université PARIS-SACLAY

A unified framework for label shift quantification

Cadre unifié pour la quantification label shift

Thèse de doctorat de l'Université Paris-Saclay

École doctorale de Mathématique Hadamard n° 574 (EDMH)
Spécialité de doctorat: Mathématiques appliquées
Graduate School : Mathématiques. Référent : Faculté des sciences d'Orsay

Thèse préparée dans l'unité de recherche **Laboratoire de mathématiques d'Orsay (Université Paris-Saclay, CNRS)** sous la direction de **Gilles Blanchard**, professeur, et le co-encadrement de **Marc Glisse**, chargé de recherche.

Thèse présentée et soutenue à Paris-Saclay, le 1er octobre 2024, par

Bastien DUSSAP

Composition du jury

Membres du jury avec voix délibérative

Christine Keribin Professeure, Université Paris-Saclay	Présidente
Pierre Alquier Professeur, ESSEC Business School de Singapore	Examinateur & Rapporteur
Alejandro Moreo Fernández Istituto di Scienza e Tecnologie dell'Informazione di Pisa	Rapporteur
Claire Boyer Professeure, Université Paris-Saclay	Examinatrice
Tabea Rebafka Maître de conférences, Paris-Sorbonne	Examinatrice

THESE DE DOCTORAT

NNT: 2024UPASM030

université PARIS-SACLAY | ÉCOLE DOCTORALE de mathématiques Hadamard (EDMH)

Titre: Cadre unifié pour la quantification label shift.
Mots clés: Théorie de l'apprentissage, Quantification, Kernel Mean Embedding, Label shift, Open set label shift, Cytométrie en flux.

Résumé: Il n'est pas rare qu'en classification supervisée, l'information recherchée ne soit pas de nature locale, c'est-à-dire associer à chaque point un label, mais de nature globale : obtenir les proportions des différents labels dans l'échantillon. Ce problème, que nous avons choisi de désigner sous le nom de "label shift quantification", mais qui porte aussi de nombreux autres noms dans la littérature, a vu depuis le milieu des années 2000 une multiplication des articles publiés. Cependant, ces travaux sont souvent menés en parallèle, issus de communautés dialoguant peu, ce qui a résulté en une bibliographie parsemée.

Dans ce manuscrit, nous proposons d'abord une revue de ces différents travaux avec un double objectif : d'une part, créer un pont entre ces communautés en présentant des résultats issus des différents domaines de recherche, et d'autre part, resituer la suite des travaux menés dans leur contexte, notamment en s'intéressant aux efforts d'unification des méthodes. Dans un second temps, nous proposons un cadre unifiant plusieurs méthodes classiques de la littérature basées sur des vectorisations par moyenne. Nous étudions les garanties théoriques de ces méthodes et montrons des résultats de robustesse lorsque l'hypothèse centrale de label shift n'est plus vérifiée. Nous proposons aussi une extension de ce travail, centrée sur des vectorisations par noyaux, utilisant l'information de la covariance et non plus seulement la moyenne. Enfin, dans un troisième temps, nous nous intéressons à l'utilisation d'une vectorisation particulière basée sur les Random Fourier Features dans des applications en cytométrie en flux.

Title: A unified framework for label shift quantification.
Keywords: Learning Theory, Quantification, Kernel Mean Embedding, Label Shift, Open set label shift, Flow cytometry.

Abstract:. In supervised classification, it is not uncommon that the information sought is not local, meaning the label associated to each data point, but global: obtaining the proportions of the different labels within the sample directly. This problem, which we have chosen to refer to as "label shift quantification" but which is also known by many other names in the literature, has seen a proliferation of publications since the mid-2000s. However, these works often proceed in parallel, coming from communities with limited dialogue, resulting in a scattered bibliography.

In this manuscript, we first provide an overview of these diverse works with a twofold aim: first, to bridge the gap between these communities by presenting results from different research areas, and on the other hand, contextualise the subsequent work, particularly focusing on efforts to unify methods. Second, we propose a framework that unifies several classical methods from the literature based on mean vectorisations. We examine the theoretical guarantees of these methods and demonstrate their robustness when the central assumption of label shift is violated. We also extend this work by focusing on kernel-based vectorisations using covariance information rather than just the mean. Finally, we explore the use of a specific vectorisation based on Random Fourier Features in applications related to flow cytometry.

Remerciements

Je tiens tout d'abord à exprimer ma profonde gratitude aux membres du jury et aux rapporteurs pour avoir accepté de participer à cette soutenance. Je remercie chaleureusement Claire Boyer, Christine Keribin, et Tabea Rebafka, ainsi que Pierre Alquier et Alejandro Moreo, pour le temps et l'énergie qu'ils ont consacrés à l'évaluation de mon manuscrit.

Je remercie chaleureusement mes directeurs de thèse pour m'avoir permis de réaliser ce travail, qui n'aurait jamais pu voir le jour sans leur implication et leur encadrement sans faille. Gilles, cette thèse te doit énormément. Merci de m'avoir d'abord accueilli en tant qu'étudiant pour un projet de M2, puis en tant que stagiaire, et enfin en tant que doctorant. Ta bienveillance, ton soutien, et tes nombreux conseils pertinents ont été inestimables. Marc, je te remercie pour nos nombreuses discussions qui ont toujours su m'aiguiller dans la bonne direction, ainsi que pour ta bienveillance et le temps que tu m'as toujours accordé. Toutes les thèses ne se déroulent pas aussi bien (mon bureau peut malheureusement en témoigner), et je me sens chanceux d'avoir pu faire un bout de chemin scientifique avec vous deux.

Je remercie chaleureusement tous les membres du Laboratoire de Mathématiques d'Orsay. Une page de six années se tourne aujourd'hui pour moi, puisque j'ai eu la chance d'y étudier pendant trois ans avant d'y poursuivre ma thèse pour trois années supplémentaires. Je tiens à exprimer ma gratitude envers tous les professeurs que j'ai eus durant mes études. Sans eux, cette thèse n'aurait évidemment pas pu voir le jour. Merci de m'avoir transmis la *rigueur intellectuelle, la réflexivité éthique et le respect des principes de l'intégrité scientifique*. Je remercie bien sûr tous les membres du laboratoire, passés et présents de l'équipe de Probabilités et Statistiques, ainsi que le personnel administratif et informatique, pour leur accueil chaleureux et bienveillant, et pour l'aide précieuse qu'ils m'ont accordée. Je souhaite également remercier Perrine Lacroix pour m'avoir encouragé à m'occuper du séminaire de statistiques des M2 au coté de Wojciech.

J'ai eu la chance d'être membre de l'équipe Datashape de l'Inria. Je tiens à remercier chaleureusement tous les membres de l'équipe, tant à Orsay qu'à Sophia Antipolis, pour leur accueil. Je remercie tout particulièrement Marc, Vincent, et Fred pour les nombreux CESFO partagés. Les plus taquins d'entre vous (Zacharie) remarqueront sans doute que mes contribu-

tions à l'analyse topologique des données sont assez *modestes* dans ce manuscrit. Cependant, faire partie de cette équipe a été une expérience extrêmement enrichissante, qui m'a permis de découvrir un domaine des mathématiques qui m'était peu familier, à l'intersection des statistiques et de la topologie algébrique. Je regrette de ne pas avoir pu explorer davantage ce domaine pendant ma thèse. Je garde un excellent souvenir des séminaires d'équipe à Porquerolles (et pas seulement parce qu'ils avaient lieu à Porquerolles). Je ne peux conclure ces remerciements sans exprimer ma gratitude envers notre assistante, Aïssatou, pour son travail remarquable.

Ce travail n'aurait pas été possible sans la collaboration avec Metafora. Je remercie chaleureusement l'ensemble de l'équipe biologie, et en particulier Nupur pour les échanges constructifs sur les données de cytométrie utilisées, ainsi que Christian pour ses explications sur le myélome multiple. Je tiens également à exprimer ma gratitude envers toute l'équipe Data Science pour avoir rendu mon expérience si agréable : Hamza, Heng, Nafise (sans qui rien n'aurait été possible, évidemment), Pierre-André, Weifeng, et un merci tout particulier à Baptiste, qui a assuré le rôle de troisième directeur. Merci pour ton implication, ta bienveillance et ton expertise. Enfin, je remercie Vincent, à l'origine de cette collaboration, pour sa confiance et l'opportunité de travailler sur des projets concrets. Je suis heureux de pouvoir continuer l'aventure avec Metafora dans les années à venir.

Un grand merci à tous les (ex-)doctorants du LMO et de Datashape avec qui j'ai partagé l'aventure, parfois (souvent ?) difficile de la thèse, et qui ont contribué à créer une ambiance de travail enrichissante. Merci à la *génération dorée* de Datashape : Alex, Alexandre, Étienne, Jérémie, Olympio, Louis, Vadim, et Wojciech, qui m'ont chaleureusement accueilli au sein de l'équipe. Du côté du LMO, merci à Jean-Baptiste, avec qui j'ai évolué en parallèle tout au long de la thèse. Je passe le flambeau à Ibrahim et à mon petit frère de thèse, Romain, pour l'organisation de l'excellent séminaire des étudiants du M2 de statistiques, et je vous souhaite plein de succès dans la poursuite de vos thèses. Enfin, merci à Alexandre, Laure, Samy, et Wojciech d'avoir partagé un bureau avec moi. Je ne peux terminer ces remerciements, sans mentionner Badr-Eddine avec qui j'ai eu la chance de collaborer dans l'écriture d'un article.

Merci à mes amis de toujours : Arthur (et maintenant Maëva), Arthur2, Aurélie, Flavien, Jocelyn, et Raphaël, pour toutes les parties de jeux de société partagées ainsi que les séances de jeux de rôle qui ont rythmé ces trois dernières années. Merci également à Thibault, dont je suis impatient de lire la thèse (et de ne rien comprendre).

Je ne peux terminer ces remerciements sans mentionner ma famille, mes soutiens de toujours : mon frère, dont j'ai suivi les traces ; ma sœur, pour tous les moments partagés à jouer à la Switch ; et mes parents, pour leur amour inconditionnel et leur soutien indéfectible tout au long de ce parcours.

Contents

Chapter 1

Introduction

1.1 Introduction en Français

Ce travail est le fruit de la collaboration entre trois entités. D'une part, sur le plan académique, de l'université Paris-Saclay et plus particulièrement de l'équipe *Probabilités et statistiques* du Laboratoire de Mathématiques d'Orsay ainsi que de l'équipe *DataShape* du centre INRIA de Saclay et d'autre part, sur le plan industriel, de l'entreprise Metafora. Il se place dans le cadre du développement du logiciel **METAflow**, logiciel d'aide à l'analyse de données de Cytométrie.

Dans ce cadre, notre travail porte sur un problème d'estimation de proportions supervisé, portant de nombreux noms dans la littérature mais que nous désignerons par **Label Shift Quantification** ou plus simplement **Quantification**.

Dans ce problème nous supposons avoir accès à plusieurs échantillons de référence, chacun ayant été tiré selon une distribution différente. Un nouvel échantillon est alors tiré que l'on suppose être issu d'un mélange de ces distributions de références. L'objectif est alors d'estimer les proportions de ce mélange.

Pour analyser les données de cytométrie, METAflow utilise un algorithme de clustering hiérarchique. L'algorithme retourne un arbre, où chaque nœud représente un sous-ensemble de points, i.e. un cluster. L'objectif de cette thèse est de faire de la quantification sur l'ensemble des nœuds de l'arbre. Dû à ces contraintes applicatives, nous avons opté pour une approche par vectorisation de distribution, en particulier en utilisant les Random Fourier Features (RFF). La deuxième étape de notre travail a été d'étudier comment ces vectorisations par RFF pouvaient aider dans le travail d'analyse des données de cytométrie au-delà du cadre de la quantification.

Table des matières

1.1.1 Label Shift Quantification

Quantification est un terme ambigu qui peut être utilisé pour désigner une grande variété de problèmes en mathématiques et en sciences en général. Tout au long de ce manuscrit, nous appellerons quantification le problème consistant à **quantifier le changement de proportion entre une distribution de référence appelée source et une nouvelle distribution appelée cible**.

La source et la cible s'écrivent comme un mélange de c distributions $(\mathbb{P}_i)_{i=1}^c$, correspondant à différentes classes, mais les poids des mélanges changent entre la source et la cible. On trouve des variantes de ce problème sous de nombreux noms dans la littérature : *class prior estimation*, *class prior change*, *prevalence estimation*, *class ratio estimation*, *learning to quantify*, *unfolding*, *domain adaptation under label shift*, *label shift adaptation*, et probablement d'autres noms que nous ne connaissons pas encore. En résulte une bibliographie très segmentée, qui peut être divisée en trois catégories en fonction de l'objectif :

- Le premier est *détecter* si le mélange de la source et de la cible est différent. On se situe alors dans la littérature des tests statistiques.
- Le second consiste à *corriger* une méthode développée sur la source et dont l'efficacité a été prouvée sur la cible. Par exemple, si on entraîne un classificateur sur la distribution source pour classer les différentes classes $(\mathbb{P}_i)_{i=1}^c$, fonctionnera-t-il sur la cible ? Et si ce n'est pas le cas, comment le corriger ? On se situe alors dans la littérature machine learning et plus précisément au problème d'adaptation de domaine.
- Le dernier problème est de *quantifier* le changement de poids, ou en d'autres termes, pouvons-nous estimer le nouveau poids du mélange dans la cible ?

Dans ce manuscrit, nous nous concentrons sur ce dernier point. La littérature la plus concernée par cette question l'appelle **Quantification**. Ce n'est probablement pas le terme le plus approprié, car il est souvent ambigu et laisse l'observateur avec une question : *Qu'est-ce qui est quantifié en quantification ?* C'est pourquoi nous préférons le terme **label shift quantification**. Nous donnerons une définition formelle de **label shift** dans le Chapitre 2 (Définition 2.1), mais pour résumer simplement, le label shift est exactement ce que nous cherchons à estimer : le changement de poids du mélange. La quantification n'est pas un sujet nouveau en soi, on peut par exemple retrouver des travaux en épidémiologie datant de 1966 [11] cherchant à estimer la "*prevalence*", c'est-à-dire les proportions, des classes dans un échantillon. De plus, de nombreux travaux de classification sont en réalité des problèmes de quantification dans lesquels l'information cherchée n'est pas une information locale sur l'appartenance de chaque point à une classe, mais une information globale sur les proportions des différentes classes dans l'échantillon. La classification n'est alors qu'un moyen de *quantifier*. Cependant, il a fallu attendre le début des années 2000, dans une série d'articles par Forman en 2005 [46], 2006 [47] et 2008 [48], pour que le sujet s'impose comme une problématique en soi et non comme une simple application de classification. Depuis, la problématique s'est structurée sous cette dénomination commune de "**quantification**" et de nombreux articles ont été écrits sur le sujet jusqu'au récent livre *Learning to quantify* d'Esuli et al. [32] publié en 2023, des *data challanges* ont été organisés [35, 36] et des *workshops* [15, 27, 71] sur le sujet ont eu lieu lors de grandes conférences internationales de *Machine Learning* (CIKM 2021 et ECML/PKDD 2022/2023). Cependant, le sujet n'est pas non plus devenu "grand public" au sein des communautés machine learning. On retrouve ainsi encore de nombreux articles traitant de quantification sans le nommer explicitement [49, 77].

Open Set Label Shift Quantification

Nous étudierons également un cadre plus général que le label shift, dans lequel les proportions des différentes classes dans la cible changent, alors que la distribution des classes ne change pas, et qu'une nouvelle classe, appelée la **contamination**, apparaît. Cette hypothèse a été à notre connaissance formulée pour la première fois dans l'article correspondant au Chapitre 3 sous le nom de *contaminated label shift* et en parallèle de manière indépendante par Garg et al. [51] sous le nom *Open Set Label Shift* (OSLS). Ce nom "open set" est en référence à l'*Open Set Domain Adaptation* (OSDA) aussi parfois appelé "*universal domain adaptation*" une hypothèse très générale faite en *adaptation de domaine*, dans laquelle les distributions des classes changent, les proportions des classes changent et une contamination apparaît dans la distribution cible. Les travaux de Garg et ses coauteurs cherchent à obtenir un classificateur ayant une bonne précision sur la donnée cible et non à déterminer les proportions (bien que leurs méthodes permettent d'obtenir ces proportions). Dans la littérature quantification, ce setting n'avait pas encore était traité avant les travaux que nous présentons dans le Chapitre 3.

1.1.2 Application à la Cytométrie en flux

La cytométrie en flux est une technologie qui permet une analyse rapide de cellules individuelles suspendues dans une solution saline. La machine permettant de réaliser ces mesures, le cytomètre, se compose d'éléments fluidiques, optiques et électroniques. Le système fluidique est composé de la solution saline dans laquelle les cellules, issues par exemple d'un prélèvement sanguin ou de moelle osseuse, sont suspendues. La machine pressurise le fluide et le fait converger vers un point afin que les cellules puissent être analysées une à une. Le système optique consiste en un laser pointant en direction du point de convergence des cellules qui excitera les cellules, provoquant des signaux lumineux redirigés par une série de filtres dichroïques et captés par des tubes photomultiplicateurs réglés à différentes longueurs d'onde allant des ultraviolet (355 nm) jusqu'au rouge (640 nm). Cette fluorescence émise est la résultants de deux phénomènes, d'une part la "couleur" naturelle de la cellule ou autofluorescence et d'autre part la fluorescence émise par un anticorps couplé à un fluorochrome ayant été placé par les cytométristes dans la solution saline lors de la préparation de l'échantillon. À cela, s'ajoutent deux mesures : la lumière diffusée aux petits angles (Forward Scatter, FSC) et la lumière diffusée à 90 degrés (Side Scatter – SSC) renseignant sur la taille, la forme ou encore la granularité des cellules. Enfin, le système électronique enregistre ces mesures et les sauvegarde sous la forme d'un fichier standardisé (*.fcs* pour *Flow Cytometry Standard*). Le fonctionnement d'un cytomètre est résumé en Figure 1.1.

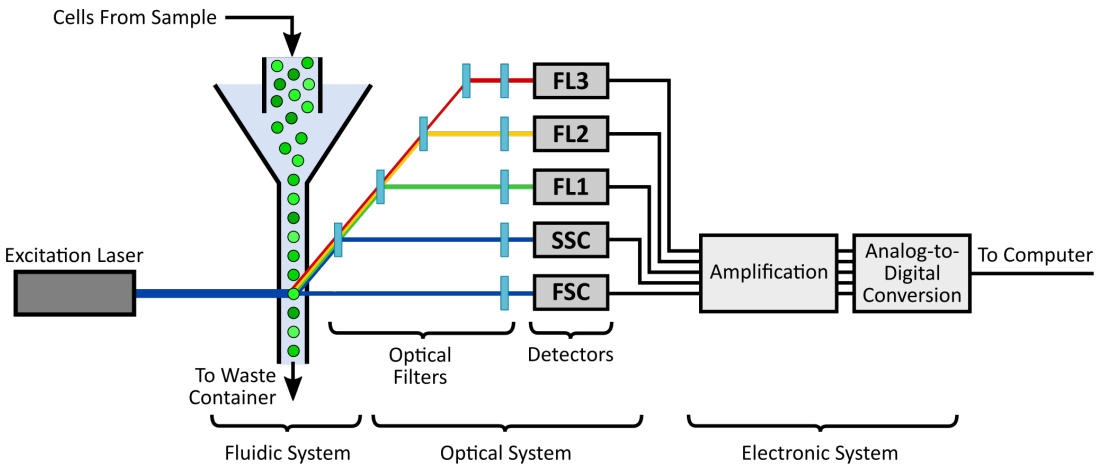


FIGURE 1.1 : Schéma d'un cytomètre en flux, illustrant les systèmes fluidiques, optiques et électroniques. Source de l'image : AAT Bioquest, Inc [8].

Du point de vue pratique, la donnée enregistrée prend la forme d'un tableau, où chaque ligne correspond à une cellule et chaque colonne correspond à un marqueur (plus précisément, à une longueur d'onde associée à un marqueur de fluorescence). La valeur d'une case représente alors l'intensité lumineuse émise par une cellule à une longueur d'onde donnée.
L'objectif des cytométristes est alors double, d'une part en amont choisir les bons marqueurs

afin de caractériser au mieux les cellules, le nombre de tubes photomultiplicateurs étant limité bien que de plus en plus conséquent. Par exemple, les cytomètres présents à Metafora possèdent entre 10 et 20 capteurs. D'autre part, en aval, les cytométristes sont responsables de l'analyse des données et plus particulièrement de l'identification de *clusters* (dans la terminologie Machine Learning) ou *gates* (dans la terminologie cytométrie). L'analyse conventionnelle des données de cytométrie se fait manuellement comme présenté en Figure 1.2.

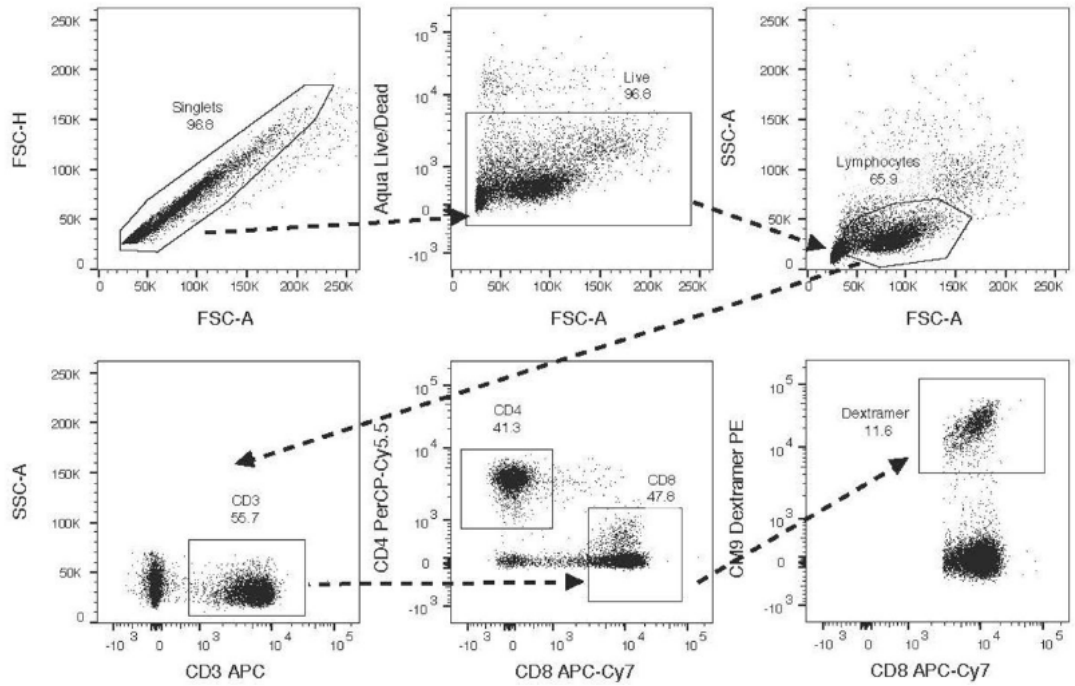


FIGURE 1.2 : Stratégie de *gating* issue de l'article de McKinnon et al. [81]. Les *gates* sont une à une dessinées en regardant la donnée au travers de 2 marqueurs bien choisis, filtrant ainsi la donnée jusqu'à ne laisser que la donnée d'intérêt. Dans cet exemple, les singlets (i.e. cellules étant passé seules devant le laser) sont d'abord sélectionnés puis les cellules vivantes, puis les lymphocytes, puis les lymphocytes exprimant le complexe protéique CD3, les lymphocytes exprimant le complexe CD8 et enfin la population d'intérêt.
Cet exemple montre bien deux limites de ce type d'analyse : l'analyse est lente (6 étapes pour arriver à la bonne population) et les gates sont dessinées manuellement par le ou la cytométristes, incorporant une part d'arbitraire dans l'analyse.

Pour pallier les défauts inhérents à l'analyse manuelle, un nombre croissant d'algorithmes, de papier ou de logiciels ont été proposés pour automatiser l'analyse des données de cytométrie, voir par exemple Cheung et al. [23]. Metafora propose **METAflow**, un logiciel utilisant principalement un algorithme de *clustering hiérarchique*, basé sur une estimation de densité

et la persistance homologique [20], permettant un *gating* automatique tout en laissant à l'utilisateur le contrôle en choisissant les branches de l'arbre de clustering à explorer.

Dans le Chapitre 5, on cherchera à estimer la proportion de certaines cellules d'intérêt, c'est-à-dire à faire de la **quantification**, dans chacun des nœuds de l'arbre. Cela induit deux contraintes sur l'algorithme que l'on recherche. D'une part, puisque les cellules d'intérêt ne représentent pas l'ensemble des cellules d'un jeu de données (parfois, cela ne représente même qu'un ensemble très faible), on se place dans le cadre de l'open set label shift. D'autre part, du fait de la structure d'arbre, on doit pouvoir résoudre le problème sur chacun des nœud sans que la complexité algorithmique ne soit trop élevée. En deuxième objectif, on regardera si les vectorisations par Random Fourier Features peuvent être utilisées pour identifier le label sur les nœuds de l'arbre directement.

1.1.3 Vectorisation de distribution

L'approche que nous allons poursuivre pendant toute cette thèse est l'utilisation de **vectorisation moyenne** (Mean Embedding).

Mathématiquement, supposons que nous avons une certaine distribution $\mathbb{P}$ sur l'espace des données $\mathcal{X}$ et soit ϕ une fonction de $\mathcal{X} \to \mathbb{R}^D$. La *vectorisation moyenne* de $\mathbb{P}$ par la fonction ϕ, que l'on notera $\Phi(\mathbb{P})$, est alors la moyenne de ϕ sous $\mathbb{P}$, i.e. $\Phi(\mathbb{P}) = \mathbb{E}_{\mathbb{P}}[\phi(X)]$. La propriété intéressante de cette fonction est qu'elle est linaire en la distribution. Autrement dit, si la distribution cible $\mathbb{Q}$ s'écrit comme un mélange de distributions de référence $\mathbb{P}_i$: $\mathbb{Q} = \sum \alpha_i \mathbb{P}_i$ alors $\Phi(\mathbb{Q}) = \sum \alpha_i \Phi(\mathbb{P}_i)$.

Une façon de résoudre le problème de quantification est alors d'utiliser les données d'entraînement pour estimer les $\Phi(\mathbb{P}_i)$ et d'utiliser la données de test pour estimer $\Phi(\mathbb{Q})$, pour ensuite chercher les proportions $\hat{\alpha}$ tel que $\sum \hat{\alpha}_i \Phi(\mathbb{P})$ est le plus proche possible de $\Phi(\mathbb{Q})$. La distance la plus naturelle à utiliser est la distance L_2 et les travaux théoriques que nous présentons sont obtenus avec elle, mais en pratique d'autres distances pourraient être utilisées.

Remarque. J'ai utilisé ici le terme de *vectorisation*, car j'ai défini ma fonction ϕ à valeur dans $\mathbb{R}^D$. En pratique, c'est cela que l'on fera tout au long de la thèse. Cependant, l'ensemble des résultats peuvent aussi s'appliquer si ϕ est à valeur dans un espace de Hilbert quelconque $\mathcal{H}$, par simplicité, j'utiliserai tout au long de cette introduction le terme vectorisation et ne ferai pas la différence entre $\mathbb{R}^D$ et $\mathcal{H}$ tandis que dans le corps de la thèse, j'utiliserai le terme *embedding* et ne parlerai que de $\mathcal{H}$ pour que les résultats soient les plus généraux possibles.

La structure d'arbre rend ces méthodes basées sur des *vectorisations par moyenne* particulièrement intéressantes. En effet, puisque la fonction Φ est linéaire pour des distributions, cela entraîne que la vectorisation d'un nœud est égale à une moyenne pondérée des vectorisations de ses enfants. Ainsi, il suffit de calculer la vectorisation des feuilles puis de faire remonter l'information vers la cime de l'arbre. Les proportions de chaque nœud se calculent

par la résolution d'un problème QP en basse dimension dont le temps de calcul est négligeable par rapport aux calculs des vectorisations.

N'importe quelle fonction ϕ peut être utilisée, par exemple, on parlera de méthodes utilisant la sortie d'un classificateur. ϕ n'est alors techniquement pas à valeur dans $\mathbb{R}^D$, mais soit dans le simplex de dimension $(c-1)$: $\Delta^c \coloneqq \{x \in \mathbb{R}^c \colon x_i \geq 0,\ \sum x_i = 1\}$ si le classificateur renvoie une probabilité, soit dans $\{0,1\}^c$ si le classificateur ne renvoie que la classe prédite. Autre exemple, on peut vectoriser en utilisant une couche intermédiaire d'un réseau de neurones. Les réseaux sont des algorithmes qui apprennent à bien séparer des données (souvent de très hautes dimensions comme des images) dans une représentation intermédiaire, puis à classifier (par un classificateur linaire) dans un second temps.

Une autre forme de vectorisation dont on reparlera tout au long de ce manuscrit, est le **Kernel Mean Embedding**. Rappelons d'abord la définition d'un noyau.

Définition. *(Noyau semi-défini).* Une fonction $k : X \times X \to \mathbb{R}$ est un noyau semi-défini positif si elle est symétrique, c'est-à-dire $k(x,y) = k(y,x)$, et si la matrice de Gram est positive :

$$\sum_{i,j=1}^{n} c_i c_j k(x_i, x_j) \geq 0, \tag{1.1}$$

pour tout $n \in \mathbb{N}$, tout choix de $x_1, \ldots, x_n \in X$ et tout $c_1, \ldots, c_n \in \mathbb{R}$. On dit qu'il est défini positif si l'égalité dans (1.1) implique $c_1 = c_2 = \ldots = c_n = 0$.

On peut montrer que pour tout noyau semi-défini positif k (que l'on appellera simplement noyau dans la suite) il existe un unique espace de fonctions $\mathcal{H}_k$ de $X \to \mathbb{R}$, appelé le RKHS (espace de Hilbert à noyau reproduisant) associé à k et une unique fonction Φ_k dans cet espace telle que $k(x,y) = \langle \Phi_k(x), \Phi_k(y) \rangle$, pour tout x, y. Le Kernel Mean Embedding (KME) d'une distribution associé au noyau k est alors défini comme :

$$\Phi_k(\mathbb{P}) \coloneqq \int_X \Phi_k(x) d\mathbb{P}(x) = \mathbb{E}_{\mathbb{P}}[\Phi_k(X)] \in \mathcal{H}_k. \tag{1.2}$$

Cet objet est plus complexe que les vectorisations que l'on a présentées précédemment car cette fois, il est à valeur dans un espace de Hilbert de dimension infinie, et l'intégrale (1.2) est définie (si elle existe) comme une intégrale de Bochner.

La distance L_2 entre la vectorisation de deux distribution est alors appelé le *Maximum Mean Discrepancy*, introduite par Gretton et al. [61] pour des tests statistiques.

Cependant, le MMD, comme toutes les méthodes à noyaux, souffre d'un problème computationnel. On ne peut accéder au MMD entre deux distributions que par le biais des produits scalaires en utilisant l'astuce du noyau (*kernel trick*) :

$$\langle \Phi_k(\mathbb{P}), \Phi_k(\mathbb{Q}) \rangle = \mathbb{E}_{\mathbb{P},\mathbb{Q}}[k(X,Y)].$$

Calculer ce produit scalaire est de complexité **quadratique** avec le nombre de points. En effet, $\Phi_k(\mathbb{P})$ même quand $\mathbb{P}$ est une distribution empirique, ne peut être calculée et stockée

facilement sur un ordinateur, car il s'agit d'une fonction. Ces deux éléments sont chacun rédhibitoires pour l'application à la cytométrie en flux.

1.1.4 Random Fourier Features

Pour surpasser cette problématique, nous utiliserons les **Random Fourier Features** (**RFF**), introduit par Rahimi et al. [98], une méthode d'approximation du noyau.

Les Random Fourier Features sont basés sur le théorème de Bochner :

Theorem 1.1. *Une fonction continue φ sur $\mathbb{R}^D$ définit un noyau semi-définie positif $k(x, y) = \varphi(x - y)$ si et seulement si φ est la transformée de Fourier d'une mesure non-négative.*

Un corollaire direct de ce résultat est que tout noyau invariant par translation continu k, associé à une fonction φ, i.e. $k(x, y) = \varphi(x - y)$ est la transformée de Fourier d'une mesure non-négative que nous dénotons Λ_k et pour laquelle $\Lambda_k(\mathbb{R}^D) = \varphi(0)$. Par conséquent si on normalise le noyau par $\varphi(0)$, Λ_k est alors une mesure de probabilité appelée la *distribution spectrale* de k.

Le noyau peut alors se réécrire comme suit :

$$k(x, y) = \mathbb{E}_{\omega\sim\Lambda_k}[e^{i\omega^T(x-y)}] = \mathbb{E}_{\omega\sim\Lambda_k}[\cos\big(\omega^T(x - y)\big)].$$

Par Monte-Carlo, en utilisant un échantillon $(\omega_i)_{i=1}^{D/2}$ iid de Λ_k, la vectorisation $z\colon \mathcal{X} \to \mathbb{R}^D$ définie par

$$z(x) = \sqrt{\frac{2}{D}}\big[\cos(\omega_i^T x),\ \sin(\omega_i^T x)\big]_{i=1}^{D/2}, \tag{1.3}$$

est telle que : $k(x, y) = \mathbb{E}\big[z(x)^T z(y)\big]$, où l'espérance est prise par rapport à $(\omega_i)_{i=1}^{D/2}$. Dans la pratique, une autre vectorisation est couramment utilisée :

$$\tilde{z}(x) = \sqrt{\frac{2}{D}}\big[\cos(\omega_i^T x + b_i)\big]_{i=1}^{D}, \tag{1.4}$$

où b_i sont des échantillons iid de la distribution uniforme sur $[0, 2\pi]$. Voir Gundersen [64] pour le calcul détaillé. Même si cette vectorisation est populaire dans la pratique, nous aimerions souligner que la deuxième version donne les pires résultats en termes de variance et de limite supérieure pour le noyau Gaussien [113].

Si la vectorisation est à valeur dans $\mathbb{R}^D$, la complexité du calcul des vectorisations est alors linéaire en le nombre de points et linéaire en le nombre de random features utilisé, c'est à dire linéaire en D. De plus, le calcul de la vectorisation d'un nuage de points se réduit à une multiplication matricielle, pour laquelle les GPU sont bien adaptés. Pour gérer les débordements de mémoire sur les GPU, nous nous appuyons sur la libraire Python *PyKeops* [19].

Succinctement, cette méthode permet donc de définir une fonction $z\colon X \to \mathbb{R}^D$ (avec D un hyper-paramètre choisi) de telle sorte que pour tout $x, y : k(x, y) \approx z(x)^T z(y)$. Cette méthode a trois avantages (par rapport à d'autres méthodes d'approximations du noyau comme pas exemple Nystrom). Premièrement, la vectorisation $z(\mathbb{P})$ est stockable sur un ordinateur, car il s'agit d'un vecteur de $\mathbb{R}^D$. Deuxièmement, le MMD entre deux distributions ce calcule en temps linéaire et non quadratique. Et enfin, avec cette fonction z, on se situe toujours dans le cadre des vectorisations présentées plus tôt. L'ensemble des théorèmes de ce manuscrit s'applique directement avec z sans avoir à modifier les preuves.
Dans les Chapitres 3 et 4 nous verrons cette fonction z comme une approximation du noyau Gaussien et donc comme une façon d'accélérer le temps de calcul, tandis que dans le Chapitre 5 la fonction z sera vue comme une vectorisation en soi, c'est-à-dire comme un moyen de caractériser une distribution sous la forme d'un vecteur utilisable dans diverses applications de **cytométrie en flux**.

1.1.5 Contributions

La première contribution (Chapitre 2) de cette thèse est une revue de littérature sur la quantification *Label shift*. L'objectif de ce chapitre est de présenter et de faire connaître au lecteur ce domaine de recherche encore méconnu de la littérature *Machine Learning*.

Ce chapitre présente une vue générale (mais non-exhaustive) des principales méthodes de la littérature, ainsi qu'une présentation des différentes approches proposées dans la littérature pour unifier ces méthodes sous un même cadre d'analyse. Par exemple Firat [44] qui représente différentes méthodes sous la forme d'un problème de régression sous contraintes afin de présenter des extensions des algorithmes du cas binaire au cas multiclasses. Cette même représentation est au cœur du package python Qunfold par Bunse [14]. Dans une approche différente, Garg et al. [52] proposer d'unifier sous une même analyse les deux algorithmes les plus utilisés en pratique (Adjusted Classify and count et Maximum Likelihood Quantification que l'on présente dans le chapitre) afin de justifier de la supériorité pratique de l'un des deux.

Ce chapitre se conclut par une présentation des protocoles de test proposé dans la littérature ainsi que d'une présentation et une extension des travaux de Sebastiani [104] portant sur le choix de la métrique à utiliser.

La deuxième contribution (Chapitre 3) est l'étude théorique et pratique de l'ensemble des méthodes basées sur les vectorisations par moyenne présentées en Section 1.1.3. Cette catégorie de méthodes regroupe plusieurs des méthodes présentées dans le Chapitre 2. La contribution de ce chapitre est double, d'une part dans le cadre *Label shift* classique on obtient une borne empirique de l'erreur d'approximation améliorant les bornes ayant été obtenues pour les différentes méthodes tout en étant plus générale car s'appliquant à toutes les méthodes de la catégorie, voir Théorème 3.1. C'est aussi le premier travail à notre connaissance qui propose d'unifier différentes méthodes de la littérature afin d'obtenir un théorème de convergence s'appliquant à toutes les méthodes. Ce résultat permet aussi d'avoir un critère

de choix de fonction de vectorisation calculable empiriquement à partir du jeu de données labellisé d'entraînement, utilisable par exemple pour choisir le meilleur paramètre de *bandwidth* lorsque la vectorisation choisie est basée sur le noyau Gaussien, le choix de *bandwidth* étant un problème classique des méthodes à noyaux. Dans le cadre *open set label shift*, on obtient là aussi une borne empirique de l'erreur d'approximation, voir le Théorème 3.3 et son Corollaire 3.2, inédite, car aucune analyse de l'erreur d'un quantifier n'a encore été proposée dans un cadre *open set label shift*. Ce résultat met en évidence une robustesse à la contamination sous une hypothèse d'orthogonalité des vectorisations du bruit avec les vectorisations des autres classes. Sous cette hypothèse, l'erreur convergera vers zéro, (plus lentement que dans le cas *label shift* cependant) tandis que sans cette hypothèse, l'erreur convergera vers un biais égal à la norme de la composante orthogonale aux vectorisations des autres classes du bruit. Pour pouvoir faire de la quantification sous *open set label shift* il faut donc une vectorisation qui assure une orthogonalité avec le bruit. La méthode utilisant un noyau Gaussien est la plus adaptée à ce type de problématique car l'orthogonalité des vectorisations est assurée à partir du moment où les distributions sont bien séparées. On met en évidence la robustesse de la vectorisation gaussienne à du bruit sur des données simulées et sur des données de cytométrie en flux issues de Metafora. Une dernière contribution, mineure pour ce chapitre, mais importante pour le Chapitre 5, est l'utilisation des Random Fourier Features [98] comme méthode pour vectoriser des distributions.

La troisième contribution (Chapitre 4) propose d'utiliser l'information contenue dans la variance et non plus simplement la moyenne des vectorisations. Il s'agit d'une extension du précédent chapitre pour de la quantification *label shift*, bien que les résultats obtenus sur la robustesse au cadre *open set label shift* soient aussi applicables à ces méthodes.
Dans la littérature des tests MMD, c'est un fait désormais bien établi (par exemple dans un article récent de Hagrass et al. [66] ou dans des articles plus vieux de Harchaoui et al. [31, 67]) que les méthodes n'utilisant que l'information contenue dans la moyenne ne sont pas minimax. En s'inspirant entre autres de leurs travaux, on propose dans ce chapitre de "régulariser" la distance utilisée dans le Chapitre 3 par une certaine matrice M. Plus précisément, là où dans le Chapitre 3 la distance utilisée était la L_2, ici on se propose d'utiliser une distance de type Mahalanobis.

En utilisant un théorème de Bernstein vectoriel, variante d'un résultat obtenue par Wolfer et al. [127], on obtient une autre borne sur l'erreur d'approximation faisant cette fois-ci intervenir la variance, voir Théorème 4.2, et on obtient alors un critère de choix de matrice M qu'on relie au *kernel Fisher Discriminant Analysis*, un algorithme de réduction de dimension.

Après une discussion sur la forme de cette matrice optimal, ainsi qu'une explication sur l'estimation pratique de la matrice, on test les nouvelles méthodes sur les même donnée que celle du chapitre précédent : données simulées et données de cytométrie en flux issues de Metafora.

La quatrième contribution (Chapitre 5) est une étude de cas sur l'usage de la vectorisation par Random Fourier Features pour la cytométrie en flux. On s'intéresse à un jeu de données réelles contenant l'analyse de moelle osseuse de 29 patients atteints de deux pathologies : le *myélome multiple* et la *Gammapathies monoclonales de signification indéterminée* (MGUS). Dans un premier temps, on montre l'intérêt de cette vectorisation pour une analyse exploratoire d'un jeu de donnée de Cytométrie. En particulier, on met en évidence l'échec de la stratégie de gating proposé pour identifier les *T cells* ainsi que les *Plasmocytes* de l'un des patients. Dans un second temps, on s'intéresse à l'usage des RFF pour 2 tâches : identifier dans l'arbre de clustering hiérarchique issue de METAflow le cluster le plus probable d'être les cellules d'intérêt et d'autre part la quantification des proportions dans chacun des nœuds de l'arbres.

Enfin, le manuscrit se termine par l'Annexe A. Celle-ci contient trois résultats de concentration de norme dans des espaces de Hilbert, utilisés au cœur de toutes les preuves de ce manuscrit. Le premier théorème est un Hoeffding vectoriel : un résultat classique basé sur l'inégalité de McDiarmid. On complète cette section en présentant un petit historique des résultats de ce type dans la littérature MMD. Le deuxième résultat de concentration est une version vectorielle de Bernstein (Théorème A.3) remontant aux travaux de Pinelis [93] dans des espaces de Banach et Yurinsky [129] dans des espaces de Hilbert. La preuve, que l'on propose dans ce manuscrit est tirée des travaux de Wolfer et al. [127] avec deux petites différences. D'une part la correction d'une typo (un $\log(1/\delta)$ qui aurait du être un $\log(2/\delta)$) et d'autre part, le point de départ de leur théorème est basé sur le résultat de Yurinsky alors que nous utilisons celui de Pinelis. La différence est cependant minime, puisque qu'elle n'impact que la constance devant le terme en $\mathcal{O}(1/n)$ de la borne : nous avons $2/3$ alors qu'ils avaient $4/3$. Enfin, pour être complet, nous présentons un troisième théorème, une inégalité de Bennett vectorielle, basé cette fois sur les travaux de Smale et al. [108] qui eux même basent leurs résultats sur un théorème de Pinelis [93]. La différence porte encore sur la constante devant le terme en $\mathcal{O}(1/n)$ avec cette fois un 4 au lieu d'un $2/3$.

1.2 Introduction in English

This work is the result of a collaboration between three entities. On the academic level, the University Paris-Saclay and more particularly the *Probabilities and Statistics* team of the *Laboratoire de Mathématiques d'Orsay*, the *DataShape* team of the *INRIA* centre in Saclay and, on an industrial level, the company Metafora. It is part of the development of **METAflow**, a software to assist in the analysis of flow cytometry data, a type of data that captures multi-parameter information about individual cells.

In this context, our work concerns a problem of supervised estimation of proportions, which has many names in the literature but which we will refer to as **Label Shift Quantification** or more simply **Quantification**.

In this problem we assume that we have access to several reference samples, each drawn according to a different distribution. A new sample is then drawn from a mixture of these reference distributions and the objective is to estimate the proportions of this mixture.

To analyse flow cytometry data, METAflow performs hierarchical clustering. The output of the software is a tree where each node represents a subset of points, i.e. a cluster. The aim of this thesis is to perform Quantification on each node of the tree. Due to these application constraints, we chose to use a distribution vectorisation approach, specifically using Random Fourier Features (RFF). The second stage of our work was to investigate how these RFF vectorisations could help in the analysis of cytometry data beyond quantification.

Contents

1.2.1 Label Shift Quantification

Quantification is an ambiguous term that can be used to refer to a wide variety of problems in mathematics and science in general. Throughout this manuscript, we will refer to quantification as the problem of **quantifying the change in proportion between a reference distribution called the source and a new distribution called the target**.

The source and the target are mixtures of c distributions $(\mathbb{P}_i)_{i=1}^c$, corresponding to different classes, but the weights of the mixtures change between the source and the target. Variations of this problem can be found under many names in the literature: *class prior estimation*, *class prior change*, *prevalence estimation*, *class ratio estimation*, *learning to quantify*, *unfolding*, *domain adaptation under label shift*, *label shift adaptation*, and probably other names that we do not yet know. The result is a highly segmented bibliography, which can be divided into three categories depending on the objective:

- The first objective is to *detect* if the weights of the source and the target are different. It is therefore a statistical testing problem.
- The second is to *correct* a method developed on the source and whose effectiveness has been proven, on the target. For example, if you train a classifier on the source distribution to classify the different classes $(\mathbb{P}_i)_{i=1}^c$, will it work on the target? And if it does not, how can it be corrected? This brings us back to the machine learning literature and, more specifically, to the domain adaptation problem.
- The final problem is to *quantify* the change in weight, or in other words: Can we estimate the new weight of the mixture in the target?

In this manuscript, we focus on this last point. The literature most concerned with this question calls it **quantification**. This is probably not the most appropriate term, as it is often ambiguous and leaves the observer with a question: *What is quantified in quantification?* This is why we prefer the term **label shift quantification**. We will give a formal definition of **label shift** in Chapter 2 (Definition 2.1), but to summarise simply, the label shift is exactly what we are trying to estimate: the change in weight of the mixture. Quantification is not a new subject in itself; for example, we can find work in epidemiology dating back to 1966 [11] seeking to estimate the '*prevalence*', i.e. the proportions of the classes in a sample. In addition, many classification studies are in fact quantification problems because the information sought is not a local information on whether each point belongs to a class, but a global information on the proportions of the different classes in the sample. Classification is then simply a means of quantifying.

It was not until the early 2000s, in a series of articles by Forman in 2005 [46], 2006 [47] and 2008 [48], that the subject emerged as a problem in its own right and not simply as an application of classification. Since then, the issue has been structured under the common name of "**quantification**" and numerous articles have been written on the subject, culminating in the recent book *Learning to quantify* by Esuli et al. [32] published in 2023, *data challenges* have been organised [35, 36] and *workshops* [15, 27, 71] on the subject have taken place at major international *Machine Learning* conferences (CIKM 2021 and ECML/PKDD 2022 and 2023). Even so, the subject has not gone mainstream in the machine learning community. There are still many articles dealing with quantification without explicitly naming it [49, 77].

Open Set Label Shift Quantification

We will also investigate a more general setting than label shift, where the proportions of the different classes in the target change, while the class distribution does not, and a new class, called the **contamination** class, appears. To our knowledge, this hypothesis was formulated for the first time in the article corresponding to Chapter 3 under the name *contaminated label shift* and in parallel and independently by Garg et al. [51] under the name *Open Set Label Shift* (OSLS). This name "open set" is in reference to *Open Set Domain Adaptation* (OSDA) also sometimes called "*universal domain adaptation*" a very general assumption made in *domain adaptation*, in which class distributions change, class proportions change, and contamination appears in the target distribution. The work of Garg and his co-authors seeks to obtain a classifier with good precision on the target data and not to determine the proportions. In the quantification literature, this setting had not yet been addressed before the work we present in Chapter 3.

1.2.2 Application to Flow Cytometry

Flow cytometry is a technology that enables rapid analysis of individual cells suspended in a saline solution. The machine used to perform these measurements, the cytometer, is made up

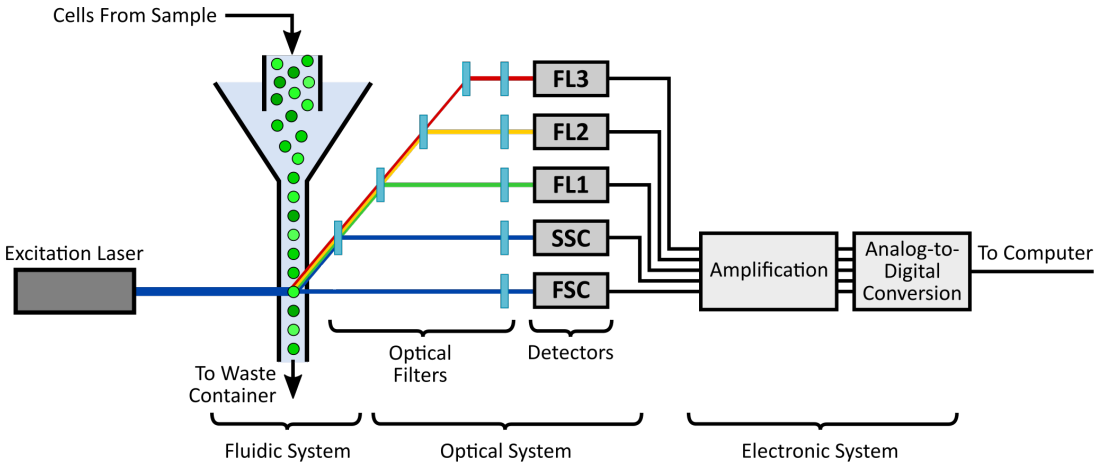


Figure 1.3: Diagram of a flow cytometer, illustrating the fluidic, optical and electronic systems. Image source: AAT Bioquest, Inc [8].

of fluidic, optical and electronic components. The fluid system consists of the saline solution in which the cells, taken from a blood or bone marrow sample for example, are suspended. The machine pressurises the fluid and makes it converge towards a point so that the cells can be analysed one by one. The optical system consists of a laser pointing towards the point of convergence of the cells, which excites the cells, causing light signals that are redirected by a series of dichroic filters and picked up by photomultiplier tubes set at different wavelengths ranging from ultraviolet (355 nm) to red (640 nm). This fluorescence is the result of two phenomena: the cell's natural "colour" or autofluorescence and the fluorescences emitted by antibodies coupled to fluorochromes that have been placed in the saline solution by the cytometrists during the preparation of the sample. In addition, there are two measurements: Forward Scatter (FSC) and Side Scatter (SSC), which provide information about the size, shape and granularity of the cells. Finally, the electronic system records these measurements and saves them in the form of a standardised file (*.fcs* for *Flow Cytometry Standard*). The operation of a cytometer is summarised in Figure 1.3.

From a practical point of view, the data recorded takes the form of a table, where each row corresponds to a cell and each column corresponds to a marker (more precisely, to a wavelength associated with a fluorescence marker). The value of a cell in the table represents the light intensity emitted by a biological cell at a given wavelength.

Cytometrists have a twofold objective: firstly, to choose the right markers to best characterise the cells, as the number of photomultiplier tubes is limited. For example, the cytometers at Metafora have between 10 and 20 sensors. Downstream, the cytometrists are responsible for analysing the data and more specifically for identifying *clusters* (in machine learning terminology) or *gates* (in cytometry terminology). Conventional analysis of cytometry data is done manually as shown in Figure 1.4.

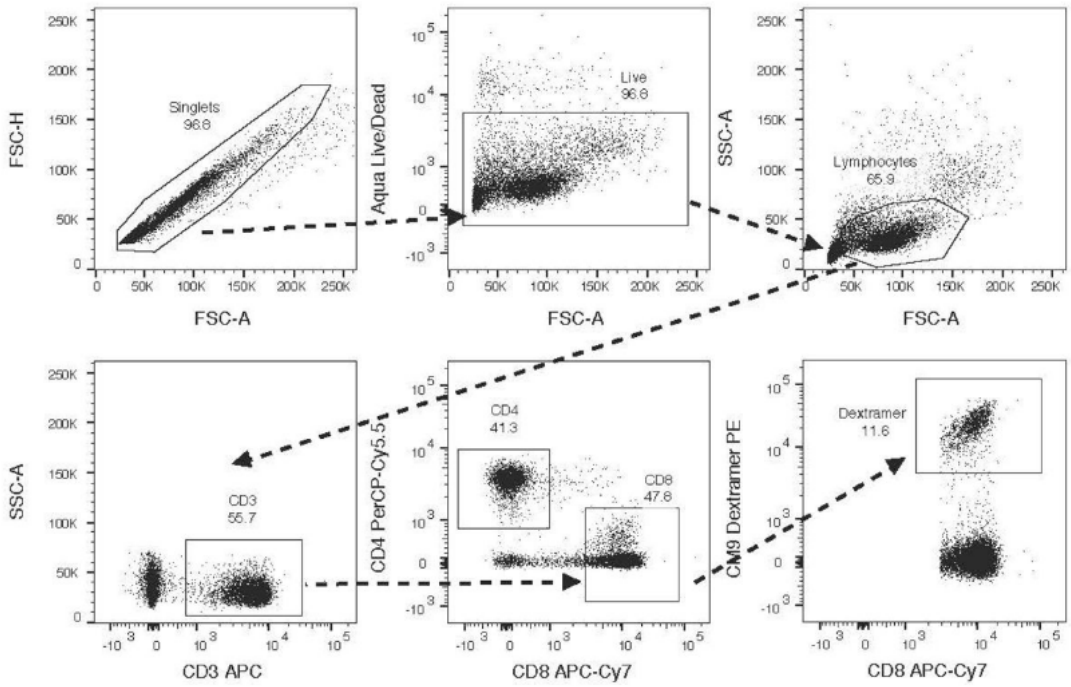


Figure 1.4: *gating* strategy from an article by McKinnon et al. [81]. The *gates* are drawn one by one by looking at the data through 2 well-chosen markers, filtering the data until only the data of interest is left. In this example, singlets (i.e. cells that have passed the laser alone) are selected first, followed by living cells, then lymphocytes, then lymphocytes expressing the CD3 protein complex, lymphocytes expressing the CD8 complex and finally the population of interest. This example illustrates two limitations of this type of analysis: the analysis is slow (6 steps to arrive at the right population) and the gates are drawn manually by the cytometrist, incorporating an element of arbitrariness into the analysis.

To overcome the shortcomings inherent in manual analysis, a growing number of algorithms, papers and software have been developed to automate the analysis of cytometry data, see for example Cheung et al. [23]. Metafora proposes **METAflow**, a software using a hierarchical clustering algorithm, based on density estimation and homological persistence, allowing automatic clustering while leaving the user in control by choosing which branches of the clustering tree to explore.

In Chapter 5 we will try to estimate the proportion of certain cells of interest, i.e. **quantification**, in each node of the tree. This imposes two constraints on the algorithm we are looking for. On the one hand, since the cells of interest do not represent all the cells in a dataset (sometimes they even represent a very small set), we are at best in the open set label shift framework. On the other hand, because of the tree structure, we need to be able to solve the problem on each of the nodes without the algorithm complexity being too high. As a second objective, we will look in to see if Random Fourier Features vectorisations can be used to identify the label on the nodes of the tree directly.

1.2.3 Distribution vectorisation

The approach we will be pursuing throughout this thesis is the use of **Mean Vectorisation** or **Mean Embedding**. We have chosen this approach because the vectorisation of a node in the tree structure can be computed using only the vectorisation of its children. Therefore, we can compute the vectorisation of all the nodes simply by computing the vectorisation of all the leaves once. We can solve the quantification problem for all nodes independently once this step has been performed.

Mathematically, suppose we have a certain distribution $\mathbb{P}$ on the data space $\mathcal{X}$ and let ϕ be a function of $\mathcal{X} \to \mathbb{R}^D$. The mean *vectorisation* of $\mathbb{P}$ by the function ϕ, which we will denote $\Phi(\mathbb{P})$, is then the mean of ϕ under $\mathbb{P}$, i.e. $\Phi(\mathbb{P}) = \mathbb{E}_{\mathbb{P}}[\phi(X)]$. The interesting property of this function is that it is linear in the distribution. In other words, if the target distribution $\mathbb{Q}$ is written as a mixture of reference distributions $\mathbb{P}_i$: $\mathbb{Q} = \sum \alpha_i \mathbb{P}_i$ then $\Phi(\mathbb{Q}) = \sum \alpha_i \Phi(\mathbb{P}_i)$.

One way to solve the quantification problem is then to use the training data to estimate each $\Phi(\mathbb{P}_i)$ and to use the test data to estimate $\Phi(\mathbb{Q})$ and then to search for the proportions $\hat{\alpha}$ such that $\sum \hat{\alpha}_i \Phi(\mathbb{P})$ is as close as possible to $\Phi(\mathbb{Q})$. The most natural distance to use is L_2 and the theoretical work we will present is obtained with it, but in practice other distances could be used.

Remark. We used the term *vectorisation* here, because the function ϕ takes values in $\mathbb{R}^D$. In practice, this is what will do throughout the thesis. However, all the results can also be applied if ϕ takes values in any Hilbert space $\mathcal{H}$, so for simplicity, we will use the term vectorisation throughout this introduction and will not distinguish between $\mathbb{R}^D$ and $\mathcal{H}$, whereas in the body of the thesis we will use the term *embedding* and will only talk about $\mathcal{H}$ so that the results are as general as possible.

The tree structure makes these methods based on *vectorisations by average* particularly

interesting. Since the Φ function is linear for distributions, this means that the vectorisation of a node is equal to a weighted average of the vectorisations of its children. So all we have to do is calculate the vectorisation of the leaves and then work our way up to the top of the tree. The proportions of each node are calculated by solving a QP problem in small dimension, the calculation time for which is negligible compared with the vectorisation calculations.

Any ϕ function can be used, for example, we will talk about methods using the output of a classifier. ϕ is then technically not valued in $\mathbb{R}^D$, but either in the simplex of dimension $(c-1)$: $\Delta^c := \{x \in \mathbb{R}^c \colon x_i \geq 0, \sum x_i = 1\}$ if the classifier returns a probability, or in $\{0,1\}^c$ if the classifier only returns the predicted class. Another example is vectorisation using an intermediate layer of a neural network. Neural networks are algorithms that learn to separate data (often very high-dimensional data such as images) in an intermediate representation, then classify (using a linear classifier) in a second stage. Another form of vectorisation, which will be discussed throughout this manuscript, is **Kernel Mean Embedding**.

First, let us recall the definition of a kernel.

Definition 1.1. *(Semidefinite kernel).* A function $k : X \times X \to \mathbb{R}$ is a semidefinite positive kernel if it is symmetric, i.e. $k(x,y) = k(y,x)$, and if the Gram matrix is positive:

$$\sum_{i,j=1}^{n} c_i c_j k(x_i, x_j) \geq 0, \tag{1.5}$$

for any $n \in \mathbb{N}$, any choice of $x_1, \ldots, x_n \in X$ and any $c_1, \ldots, c_n \in \mathbb{R}$. It is said to be definite positive if equality in (1.5) implies $c_1 = c_2 = \ldots = c_n = 0$.

We can show that for any semidefinite positive kernel k (which we will simply call kernel in the following) there exists a unique space of real functions on X, called the RKHS (reproducing kernel Hilbert space) associated to k and a unique function Φ_k in this space such that $k(x,y) = \langle \Phi_k(x), \Phi_k(y) \rangle$, for all x, y. The Kernel Mean Embedding (KME) of a distribution associated to the kernel k is then defined as :

$$\Phi_k(\mathbb{P}) := \int_X \Phi_k(x) d\mathbb{P}(x) = \mathbb{E}_{\mathbb{P}}[\Phi_k(X)] \in \mathcal{H}_k. \tag{1.6}$$

This object is more complex than the vectorisations we presented earlier, because this time it is valued in an infinite-dimensional Hilbert space, and the integral (1.6) is defined (if it exists) as a Bochner integral.

The L_2 distance between the vectorisation of two distributions is then called the *Maximum Mean Discrepancy*, introduced by Gretton et al. [61] for statistical testing.

However, MMD, like all kernel methods, suffers from a computational problem. The MMD between two distributions can only be accessed via scalar products using the kernel trick:

$$\langle \Phi_k(\mathbb{P}), \Phi_k(\mathbb{Q}) \rangle = \mathbb{E}_{\mathbb{P},\mathbb{Q}}[k(X,Y)].$$

Calculating this scalar product is **quadratic** with the number of points. Indeed, $\Phi_k(\mathbb{P})$, even when $\mathbb{P}$ is an empirical distribution, can not be easily stored on a computer as it is a function. These two elements are both prohibitive for the application to flow cytometry.

1.2.4 Random Fourier Features

To overcome this problem, we will use the **Random Fourier Features** [98] or *RFF*, a kernel approximation method.

Random Fourier Features are based on Bochner's theorem :

Theorem 1.2. *A continuous function φ on $\mathbb{R}^D$ defines a semidefinite positive kernel $k(x, y) = \varphi(x - y)$ if and only if φ is the Fourier transform of a non-negative measure.*

A direct corollary of this result is that any continuous translation invariant kernel k, associated to a function φ, where $k(x, y) = \varphi(x - y)$ is the Fourier transform of a non-negative measure which we denote Λ_k and for which $\Lambda_k(\mathbb{R}^D) = \varphi(0)$. Therefore if we normalise the kernel by $\varphi(0)$, Λ_k is then a probability measure called the *spectral distribution* of k.

The kernel can be rewritten as follows:

$$k(x, y) = \mathbb{E}_{\omega\sim\Lambda_k}[e^{i\omega^T(x-y)}] = \mathbb{E}_{\omega\sim\Lambda_k}[\cos\big(\omega^T(x - y)\big)].$$

By Monte-Carlo, using a sample $(\omega_i)_{i=1}^{D/2}$ iid of Λ_k, the vectorisation $z\colon \mathcal{X} \to \mathbb{R}^D$ defined by

$$z(x) = \sqrt{\frac{2}{D}}\big[\cos(\omega_i^T x),\ \sin(\omega_i^T x)\big]_{i=1}^{D/2}, \tag{1.7}$$

is such that: $k(x, y) = \mathbb{E}\big[z(x)^T z(y)\big]$, where the expectation is taken with respect to $(\omega_i)_{i=1}^{D/2}$. In practice, another vectorisation is commonly used:

$$\tilde{z}(x) = \sqrt{\frac{2}{D}}\big[\cos(\omega_i^T x + b_i)\big]_{i=1}^{D}, \tag{1.8}$$

where b_i are iid samples of the uniform distribution on $[0, 2\pi]$. See Gundersen [64] for the detailed calculation. Although this vectorisation is popular in practice, we would like to point out that the second version gives the worst results in terms of variance and upper bound for the Gaussian kernel [113].

If the vectorisation is valued in $\mathbb{R}^D$, then the complexity of computing the vectorisations is linear in both the number of points and the number of random features, i.e. linear in D. In addition, calculating the vectorisation of a point cloud can be reduced to matrix multiplication, for which GPUs are well suited. To deal with memory overflows on GPUs, we use the Python package *PyKeops* [19].

To sum up, this method allows us to define a function $z\colon X \to \mathbb{R}^D$ (with D a chosen hyper-parameter) such that for all x, y: $k(x, y) \approx z(x)^T z(y)$. This method has three advantages (over other kernel approximation methods such as Nystrom). Firstly, the vectorisation $z(\mathbb{P})$ can be stored on a computer, as it is a vector of $\mathbb{R}^D$. Secondly, the MMD between two distributions is calculated in linear and not quadratic time. And finally, with this z function, we are still within the framework of the vectorisations presented earlier. All the theorems in this manuscript apply directly with z without having to modify the proofs.

In Chapters 3 and 4 we will see this function z as an approximation to the Gaussian kernel and therefore as a way of speeding up computation time, whereas in Chapter 5 the function z will be seen as a vectorisation in itself, i.e. as a means of characterising a distribution in the form of a vector which can be used in various applications of **flow cytometry**.

1.2.5 Contributions

The first contribution (Chapter 2) of this thesis is a literature review on label shift quantification. The aim of this chapter is to introduce the reader to this still little-known area of research in *machine learning*.

This chapter presents a general (but not exhaustive) overview of the main methods in the literature, as well as a presentation of the different approaches proposed in the literature to unify these methods under a single analysis framework. For instance Firat [44] represents different methods in the form of a constrained regression problem in order to present extensions of the algorithms from the binary case to the multiclass case. This same representation is at the heart of the python package Qunfold by Bunse [14]. In a different approach, Garg proposes to unify under a single analysis the two most commonly used algorithms in practice (Adjusted Classify and count and Maximum Likelihood Quantification which we both present in the chapter) in order to justify the practical superiority of one over the other.

This chapter concludes with a presentation of the test protocols proposed in the literature as well as a presentation and an extension of the work of Sebastiani [104] concerning the choice of metric to use.

The second contribution (Chapter 3) is a theoretical and practical study of the set of methods based on mean vectorisations presented in Section 1.2.3. This category of methods includes several of the methods presented in Chapter 2. The contribution of this chapter is twofold: on the one hand, within the classical framework of label shift, we obtain an empirical bound on the approximation error that improves on the bounds obtained for the various methods, while at the same time being more general because it applies to all the methods in the category, see Theorem 3.1. This is also the first work to our knowledge that proposes to unify different methods in the literature in order to obtain a convergence theorem that applies to all methods. This result also provides a criterion for choosing a vectorisation function that can be calculated empirically from the labelled training dataset, which can be used, for example, to choose the best *bandwidth* parameter when the chosen vectorisation

is based on the Gaussian kernel, the choice of *bandwidth* being a classic problem for kernel methods. In the *open set label shift* framework, we also obtain an empirical bound on the approximation error, see Theorem 3.3 and its Corollary 3.2, which is novel, as no analysis of the error of a quantifier has yet been proposed in an *open set label shift* framework. This result highlights a robustness to contamination under the assumption of orthogonality of the noise vectors with the vectors of the other classes. Under this assumption, the error will converge towards zero (although more slowly than in the case of *Label shift*), whereas without this assumption, the error will converge towards a bias equal to the norm of the component orthogonal to the vectors of the source. Therefore, to be able to perform quantification under *open set label shift* one need a vectorisation that ensures orthogonality with the noise. The method using a Gaussian kernel is the most suitable for this type of problem, since the orthogonality of the vectorisations is guaranteed as soon as the distributions are well separated.

We demonstrate the robustness of Gaussian vectorisation to noise on simulated data and on cytometry data from Metafora. A final contribution, minor for this chapter, but important for Chapter 5, is the use of Random Fourier Features [98] a classic method in the literature for accelerating the computation time of kernel methods, as a method for vectorising distributions.

The third contribution (Chapter 4) proposes to use the information contained in the variance rather than just the mean of the vectorisations. This is an extension of the previous chapter for label shift quantification, although the results obtained for open set label shift are also applicable to these methods.

In the MMD literature on hypothesis testing, it is now well established (for instance in a recent article by Hagrass et al. [66] or in older articles by Harchaoui et al. [31, 67]) that methods using only the information contained in the mean are not minimax. Inspired by their work, we propose in this chapter to "regularise" the distance used in Chapter 3 by a certain matrix M. More precisely, while in Chapter 3 the distance used was the L_2, here we propose to use a Mahalanobis-type distance. Using a Bernstein vector theorem, a variant of a result obtained by Wolfer et al. [127], we obtain another bound on the approximation error, this time involving the variance, see Theorem 4.2. Using this theorem we then obtain a criterion for the choice of M which we link to *kernel Fisher Discriminant Analysis*, a dimension reduction algorithm. After discussing the form of this optimal matrix and explaining the practical estimation of the matrix, we test the new methods on the same data as in the previous chapter: simulated data and flow cytometry data from Metafora.

The fourth contribution (Chapter 5) is a case study on the use of Random Fourier Features vectorisation for flow cytometry. We are interested in a real dataset containing the bone marrow analysis of 29 patients suffering from two pathologies: multiple myeloma and monoclonal gammapathies of undetermined significance (MGUS).

Firstly, we demonstrate the value of this vectorisation for an exploratory analysis of a

Cytometry dataset. In particular, we highlight the failure of the proposed gating strategy to identify the *T cells* as well as the *Plasmocytes* of one of the patients. Secondly, we look at the use of RFFs for 2 tasks: identifying in the hierarchical clustering tree derived from METAflow the cluster most likely to be the cells of interest and quantifying the proportions in each of the nodes of the tree.

Finally, the manuscript ends with the Appendix A, which contains three results on norm concentration in Hilbert spaces, used in all the proofs in this manuscript. The first theorem is a vectorial Hoeffding: a classical result based on McDiarmid's inequality. We complete this section by presenting a short history of results of this type in the MMD literature. The second concentration result is a vector version of Bernstein (Theorem A.3) going back to the work of Pinelis [93] in Banach spaces and Yurinsky [129] in Hilbert spaces. The proof proposed in this manuscript is taken from Wolfer et al. [127] with two small differences. Firstly, a typo has been corrected (a $\log(1/\delta)$ which should have been a $\log(2/\delta)$) and secondly, the starting point of their theorem is based on Yurinsky's result whereas we use Pinelis' result. However, the difference is minimal, since it only affects the constancy in front of the $\mathcal{O}(1/n)$ term of the bound: we have $2/3$ whereas they had $4/3$. Finally, for completeness, we present a third theorem, a vector Bennett inequality, this time based on the work of Smale et al. [108] who themselves base their results on a theorem of Pinelis [93]. The difference again concerns the constancy in front of the $\mathcal{O}(1/n)$ term, this time with a 4 instead of a $2/3$.

Chapter 2

Review on Label Shift Quantification

This chapter offers an overview of the quantification problem, drawing directly from prior works that have addressed this topic, namely, the recent book *Learning to quantify* by Esuli et al. [32], and a review by Gonzalez et al. [58]. However, we will try both to reduce the scope of their work by focusing only on the problem of label shift quantification, and thus not discussing parallel problems such as ordinal quantification [39] or multi-label quantification [84], and to extend their work by looking at areas of the literature that are less well known to this community.

After a brief overview of the possible applications of quantification, we will present and discuss a number of methods from the literature, as well as work aimed at unifying these methods within a single framework. Finally, we present the experimental protocols proposed in the literature.

Contents

2.1 Introduction

One of the central problems in machine learning is **classification**. Suppose we have a space $\mathcal{X}$ (usually $\mathbb{R}^d$), a set of c labels $\mathcal{Y} = \{1, \cdots, c\}$ and a probability distribution $\mathbb{P}(X, Y)$ on $\mathcal{X} \times \mathcal{Y}$. Throughout this chapter we will use $\mathbb{P}_X(\cdot)$ and $\mathbb{P}_Y(\cdot)$ to denotes the marginal distribution on $\mathcal{X}$ and $\mathcal{Y}$ of $\mathbb{P}$. In particular, for all $k \in [c]$, $\mathbb{P}_Y(k)$ represent the proportions of point of class k we expect to have in a sample. We will also use $\mathbb{P}(\cdot|y = k)$ to denote the class-k conditional distribution and for all point $x \in \mathcal{X}$, $\mathbb{P}(y = k|x)$ represent the posterior probability, i.e. the conditional distribution of the label y given x and can be understood as the probability that the point x belongs to class k.

The objective is to find a function (a *classifier*) $f\colon \mathcal{X} \mapsto \mathcal{Y}$ that, for a given data point $x \in \mathcal{X}$, predicts the label y associated. Alternatively, we may look for a function $f\colon \mathcal{X} \to \Delta^c$ that, for a given data point $x \in \mathcal{X}$, returns a probability distribution over $\mathcal{Y}$. In this case, $f_j(x)$ is (up to normalisation of f) supposed to estimate $\mathbb{P}(y = j|x)$.

We define a loss function $L\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ or alternatively $L\colon \Delta^c \times \Delta^c \to \mathbb{R}$ and we seek to minimise the expected classification error :

$$\mathcal{E}(f) := \mathbb{E}_{\mathbb{P}(X,Y)}[L(f(X), Y)]. \tag{2.1}$$

Therefore, in classification, the information we seek is related to the individual x. **Quantification** is a related issue where we are interested in global information about the dataset. For a given sample, we want to obtain the **proportions** of the different labels present in the sample.

Suppose we have two distributions, $\mathbb{P}(X, Y)$ and $\mathbb{Q}(X, Y)$. The first one is the distribution of the training data and will be referred to as the *source* distribution, while the second one is the distribution of the test data and will be referred to as the *target* distribution. Without assumptions about the distributions $\mathbb{P}$ and $\mathbb{Q}$, no results can be found. There exist a wide range of assumptions that can be made about the change in distribution, see Quiñonero-Candela et al. [97] or Storkey [112], but the most classical assumption in quantification is called **Label Shift**.

Definition 2.1 (Label Shift)**.** Let $\mathbb{P}(X, Y)$ and $\mathbb{Q}(X, Y)$ be two distributions on $\mathcal{X} \times \mathcal{Y}$. We say that $\mathbb{P}$ and $\mathbb{Q}$ follow the label shift assumption if $\forall i \in [c]$ and $\forall x \in \mathcal{X}$:

$$\mathbb{P}(x|y = i) = \mathbb{Q}(x|y = i),$$

the distribution $\mathbb{Q}$ can therefore be decomposed as follows:

$$\mathbb{Q}_X(x) = \sum_{i=1}^{c} \mathbb{Q}_Y(i)\mathbb{P}(x|y = i). \tag{$\mathcal{LS}$}$$

In other words, the conditional distribution of x given y does not change, but the proportions can change.

Note that this assumption is also sometimes referred to as *Prior probability shift* or *Target shift*. This distribution *shift* is related to the class of so-called $\mathcal{Y} \to \mathcal{X}$ problems. This terminology, derived at least from Fawcett et al. [41], describes problems where the labels y determine the distributions of the data X through a *causal mechanism*. Cytometry data (see Section 1.2.2) is a good example of a $\mathcal{Y} \to \mathcal{X}$ problem for which the label shift assumption seems plausible. From one patient to another, we do not expect the distribution of a particular cell type to change (or at least we hope not) but we can expect the proportions of cells to change.

Suppose we have access to a labelled training dataset $\{x_i, y_i\}_{i=1}^n \in (\mathcal{X} \times \mathcal{Y})^n$, drawn iid from the source distribution $\mathbb{P}(X, Y)$, and a unlabelled test dataset $\{x_{n+j}\}_{j=1}^m$, drawn iid from the target marginal distribution $\mathbb{Q}_X(\cdot)$. Using these two datasets, the goal is to estimate either the target proportions $\alpha^* \coloneqq \mathbb{Q}_Y(\cdot)$ or the empirical target proportions in sample $\{x_{n+j}\}_{j=1}^m$ denoted $\tilde{\alpha}$. We denote by n and m the number of points in the source and in the target samples and n_i and m_i the number of points of class i in the source and target. Note in particular that $\tilde{\alpha}_i = m_i/m$.

Many methods in the literature do not aim at estimating the proportions in the target directly but rather the ratio of proportions $w(\cdot) := \mathbb{Q}_Y(\cdot)/\mathbb{P}_Y(\cdot)$, or alternatively the empirical counterpart $\tilde{w}(i) = m_i/m \times n/n_i$. From a practical standpoint, both approaches lead to the same results, since we have access to a natural estimator for $\mathbb{P}_Y(\cdot)$, i.e. $\hat{\mathbb{P}}_Y(i) \coloneqq n_i/n$. Therefore, we can convert $\hat{w}$ to $\hat{\alpha}$ simply by multiplying with $\hat{\mathbb{P}}_Y$, and vice versa.

From a theoretical perspective, however, the results in the literature often focus on bounding the quadratic approximation error, which in some cases corresponds to $\|\hat{\alpha} - \alpha^*\|_2$, and in other cases to $\|\hat{w} - w\|_2$. To go from one to the other implies taking into account the approximation error of $\hat{\mathbb{P}}_Y$ which adds a term in $\mathcal{O}\left(n^{-1/2}\right)$ to the bound. Therefore, it is not always easy to compare the results of different methods. Nevertheless, it is essential to keep in mind that the estimation of w and α^* corresponds to two viewpoints of the same problem.

2.1.1 Applications

Quantification is often seen as a sub-problem of classification in the sense that a classifier f can be used as a quantifier simply by counting the outputs of the classifier on the target data. It is now widely known in the quantification literature that this approach is sub-optimal because a vanilla classifier is not robust to distribution shift. (we will discuss this point in the next section and in Section 2.2.1). Since classification is ubiquitous in machine learning, developing methods that are robust to any kind of shift (including label shift) has been a widespread subject. A way to create robust classifiers, as we shall see next, is to estimate the proportions of the classes by using a quantification procedure. However, the scope of quantification methods is not limited to classification. An idea that has been pushed by the quantification community is that quantification is a problem on its own. Forman [46] (2005), already noted that many problems in machine are not interested by a local information such as the class of a point x but a global information on the proportions of the sample. Unfortunately, the sub-optimality of a simple classification method to estimate the proportions is not a widely known fact.

Let us take a look at the different applications of quantification.

Classification under Label Shift For classification application, it is crucial to consider the label shift as the classification error on the target is not equivalent to the classification error on the source, but to a weighted version of it:

$$\begin{aligned}
\mathbb{E}_{\mathbb{Q}(X,Y)}[L(f(X),Y)] &= \int_{\mathcal{X}\times\mathcal{Y}} L(f(x),y)\ \mathbb{Q}(x,y)dxdy \\
&= \int_{\mathcal{X}\times\mathcal{Y}} L(f(x),y)\ \mathbb{Q}(x|y)\mathbb{Q}_Y(y)dxdy \\
&= \int_{\mathcal{X}\times\mathcal{Y}} L(f(x),y)\ \mathbb{P}(x|y)\mathbb{Q}_Y(y)dxdy \\
&= \int_{\mathcal{X}\times\mathcal{Y}} L(f(x),y)\ \mathbb{P}(x,y)\mathbb{Q}_Y(y)/\mathbb{P}_Y(y)dxdy \\
&= \mathbb{E}_{\mathbb{P}(X,Y)}[L(f(X),Y)w(Y)], \qquad (2.2)
\end{aligned}$$

Since classifiers are trained to minimise the expected error (2.1) and not (2.2), they do not provide any theoretical guarantees on new data that suffer form label shift, unless we make clear assumption on the "range" of that shift, for instance $\|w\| \leq C$. This is illustrated in Figure 2.1. Equation (2.2) shows that under Label Shift a classifier applied directly to a new dataset is suboptimal. We will discuss the use of classifiers for quantification in the following sections.

In this setting, quantification methods can be used to estimate the ratios w and then train a classifier by minimising Equation (2.2) instead of the traditional loss function. Alternatively, and less costly for complex classifiers such as neural networks, instead of retraining the classifier we can use Bayes' theorem :

$$\mathbb{Q}(y=i|x) \propto w(i)\mathbb{P}(y=i|x),$$

where the posterior distribution $\mathbb{P}(\cdot|x)$ is approximated by $f(x)$.

Fairness Fabris et al. [40] used quantification to address the *fairness-under-unawareness* problem. In this setting one wants to assert the *fairness* of an algorithm w.r.t. certain sensitive attributes (such as the gender or the race) when legal barriers complicate or prohibit the collection of sensitive attributes. In other words, *Fairness-under-unawareness* is the problem of measuring group fairness when the values of the sensitive attributes are unknown.

One way to measure fairness is to use *demographic disparity*, i.e. the difference between "the probability that the classifier outputs positive when the sensitive attribute is **present**" minus "the probability that the classifier outputs positive when the sensitive attribute is **absent**". Unfortunately, to compute an estimate of these two quantities, we need access to the value of the sensitive attribute in our test sample, which is often not available in large numbers for legal or technical reasons. One way around this would be to train a classifier to estimate the sensitive attribute using the non-sensitive ones, but this raises ethical concerns. The authors show that we do not need access to the sensitive attribute of each point, but only to the proportions of these attributes in the target sample

To do this, we need 2 training data sets. The first, which is typically large, will contain only the non-sensitive attributes, while the second will contain all available attributes. We

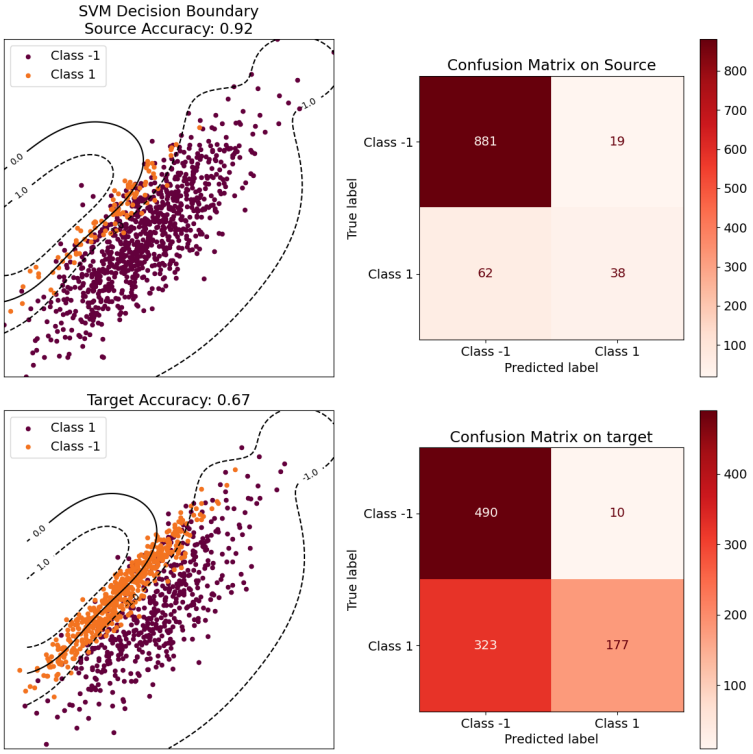


Figure 2.1: In the top row, the decision boundary and confusion matrix of a SVM using the Gaussian kernel on imbalanced data are shown. The accuracy for the first class is low, but the overall accuracy is good as the first class comprises only 10% of the total source data. In the bottom row, the same plot is displayed for the target data. This time, the first class accounts for 50% of the data, leading to a notable decrease in accuracy because the marginal accuracy of the first class remains poor.

use the first to train a classifier (for which we want to compute fairness) and the second to train a quantifier.

This approach has the advantage of providing a form of anonymisation for sensitive parameters, as the quantification returns a vector of proportions on the test set, rather than a function that maps each data point to a sensitive attribute.

However, the authors used a classifier-based quantification method in their experiments, which negates the benefits of quantification. We could measure *Fairness-under-unawareness* using quantifiers that do not rely on classifiers (this is the topic of Section 2.2.5), leading to anonymisation of the sensitive attribute as desired.

Conformal prediction Another application of quantification is conformal prediction. It is a particular instance of classification (or regression) where the objective is to obtain for a new data (X_{n+1}, Y_{n+1}) a set function $C\colon \mathcal{X} \to 2^{\mathcal{Y}}$ with the guarantee that:

$$\mathbb{P}(Y_{n+1} \in C(X_{n+1})) \geq 1 - \gamma,$$

see Shafer et al. [106] for a tutorial on conformal prediction.

Using general results on conformal prediction under distribution shift [121], Podkopaev et al. [94] showed that classical methods from the conformal literature could be adapted to yield asymptotic γ coverage as long as we had a consistent estimator of w.

Direct applications The aspect of quantification that has been most thoroughly explored in the literature is the idea that quantification is an end in itself. In many areas of research, the information sought is not local information about each data point, but global information about the entire dataset. To quote Hopkins et al. [69]:

"*Policy makers or computer scientists may be interested in finding the needle in the haystack [...] but social scientists are more commonly interested in characterizing the haystack. Certainly, individual document classifications, when available, provide additional information to social scientists [...] but they do not usually constitute the* ***ultimate quantities of interest***."

Without providing an exhaustive review of all the domains and articles (see chapter 2 of [32]), we can mention some areas where singular point-wise information is not what is sought.

This is the case, for example, in sentiment analysis [39, 50, 69], where we want to estimate the proportion of text documents that convey positive or negative opinions. For example, for a particular news item, how many tweets about it are positive. In epidemiology, one may be interested in estimating disease prevalence rates across various geographical regions, time periods, age groups, or genders. Daughton et al. [26] used quantification methods on social media posts to derive data on public health trends such as flu vaccination rates. Without going into too much detail, quantification has also been used in ecological modelling [57], market research and political science [9]. A last application of interest to us is flow cytometry analysis [30, 49], because in this context the proportions of cells in a sample can be used as a diagnostic criteria.

2.1.2 Related literature

The term "quantification", introduced by Forman [46], is not the starting point of the literature (see for example, the earlier works of Saerens et al. [101]). Moreover, as we have seen, the problem can also be viewed as the first step of a broader problem of improving classifiers under label shift, as in the work of Saerens. This results in a fragmented literature.

A large number of articles deal with the quantification problem itself, without explicitly using this term or mentioning this literature [2, 4, 49, 70, 73, 77] and vice versa, the results

of these articles are not necessarily known in the quantification literature. One of the aims of this chapter is also to bridge the gap between these different literatures.

2.2 Methods

Numerous methods have been proposed in the literature so far, many of which are already implemented in the Python packages QuaPy [83], QuantificationLib [18] and qunfold [14].

Taxonomies of these methods have already been proposed. Gonzalez et al. [58] divided the methods into 3 categories:

1. *Classify, Count, and Correct*: These methods use a classifier to make an initial approximation of the proportions and then correct the results to account for its bias.
2. *Methods based on adapting traditional classification algorithms to quantification learning*: These methods also use classifiers, but they are trained with a quantification objective, whereas the methods in the first category use any black-box classifier.
3. *Methods based on distribution matching*: These methods aim to estimate the proportions by finding the mixture of $\mathbb{P}(x|y=i)$ closest to $\mathbb{Q}_X(x)$.

Esuli et al. [32] and Quapy adopt the same taxonomy, with methods from categories 1 and 2 grouped into a larger family of methods called "aggregative methods". These are methods that require an initial classification step, as opposed to the non-aggregative methods corresponding to the third category. However, this way of categorizing does not align with more recent research that has shown the connections between different methods [17, 30, 44, 52]. We will discuss these works in Section 2.3. We propose here a non-exhaustive overview of quantification methods, based on the work of Gonzalez et al. [58] and Esuli et al. [32].

Binary and multi-class

Regardless of the taxonomy, the methods we present in this section can be divided into two categories: methods that can deal with more than two classes, and methods that are restricted to two classes. The former will be referred to as "multi-class", while the latter will be referred to as "binary". Actually, only a few of the methods we will present are true binary methods, since most of them can be extended (at least in theory) to the multi-class setting. However, these methods are often written as minimisation procedures, and the authors do not present a way to solve the problem with more classes, except by a trivial (and inefficient) grid search in $\Delta^c \coloneqq \{x \in \mathbb{R}^c \colon x \geq 0,\ \sum x_i = 1\}$.

The classical way in the literature to compare "binary" method in a multi-class setting is the so-called *One-vs-Rest* (OVR) procedure (sometimes also called One-vs-All). This is the same technique as in classification, it consists of fitting one quantifier per class k, where each quantifier is fitted using two classes $\oplus = \{y = k\}$ vs $\ominus = \{y \neq k\}$ and then aggregating the results. Schumacher and his co-authors [103] compared OVR to multi-class and state :

"*By contrast, extending predictions from binary quantifiers to the multiclass case in a one-vs-rest fashion does not appear to yield competitive results, even when using strong base quantifiers such as the Median Sweep or the DyS framework.*"

Recently, Donyavi et al. [28] gave a simple explanation on this poor performance: the label shift ($\mathcal{LS}$) hypothesis that is key for all quantification methods is not satisfied. Indeed, let us compute the marginal distribution of the $\ominus$ class in the source and in the target:

$$\sum_{l\neq k} \frac{\mathbb{P}_Y(l)}{\mathbb{P}_Y(\ominus)}\mathbb{P}(x|y=l) = \mathbb{P}(x|\ominus) \neq \mathbb{Q}(x|\ominus) = \sum_{l\neq k} \frac{\mathbb{Q}_Y(l)}{\mathbb{Q}_Y(\ominus)}\mathbb{P}(x|y=l). \tag{2.3}$$

In other words, the binary quantifier that is train to estimate the proportion $q_Y(k)$ will be applied on data that do not satisfy the label shift hypothesis. "*Therefore, OVR quantification approaches are doomed to underperform*" [28].

Note that in the multi-class setting, we want to estimate a vector, while in the binary setting we only want to estimate the proportion of the first class. To keep the notation simple, we use the same notation α^* for the vector of proportions in the multi-class setting, α^* is then a vector in Δ^c, and the proportion of the first class in the binary setting, α^* is then a real value in $[0,1]$ and the proportions are then given by $(\alpha^*, 1-\alpha^*)$.

To differentiate between the two object, we make it clear whether we are in the multi-class setting or the binary setting.

2.2.1 Classify Count and Correct

Classify and Count

We have seen that quantification and classification are two related problems. The naive method, therefore, consists of using a classifier trained on the source data and applying it to the target data. The estimation of proportions is then given by:

$$\hat{\alpha}_{\text{cc}} = \left(\frac{1}{m}\sum_{j=1}^{m} \mathbf{1}\{f(x_{n+j}) = i\} \right)_i. \tag{2.4}$$

In the quantification literature, this method is referred to as *Classify and Count* (**CC**) [48]. However, this approach offers no guarantee. It is a well known fact in classification that a change in the marginal distribution of labels leads to a decrease in classification performance (see for instance He et al. [68]).

More precisely, it can be shown that the classification error can be decomposed as a sum weighted by the proportions of conditional classification errors (see for instance Gilet et al. [56]):

$$\mathbb{E}_{\mathbb{P}(X,Y)}[L(f(X),Y)] = \sum_{i=1}^{c} \left[\mathbb{P}_Y(i) \sum_{j=1}^{c} L_{ij} \mathbb{P}(f(x) = j | y = i) \right],$$

where $L_{ij} := L(i,j)$. Thus, in practice, it is possible to have both a classifier with good overall classification errors and high conditional errors on small classes, see for instance Figure 2.1.

It has been confirmed empirically in multiple experiments, for instance in Schumacher et al. [103], that **CC** yield poor quantification results without adjustments in both binary and multi-class settings. For instance, if we take the data and the classifier of Figure 2.1, the kernel SVM trained on the biased training data has a large error on the positive class. The corresponding **CC** estimator returns $\hat{\alpha} = (0.813, 0.187)$ instead of $\alpha^* = (0.5, 0.5)$.

The reason why **CC** does not work is because $\hat{\alpha}_{\text{cc}}$ is not the estimate of the desired quantity $\alpha^* := \mathbb{Q}_Y(i)$, but of $\alpha_{cc} := \mathbb{Q}(f(x) = i)$. Using Bayes' theorem along with the Label Shift assumption:

$$\begin{aligned} \alpha_{cc} &= \sum_{j=1}^{c} \mathbb{Q}(f(x) = i | y = j) \mathbb{Q}_Y(j) \\ &= \sum_{j=1}^{c} \mathbb{P}(f(x) = i | y = j) \mathbb{Q}_Y(j) \\ &= C_{\hat{y}|y} \times \alpha^* \end{aligned} \tag{2.5}$$

Where $C_{\hat{y}|y}$ is the conditional confusion matrix of f. In other words, $\hat{\alpha}_{\text{cc}}$ will only be a good estimate of α^* if $C_{\hat{y}|y}$ is the identity matrix, i.e. the classifier is perfect or alternatively if α^* is an eigenvector of $C_{\hat{y}|y}$. We will come back to this second option in Section 2.2.4.

In the binary case, a theorem by Forman [48] summarises the failure of **CC**.

Theorem 2.1. *For an imperfect classifier even if the number of points in the target is infinite, the **CC** method will underestimate the true proportion of positives α in the target for $\alpha > \bar{\alpha}$, and overestimate for $\alpha < \bar{\alpha}$, where $\bar{\alpha}$ is the particular proportion at which the **CC** method estimates correctly; that is, the **CC** method estimates exactly $\bar{\alpha}$ for a test set having $\bar{\alpha}$ positives.*

This theorem can be visualised in Figure 2.2.

Two approaches can be proposed to fix this: ante-hoc and post-hoc. In the first approach, we "correct" the classifier before seeing the target distribution. For instance, we can try to learn a classifier that will perform optimally regardless of the target proportions. This is the idea behind the *Prior Probability Shift* literature [12, 55, 56, 120], where the objective is precisely to obtain classifiers that are robust to changes in the proportions of the classes, i.e. a label shift. Unfortunately, no comparative study has been conducted yet to determine if the methods from the *Prior Probability Shift* literature yield good quantifiers when use with a Classify and Count procedure.

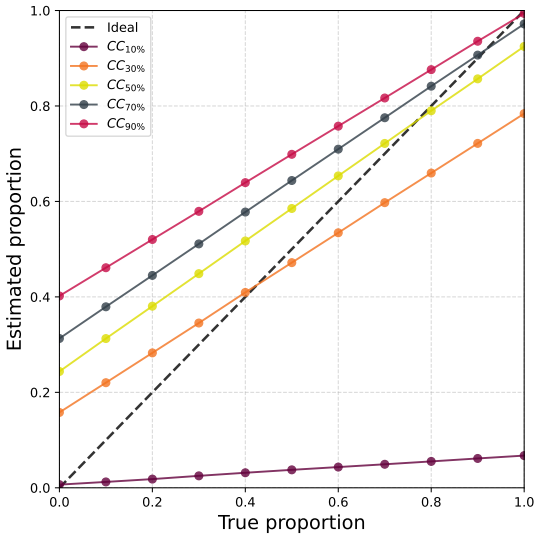


Figure 2.2: In a binary case where we only need to estimate the proportion of the first class, (i.e. positive label) we train a Quadratic discriminant analysis (QDA) on the source and apply it to the target where we change the proportion of the positive label, from 0 to 1. Each coloured line represents a QDA trained on the source, where the percentage of positive label in the source used to train the QDA changes. For instance $CC_{10\%}$ was trained on a source with only 10% of positive label. This is an illustration of the phenomenon described by Forman in Theorem 2.1. This figure is inspired by that of Esuli et al. [32].

The second class of approach, more traditional in quantification literature, consists of using a black-box classifier and applying post-hoc calibrations. The name used in the quantification literature for theses methods is *Adjusted Classify and Count* and was dubbed *Black-box shift estimation* by Lipton et al. [77].

Adjusted Classify and count

Following Equation (2.5), one way to estimate α^* when we have an estimate of α_{cc} is to use a held-out validation set to estimate $C_{\hat{y}|y}$ and solve the linear system:

$$\hat{\alpha} = \hat{C}_{\hat{y}|y}^{-1}\hat{\alpha}_{\text{cc}} \tag{2.6}$$

Where, $\hat{\alpha}$ is the estimated vector of proportions, $\hat{C}_{\hat{y}|y}$ is the empirical conditional confusion matrix and $\hat{\alpha}_{\text{cc}}$ is the vector obtained through Classify and Count (2.4).

The approach sometimes attributed to Forman [46, 48] in the quantification literature is actually older and has a longer history in epidemiology [11].

In its binary version [11, 48], where there are only two classes, and we are interested in estimating the proportion of the first class α^*, Equation (2.5) is then written as:

$$\alpha^* = \frac{\alpha_{cc} - \mathit{fpr}}{\mathit{tpr} - \mathit{fpr}}, \tag{2.7}$$

where $\mathit{tpr} = \mathbb{P}(f(x) = 1|y = 1)$ is the *true positive rate*, and $\mathit{fpr} = \mathbb{P}(f(x) = 1|y = -1)$ is the *false positive rate*. Note that there is no guarantee that $\hat{\alpha} \in [0, 1]$ when we replace *tpr*, *fpr* and α_{cc} by approximations in Equation (2.7).

Going back to the example in Figure 2.1, from the training confusion matrix we can see that $\mathit{fpr} = 0.02$ and $\mathit{tpr} = 0.38$. The values of *fpr* and *tpr* obtained on the test data are $\mathit{fpr} = 0.02$ and $\mathit{tpr} = 0.35$. The estimation of *tpr* is quite precise, resulting in an estimate using Equation (2.7) of $\hat{\alpha} = (0.54, 0.46)$ which is close to the real solution $\alpha^* = (0.5, 0.5)$.

Forman [47] proposes, in the binary case, an ante-hoc procedure, called threshold policy, to enhance the method even more. A binary classifier is of the form $f(x) = \mathbf{1}_{\phi(x)\geq 1/2}$, meaning the classifier predicts class 1 if the model's output is greater than or equal to $1/2$. Forman suggests changing this threshold to $f_\delta(x) = \mathbf{1}_{\phi(x)\geq\delta}$, where $\delta \in (0, 1)$.

The purpose of this method is to find "a threshold that admits more true positives and many more false positives, yielding worse classifier accuracy but better quantifier accuracy" [47]. Forman proposed 4 heuristics for the choice of threshold δ (see Figure 2.3):

1. *Max*: Selects the threshold that maximizes the numerator $\mathit{tpr} - \mathit{fpr}$.
2. *X*: Selects the threshold such that $1 - \mathit{tpr} = \mathit{fpr}$.
3. *T50*: Selects the threshold such that $\mathit{tpr} = 0.5$.
4. *Median Sweep* (MS): The estimated proportion is given by the median of the proportions obtained for each threshold δ.

For the multi-class case, Lipton et al. [77] proposed a version of Adjusted Classify and Count, without the threshold policy, under the name *Black-Box Shift Estimator* (BBSE). In the case where we want to estimate the ratios rather than the proportions, Equation (2.6) can be rewritten as:

$$\hat{w} = \hat{C}_{\hat{y},y}^{-1}\hat{w}_{\mathrm{cc}}. \tag{2.8}$$

Here, $\hat{C}_{\hat{y},y}$ is the estimated confusion matrix where $(C_{\hat{y},y})_{i,j} = \mathbb{P}(f(x) = i, y = j)$, and $\hat{w}_{\mathrm{cc}}(i) = \hat{\alpha}_{\mathrm{cc}}(i) \times \hat{\mathbb{P}}_Y(i)$ is the empirical estimate of the ratios using classify and count.

Lipton and his co-authors provided theoretical guarantees on the estimation error of the ratios.

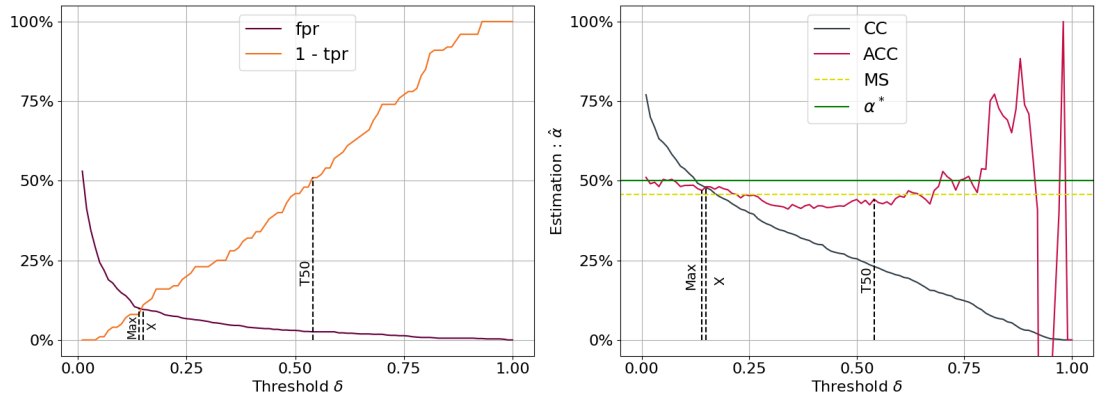


Figure 2.3: Figure inspired by Forman [47] where we use the same data and classifier as in Figure 2.1. On the **left**, *fpr* and $(1 - tpr)$ are shown as functions of the threshold threshold δ of the classifier. A classifier with a perfect threshold would have a region where both curves are equal to 0. The vertical dashed lines represent the different values of δ for the different heuristics presented: *Max*, *X* and *T50*. On the **right**, the estimate proportion of Classify and Count (**CC**) and Adjusted Classify and Count (**ACC**) are shown with respect to δ. The estimate proportion of the *Median Sweep* (MS) heuristic is shown as a yellow dashed lines. In green, the true proportion $\alpha^* = 0.5$. In this example, a well chosen threshold can achieve a higher accuracy than using the ACC method directly.

Theorem 2.2 ([77]). *Let λ_{min} be the smallest eigenvalue of the confusion matrix of f. There exists a constant $C > 0$ such that for all $n > 80\log(n)2\lambda_{min}^{-2}$, with probability at least $1 - 3cn^{-10} - 2cm^{-10}$ we have :*

$$\|\hat{w} - w\|_2^2 \leq \frac{C}{\lambda_{min}^2}\left(\|w\|^2\frac{\log n}{n} + c\frac{\log m}{m}\right),$$

where we recall that c is the number of classes, n the number of points in the source and m the number of points in the target.

This bound was improved by Azizzadenesheli et al. [4] and in Chapter 3 of this manuscript. There are two variants of this method in the literature. The first one consists of solving a quadratic problem instead of inverting the matrix:

$$\hat{\alpha} = \underset{\alpha \in \Delta^c}{\operatorname{argmin}} \ \|\alpha\hat{C}_{\hat{y}|y} - \hat{\alpha}_{\mathrm{cc}}\|^2, \tag{2.9}$$

where $\Delta^c = \{\alpha \in [0,1]^c \mid \sum_{i=1}^c \alpha_i = 1\}$, is the $(c-1)$-dimensional simplex.

This version was used by Hopkins et al. [69] for text content analysis and in Chapter 3 as a specific case of our general method: Distribution Feature Matching.

As discussed in Chapter 3, the difference between (2.6) and (2.9) lies in the fact of imposing the condition $\alpha \geq 0$, the condition $\sum_{i=1}^{c} \alpha_i = 1$ being already imposed by the nature of $\hat{C}_{\hat{y}|y}$ and $\hat{\alpha}_{\text{cc}}$. Both approaches coincide when $\hat{\alpha} \geq 0$, but in other cases, since we are looking for proportions, (2.9) will be more reliable than (2.6).

In the case where we want to estimate the ratios, (2.9) is written as:

$$\hat{w} = \underset{w \in \mathcal{W}}{\operatorname{argmin}} \ \|w\hat{C}_{\hat{y},y} - \hat{w}_{\text{cc}}\|^2, \tag{2.10}$$

where $\mathcal{W}$ is the set of vectors such that $w \geq 0$ and $\sum_{i=1}^{c} w_i \times \mathbb{P}_Y(i) = 1$. Since the proportions of the source are unknown, they are replaced by $\hat{\mathbb{P}}_Y := n_i/n$, the natural estimation of $\mathbb{P}_Y$. This approach was used by Tachet et al. [114] as a building block for classification in a domain adaptation setting, which is more general than label shift.

Another variant, introduced by Azizzadenesheli et al. [4], in the case where we want to estimate the ratios, consists in regularising the objective (2.9) by penalising it with $\|\hat{w} - \mathbf{1}\|$. More precisely:

$$\begin{aligned} \hat{\theta} &= \underset{\theta}{\operatorname{argmin}} \ \|\theta\hat{C}_{\hat{y},y} - (\hat{w}_{\text{cc}} - \mathbf{1}\hat{C}_{\hat{y},y})\|_2 + \gamma\|\theta\|_2, \\ \hat{w} &= \mathbf{1} + \lambda\hat{\theta}. \end{aligned}$$

Here, γ is a hyperparameter that controls the strength of the regularisation term, and $\mathbf{1}$ is the vector of ones. The regularisation term encourages the estimated ratios $\hat{w}$ to be close to the vector of ones, which means that the method seeks solutions where the estimated label shift is close to zero.

In the absence of label shift, where $w = \mathbf{1}$, the parameter $\theta = w - \mathbf{1}$ represents the "amount of shift". The value of λ is chosen to be proportional to the number of points used during training. The aim of the regularisation is to make the quantifier less sensitive to the singularity of the $\hat{C}_{\hat{y},y}$ matrix, as Forman proposed with his threshold policy.

CDE-iteration

As mentioned above, one way to improve the performance of a classifier on a new dataset resulting from label shift is to train it with weights. This approach is known as Cost Sensitive Learning [42]. In this setting, we define a *cost matrix* C, where $C_{i,j}$ represents the cost of estimating class i when the true class is j. For two classes, we only need to define two values : c_{-1} the cost of a false positives, i.e. the price paid for classifying a sample as positive when it is not, and c_1 the cost of a false negatives. In Equation (2.2), the ratio of the proportions $w(y)$ acts as the weight (c_{-1}, c_1).

There is therefore a strong connection between the proportions and the weights (see Figure 2.4). The costs that should be used when building the classifier satisfy :

$$\frac{c_1}{c_{-1}} = \frac{\mathbb{P}_Y(-1)\mathbb{Q}_Y(1)}{\mathbb{Q}_Y(-1)\mathbb{P}_Y(1)}. \tag{2.11}$$

For example, in the case shown in Figure 2.1, we have $\frac{\mathbb{P}_Y(-1)}{\mathbb{P}_Y(1)} = 9$, if we denote $z := \mathbb{Q}_Y(1)$ and fix $c_{-1} = 1$[1], we obtain $c_1 = \frac{9z}{1-z}$. In Figure 2.4 we keep the same setting with $\mathbb{Q}_Y(1) = 0.7$ and thus $c_1 = 21$ and with show the difference between learning without weights ($c_1 = c_{-1}$) and learning with the optimal ones.

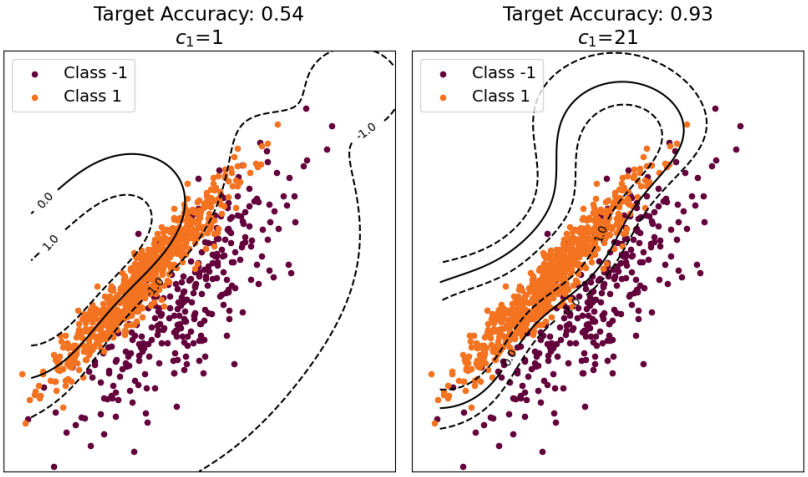


Figure 2.4: The impact of learning weight on accuracy is evident when considering imbalanced source and target distributions. Let us revisit the example from Figure 2.1. In the source distribution, the classes are heavily imbalanced with a ratio of $(0.9, 0.1)$, while in the target distribution, the imbalance is in the opposite direction with a ratio of $(0.3, 0.7)$. On the left, the decision boundary and accuracy of a *vanilla* kernel SVM ($c_1 = 1$) and on the right the decision boundary and accuracy of a cost sensitive kernel SVM with the theoretical weight $c_1 = 21$.

The choice of c_1 is therefore equivalent to making an assumption about the proportion $\mathbb{Q}_Y$ one is trying to estimate. It is expected that the accuracy would be optimal for the true choice of c_1 (see Figure 2.5).

Class Distribution Estimation Iterate or CDE-iteration proposed by Xue et al. [128] involves training a classifier with initial weights (c_{-1}, c_1) set to $(1, 1)$, estimating the proportions using Classify and Count with this classifier, calculating the weights using Equation (2.11), and repeating this process until convergence.

[1]Multiply the loss function of Equation (2.2) by a constant does not change the value of the minimum, so the values c_1 and c_{-1} are not important, what matters is the ratio between the two values. Therefore we can set $c_{-1} = 1$ without a loss of generality.

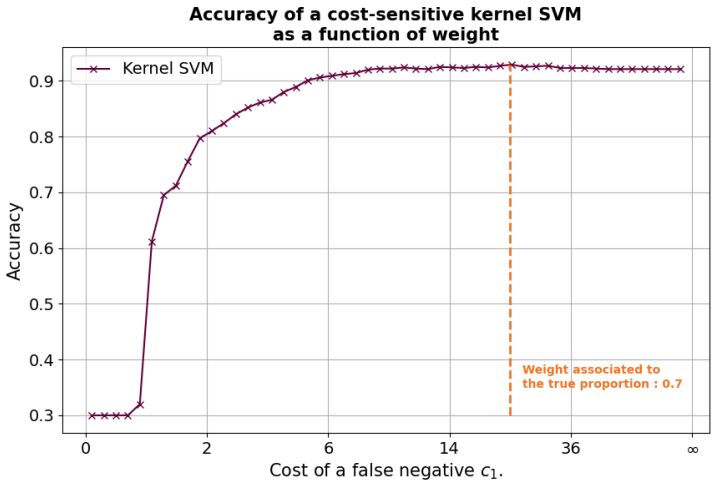


Figure 2.5: Several kernel SVMs with different costs are trained on the same setting as in Figure 2.4, where the classes are highly imbalanced with a ratio of $(0.9, 0.1)$ in the training set and the imbalance is in the opposite direction with a ratio of $(0.3, 0.7)$ in the test set. The figure shows the accuracy of the kernel SVMs obtained on the test data set based on the choice of cost c_1 (equivalent to choosing a proportion of positive in the test z, since according to Equation (2.11), $z = 9c_1/(1 - c_1)$). The weight associated with the true proportion $x = 0.7$ is indicated by the orange dashed line.
CDE is based on the premise that accuracy should be optimal for the true choice of c_1. However, as we can see in this example, this accuracy is within a plateau and other choice of cost c_1 (associated to different proportions z) would lead to similar accuracy.

Tasche [116] proposed an alternative version of this method that does not involve re-training. They note that for given weights (c_{-1}, c_1), the Bayes estimator is given by the formula:

$$g_{c_{-1},c_1}(x) = \begin{cases} -1 & \text{if } \mathbb{P}(y = -1|x) \geq \frac{c_1}{c_{-1}+c_1} \\ 1 & \text{else} \end{cases} \tag{2.12}$$

Thus, the method proposed by Tasche, which could be called Bayes-CDE, involves training and then estimating the weights using Classify and Count, and repeating the process by updating the threshold of the estimator $\hat{g}_{c_{-1},c_1}$.

Neither CDE-iteration nor Bayes-CDE are competitive methods in the literature,as shown by the comparative experiments of Schumacher et al. [103]. This can be explained, on one hand, by the exemple of Figure 2.5 and on the other hand, by the work of Tasche [116], which defines a notion of Fisher Consistency for quantification that this methods does not

satisfy. In other words, the method does not necessarily achieve the correct proportions when the distributions $\mathbb{P}(x|y=i)$ and $\mathbb{Q}_X(x)$ are directly known or equivalently when the number of points goes to infinity.

2.2.2 Methods that use a soft classifier

The methods presented above only used the predicted class of the classifier. Suppose that the classifier is "soft" meaning that it outputs a distribution on the space of labels, $f(x) \in \Delta^c$ is then interpreted as an approximation of $\mathbb{P}(y|x)$. An important property of soft-classifier is the calibration. Let us recall the definition of a calibrated estimator.

Definition 2.2. A classifier $f\colon \mathcal{X} \mapsto \Delta^c$ is said to be canonically calibrated on the source distribution $\mathbb{P}$ if:

$$\forall x \in \mathcal{X} \text{ and } \forall j \in \mathcal{Y},\ \mathbb{P}(y=j|f(x)) = f_j(x).$$

Intuitively, a classifier is calibrated if "*among test instances receiving a predicted probability vector s, the class distribution is (approximately) distributed as s*" [107].

In general, calibration matters because the outputs of such classifier can directly be interpreted as a confidence level, but in quantification it is even more important because we can directly estimate the proportions with a calibrated classifier. Indeed roughly speaking, for a given value $s \in [0,1]^c$, if we look at all the points x such that $f_j(x) = s_j$, we know that $s_j\%$ of them will be of class j. Some classifier are naturally calibrated (at least when the number of points goes to infinite) such as logistic regression [131], because the loss used during training is a calibration loss, while other are known to be overconfident such as neural network [65] and need to be calibrated using hold-out validation sets and post-hoc calibration methods (see Silva et al. [107] for an overview on calibration).

The importance of calibration for quantification was brought to light by Garg et al. [52] and Alexandari et al. [2] who showed that quantification with calibration is "*hard to beat*" for at least two methods: BBSE and MLLS. We present the results on the importance of calibration for MLLS in Section 2.2.3 and we come back to BBSE in Section 2.3.4.

The reason why calibration is not discussed for the other methods in this section is not because it is not important, but rather because the authors of those methods have not proposed a theoretical study of the approximation error of their algorithms. We strongly believe that these methods would benefit from a post-hoc calibration procedure.

Probabilistic Adjusted Classify and count

Adjusted Classify and count (or BBSE) presented above can also be adapted to a soft classifier. Bella et al. [7] proposed 2 variations in their work. The first one is called *Probability Average* (PA), but it is commonly referred to as *Probabilistic Classify and Count* (PCC) in the quantification literature, serving as the probabilistic counterpart of CC. It simply involves

averaging the classifier outputs on the target data:

$$\hat{\alpha}_{\text{pcc}} = \frac{1}{m}\sum_{j=1}^{m} f(x_{n+j}). \tag{2.13}$$

Naturally, this naive estimator faces the same issues as the standard CC estimator (refer to Corollary 6 in Tasche [115]).

The other variation, known as *Scaled Probability Average* (SPA) or frequently referred to as *Probabilistic Adjusted Classify and Count* (PACC) or *Probabilistic Adjusted Count* (PAC) in the quantification literature, is the probabilistic counterpart of ACC in its binary version. It involves replacing $\hat{\alpha}_{\text{cc}}$ with $\hat{\alpha}_{\text{pcc}}$ in Equation (2.7), *tpr* with the average of the estimator for class 1 from the source, and *fpr* with one minus the average of the estimator for class 0 from the source.

Lastly, Lipton et al. [77] introduced a probabilistic version of BBSE (which is a multiclasses PAC). This approach replaces the confusion matrix in Equation (2.6) with its probabilistic equivalent. Theorem 2.2 is also valid for this probabilistic BBSE.

Hellinger Distance-y

For every classifier f, we can examine the histogram (with a fixed number of bins b) of the classifier's outputs. If we denote $P_i := P_i(b)$ the population histogram with b bins corresponding to the output of the classifier on the distribution $\mathbb{P}(x|y=i)$, and Q the population histogram corresponding to the outputs of the classifier on the target distribution, then because of the Label Shift hypothesis $(\mathcal{LS})$, we have :

$$\alpha^* P_1 + (1-\alpha^*)P_2 = Q \tag{2.14}$$

The method proposed by González-Castro et al. [59] is to use the Hellinger distance to find the weight $\alpha \in [0,1]$ that minimise the distance between $\hat{Q}$ and $\sum_i \alpha_i \hat{P}_i$, where $\hat{P}_i$ and $\hat{Q}$ are the histograms estimated from the data :

$$\hat{\alpha} = \underset{\alpha\in[0,1]}{\operatorname{argmin}} \ \text{HD}\left(\sum_{i=1}^{c}\alpha_i\hat{P}_i, \hat{Q}\right). \tag{$\mathcal{HD}y$}$$

The find the minimum of $(\mathcal{HD}y)$, the authors proposed a simple grid search on $[0,1]$.

An important caveat is that the method depends on the number of bins used to compute the histograms. This hyperparameter has a important impact on the results, but the authors did not propose a criterion for choosing it. Instead, they proposed an ensemble procedure to aggregate the results, which we return to in Section 2.2.6.

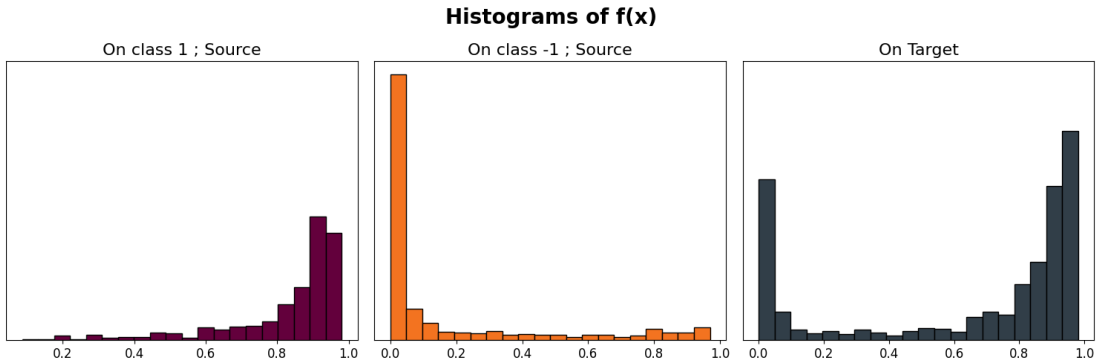


Figure 2.6: Figure inspired by Gonzalez et al. [59] with the data for Figure 2.1. The figure illustrates the output distributions of a kernel SVM for the two classes in the source (left and middle) and the target (right).

Distribution y-Similarity

Maletzke et al. [79] proposed an extension of the previous method by replacing the Hellinger distance with other distances. This results in a range of methods (including HDy and Wasserstein-y that we present next).

Among the distances proposed to compare the histograms, we can list Squared Euclidean, Manhattan, Jensen-Shannon divergence, Hellinger (HDy) or Topsøe, which seem to give the best results in their experiments, although it is not clear whether this result depends on the data they choose or if there exists a theoretical argument to justify it.

As explained in detail by Moreo et al. [85], Firat [44] was the first to propose an extension for HDy and DyS to the multi-class setting. Their idea is to compute an histogram (with b bins) for each of the c marginal, and then concatenate the vectors to obtain a $(c \times b)$ vectorial representation of the data. As Moreo pointed out, this has the (unwanted) side effect of giving a representation that is no longer a probability distribution, since they sum to c, which makes it strange to use distances such as Hellinger or Wasserstein. Dividing the vector by c would lead to a vectorial representation that is artificially bounded by $1/c$. An alternative would be to compute the mean Hellinger distance of each marginal. As pointed out by Moreo and his co-authors the two procedures are equivalent up to constant.

One way to fix this would be to simply compute a histogram directly on the simplex, but this scales combinatorially with the number of classes and bins, making the method unsuitable for large numbers of classes. Therefore if we want to extend HDy and DyS to the multi-class setting we have a trade off to make. Either the problem scales linearly with the number of classes, but we only consider the information contained in the marginals, or the problem scales exponentially with the number of classes, but we have access to all the information.

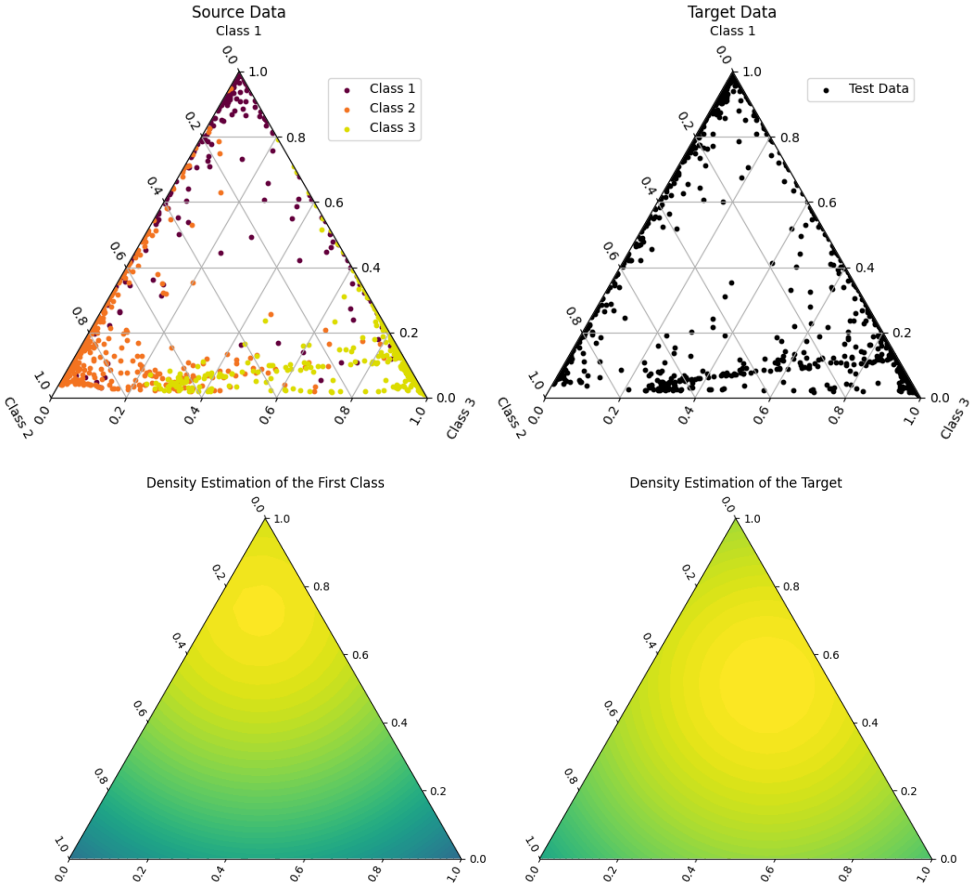


Figure 2.7: Figure Inspired Moreo et al. [85]. The figure illustrates the output distributions of a kernel SVM for the three classes in the source (upper left) and the target (upper right). Instead of comparing the two distributions using the histograms, Moreo et al. suggested first estimating the density on the simplex using a KDE and then comparing the density.

Kernel Density

Moreo et al. [85] proposed a new method to harness the power of classifiers without suffering from the pitfalls of histogram representations. To do this, their method, called KDEy, relies on a *kernel density estimation* of the PDF of $f(x)$ with a Gaussian kernel. The method is illustrated in Figure 2.7. For each class, we compute $L_i(f) = 1/n_i \sum_{i=1}^{n_i} \delta_{f(x_i)}$ the empirical distribution of the classifier outputs, where δ_x is the Dirac measure on x. Then, using a classical Gaussian KDE, they estimate the PDF of this distribution $\hat{p}_{L_i(f)}$. The goal is now the same as in HDy and DyS: find the mixture of $\hat{p}_{L_i(f)}$ that is "closest" to $\hat{q}_{L(f)}$, where $\hat{q}_{L(f)}$ is the KDE estimate of the classifier outputs on the target distribution.

$$\min_{\alpha \in \Delta^c} \mathcal{D}\left(\sum_{i=1}^{c} \alpha_i \hat{p}_{L_i(f)}, \hat{q}_{L(f)}\right) \tag{2.15}$$

They proposed 3 methods, all of which use a different $\mathcal{D}$, and all of which differ in the optimisation procedure used to solve (2.15): The first uses the Hellinger distance as $\mathcal{D}$, which leads to a natural extension of $(\mathcal{HD}y)$. In fact, this first approach is more general, as it can be used for any f-divergences, such as the Total Variation Distance, Jensen-Shannon or Topsøe. For this class of functions D, we can solve (2.15) using a Monte Carlo approximation. The second uses the Cauchy-Schwarz divergence because we can compute a close form solution. Finally, they suggest using the Kullback-Leibler divergence. The KL divergence is also an f-divergence and so we could use a Monte Carlo approximation, but since KL is closely related to the maximum likelihood framework, we can solve (2.15) using the EM algorithm. This last method, which the article shows to be the best in their experiments, can be seen as a natural extension of the maximum likelihood framework that we present in detail in Section 2.2.3.

Wasserstein Distance

Let us recall the definition of the p-Wasserstein distance between two empirical distributions: $a = \sum_{i=1}^{n} \gamma_i \delta_{a_i}$ and $b = \sum_{j=1}^{m} \sigma_j \delta_{b_j}$, where (a_i) and (b_j) are points in the space $\mathcal{X}$, $\gamma \in \Delta^n$, and $\sigma \in \Delta^m$.

We denote $\Pi(\gamma, \sigma) = \{M \in \mathbb{R}^{n \times m} \colon M1_n = \gamma \text{ and } M^T 1_m = \sigma\}$ the space of coupling matrices between γ and σ. The Wasserstein distance between a and b, is given by:

$$W_p^p(a, b) = \inf_{M \in \Pi(\gamma, \sigma)} \sum_{i,j}^{n,m} \|a_i - b_j\|^p M_{i,j}. \tag{2.16}$$

This distance could be applied to histograms, resulting in a DyS method. However, Maletzke et al. [79] proposed a variant, called SORD, in which they use the "earthmover distance" i.e. W_1, not on the histograms, but on the outputs directly as we presented in the previous section, leading to the following minimisation problem:

$$\operatorname*{argmin}_{\alpha \in \Delta^c} \; W_1\left(\sum_{i=1}^{c} \alpha_i \hat{p}_{L_i(f)}, \hat{q}_{L(f)}\right). \tag{2.17}$$

The main advantage of SORD is that it no longer depends on the number of bins b and it can naturally extended to the multiclass setting. However the minimisation procedure proposed by the authors to solve (2.17) is only applicable in the binary case.

Freulon et al. [49] proposed a similar method, which we present in Section 2.2.5, where they solve (2.17) directly on the data without first transforming the data using a classifier. We believe that the minimisation procedure they developed for their method could be used to solve (2.17).

Forman's Mixture Model (FMM)

Forman [48] proposed to use the cumulative distribution function (CDF) of the outputs (see Figure 2.8). To compare the cdf, he used what he called the *PP-Area* metric. As pointed out by Firat [44], this is equivalent to using the L_1 norm on the cumulative sum (*cumsum*) of the bins in the CDF. Castaño et al. [17] also show that this is equivalent to the method SORD presented above. In fact, this equivalence is due to a general fact about Wasserstein distance in one dimension: the W_p distance between two densities is equivalent to the L_p norm of the two CDFs.

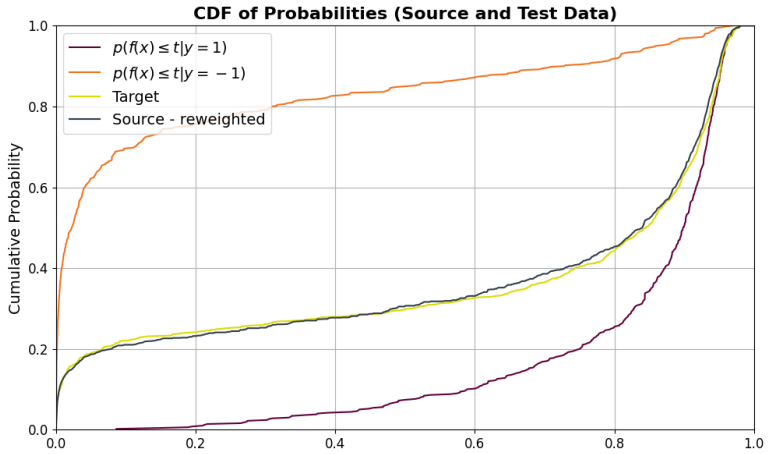


Figure 2.8: Orange line represents the cumulative distribution function (CDF) of the outputs of the SVM for the first class in the source, while prune is for the second class. The CDF of the target is shown in yellow. Additionally, the CDF of the source, reweighted using the true weight $(0.7, 0.3)$ of the target, is illustrated in gray.

2.2.3 Maximum Likelihood Quantification

Another way to view quantification is to consider it as a parametric model for which we aim to estimate the parameter. More precisely, let

$$\mathcal{M} := \left\{ \mathbb{P}_\alpha := \sum_{i=1}^{c} \alpha_i \mathbb{P}(x|y=i);\ \alpha \in \Delta^c \right\}, \tag{2.18}$$

be a parametric model. Under the label shift assumption, we know that $\mathbb{Q} \in \mathcal{M}$. As always, we can view the problem from the perspective of ratios, and in this case, we set up the model as $\mathcal{M}_\mathcal{W} := \{\sum_{i=1}^{c} w_i \mathbb{P}(x|y=i),\ w \in \mathcal{W}\}$, where $\mathcal{W} := \{w \in \mathbb{R}^c \mid \sum_{i=1}^{c} w_i \mathbb{P}_Y(i) = 1\}$.

The maximum likelihood algorithm was introduced by Saerens et al. [101] as an alternative to the classical "confusion matrix" method presented earlier. The method is called *MLLS* (*Maximum Likelihood Label Shift*) in Alexandari et al. [2] and Garg et al. [52], *EMQ* (*Expectation Maximisation Quantifier*) in Saerens et al. [101], and SLD algorithm, named after the authors of the original article (*Saerens-Latinne-Decaestecker*), in the quantification literature, [32, 33, 83].

We will stick to MLLS, firstly because a significant part of the analyses we will discuss come from articles that refer to it by this name, and secondly because the estimator is independent of the optimisation method used (expectation maximisation).

A way to find the best parameter α in $\mathcal{M}$ is to use the maximum likelihood estimator. Given a sample $\{x_1, \cdots, x_m\}$ drawn from the true target marginal distribution q, we seek to maximise the log-likelihood defined as:

$$\begin{aligned} l(\{x_1, \cdots, x_m\}; \alpha) &= \sum_{k=1}^{m} \log \sum_{i=1}^{c} \mathbb{Q}(x_k | y = i)\alpha_i \\ &= C + \sum_{k=1}^{m} \log \sum_{i=1}^{c} \frac{\mathbb{P}(y = i|x_k)}{\mathbb{P}_Y(i)}\alpha_i. \end{aligned} \tag{2.19}$$

For the detailed calculations, see, for example, Alexandari et al. [2]. The quantities $\mathbb{P}(y = i|x_k)$ and $\mathbb{P}_Y(i)$ are unknown, so we have to replace them with plug-in estimates: $\hat{\mathbb{P}}(y|x_k) := f(x_k)$ and $\hat{\mathbb{P}}_Y(i)$, where f is an estimator obtained by training on the source data. Let us present some of the results proved by Alexandari et al. [2].

Lemma 2.1. *Regardless of the estimators f and $\hat{p}$, the function $\sum_{k=1}^{m} \log \sum_{i=1}^{c} \frac{f_i(x_k)}{\hat{\mathbb{P}}_Y(i)}\alpha_i$ is concave on α and bounded by above as long as $\hat{\mathbb{P}}_Y > 0$.*

Thus, for given f and $\hat{\mathbb{P}}_Y$, the maximum likelihood problem will have a unique maximiser. The EM algorithm used by Saerens et al. [101] is, therefore, only a means to maximise the likelihood and not a key feature of the method.

To find the ratios $w_i = \frac{\mathbb{Q}_Y(i)}{\mathbb{P}_Y(i)}$ instead of the proportions α_i, Garg et al. [52] reformulated the problem as follows:

$$\tilde{w} := \underset{w \in \mathcal{W}}{\operatorname{argmax}} \sum_{k=1}^{m} \log \sum_{i=1}^{c} \mathbb{P}(y = i|x_k)w_k. \tag{2.20}$$

And in the case where the classifier f is used:

$$\tilde{w}_f := \underset{w \in \mathcal{W}}{\operatorname{argmax}} \sum_{k=1}^{m} \log(f(x_k)^T w) \tag{2.21}$$

The work of Alexandari et al. [2] has shown the practical superiority of MLLS over BBSE when the estimator f used is well-calibrated, see Definition 2.2.

Under certain technical assumptions, Garg et al. [52] obtained a convergence theorem for well-calibrated classifiers f.

Theorem 2.3 ([52]). *For any calibrated classifier f, we have:*

$$\|\tilde{w}_f - w\| \leq \frac{\mathcal{O}(m^{-1/2})}{\min_i \ \mathbb{P}_Y(i) \ \sigma_f},$$

with σ_f being the smallest eigenvalue of the matrix $\mathbb{E}_{\mathbb{Q}}[f(X)f(X)^T]$ and m the number of points in the target.

The absence of dependence on the number of points n in the source may be surprising. In reality, this dependence arises from assuming the classifier is calibrated. By using Lemma 5 of the same article, we obtain another convergence theorem, that take into account the *post-hoc* calibration.

Corollary 2.1 ([52]). *For any classifier f after a post-hoc calibration procedure, we have:*

$$\|\tilde{w}_f - w\| \leq \frac{\mathcal{O}(m^{-1/2}) + \mathcal{O}(n_v^{-1/2})}{\min i \ \mathbb{P}_Y(i) \ \sigma_f},$$

where n_v is the number of points used during the calibration procedure ($n_v < n$).

Throughout the thesis we have been able to test our methods in different settings (simulated data, real data, image data) and to compare them with some of the methods presented in this chapter. Provided that an appropriate calibration method is used, for instance the bias-corrected calibration in Alexandari's paper, MLLS outperforms the other methods we were able to test. This method also has the advantage of being fast (without taking into account the training time) and of not being affected by the curse of dimensionality, since it operates on the output space of the classifier of dimension the number of classes.

This result has also been observed in previous experiments: Alexandari et al. [2], Moreo et al. [86] and, to a lesser extent, Schumacher et al. [103].

However, more recent work based on ensemble methods, which we present in Section 2.2.6, was able to outperform MLLS in a quantification contest [34], and the method of Moreo et al. [85] KDEy that we presented in Section 2.2.2 combined with the KL divergence (this method called KDEy-ML can be seen as a natural extention of MLLS) was also shown to perform better.

2.2.4 Classifier trained for Quantification

The previous methods used a classifier trained on the source distribution and applied a *post-hoc* correction method to estimate proportions by correcting the biases of classify and count. Other *ante-hoc* methods from the *imbalance classification literature* [12, 55, 56, 120], although less explored in the quantification literature, aims to train a classifier that performs

well regardless of the proportions in the target. As previously said, no comparative study has been conducted yet to determine if the methods from this literature yield good quantifiers.

In this section, we will focus on *ante-hoc* methods that aim to train classifiers with a loss function designed for quantification. We have said that **CC** only works if the confusion matrix is the identity matrix, in other words, if the classifier is perfect. However, according to Equation (2.5), another scenario is possible: if the vector of proportions we want to estimate, α^*, is an eigenvector associated with the eigenvalue 1 of the confusion matrix $C_{\hat{y}|y}$.

In the case of binary classification, we can give a concrete meaning to this scenario: if $\mathbb{P}(f(x) = 1|y = 0) = \mathbb{P}(f(x) = 0|y = 1)$ then $\alpha^* = C_{\hat{y}|y}\alpha^*$ and therefore $\alpha_{cc} = \alpha^*$.

Indeed, even with many classification errors, if the number of false positives is equal to the number of false negatives, then the estimated proportions will be accurate. Here we see the global nature of the information sought in quantification.

The algorithms presented in this section are based on this idea, we are looking for a classifier that does not minimise classification errors, but rather compensates them.

Quantification Trees

Quantification Trees, and by extension Quantification Forest, is a family of methods proposed by Milli et al. [82] as an adaptation of Decision Trees, where the splitting criterion is designed for quantification.

Two criteria were proposed. The first one, called "Classification Error Balancing", tries to balance the number of false positive and false negative in a split. For a given subset of points S, for each class $y \in \mathcal{Y}$, we note $\mathrm{FP}_y(S)$ the number of false positive in S and $\mathrm{FN}_y(S)$ the number of false negatives. The criterion for a split is given by :

$$E(S) = \|E_y(S)\|_2^2, \text{ where } E_y(S) = |\mathrm{FP}_y(S) - \mathrm{FN}_y(S)|$$

The second, "Classification-Quantification Balancing", is a mixture of a quantification criterion and a classification criterion:

$$\bar{E}(S) = \|\bar{E}_y(S)\|_2^2, \text{ where } \bar{E}_y(S) = |\mathrm{FP}_y(S) - \mathrm{FN}_y(S)| \times |\mathrm{FP}_y(S) + \mathrm{FN}_y(S)|$$

$|\mathrm{FP}_y(S) - \mathrm{FN}_y(S)|$ measures the quantification error, while $|\mathrm{FP}_y(S) + \mathrm{FN}_y(S)|$ measures the classification error.

Just as we obtain random forests from decision trees by using subsets of features and subsets of data to create multiple "independent" trees, the authors have proposed quantification forests. For each subset, a quantification tree is calculated and the proportions of each tree are averaged to obtain the estimate.

Using Non-decomposable measures

One way to make a classifier suitable for quantification is to replace the loss function used during training with a loss more suitable for quantification, in Section2.4.1 we present such losses. However, training a classifier with these losses is complex because the metrics are not additive with respect to the points (x_i, y_i). For the reasons explained above, a classification error on a point can lead to an improvement in the quantification error. This type of error is referred to as multivariate performance measures or non-decomposable measures.

$$\Delta(f) = \Delta((f(x_1), \cdots, f(x_n)); (y_1, \cdots, y_n)) \tag{2.22}$$

Examples in this category include the F1-score, the ROC area or the f-divergence calculated on the class proportions.

Training a classifier on this type of error is non-trivial, but several papers in the literature have addressed the problem. Joachims [72] proposed a polynomial time optimisation method for an SVM on this type of measurement and direct application of this SVM for quantification was proposed by Esuli et al. [38]. In their paper, the authors chose the Kullback-Leibler divergence between the estimated proportions and the true proportions as a non-decomposable measure. Note that an extension of this work could be to use any of the metrics presented in Section 2.4.1, as long as it can be expressed in terms of the confusion matrix. Barranquero et al. [5] has proposed a variant of this method. Based on the idea that a good quantifier should also be a good classifier, Barranquero and his co-authors proposed a new measure, called the Q-measure, which balances classification and quantification errors. More precisely, if we denote *cperf* as the classification error and *qperf* as the quantification error, then the Q-measure is given by

$$Q_\beta = (1 + \beta^2) \frac{\textit{cperf} \times \textit{qperf}}{\beta^2 \textit{cperf} + \textit{qperf}}, \tag{2.23}$$

with β the trade-off parameter. Unfortunately, the SVM proposed by Joachims does not scale well with either the number of classes or the number of points.

Narasimhan et al. [88] have shown that for certain non-decomposable measures, such as the F1 score, an iterated stochastic gradient descent algorithm converges (exponentially fast in the number of iterations) to a minimiser. However, this result is restricted to the binary case and, as with Joachims, to linear classifiers.

More recently, Sanyal et al. [102] proposed an optimisation method, inspired by Narasimhan's method, for neural networks. The idea of this method is to split the network into two sets of layers. The first layers of the network are trained only once, while Narasimhan's method is applied to the last layers of the network. For example, if we restrict ourselves to the last layer, then Sanyal's procedure simply consists of applying Narasimhan's method with a data transformation step performed by the network.

2.2.5 Non-aggregative methods

So far, all the methods presented here have used a classifier in one way or another. There are two reasons why there are so many aggregative methods, i.e. methods that aggregate the outputs of a classifier. On the one hand, classification is a very classical problem with a rich and well-defined literature, since quantification is a sub-problem of classification, it is logical that the development of quantification started with modified classification methods. On the other hand, one of the main motivations for quantification, as discussed in the introduction, is to improve classification by taking into account the proportions. This implies that a pre-trained classifier is available, and therefore using it as a quantifier is probably the best solution in terms of computational complexity.

As Esuli et al. pointed out in their book "Learning to Quantify" [32], classification is a more "complex" problem than quantification in the sense that every classifier is a quantifier, but the reverse is not true. The training step of a classifier could therefore be a superfluous method to obtain proportions.

In this section we present methods that do not rely on a classifier.

Hellinger Distance-x

A similar approach to HDy, also proposed by González-Castro et al. [59], is to compute histograms directly from the data and to solve:

$$\hat{\alpha} = \underset{\alpha \in \Delta^c}{\operatorname{argmin}} \ \mathsf{HD}\left(\sum_{i=1}^{c} \alpha_i P_i, Q\right) \qquad (\mathcal{HD}x)$$

where P_i is the histogram of class i and Q is the histogram of the target. For multidimensional data we have the same problem as for $(\mathcal{HD}y)$ and Dys, and the same discussion applies here.

The proposed methodology has two defects. First, the authors only proposed this method for the two-class setting, and therefore did not propose a minimisation procedure to solve $(\mathcal{HD}x)$ other than an extensive search on $[0, 1]$, which does not scale well with the number of classes. Secondly, this method depends on the number of bins used to compute the histograms. This hyperparameter has a big impact on the results, but the authors did not propose a criterion for choosing it. Instead, they proposed an ensemble procedure to aggregate the results, which we detail in Section 2.2.6

Similar to how DyS is a generalisation of HDy for distances other than the Hellinger distance, one could propose Distribution x-Similarity (DxS) as a generalisation of HDx using other distance functions. However, to the best of our knowledge, this approach has not been proposed in the literature although the distribution feature matching approach we present in Chapter 3 of this manuscript includes this method when the distance used is the L_2 distance.

ReadMe

The method ReadMe by Hopkins et al. [69] stands out in this list, as it was specifically designed for text categorisation. However this method is dated, as it predates modern deep neural network based text processing methods.

Starting from a dictionary of size K, a text is transformed into a binary vector S of $\{0,1\}^K$, where $S(D)_i = 1$ if the i-th word from the dictionary is present in the text. Analogous to the calculation done to get BBSE, we get the following equations:

$$\mathbb{Q}(S) = \sum_{i=1}^{c} \mathbb{P}(S|y=i)\mathbb{Q}_Y(i) \tag{2.24}$$

Here, $q(S)$ is the probability of having the profile S in the target, $\mathbb{P}(S|y=i)$ is the probability of having the profile S in documents of category i (note that under the label shift assumption, $\mathbb{P}(S|y=i) = \mathbb{Q}(S|y=i)$), and $\mathbb{Q}_Y(\cdot)$ are the proportions we want to estimate.

Due to the sparsity of the problem, and since $2^K \gg n$, it is challenging to accurately estimate these quantities when the dictionary size is large. To address this issue, Hopkins proposed a bagging strategy, where a small number of words from the dictionary is used at each iteration. Equation (2.24) is solved for this reduced dictionary. The results are then aggregated. The resulting system of equations can be solved by least squares minimisation, as proposed in Equation (2.9).

Maximum Mean Discrepancy

We recall that for any symmetric and semidefinite positive kernel k defined on $\mathcal{X}$, one can associate a Hilbert space denoted $\mathcal{H}_k$, or simply $\mathcal{H}$ when there is no ambiguity, and a "feature" mapping $\Phi : \mathcal{X} \to \mathcal{H}$ such that $\langle \Phi(x), \Phi(y) \rangle = k(x,y)$.

This mapping can be extended to the space of distributions by taking the expectation:

$$\Phi\colon \mathbb{P} \mapsto \Phi(\mathbb{P}) := \mathbb{E}_{X\sim\mathbb{P}}[\Phi(X)] \in \mathcal{H}. \tag{2.25}$$

This embedding is called Kernel Mean Embedding (KME). We refer the reader to Muandet et al. [87] for a survey on KME. Note that it is sufficient to have $\mathbb{E}_{\mathbb{P}}[\sqrt{k(X,X)}] < \infty$ to ensure the existence of $\Phi(\mathbb{P})$, see Smola et al. [109].

An important property of this mapping is that, even though we do not have direct access to it because it is an infinite dimensional vector, we can still compute scalar products between mappings using the formula $\langle \Phi(\mathbb{P}), \Phi(\mathbb{Q}) \rangle_{\mathcal{H}} = \mathbb{E}_{(X,Y)\sim\mathbb{P}\otimes\mathbb{Q}}[k(X,Y)]$, this property is known in the machine learning literature as the kernel trick.

The function :

$$D_\Phi(\mathbb{P}, \mathbb{Q}) = \|\Phi(\mathbb{P}) - \Phi(\mathbb{Q})\|_{\mathcal{H}},$$

is a pseudo-distance on the space of measures on $\mathcal{X}$, called the *Maximum Mean Discrepancy* or MMD [62]. Using the kernel trick, we can compute the MMD between two distributions:

$$D_\Phi^2(\mathbb{P}, \mathbb{Q}) = \mathbb{E}_{\mathbb{P},\mathbb{P}}[k(X, X)] + \mathbb{E}_{\mathbb{Q},\mathbb{Q}}[K(Y, Y)] - 2\mathbb{E}_{\mathbb{P},\mathbb{Q}}[k(X, Y)].$$

In the literature, we found two articles dealing with quantification using kernel methods. Iyer et al. [70] used the Gaussian kernel defined as

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right),$$

and Kawakubo et al. [73] used the energy kernel defined as

$$k(x, y) = \|x\| + \|y\| - \|x - y\|,$$

which is indeed a kernel (see Sejdinovic et al. [105]).

However, these methods can be generalised to any kernel. In particular, to deal with the quadratic complexity of kernels methods we propose in Chapter 3 to use Random Fourier Features.

Both Iyer and Kawakubo proposed to use the MMD distance to search for the mixture of the source embeddings that is closest to the embedding of the target. More precisely, Kernel-based method for quantification solve the following minimisation problem:

$$\hat{\alpha} = \underset{\alpha \in \Delta^c}{\operatorname{argmin}} \ \mathsf{MMD}\left(\sum_{i=1}^{c} \alpha_i \mathbb{P}(x|y = i), \mathbb{Q}_X(x)\right) \tag{2.26}$$

We can minimise (2.26) using any available QP solver, as described in Section 3.4.1, and so this method is suitable even when the number of classes is large.

Regularized Wasserstein distances

Freulon et al. [49] proposed to do quantification using the Wasserstein distance. This method is equivalent to the one presented above, except that we replace the Maximum Mean Discrepancy by the Wasserstein Distance. The estimation is given by the following optimisation problem:

$$\hat{\alpha} = \min_{\alpha \in \Delta^c} \ W_2\left(\sum_{i=1}^{c} \alpha_i \hat{\mathbb{P}}(x|y = i), \hat{\mathbb{Q}}_X(x)\right) \tag{2.27}$$

where $\hat{\mathbb{P}}(x|y = i)$ and $\hat{\mathbb{Q}}_X(x)$ denote the empirical distributions.

Two problems arise when we want to solve (2.27). First, computing the Wasserstein distance has a complexity of $\mathcal{O}(n^3 \log(n))$, which makes it impractical for large data, even more so than kernel methods. Secondly, for two classes, a simple grid search is sufficient to find the minimum, but if one wants to apply the method in a multi-class setting, an efficient minimisation procedure is necessary.

Freulon and his co-authors solve these problems by replacing the Wasserstein distance by the entropic regularization of W_2, introduced by Cuturi [25], because the optimisation

problem can be solved efficiently on a GPU. The entropic regularization smooths the original formulation of the Wasserstein distance. More precisely, with the same notation as in Equation (2.16) the distance between two distributions a and b is defined as:

$$W^{\varepsilon}(a,b) = \inf_{M \in \Pi(\gamma,\sigma)} \sum_{i,j}^{n,m} \|a_i - b_j\|_2 M_{i,j} + \varepsilon H(M), \tag{2.28}$$

where H is the entropy function defined as $H(M) = \sum_{i,j}(\log(M_{i,j}) - 1)M_{i,j}$.

Based on this definition, Freulon proposed a minimisation procedure that is also applicable in the multi-class setting. Without going to much into the details, simply note that the regularised Wasserstein distance between two distributions a and b can equivalently be written as:

$$W^{\epsilon}(a,b) = \max_{u} \ \mathbb{E}_{Y \sim b}[g_{\varepsilon,a}(Y,u)], \tag{2.29}$$

for a certain function $g_{\varepsilon,a}$. From this, Freulon and his co-authors propose two minimisation procedures to solve (2.27), both involving stochastic gradient descent.

Note that W^{ε} does not define a distance between distributions. An extension of the work of Freulon et al. as mentioned by the authors, could be to use the Sinkhorn Divergence defined as:

$$S_{\varepsilon}(a,b) = W^{\varepsilon}(a,b) - W^{\varepsilon}(a,a) - W^{\varepsilon}(b,b), \tag{2.30}$$

which has all the desired property: "positivity, convexity, metrization of the convergence in law and scalability to large datasets" [43]. In particular, it can be shown (see Ramdas et al. [99]) that the Sinkhorn divergences interpolate between W_2 and the MMD distance associated to the energy kernel we defined above.

Deep Quantification Networks

The use of a deep neural networks for quantification has already been presented in this chapter. For instance, Sanyal et al. [102], based on the work of Narasimhan et al. [88], proposed to train a deep neural network with a loss function designed for quantification, see Section 2.2.4. The classifier-based methods presented in Sections 2.2.1, 2.2.2 and 2.2.3 can all use a neural network as their classifier and in particular, MLLS (Section 2.2.3) has been shown to be particularly efficient when used with a calibrated deep neural network, see [2, 52]. Moreover, in Chapter 3, we present a framework that embeds the distribution using the penultimate layer of a neural network trained for classification. However, none of the methods discussed so far were architectures specifically designed for quantification.

Motivated by the success of deep learning in machine learning, Qi et al. [96] proposed *Deep Quantification Networks* (DQN), a framework to use the expressive power of neural networks to perform quantification without the unnecessary step of classification.

Starting with a source data set $\{x_i\}_{i=1}^{n}$, the algorithm first performs a splitting strategy to create a set of K bags, each containing n/K points. Then, for each point of each bag,

a "feature extraction" function is used to embed the data in a high-dimensional space. This embedding can be done either by defining an architecture to be trained during the training of the DQN, or alternatively by fine tuning a pre-existing neural network. The third step is to aggregate the embedded data using, for example, a max, a mean or a median. With these 3 steps, each bag is now represented by a single vector, using a fully connected layer and a softmax, the model outputs a prediction of the proportions in the bag as if the bags were single points in a classical classification problem.

The two key elements of DQN are the splitting strategy and the aggregation method. We come back to the splitting strategy in Section 2.4.1 because this issue also arises in the design of evaluation protocols for quantification, and we focus here on the choice of aggregation method. As pointed out by Pérez-Mon et al. [95], this layer is part of a wider literature on the design of neural networks for use on *sets* [75, 132], where a *set* is an unordered, arbitrarily long sequence of data points $\{x_i\}$. To do this, the authors of this literature use a *permutation-invariant* layer and a pooling strategy to reduce the embeddings of each element of the set to a single embedding that can be processed with a dense fully connected layer and a softmax. The simplest form of *permutation-invariant* are the max, mean and median proposed by Zaheer et al. [132] and used by Qi et al. for their DQN, but Lee et al. [75] proposed to use an attention mechanism that has not been used for quantification yet.

For quantification, Pérez-Mon et al. proposed a different approach. They use *differentiable histogram layers*, a type of *permutation-invariant* layer proposed by Yusuf et al. [130], which allows the histograms of each feature to be approximated by a differentiable function that can be trained by back-propagation. Their method HistNetQ was shown to outperform MLLS in a quantification competition [35].

2.2.6 Aggregation of quantifiers

As we have seen in the previous sections, we have a wide choice of algorithms at our disposal. One way to harness this diversity is to use different quantifiers and aggregate the results. This idea has already been presented in previous sections, for example Forman's Median Sweep Threshold policy (Figure 2.3) for ACC consists of taking the median of the proportions obtained for each threshold. To choose the number of bins in $(\mathcal{HD}x)$ (or DyS), since we do not have a theoretical criterion, the authors proposed to use different numbers of bins and then aggregate the results with the median.

However, the two methods presented have aggregated the results because they lack a criterion for choosing the hyperparameters and not so much to exploit the diversity of the methods available. This is not the case for quantification forests, presented in Section 2.2.4, that proposed to train multiple quantification trees with different subsets of features and samples, and then average the results of each tree to estimate the proportions similar to the classical classification forest.

In this section, we present quantification methods that use multiple quantifiers.

Ensemble models for quantification

Ensemble methods in classification are techniques that combine multiple individual machine learning models to improve the overall performance of the classifier. In this family of methods we count for instance Gradient-boosted trees, Random Forest or Bagging methods in general. This kind of method is widely known to be efficient, for instance XGBoost [21] is the winner of many data challenges on Kaggle.

Pérez-Gállego et al. [90] proposed to adapt this kind of strategy to quantification. The principle is the same as in the classification case. Starting from a source sample, we generate a list of proportion vectors $\beta^1, \cdots, \beta^k$, where k is the number of different models in the ensemble. For each of these proportions we subsample bags from the source enforcing the sampled proportions to be equals to β^i. Each model is then train on its bag. On the target the final estimate is simply the mean of the estimate of each model.

In their papers Pérez-Gállego et al. proposed to use CC (2.4), ACC (2.7) and HDy $(\mathcal{HD}y)$ but we could use other methods presented in this chapter.

This simple approach was enhanced by the same authors [89] with a dynamic strategy to aggregate the outputs of each model. The key idea, is to only use for prediction the classifier that were trained with bags that had proportion vectors β^i close to the target proportions. As it is unknown, the model first estimates $\hat{\alpha}$ using all the models, then ranks the models according to $D(\hat{\alpha}, \beta^i)$ where D is a suitable distance or divergence, and finally only uses for the aggregating phase $\sigma\%$ of the model, where σ is an hyperparameter.

This method therefore depends on two key elements: a way to sample proportions β^i and a distance D to compare the proportions. We come back to this in Section 2.4.

MC-SQ and MC-MQ

Donyavi and her co-authors [28] proposed an extension of this approach with two algorithms: **MC-SQ** for *multiple classifier, single quantifier* and **MC-MQ** for *multiple classifier, multiple quantifier*. The idea this time is not to split the source into different bags, but to use k different classifiers on the source, and then use l aggregative quantifier algorithms, each with all the classifiers leading to $k \times l$ estimates, which are aggregated using the median (but it could be any aggregation function) to get the final results.

QuaNet: Deep learning for Quantification

Esuli et al. [37] designed QuaNet, a recurrent architecture that takes as input a sample from the target distributions and outputs the estimated proportions.

To do so, they first train a deep neural network f on the source distributions. For a target sample, QuaNet first compute the embeddings of all points (namely the penultimate layer of f) that they sort using the approximation given by the network of $\mathbb{P}(y = 1|x)$. QuaNet interprets this list as a time series indexed by the probability that the embeddings belong

to the first class. The data pass through a bidirectional LSTM [60] which outputs a high-dimensional vectorisation. Meanwhile, using the network f, they estimate the proportions using the method based on multiple classifiers that we presented: CC (2.4), ACC (2.7), PCC (2.13) and PACC.

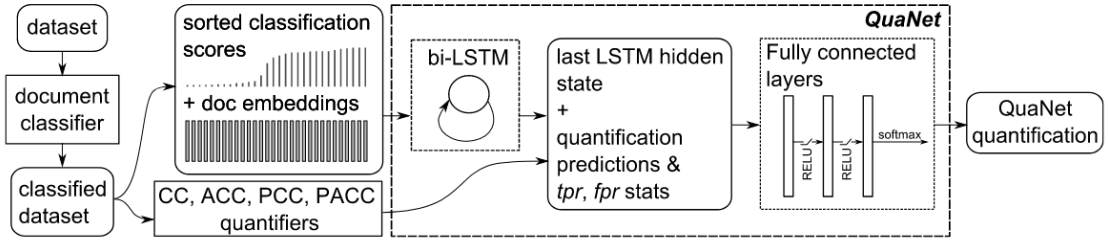


Figure 2.9: Architecture of QuaNet, taken from [37].

The embedding and output of the quantifiers then pass through a dense neural network. Figure 2.9 shows the QuaNet architecture.

QuaNet and DQL both have in common the use of a dedicated neural network architecture for quantification, but while DQL is a non-aggregative method that learns to embed the data to perform quantification, QuaNet aggregates the results of multiple classifier-based quantifiers.

QuaNet has a number of shortcomings that we would like to highlight. First, the method is only suitable for the binary case, as is the case for a number of methods in this section. Second, training such a model is complex, since the data the model takes as input is not a sample from the source, but a list of distributions, this implies having access to a large number of data sets, as it is a well-known shortcoming of neural networks that they require large data sets to be effective. Note that in this case we need at least one large dataset to train f, but also a large number of datasets to train the LSTM. Alternatively, if we only have one source sample, we can split the sample in half, use one to train f, and with a resampling scheme to generate multiple prevalence on the second sample, we can train the LSTM with the same strategy as in DQL. Third, the quantifiers used as input to the LSTM are too redundant and include two methods that are widely known to be ineffective: CC and PCC. However, this can easily be remedied by using some of the many methods we have presented.

2.3 Unification and joint analysis

Several works in the literature have attempted to unify the various methods presented in the previous sections under a common framework.

In the quantification literature, Firat [44] was the first to notice that most of the methods presented in the previous section can be recast as a constrained multivariate regression task. He uses this new framework to extend existing binary quantifiers to the multiclass setting.

This approach defines two objects: a *feature transformation function* and a *loss function*. This approach is at the heart of the Python package $\mathrm{qunfold}$ (Bunse [14], 2023). However, this framework was not proposed to derive a common analysis, which explains why the framework is so general. In fact we will see that it encapsulates almost all the methods we have presented.

Bunse [13] (2022) showed the link between quantification and a physic problem called *unfolding* that we will not present here. His work is closely related to that of Firat, with the addition of a regularising term. Once again, this work was not done to derive a common theoretical analysis.

More recently, Garg et al. [52] were the first to propose a theoretical study of MLLS, which we partly presented in Section 2.2.3. In their analysis, they presented a common framework that regroup both BBSE and MLLS to explained why MLLS gives better results than BBSE, we present their work in Section 2.3.4.

Finally, we unify several methods from the literature, including BBSE, ACC, and the Maximum Mean Discrepancy-based method, and derive a common analysis in Chapter 3.

2.3.1 Distribution Matching

The unifications proposed by Firat and Bunse are based on what we will call distribution matching.

In Section 2.2.3, we presented quantification as a parametric model (2.18). The goal of distribution matching is to find the distribution $\mathbb{P}_\alpha$ that is closest to $\mathbb{Q}$ with respect to any discrepancy or divergence. To do this, we define a feature mapping $\Phi\colon \mathcal{M}(\mathcal{X}) \mapsto \mathcal{Z}$ that maps any distribution on $\mathcal{X}$ to any space $\mathcal{Z}$ and a distance $\mathcal{D}$ that has all the desired properties (minimal for α^*, convex, fast to compute and minimise) to solve:

$$\min_{\alpha \in \Delta^c} \mathcal{D}(\Phi(\mathbb{P}_\alpha), \Phi(\mathbb{Q})). \tag{2.31}$$

Let us see how we can rewrite some of the previous methods as (2.31).

Adjusted Classify and Count Let us assume that the classifier f is not probabilistic. In this case, let $\phi\colon x \mapsto (\mathbf{1}\{f(x) = i\})_{i=1}^c$, and for a distribution $\mathbb{P}$, denote $\Phi(\mathbb{P}) := \mathbb{E}_{\mathbb{P}}[\phi(X)]$. Then, $\Phi(\mathbb{P}(x|y = i))_j = \mathbb{P}(f(x) = j|y = i)$, which implies that $\sum_{i=1}^c \alpha_i \Phi(\mathbb{P}(x|y = i)) = \alpha C_{\hat{y}|y}$.

The Distribution Matching method associated with this function Φ and the L_2 norm corresponds to the formulation (2.9) of Adjusted Classify and Count.

The method proposed by Azizzadenesheli et al. [4] is a variant of BBSE where the function $\mathcal{D}$ is the L_2 norm and a penalisation term, so this method can also be cast as a distribution matching procedure.

Forman's threshold policy The threshold policy consists in applying to the source an ACC with a classifier f, whose threshold has been optimised according to a criterion

(see Figure 2.3). The threshold policy is therefore a distribution matching method with an ante-hoc procedure, which hypothetically gives a worse classifier but a better quantifier.

Probabilistic Adjusted Classify and count This setting is the same as the one we described above, except that the function Φ is now the output of a soft classifier: $\phi(x) = f(x)$.

Hellinger distance x and y The methods proposed by González-Castro et al. [59] use as their names indicate, the Hellinger distance between either histograms of the outputs of the classifier $(\mathcal{HD}y)$ or histograms of the data directly $(\mathcal{HD}x)$. Thus, they are distribution matching methods with $\mathcal{D}$ the Hellinger distance and Φ the histogram function of the classifier outputs or the data.

Dys proposed by Maletzke et al. [79] is a variation of HDy where they change the distance function $\mathcal{D}$.

Kernel Density The methods proposed by Moreo et al. [85] embed the data using a density estimation of the output of a classier f. The authors proposed to use 3 distances D: the Hellinger distance, the Cauchy-Schwarz divergence and the KL divergence. However, the framework can be generalised by using any distance D, such as a f-divergence or the L_2 norm.

SORD The SORD method, also proposed by Maletzke, consists, as we have explained, in minimising the Wasserstein distance between the outputs of the classifier, see (2.17). This directly gives us our functions Φ and $\mathcal{D}$.

Kullback–Leibler divergence We presented MLLS (Section 2.2.3) as to find the parameter α that maximise the likelihood. It is a well-known fact in statistics that maximising the likelihood of a model is equivalent to minimising the Kullback-Leibler divergence between the real distribution and the probability defined by the parameter α.

$$
\begin{aligned}
KL(\mathbb{Q}, \mathbb{P}_\alpha) &= \int_{\mathcal{X}} \mathbb{Q}_X(x) \ \log\left(\frac{\mathbb{Q}_X(x)}{\mathbb{P}_\alpha(x)}\right) dx \\
&= \underbrace{\int_{\mathcal{X}} \mathbb{Q}_X(x) \ \log\left(\mathbb{Q}_X(x)\right) dx}_{C_1} - \int_{\mathcal{X}} \mathbb{Q}_X(x) \ \log\left(\mathbb{P}_\alpha(x)\right) dx \\
&= C_1 - \int_{\mathcal{X}} \mathbb{Q}_X(x) \ \log\left(\sum_{i=1}^{c} \alpha_i \mathbb{P}(x|y=i)\right) dx \\
&= C_1 - \int_{\mathcal{X}} \mathbb{Q}_X(x) \ \log\left(\sum_{i=1}^{c} \alpha_i \frac{\mathbb{P}(y=i|x)\mathbb{P}_X(x)}{\mathbb{P}_Y(i)}\right) dx \\
&= C_1 - C_1' - \int_{\mathcal{X}} \mathbb{Q}_X(x) \ \log\left(\sum_{i=1}^{c} \alpha_i \frac{\mathbb{P}(y=i|x)}{\mathbb{P}_Y(i)}\right) dx.
\end{aligned}
$$

Replacing $\mathbb{Q}$ with the empirical distribution of the target sample yields Equation (2.19) up to additive term. A similar calculation using the parametrisation $\mathcal{M}_{\mathcal{W}}$ yields Equation (2.20) up to additive term.

ReadMe ReadMe [69] was specifically design for text categorisation (2.24). The embedding function Φ is a vector of dimension K, where $\Phi(x)_i = 1$ if the i-th word of a dictionary is present in the document x and the function $\mathcal{D}$ is as in BBSE the L_2 norm.

Maximum Mean Discrepancy The methods proposed by Iyer et al. [70] and Kawakubo et al. [73] both used the kernel methods (the Gaussian kernel and the energy kernel, respectively) to perform quantification. We extend their methods to any kernel, including kernels obtained with Random Fourier Features in Chapter 3.

The function Φ associated with these methods is the kernel mean embedding and the distance is the Hilbert distance of the corresponding RKHS.

Regularised Wasserstein distance Freulon et al. [49] proposed to use a regularised Wasserstein distance directly on the data, as opposed to SORD which did it on the outputs of a classifier and with the classical Wasserstein distance. In this case, $\mathcal{D}$ is the regularised Wasserstein distance and Φ is the identity function.

If we do the alternative described by Freulon et al., using the Sinkhorn divergence instead of the regularised Wasserstein distance, then we are again in the distribution matching framework, where $\mathcal{D}$ is the Sinkhorn divergence [43].

2.3.2 Distribution Feature Matching

In Chapter 3 we study a general framework that we call *Distribution Feature Matching* (DFM). DFM is a special case of Distribution Matching where the distance function is an Hilbertian norm and the embedding function Φ is of the form $\Phi(\mathbb{P}) = \mathbb{E}_{\mathbb{P}}[\Phi(x)]$. This method unites under the same analysis the Maximum Mean Discrepancy methods of [70] and [73], the methods that use the confusion matrix of a classifier, the method of Hopkins et al. for text classification and a variation of the method of Moreo et al. [85], who propose to use a KDE on the output of the classifier f if we choose $\mathcal{D}$ as L_2 norm. Moreover, the error bounds obtained are better than those known in the literature, namely those of Iyer [70] and Lipton [77].

We provide theoretical guarantees on the estimation error of the proportions for all DFM methods in Chapter 3. See Table 2.1 for an overview of the Distribution Matching framework.

Method	Embedding function Φ	Distance function $\mathcal{D}$	DFM
BBSE	$\mathbf{1}\{f(x) = i\}$	L_2	Yes
Threshold policy	$\mathbf{1}\{f(x) = i\}$	L_2	Yes
PACC	$f(x)$	L_2	Yes
HDy	Histogram of $f(x)$	Hellinger distance	No
HDx	Histogram of x	Hellinger distance	No
Dys	Histogram of $f(x)$	Any distance	No
Dys	Histogram of $f(x)$	L_2	Yes
FMM	cumsum of the cdf of f	L_1	No
SORD	$f(x)$	W_2	No
KDEy	KDE of $f(x)$	Multiple Choice	No
KDEy	KDE of $f(x)$	L_2	Yes
ReadMe	Special embedding	L_2	Yes
Kernel methods	KME	MMD	Yes
Wasserstein	Identity function	Regularised W_2	No
Sinkhorn	Identity function	Sinkhorn	No

Table 2.1: Overview of the Distribution Matching framework.

With this restricted definition of Distribution Matching we were able to obtain theoretical results on the convergence of the estimator. See Theorem 3.1.

2.3.3 Methods that do not fit the framework

Some of the methods we presented in Section 2.2 were not categorised as distribution matching methods. Let us review these methods.

Classify and Count Actually Classify and Count is a distribution matching method, where $\Phi(\mathbb{P})_i = \mathbb{E}_{\mathbb{P}}[\mathbf{1}\{f(x) = i\}]$ and $\mathcal{D}$ is the function $\mathcal{D}(a, b) = b$. With these notations (2.31) is no longer an optimisation problem because it is equal to $\Phi(\mathbb{Q}_X(x))$. We choose to exclude classify and count from this category because it does not follow the "philosophy" of Distribution matching, that is to say: find the distribution $\mathbb{P}_\alpha$ in $\mathcal{M}$ that is the "closest" to the true distribution $\mathbb{Q}$ where the definition of closest is given by the embedding Φ and the distance function $\mathcal{D}$.

Methods of section 2.2.4 These methods were designed to minimise a "quantification" criterion rather than a "classification" criterion. This ante-hoc procedure produces worse classifiers in the hope of having better quantifiers. To quantify, these methods use the classifier and a classify and count procedure.

Deep Quantification Networks DQN learns to represent a distribution as a vectorial representation that encapsulates both the source embeddings $\mathbb{P}(x|y = i)$ and the proportions of the classes $\mathbb{P}_Y(i)$ of a given bag. Once this intermediate embedding has been learned, the model outputs the probability using a fully connected layer, i.e. a linear classification and a softmax.

We can not call this a distribution matching method because the intermediate embedding $\Phi(\mathbb{Q}_X(x))$ will be no closer to $\Phi(\mathbb{P}_{\alpha^*})$ than any other $\mathbb{P}_\alpha$.

Ensemble methods Methods that aggregate multiple quantifiers, from Section 2.2.6, can not be cast as distribution matching, but the quantifiers used as an intermediate step might be.

2.3.4 Comparison of BBSE and MLLS

Garg et al. [52] proposed a method that unifies BBSE and MLLS[2]. The perspective taken by Garg and their co-authors is similar to distribution matching but more restricted to their objective.

The idea is the same as with DFM: map each point of the dataset to some *space*, embed each class by taking the mean and find the proportions by minimising a distance, but the *space* on which they embed is more restricted than the general case, since it is a probabilistic space.

In DFM we define $\Phi(x)$ as a mapping from the data space $\mathcal{X}$ to an Hilbert space $\mathcal{H}$, while in Garg et al., a point is associated with a distribution $g(x) \in \mathcal{P}(\mathcal{Z})$ in a space to be defined as $\mathcal{Z}$. They then define $\mathbb{P}(z|y = i)$ as $\int_{\mathcal{X}} g(x)\mathbb{P}(x|y = i)dx$ and $\mathbb{Q}(z)$ as $\int_{\mathcal{X}} g(x)\mathbb{Q}_X(x)dx$.

[2]They called their method: *generalized distribution matching*, but note that this is not the same as *distribution matching* and is in fact less general than DM.

If we have a classifier we can take $g = f$ because it defines a distribution over Δ^c. $\mathbb{P}(z|y)$ is now a matrix in $\mathbb{R}^{c\times c}$ equal to the confusion matrix of the classifier while $\mathbb{Q}(z)$ is now a vector equals to α_{cc} the estimation of the proportions by classify and count (Section 2.2.1).

The choice of D then remains. If we use KL we obtain MLLS and if we use L_2 we obtain BBSE.

The difference between the two methods lies in the choice of distance but also, as they pointed out, on the choice of calibration. The following theorem by Vaicenavicius et al. [123] indicates that the confusion matrix is a calibration technique.

Theorem 2.4 (Vaicenavicius et al. [123])**.** *For any function h, the function*

$$f\colon x \mapsto \mathbb{P}(y|h(x)),$$

is a calibrated classifier.

In particular, the function $\bar{f}(x) = \mathbb{P}(y|\operatorname{argmax} f(x))$ is a calibrated predictor, and this vector can be estimated by the $(\operatorname{argmax} f(x))$-th column of the confusion matrix.

To summarize, MLLS involves training a classifier on the source data, **calibrating** this estimator using an explicit post-hoc method, and then minimizing quantification using the KL distance. The BBSE method, on the other hand, involves training a classifier on the source data and using the L_2 distance while **implicitly calibrating** the classifier using the confusion matrix. Garg and their co-authors propose a hybrid method that uses the estimator calibrated by the confusion matrix but still uses the KL distance. This hybrid method, named MLLS-CM (CM for Confusion Matrix), is experimentally shown to be inferior to MLLS while being (roughly) equivalent to BBSE.

In summary, Garg et al. affirm that MLLS is better than BBSE not because of the KL distance but because the calibration method using the confusion matrix is less effective than other post-hoc methods. This was confirmed in their experiments as well as those of Alexandari et al. [2] that show empirically that MLLS outperforms BBSE for a good choise of calibration, namely Bias-Corrected Temperature Scaling (BCTS).

2.4 Evaluation of quantifiers

As with all learning methods, quantification algorithms need to be compared through experimentation. The difficulty in evaluating them is that the estimation is not based on a single data point, as in classification, but on a complete data set. Therefore, it requires access not to a training set (the source) and a test set (the target) as in classification, but to a collection of sources and targets, with each source/target pair having to satisfy the label shift hypothesis ($\mathcal{LS}$). Moreover, unlike the classical classification framework, where we can measure the error of a classifier using the accuracy of the model, in quantification we have to define a metric to compare $\hat{\alpha}$ and α.

Two elements must be defined when designing an evaluation protocol: a metric D to evaluate the error of a single estimate, and a set of data sets that satisfy the label shift assumption ($\mathcal{LS}$), either from real applications or generated.

In this section we only focus on the protocol for comparing quantifiers in direct application (see Section 2.1.1), as the other application areas we have presented have their own evaluation protocols. First we discuss the choice of metric, and then the data generation process.

2.4.1 Evaluation metrics

This section is a reinterpretation of the work of Sebastiani [104] (also chapter 3.1 of *Learning to quantify* by Esuli et al. [32]). In their work, the authors propose an "axiomatic approach to evaluation measures for quantification", which consists in defining a set of properties (or axioms) that a good measure D should satisfy. In this section we present the axioms proposed by the author, discuss their validity, and show that these axioms are incompatible. In addition, we discuss the way in which multiple results can be aggregated, a topic that is only hinted at in his work.

Throughout this section we will refer to the measures as D and assume that we have two vectors of proportions: $\alpha, \hat{\alpha} \in \Delta^c$. The true proportion is the vector α while the estimate is the vector $\hat{\alpha}$.

Classical metrics used in Quantification.

Let us start by presenting the classical metrics used in the quantification literature. We refer the reader to the work of Sebastiani [104] (Table 2) to find out which article has used which metric.

Definition 2.3 (Absolute error **AE**)**.**

$$\mathrm{AE}(\alpha, \hat{\alpha}) = \frac{1}{c}\sum_{i=1}^{c} |\alpha_i - \hat{\alpha}_i| = \frac{1}{c}\|\alpha - \hat{\alpha}\|_1.$$

Definition 2.4 (Relative Absolute error **RAE**)**.**

$$\mathrm{RAE}(\alpha, \hat{\alpha}) = \frac{1}{c}\sum_{i=1}^{c} \frac{|\alpha_i - \hat{\alpha}_i|}{\alpha_i}.$$

Definition 2.5 (Square error **SQ**)**.**

$$\mathrm{AE}(\alpha, \hat{\alpha}) = \frac{1}{c}\sum_{i=1}^{c} (\alpha_i - \hat{\alpha}_i)^2 = \frac{1}{c}\|\alpha - \hat{\alpha}\|_2^2.$$

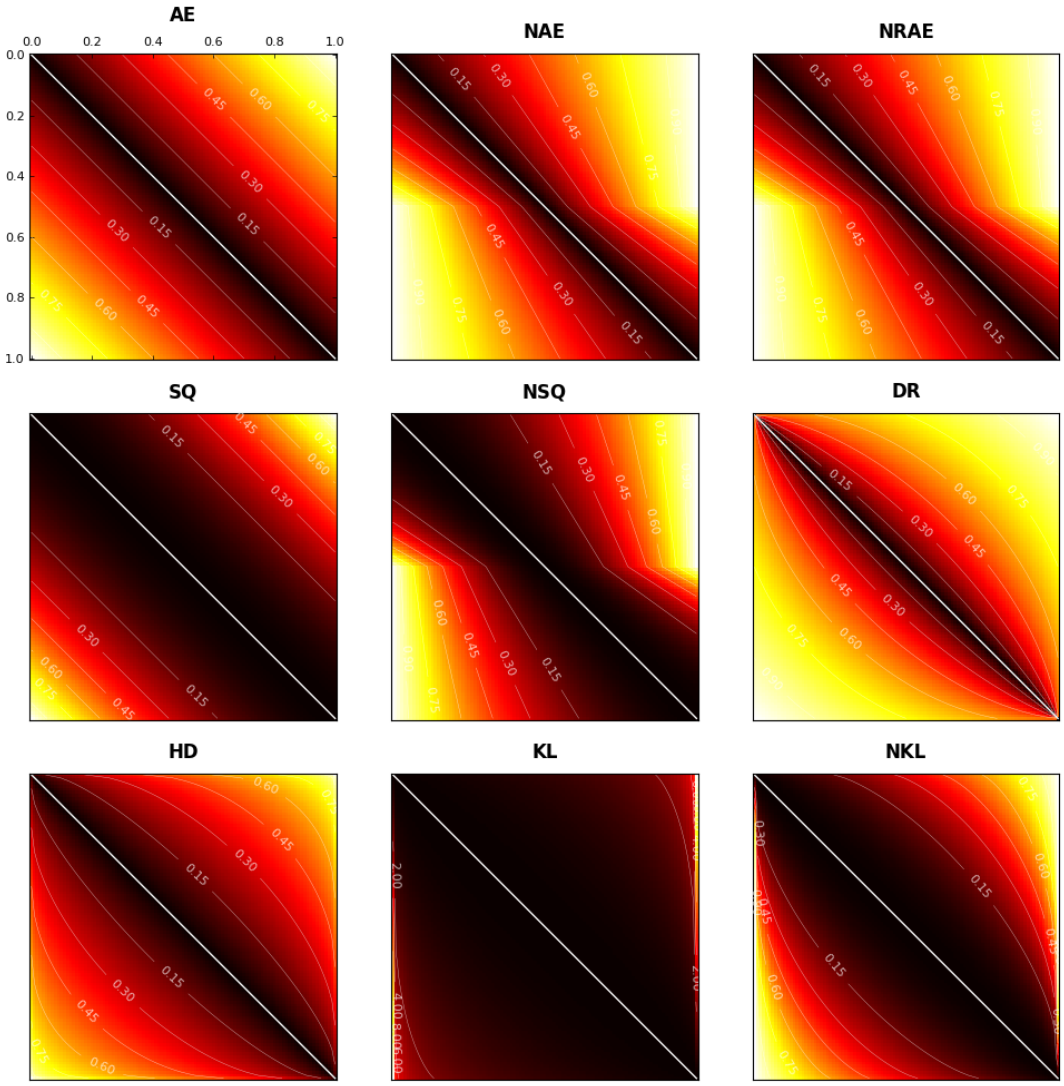


Figure 2.10: This figure, inspired by that of Sebastiani [104] (Figure 1), shows the distance between α on the x axis and $\hat{\alpha}$ on the y axis for a selection of metrics D. Darker areas represent small errors.

Definition 2.6 (Discordance Ratio **DR**).

$$\begin{aligned}\mathrm{DR}(\alpha,\hat{\alpha}) &= 1 - \frac{1}{c}\sum_{i=1}^{c}\frac{\min(\alpha_i,\hat{\alpha}_i)}{\max(\alpha_i,\hat{\alpha}_i)} \\ &= \frac{1}{c}\sum_{i=1}^{c}\frac{|\alpha_i-\hat{\alpha}_i|}{\max(\alpha_i,\hat{\alpha}_i)}.\end{aligned}$$

Definition 2.7 (Kullback-Leibler Divergence **KL**)**.**

$$\mathrm{KL}(\alpha, \hat{\alpha}) = \sum_{i=1}^{c} \alpha_i \log \frac{\alpha_i}{\hat{\alpha}_i}.$$

Definition 2.8 (Pearson Divergence **PD** or χ^2 divergence)**.**

$$\mathrm{PD}(\alpha, \hat{\alpha}) = \frac{1}{c} \sum_{i=1}^{c} \frac{(\alpha_i - \hat{\alpha}_i)^2}{\hat{\alpha}_i}.$$

Definition 2.9 (Total Variation **TV**)**.**

$$\mathrm{TV}(\alpha, \hat{\alpha}) = \sup |\alpha_i - \hat{\alpha}_i| = \|\alpha - \hat{\alpha}\|_\infty.$$

For reasons we will explain in the next section, it may be advantageous to have a metric for which the error range, i.e. the worst error an estimate of the true proportions can make, is bounded by a constant M. One way to achieve this is to "normalise" the metric D by dividing $D(\alpha, \hat{\alpha})$ by $D(\alpha, \tilde{\alpha})$, where $\tilde{\alpha}$ is the worst possible estimate. We call this the "perverse" estimate of α. This defines a new category of metrics called normalised metrics:

Definition 2.10 (Normalised Absolute error **NAE**)**.**

$$\begin{aligned}\mathrm{NAE}(\alpha, \hat{\alpha}) &= \frac{\mathrm{AE}(\alpha, \hat{\alpha})}{\max\limits_{\tilde{\alpha}} \mathrm{AE}(\alpha, \tilde{\alpha})} \\ &= \frac{\sum_{i=1}^{c} |\alpha_i - \hat{\alpha}_i|}{2(1 - \min\ \alpha_i)}.\end{aligned}$$

Definition 2.11 (Normalised Relative Absolute error **NRAE**)**.**

$$\begin{aligned}\mathrm{NRAE}(\alpha, \hat{\alpha}) &= \frac{\mathrm{RAE}(\alpha, \hat{\alpha})}{\max\limits_{\tilde{\alpha}} \mathrm{RAE}(\alpha, \tilde{\alpha})} \\ &= \frac{\sum_{i=1}^{c} |\alpha_i - \hat{\alpha}_i| / \alpha_i}{c - 1 + \frac{1 - \min\ \alpha_i}{\min\ \alpha_i}}.\end{aligned}$$

Definition 2.12 (Normalised Square error **NSQ**)**.**

$$\begin{aligned}\mathrm{NSQ}(\alpha, \hat{\alpha}) &= \frac{\mathrm{SQ}(\alpha, \hat{\alpha})}{\max\limits_{\tilde{\alpha}} \mathrm{SQ}(\alpha, \tilde{\alpha})} \\ &= \frac{\sum_{i=1}^{c} |\alpha_i - \hat{\alpha}_i|}{(1 - \min\ \alpha_i)^2 + \sum_{i=1}^{c} \alpha_i^2 - \min\ \alpha_i^2}.\end{aligned}$$

Since the perverse estimate of α gives an infinite error, we can not use the same strategy to normalise the KL divergence. What has been proposed in the literature is to use the sigmoid function.

Definition 2.13 (Normalised Kullback-Leibler Divergence **NKL**)**.**

$$\mathrm{NKL}(\alpha, \hat{\alpha}) = 2\sigma(\mathrm{KL}(\alpha, \hat{\alpha})) - 1.$$

with $\sigma\colon x \mapsto \frac{1}{1+\exp^{-x}}$, the sigmoid function.

Remark. Some of the metrics we define in this section are not defined if either α or $\hat{\alpha}$ has a zero coordinate. What has been proposed in the literature is to "smooth" the proportion by a factor ε:

$$\alpha_\varepsilon = \frac{\varepsilon + \alpha}{\sum_{i=1}^{c} \varepsilon + \alpha_i},$$

where $\varepsilon \sim 1/2n$ is taken in practice. The heuristic behind this value is that if $\hat{\alpha} = 0$ it can either mean that the true proportions are zero or that the sample is too small to have a data point from that label i.e. $\alpha_i \leq 1/n$.

The axioms

We present the seven axioms with graphs to illustrate them. Some of these are only shown for $c = 2$. In their original formulation they were shown for an arbitrary $c \geq 2$, but it was assumed that α was equal to $\hat{\alpha}$ at all but two coordinates. To keep the formulation simple (and to be able to visualise it) we restrict the definition to two classes.

Moreover, the **(IND)** property implies that if a property presented for $c = 2$ is verified by D, then it also satisfies the property in the general formulation.

Axiom 1. *Identity of indiscernible* ***(IoI)****.*
For all proportions $\alpha, \hat{\alpha} \in \Delta^c$,

$$D(\alpha, \hat{\alpha}) = 0 \iff \alpha = \hat{\alpha}.$$

Axiom 2. *Non-negativity* ***(NN)****.*
For all proportions $\alpha, \hat{\alpha} \in \Delta^c$,

$$D(\alpha, \hat{\alpha}) \geq 0.$$

Axiom 3. *Maximum* ***(MAX)****.*
There exists a value $M \in [0, +\infty[$*, such that for all proportions* $\alpha \in \Delta^c$*, there exists a "perverse" approximation* $\tilde{\alpha} \coloneqq \tilde{\alpha}(\alpha)$ *such that*

$$D(\alpha, \tilde{\alpha}) = M, \text{ and } \forall \hat{\alpha} \in \Delta^c,\ D(\alpha, \hat{\alpha}) \leq M.$$

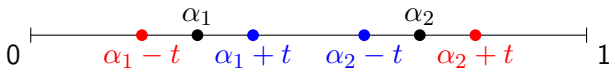
Axiom 4. *Strict monotonicity* ***(MON)****.*
For all proportions $\alpha \in \Delta^2$ *and for all values* $t \in [-\alpha_1, \alpha_2]$ *where* α_i *denotes the i-th coordinate of the vector* α*, let us write* $\alpha_t = (\alpha_1 + t, \alpha_2 - t)$*. It holds:*

$$t \mapsto D(\alpha, \alpha_t) \text{ is decreasing on } [-\alpha_1, 0],$$
$$t \mapsto D(\alpha, \alpha_t) \text{ is increasing on } [0, \alpha_2].$$

For instance:

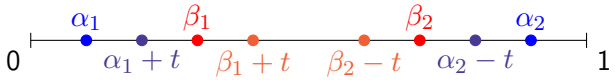
$$D(\alpha, \alpha_{-0.1}) < D(\alpha, \alpha_{-0.2}).$$

Axiom 5. *Impartiality **(IMP)**.*
For all proportions $\alpha \in \Delta^2$ and for all values $t \in [0, \min(\alpha_1, \alpha_2)]$, the approximation $\alpha_t = (\alpha_1 + t, \alpha_2 - t) \in \Delta^2$ and the approximation $\alpha_{-t} = (\alpha_1 - t, \alpha_2 + t) \in \Delta^2$ yield the same error:



$$D(\alpha, \alpha_t) = D(\alpha, \alpha_{-t}).$$

Axiom 6. *Relativity **(REL)** and Absoluteness **(ABS)**.*
For every proportions $\alpha, \beta \in \Delta^2$, that satisfy $\alpha_1 < \beta_1 \leq \beta_2 < \alpha_2$. Let us take t, such that $\alpha_t = (\alpha_1 + t, \alpha_2 - t) \in \Delta^2$ and $\beta_t = (\beta_1 + t, \beta_2 - t) \in \Delta^2$ two approximations. Then it holds:



$$D(\alpha, \alpha_t) > D(\beta, \beta_t), \quad \textbf{(REL)}$$
$$D(\alpha, \alpha_t) = D(\beta, \beta_t). \quad \textbf{(ABS)}$$

Axiom 7. *Independence **(IND)**. Let us write $C_1 \subset \{1, \cdots, c\}$ a subset of the classes.*
For every proportions $\alpha \in \Delta^c$, and every estimation $\hat{\alpha}^1, \hat{\alpha}^2 \in \Delta^c$, such that $\hat{\alpha}^1_i = \hat{\alpha}^2_i$ for every $i \notin C_1$. Then $D(\alpha, \hat{\alpha}^1) \leq D(\alpha, \hat{\alpha}^2)$ if and only if

$$D(\alpha_{C_1}, \hat{\alpha}^1_{C_1}) \leq D(\alpha_{C_1}, \hat{\alpha}^2_{C_1}),$$

where $\alpha_{C_1} \in \Delta^{|C_1|}$ are the proportions restricted to C_1. Same for $\hat{\alpha}^1_{C_1}$ and $\hat{\alpha}^2_{C_1}$.

Let us break down these axioms. The properties (**Iol**) and (**NN**) imply that D is a divergence but not a distance since D is not necessary symmetric and does not satisfy the triangle inequality. We do not see any reason why we should impose the triangle inequality on D and the symmetry is not useful since we are comparing a true distribution and an estimate, so we are free to choose whether we want to compute $D(\alpha, \hat{\alpha})$ or $D(\hat{\alpha}, \alpha)$.

Property (**MON**) means that "all other things being equal, a higher prediction error on one class (obviously matched by a higher prediction error of opposite sign on another class)

implies a higher quantification error as measured by D". [104]. Property (**IND**), only implies that we need to compare two estimates at the coordinates where they differ.

As we can see in Table 2.2, these four properties (**IoI**), (**NN**), (**MON**) and (**IND**) are satisfied by all metrics commonly used in quantification. These are primary properties, and we can not think of any cases where we would want a metric D that does not satisfy one of these properties.

The discussion in the article [104] focuses mainly on the last four properties.

Property (**IMP**) states that an underestimation or an overestimation of the same absolute value should be equally penalised. Property (**REL**) implies that making mistakes of the same absolute value on small classes is worse than making mistakes on larger classes, whereas (**ABS**) implies that the impact of errors is the same regardless of class size. There are certain scenarios where (**REL**) is desirable, when we want to penalise errors on small classes more severely. For instance, if we know in advance that the classes are highly unbalanced, such as in a flow cytometry dataset. But there are also scenarios where (**ABS**) is desired. For example, Sebastiani [104] presented a setting where we want to estimate the percentage of passengers who will show up for a flight, where overestimating or underestimating by a certain percentage would have the same cost to the airline, regardless of the true proportion of people who do not show up. There is no compelling reason to use a metric that does not satisfy any of these properties. However, in that scenario it is not clear if (**IMP**) is desirable.

Property (**MAX**) implies that the range of values does not depend on the true proportion that we aim to estimate. This property is important in settings where we want to assess the quality of a given quantifier on multiple samples, each characterized by its own true proportions and its own number of classes. For instance, consider the **absolute error (AE)**, i.e. the L_1 norm divided by the number of classes. The error range of **(AE)** is $[0, \frac{2}{c}(1 - \min\ \alpha_i)]$. The smallest range possible occurs when $\min\ \alpha_i$ is equal to $\frac{1}{c}$, and in that case, the error range is $[0, \frac{2}{c} - \frac{2}{c^2}]$. The largest possible range occurs when $\min\ \alpha_i$ is equal to 0, and in that case, the error range is $[0, \frac{2}{c}]$. This is a problem if the number of classes is small; for $c = 2$, the range changes from $[0, 1/2]$ to $[0, 1]$ depending on the true distribution.

Therefore, if we aggregate the error measured with (**AE**) by a mean or a median, the results will be unreliable as an error of $1/2 \in [0, 1]$ for a given proportion is "better" than an error of $1/2 \in [0, 1/2]$ on another. This issue is less problematic if the number of classes is large; for instance, for $c = 10$, the range changes from $[0, 0.18]$ to $[0, 0.2]$, however if we apply a mean on experiments with different numbers of classes, the results become even more confusing.

After a close examination of all the metrics used in the literature, Sebastiani [104] found that no metric satisfied all the properties at once[3] The article concluded by suggesting that "more research is needed to identify or synthesise a truly adequate such measure". Unfortunately, the main theorem we will establish in this section states that the seven axioms

[3]Note that (**REL**) and (**ABS**) are by definition mutually exclusive. When we say "satisfies all properties at once", we mean that it satisfies either (**REL**) or (**ABS**) and all the other properties.

	IoI	**NN**	**MAX**	**MON**	**IMP**	**REL**	**ABS**	**IND**
AE	Yes	Yes	No	Yes	Yes	No	Yes	Yes
NAE	Yes	Yes	Yes	Yes	Yes	No	No	Yes
RAE	Yes	Yes	No	Yes	Yes	Yes	No	Yes
NRAE	Yes	Yes	Yes	Yes	Yes	No	No	Yes
SQ	Yes	Yes	No	Yes	Yes	No	Yes	Yes
NSQ	Yes	Yes	Yes	Yes	Yes	No	No	Yes
DR	Yes	Yes	No	Yes	No	No	No	Yes
KLD	Yes	Yes	No	Yes	No	No	No	Yes
NKLD	Yes	Yes	Yes	Yes	No	No	No	Yes
PD	Yes	Yes	No	Yes	No	No	No	Yes
TV	Yes	Yes	No	Yes	Yes	No	Yes	Yes

Table 2.2: Property of the metics as reported by Sebastiani [104], the proofs or the counter-example can be found in the article. Note however that the Discordance ratio **(DR)** does not check **REL**, contrary to what they announced. See Proposition 2.1.

presented are mutually exclusive, leading to a context-dependent choice of metric.

To show this, we must first characterise for a metric D the set of "perverse estimates" i.e. the worst possible estimations.

Lemma 2.2. *If a function D satisfies (**MON**) and (**IMP**), then for given proportion vector $\alpha \in \Delta^2$:*

$$\tilde{\alpha}_i = \begin{cases} 1 & \text{if } i = \text{argmin } \alpha_i \\ 0 & \text{else} \end{cases}$$

is the perverse estimation of D.
If $\alpha_1 = \alpha_2$ both $\tilde{\alpha} = (1, 0)$ and $\tilde{\alpha} = (0, 1)$ are "perverse".

Proof. Let us suppose without loss of generality that $\alpha_1 < \alpha_2$.
Let us write $\alpha_t = (\alpha_1 + t, \alpha_2 - t)$, since $\alpha_1 \leq \alpha_2 : t \in [-\alpha_1, \alpha_2]$. By definition of **IMP** we have :

$$D(\alpha, \alpha_t) = D(\alpha, \alpha_{-t}),$$

using (**MON**), we conclude that $D(\alpha, \alpha_t)$ is maximal when $|t|$ is maximal, i.e. $t = \alpha_2$. □

We can now state and prove the main theorem of this section: The axioms of the Sebastiani [104] are mutually exclusive, i.e. no function D can satisfy (**IoI**), (**NN**), (**MON**), (**MAX**), (**IMP**), (**IND**), and (**REL**) or (**ABS**) at the same time.

Theorem 2.5 (Incompatibility of the axioms)**.** *Suppose that a metric D satisfies (**MON**) and (**IMP**). Then it can not satisfy both (**MAX**) and (**REL**) or (**MAX**) and (**ABS**) at the same time.*

Proof. Let us take $\alpha, \beta \in \Delta^2$, that satisfy $\alpha_1 < \beta_1 \leq \beta_2 < \alpha_2$, $t \in [-\alpha_1, \beta_2]$, $\alpha_t = (\alpha_1 + t, \alpha_2 - t) \in \Delta^2$ and $\beta_t = (\beta_1 + t, \beta_2 - t) \in \Delta^2$.

Let us take $t = \beta_2$, by Lemma 2.2, β_t is the perverse estimation of β and by **MAX** : $D(\beta, \beta_t) = M$.

1. (**REL**) is not possible because of (**MAX**) as $D(\alpha, \alpha_t) \leq M$.
2. (**ABS**) is not possible as α_t is not the perverse estimation of α.

□

Theorem 2.5 establishes that the seven axioms presented are mutually exclusive, leading to a context-dependent choice of metric. Note, however, that the property (**MAX**) is not required if we compute the error on only one experiment. In other words, (**MAX**) is the only property that is not relative to the metric itself, but is relative to the metric when we compare results on different experiments. In a sense, this is a "global" or "aggregation" property, while the other properties are "local".

Therefore, we can argue that this "global" property is not as important as the other properties. If we only do one experiment, the question does not arise. If we have several experiments, we still can choose to aggregate the results using the arithmetic mean because the problem posed by the lack of (**MAX**) may not be as important as [104] suggests. For example, with the absolute error (**AE**), the range of error between two true distributions is at worst twice as large and if the number of classes is high the range is almost null. Such variations are not catastrophic for the arithmetic mean that we compute.

If we still want to mitigate this problem, we can rely on a different aggregating strategy than the arithmetic mean. What we propose to overcome Theorem 2.5 is to take a metric that satisfies all the desired properties (namely (**REL**)/(**ABS**) and (**IMP**)) depending on the context, and to choose a clever aggregation strategy to overcome the absence of the (**MAX**) property. However, as we shall see, there is no "free lunch" in the choice of metric, and our aggregation strategy also introduces shortcomings that must be acknowledged when used.

Instead of computing the difference between the arithmetic means or the medians as it is usually done in the literature, we can compute the ratio between the geometric means.

If we output two sets of errors $e_i \in [0, E_i]$ and $e'_i \in [0, E_i]$ then we can rewrite e_i and e'_i as $\lambda_i E_i$ and $\lambda'_i E_i$. with λ_i and λ'_i in $[0, 1]$. If we compare the two methods using the arithmetic mean then

$$\frac{1}{n}\sum_{i=1}^{c}(e_i - e'_i) = \frac{1}{n}\sum_{i=1}^{c}(\lambda_i - \lambda'_i)E_i,$$

which is a problem because the setting for which the range is high will count more on the global error. But if we compute the geometric mean:

$$\frac{\sqrt[n]{\prod_{i=1}^{n} e_i}}{\sqrt[n]{\prod_{i=1}^{n} e'_i}} = \sqrt[n]{\prod_{i=1}^{n} \frac{\lambda_i}{\lambda'_i}},$$

then all problems and errors are equally considered. The major problem with this strategy is that if only one error is zero then the geometric mean equals zero.

Alexandari et al. [2] suggested combining the aggregation strategy (in their case a median rather than a geometric mean) with a paired Wilcoxon signed-rank test. If the p-value of the test is greater than a fixed threshold (for example, 0.01), then one of the two methods is *significantly*[4] better than the other, but it does not say which is the best.

Additional remarks

Proposition 2.1. *Discordance ratio does not satisfy* ***REL****, contrary to what Sebastiani [104] announced.*

Proof. Take, $\alpha = (0.25, 0.75)$, $\beta = (0.5, 0.5)$ and $t = 0.5$.
In that case,

$$D(\alpha, \alpha_t) = 1 - \frac{1}{2}\left(\frac{0.25}{0.75} + \frac{0.25}{0.75}\right) = 2/3,$$
$$D(\beta, \beta_t) = 1 - \frac{1}{2}\left(\frac{0.5}{1} + \frac{0}{0.5}\right) = 3/4,$$

so $D(\alpha, \alpha_t) < D(\beta, \beta_t)$, contradicting (**REL**). □

Sebastiani [104] introduced **REL** and **ABS** but a third option is possible:

Definition 2.14. Anti-Relativity **(AREL)**.
For every proportions $\alpha, \beta \in \Delta^c$, that satisfy $\alpha_1 < \beta_1 \leq \beta_2 < \alpha_2$. Let us take $t \in [-\alpha_1, \beta_2]$ such that $\alpha_t = (\alpha_1 + t, \alpha_2 - t) \in \Delta^c$ and $\beta_t = (\beta_1 + t, \beta_2 - t) \in \Delta^c$ two approximations. Then it holds:

$$D(\alpha, \alpha_t) < D(\beta, \beta_t).$$

Several methods presented before satisfied this property.

[4] We note that the notion of "statistical significance" is widely criticised by statisticians, see Amrhein et al. [3]

Proposition 2.2. ***NAE***, ***NRAE*** *and* ***NSQ*** *satisfy* ***AREL***.

Proof. A direct computation shows that :

- $NAE(\alpha, \alpha_t) = \frac{t}{\alpha_2}$,
- $NRAE(\alpha, \alpha_t) = 2t(1 - \alpha_2)$,
- $NSQ(\alpha, \alpha_t) = \frac{t}{\alpha_2^2}$.

We conclude, using $\alpha_2 > \beta_2$. □

These 3 methods also satisfy **MAX**, so there are methods that satisfy all the axioms if we replace **REL**/**ABS** with **AREL**. However, **AREL** means that we penalise more errors made on the "middle class" than on the small and large classes, and we do not see any reason why we would want to have this property in our metric.

2.4.2 Evaluation protocols

The easiest way to test a quantification method is to have multiple datasets from different sources and perform quantification on them, as we would in a real situation where we want to apply quantification. For example, in Chapters 3 and 4 we have access to a set of 29 patients, so we can use each patient alternately as a source and as a target, creating 29×28 sets on which to compare the data. However, we need to make sure that the different distributions satisfy the label shift assumption ($\mathcal{LS}$) before testing the methods, which may turn out to be wishful thinking on real data. What is more, we are limited by the number of samples available to us and, finally, we do not have access to any form of hyper-parameter that would allow us to control the difficulty of the task i.e. the "amount" of shift.

For these reasons, the community has discussed and proposed mechanisms to create *bags* (i.e. subsamples) from one or more sources, where each bag has its own proportions α to be estimated. This discussion can be linked to the splitting strategy we discuss for deep quantification learning [95, 96], which requires a set of bags to train the model. Two approaches have been proposed in the quantification literature: the *Natural Prevalence Protocol (NPP)*, introduced by Esuli et al. [38], and the *Artificial Prevalence Protocol (APP)*, going back at least to the seminal work of Saerens et al. [101]. The two approaches differ not in the nature of the data used, which can be artificial or real, but in the mechanism for generating the *bags*.

The first framework (NPP) consists of taking a very large data set and dividing it into uniformly drawn samples. This approach has the advantage of ensuring that the label shift hypothesis is verified on the bags, but the proportions of the different bags do not differ sufficiently and so we can not measure the robustness of the methods to 'hard' changes in the proportion distribution. Therefore, the method is now deprecated in the literature.

The most used protocol in the literature is the artificial one (APP). The idea of this approach is to control the amount of shift in the target distribution by either subsampling in a dataset, with respect to given proportions α, a bag that will satisfy $q_Y(\cdot) = \alpha$ or alternatively, if the data is generated, generating a mixture with the given proportions. Different samples can be generated, each characterised by different proportions, so the key element of APP is the generation mechanism of the proportions α. When it was first introduced, the protocol was presented in the two-class setting. If $c = 2$, we can simply do a grid-based exploration. For each $\alpha \in [0, 1]$, a vector of proportions can be generated by taking $(\alpha, 1 - \alpha)$. This is particularly interesting because we can then plot the errors against α as a simple line graph. The main disadvantage is that the resulting proportions are not "*natural*". For instance, if the two classes are *healthy cells* and *cancerous cells*, it would be odd to have 95% *cancerous cells* and only 5% *healthy cells*. However, this can easily be fixed by putting conditions on α.

The protocol can be generalised for $c > 2$ by creating a grid on the simplex Δ^c, but this becomes intractable as the number of classes increases. For LeQua 2022, Esuli et al. [34] mitigated this problem by replacing the extensive grid-based exploration by sampling α uniformly on the simplex with the so-called Kraemer sampling algorithm. However, we still do not have access to hyperparameters to control the difficulty of the problem.

In the label shift literature, authors have proposed another form of artificial protocol where the proportions are also sampled from a distribution on the simplex but not the uniform one. Lipton et al. [77] suggested using **tweak-one shift**. The idea is to assign a probability ρ to one class and then distribute the remaining mass to the other classes, either equally across the classes or according to the source proportions. The results are easy to plot, the proportions are more "natural", and we can play with two hyperparameters : ρ and the class taken. In a sense, this approach is similar to the artificial prevalence proposed by Saerens in the two-class setting. Another protocol, also used by Lipton but also by Alexandari et al. [2], is the **Dirichlet shift**. The idea is to generate the proportions with the Dirichlet distribution of parameter $\tau\beta$, with $\beta \in \Delta^c$ and $\tau > 0$. For β we can either take the source proportions, the vector $[1/c, \cdots, 1/c]$ or any prior knowledge on the target proportions. The parameter τ does not change the mean of the Dirichlet distribution ($\mathbb{E}[\mathcal{D}(\tau\beta)] = \beta$) but it changes the variance, higher values of τ leading to proportions that are almost sparse, while small values lead to proportions that are close to β, see Figure 2.11.

This approach has the advantage of having a hyperparameter to control the difficulty, while ensuring that the average proportions are *natural* (thanks to β).

2.4.3 Conclusion on the evaluation protocols

As we have seen, the evaluation of protocols is a more complex issue in quantification than in more classical machine learning problems.

For the choice of metric, we believe (as it was the case in [104]) that two metrics stand out: the absolute error (**AE**) and the relative absolute error (**RAE**). The absolute error satisfied the (**ABS**) property, so we want to use it in situations where the classes are balanced or in

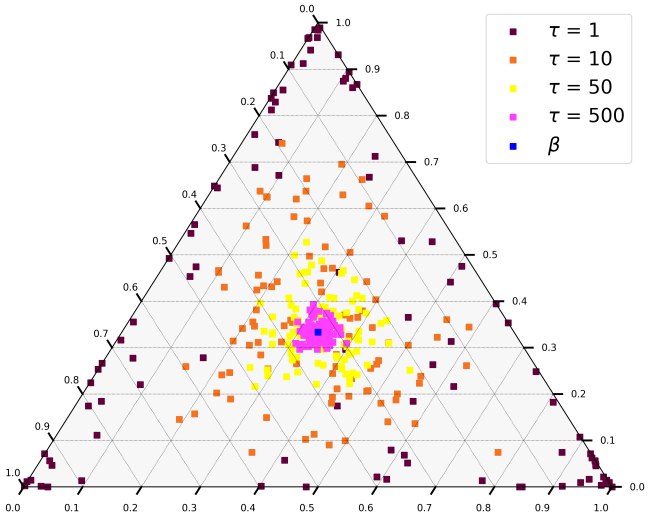


Figure 2.11: Ternary plot of Dirichlet distributions in dimension three, used to sample the target proportions for $\tau = 1, 10, 50, 500$. The proportions β are in blue. When τ is small, the proportions are almost sparse (i.e. close to the border) and when τ is large the proportions are more balanced around β.

situations where making errors on small classes does not need to be penalised more. On the other hand, the relative absolute error must be used in situations where we want to penalise errors on small classes more. These two measures do not satisfy the property (**MAX**), but this problem can be mitigated as we have seen. This choice of metric is common in the quantification literature, for example (**RAE**) is the metric used for the quantification data challenges : LeQua2022 [35] and LeQua2024 [36].

For the evaluation protocol, we propose to use the **Dirichlet shift** for the reasons we have explained. This choice is not common in the literature, the authors often prefer to use the Kraemer sampling algorithm to sample uniformly on the simplex.

Chapter 3

Quantification with Distribution Feature Matching

This chapter is a modified version of the article B. Dussap, G. Blanchard, BE. Chérief-Abdellatif : Label Shift Quantification with Robustness Guarantees via Distribution Feature Matching published in *Machine Learning and Knowledge Discovery in Databases: Research Track. ECML PKDD 2023. Lecture Notes in Computer Science, vol 14173. Springer, 2023.* [30]. The version contained in this manuscript includes new experiments, more detailed explanations of the theorems, and a new introduction that is more in line with the content of the other chapters.

In this chapter, we study our framework introduced in Section 2.3.2: *Distribution Feature Matching* (DFM). We derive a general performance bound for DFM procedures under *Label Shift*, which improves on previous bounds derived in particular cases in several key aspects. We then extend this analysis by departing from the exact label shift hypothesis, in particular in the case of *open set label shift*. These theoretical results are confirmed by numerical studies on simulated and real datasets. We also present an efficient, scalable and robust version of kernel-based DFM using Random Fourier Features.

Contents

3.1 Introduction

As we discussed in Section 2.3.1 of the previous chapter, a significant proportion of quantification methods can be recast under a common framework called *Distribution Matching*. Since we have access to a sample of each source class and a sample of the target, and since we know that under label shift the target is a mixture of the source class distributions, the strategy is to search for the source mixture that is *closest* (in some sense to be defined) to the target. A significant part of the quantification method proposed in the literature consists in defining the right notion of distance to use.

However, this unified view of quantification can not be used to derive a common analysis, because the different methods are too different. In this chapter, we introduce a new framework called *Distribution Feature Matching* (DFM), which is less general than *distribution matching*

and for which we can give a unifying theoretical analysis. This framework uses the mean vectorisation, or mean embedding, introduced in Section 1.2.3. It generalises existing methods such as Black-Box Shift Estimation (BBSE) [77], Kernel Mean Matching (KMM) [70, 133] and its variant Energy Distance Matching [73].

We also introduce Random Fourier Feature Matching (RFFM), another special case of KMM based on random Fourier features, introduced in Section 1.2.4. The idea of using RFFs has been considered in numerous works to speed up the computation of kernel methods, but to the best of our knowledge, it has never been exploited in general estimation or label shift quantification problems.

Since the aim of this work is also to apply quantification methods on flow cytometry data, and since such datasets are often "contaminated" by dead cells, doublets (i.e. two cells that are hit together when they pass in front of the laser), or simply other types of cells that are not labelled in the current experiment, we want methods that are robust to this kind of contamination.

To this end, we introduce *soft-DFM*, an extension of DFM and we study the robustness of this procedure to contamination under *open set label shift*.

Notations

From a formal point of view, consider a covariate space $\mathcal{X}$, typically a subset of $\mathbb{R}^d$, and a label space $\mathcal{Y} = [c] \coloneqq \{1, \dots, c\}$. We define the two *source* and *target* domains as different probability distributions over the covariate-label space pair $\mathcal{X} \times \mathcal{Y}$. The target label distribution is denoted $\alpha^* = (\alpha_i^*)_{i=1}^c$ while each class-i conditional target distribution is denoted $\mathbb{Q}_i$. Similarly, the source label distribution is denoted $\beta^* = (\beta_i^*)_{i=1}^c$ while each class-i conditional source distribution is denoted $\mathbb{P}_i$. We will consider the *label shift* setting :

$$\forall i = 1, \dots, c\,, \quad \mathbb{P}_i = \mathbb{Q}_i, \tag{$\mathcal{LS}$}$$

and the *open set label shift* setting which involves the contamination of the target by a new class. We assume that the target domain is $\mathcal{X} \times \tilde{\mathcal{Y}}$, with $\tilde{\mathcal{Y}} = \{0, \dots, c\}$ and that the label shift hypothesis is still verified for the class $\{1, \dots, c\}$:

$$\begin{aligned} &\mathbb{Q} = \sum_{i=1}^{c} \alpha_i^* \mathbb{Q}_i + \alpha_0^* \mathbb{Q}_0 \\ &\forall i = 1, \dots, c\,, \quad \mathbb{P}_i = \mathbb{Q}_i. \end{aligned} \tag{$\mathcal{OSLS}$}$$

The distribution $\mathbb{Q}_0$ is seen as a noise or a contamination, for which we have no prior knowledge nor sample. Therefore, our objective in this contaminated scenario is to be robust to a large class of noise distributions. In Section 3.3.3, we will give insight on the kind of contamination we can be robust to.

In both settings, we suppose a source dataset $\{(x_j, y_j)\}_{j \in [n]} \in (\mathcal{X} \times \mathcal{Y})^n$ and a target dataset $\{x_{n+j}\}_{j \in [m]} \in \mathcal{X}^m$ are given. All data points from the source (respectively the target)

dataset are independently sampled from the source (resp. the target) domain. We have access to the source labels y_j but not to the target labels which are not observed. We denote by $\hat{\mathbb{P}}_i \coloneqq \sum_{j\in[n]:y_j=i} \delta_{x_j}(\cdot)/n_i$ the empirical source class-i conditional distribution, where δ_{x_j} denotes the Dirac measure at point x_j and n_i the number of instances labeled i in the source dataset. Note that $n_1+\ldots+n_c = n$. In the same manner, we denote $\hat{\mathbb{Q}} \coloneqq \sum_{j\in[m]} \delta_{x_{n+j}}(\cdot)/m$ the empirical target distribution. We finally denote by $\tilde{\beta}$ the empirical proportions in the source dataset, i.e. $\tilde{\beta}_i \coloneqq n_i/n$.

3.2 Distribution Feature Matching

Let us present our general framework, Distribution Feature Matching or DFM.

Definition 3.1 (Distribution Feature Matching)**.** Let $\Phi : \mathcal{X} \to \mathcal{F}$ be a fixed feature mapping from $\mathcal{X}$ into a Hilbert space $\mathcal{F}$ (possibly $\mathcal{F} = \mathbb{R}^D$). Suppose we can extend the mapping Φ to probability distributions on $\mathcal{X}$ by taking expectation, i.e. $\Phi\colon \mathbb{P} \mapsto \Phi(\mathbb{P}) \coloneqq \mathbb{E}_{X\sim\mathbb{P}}[\Phi(X)] \in \mathcal{F}$.

We call *Distribution Feature Matching* (DFM) any estimation procedure that can be formulated as the minimiser of the following problem:

$$\hat{\alpha} = \operatorname*{arg\,min}_{\alpha\in\Delta^c} \left\| \sum_{i=1}^{c} \alpha_i \Phi(\hat{\mathbb{P}}_i) - \Phi(\hat{\mathbb{Q}}) \right\|_{\mathcal{F}}^2 \tag{$\mathcal{P}$}$$

where Δ^c is the $(c-1)$-dimensional simplex.

In the contamination setting, we aim at finding the proportions of the non-noise classes of the target. As these proportions do not sum to one, the "hard" condition $\sum_i \alpha_i = 1$ is no longer needed. One way to overcome this is to introduce a fictitious "dummy" class in the source that formally has a vectorisation equal to 0 (note that adding a dummy class is a computational and theoretical convenience; we do not require to have a real distribution $\mathbb{P}_0$ that maps to 0 in the feature space for the results to hold). If we write $\Phi(\hat{\mathbb{P}}_0) \coloneqq 0$ one can see that:

$$\operatorname*{arg\,min}_{\alpha\in\Delta^{c+1}} \left\| \sum_{i=0}^{c} \alpha_i \Phi(\hat{\mathbb{P}}_i) - \Phi(\hat{\mathbb{Q}}) \right\|_{\mathcal{F}}^2 = \operatorname*{arg\,min}_{\alpha\in\mathsf{int}(\Delta^c)} \left\| \sum_{i=1}^{c} \alpha_i \Phi(\hat{\mathbb{P}}_i) - \Phi(\hat{\mathbb{Q}}) \right\|_{\mathcal{F}}^2, \tag{$\mathcal{P}_2$}$$

where $\mathrm{int}(\Delta^c) \coloneqq \{x \in \mathbb{R}^c\colon x \geq 0,\ \sum x_i \leq 1\}$. A procedure that solves $\mathcal{P}_2$ will be called *soft*-DFM.

In Section 3.3, we will present theoretical results, in the classical label shift hypothesis ($\mathcal{LS}$), for DFM methods under an identifiability and boundedness assumption.

In Section 3.3.3, we will show a general result for ("hard") DFM methods when the label shift hypothesis is not verified. As a corollary of these bounds we will directly obtain corresponding guarantees for the *soft*-DFM methods.

In the remainder of this section, we will show the link between DFM and other classical label shift quantification algorithms. However, any black-box feature mapping will be suitable for the results of Section 3.3.

3.2.1 Kernel Mean Matching for Label Shift

Iyer et al. [70] used Kernel Mean embedding (KME) as their mapping. We refer the reader to Muandet et al. [87] for a survey on KME. We briefly recall that for any symmetric and semidefinite positive kernel k defined on $\mathcal{X}$, one can associate a unique Hilbert space denoted $\mathcal{H}_k$, or simply $\mathcal{H}$ when there is no ambiguity, and a feature map $\Phi : \mathcal{X} \to \mathcal{H}$ such that $\langle \Phi(x), \Phi(y) \rangle = k(x, y)$.

This mapping can be extended to the space of distributions by taking expectations, which constitutes the principle of KME. Note that it is sufficient to have $\mathbb{E}_{\mathbb{P}}[\sqrt{k(X, X)}] \leq \infty$ to ensure the existence of $\Phi(\mathbb{P})$, see Smola et al. [109]. An important property of this mapping is that, even if we do not have direct access to the mapping as it is an element of the infinite-dimensional Hilbert space $\mathcal{H}$, we still can compute scalar products between mappings using the so-called *kernel trick*:

$$\langle \Phi(\mathbb{P}), \Phi(\mathbb{Q}) \rangle_{\mathcal{H}} = \mathbb{E}_{(X,Y)\sim\mathbb{P}\otimes\mathbb{Q}}[k(X, Y)],$$

which provides a way to find an explicit solution of Equations $(\mathcal{P})$ and $(\mathcal{P}_2)$ in practice.

For any kernel k, the function $D_\Phi(\mathbb{P}, \mathbb{Q}) = \|\Phi(\mathbb{P}) - \Phi(\mathbb{Q})\|_{\mathcal{H}}$ is a pseudo-distance on the space of measures on $\mathcal{X}$, called the *Maximum Mean Discrepancy* (MMD) [61]. D_Φ will be a distance if and only if the mapping Φ is injective. Kernel that satisfy this assumption, for instance the Gaussian kernel $k(x, y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma}\right)$, and their corresponding RKHS, are said to be *characteristic*.

In the paper that introduced MMD-based technique for quantification under label shift, Iyer et al. [70] did not properly named their method. However, in the closely-related setting of classification under covariate-shift, Gretton et al. [63] also introduced a MMD-based technique similar to the one we presented here and called it *Kernel Mean Matching* (KMM). This is the name we will use in the latter.

3.2.2 Energy Distance

Kawakubo et al. [73] proposed to used the *energy distance* to do quantification. The *energy distance* is defined as a weighted L2 distance between the characteristic functions of the two distributions but can be equivalently expressed as:

$$\textbf{EnergyDistance}(\mathbb{P}, \mathbb{Q}) = 2\mathbb{E}_{\mathbb{P},\mathbb{Q}}\big[\|X - Y\|\big] - \mathbb{E}_{\mathbb{P},\mathbb{P}}\big[\big\|X - X'\big\|\big] - \mathbb{E}_{\mathbb{Q},\mathbb{Q}}\big[\big\|Y - Y'\big\|\big].$$

Proposition 3.1. *The* **EnergyDistance** *is a particual case of MMD, with the so-called Energy kernel:*

$$k(x, y) = \|x\| + \|y\| - \|x - y\|.$$

Proof. One important property of MMD is that it can be represented using the associated kernel k as

$$\mathbf{MMD}^2(\mathbb{P}, \mathbb{Q}) = \mathbb{E}_{\mathbb{P},\mathbb{P}}\big[k(X, X')\big] + \mathbb{E}_{\mathbb{Q},\mathbb{Q}}\big[k(Y, Y')\big] - 2\mathbb{E}_{\mathbb{P},\mathbb{Q}}[k(X, Y)]. \tag{3.1}$$

Now if we plug the definition of the Energy kernel in 3.1, we obtain:

$$\begin{aligned}
\mathbb{E}_{\mathbb{P},\mathbb{P}}\big[k(X, X')\big] &= 2\mathbb{E}_{\mathbb{P}}\big[\|X\|\big] - \mathbb{E}_{\mathbb{P},\mathbb{P}}\big[\big\|X - X'\big\|\big] \\
\mathbb{E}_{\mathbb{Q},\mathbb{Q}}\big[k(Y, Y')\big] &= 2\mathbb{E}_{\mathbb{Q}}\big[\|Y\|\big] - \mathbb{E}_{\mathbb{Q},\mathbb{Q}}\big[\big\|Y - Y'\big\|\big] \\
\mathbb{E}_{\mathbb{P},\mathbb{Q}}[k(X, Y)] &= \mathbb{E}_{\mathbb{P}}\big[\|X\|\big] + \mathbb{E}_{\mathbb{Q}}\big[\|Y\|\big] - \mathbb{E}_{\mathbb{P},\mathbb{Q}}\big[\|X - Y\|\big]
\end{aligned}$$

$$\begin{aligned}
\mathbf{MMD}^2(\mathbb{P}, \mathbb{Q}) &= 2\mathbb{E}_{\mathbb{P},\mathbb{Q}}\big[\|X - Y\|\big] - \mathbb{E}_{\mathbb{P},\mathbb{P}}\big[\big\|X - X'\big\|\big] - \mathbb{E}_{\mathbb{Q},\mathbb{Q}}\big[\big\|Y - Y'\big\|\big]. \\
&= \mathbf{EnergyDistance}(\mathbb{P}, \mathbb{Q})
\end{aligned}$$

The fact that $k(x, y) = \|x\| + \|y\| - \|x - y\|$ is indeed a reproducing kernel can be found in [105]. □

As a result, Kawakubo's method is, in fact, a particular example of kernel mean matching and therefore of distribution feature matching. This method has the convenient advantage of not depending on hyperparameters to be optimized, as is the case with the bandwidth σ in the Gaussian Kernel.

3.2.3 BBSE as Distribution feature matching

Black-Box Shift Estimation (BBSE) or Adjusted Classify and Count (ACC), is a method using the output of a black-box classifier f to estimate the proportions in the target. To take into account the bias of the training data (i.e. the source) Lipton et al. [77] used the confusion matrix.

To understand how Black-Box Shift Estimation can be cast as a Distribution Feature Matching procedure, we start from its original formulation, presented in Equation (2.5), as seeking the vector of proportions α that satisfies $\hat{\alpha}_{\text{cc}} = C_{\hat{y}|y}\alpha$, where $C_{\hat{y}|y}$ is the estimated conditional confusion matrix defined as $(C_{\hat{y}|y})_{ij} = \frac{1}{n_i}\sum \mathbf{1}\{f(x_l) = i \text{ and } y_l = j\}$ and $\hat{\alpha}_{\text{cc}}$ is the classify and count estimator, i.e. the empirical mean of the observed outputs of the black-box classifier f on the target data, $(\hat{\alpha}_{\text{cc}})_i = \frac{1}{n}\sum_l \mathbf{1}\{f(x_l) = i\}$. The BBSE estimate is then $\hat{\alpha} = {C_{\hat{y}|y}}^{-1}\hat{\alpha}_{\text{cc}}$ ($C_{\hat{y}|y}$ is explicitly assumed invertible by Lipton et al. [77]).

Proposition 3.2. *The BBSE estimator based on the black-box classifier f is the same as the solution of the DFM problem $(\mathcal{P})$ using the feature mapping $\Phi(x) = (\mathbf{1}\{f(x) = i\})_i \in \mathbb{R}^c$, where the positivity constraint on α is dropped.*

Proof. It is straightforward to check that for the mentioned feature mapping, $\Phi(\hat{\mathbb{P}}_i)$ in the DFM setting is exactly the i-th column of $C_{\hat{y}|y}$ in the BBSE notation, and $\Phi(\hat{\mathbb{Q}}) = \hat{\alpha}_{\text{cc}}$. Hence the DFM objective in $(\mathcal{P})$ rewrites to $\|\alpha^T C_{\hat{y}|y} - \hat{\alpha}_{\text{cc}}\|^2$, and since $C_{\hat{y}|y}$ is assumed invertible, in that setting the unconstrained solution is $\hat{\alpha} = C_{\hat{y}|y}^{-1}\hat{\alpha}_{\text{cc}}$. Furthermore, the sum-1 condition $\mathbf{1}^T\alpha = 1$ (where $\mathbf{1}$ denotes a vector of ones of dimension c) is automatically satisfied for the unconstrained solution: obviously it holds $\mathbf{1}^T C_{\hat{y}|y} = \mathbf{1}^T$, hence $\mathbf{1}^T = \mathbf{1}^T C_{\hat{y}|y}^{-1}$, and $\mathbf{1}^T\hat{\alpha}_{\text{cc}} = 1$, so that $\mathbf{1}^T C_{\hat{y}|y}^{-1}\hat{\alpha}_{\text{cc}} = 1$. □

In the experiments to come, we will use BBSE+, our modified version of BBSE including the positivity constraint. The experimental results are slightly better for BBSE+. The reason is that in many cases, due to the presence of small classes in the source and in the target, BBSE returns negative proportions. When it does not output negative values, the two algorithms are the same.

This version of BBSE (with the positive constraint) already existed in the literature, it has been used for text content analysis by Hopkins et al. [69] and as a building block for classification, in a domain adaptation setting more general than label shift by Tachet et al. [114].

3.3 Theoretical guarantees

We now provide statistical guarantees for DFM and *soft*-DFM procedures. In particular, we provide a high-probability upper bound on the Euclidean distance $\|\hat{\alpha} - \alpha^*\|_2$, between the estimated proportions $\hat{\alpha}$ and the true proportions α^* under both label shift and open set label shift. Moreover, the bounds are true if we replace the true proportions α^* of the target by the empirical proportions $\tilde{\alpha}$ of the target sample. More precisely, if we note m_i the number of points of class i in the target sample, then $\tilde{\alpha} = m_i/m$.

We make the following identifiability hypothesis on the mapping Φ:

$$\sum_{i=1}^{c} \beta_i \Phi(\mathbb{P}_i) = 0 \iff \beta_i = 0 \ \forall i = 1, \dots, c, \tag{$\mathcal{A}_1$}$$

and

$$\exists C > 0 : \qquad \|\Phi(x)\|_{\mathcal{F}} \leq C \text{ for all } x. \tag{$\mathcal{A}_2$}$$

If we use KMM, the boundedness property is satisfied as soon as the kernel is bounded (for instance the Gaussian kernel, or any continuous kernel on a compact space). For BBSE, the boundedness is verified with $C = 1$.

Concerning Condition $\mathcal{A}_1$, it is satisfied in the KMM case as long as the kernel is characteristic and the distributions $\mathbb{P}_i$ are linearly independent (which is the minimal assumption for the class proportions to be identifiable). These assumption have been previously used by Iyer et al. [70] for KMM. Similarly, for BBSE, Lipton et al. [77] also assumed identifiability and required that the expected classifier outputs for each class to be linearly independent.

We introduce the following notations and state our main theorem:

Definition 3.2. We denote $\hat{\boldsymbol{G}}$ the Gram matrix, resp. $\hat{\boldsymbol{M}}$ the centered Gram matrix of $\{\Phi(\hat{\mathbb{P}}_1), \cdots, \Phi(\hat{\mathbb{P}}_c)\}$. That is, $\hat{\boldsymbol{G}}_{ij} = \langle \Phi(\hat{\mathbb{P}}_i), \Phi(\hat{\mathbb{P}}_j) \rangle$ and $\hat{\boldsymbol{M}}_{ij} = \langle \Phi(\hat{\mathbb{P}}_i) - \overline{\Phi}, \Phi(\hat{\mathbb{P}}_j) - \overline{\Phi} \rangle$ with $\overline{\Phi} = c^{-1} \sum_{k=1}^{c} \Phi(\hat{\mathbb{P}}_k)$. Furthermore, let $\Delta_{\min}$ be the second smallest eivenvalue of $\hat{\boldsymbol{M}}$ and $\lambda_{\min}$ the smallest eigenvalue of $\hat{\boldsymbol{G}}$. In particular, it holds $\Delta_{\min} \geq \lambda_{\min}$.

Theorem 3.1. *If the label shift hypothesis $(\mathcal{LS})$ holds, and if the mapping Φ satisfies Assumptions $(\mathcal{A}_1)$ and $(\mathcal{A}_2)$, then for any $\delta \in (0,1)$, with probability greater than $1-\delta$, the solution $\hat{\alpha}$ of $(\mathcal{P})$ satisfies:*

$$\|\hat{\alpha} - \alpha^*\|_2 \leq \frac{2CR_{c/\delta}}{\sqrt{\Delta_{min}}} \left(\sqrt{\frac{\|w\|_1}{n}} + \frac{1}{\sqrt{m}} \right) \tag{3.2}$$

$$\leq \frac{2CR_{c/\delta}}{\sqrt{\Delta_{min}}} \left(\frac{1}{\sqrt{\min_i n_i}} + \frac{1}{\sqrt{m}} \right), \tag{3.3}$$

where $R_x = 1 + \sqrt{2\log(2x)}$, $w = \alpha^*/\tilde{\beta}$.

The same result holds when replacing α^* by the (unobserved) vector of empirical proportions $\tilde{\alpha}$ in the target sample, both on the left-hand side and in the definition of w.

Moreover, under the same assumptions, the solution $\hat{\alpha}_{\text{soft}}$ of $(\mathcal{P}_2)$ satisfies the same bounds with Δ_{min} replaced by λ_{min}.

The proof can be found in Section 3.5.1.

The first inequality of the theorem, Equation (3.2), depends on the true proportions we want to estimate α^* in w. $\|w\|_1$ can be understood as an "*amount of shift*". $\|w\|_1 = c$ in the best case scenario, i.e. when the empirical proportions are equal to the proportions we want to estimate (i.e. no shift) or when we have the same number of points in each class of the labelled data, indeed when $\tilde{\beta}_i = 1/c$, we have $\|w\|_1 = \sum_{i=1}^{c} \alpha_i^*/\tilde{\beta}_i = c \times \sum_{i=1}^{c} \alpha_i^* = c$ because $\alpha^* \in \Delta^c$.

To remove the dependence on the true proportions, we take the worst case scenario, which is obtained when $\alpha_i = 1$ for the smallest $\tilde{\beta}_i$ and 0 for the others, i.e. when the class for which we have the least information becomes the only class in the target. In this case we obtain Equation (3.3).

The dependence in α^* in the first equation is actually desirable. Indeed, in situations where one of the classes i on the source domain is rare (for example, the cancer cells in a flow cytometry sample), then the rate $(\min_i n_i)^{-1/2}$ in Inequality (3.3) explodes, which is

not the case of the rate $\sqrt{\|w\|_1/n}$ in Inequality (3.2) when the source and target proportions are similar, as the weight vector w reflects the similarity between the source and target distributions.

Remark. When Φ is a black-box classifier, the embeddings of the distribution are the column of the confusion matrix, i.e. vectors in Δ^c. According to Theorem 3.2 that we present in Section 3.3.2, the confusion matrix that maximise Δ_{min} is the identity.
We presented in Chapter 2 a line of research that trained a classifier to be a good quantifier without being a good classifier. According to our study, for BBSE (or ACC as we also called it in the first chapter), the best classifier to use for quantification is a good classifier. This result was also pointed out by Tasche [117] for the Maximum Likelihood Estimator, where he shows that the asymptotic variance of the quantifier is proportional to the inverse of the mean squared error on the target. In other words, a better classifier reduces the asymptotic variance of the quantifier.

3.3.1 Comparison to related literature

We compare our result to Theorem 1 of [70] (*Kernel Mean Matching*) and Theorem 3 of [77] (*BBSE*), which as we have mentioned earlier hold under the same assumptions as we make here.

Concerning KMM, a comparison between our inequality (3.3) and Theorem 1 in [70] shows that our bound is tighter than theirs, which is of leading order

$$\frac{c}{\sqrt{m}} + \sum_i \frac{1}{\sqrt{n_i}} \quad \text{vs ours in} \quad \frac{1}{\sqrt{m}} + \max_i \frac{1}{\sqrt{n_i}}$$

up to logarithmic factors. Thus, Theorem 3.1 improves upon the previous upper bound by a factor of c with respect to the term in m, and reduces the sum into a maximum regarding the number of instances per class n_i, which may also decrease the order of by factor c when the classes are evenly distributed in the source dataset. Furthermore, Inequality (3.2) even significantly improves over both Inequality (3.3) and Theorem 1 in [70]. Hence, our theorem significantly improves the existing bound for KMM established by [70].

Similarly, our bound (3.2) applied to BBSE also improves Theorem 3 in [77]. In particular, when both inequalities are formulated with the same probability level (for instance $1-\delta$), our bound for BBSE is tighter by a factor $\sqrt{c}$ w.r.t. the term in m than the guarantee provided by [77]. Note however that contrary to our result and to Theorem 1 in [70], the bound of [77] does not involve any empirical quantity that can be computed using the source dataset.

Another key component of the bounds is the second smallest eigenvalue Δ_{min} of the centered Gram matrix, which replaces the minimum singular value λ_{min} of the Gram matrix in the case of KMM (see Theorem 1 in [70]) and the smallest eigenvalue of the confusion matrix divided by the infinite norm of the source proportions in the case of BBSE (see Inequality (3) of Theorem 3 in [77]), and improves upon both of them.

This improvement is particularly important when the two source classes are unbalanced. For instance, in a two-class setting with a black-box classifier feature map and $\beta^* = (p, 1-p)$, the theoretical version of Δ_{min} is equal to 1 when the classifier is perfect and replaces the factor that would be $\min\left(\frac{p}{1-p}, \frac{1-p}{p}\right) < 1$ in the bound of [77]. When the classifier is not perfect but both classes share the same classification accuracy $a \in (1/2, 1)$, then $\Delta_{\mathsf{min}} = 2a - 1$, which strictly improves the factor of [77] except when both classes are equally balanced, in which case both quantities are equal.

3.3.2 Criterion for the choice of feature mapping

A classical problem of kernel method that used the Gaussian kernel is the choice of bandwidth σ. In our case the problem is more general, because we have to choose a feature map Φ. The quantity Δ_{min} (which is empirical) serves as a natural **criterion** for selecting the hyperparameters of the feature map, because the dependence in our bound only appears in Δ_{min}, which can then be maximized using the labelled training dataset.

To fully understand the nature of Δ_{min}, we can interpret DFM as the projection of the target embedding onto the convex hull of the source embeddings. Our estimation is then the barycentric coordinate of the projection, see Figure 3.1. In contrast to λ_{min}, Δ_{min} can be understood as a geometric property of this convex hull. Indeed, if we shift all the distribution embeddings by the same vector $b \in \mathcal{F}$, we obtain a new Gram matrix $\hat{\boldsymbol{G}}_b$ and we can show that $\Delta_{\mathsf{min}}(\hat{\boldsymbol{G}}) = \Delta_{\mathsf{min}}(\hat{\boldsymbol{G}}_b)$, which is obviously not the case for $\lambda_{\mathsf{min}}(\hat{\boldsymbol{G}})$ and $\lambda_{\mathsf{min}}(\hat{\boldsymbol{G}}_b)$.

To better understand this, let us state the following proposition:

Proposition 3.3. *For any number of classes c, Δ_{min} is equal to* $\min\limits_{\substack{\|u\|_2=1 \\ \mathbf{1}^T u=0}} u^T \hat{\boldsymbol{G}} u$.

The proof can be found in Section 3.5.4.

Corollary 3.1. *For two classes, $\Delta_{min} = \frac{1}{2}\left\|\Phi(\hat{\mathbb{P}}_1) - \Phi(\hat{\mathbb{P}}_2)\right\|^2$.*

Proof. In two dimensions, the conditions $\|x\| = 1$ and $\mathbf{1}^T x = 0$ can only be verified for 2 points, $x = \left(\sqrt{\frac{1}{2}}, -\sqrt{\frac{1}{2}}\right)$ and $x = \left(-\sqrt{\frac{1}{2}}, \sqrt{\frac{1}{2}}\right)$. If we compute $x^T \hat{\boldsymbol{G}} x$ for these two points we obtain : $\frac{1}{2}\left\|\Phi(\hat{\mathbb{P}}_1)\right\|^2 - \left\langle \Phi(\hat{\mathbb{P}}_1), \Phi(\hat{\mathbb{P}}_2) \right\rangle + \frac{1}{2}\left\|\Phi(\hat{\mathbb{P}}_2)\right\|^2$ which is equal to $\frac{1}{2}\left\|\Phi(\hat{\mathbb{P}}_1) - \Phi(\hat{\mathbb{P}}_2)\right\|^2$. □

From a geometric point of view, it is clear that the larger the convex hull (i.e. the line connecting the two features in situations where there are two classes), the less the barycentric coordinate will be affected by a small perturbation of the weights. From a statistical point of view, if our mixture is composed of two very different distributions, it will be intuitively easier to distinguish them in a new sample.

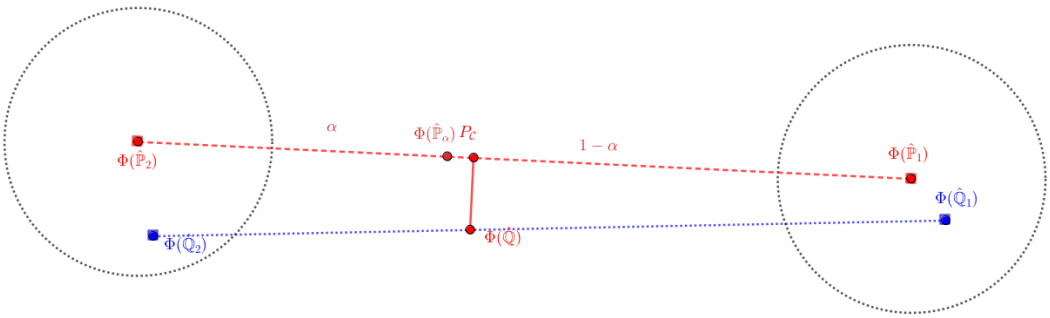


Figure 3.1: One way to understand the DFM procedure is to view it as the projection of the target embedding onto the convex hull of the source embeddings. In this example, we have two classes, with $(\alpha, 1-\alpha) = (0.4, 0.6)$. The points $\Phi(\hat{\mathbb{P}}_1)$ and $\Phi(\hat{\mathbb{P}}_2)$ in red are known; these are the embeddings of the source. The points $\Phi(\hat{\mathbb{Q}}_1)$ and $\Phi(\hat{\mathbb{Q}}_1)$ in blue are unknown, as we do not have access to the labels in the target distribution, but we do know $\Phi(\hat{\mathbb{Q}})$ in red. Equation $\mathcal{P}$ simplifies to projecting the point $\Phi(\hat{\mathbb{Q}})$ onto the convex hull of the sources (here, the dotted line $\mathcal{P}_C$ in red). The estimated proportions can be read from the distances between $\mathcal{P}_C$ and the source embeddings. $\Phi(\hat{\mathbb{P}}_\alpha)$ in red represents the embedding of the source distribution reweighted by the true target proportions α, i.e. the point we would have liked to have been projected onto in order to correctly estimate the true proportions.
Due to the label shift hypothesis $\mathcal{LS}$, we know that $\Phi(\hat{\mathbb{P}}_i)$ and $\Phi(\hat{\mathbb{Q}}_i)$ are close, and we can bound their distance using Theorem A.1. The black circles represent these distances.

Understanding Δ_{min} is less intuitive when the number of classes is greater than 2, because we do not have access to a closed-form formula as simple as the distance between the two embeddings. We propose here an approximation of Δ_{min} by a more easily understandable quantity.

Definition 3.3. For any number of classes c and any vectors $\{b_1, \cdots, b_c\}$, we define:

$$\Gamma(b_1, \cdots, b_c) := \min_{(I_1, I_2) \in \mathcal{P}_2(c)} d^2(C_{I_1}, C_{I_2}),$$

with the following:

$$\mathcal{P}_2(c) = \Big\{ I_1, I_2 \subset \{1, \cdots, c\} |\ |I_1 \cap I_2| = 0, |I_1 \cup I_2| = c, |I_1| \text{ and } |I_2| > 0 \Big\}$$

$$C_I = \left\{ \sum_{j \in I} \lambda_j b_j \mid \lambda \in \Delta^{|I|} \right\}$$

$$d^2(A, B) = \inf_{\substack{x \in A \\ y \in B}} \|x - y\|_2^2$$

In other words, $\Gamma \coloneqq \Gamma\Big(\Phi(\hat{\mathbb{P}}_1) \cdots, \Phi(\hat{\mathbb{P}}_c)\Big)$ is the minimal distance between the convex hull of a subset of the source distribution and the convex hull of the complementary subset. It represents how close we are from writing a barycenter as the mixture of two disjoint subset of the sources. If Γ equals 0, then it means that the embeddings of the distributions (alternatively, the distributions themselves if the mapping is injective) are not linearly independent, thus breaking assumption $\mathcal{A}_1$. On the other hand, if Γ is large, then it means not only that all classes are far from each other, but every combination of classes is far from each other. From both a geometrical and statistical point of view, we understand that a large Γ is suitable. In the case of two classes, we have $\Gamma = 2\Delta_{\mathsf{min}}$, but we do not have this equality for a higher number of classes. What we can prove, however, is that Δ_{min} is lower-bounded and upper-bounded by Γ up to constants.

Theorem 3.2. *For any number of classes c and any vectors $\{b_1, \cdots, b_c\}$:*

$$K_{\max(c)}\ \Gamma(b_1, \cdots, b_c) \geq \Delta_{min}(b_1, \cdots, b_c) \geq \frac{1}{2}\ \Gamma(b_1, \cdots, b_c), \tag{3.4}$$

where $K_{\max(c)} = \frac{c}{4}$ if c is even and $\frac{(c+1)(c-1)}{4c}$ if c is odd.

Remark. If $c = 2$, $K_{\max(2)} = 1/2$ and $\Delta_{\min} = 1/2\Gamma$ as already proved in Corollary 3.1.

The proof can be found in Section 3.5.5.

The upper bound of the theorem is tight, indeed if we take b_i as the canonical base of $\mathbb{R}^c$, then $\Delta_{\mathsf{min}}(b_1, \cdots, b_c) = 1$ and $\Gamma(b_1, \cdots, b_c) = K^{-1}_{\max(c)}$.

On the other hand, we did not found evidence yet that the lower bound was tight for more than two classes.

3.3.3 Robustness to contamination

We now present a theoretical analysis of the robustness of the method with respect to the open set label shift assumption. First, let us obtain a general result when label shift is not verified. A naive approach would simply include the bias term $\|\Phi(\mathbb{P}_i) - \Phi(\mathbb{Q}_i)\|_{\mathcal{F}}$ in the bound but we propose here something more tight that put into light the robustness of *soft*-DFM to certain types of contamination.

Theorem 3.3. *Denote by $\hat{\alpha}$ the minimiser of the DFM problem $\mathcal{P}$. Denote $\bar{V}$ the affine span of the vectors $\Phi(\mathbb{P}_i)$ and $\mathcal{C}$ the convex hull of those same vectors. Denote $\Pi_{\bar{V}}$ and $\Pi_{\mathcal{C}}$ the orthogonal resp. convex projection onto $\bar{V}$ and $\mathcal{C}$.*
Suppose the same assumptions as in Theorem 3.1 hold, except for the exact label shift assumption $\mathcal{LS}$. Then, with probability greater than $1-\delta$:

$$\|\hat{\alpha} - \alpha^*\|_2 \leq \frac{1}{\sqrt{\Delta_{min}}}\Big(3\epsilon_n + \varepsilon_m + \sqrt{2\epsilon_n}B^{\perp} + B^{\|}\Big), \tag{3.5}$$

with:

$$\epsilon_n = C\frac{R_{c/\delta}}{\sqrt{\min_i n_i}}; \qquad \varepsilon_m = C\frac{R_{1/\delta}}{\sqrt{m}}; \tag{3.6}$$

$$B^{\perp} = B^{\perp}(\mathbb{P}, \mathbb{Q}) = \sqrt{\|\Phi(\mathbb{Q}) - \Pi_{\mathcal{C}}(\Phi(\mathbb{Q}))\|_{\mathcal{F}}};$$
$$B^{\|} = B^{\|}(\mathbb{P}, \mathbb{Q}) = \max_i \|\Phi(\mathbb{P}_i) - \Pi_{\bar{V}}(\Phi(\mathbb{Q}_i))\|_{\mathcal{F}}.$$

The proof can be found in Section 3.5.2.

Observe that the bound (3.5) shows robustness of a DFM procedure against perturbations $B^{\perp}$ that are "orthogonal" to the source space $\bar{V}$ in feature space. In particular, *consistency* (i.e. convergence of the bound to 0 as the sample sizes grow to infinity) is still granted if $\mathbb{Q}_i \neq \mathbb{P}_i$ but $\Pi_V(\mathbb{Q}_i) = \mathbb{P}_i$. The procedure is robust with respect to perturbations in the sources that are orthogonal to $\bar{V}$ in feature space. Which type of perturbation of the distributions will result in (close to) orthogonal shifts in feature space very much depends on the feature mapping used. For BBSE, the feature space is of the same dimension as the number of sources, thus under condition $(\mathcal{A}_1)$, $\bar{V}$ will coincide with E_1, the affine space of vectors summing to one. Since any distribution will be also mapped to E_1, the orthogonal component will always be 0. Thus, we expect no particular robustness property for BBSE methods. For more general feature maps, such as kernel methods or any other vectorisations, this orthogonality property remains to be investigated in general, but we will exhibit below a favourable scenario for KMM in the *Open Set Label Shift* setting $\mathcal{OSLS}$.

We now state a corollary in the $\mathcal{OSLS}$ scenario. To do so, we recall that, in this case, we use the *soft*-DFM procedure $\mathcal{P}_2$. We are now in a particular case, where the only difference between source and target is that the unknown noise class $\mathbb{Q}_0$ is formally replaced by the dummy class having feature map equal to 0 in the source. Introduce $V \coloneqq \mathrm{Span}\{\Phi(\mathbb{P}_i), i \in [c]\}$ (i.e. vector span rather than affine span for $\bar{V}$ previously) and let Π_V be the orthogonal projection on V.

Corollary 3.2. *Denote by $\hat{\alpha}_{\mathrm{soft}}$ the minimiser of the soft-DFM problem $\mathcal{P}_2$. Assume the open set label shift hypothesis $(\mathcal{OSLS})$ holds, and the mapping Φ satisfies Assumptions $(\mathcal{A}_1)$ and $(\mathcal{A}_2)$. Then, with probability greater than $1-\delta$:*

$$\|\hat{\alpha}_{\mathrm{soft}} - \alpha^*\|_2 \leq \frac{1}{\sqrt{\lambda_{min}}}\Big(3\epsilon_n + \varepsilon_m + \sqrt{2\alpha_0\ \epsilon_n\ \|\Phi(\mathbb{Q}_0)\|_{\mathcal{F}}} + \|\Pi_V(\Phi(\mathbb{Q}_0))\|_{\mathcal{F}}\Big),$$

with $\epsilon_n, \varepsilon_m$ defined as in (3.6).

The proof can be found in Section 3.5.3.

In the particular case of KMM with a translation-invariant kernel $k(x, y) = \varphi(x - y)$, for any distributions $\mathbb{P}, \mathbb{P}'$ it holds $\langle \Phi(\mathbb{P}), \Phi(\mathbb{P}') \rangle = \mathbb{E}_{(X,Y)\sim\mathbb{P}\otimes\mathbb{P}'}[\varphi(X - Y)]$. Thus, if φ is rapidly decaying with distance (this is the case of the Gaussian kernel), the feature mappings $\Phi(\mathbb{P})$ and $\Phi(\mathbb{P}')$ will be close to orthogonal (have a scalar product close to 0) whenever the distributions $\mathbb{P}, \mathbb{P}'$ are well-separated. From this analysis, we anticipate that KMM with a Gaussian kernel will be robust against contaminations distributions $\mathbb{Q}_0$ whose main mass is far away from the source distributions, since its representation $\Phi(\mathbb{Q}_0)$ will then be close to orthogonal to V in feature space.

3.4 Algorithms and applications

In this section, we will apply four methods on both synthetic and real datasets.

We choose to test three soft-DFM methods: KMM using the Energy Kernel [73], our modified version of BBSE [77] and KMM using the Gaussian kernel [70]. We compare those methods to Classify and Count (CC), presented in Section 2.2.1. KMM was enhanced with Random Fourier Features to reduce the algorithmic complexity and therefore the computing time.

The main objective of the experiments is, in view of our theoretical results of Section 3.3.3 and particularly Corollary 3.2, to test robustness properties of several DFM methods against contamination of the the target dataset by different types of noise. Moreover, we want to check if the *soft* version presented in Section 3.2 leads to improved results in some cases, and will not hurt the results in the others.

All the computations were done on a computer equipped with a NVIDIA RTX A2000 Laptop.

3.4.1 Optimisation problem

Whatever the chosen mapping, solving $(\mathcal{P})$ amounts to solving a Quadratic Programming (QP) in dimension c. Indeed, we can rewrite the problem as:

$$\begin{aligned} &\text{minimise } \frac{1}{2}\alpha^T \hat{\boldsymbol{G}}\alpha + q^T\alpha \\ &\text{subject to } \alpha \succeq 0_c \text{ and } \mathbf{1}_c^T\alpha = 1, \end{aligned} \tag{QP}$$

with $q = \left(\langle \Phi(\hat{\mathbb{P}}_i), \Phi(\hat{\mathbb{Q}}) \rangle\right)_{i=1}^{c}$. This is a c-dimensional QP problem, which can be solved efficiently using any available QP solver.

The computational bottleneck is the computation of the Gram matrix $\hat{\boldsymbol{G}}$ and the vector q. Using KMM directly leads to a complexity of $O(n(n + m))$, as computing q requires evaluating the kernel for every pair of points from the source and the target and computing

$\hat{\boldsymbol{G}}$ requires evaluating the kernel for every pair of points between the source classes. Moreover, one needs to have permanent access to the source distributions, as computing q requires both the source and target raw dataset.

Due to such issues, kernel matrix approximations are often used in order to reduce the computational cost of kernel methods [16, 100]. In our case we use the well-established principle of Random Fourier Features (RFF) approximation [98] that we presented in the Section 1.2.4.

In the experiments, we will call Random Fourier Features Matching (RFFM) the DFM method that uses the RFF embedding as a feature mapping. RFFM can be used with any translation invariant kernel as long as we are able to sample from its *spectral distribution* Λ_k. We choose to stick to the classical Gaussian kernel: $k(x, y) = \exp\left(-\|x - y\|^2/(2\sigma^2)\right)$ where the parameter σ is optimised using the criterion derived from (3.2) and Λ_k is the centred Gaussian distribution with variance σ.

With this implementation, we can solve $(\mathcal{P})$ for high-dimensional data with a very large number of points in less than a second. For example, for two datasets containing 10^6 points in dimension 5, $(\mathcal{P})$ is solved in less than a second on a GPU, whereas it takes nearly 2 minutes when we use exact KMM.

Note that here RFFM is only used as a way to speed up the computation, and hence we would obtain similar results with a classical KMM using the Gaussian kernel.

3.4.2 Experiments

We test two settings. The first one is completely artificial, we generate data using Gaussian mixtures in $\mathbb{R}^D$. We contaminate the source distributions with a new Gaussian and we control the difficulty of the task with a set of hyperparameters, including but not limited to the percentage ε of noise in the target distribution.
The second one is a real scenario where we use cytometry data from Metafora. We have a set of 29 patients and we use what is called a TBNK panel to extract four types of cells: *T cells, B cells, NK cells and Plasmocytes* and we use the *T cells* as noise.

To evaluate the error of our estimates, we use 3 pieces of information as detailed in Section 2.4, the geometric means of the *absolute errors*, the median rank of the method and a one-tailed Wilcoxon test (with a threshold of 0.01) to determine whether two methods are "significantly" equivalent.

The *absolute errors* between two vectors is defined as the L_1 norm divided by the number of classes. To clarify we multiply this by 100. Therefore an error of 3.0 can be understood as, "*the average approximation error of the class proportions is* 3%". One important remark to keep in mind is that CC and BBSE output probability distribution on Δ^c while the target proportions of the non-noise classes live in $\mathrm{int}(\Delta^c)$, in other word these method are intrinsically biased by the percentage of noise ε. If we suppose that a classifier perfectly classify the non-noise data in the target, then the *absolute errors* as defined above will be equals to $\varepsilon/c \times 100$. This is the error we expect for a non robust method and we hope that soft-DFM will break

this hard threshold.

This methodology is inspired on the one hand by the work of Sebastiani et al. [104] for the choice of metric and on the other hand by the experiments carried out by Alexandari et al. [2] as explained in Section 2.4.

Gaussian Mixture

In this setting, the source consists of a list of c Gaussian distributions: $\mathbb{P}_1, \cdots, \mathbb{P}_c$ in $\mathbb{R}^D$. The mean of each Gaussian is located on the D-dimensional sphere of radius R. Each Gaussian has its own covariance matrix, generated using a random orthonormal basis of $\mathbb{R}^D$ and D eigenvalues sampled from a uniform distribution on $[0, d]$.

The proportions of the source are balanced, $\mathbb{P}_Y(i) = 1/c$, and we sample $n = 10000$ points. The proportions of the target are sampled from a Dirichlet distribution: $\mathrm{Dir}(\tau/c, \cdots, \tau/c)$, where $\tau > 0$, and we also sample $m = 10000$ points. The mean of the Dirichlet distribution is $\mathbb{P}_Y(y)$ and the hyperparameter τ controls the variance of the distribution. A high τ will lead to target proportions similar to those of the source, while small τ will lead to sparse proportions (see Figure 2.11). Finally, the target is contaminated with a new Gaussian distribution. This contamination, denoted as $\mathbb{Q}_0$, is generated with the same procedure as for the source. The same dimension D and range of eigenvalues d are used, but the radius of the ball is set as $\rho \times R$, where $\rho > 0$ is a new parameter that controls the distance between the source distributions and the noise. The last parameter is the percentage of noise ϵ.

Therefore, we have a set of 7 hyperparameters that we can adjust:

1. The dimension of the data : D.
2. The number of classes : c.
3. The radius of the sphere : R.
4. The range of eigenvalues for the covariance matrices : d.
5. The parameter of the Dirichlet distribution used for the target proportions : τ.
6. The distance between the noise and the source : ρ.
7. The percentage of noise in the target : ε.

Throughout the experiments, the radius is fixed at $R = 2$, the range of eigenvalues at $d = 1$, the variance of the Dirichlet distribution at $\tau = 50$ and the number of classes taken at $c = 5$. The dimensions taken are $D = 2$, $D = 5$ and $D = 10$. This results in three different settings for which we can see the effect of the ε and ρ parameters.

We tested two settings, one with $\rho = 1$ and one with $\rho = 10$. In the first setting, the noise is not different from the source distributions, while in the other case the noise is far away from the source (see Figure 3.2). For CC and BBSE we use a logistic regression. For

BBSE, we split the source distribution in two, one for training and the other to estimate the confusion matrix of the classifier. For RFFM, we used 1000 random Fourier features and we used the theoretical criterion for the choice of bandwidth σ, as explained in Section 3.3.2. The results can be found in table 3.1 and table 3.2.

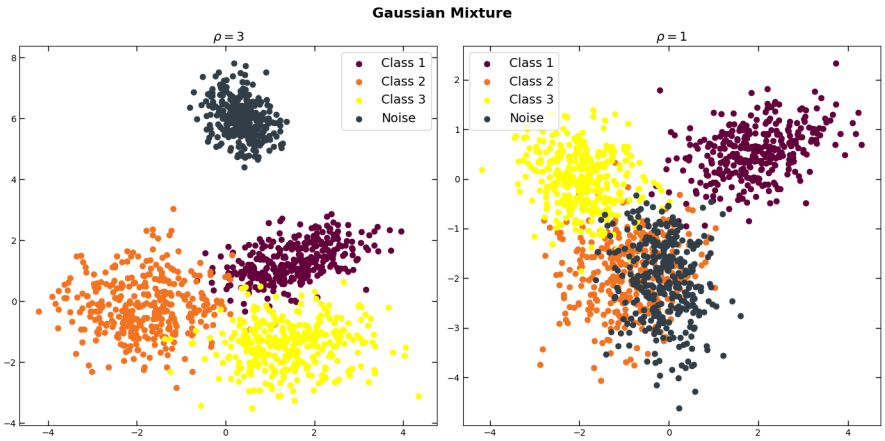


Figure 3.2: On both plots $d = 2$, $c = 3$, $R = 2$, $D = 1$ and $\varepsilon = 0.25$. With these parameters the plum, orange and yellow sources are well separated. On the left $\rho = 3$ and on the right $\rho = 1$. In the light of our theoretical analysis, we expect robustness for RFFM for the left setting and no robustness for the right setting.

In the absence of noise contamination in the target, all methods give excellent results, with a small advantage for RFFM, because the source distributions are easy to distinguish. Obviously, the results deteriorate as the level of contamination increases but the level of deterioration depends on whether we are in the close ($\rho = 1$) or far ($\rho = 10$) setting.

In the close setting, no method is robust. In our setting, with $\varepsilon = 0.2, 0.5, 0.7$ the "hard threshold" we discussed earlier are $4, 10$ and 14. As we can see neither RFFM nor EnergyQuantifier break this limits.

One the other hand when the new class is far away, RFFM significantly outperforms the others. As discussed following Corollary 3.2, this is because when the contamination is far away from the other classes, the Gaussian embedding of the noise distribution is close to orthogonal to the source KMEs. This property does not hold when a class is added close to the others and can therefore be more easily confounded. The results align well with our theoretical analysis.

While Theorem 3.3 and Corollary 3.2 also hold for the Energy kernel as well (assuming bounded data) or BBSE, we do not observe any robustness against noise. Again, this is in line with the theoretical study for BBSE, for which we expected no robustness. Concerning the Energy kernel, we surmise that the lack of robustness comes from the fact that $k(x, y)$ can

Percentage of noise ϵ	**Quantifier**	Number of classes = 5		
		dim = 2	dim = 5	dim = 10
0.0	CC	1.50 ; 3.0	0.78 ; 4.0	0.58 ; 4.0
0.0	BBSE	1.45 ; 3.0	0.45 ; 3.0	**0.37 ; 2.0**
0.0	RFFM	**0.65 ; 1.0**	**0.30 ; 1.0**	**0.32 ; 2.0**
0.0	EnergyQuantifier	1.21 ; 3.0	0.39 ; 2.0	0.38 ; 3.0
0.2	CC	**4.94 ; 1.0**	**4.30 ; 2.0**	**4.06 ; 1.75**
0.2	BBSE	8.54 ; 3.0	4.59 ; 2.0	4.22 ; 2.0
0.2	RFFM	**5.09 ; 2.0**	**4.34 ; 2.0**	4.49 ; 3.0
0.2	EnergyQuantifier	9.72 ; 4.0	7.40 ; 4.0	5.61 ; 4.0
0.5	CC	**10.68 ; 1.0**	**10.06 ; 1.0**	**10.01 ; 1.0**
0.5	BBSE	15.36 ; 3.0	11.36 ; 3.0	10.56 ; 2.0
0.5	RFFM	**9.76 ; 2.0**	**10.68 ; 2.0**	11.34 ; 3.0
0.5	EnergyQuantifier	15.68 ; 3.5	14.94 ; 4.0	13.05 ; 4.0
0.7	CC	**14.33 ; 2.0**	**14.03 ; 1.0**	**14.00 ; 2.0**
0.7	BBSE	17.85 ; 3.0	15.49 ; 3.0	14.39 ; 2.5
0.7	RFFM	**12.82 ; 1.0**	**14.39 ; 2.0**	**14.43 ; 2.5**
0.7	EnergyQuantifier	18.29 ; 4.0	16.80 ; 4.0	15.65 ; 4.0

Table 3.1: **Gaussian Mixture: Comparison of CC, BBSE, RFFM and EnergyQuantifier when $\rho = 1$ (close setting)**. The value before the semicolon is the geometric mean of the absolute error (multiplied by 100 for clarity) over 50 repetitions. Value after the semicolon is the median rank of a method relative to the other method in the same setting (same dimension, number of classes and percentage of noise). A bold value in a group is not significantly different from the best-performing method in the group (also in bold), as measured by a paired Wilcoxon test at $p < 0.01$.

Percentage of noise ϵ	**Quantifier**	Number of classes = 5		
		dim = 2	dim = 5	dim = 10
0.0	CC	1.50 ; 3.0	0.78 ; 4.0	0.58 ; 4.0
0.0	BBSE	1.45 ; 3.0	0.45 ; 3.0	**0.37 ; 2.0**
0.0	RFFM	**0.65 ; 1.0**	**0.30 ; 1.0**	**0.32 ; 2.0**
0.0	EnergyQuantifier	1.21 ; 3.0	0.39 ; 2.0	0.38 ; 3.0
0.2	CC	5.23 ; 2.0	4.59 ; 2.0	4.41 ; 2.0
0.2	BBSE	9.90 ; 3.0	5.93 ; 3.0	5.36 ; 3.0
0.2	RFFM	**1.17 ; 1.0**	**0.69 ; 1.0**	**0.70 ; 1.0**
0.2	EnergyQuantifier	13.83 ; 4.0	12.43 ; 4.0	9.43 ; 4.0
0.5	CC	10.87 ; 2.0	10.34 ; 2.0	10.25 ; 2.0
0.5	BBSE	17.42 ; 3.0	14.77 ; 3.0	13.53 ; 3.0
0.5	RFFM	**2.19 ; 1.0**	**1.66 ; 1.0**	**1.81 ; 1.0**
0.5	EnergyQuantifier	19.99 ; 4.0	18.83 ; 4.0	17.14 ; 4.0
0.7	CC	14.47 ; 2.0	14.22 ; 2.0	14.16 ; 2.0
0.7	BBSE	19.78 ; 3.0	18.41 ; 3.0	17.93 ; 3.0
0.7	RFFM	**2.54 ; 1.0**	**2.13 ; 1.0**	**2.62 ; 1.0**
0.7	EnergyQuantifier	20.65 ; 4.0	19.93 ; 4.0	19.32 ; 4.0

Table 3.2: **Gaussian Mixture: Comparison of CC, BBSE, RFFM and EnergyQuantifier when $\rho = 10$ (far setting)**. The value before the semicolon is the geometric mean of the absolute error (multiplied by 100 for clarity) over 50 repetitions. Value after the semicolon is the median rank of a method relative to the other method in the same setting (same dimension, number of classes and percentage of noise). A bold value in a group is not significantly different from the best-performing method in the group (also in bold), as measured by a paired Wilcoxon test at $p < 0.01$.

take large values even when $\|x - y\|$ is large, hence near-orthogonality of the noise distribution to the source does not hold in the corresponding feature space, in contrast to the Gaussian kernel.

Flow Cytometry

In this setting we use flow cytometry data from Metafora. For each patient we have access to four cell types: *T cells, B cells, NK cells and plasmocytes*. This dataset is a subset of a larger dataset that we present in detail in Chapter 5.

To perform quantification, we propose an APP approach with a Dirichlet shift as described in Section 2.4, where we remove all *T cells* to use them as noise. We choose to use T cells as noise because they are the most abundant cells in the sample, see Figure 3.3.

For each patient, we split the data set in half, using the first half as the source and resampling (with replacement) the other half so that the target proportions are $\alpha \sim \mathcal{D}(\tau\beta)$, where τ is a hyperparameter controlling the amount of shift and β is the proportions of the source. In Figure 3.4 we show the target proportions obtained for $\tau = 10$ and $\tau = 500$, the two values we chose in our experiments. We repeat the process 50 times for each patient, each ρ and each percentage of noise ε.

Percentage of noise ϵ	**Quantifier**	**Amount of shift**	
		$\rho = 10$	$\rho = 500$
0.0	CC	0.21 ; 2.0	0.21 ; 2.0
0.0	BBSE	**0.17 ; 2.0**	**0.17 ; 2.0**
0.0	RFFM	0.25 ; 2.0	0.23 ; 2.0
0.1	CC	3.53 ; 2.0	3.52 ; 2.0
0.1	BBSE	3.57 ; 3.0	3.57 ; 3.0
0.1	RFFM	**1.55 ; 1.0**	**1.54 ; 1.0**
0.2	CC	6.86 ; 2.0	6.85 ; 2.0
0.2	BBSE	7.08 ; 3.0	7.08 ; 3.0
0.2	RFFM	**2.97 ; 1.0**	**2.97 ; 1.0**
0.3	CC	10.18 ; 2.0	10.17 ; 2.0
0.3	BBSE	10.59 ; 3.0	10.60 ; 3.0
0.3	RFFM	**4.40 ; 1.0**	**4.42 ; 1.0**

Table 3.3: **Flow Cytometry: Comparison of CC, BBSE and RFFM**. The value before the semicolon is the geometric mean of the absolute error (multiplied by 100 for clarity) over 50 repetitions. Value after the semicolon is the median rank of a method relative to the other method in the same setting (same dimension, number of classes and percentage of noise). A bold value in a group is not significantly different from the best-performing method in the group (also in bold), as measured by a paired Wilcoxon test at $p < 0.01$.

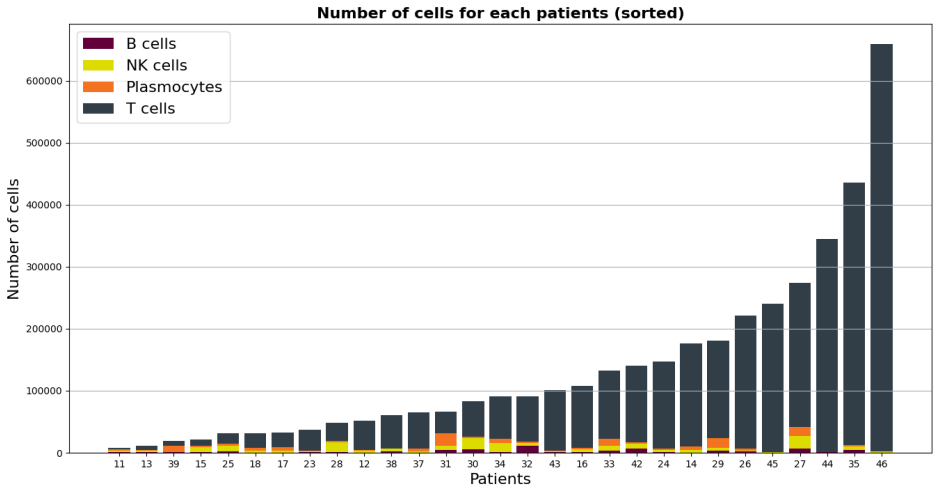


Figure 3.3: Number of cells for each patient. The patients are numbered from 1 to 46, but we discarded 17 of them because we did not have access to all the labels.

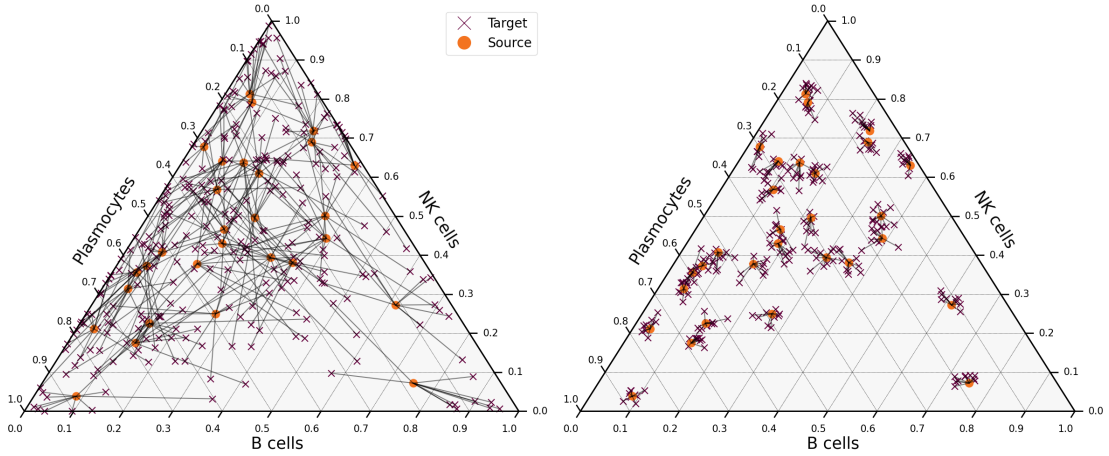


Figure 3.4: We show the source proportions as orange dots and the target proportions as purple crosses. For each repetition, the source proportions do not change (so there are only 29 orange dots), but the target proportions are sampled by a Dirichlet distribution centred on the source proportions. We draw a line between a target and its corresponding source. On the left we see the case $\tau = 10$ (higher variance) and on the right $\tau = 500$ (lower variance). Note that we only show 10 repetitions here for clarity.

The results can be found in the table 3.3. As we can see RFFM is robust to contamination of the samples by *T cells* not as robust as it was in the synthetic but robust enough to outperform the other methods and break the "hard threshold". This suggests that the embedding obtained using RFF, have interesting properties that we will use in Chapter 5 for cells identification. Note that the parameter ρ that control the *amount of shift* has no impact on the results, which is in line with the theoretical work. As we said, when we have the same number of points in each class of the labelled data as it is the case here, we are in the *best case scenario* and the bound (3.2) is minimal whatever the choice of proportions α^* (or $\tilde{\alpha}$).

3.5 Proofs

3.5.1 Proof of Theorem 3.1.

The building block used in the proof of the theorem is a vector norm concentration inequality in Hilbert spaces. More precisely, in this chapter, we use a Hoeffding-based inequality. This is an old result that has appeared in different versions in the literature. Appendix A is devoted to such results and contains a history of this argument as it has been used in the Kernel Mean Embedding literature, as well as a self-contained proof (See Theorem A.1).

We now turn to the proof of Theorem 3.1.

Proof. Let $\alpha \in \Delta^c$ be fixed. Later we will consider the choices $\alpha = \alpha^*$ or $\alpha = \tilde{\alpha}$ (the population resp. empirical class proportions in the target domain), and will specify this when needed, but a large part of the argument holds for any α.

Given a feature map Φ, let us use the notation $\hat{\boldsymbol{G}}$, introduce in Definition 3.2, for the gram matrix of the empirical source embedding. Let us note D_Φ the function defined by $D_\Phi(\mathbb{P}_1, \mathbb{P}_2) = \|\Phi(\mathbb{P}_1) - \Phi(\mathbb{P}_2)\|$. Note that D_Φ will be a distance if and only if the mapping Φ is injective and will be a pseudo-distance otherwise.

It holds:

$$
\begin{aligned}
D_\Phi\left(\sum_{i=1}^c \hat{\alpha}_i \hat{\mathbb{P}}_i, \sum_{i=1}^c \alpha_i \hat{\mathbb{P}}_i\right)^2 &= \left\|\sum_{i=1}^c \hat{\alpha}_i \Phi(\hat{\mathbb{P}}_i) - \sum_{i=1}^c \alpha_i \Phi(\hat{\mathbb{P}}_i)\right\|^2 \\
&= \left\|\sum_{i=1}^c (\hat{\alpha}_i - \alpha_i) \Phi(\hat{\mathbb{P}}_i)\right\|^2 \\
&= (\hat{\alpha} - \alpha)\ \hat{\boldsymbol{G}}\ (\hat{\alpha} - \alpha) \\
&\overset{(\dagger)}{\geq} \left(\min_{\substack{\|u\|_2=1 \\ \mathbf{1}^T u = 0}} u^T \hat{\boldsymbol{G}} u\right) \|\hat{\alpha} - \alpha\|^2 \\
&\overset{(\ddagger)}{=} \Delta_{\min}\ \|\hat{\alpha} - \alpha\|^2,
\end{aligned}
$$

with equality (‡) proven in Proposition 3.3. For (†), note that we could have written $\geq \left(\min_{\|u\|=1} u^T \hat{\boldsymbol{G}} u\right) \|\hat{\alpha} - \alpha\|^2$, i.e. the definition of the smallest eigenvalue of the empirical gram matrix $\hat{\boldsymbol{G}}$, but since $\sum_{i=1}^c (\hat{\alpha}_i - \alpha_i) = 0$, we can "add" the condition $\mathbf{1}^\top u = 0$ to the definition of the smallest eigenvalue.

Thus in order to bound $\|\hat{\alpha} - \alpha\|$ we have to upper-bound: $D_\Phi\left(\sum_{i=1}^c \hat{\alpha}_i \hat{\mathbb{P}}_i, \sum_{i=1}^c \alpha_i \hat{\mathbb{P}}_i\right)$. By the triangle inequality, this is upper-bounded by $D_\Phi\left(\sum_{i=1}^c \hat{\alpha}_i \hat{\mathbb{P}}_i, \hat{\mathbb{Q}}\right) + D_\Phi\left(\hat{\mathbb{Q}}, \sum_{i=1}^c \alpha_i \hat{\mathbb{P}}_i\right)$. By definition, $\hat{\alpha}$ is the minimiser of $D_\Phi\left(\sum_{i=1}^c \gamma_i \hat{\mathbb{P}}_i, \hat{\mathbb{Q}}\right)$ for $\gamma \in \Delta^c$, hence we have

$D_\Phi\left(\sum_{i=1}^c \hat{\alpha}_i \hat{\mathbb{P}}_i, \hat{\mathbb{Q}}\right) \leq D_\Phi\left(\sum_{i=1}^c \alpha_i \hat{\mathbb{P}}_i, \hat{\mathbb{Q}}\right)$. Hence, we can upper-bound the quantity by $2D_\Phi\left(\sum_{i=1}^c \alpha_i \hat{\mathbb{P}}_i, \hat{\mathbb{Q}}\right)$.

Using the triangle inequality once again, we can upper bound $D_\Phi\left(\sum_{i=1}^c \alpha_i \hat{\mathbb{P}}_i, \hat{\mathbb{Q}}\right)$ by :

$$D_\Phi\left(\sum_{i=1}^c \alpha_i \hat{\mathbb{P}}_i, \sum_{i=1}^c \alpha_i \mathbb{P}_i\right) + D_\Phi\left(\sum_{i=1}^c \alpha_i \mathbb{P}_i, \hat{\mathbb{Q}}\right). \tag{3.7}$$

Using Theorem A.1 and the union bound, it holds with probability greater than $1 - \delta/2$:

$$\begin{aligned} D_\Phi\left(\sum_{i=1}^c \alpha_i \hat{\mathbb{P}}_i, \sum_{i=1}^c \alpha_i \mathbb{P}_i\right) &\leq \sum_{i=1}^c \alpha_i D_\Phi\left(\hat{\mathbb{P}}_i, \mathbb{P}_i\right) \\ &\leq \sum_{i=1}^c \alpha_i C \frac{1 + \sqrt{2\log(2c/\delta)}}{\sqrt{n_i}} \\ &= C R_{c/\delta} \sum_{i=1}^c \frac{\alpha_i}{\sqrt{n_i}}, \end{aligned} \tag{3.8}$$

with $R_x = 1 + \sqrt{2 \log 2x}$.
Since $n_i = n\tilde{\beta}_i$, it holds $\sum_{i=1}^c \frac{\alpha_i}{\sqrt{n_i}} = \frac{1}{\sqrt{n}} \sum_{i=1}^c \frac{\alpha_i}{\sqrt{\tilde{\beta}_i}}$. If we write $w_i = \frac{\alpha_i}{\tilde{\beta}_i}$, then by Hölder's inequality:

$$\sum_{i=1}^c \frac{\alpha_i}{\sqrt{\tilde{\beta}_i}} = \sum_{i=1}^c \sqrt{\frac{\alpha_i}{\tilde{\beta}_i}} \sqrt{\alpha_i} = \sum_{i=1}^c (\sqrt{w_i}\sqrt{\alpha_i}) \leq \sqrt{\|w\|_1} \underbrace{\sqrt{\sum_{i=1}^c \alpha_i}}_{=1}. \tag{3.9}$$

So that $D_\Phi\left(\sum_{i=1}^c \alpha_i \hat{\mathbb{P}}_i, \sum_{i=1}^c \alpha_i \mathbb{P}_i\right) \leq C R_{c/\delta} \sqrt{\frac{\|w\|_1}{n}}$

We finally turn to bounding the term $D_\Phi\left(\sum_{i=1}^c \alpha_i \mathbb{P}_i, \hat{\mathbb{Q}}\right)$. This is the only point of the proof where we need to specify α. Under $(\mathcal{LS})$ this is equal to $D_\Phi\left(\sum_{i=1}^c \alpha_i \mathbb{Q}_i, \hat{\mathbb{Q}}\right)$. We now distinguish between two possibilities:

- If $\alpha = \alpha^*$, then $\sum_{i=1}^c \alpha_i^* \mathbb{Q}_i = \mathbb{Q}$, so that using Theorem A.1, it holds with probability greater than $1 - \delta/2$:

$$D_\Phi\left(\sum_{i=1}^c \alpha_i^* \mathbb{Q}_i, \hat{\mathbb{Q}}\right) = D_\Phi\left(\mathbb{Q}, \hat{\mathbb{Q}}\right) \leq C \frac{R_{1/\delta}}{\sqrt{m}}. \tag{3.10}$$

- For $\alpha = \tilde{\alpha}$, it holds $\tilde{\alpha}_i = m_i/m$, where m_i is the number of target sample points

of class i. We then get

$$D_\Phi\left(\sum_{i=1}^{c}\tilde{\alpha}_i\mathbb{Q}_i,\hat{\mathbb{Q}}\right) = \left\|\frac{1}{m}\sum_{i=1}^{c}m_i\Phi(\mathbb{Q}_i) - \frac{1}{m}\sum_{j=n+1}^{n+m}\Phi(x_j)\right\|$$
$$= \left\|\frac{1}{m}\sum_{j=n+1}^{n+m}\Phi(\mathbb{Q}_{y_j}) - \frac{1}{m}\sum_{j=n+1}^{n+m}\Phi(x_j)\right\|$$
$$= \left\|\frac{1}{m}\sum_{j=n+1}^{n+m}(\Phi(x_j) - \Phi(\mathbb{Q}_{y_j}))\right\|.$$

Now, notice that conditionally to the labels $(y_j)_{j=n+1}^{n+m}$, the target sample points x_j are independent, not identically distributed but with respective class conditional distribution $\mathbb{Q}_{y_j}$. We can therefore still appeal to Theorem A.1, and conclude that it holds with probability greater than $1-\delta/2$:

$$\left\|\frac{1}{m}\sum_{j=n+1}^{n+m}(\Phi(x_j) - \Phi(\mathbb{Q}_{y_j}))\right\| \leq C\frac{R_{1/\delta}}{\sqrt{m}}. \tag{3.11}$$

In both cases we therefore get the same bound for this last term.

We bound (3.7) by putting together (3.8), (3.9) and either (3.10) (for $\alpha = \alpha^*$) or (3.11) (for $\alpha = \tilde{\alpha}$). $R_{1/\delta} \leq R_{c/\delta}$, gives the first claimed inequality of the theorem.

To obtain the second claimed inequality, we can see that the worst case scenario for w is obtain when $\alpha_i = 1$ for the smallest $\tilde{\beta}_i$ and 0 for the others.

Hence, $\sqrt{\frac{\|w\|_1}{n}} \leq \max_i \frac{1}{\sqrt{n_i}}$. □

3.5.2 Proof of Theorem 3.3

First, let us prove a lemma that we will use during the proof.

Lemma 3.1. *Let $\mathcal{H}$ be a Hilbert space, $\mathcal{C}$ be a closed convex subset and $\bar{V}$ an affine subspace of $\mathcal{H}$ such that $\mathcal{C} \subset \bar{V} \subset \mathcal{H}$. For every $x \in \mathcal{H}$ we have*

$$\Pi_{\mathcal{C}}(x) = \Pi_{\mathcal{C}}(\Pi_{\bar{V}}(x)),$$

with $\Pi_{\mathcal{C}}$ and $\Pi_{\bar{V}}$ the minimum distance projection functions onto $\mathcal{C}$ and $\bar{V}$.

Proof. Let us take $x \in \mathcal{H}$ and $c \in \mathcal{C}$. Note $p = \Pi_{\bar{V}}(x)$, since $c \in \bar{V}$ we can use Pythagoras' theorem :

$$\|x-c\|^2 = \|x-p\|^2 + \|p-c\|^2.$$

The point c that minimises $\|x-c\|^2$, i.e $\Pi_{\mathcal{C}}(x)$, is the same point c that minimises $\|p-c\|^2$, i.e $\Pi_{\mathcal{C}}(\Pi_{\bar{V}}(x))$. □

Let us now prove Theorem 3.3.

Proof. Below, we use the notation α for α^*.

Given a feature map Φ, let us use the notation $D_\Phi(\mathbb{P}_1, \mathbb{P}_2) = \|\Phi(\mathbb{P}_1) - \Phi(\mathbb{P}_2)\|$. Note that D_Φ will be a distance if and only if the mapping Φ is injective and will be a pseudo-distance otherwise. We will use the notation introduced in Definition 3.2.

We write $\mathcal{C}_n = \text{ConvHull}\Big\langle \Phi(\hat{\mathbb{P}}_1), \cdots, \Phi(\hat{\mathbb{P}}_c) \Big\rangle$, the convex hull of the mapped empirical distributions $\hat{\mathbb{P}}_i$, and $\Pi_{\mathcal{C}_n}$ the projection onto this convex. In the same fashion let us write $\mathcal{C} = \text{ConvHull}\langle \Phi(\mathbb{P}_1), \cdots, \Phi(\mathbb{P}_c) \rangle$, the convex hull of the mapped sources distributions and the associated projection $\Pi_{\mathcal{C}}$. Finally let us write $\bar{V}$ the affine subspace generated by the $\Phi(\mathbb{P}_i)$, namely $\bar{V} = \{\sum_{i=1}^c \lambda_i \Phi(\mathbb{P}_i) \mid \sum_{i=1}^c \lambda_i = 1\}$.

With the same computation as in the proof of Theorem 3.1 (see 3.5.1), we have:

$$\|\hat{\alpha} - \alpha\| \leq \frac{1}{\sqrt{\Delta_{\min}}} D_\Phi\left(\sum_{i=1}^c \hat{\alpha}_i \hat{\mathbb{P}}_i, \sum_{i=1}^c \alpha_i \hat{\mathbb{P}}_i\right).$$

Once again we have to upper-bound $D_\Phi\Big(\sum_{i=1}^c \hat{\alpha}_i \hat{\mathbb{P}}_i, \sum_{i=1}^c \alpha_i \hat{\mathbb{P}}_i\Big)$. By the triangle inequality:

$$D_\Phi\left(\sum_{i=1}^c \hat{\alpha}_i \hat{\mathbb{P}}_i, \sum_{i=1}^c \alpha_i \hat{\mathbb{P}}_i\right) \leq \underbrace{\left\|\sum_{i=1}^c \hat{\alpha}_i \Phi(\hat{\mathbb{P}}_i) - \Pi_{\mathcal{C}}(\Phi(\mathbb{Q}))\right\|}_{(1)} + \underbrace{\left\|\Pi_{\mathcal{C}}(\Phi(\mathbb{Q})) - \sum_{i=1}^c \alpha_i \Phi(\hat{\mathbb{P}}_i)\right\|}_{(2)}. \tag{3.12}$$

Let us analyse the second term first. We use the triangle inequality:

$$\left\|\Pi_{\mathcal{C}}(\Phi(\mathbb{Q})) - \sum_{i=1}^c \alpha_i \Phi(\hat{\mathbb{P}}_i)\right\| \leq \left\|\Pi_{\mathcal{C}}(\Phi(\mathbb{Q})) - \sum_{i=1}^c \alpha_i \Phi(\mathbb{P}_i)\right\| + D_\Phi\left(\sum_{i=1}^c \alpha_i \mathbb{P}_i, \sum_{i=1}^c \alpha_i \hat{\mathbb{P}}_i\right).$$

The second term has already appeared in Section 3.5.1. Using 3.8 and 3.9, the second term can be bounded, with probability greater than $1 - \delta/2$, by $CR_{c/\delta}\sqrt{\|w\|_1/n}$.

For the first term we will require three elements:

1. $\sum_{i=1}^c \alpha_i \Phi(\mathbb{P}_i)$ lies in $\mathcal{C}$.
2. For all x, $\Pi_{\mathcal{C}}(x) = \Pi_{\mathcal{C}}(\Pi_{\bar{V}}(x))$, see Lemma 3.1.
3. $\Pi_{\mathcal{C}}$ is a contraction.

With that in mind,

$$\begin{aligned}\left\|\sum_{i=1}^{c}\alpha_i\Phi(\mathbb{P}_i)-\Pi_{\mathcal{C}}(\Phi(\mathbb{Q}))\right\| &= \left\|\Pi_{\mathcal{C}}\left(\sum_{i=1}^{c}\alpha_i\Phi(\mathbb{P}_i)\right)-\Pi_{\mathcal{C}}\Big(\Pi_{\bar{V}}\big(\Phi(\mathbb{Q})\big)\Big)\right\| \\ &\leq \left\|\sum_{i=1}^{c}\alpha_i\Big(\Phi(\mathbb{P}_i)-\Pi_{\bar{V}}\big(\Phi(\mathbb{Q})\big)\Big)\right\| \\ &\leq \max_i\left\|\Phi(\mathbb{P}_i)-\Pi_{\bar{V}}\big(\Phi(\mathbb{Q})\big)\right\| \\ &= B^{\parallel}(\mathbb{P},\mathbb{Q}).\end{aligned}$$

Therefore, we can upper bound the second term of (3.12) as such :

$$\left\|\Pi_{\mathcal{C}}(\Phi(\mathbb{Q}))-\sum_{i=1}^{c}\alpha_i\Phi(\hat{\mathbb{P}}_i)\right\| \leq CR_{c/\delta}\sqrt{\frac{\|w\|_1}{n}}+B^{\parallel}(\mathbb{P},\mathbb{Q}).$$

Let us turn to the first term of (3.12). By the representation of DFM as a projection presented in Figure 3.1, it holds $\sum_{i=1}^{c}\hat{\alpha}_i\Phi(\hat{\mathbb{P}}_i)=\Pi_{\mathcal{C}_n}(\Phi(\hat{\mathbb{Q}}))$. Using the triangle inequality,

$$\left\|\Pi_{\mathcal{C}_n}(\Phi(\hat{\mathbb{Q}}))-\Pi_{\mathcal{C}}(\Phi(\mathbb{Q}))\right\| \leq \left\|\Pi_{\mathcal{C}_n}(\Phi(\hat{\mathbb{Q}}))-\Pi_{\mathcal{C}_n}(\Phi(\mathbb{Q}))\right\|+\left\|\Pi_{\mathcal{C}_n}(\Phi(\mathbb{Q}))-\Pi_{\mathcal{C}}(\Phi(\mathbb{Q}))\right\|.$$

Since $\Pi_{\mathcal{C}_n}$ is a contraction, we have $\left\|\Pi_{\mathcal{C}_n}(\Phi(\hat{\mathbb{Q}}))-\Pi_{\mathcal{C}_n}(\Phi(\mathbb{Q}))\right\| \leq D_\Phi(\mathbb{Q},\hat{\mathbb{Q}})$.

That we can bound by $CR_{1/\delta}/\sqrt{m}$, like we did in Section 3.5.1 using Theorem A.1.

The only thing left to bound is the term $\|\Pi_{\mathcal{C}_n}(\Phi(\mathbb{Q}))-\Pi_{\mathcal{C}}(\Phi(\mathbb{Q}))\|$. For this we use a result by Alber et al. [1] relating the distance between convex projections onto two different convex sets in relation to their Hausdorff distance. Recall the definition of the Hausdorff distance between two sets:

Definition 3.4. Let X and Y be two non-empty subsets of a metric space (M,d). We define their Hausdorff distance $H(X,Y)$ by:

$$H(X,Y)=\max\left\{\sup_{x\in X}d(x,Y),\sup_{y\in Y}d(X,y)\right\}$$

Using Theorem 3.6 and Remark 3.7 of [1] :

$$\|\Pi_{\mathcal{C}_n}(\Phi(\mathbb{Q}))-\Pi_{\mathcal{C}}(\Phi(\mathbb{Q}))\| \leq \sqrt{2H(\mathcal{C},\mathcal{C}_n)(r+d)},$$

where $r=\operatorname{dist}(0,\Phi(\mathbb{Q}))$, $d=\max\{\operatorname{dist}(0,\mathcal{C}),\operatorname{dist}(0,\mathcal{C}_n)\}$ and 0 the origin. Since the problem has a geometrical nature and is invariant by translation, we can translate everything so that $\Phi(\mathbb{Q})$ is the origin of the space. Hence, the bound reads $\sqrt{2H(\mathcal{C},\mathcal{C}_n)d}$

with $d = \max\{\text{dist}(\Phi(\mathbb{Q}), \mathcal{C}), \text{dist}(\Phi(\mathbb{Q}), \mathcal{C}_n)\}$. Let us take care of the Hausdorff distance first:

$$\begin{aligned}
\sup_{x\in\mathcal{C}} d(x, \mathcal{C}_n) &= \sup_{x\in\mathcal{C}} \left\| x - \Pi_{\mathcal{C}_n}(x) \right\| \\
&= \sup_{\lambda\in\Delta^c} \inf_{\beta\in\Delta^c} \left\| \sum_i \lambda_i \Phi(\mathbb{P}_i) - \sum_i \beta_i \Phi(\hat{\mathbb{P}}_i) \right\| \\
&\leq \sup_{\lambda\in\Delta^c} \left\| \sum_i \lambda_i \Phi(\mathbb{P}_i) - \sum_i \lambda_i \Phi(\hat{\mathbb{P}}_i) \right\| \\
&\leq \max_i \; D_\Phi(\mathbb{P}_i, \hat{\mathbb{P}}_i).
\end{aligned}$$

A similar argument holds for $\sup_{x\in\mathcal{C}_n} \; d(x, \mathcal{C})$, and hence $H(\mathcal{C}, \mathcal{C}_n) \leq \max_i \; D_\Phi(\mathbb{P}_i, \hat{\mathbb{P}}_i)$.

We could simply bound d by 2 but we would obtain a loose bound. Instead, if we write $\Pi_{\mathcal{C}}(\Phi(\mathbb{Q})) = \sum_{i=1}^c \lambda_i \Phi(\mathbb{P}_i)$ then

$$\begin{aligned}
d(\Phi(\mathbb{Q}), \mathcal{C}_n) &= \|\Phi(\mathbb{Q}) - \Pi_{\mathcal{C}_n}(\Phi(\mathbb{Q}))\| \\
&\leq \left\| \Phi(\mathbb{Q}) - \sum_{i=1}^c \lambda_i \Phi(\hat{\mathbb{P}}_i) \right\| \\
&\leq \|\Phi(\mathbb{Q}) - \Pi_{\mathcal{C}}(\Phi(\mathbb{Q}))\| + \left\| \sum_{i=1}^c \lambda_i \Phi(\mathbb{P}_i) - \sum_{i=1}^c \lambda_i \Phi(\hat{\mathbb{P}}_i) \right\| \\
&\leq \|\Phi(\mathbb{Q}) - \Pi_{\mathcal{C}}(\Phi(\mathbb{Q}))\| + \max_i \; D_\Phi(\mathbb{P}_i, \hat{\mathbb{P}}_i).
\end{aligned}$$

Finally, using Theorem A.1, we get with probability higher than $1 - \delta/2$:

$$\begin{aligned}
\sqrt{2H(\mathcal{C}, \mathcal{C}_n)d} &\leq \sqrt{2} \; \max_i \; D_\Phi(\mathbb{P}_i, \hat{\mathbb{P}}_i) + \sqrt{2 \max_i D_\Phi(\mathbb{P}_i, \hat{\mathbb{P}}_i) \; \|\Phi(\mathbb{Q}) - \Pi_{\mathcal{C}}(\Phi(\mathbb{Q}))\|} \\
&= \sqrt{2} \; \max_i \; D_\Phi(\mathbb{P}_i, \hat{\mathbb{P}}_i) + \sqrt{2 \max_i D_\Phi(\mathbb{P}_i, \hat{\mathbb{P}}_i)} \; B^\perp(\mathbb{P}, \mathbb{Q}). \\
&\leq \sqrt{2} \; \max_i \; \frac{CR_{c/\delta}}{\sqrt{n_i}} + \sqrt{2 \max_i \; \frac{CR_{c/\delta}}{\sqrt{n_i}}} \; B^\perp(\mathbb{P}, \mathbb{Q}).
\end{aligned}$$

By putting everything together, with probability at least $(1 - \delta)$, we have:

$$\begin{aligned}
\|\hat{\alpha} - \alpha^*\| \leq \frac{1}{\sqrt{\Delta_{\min}}} \Big(& \frac{CR_{1/\delta}}{\sqrt{m}} + \sqrt{2} \; \max_i \; \frac{CR_{c/\delta}}{\sqrt{n_i}} + \sqrt{2 \max_i \; \frac{CR_{c/\delta}}{\sqrt{n_i}}} \; B^\perp(\mathbb{P}, \mathbb{Q}) \\
& + CR_{c/\delta} \sqrt{\frac{\|w\|_1}{n}} + B^\parallel(\mathbb{P}, \mathbb{Q}) \Big) \\
\leq \frac{1}{\sqrt{\Delta_{\min}}} \Big(& c_1 \; \max_i \; \frac{CR_{c/\delta}}{\sqrt{n_i}} + \frac{CR_{1/\delta}}{\sqrt{m}} + \sqrt{2 \max_i \; \frac{CR_{c/\delta}}{\sqrt{n_i}}} \; B^\perp + B^\parallel \Big)
\end{aligned}$$

with $c_1 = 1 + \sqrt{2} \leq 3$. □

3.5.3 Proof of Corollary 3.2

We directly apply Theorem 2 with the "dummy" class $\Phi(\mathbb{P}_0) \coloneqq 0$. If we write $\bar{\boldsymbol{G}}$ the Gram matrix of $\left\{\Phi(\mathbb{P}_0), \Phi(\hat{\mathbb{P}}_1) \cdots \Phi(\hat{\mathbb{P}}_c)\right\}$ then, as the first column of this matrix is zero,

$$\Delta_{\mathsf{min}}(\bar{\boldsymbol{G}}) = \min_{\substack{\|u\|_2=1 \\ \mathbf{1}^T u=0}} u^T \bar{\boldsymbol{G}} u = \min_{\|u\|=1} u^T \hat{\boldsymbol{G}} u = \lambda_{\mathsf{min}}.$$

The affine subspace $\mathrm{AffSpan}\{\Phi(\mathbb{P}_i), i \in [0, \cdots, c]\}$ is equal to $V \coloneqq \mathrm{Span}\{\Phi(\mathbb{P}_i), i \in [c]\}$ and the convex hull $\mathrm{ConvHull}\{\Phi(\mathbb{P}_i), i \in [0, \cdots, c]\}$ is equal the interior of C.

As we are in the open set label shift setting : $\mathbb{P}_i = \mathbb{Q}_i$ and hence

$$B^{\|}(\mathbb{P}, \mathbb{Q}) = \max_i \ \|\Phi(\mathbb{P}_i) - \Pi_V(\Phi(\mathbb{Q}_i))\| = \|\Pi_V(\Phi(\mathbb{Q}_0))\|$$

The "orthogonal" term $B^{\perp}(\mathbb{P}, \mathbb{Q})$ can be bounded by $\sqrt{\alpha_0 \ \|\Phi(\mathbb{Q}_0)\|}$ as follows:

$$\begin{aligned} B^{\perp}(\mathbb{P}, \mathbb{Q})^2 &= \left\|\Phi(\mathbb{Q}) - \Pi_{\mathrm{int}(C)}(\Phi(\mathbb{Q}))\right\| \\ &\overset{(\dagger)}{\leq} \left\|\sum_{i=1}^{c} \alpha_i^* \Phi(\mathbb{P}_i) + \alpha_0^* \Phi(\mathbb{Q}_0) - \sum_{i=1}^{c} \alpha_i^* \Phi(\mathbb{P}_i)\right\| \\ &= \alpha_0 \|\Phi(\mathbb{Q}_0)\|, \end{aligned}$$

for the inequality (†), we use the fact that $\sum_{i=1}^{c} \alpha_i^* \Phi(\mathbb{P}_i) \in \mathrm{int}(C)$.

3.5.4 Proof of Proposition 3.3

Proof. Let us take $u \in \mathbb{R}^c$ such that $\|u\| = 1$, $\mathbf{1}^T u = 0$ and $\boldsymbol{P}$ the projection matrix on $< \mathbf{1} >^{\perp}$, such that $\boldsymbol{P}u = u$. We have:

$$\begin{aligned} u^T \hat{\boldsymbol{G}} u &= (\boldsymbol{P}u)^T \hat{\boldsymbol{G}} (\boldsymbol{P}u) \\ &= u^T (\boldsymbol{P}\hat{\boldsymbol{G}}\boldsymbol{P}) u. \end{aligned}$$

Observe that $\boldsymbol{P}\hat{\boldsymbol{G}}\boldsymbol{P}$ is a symmetric matrix of rank $c-1$, hence the eigenvectors v_i (associated to the eigenvalues $\lambda_1 \geq \lambda_2 \cdots \geq \lambda_c = 0$) form an orthonormal basis of $\mathbb{R}^c$. Since $u \in < \mathbf{1} >^{\perp}$, then $u \in \mathrm{Span}(v_1, \cdots, v_{c-1})$ so that there exist $\alpha \in \mathbb{R}^{c-1}$ such that $u = \sum_{i=1}^{c-1} \alpha_i v_i$, and since $\|u\| = 1$ then $\|\alpha\| = 1$.

With that in mind :

$$\begin{aligned} u^T(\boldsymbol{P}\hat{\boldsymbol{G}}\boldsymbol{P})u &= \left(\sum_{i=1}^{c-1}\alpha_i v_i\right)^T\left(\sum_{i=1}^{c-1}\alpha_i(\boldsymbol{P}\hat{\boldsymbol{G}}\boldsymbol{P})v_i\right) \\ &= \left(\sum_{i=1}^{c-1}\alpha_i v_i\right)^T\left(\sum_{i=1}^{c-1}\alpha_i\lambda_i v_i\right) \\ &= \sum_{i,j=1}^{c-1}\lambda_i\alpha_i\alpha_j\langle v_i, v_j\rangle \\ &= \sum_{i=1}^{c-1}\lambda_i\alpha_i^2. \end{aligned}$$

Hence $\min_{\substack{\|u\|_2=1 \\ \mathbf{1}^T u=0}} u^T\hat{\boldsymbol{G}}u$ is equal to $\min_{\|\alpha\|=1}\sum_{i=1}^{c-1}\lambda_i\alpha_i^2$. Using the change of variable $\beta_i = \alpha_i^2$, this is equivalent to find : $\min_{\beta\in\Delta^{c-1}} < \lambda, \beta >$, which is equal to λ_{c-1}. A straightforward computation of $\boldsymbol{P}\hat{\boldsymbol{G}}\boldsymbol{P}$ with $\boldsymbol{P} = I_c - \frac{1}{c}\,\mathbf{1}^T\mathbf{1}$, shows that $\boldsymbol{P}\hat{\boldsymbol{G}}\boldsymbol{P} = \hat{\boldsymbol{M}}$. □

3.5.5 Proof of Theorem 3.2

Before we proceed, it is important to note that for any number of classes c and any vectors $\{b_1, \cdots, b_c\}$, if we write $\hat{\boldsymbol{G}}$ the gram matrix of the vectors, Proposition 3.3 can be reformulated as :

$$\begin{aligned} \Delta_{\mathsf{min}} &\coloneqq \Delta_{\mathsf{min}}(b_1, \cdots, b_c) \\ &= \min_{\substack{\|u\|_2=1 \\ \mathbf{1}^T u=0}} u^T\hat{\boldsymbol{G}}u \\ &= \min_{\mathbf{1}^T u=0} \frac{u^T\hat{\boldsymbol{G}}u}{\|u\|_2^2}. \end{aligned}$$

In the same fashion, note that the definition of Γ (Definition 3.3) can also be reformulated as:

$$\begin{aligned}\Gamma \coloneqq \Gamma(b_1, \cdots, b_c) &= \min_{(I_1,I_2)\in\mathcal{P}_2(c)} d^2(C_{I_1}, C_{I_2}),\\ &= \min_{\substack{\|v\|_1=2\\ \mathbf{1}^T v=0}} \left\|\sum_{i=1}^{c} v_i b_i\right\|^2\\ &= \min_{\substack{\|v\|_1=2\\ \mathbf{1}^T u=0}} v^T \hat{\boldsymbol{G}} v\\ &= 4 \min_{\mathbf{1}^T v=0} \frac{v^T \hat{\boldsymbol{G}} v}{\|v\|_1^2}.\end{aligned}$$

Finally, we will need the following lemma during the proof. It can be proved using the Lagrange multiplier.

Lemma 3.2. *If we note Δ^k the $(k-1)$-dimensional simplex, we have:*

$$\inf_{u\in\Delta^k} \|u\|_2^2 = \frac{1}{k}$$

We can move on to the proof of Theorem 3.2.

Proof. Let us write u^* the miniser of $\min_{\mathbf{1}^T u=0} \frac{u^T \hat{\boldsymbol{G}} u}{\|u\|_2^2}$ and v^* the minimiser of $\min_{\mathbf{1}^T v=0} \frac{v^T \hat{\boldsymbol{G}} v}{\|v\|_1^2}$.

For the upper bound we can do the following:

$$\frac{\Delta_{\min}}{\Gamma/4} = \frac{\min_{\mathbf{1}^T u=0} \frac{u^T \hat{\boldsymbol{G}} u}{\|u\|_2^2}}{\min_{\mathbf{1}^T v=0} \frac{v^T \hat{\boldsymbol{G}} v}{\|v\|_1^2}} \leq \frac{\|v^*\|_1^2}{\|v^*\|_2^2} = \frac{4}{\|v^*\|_2^2} \leq \frac{4}{\min\limits_{\substack{\|x\|_1=2\\ \mathbf{1}^T x=0}} \|x\|_2^2}.$$

Therefore we have the upper bound:

$$\Delta_{\min} \leq \left(\min_{\substack{\|x\|_1=2\\ \mathbf{1}^T x=0}} \|x\|_2^2\right)^{-1} \Gamma. \tag{3.13}$$

In the same fashion, for the lower bound:

$$\frac{\Gamma/4}{\Delta_{\min}} = \frac{\min_{\mathbf{1}^T v=0} \frac{v^T \hat{\boldsymbol{G}} v}{\|v\|_1^2}}{\min_{\mathbf{1}^T u=0} \frac{u^T \hat{\boldsymbol{G}} u}{\|u\|_2^2}} \leq \frac{\|u^*\|_2^2}{\|u^*\|_1^2} = \frac{1}{\|u^*\|_1^2} \leq \frac{1}{\min\limits_{\substack{\|y\|_2=1\\ \mathbf{1}^T y=0}} \|y\|_1^2}.$$

Therefore we have the lower bound:

$$\Delta_{\min} \geq 1/4 \left(\min_{\substack{\|y\|_2=1\\ \mathbf{1}^T y=0}} \|y\|_1^2\right) \Gamma. \tag{3.14}$$

All is left to do to find the constants is two find the solutions of the two minimisation problems in Equation (3.13) and (3.14).

For the problem of Equation (3.13), note that the condition $\|x\|_1 = 2$ and $\mathbf{1}^T x = 0$ together imply that the optimal solution x^* satisfy $\left\|x_+^*\right\|_1 = \left\|x_-^*\right\|_1 = 1$ where, x_+^* are the positive coordinate of x^* and x_-^* are the negative coordinate. We do not know how many coordinates are positive in x^*, but if we write that number c_1, then $x_+^* \in \Delta^{c_1}$ and $x_-^* \in \Delta^{c-c_1}$. Hence, we can rewrite the problem as:

$$\min_{\substack{\|x\|_1=2 \\ \mathbf{1}^T x=0}} \|x\|_2^2 = \min_{c_1<c} \min_{\substack{x_1\in\Delta^{c_1} \\ x_2\in\Delta^{c-c_1}}} \|x_1\|^2 + \|x_2\|^2.$$

Using Lemma 3.2:

$$\begin{aligned}
\min_{\substack{\|x\|_1=2 \\ \mathbf{1}^T x=0}} \|x\|_2^2 &= \min_{c_1<c} \min_{\substack{x_1\in\Delta^{c_1} \\ x_2\in\Delta^{c-c_1}}} \|x_1\|^2 + \|x_2\|^2. \\
&= \min_{c_1<c} \frac{1}{c_1} + \frac{1}{c-c_1} \\
&= \frac{4}{c}, \text{ if c even.} \\
&= \frac{4c}{(c+1)(c-1)}, \text{ if c odd.}
\end{aligned}$$

This conclude the proof of the upper bound.

For the problem of Equation (3.14), note that if we take $\bar{y} = \left(-1/\sqrt{2}, 1/\sqrt{2}, 0 \cdots, 0\right)$, $\bar{y}$ satisfy the two conditions and $\|\bar{y}\|_1^2 = 2$. Therefore, the value we search is at least smaller than 2. Let us suppose that it is strictly inferior.

We have $\|y^*\|_1 < \sqrt{2}$. Since $\mathbf{1}^T y^* = 0$, then $\left\|y_+^*\right\|_1 = \left\|y_-^*\right\|_1$. We can now conclude:

$$\begin{aligned}
\left\|y_+^*\right\|_2 \leq \left\|y_+^*\right\|_1 &= \frac{1}{2}\|y^*\|_1 < \frac{1}{\sqrt{2}} \\
\left\|y_-^*\right\|_2 \leq \left\|y_-^*\right\|_1 &= \frac{1}{2}\|y^*\|_1 < \frac{1}{\sqrt{2}},
\end{aligned}$$

which leads to the following contradiction:

$$\|y^*\|_2^2 = \left\|y_+^*\right\|_2^2 + \left\|y_-^*+\right\|_2^2 < 1.$$

Therefore, $\min\limits_{\substack{\|y\|_2=1 \\ \mathbf{1}^T y=0}} \|y\|_1^2 = 2$, which conclude proof of the lower bound. □

Chapter 4

Covariance-aware Distribution Feature Matching

In this chapter, we study an extension of *Distribution Feature Matching* for label shift quantification called Mahalanobis Distribution Feature Matching or M-DFM.

What we propose in this chapter is to take into account the covariance structure of the data by changing the distance. The metric used in the previous chapter was the ambient metric of the space on which we embed the data, the L_2 distance if we used a vectorisation or the *maximum mean discrepancy* if we used kernel mean embedding, in this chapter we will use a *Mahalanobis*-type distance using a matrix (or linear operator) M.

We present a general performance bound for this new framework under *label shift* by replacing the Hoeffding theorem, which was at the heart of the proof of Theorem 3.1, with a Bernstein theorem, which allows us to derive a bound that reveals the covariance of the source embeddings. Using this new theorem, we present a criterion for the choice of M as well as a close-form solution of its minimiser.

Finally, we present a numerical study on simulated and real data sets to demonstrate the usefulness of this extension.

Contents

4.1 Introduction

We keep the same notation as in Chapter 3. See the corresponding paragraph in Section 3.1.

We will note $\Phi\colon \mathcal{X} \to \mathcal{F}$ a feature map from the covariate space to a Hilbert space $\mathcal{F}$. The embedding could be, for instance, the output of a (soft-)classifier, a hidden layer of a neural network, the implicit embedding function induced by a kernel k, or the Random Fourier Feature Embedding.

Whatever the choice of Φ, we note $\Phi(\mathbb{P}) \coloneqq \mathbb{E}_{\mathbb{P}}[\Phi(X)]$ the mean embedding of $\mathbb{P}$ by Φ and $\Sigma_i \coloneqq \mathbb{E}_{\mathbb{P}}[(\Phi(X) - \Phi(\mathbb{P}))^{\otimes 2}]$ the covariance matrix (or covariance operator) of the push-forward distribution of $\mathbb{P}$ by Φ.

Finally, we denote by $M\colon \mathcal{F} \to \mathcal{F}$ a matrix (or a linear operator) that is independent of the data.

4.1.1 Distribution Feature Matching

In the previous chapter we took the distribution matching point of view of quantification, presented in Equation (2.31), where we searched for the mixture of $\hat{\mathbb{P}}_i$ that is closest to $\hat{\mathbb{Q}}$. To do this, we have proposed a general framework called Distribution Feature Matching (**DFM**), where we embed the distributions by taking the mean of the feature map Φ under the distributions. What we have discussed in Sections 2.3 and 3.2 is that some methods such as ACC/BBSE [46, 48, 77] or Kernel methods [70, 73, 85] can be recast as particular instances of DFM. Let us recall the definition of the DFM framework.

Definition 4.1 (Distribution Feature Matching)**.** Using the notations defined above, we call *Distribution Feature Matching* (DFM) any estimation procedure that can be formulated as the minimiser of the following problem:

$$\hat{\alpha} = \operatorname*{arg\,min}_{\alpha \in \Delta^c} \left\| \sum_{i=1}^{c} \alpha_i \Phi(\hat{\mathbb{P}}_i) - \Phi(\hat{\mathbb{Q}}) \right\|_{\mathcal{F}}^2 \tag{$\mathcal{P}$}$$

where Δ^c is the $(c-1)$-dimensional simplex.

In Section 2.2.2 we presented and discussed methods that use the output of a soft classifier. The idea behind this category of methods was to use the full information given by a soft classifier and not just the argmax. What we propose in this chapter is of the same nature: use all the information provided by the feature map Φ.

The embeddings proposed in DFM use the **means** of the embeddings and thus completely forget about the points that were used to compute those means. In particular the method is completely agnostic to the **variance** both in the algorithm and in the theoretical analysis, see Theorem 3.1. The algorithm ignores the covariance by nature (we only compute the mean), while the theorem ignores it because at the heart of the proof is a Hoeffding inequality (Theorem A.1) that hides the variances under a constant C.

Figure 4.1 illustrates why we would want to take the variance into account.

4.2 Mahalanobis Distribution Feature Matching

Let us introduce our general framework, Mahalanobis Distribution Feature Matching or M-DFM.

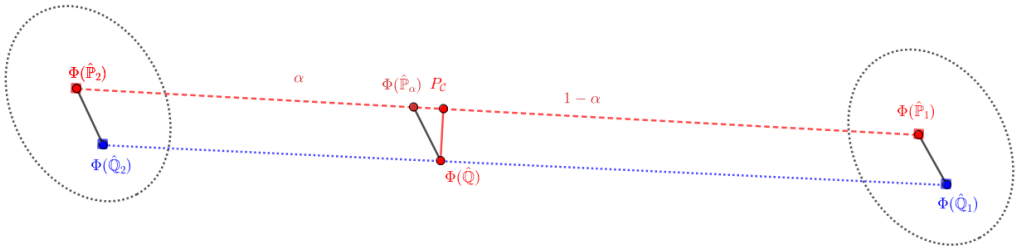


Figure 4.1: This Figure is analogous to Figure 3.1. It shows that the DFM procedure can be thought of as the projection onto the convex hull of the source embeddings of the target embedding, except that this time the covariance matrices are not multiples of the identity as we have showed in Figure 3.1. If, instead of projecting with respect to the ambient metric, as we do in DFM, we had projected $\Phi(\hat{\mathbb{Q}})$ with respect to the metric induced by the covariances, we would have been closer to the real solution.

Definition 4.2 (Mahalanobis Distribution Feature Matching)**.** Using the notations defined above, we call *Mahalanobis Distribution Feature Matching* (M-DFM) any estimator that can be formulated as the minimiser of the following problem:

$$\hat{\alpha} = \underset{\alpha \in \Delta^c}{\arg\min} \left\| M \left(\sum_{i=1}^{c} \alpha_i \Phi(\hat{\mathbb{P}}_i) - \Phi(\hat{\mathbb{Q}}) \right) \right\|_{\mathcal{F}}^2 \qquad (\mathcal{P}_{\mathcal{M}})$$

where Δ^c is the $(c-1)$-dimensional simplex.

Strictly speaking, $(\mathcal{P}_{\mathcal{M}})$ can only be called Mahalanobis if we choose $M = \Sigma^{-1/2}$, where Σ denotes the covariance matrix (or operator) of the true distributions $\sum_{i=1}^{c} \alpha_i^* \Phi(\mathbb{P}_i)$. Nevertheless, we stick to this terminology because the true Mahalanobis distance and our new approach share the same philosophy : taking into account the variance of the distributions in the metric.

4.2.1 Link with the Maximum Kernel Fisher Discriminant Ratio

As recall from Section 3.2.1, when the function Φ used is the Kernel Mean Embedding [87], the function $D_\Phi(\mathbb{P}, \mathbb{Q}) = \|\Phi(\mathbb{P}) - \Phi(\mathbb{Q})\|$ is called the Maximum Mean Discrepancy [61], originally proposed for two-sample testing.

It is a known fact, at least since the works of Harchaoui et al. [31, 67] that MMD is suboptimal. More recently, Hagrass et al. [66] proposed a in-depth analysis of the suboptimality

of MMD and show that "*the popular MMD (maximum mean discrepancy) two-sample test is not optimal in terms of the separation boundary measured in Hellinger distance*".

Harchaoui and his co-author proposed to "*incorporate the covariance structure of the probability distributions into the test statistics*"[1] in the same fashion as what we propose in this chapter. The resulting test statistic is called the *Maximum Kernel Fisher Discriminant Ratio*.

4.3 Theoretical Analysis of M-DFM

We now provide statistical guarantees for M-DFM. First, we note that the theoretical analysis we gave for DFM in Section 3.3 can still be applied to M-DFM when we apply it to $\left(M\Phi(\hat{\mathbb{P}}_i)\right)_{i=1}^c$. We call this analysis a *Hoeffding based analysis* of M-DFM because the proofs are based on a Hoeffding inequality presented in Appendix A.1. The shortcoming of this theorem is that it hides the covariance information under a constant C, which explains why the covariance of the source embeddings does not appear in the bounds we presented.

To overcome this, we propose an analysis of M-DFM, this time using a Bernstein inequality presented in Appendix A.2.

The matrix M is assumed to be **independent** of the data in this analysis.

4.3.1 Hoeffding based Analysis of M-DFM

We make the same identifiability hypothesis on the mapping Φ, namely the linear independence of the distribution embeddings $\mathcal{A}_1$ and the boundedness hypothesis $\mathcal{A}_2$, as we did for DFM. See Section 3.3 for further explanation.

However, the linear independence of the source embeddings does not imply the linear independence of $\left(M\Phi(\hat{\mathbb{P}}_i)\right)_{i=1}^c$. Therefore we make the following hypothesis about the operator M.

$$\sum_{i=1}^{c} \beta_i \Phi(\mathbb{P}_i) \in \ker(M) \iff \beta = 0, \tag{$\mathcal{A}_3$}$$

where $\ker(M)$ denotes the kernel of the linear operator M. This condition is checked, for example, when M has full rank.

The boundedness of the source embeddings implies the boundedness of $\left(M\Phi(\hat{\mathbb{P}}_i)\right)_{i=1}^c$ but the constant change. We have:

$$\|M\Phi(x)\| \leq \|M\|_{\text{op}} C,$$

where $\|M\|_{\text{op}}$ is the operator norm of M.

We introduce the following notation and give a corollary of Theorem 3.1 for M-DFM.

[1] Their test is a kernel extension of the classical Hotteling's T^2-statistic [76]

Definition 4.3 (Gram matrices)**.** We denote $\hat{\boldsymbol{G}}$ the Gram matrix, and $\hat{\boldsymbol{G}}^M$ the Mahalanobis Gram matrix of the empirical source embedding.

That is, if we write $\hat{V} = \left[\Phi(\hat{\mathbb{P}}_i), \cdots, \Phi(\hat{\mathbb{P}}_c)\right] \in \mathbb{R}^{D\times c}$ then $\hat{\boldsymbol{G}} = \hat{V}^\top \hat{V}$, i.e. $\hat{\boldsymbol{G}}_{i,j} = \left\langle \Phi(\hat{\mathbb{P}}_i), \Phi(\hat{\mathbb{P}}_j) \right\rangle$ and $\hat{\boldsymbol{G}}^M = \hat{V}^\top (M^\top M)\hat{V}$, i.e. $\hat{\boldsymbol{G}}^M_{i,j} = \left\langle M\Phi(\hat{\mathbb{P}}_i), M\Phi(\hat{\mathbb{P}}_j) \right\rangle$.

Furthermore, let $\lambda_{\min}(N)$ be the smallest eigenvalue of any matrix N, i.e. $\lambda_{\min}(N) \coloneqq \min_{\|u\|=1} u^T N u$, and $\Delta_{\min}(N)$ be the smallest eigenvalue of any matrix N restricted to the space $< 1 >^\perp$, i.e. $\Delta_{\min}(N) \coloneqq \min_{\substack{\|u\|_2=1 \\ \mathbf{1}^T u=0}} u^T N u$. In particular, it holds $\Delta_{\min}(N) \geq \lambda_{\min}(N)$.

A direct application of Theorem 3.1 gives the following corollary.

Corollary 4.1 (Hoeffding based)**.** *If the label shift hypothesis $\mathcal{LS}$ holds, if the mapping Φ satisfies the assumptions $(\mathcal{A}_1)$ and $(\mathcal{A}_2)$, if the operator M satisfies the assumption $\mathcal{A}_3$, then for any $\delta \in (0,1)$, with probability greater than $1-\delta$, the solution $\hat{\alpha}$ of $(\mathcal{P}_{\mathcal{M}})$ satisfies:*

$$\|\hat{\alpha} - \alpha^*\| \leq \frac{2\|M\|_{\mathrm{op}} C R_{c/\delta}}{\sqrt{\Delta_{min}(\hat{\boldsymbol{G}}^M)}} \left(\sqrt{\frac{\|w\|_1}{n}} + \frac{1}{\sqrt{m}} \right) \tag{4.1}$$

$$\leq \frac{2\|M\|_{\mathrm{op}} C R_{c/\delta}}{\sqrt{\Delta_{min}(\hat{\boldsymbol{G}}^M)}} \left(\frac{1}{\sqrt{\min_i n_i}} + \frac{1}{\sqrt{m}} \right) \tag{4.2}$$

with $w = \frac{\alpha^*}{\hat{\beta}_i}$ and $R_x = 1 + \sqrt{2\log 2x}$.

The same result holds when replacing α^* by the (unobserved) vector of empirical proportions $\tilde{\alpha}$ in the target sample, both on the left-hand side and in the definition of w.

The proof of Corollary 4.1 is exactly the same as the proof of Theorem 3.1, which can be found in Section 3.5.1. The conditions $\mathcal{A}_1$ and $\mathcal{A}_3$ ensure that $\Delta_{\min}(\hat{\boldsymbol{G}}^M)$ is not null, while the condition $\mathcal{A}_2$ is responsible for the constant $\|M\|_{\mathrm{op}} C$.

As discussed in detail in Section 3.3.2, $\Delta_{\min}(\hat{\boldsymbol{G}})$ is an empirical quantity that can be minimised on the training dataset to choose the feature mapping Φ, or equivalently to choose the bandwidth σ of the Gaussian kernel. We discussed the geometric property of this quantity and explained why $\Delta_{\min}$ was the natural quantity to appear rather than $\lambda_{\min}$ in Sections 3.3.1 and 3.3.2.

For M-DFM the situation is different because we have two hyperparameters to choose: Φ (or the bandwidth σ) and an operator M. These two choices are interdependent, as the best M depends on the embeddings and the best Φ is likely to depend on the choice of M. The criterion for M-DFM is more complex than that for DFM because it involves both the Mahalanobis Gram matrix $\hat{\boldsymbol{G}}^M$ and the operator norm of M.

The following theorem shows that, according to Corollary 4.1, no matrix M can be better than the identity.

Theorem 4.1. *For any given feature map Φ which satisfies Conditions $\mathcal{A}_1$ and $\mathcal{A}_2$, for any linear operators M that satisfies Conditions $\mathcal{A}_3$ we have*

$$\frac{\|M\|_{\text{op}}}{\sqrt{\Delta_{min}(\hat{\boldsymbol{G}}^M)}} \geq \frac{\|I_d\|_{\text{op}}}{\sqrt{\Delta_{min}(\hat{\boldsymbol{G}})}}. \tag{4.3}$$

The proof can be found in Section 4.6.1.

In other words, we can not do better than taking $M = I_d$, and the bound (4.1) is the same as in (3.2), where we use DFM. From this analysis, we have nothing to gain by using a Mahalanobis distance.

We have called Corollary 4.1 a *Hoeffding*-based analysis of M-DFM because the heart of the proofs is based on a Hoeffding inequality, which is presented and proved in Appendix A.1.

The Hoeffding inequality is a probability upper bound on the distance between a sum of bounded independent random variables and the mean of the expectations. However, this upper bound hides the covariance information below the bound of the random variables, in our case $\|M\|_{\text{op}}C$. It is therefore not surprising that this analysis does not allow us to obtain a value of M depending on the variance.

4.3.2 Bernstein-based Analysis of M-DFM

To take into account the variance and to get a better criterion for the choice of the operator M, we have to rely on a Bernstein-based inequality instead of a Hoeffding inequality. We describe this variance-aware alternative to Hoeffding in detail in Appendix A.2, and we also present a variant based on Bennett in Appendix A.3, which we will not use in our analysis because the constants are less sharp.

The proof follows the same path as that of Theorem 3.1, except that we replace Theorem A.1 with Theorem A.3. See the proof in Section 4.6.2.

Theorem 4.2 (Bernstein based)**.** *If the Label Shift hypothesis $\mathcal{LS}$ holds, if the mapping Φ satisfies assumptions $(\mathcal{A}_1)$ and $(\mathcal{A}_2)$, if the operator M satisfies assumptions $\mathcal{A}_3$, then for any $\delta \in (0,1)$, with probability greater than $1-\delta$, the solution $\hat{\alpha}$ of $(\mathcal{P}_{\mathcal{M}})$ satisfies:*

$$\|\hat{\alpha} - \tilde{\alpha}\| \leq R_1(\delta, c)\frac{\|M\|_{\text{op}}C}{\sqrt{\Delta_{min}(\hat{\boldsymbol{G}}^M)}}\left(\frac{\|w\|_1}{n} + \frac{1}{m}\right) \tag{4.4}$$

$$+ R_2(\delta, c)\sqrt{\frac{Tr(M\Sigma_{\tilde{\alpha}}M^\top)}{\Delta_{min}(\hat{\boldsymbol{G}}^M)}}\left(\sqrt{\frac{\|w\|_1}{n}} + \frac{1}{\sqrt{m}}\right), \tag{4.5}$$

with $w = \tilde{\alpha}/\tilde{\beta}$, $R_1(\delta, c) = \frac{4}{3}\log(4c/\delta)$, $R_2(\delta, c) = 2\sqrt{2\log(4c/\delta)}$ and $\Sigma_{\tilde{\alpha}} = \sum_{i=1}^{c}\tilde{\alpha}\Sigma_i$.

Remark. The theorem is still true if we replace $\Delta_{\min}(\hat{\boldsymbol{G}}^M)$ with $\lambda_{\min}(\hat{\boldsymbol{G}}^M)$, but unlike Theorem 3.1 we can not replace the empirical proportions $\tilde{\alpha}$ with the true proportions of the target α^*.

The upper bound of $\|\hat{\alpha} - \tilde{\alpha}\|$ depends on the value we want to estimate $\tilde{\alpha}$, as it appears in $\|w\|_1$ and in $Tr(M\Sigma_{\tilde{\alpha}}M^\top)$. The dependency in $\|w\|_1$ can be removed as we did in Equation (3.3) where we replaced $\|w\|_1$ with the worst case scenario where $\tilde{\alpha}_i = 1$ for the smallest $\tilde{\beta}_i$ and 0 for the others. Put simply, the worst case is when the class for which we have the least information in the source distribution (i.e. the smallest $\tilde{\beta}_i$) becomes the only class in the target. In this case, $\frac{1}{n}\|w\|_1 = \min_i \frac{1}{n_i}$. In situations where one of the classes is rare on the source domain, the term $\min_i \frac{1}{n_i}$ explodes, which is not the case with the formulation in Equation (4.5), so this dependency in $\|w\|$ is actually desirable. On the other hand, note that the best case scenario for $\|w\|_1$ is obtained when either $\tilde{\alpha}$ equals $\tilde{\beta}$ or when $\beta_i = \frac{1}{c}$. In both cases : $\|w\|_1 = c$.

However, the dependence of $\tilde{\alpha}$ on $Tr(M\Sigma_{\tilde{\alpha}}M^\top)$ can not be removed unless we make the unlikely assumption that all classes have the same covariance matrix Σ. In this case, $\Sigma_{\tilde{\alpha}} = \Sigma$. As explained in Figure 4.2, this $\tilde{\alpha}$ dependence in the covariance matrix is desirable.

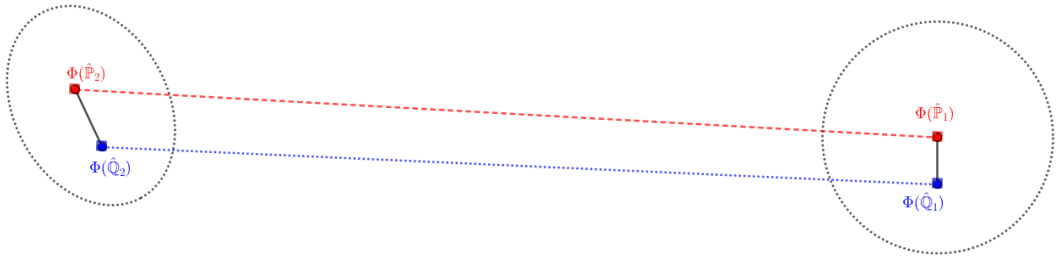


Figure 4.2: Analogous to Figure 4.1, but this time the two classes do not share the same covariance matrix. The metric to use to project onto the simplex (here the red dotted line) depends on the position of $\Phi(\hat{\mathbb{Q}})$. If we are close to the first class, we want to project using $M = \mathrm{Id}$, whereas if we are close to the second class, we want to project using the covariance matrix of the second class.

The bound is the sum of two terms. The first one (4.4) converges to zero at rate $\mathcal{O}(1/n)$. The dominant term is (4.5) as it converges at rate $\mathcal{O}(1/\sqrt{n})$, but this time neither C nor $\|M\|_{\mathrm{op}}$ appear and are replaced by $\sqrt{Tr(M\Sigma_{\tilde{\alpha}}M^\top)}$. For a sufficient number of points n, the first term will be close to zero and only the second term will remain. We can ask ourselves if we have gained anything by replacing $\|M\|_{\mathrm{op}}C$ with $\sqrt{\mathrm{Tr}(M\Sigma_{\tilde{\alpha}}M^\top)}$.

We have :

$$\begin{aligned}\mathrm{Tr}(M\Sigma_{\tilde{\alpha}}M^\top) &= \mathrm{Tr}(\Sigma_{\tilde{\alpha}}M^\top M)\\ &\leq \mathrm{Tr}(\Sigma_{\tilde{\alpha}})\ \left\|M^\top M\right\|_{\mathrm{op}}\\ &\leq \mathbb{E}\Big[\|X-\mathbb{E}[X]\|^2\Big]\ \|M\|_{\mathrm{op}}^2\\ &\leq C^2\|M\|_{\mathrm{op}}^2\end{aligned}$$

This is not a proof that the Bernstein-based bound (4.5) is sharper than the Hoeffding one (3.2), because the constants in front of these terms are not the same (in fact the constant in (3.2) is ~ 0.6 times smaller than that of (4.5) for $\delta = 0.05$), but due to the nature of the bound (the trace appears and not the operator norm) there is something to be gained by using a M other than the identity.

4.4 Optimal choice of M

Theorem 4.1 based on Hoeffding could not be used to derive a criterion for the optimal choice of M. The Bernstein based theorem, on the other hand, can be used for M.

In this section we assume that we use a given finite dimensional embedding Φ, i.e. $\mathcal{F} = \mathbb{R}^D$. This is the case, for instance, when we use MMD with Random Fourier Features, or when we embed the data using a classifier. In this framework, the operator M is now a matrix, which we assume is square ($M \in \mathbb{R}^{D\times D}$), but could also be in $\mathbb{R}^{D\times K}$, since as we explained before what matters is $M^\top M \in \mathbb{R}^{D\times D}$.

There are two places where the dependence of M appears in Theorem 4.2. In Equation (4.4) we have the same criterion as in (4.3):

$$\frac{\|M\|_{\mathrm{op}}}{\sqrt{\Delta_{\mathsf{min}}(\hat{\boldsymbol{G}}^M)}},$$

but we can ignore this term because it is in front of something that converges to zero at rate $\mathcal{O}(1/n)$. The term we want to minimise is the one in Equation (4.5) :

$$\sqrt{\frac{\mathrm{Tr}(M\Sigma_{\tilde{\alpha}}M^\top)}{\Delta_{\mathsf{min}}(\hat{\boldsymbol{G}}^M)}}. \tag{4.6}$$

4.4.1 Connection with Fisher Discriminant Analysis

Let us explore the connection between Fisher Discriminant Analysis (FDA), introduced by Fisher [45] and the Criterion (4.6). First, let us recall the principle of FDA (see for instance Ghojogh et al. [54] for a detailed tutorial on FDA).

Suppose we have access to a data set $\{(z_i, y_i)\}_{i=1}^n \in \mathbb{R}^d \times \mathcal{Y}$, the goal of FDA is to find a lower dimensional subspace that separates the classes as much as possible while minimising the spread of each class. In other words, if we take $U = [u_1, \cdots, u_c]$ as a basis of a subspace of $\mathbb{R}^D$, we search for the U that maximises the variance *between classes* when projected onto $\mathrm{Span}(U)$, while minimising the variance *within classes* when projected onto $\mathrm{Span}(U)$. This results in the following minimisation:

$$\min_U \ \frac{\mathrm{Tr}(U^T S_W U)}{\mathrm{Tr}(U^T S_B U)}, \tag{4.7}$$

with

$$S_W = \frac{1}{n} \sum_{j=1}^{c} \sum_{i=1}^{n_j} \mathbf{1}\{y_j = i\}(z_i - \mu_j)(z_i - \mu_j)^T; \tag{4.8}$$

$$S_B = \sum_{j=1}^{c} \tilde{\alpha}_j (\mu_j - \mu)(\mu_j - \mu)^T, \tag{4.9}$$

where μ_j is the mean of class i and μ the overall mean. With our notation, $z_i = \Phi(x_i)$, $\mu_j = \Phi(\hat{\mathbb{P}}_j)$ and $\mu \coloneqq \mu_{\tilde{\alpha}} = \sum_{j=1}^c \tilde{\alpha} \mu_j = \Phi(\hat{\mathbb{P}})$.

The criterion (4.6) is of the same kind. As we will see, the term $\mathrm{Tr}(M \Sigma_{\tilde{\alpha}} M^\top)$ defines a variance *within classes*, while the term $\Delta_{\mathsf{min}}(\hat{\boldsymbol{G}}^M)$ can be understood as a *variance between classes*. However, the problems are seen from different angles. In our case we are not looking for a $\mathrm{Span}(U)$ to project onto, but for a metric defined by $M^\top M$. This can be seen from the fact that in one case we minimise over $U \in \mathbb{R}^{D \times c}$ and in the other over $M \in \mathbb{R}^{D \times D}$. However, if we note $A \coloneqq UU^\top$ and $B \coloneqq M^\top M$, the two criteria can be rewritten

$$\min_A \ \frac{\mathrm{Tr}(A S_W)}{\mathrm{Tr}(A S_B)} \qquad \text{and} \qquad \min_B \ \frac{\mathrm{Tr}(B \Sigma_{\tilde{\alpha}})}{\Delta_{\mathsf{min}}(\hat{V}^\top B \hat{V})},$$

where we recall that $\hat{V}$ was defined in Definition 4.3 as the source embeddings matrix. The two minimisations can be performed on the same space of A and B, but to go back from A to U and from B to M we have to make different assumptions for A and B. For A we have to assume that its rank is equal to the number of classes c, while for B we have to assume that B is symmetric semidefinite positive. This is the first difference between the two frameworks.

For the *within-classes* variance S_W, we have :

$$\begin{aligned} S_W &= \frac{1}{n}\sum_{j=1}^{c}\sum_{i=1}^{n_j}\mathbf{1}\{y_j = i\}(z_i - \mu_j)(z_i - \mu_j)^T \\ &= \sum_{j=1}^{c}\frac{n_j}{n}\frac{1}{n_j}\sum_{i=1}^{n_j}\mathbf{1}\{y_j = i\}(z_i - \mu_j)(z_i - \mu_j)^T \\ &= \sum_{j=1}^{c}\tilde{\alpha}_j\hat{\Sigma}_i, \end{aligned}$$

where $\hat{\Sigma}_i$ is the empirical covariance matrix of class j. The two numerators have the same nature, they both represent the variance *within classes* of the sources. However, in FDA this variance is purely empirical, whereas in our criterion the numerator is a mixture of an empirical part $\tilde{\alpha}$ and a theoretical part Σ_i.

The change that occurs in the denominators is more profound. We said in Definition 3.2 that for a matrix N, $\Delta_{\mathsf{min}}(N)$ can be defined as the second smallest eigenvalue of the centred Gram matrix. If we note that $\lambda^{(-2)}$ is the second smallest eigenvalue, then according to the proof in Section 3.5.4, $\Delta_{\mathsf{min}}(N) = \lambda^{(-2)}(\boldsymbol{P}N\boldsymbol{P})$, where $\boldsymbol{P} \coloneqq I_d - \frac{1}{c}\mathbf{1}$ is a projection matrix. When N is a Gram matrix, $\boldsymbol{P}N\boldsymbol{P}$ is the centred Gram matrix. With all this in mind, let us upper bound the second smallest eigenvalue with the trace. Since the smallest eigenvalue of $\boldsymbol{P}\hat{V}^\top B\hat{V}\boldsymbol{P}$ is zero, we have :

$$\Delta_{\mathsf{min}}(\hat{V}^\top B\hat{V}) = \lambda^{(-2)}(\boldsymbol{P}\hat{V}^\top B\hat{V}\boldsymbol{P}) \leq (c-1)^{-1}\operatorname{Tr}(\boldsymbol{P}\hat{V}^\top B\hat{V}\boldsymbol{P}) = (c-1)^{-1}\operatorname{Tr}(B\hat{V}\boldsymbol{P}\hat{V}^\top).$$

We have :

$$\begin{aligned} \hat{V}\boldsymbol{P}\hat{V}^\top &= \hat{V}\hat{V}^\top - \frac{1}{c}\hat{V}\mathbf{1}\hat{V}^\top \\ &= \hat{V}\hat{V}^\top - \frac{1}{c^2}\hat{V}\mathbf{1}(\hat{V}\mathbf{1})^\top \\ &= \hat{V}\hat{V}^\top - c\mu\mu^\top \\ &= c \times \left(\sum_{i=1}\frac{1}{c}\mu_i\mu_i^\top - \mu\mu^\top\right), \end{aligned}$$

and therefore :

$$\begin{aligned} \Delta_{\mathsf{min}}(\hat{V}^\top B\hat{V}) &\leq \frac{c}{c-1}\operatorname{Tr}\left(B\sum_{i=1}\frac{1}{c}\mu_i\mu_i^\top - \mu\mu^\top\right) \\ &\overset{(\dagger)}{=} \frac{c}{c-1}\operatorname{Tr}(BS_B). \end{aligned}$$

where † is only true if $\tilde{\alpha}_i = 1/c$.

This highlights two differences between the FDA criterion (4.7) and ours (4.6). We replace the second smallest eigenvalue by the trace and our criterion does not depend the true value of $\tilde{\alpha}$. Note that in their case it is not a problem because the true labels are known, FDA is not a method to do classification but to do dimension reduction.

4.4.2 Optimal M

The intuition we have about the optimal M is that it should act on the subspace $\mathrm{Span}(\hat{V})$ to maximise $\Delta_{\mathsf{min}}(\hat{\boldsymbol{G}}^M)$ by minimising the variance on that subspace. On the orthogonal complement of $\mathrm{Span}(\hat{V})$, however, it is better to "kill" the action of M to minimise the variance, i.e. the kernel of M should contain the complement of $\mathrm{Span}(\hat{V})$.

To define the optimal M, we must first define the pseudo-inverse of a matrix.

Definition 4.4 (Moore–Penrose inverse)**.** Let $A \in \mathbb{R}^{D\times c}$ be a matrix. The pseudo-inverse of A, denoted by A^+, is the only matrix in $\mathbb{R}^{c\times D}$ that satisfies the four equations:

1. $AA^+A = A$
2. $A^+AA^+ = A^+$
3. $(AA^+)^\top = AA^+$
4. $(A^+A)^\top = A^+A$

Proposition 4.1 (Moore–Penrose inverse)**.** *The pseudo-inverse of a matrix A satisfies these properties:*

1. $A^+A = \Pi_{\mathrm{range}(A^\top)}$ *and* $AA^+ = \Pi_{\mathrm{range}(A)}$.
2. $A^+ = (A^\top A)^+A^\top = A^\top(AA^\top)^+$.
3. *When the rank of A is equal to its number of columns* $A^+ = (A^\top A)^{-1}A^\top$.
4. *If $A = PDQ$ is the SVD decomposition of A, we have $A^+ = PD^+Q$, with $D^+_{ii} = D^{-1}_{ii}$ if $D_{ii} \neq 0$.*

Where Π_S denotes the orthogonal projector on a linear subspace S.

We can now characterise the class of optimal M. Two remarks beforehand: first, as pointed out earlier, we are not looking for the optimal M, but the optimal $M^\top M \in \mathbb{R}^{D\times D}$, which we show to be unique up to a multiplicative factor. Second, we replace the optimal criterion (4.6) by a proxy (4.10) where we replace Δ_{min} in the denominator by λ_{min}.

Theorem 4.3. *Let us write $W = \Sigma_{\tilde{\alpha}}^{-1/2}\hat{V}$ and PDQ the SVD decomposition of W, with $P \in O(D)$, $Q \in O(c)$ and $D \in \mathbb{R}^{D\times c}$ a rectangular diagonal matrix with positive real numbers on the diagonal, i.e. $D = [D_0, 0]^T$, with $D_0 = \mathrm{Diag}(\sigma_1, \cdots, \sigma_c) \in \mathbb{R}^{c\times c}$.*

For any given feature map Φ that satisfies the conditions $\mathcal{A}_1$ and $\mathcal{A}_2$, the matrices M that minimise the criterion:

$$\frac{\mathrm{Tr}(M\Sigma_{\tilde{\alpha}}M^\top)}{\lambda_{min}(\hat{\boldsymbol{G}}^M)}, \tag{4.10}$$

satisfy up to a multiplicative factor :

$$M^\top M = \Sigma_{\tilde{\alpha}}^{-1/2}\Big(WW^\top\Big)^+\Sigma_{\tilde{\alpha}}^{-1/2}. \tag{$\mathcal{M}$}$$

For these matrices, Criterion (4.10) is then equal to

$$\mathrm{Tr}\Big(\big(WW^\top\big)^+\Big) = \sum_{i=1}^{c}\sigma_i^{-2}. \tag{4.11}$$

Proof can be found in Section 4.6.3.

It is important to note that the formulation $(\mathcal{M})$ depends on $\hat{V}$, while Theorem 4.2 suppose an M that is independent of the data. Therefore, paradoxically, the theorem does not hold for the optimal M. They is still work to be done to fix this issue.

One approach is to apply Theorem 4.2 to $M_*^\top M_* := \Sigma_{\tilde{\alpha}}^{-1/2}\Big(\Sigma_{\tilde{\alpha}}^{-1/2}VV^\top\Sigma_{\tilde{\alpha}}^{-1/2}\Big)^+\Sigma_{\tilde{\alpha}}^{-1/2}$, i.e. for the population version of the optimal M. We can then study $\Delta_{\mathsf{min}}(\hat{\boldsymbol{G}}^M)/\Delta_{\mathsf{min}}(\hat{\boldsymbol{G}}^{M_*})$ asymptotically to obtain a bound that applies to the optimal M with the same rate of convergence.

This work is beyond the scope of this manuscript and is left for future research.

Remark. We have simplified by replacing $\Delta_{\min}(\hat{\boldsymbol{G}}^M)$ with $\lambda_{\min}(\hat{\boldsymbol{G}}^M)$ in the formulation of the criterion. While it is not clear whether $(\mathcal{M})$ is the minimum of the original criterion (4.6), we can note that, with this M: $\Delta_{\min}(\hat{\boldsymbol{G}}^M) = \lambda_{\min}(\hat{\boldsymbol{G}}^M) = 1$ and therefore the value of the criterion is equal to (4.11) in both cases.

To grasp the nature of $(\mathcal{M})$ let us look at $M^\top M$ as a bilinear form on $\mathbb{R}^D$. Let us write $E = (e_{c+1}, \cdots, e_D)$ an orthonormal basis of the orthogonal of $\mathrm{Span}(V)$ with respect to Σ^{-1}:

$$\begin{aligned} e_i\Sigma^{-1}e_j &= 0 \quad \text{if } i \neq j \\ e_i\Sigma^{-1}e_i &= 1. \end{aligned}$$

For any $\alpha, \beta \in \mathbb{R}^c$ and $\gamma \in \mathbb{R}^{D-c}$, if we note $x = \hat{V}\alpha \in \mathrm{Span}(\hat{V})$ and $y = \hat{V}\beta + E\gamma \in \mathbb{R}^D$ then

$$x^\top M^\top My = \alpha^\top\beta.$$

Therefore, on $\mathrm{Span}(\hat{V})$, $M^\top M$ acts as if the source embeddings were an orthonormal basis of the subspace and elsewhere it acts as a projector on $\hat{V}$ with respect to Σ^{-1}.

Intuitively, the reason why $M^\top M$ acts as a hybrid between a Euclidean distance on $\hat{V}$ and a Mahalanobis distance on $\hat{V}^\perp$ is that on the one hand we use the L_2 metric to compare our approximation ($\|\tilde{\alpha} - \hat{\alpha}\|_2$), but on the other hand we use a Mahalanobis metric to compare the embeddings.

Solution of $(\mathcal{P}_\mathcal{M})$ with the optimal M. With all that in mind, let us write the embedding of the target as $\Phi(\hat{\mathbb{Q}}) = \hat{V}\beta + E\gamma$. Then the solution of $(\mathcal{P}_\mathcal{M})$ when we take an optimal M, can be rewritten as:

$$\begin{aligned}\hat{\alpha} &= \operatorname*{arg\,min}_{\alpha \in \Delta^c} \left\| M\left(\sum_{i=1}^{c} \alpha_i \Phi(\hat{\mathbb{P}}_i) - \Phi(\hat{\mathbb{Q}})\right)\right\|_{\mathcal{F}}^2 \\ &= \operatorname*{arg\,min}_{\alpha \in \Delta^c} \alpha^T\alpha - 2\alpha^T\beta. + \beta^\top\beta \\ &= \operatorname*{arg\,min}_{\alpha \in \Delta^c} \|\alpha - \beta\|_2^2\end{aligned}$$

In particular, if $\beta \in \Delta^c$ then $\hat{\alpha} = \beta$.

Therefore we can summarize the effect of this optimal M as such:

The embedding of the target $\Phi(\hat{\mathbb{Q}})$ is first projected onto $\hat{V}$ with respect to the covariance operator as desired in Figure 4.1. However, unlike the figure, the covariance is ignored on the space $\hat{V}$ and only the geometric nature of the convex hull is taken into account, because if we assume that the target embedding belongs to the Span of the source embeddings, then the solution is given by

$$\hat{\alpha} = \Pi_{\Delta^c}\left(\hat{V}^+\Phi(\hat{\mathbb{Q}})\right),$$

where Π_{Δ^c} is the projection onto the simplex.

4.4.3 Taking the pooled covariance matrix

As desired, the covariance matrix appears in the formulation of M. However, the covariance is not taken into account in the Span of the source embeddings, as we wanted in Figure 4.1. One way to have this would be to take the pooled covariance matrix: $\Sigma_{\tilde{\alpha}}$ directly.

Solution of $(\mathcal{P}_\mathcal{M})$ with the pooled covariance matrix. The solution of $(\mathcal{P}_\mathcal{M})$ when we take $M = \Sigma_{\tilde{\alpha}}^{-1/2}$ can be rewritten:

$$\begin{aligned}\hat{\alpha} &= \underset{\alpha\in\Delta^c}{\arg\min} \left\| \Sigma_{\tilde{\alpha}}^{-1/2}\left(\sum_{i=1}^{c}\alpha_i\Phi(\hat{\mathbb{P}}_i) - \Phi(\hat{\mathbb{Q}})\right)\right\|_{\mathcal{F}}^2 \\ &= \underset{\alpha\in\Delta^c}{\arg\min}\ \alpha^T\left(\hat{V}^\top\Sigma_{\tilde{\alpha}}^{-1}\hat{V}\right)\alpha - 2\alpha^T\left(\hat{V}^\top\Sigma_{\tilde{\alpha}}^{-1}\hat{V}\right)\beta + \beta^\top\left(\hat{V}^\top\Sigma_{\tilde{\alpha}}^{-1}\hat{V}\right)\beta \\ &= \underset{\alpha\in\Delta^c}{\arg\min}\ \|\alpha-\beta\|_W^2,\end{aligned}$$

with $W = \Sigma_{\tilde{\alpha}}^{-1/2}\hat{V}$.

In particular, if $\beta \in \Delta^c$ then $\hat{\alpha} = \beta$.

The effect of the pooled covariance matrix can be summarize as such:

As for the optimal M, the embedding of the target $\Phi(\hat{\mathbb{Q}})$ is first projected onto V with respect to the covariance operator, and if the target embedding belongs to the convex hull of the source embeddings, i.e. $\beta \in \Delta^c$, the two matrices give the same results. If not, the pooled covariance matrix and the optimal M differ in the way the vector β is projected onto the simplex. For the optimal M we projected with respect to the ambient matrix in $\mathbb{R}^c$, while for $\Sigma_{\tilde{\alpha}}$ we projected with respect to the matrix W.

4.4.4 Approximation of M and choice of bandwidth

For a given feature map Φ, we have a closed form solution of M that minimises the Criterion (4.10) given by Equation $(\mathcal{M})$. This closed form solution depends on $\hat{V}$ the empirical source covariance matrix that can be computed for a given dataset but unfortunately also on the covariance matrices of the source embeddings $(\Sigma_i)_{i=1}^c$ and on the true proportions we want to estimate $\tilde{\alpha}$, two quantities that are not available to us.

We can estimate the covariance matrices of the source embeddings with the usual covariance estimator:

$$\hat{\Sigma}_i = \frac{1}{n_i - 1}\sum_{j=i}^{n}\mathbf{1}\{y_j = i\}(\Phi(x_j) - \Phi(\mathbb{P}_i))(\Phi(x_j) - \Phi(\mathbb{P}_i))^\top.$$

However, for any true proportions $\tilde{\alpha}$ the matrix $\hat{\Sigma}_{\tilde{\alpha}} := \sum_{i=1}^c \tilde{\alpha}_i\hat{\Sigma}_i$ will most likely be singular.

We propose to define $\mathcal{I}(\hat{\Sigma}_i;\gamma,\alpha) \approx \Sigma_\alpha^{-1/2}$, a regularised version of the square root of the inverse covariance matrix, with regularization parameter γ and proportion vector α. Two classical regularisation choices are the Tikhonov regularisation :

$$\mathcal{I}(\hat{\Sigma}_i;\gamma,\alpha) = \left(\hat{\Sigma}_\alpha + \gamma I\right)^{-1/2},$$

and the spectral truncation of rank d :

$$\mathcal{I}(\hat{\Sigma}_i;1/d,\alpha) = \sum_{p=1}^{d}\lambda_p^{-1/2}(e_p e_p^\top),$$

where (λ_p, e_p) denote the sequence of eigenvalues and eigenvectors of the matrix $\hat{\Sigma}_\alpha$.

In the following, we will use the Tikhonov regularisation and therefore we have a new hyperparameter to deal with : the regularisation parameter γ.

For the proportions $\tilde{\alpha}$, we explained in Figure 4.2 that it is desirable to have this dependence on the true proportions in the bound. In practice, however, we need to have an estimate of the proportions before computing M. We propose a procedure where the proportions are first estimated using DFM, and then the output is used to compute an estimate of the matrix M used for M-DFM.

This *two-step* procedure can be enhanced even more by proposing a *multi-step* procedure, where we perform k steps of M-DFM where the matrix M is computed using the proportions obtained during the previous step. We stop the algorithm after k steps of after convergence of the iterations.

In practice, during the experiments we have not seen any difference between the *two-step* and *multi-step* procedures.

Remark. The proof of Theorem 4.2, and in particular the Bernstein's theorem presented in Appendix A.3 on which the bound is based, suppose that the matrix M used is deterministic. Therefore the theorem is not true if the data are used to estimate both M and the embeddings $\Phi(\mathbb{P}_i)$. We already mention the dependence in $\hat{V}$ and said that in practice we could use $\hat{V}$ directly. For the dependence in Σ_i, we can split the data in two, used a part for the embeddings and the other part for M.

Choice of bandwidth σ If we rely on the Gaussian kernel we need a criterion to choose the bandwidth σ. The only term in Equation (4.5) that depends on the bandwidth is (4.10). So the criterion to choose the bandwidth is the same as the one use to choose M. But as we said above, this criterion depends on $\Sigma_{\tilde{\alpha}}^{-1/2}$ so we have to rely on an estimation which also depends on a hyperparameter γ.

Similarly to DFM, where we used Δ_{min} (see Section 3.3.2) as a criterion that we maximised by an exhaustive grid search on $\mathbb{R}^+$, for M-DFM we propose to do an exhaustive grid search on both γ and σ to maximise (4.11).

However, to reduce the computation time, we propose an alternative where we first estimate the σ that maximises Δ_{min} and then use (4.11) to estimate the best γ parameter. This procedure replaces one grid search in two dimensions with two grid searches in one dimension, and avoids having to compute many covariance matrices.

4.5 Experiments

In this section we first show that using the covariances actually improves the results and was not just a theoretical work.

We pointed out that there are two possible choices of M: the optimal M and the pooled covariance matrix. Theoretically, the optimal M should give better results, but as we shall see, the two matrices give similar results. In a second step, we will try to understand why the results are similar.

4.5.1 Impact of M on Quantification

First, let us describe the experiments we performed to evaluate the quantification performance of M-DFM. We tested the same two settings as in Section 3.4.2. The first setting is completely artificial, the data is generated using a Gaussian mixture in $\mathbb{R}^d$. In the second setting, we use a flow cytometry dataset which is a subset of a larger dataset that we present in detail in Chapter 5.

The four methods tested are all part of the M-DFM framework and for each method we use $M^\top M = I$, $M^\top M = \mathcal{M}$ and $M^\top M = \Sigma_{\tilde{\alpha}}^{-1}$. The first case is a simple DFM procedure, which we compare to either M-DFM with the optimal criterion or M-DFM with the covariance matrix directly.

Let us first look at the four embeddings we are considering in this section:

Names	Embedding functions Φ
BBSE	$\hat{f}(x)$
RFFM	$\sqrt{\frac{2}{D}}\left[\cos(\omega_i^T x),\ \sin(\omega_i^T x)\right]_{i=1}^{D/2}$
FourierClassifer	$\sqrt{\frac{2}{D}}\left[\cos(\omega_i^T \hat{f}(x)),\ \sin(\omega_i^T \hat{f}(x))\right]_{i=1}^{D/2}$
MeanDFM	x

Table 4.1: Overview of the four embeddings we consider in this section. $\hat{f}$ is a soft classifier trained on the source and ω_i is sampled from the *spectral distribution* of a kernel, in our case the Gaussian one.

The first two, BBSE and RFFM, were introduced in Chapter 3. FourierClassifier is similar to Moreo et al.'s KDEy [85], which was presented in Section 2.2.2. The embedding is obtained by computing the RFF embedding not directly on the data, but on the output of a classifier $\hat{f}$. The last method, MeanDFM, uses the identity function. In this case, the embedding of a distribution is simply its mean.

Gaussian Mixture

We generate the data with the same Gaussian mixture as in Section 3.4.2. The parameters are also the same : the radius is $R = 2$, the range of the eigenvalues is $d = 1$, the variance of the Dirichlet distribution is $\tau = 50$, the number of classes taken is $c = 2$ and $c = 5$ and

the dimensions are $D = 2$, $D = 5$ and $D = 10$. The only difference from the experiments we conducted in the previous chapter is that we do not contaminate the data with noise.

All the results are for the 6 different settings are mixed together in Table 4.2.

Flow Cytometry

We use the flow cytometry data from Metafora. This dataset is a subset of a larger dataset that we present in detail in Chapter 5. For each patient we have access to four cell types: *T cells, B cells, NK cells and plasmocytes*, but we discard the *T cells* because they are overrepresented in the dataset, see Figure 3.3. The results can be found in Table 4.3.

Quantifier	Mahalanobis		Classique
	Optimal	$\Sigma_{\tilde{\alpha}}^{-1/2}$	Identity
BBSE	**0.355 ; 2.5**	**0.355 ; 2.5**	0.357 ; 1.0
RFFM	**0.237 ; 1.5**	**0.237 ; 1.5**	0.266 ; 3.0
FourierClassifier	**0.323 ; 1.5**	**0.323 ; 1.5**	0.348 ; 3.0
MeanDFM	**0.696 ; 2.5**	**0.697 ; 2.5**	**0.625 ; 1.0**

Table 4.2: **GaussianMixture: Comparison of BBSE, RFFM, FourierClassifier and MeanDFM**. The value before the semicolon is the geometric mean of the absolute error (multiplied by 100 for clarity) over 50 repetitions. Value after the semicolon is the median rank of a method relative to the other method in the same setting (same embedding but different matrices M). A bold value in a group is not significantly different from the best-performing method in the group (also in bold), as measured by a paired Wilcoxon test at $p < 0.01$.

The use of the covariance, either by taking the optimal M or the covariance directly, significantly enhance the results for RFFM and FourierClassifier. For BBSE and MeanDFM, the results are either improved of remain the same depending on the data. Note that in Table 4.2, for BBSE the results are statistically better (according to the Wilcoxon test with p-value less than 0.01) when we use the covariance, the geometric means are also smaller, but the median rank of the classical method is better than the median rank of the other methods. This could imply that the classical method is more consistent across different conditions, whereas the various covariance-based methods might be more effective on average but less robust in certain cases.

Robustness to open set label shift

Even if it is not the subject of this chapter, we point out that the results on robustness under open set label shift presented in the previous chapter (see Theorem 3.3 and corollary 3.2), are

Quantifier	Mahalanobis		Vanilla
	Optimal	$\Sigma_{\tilde{\alpha}}^{-1/2}$	Identity
BBSE	**0.166 ; 2.0**	**0.166 ; 2.5**	**0.165 ; 1.5**
RFFM	**0.151 ; 1.5**	**0.151 ; 1.5**	0.205 ; 3.0
FourierClassifier	**0.138 ; 1.5**	**0.138 ; 1.5**	0.156 ; 3.0
MeanDFM	**0.329 ; 1.5**	**0.329 ; 1.75**	0.364 ; 3.0

Table 4.3: **Flow Cytometry: Comparison of BBSE, RFFM, FourierClassifier and MeanDFM**. The value before the semicolon is the geometric mean of the absolute error (multiplied by 100 for clarity) over 50 repetitions. Value after the semicolon is the median rank of a method relative to the other method in the same setting (same embedding but different matrices M). A bold value in a group is not significantly different from the best-performing method in the group (also in bold), as measured by a paired Wilcoxon test at $p < 0.01$.

still true for the embeddings $\left(M\Phi(\hat{\mathbb{P}}_i)\right)_{i=1}^{c}$ but with different constants, in the same fashion as in Section 4.3.1.

We said that robustness to contamination is ensured as long as the target embedding is orthogonal to the source embeddings. For M-DFM, the conclusion is the same, except that the target must be orthogonal w.r.t. the metric M.

In Table 4.4 we show the results of the same experiment as in the previous chapter for the *far setting*, i.e. the noise is far from the source distributions. In Table 4.5 we show the results for the flow cytometry dataset, where the *T cells* are used as a contamination.

As we can see, MahalanobisRFFM is robust to contamination, but the results are slightly worse than RFFM for the real data and equivalent for the synthetic data.

Percentage of noise ϵ	**Quantifier**	Number of classes = 5		
		dim = 2	dim = 5	dim = 10
0.0	BBSE	1.45 ; 4.0	0.45 ; 3.0	0.37 ; 3.0
0.0	RFFM	0.65 ; 2.0	0.30 ; 2.0	0.32 ; 2.0
0.0	FourierClassifier	1.28 ; 4.0	0.45 ; 4.0	0.37 ; 4.0
0.0	MahalanobisRFFM	**0.58 ; 2.0**	**0.26 ; 1.0**	**0.27 ; 1.0**
0.0	MahalanobisFourierClassifier	1.17 ; 3.0	0.43 ; 3.0	0.36 ; 3.0
0.2	BBSE	9.90 ; 5.0	5.93 ; 5.0	5.36 ; 5.0
0.2	RFFM	**1.17 ; 1.0**	**0.69 ; 1.0**	**0.70 ; 1.0**
0.2	FourierClassifier	5.82 ; 4.0	5.32 ; 4.0	5.07 ; 4.0
0.2	MahalanobisRFFM	**1.21 ; 2.0**	**0.71 ; 2.0**	0.77 ; 2.0
0.2	MahalanobisFourierClassifier	5.83 ; 3.0	4.98 ; 3.0	4.82 ; 3.0
0.5	BBSE	17.42 ; 5.0	14.77 ; 5.0	13.53 ; 5.0
0.5	RFFM	**2.19 ; 1.0**	**1.66 ; 1.5**	**1.81 ; 1.0**
0.5	FourierClassifier	11.24 ; 4.0	13.24 ; 4.0	12.76 ; 4.0
0.5	MahalanobisRFFM	**2.38 ; 2.0**	**1.67 ; 1.5**	**1.85 ; 2.0**
0.5	MahalanobisFourierClassifier	10.87 ; 3.0	11.76 ; 3.0	11.45 ; 3.0
0.7	BBSE	19.78 ; 5.0	18.41 ; 5.0	17.93 ; 5.0
0.7	RFFM	**2.54 ; 1.0**	**2.13 ; 1.0**	**2.62 ; 1.0**
0.7	FourierClassifier	13.29 ; 4.0	16.90 ; 4.0	17.13 ; 4.0
0.7	MahalanobisRFFM	**2.60 ; 2.0**	2.25 ; 2.0	**2.66 ; 2.0**
0.7	MahalanobisFourierClassifier	12.95 ; 3.0	15.42 ; 3.0	15.48 ; 3.0

Table 4.4: **Gaussian Mixture: Comparison of BBSE, RFFM, FourierClassifier with and without mahalanobis when $\rho = 10$ (far setting)**. The value before the semicolon is the geometric mean of the absolute error (multiplied by 100 for clarity) over 50 repetitions. Value after the semicolon is the median rank of a method relative to the other method in the same setting (same dimension, number of classes and percentage of noise). A bold value in a group is not significantly different from the best-performing method in the group (also in bold), as measured by a paired Wilcoxon test at $p < 0.01$.

Percentage of noise ϵ	Quantifier	Amount of shift	
		$\rho = 10$	$\rho = 500$
0.0	BBSE	0.17 ; 3.0	0.17 ; 3.0
0.0	RFFM	0.25 ; 5.0	0.23 ; 5.0
0.0	MahalanobisRFFM	**0.16 ; 3.0**	**0.16 ; 3.0**
0.0	FourierClassifier	0.16 ; 3.0	**0.16 ; 3.0**
0.0	MahalanobisFourierClassifier	0.14 ; 2.0	0.14 ; 2.0
0.1	BBSE	3.57 ; 5.0	3.57 ; 5.0
0.1	RFFM	**1.55 ; 1.0**	**1.54 ; 1.0**
0.1	MahalanobisRFFM	1.61 ; 2.0	1.61 ; 2.0
0.1	FourierClassifier	3.16 ; 3.0	3.16 ; 3.0
0.1	MahalanobisFourierClassifier	3.24 ; 4.0	3.25 ; 4.0
0.2	BBSE	7.08 ; 5.0	7.08 ; 5.0
0.2	RFFM	**2.97 ; 1.0**	**2.97 ; 1.0**
0.2	MahalanobisRFFM	3.15 ; 2.0	3.14 ; 2.0
0.2	FourierClassifier	6.25 ; 3.0	6.25 ; 3.0
0.2	MahalanobisFourierClassifier	6.42 ; 4.0	6.42 ; 4.0
0.3	BBSE	10.59 ; 5.0	10.60 ; 5.0
0.3	RFFM	**4.40 ; 1.0**	**4.42 ; 1.0**
0.3	MahalanobisRFFM	4.68 ; 2.0	4.68 ; 2.0
0.3	FourierClassifier	9.34 ; 3.0	9.35 ; 3.0
0.3	MahalanobisFourierClassifier	9.58 ; 4.0	9.59 ; 4.0

Table 4.5: **Flow Cytometry: Comparison of BBSE, RFFM and FourierClassifer with and without mahalanobis**. The value before the semicolon is the geometric mean of the absolute error (multiplied by 100 for clarity) over 50 repetitions. Value after the semicolon is the median rank of a method relative to the other method in the same setting (same dimension, number of classes and percentage of noise). A bold value in a group is not significantly different from the best-performing method in the group (also in bold), as measured by a paired Wilcoxon test at $p < 0.01$.

4.5.2 Criterion and effective dimension

We proposed a controlled experiments where the embedding Φ is the identity matrix and the source distribution is a mixture of 5 Gaussians in dimension D. Therefore, we directly have access to the true covariance matrix $\Sigma_{\tilde{\alpha}}$.

We suppose that the Gaussian have an *effective dimension*, i.e. only a subspace of dimension $D_{\text{eff}} < D$ contains variance information. To simulate this, we generate D eigenvectors, then for the first D_{eff} eigenvectors by sampling at random D vectors and applying a QR decomposition, a corresponding eigenvalue is sampled uniformly on $[1, 2]$. For each of the remaining $D - D_{\text{eff}}$ eigenvectors, an eigenvalue is sampled from a uniform distribution on $[0, 0.01]$ (we do not set these eigenvalues directly to 0 because we need the covariance matrix to be invertible). We do this for each class independently, but the eigenvectors associated with each eigenvalue are kept for each class, so the *effective dimension* of $\Sigma_{\tilde{\alpha}}$ is D_{eff}.

Figure 4.3 shows the eigenvalues of each class.

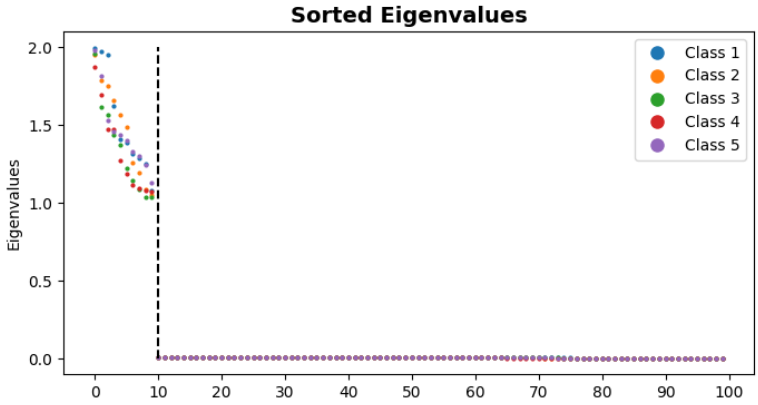


Figure 4.3: Sorted eigenvalues of each class. Here $D = 100$ and $D_{\text{eff}} = 10$.

In Figure 4.4, we plot the value of Criterion (4.6) against the effective dimension D_{eff}, for the different choices of M we used in the experiments: the identity matrix, the covariance and the optimal M. In Figure 4.5, we plot the estimation error (using the L_2 distance) against the effective dimension D_{eff} for each M. For each effective dimension, the experiments are repeated 20 times.

As we can see, when the effective dimension is small, the criterion for the optimal M and for the pooled covariance matrix are almost equal. But what we can also see, is that even in cases where the criterion for the pooled covariance matrix is clearly suboptimal the results are the same. More work must be carried out to understand why the two choices yield similar results. This experiment hints that the criterion we derive from the analysis is not correlated with the quality of the quantifier.

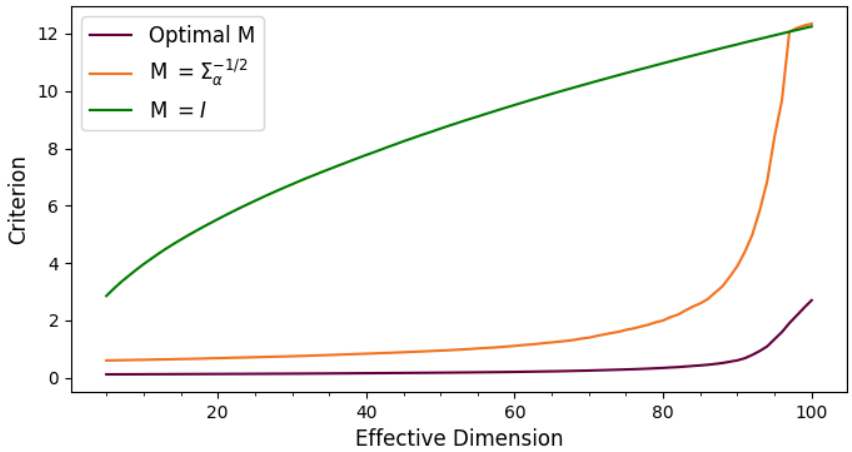


Figure 4.4: Value of criterion (4.6) for different effective dimensions D_{eff}, depending on the choice of matrix M.

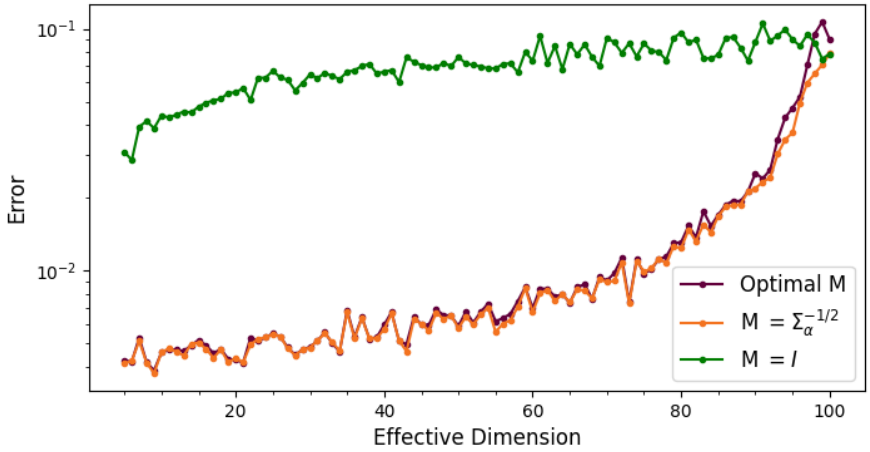


Figure 4.5: L_2 distance between the estimation $\hat{\alpha}$ and the true proportions α^* for different choice of M.

4.6 Proofs

4.6.1 Proof of Theorem 4.1.

Proof. Using Definition 4.3 of $\Delta_{\min}$:

$$\begin{aligned}\sqrt{\Delta_{\min}(\hat{\boldsymbol{G}})} &= \sqrt{\Delta_{\min}(\hat{V}^\top \hat{V})} \\ &= \min_{\substack{\|u\|_2=1 \\ \mathbf{1}^T u=0}} \sqrt{u^\top \hat{V}^\top \hat{V} u} \\ &= \min_{\substack{\|u\|_2=1 \\ \mathbf{1}^T u=0}} \left\|\hat{V}u\right\|\end{aligned}$$

Let us note u^* a minimiser.

Using the properties of the operator norm $\|\cdot\|_{\text{op}}$ we get for all operator M:

$$\begin{aligned}\sqrt{\Delta_{\min}(\hat{\boldsymbol{G}}^M)} &= \sqrt{\Delta_{\min}(\hat{V}^\top M^\top M \hat{V})} \\ &= \min_{\substack{\|u\|_2=1 \\ \mathbf{1}^T u=0}} \sqrt{u^\top \hat{V}^\top M^\top M \hat{V} u} \\ &= \min_{\substack{\|u\|_2=1 \\ \mathbf{1}^T u=0}} \left\|M\hat{V}u\right\| \\ &\leq \left\|M\hat{V}u^*\right\| \\ &\leq \|M\|_{\text{op}} \left\|\hat{V}u^*\right\| \\ &= \|M\|_{\text{op}} \sqrt{\Delta_{\min}(\hat{\boldsymbol{G}})}\end{aligned}$$

Using this inequality we can conclude that for all M:

$$\frac{\|M\|_{\text{op}}}{\sqrt{\Delta_{\min}(\hat{\boldsymbol{G}}^M)}} \geq \frac{1}{\sqrt{\Delta_{\min}(\hat{\boldsymbol{G}})}}.$$

□

4.6.2 Proof of Theorem 4.2.

Proof. Throughout the proof, we will use the notation α for $\tilde{\alpha}$ to simplify the reading. It holds :

$$\begin{aligned} D_\Phi^M\left(\sum_{i=1}^c \hat{\alpha}_i\hat{\mathbb{P}}_i, \sum_{i=1}^c \alpha_i\hat{\mathbb{P}}_i\right)^2 &= \left\| M\left(\sum_{i=1}^c \hat{\alpha}_i\Phi(\hat{\mathbb{P}}_i) - \sum_{i=1}^c \alpha_i\Phi(\hat{\mathbb{P}}_i)\right)\right\|^2 \\ &= \left\|\sum_{i=1}^c (\hat{\alpha}_i - \alpha_i) M\Phi(\hat{\mathbb{P}}_i)\right\|^2 \\ &= (\hat{\alpha}-\alpha)\ \hat{\boldsymbol{G}}^M\ (\hat{\alpha}-\alpha) \\ &\geq \left(\min_{\substack{\|u\|_2=1 \\ \mathbf{1}^T u=0}} u^T\hat{\boldsymbol{G}}^M u\right)\ \|\hat{\alpha}-\alpha\|^2 \\ &= \Delta_{\min}(\hat{\boldsymbol{G}}^M)\ \|\hat{\alpha}-\alpha\|^2, \end{aligned}$$

Thus in order to bound $\|\hat{\alpha}-\alpha\|$ we have to upper-bound: $D_\Phi^M(\sum_{i=1}^c \hat{\alpha}_i\hat{\mathbb{P}}_i, \sum_{i=1}^c \alpha_i\hat{\mathbb{P}}_i)$. By the triangle inequality, this is upper bounded by $D_\Phi^M(\sum_{i=1}^c \hat{\alpha}_i\hat{\mathbb{P}}_i, \hat{\mathbb{Q}}) + D_\Phi^M(\hat{\mathbb{Q}}, \sum_{i=1}^c \alpha_i\hat{\mathbb{P}}_i)$. By definition, $\hat{\alpha}$ is the minimiser of $D_\Phi^M(\sum_{i=1}^c \gamma_i\hat{\mathbb{P}}_i, \hat{\mathbb{Q}})$ for $\gamma \in \Delta^c$, hence we have $D_\Phi^M(\sum_{i=1}^c \hat{\alpha}_i\hat{\mathbb{P}}_i, \hat{\mathbb{Q}}) \leq D_\Phi^M(\sum_{i=1}^c \alpha_i\hat{\mathbb{P}}_i, \hat{\mathbb{Q}})$. Hence, we can upper bound the quantity by $2D_\Phi^M(\sum_{i=1}^c \alpha_i\hat{\mathbb{P}}_i, \hat{\mathbb{Q}})$.

Using the triangle inequality once again, we can upper bound $D_\Phi^M(\sum_{i=1}^c \alpha_i\hat{\mathbb{P}}_i, \hat{\mathbb{Q}})$ by

$$D_\Phi^M\left(\sum_{i=1}^c \alpha_i\hat{\mathbb{P}}_i, \sum_{i=1}^c \alpha_i\mathbb{P}_i\right) + D_\Phi^M\left(\sum_{i=1}^c \alpha_i\mathbb{P}_i, \hat{\mathbb{Q}}\right). \tag{4.12}$$

Using Theorem A.3 (Bernstein) and the union bound, it holds with probability greater than $1-\delta/2$:

$$\begin{aligned} D_\Phi^M\left(\sum_{i=1}^c \alpha_i\hat{\mathbb{P}}_i, \sum_{i=1}^c \alpha_i\mathbb{P}_i\right) &\leq \sum_{i=1}^c \alpha_i D_\Phi^M(\mathbb{P}_i, \hat{\mathbb{P}}_i) \\ &\leq \sum_{i=1}^c \alpha_i\left(\sigma_1(M)\frac{2}{3}\frac{C\log 4c/\delta}{n_i} + \sqrt{\frac{2\log 4c/\delta}{n_i}\operatorname{Tr}(M\Sigma_i M^\top)}\right) \\ &\leq \frac{2}{3}\log 4c/\delta\sigma_1(M)C\sum_{i=1}^c \frac{\alpha_i}{n_i} + \sqrt{2\log 4c/\delta}\sum_{i=1}^c \frac{\alpha_i}{\sqrt{n_i}}\sqrt{\operatorname{Tr}(M\Sigma_i M^\top)}, \end{aligned}$$

Since $n_i = n\tilde{\beta}_i$, it holds $\sum_{i=1}^c \frac{\alpha_i}{n_i} = \frac{1}{n}\sum_{i=1}^c \frac{\alpha_i}{\tilde{\beta}_i} = \frac{\|w\|_1}{n}$ with $w = \frac{\alpha}{\tilde{\beta}}$.

By Hölder's inequality:

$$\begin{aligned}\sum_{i=1}^{c} \frac{\alpha_i}{\sqrt{\tilde{\beta}_i}} \sqrt{Tr(M\Sigma_i M^\top)} &= \sum_{i=1}^{c} \sqrt{\frac{\alpha_i}{\tilde{\beta}_i}} \sqrt{\alpha_i Tr(M\Sigma_i M^\top)} \\ &\leq \sqrt{\|w\|_1} \sqrt{\sum_{i=1}^{c} \alpha_i Tr(M\Sigma_i M^\top)} \\ &\leq \sqrt{\|w\|_1} \sqrt{Tr(M\Sigma_\alpha M^\top)},\end{aligned}$$

so that :

$$\begin{aligned}D_\Phi^M \left(\sum_{i=1}^{c} \alpha_i \hat{\mathbb{P}}_i, \sum_{i=1}^{c} \alpha_i \mathbb{P}_i \right) \leq \frac{2}{3} \log 4c/\delta \sigma_1(M) C \frac{\|w\|_1}{n} \\ + \sqrt{2 \log 4c/\delta} \sqrt{Tr(M\Sigma_\alpha M^\top)} \sqrt{\frac{\|w\|_1}{n}}.\end{aligned}$$

We finally turn to bounding the term $D_\Phi^M \left(\sum_{i=1}^{c} \alpha_i \mathbb{P}_i, \hat{\mathbb{Q}} \right)$. It holds $\alpha_i = m_i/m$ where m_i is the number of target sample points of class i. We then get:

$$\begin{aligned}D_\Phi^M \left(\sum_{i=1}^{c} \alpha_i \mathbb{Q}_i, \hat{\mathbb{Q}} \right) &= \left\| M \left(\frac{1}{m} \sum_{i=1}^{c} m_i \Phi(\mathbb{Q}_i) - \frac{1}{m} \sum_{j=n+1}^{n+m} \Phi(x_j) \right) \right\| \\ &= \left\| M \left(\frac{1}{m} \sum_{j=n+1}^{n+m} \Phi(\mathbb{Q}_{y_j}) - \frac{1}{m} \sum_{j=n+1}^{n+m} \Phi(x_j) \right) \right\| \\ &= \left\| M \left(\frac{1}{m} \sum_{j=n+1}^{n+m} \big(\Phi(\mathbb{Q}_{y_j}) - \Phi(x_j) \big) \right) \right\|\end{aligned}$$

Now, notice that conditionally to the labels $(y_j)_{j=n+1}^{n+m}$, the target sample points x_j are independent, not identically distributed but with respective class conditional distribution $\mathbb{Q}_{y_j}$. We can therefore still appeal to Theorem A.3, and conclude that it holds with probability greater than $1 - \delta/2$:

$$\begin{aligned}D_\Phi^M \left(\sum_{i=1}^{c} \alpha_i \mathbb{Q}_i, \hat{\mathbb{Q}} \right) &\leq \sigma_1(M) \frac{2}{3} \frac{C}{m} \log 4/\delta + \sqrt{\frac{2 \log 4/\delta}{m} Tr(M\overline{\Sigma}M^\top)} \\ &\leq \sigma_1(M) \frac{2}{3} \frac{C}{m} \log 4c/\delta + \sqrt{\frac{2 \log 4c/\delta}{m} Tr(M\overline{\Sigma}M^\top)}\end{aligned}$$

Where $\overline{\Sigma} \coloneqq \frac{1}{m} \sum_{j=n+1}^{n+m} \Sigma_{\Phi(X_j)} = \sum_{i=1}^{c} \alpha_i \Sigma_{\Phi(\mathbb{Q}_i)} = \Sigma_\alpha$.

We conclude with a final union bound to bound the two terms of Equation (4.12). $D_\Phi^M(\sum_{i=1}^c \hat{\alpha}_i \hat{\mathbb{P}}_i, \sum_{i=1}^c \alpha_i \hat{\mathbb{P}}_i)$ was bounded by 2 times (4.12), which explain the constant R_1 and R_2 in Theorem 4.2. □

4.6.3 Proof of Theorem 4.3.

Let us first prove a lemma that we will use during the proof.

Lemma 4.1. *Let $D = \mathrm{Diag}(d_1, \cdots, d_c) \in \mathbb{R}^{c\times c}$ be a diagonal matrix with positive values. For any matrix $N \in \mathbb{R}^{c\times(D-c)}$, the matrix*

$$A(N) \coloneqq \begin{bmatrix} D & N \\ N^T & 0 \end{bmatrix},$$

is positive if and only if $N = 0$.

Proof. If $N = 0$ the matrix $A(0)$ is positive because for all $x \neq 0 \in \mathbb{R}^D$, $x^T A(0)x = \sum_{i=1}^c d_i x_i^2 \geq 0$. If $N \neq 0$, if we write $x = (x_1, x_2) \in \mathbb{R}^D$ with $x_1 \in \mathbb{R}^c$ and $x_2 \in \mathbb{R}^{(D-c)}$, then

$$x^T A(N)x = x_1^\top D x_1 + 2x_1^\top N x_2.$$

So if we take any vector x_2 such that $\|Nx_2\| > 0$ (such vector exists because N is not null) and $x_1 = -Nx_2 \times \varepsilon$ for $\varepsilon > 0$. We have:

$$x^T A(N)x = \varepsilon^2 \left\| \sqrt{D} N x_2 \right\|^2 - 2\varepsilon \|Nx_2\|,$$

so for ε small enough $x^T A(N)x < 0$. □

Let us now prove Theorem 4.3.

Proof. Let us introduce some notations.

Throughout the proof we will note $\Sigma = \Sigma_{\tilde{\alpha}}$. We note $W = \Sigma^{-1/2}\hat{V} \in \mathbb{R}^{D\times c}$ and $A = \Sigma^{1/2} M^\top M \Sigma^{1/2} \in \mathbb{R}^{D\times D}$.

With these notations,

$$\mathrm{Tr}(M\Sigma M^\top) = \mathrm{Tr}(\Sigma^{1/2} M^\top M \Sigma^{1/2}) = \mathrm{Tr}(A),$$

and

$$\lambda_{\min}(\hat{V}^\top M^\top M \hat{V}) = \lambda_{\min}(\hat{V}^\top \Sigma^{-1/2} A \Sigma^{-1/2} \hat{V}) = \lambda_{\min}(W^\top A W).$$

Let us write $W \coloneqq PDQ$ the SVD decomposition of W, with $P \in O(D)$, $Q \in O(c)$ and $D \in \mathbb{R}^{D\times c}$ a rectangular diagonal matrix with positive real numbers on the diagonal, i.e. $D = [D_0, 0]^T$, with $D_0 = \mathrm{Diag}(\sigma_1, \cdots, \sigma_c) \in \mathbb{R}^{c\times c}$. If we write $A' = P^\top A P$, we then have

: $\lambda_{\min}(W^\top AW) = \lambda_{\min}(Q^T D^\top P^\top APDQ) = \lambda_{\min}(D^\top A' D)$ and $\mathrm{Tr}(A) = \mathrm{Tr}(P^\top AP) = \mathrm{Tr}(A')$. With all these notations, the criterion (4.6) can be rewritten:

$$\frac{\mathrm{Tr}(A')}{\lambda_{\min}(D^\top A' D)}. \tag{4.13}$$

If we write $A' = \begin{bmatrix} A'_0 & A'_1 \\ A'_2 & A'_3 \end{bmatrix}$, with $A'_0 \in \mathbb{R}^{c\times c}, A'_1 \in \mathbb{R}^{c\times(D-c)}, A'_2 \in \mathbb{R}^{(D-c)\times c}, A'_3 \in \mathbb{R}^{(D-c)\times(D-c)}$ and $A'_1 = (A'_2)^\top$ because A is symmetric.

Then, $D^\top A' D = D_0 A'_0 D_0$ and $\mathrm{Tr}(A') = \mathrm{Tr}(A'_0) + \mathrm{Tr}(A'_3)$. So to minimize (4.13) we have to set the diagonal of A'_3 to zero and since A'_3 is symmetric and positive semidefinite (as a sub-matrix of A' which is itself symmetric and positive semidefinite) it implies that $A'_3 = 0$, while A'_1 does not appear in the criterion and therefore can be chosen freely.

Let us write $A''_0 = D_0 A'_0 D_0$, so that $\lambda_{\min}(D^\top A' D) = \lambda_{\min}(A''_0)$ and $\mathrm{Tr}(A') = \mathrm{Tr}(D_0^{-1} A''_0 D_0^{-1}) = \mathrm{Tr}(A''_0 D_0^{-2})$.

Since A is symmetric A', A'_0 and finally A''_0 are also symmetric. Using the spectral theorem, there exist scalars $a''_1 \geq \cdots \geq a''_c \geq 0$ and $(u_i)_{i=1}^c$ an orthonormal base of $\mathbb{R}^c$ such that $A''_0 = \sum_{i=1}^c a''_i u_i u_i^\top$. Finally, if $a''_c > 0$, (4.13) can be rewritten :

$$\frac{\mathrm{Tr}(A''_0 D_0^{-2})}{\lambda_{\min}(A''_0)} = \frac{1}{a''_c} \sum_{i=1}^{c} \sigma_i^{-2} a''_i.$$

The minimum of this last equation is obtained by taking $a''_1 = \cdots = a''_i$, i.e. by taking $A''_0 = a \times I_c$, for $a > 0$. To fix the ideas, let us write $a = 1$ for the rest of the proof. The minimum is then equals to $\mathrm{Tr}(D_0^{-2})$.

If $A''_0 = I_c$ then $A'_0 = D_0^{-1} A''_0 D_0^{-1} = D_0^{-2}$ and so

$$A = PA'P^\top = P \begin{bmatrix} D_0^{-2} & A'_1 \\ (A'_1)^\top & 0 \end{bmatrix} P^\top.$$

But $W = PDQ$ so that

$$WW^\top = PD^\top DP^\top = P \begin{bmatrix} D_0^2 & 0 \\ 0 & 0 \end{bmatrix} P^\top.$$

Hence, $A = (WW^\top)^+ + P \begin{bmatrix} 0 & A'_1 \\ (A'_1)^\top & 0 \end{bmatrix} P^\top$, and since $M^\top M = \Sigma^{-1/2} A \Sigma^{-1/2}$ then

$$M^\top M = \Sigma^{-1/2} \Big(WW^\top \Big)^+ \Sigma^{-1/2} + \Sigma^{-1/2} P \begin{bmatrix} 0 & A'_1 \\ (A'_1)^\top & 0 \end{bmatrix} P^\top \Sigma^{-1/2}. \tag{4.14}$$

The minimum was equal to $\mathrm{Tr}(D_0^{-2})$ which is equal to $\mathrm{Tr}((\Sigma^{-1/2} \hat{V} \hat{V}^\top \Sigma^{-1/2})^+)$.

In this proof, we minimised over $B \coloneqq M^\top M$ but in the formulation of the problem we minimise over M. For Equation (4.14) to be valid, we need B to be positive.

We have :

$$P^\top \Sigma^{1/2} B \Sigma^{1/2} P = \begin{bmatrix} D_0^{-2} & A_1' \\ (A_1')^\top & 0 \end{bmatrix}.$$

According to Lemma 4.1, $\begin{bmatrix} D_0^{-2} & A_1' \\ (A_1')^\top & 0 \end{bmatrix}$ is positive if and only if $A_1' = 0$. □

Chapter 5

A case study on Multiple Myeloma

In this chapter we explore the use of Random Fourier Features [98] in METAflow.

METAflow is the software developed at Metafora to perform automatic clustering of flow cytometry datasets. The software uses an algorithm based on a density estimator to obtain a hierarchical clustering, i.e. a tree where the nodes are clusters. This algorithm makes it possible to obtain clusters faster than the manual gating method and more "natural" in terms of density, while leaving the user in control of the exploration of the tree. In this exploratory chapter, we investigate how the RFF embedding, introduced in Section 1.2.4, can be used to analyse flow cytometry in a practical example. The RFF embeddings are a characterisation of the node distributions, which we can compute quickly at each node by averaging the leaf embeddings.

After a presentation of the clinical context of this chapter, we show how the embeddings can be use as a complementary tool for exploratory analysis. Then, we propose to investigate distribution feature matching (DFM), introduced in Chapter 3, in this practical setting. The goal is to estimate the proportions of different reference types of cells, in our case different types of white blood cells, for each node in the METAflow tree. Furthermore, we investigate the use of the embeddings to automatically label the nodes of the tree, i.e. for each type of cell of interest, find the node of the tree that is most likely to be that type of cell.

Contents

5.1 Introduction

Multiple myeloma (MM) is a *haematological malignancy* (i.e. a blood cancer) that affects plasma cells, a type of white blood cell that produces antibodies (see Figure 5.1). In multiple myeloma, abnormal plasma cells grow out of control in the bone marrow. This leads to weakened bones, anaemia, kidney damage and a weakened immune system. Common symptoms include bone pain, fatigue, frequent infections and unexplained weight loss. Multiple myeloma is considered a relatively rare cancer, but it is the second most common blood cancer. Approximately 100,000 people worldwide die from multiple myeloma each year.

This pathology develops from an asymptomatic stage of plasma cell proliferation called monoclonal gammopathy of undetermined significance (MGUS). MGUS is the most common plasma cell disorder [110] and is characterised by the presence of an abnormal protein in the blood like MM but MGUS does not cause any symptoms and is usually found incidentally during routine blood tests. MGUS is present in more than 3% of people aged 50 and over, and about 1% progress to multiple myeloma each year. Therefore, most people with MGUS never develop complications and do not require treatment, but it is important to monitor people with MGUS regularly as it can progress to multiple myeloma.

We refer the reader to Kyle et al. [74] for a historical perspective on multiple myeloma and to Cowan et al. [24] for a clinical presentation.

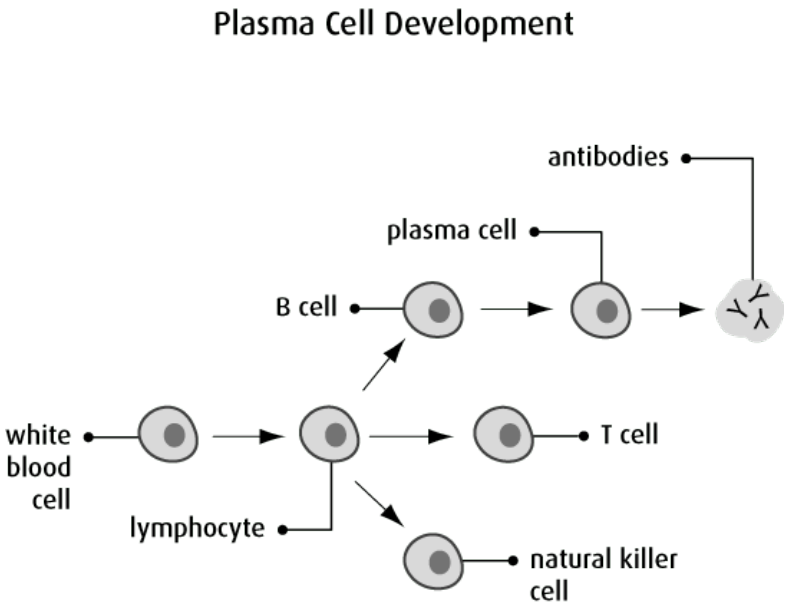


Figure 5.1: Diagram of plasma cell development from the Canadian Cancer Society [110].

Common questions asked by patients and clinicians are "What is the risk of progression from MGUS to MM" and "Is there anything that can or should be done to delay or prevent progression?". Currently, clinicians lack reliable biological predictors of progression from MGUS to MM. In the absence of such markers, MGUS patients are currently risk stratified on the basis of a few selected clinical variables identified in epidemiological studies. Better markers are needed to define high-risk (versus low-risk) MGUS and to better predict individual risk of MM progression.

This is the aim of the "Multiple Myeloma" project at Metafora : investigate the use of RBDs (Definition 5.1) as immunophenotypic markers to differentiate the plasma cells of MGUS and MM. For the remainder of this chapter, we will refer to the plasma cells as *plasmocytes*.

Definition 5.1 (RBD)**.** The Receptor-Binding Domain (RBD) is a component found on the surface of certain viruses, notably coronaviruses like SARS-CoV-2 (COVID-19). It is a specific region of the virus's spike protein, which protrudes from the viral envelope and facilitates the virus's entry into host cells. The RBD is responsible for binding to specific receptor molecules on the surface of host cells, initiating the process of viral infection.

Actually, all the lymphocytes and not just the *plasmocytes* are of interest in this study, see Figure 5.1. The four types of lymphocytes we will consider are the *T cells*, the *B cells*, the *NK cells* (for "Natural Killer"), and the *plasmocytes*.

5.1.1 Multiple Myeloma datasets

We have access to a cohort of 29 patients enumerated from 11 to 49 (we excluded some of the patients because we lacked the labels).

For each patient we have access to a **bone marrow sample** analysed by flow cytometry, (we have described the operation of a flow cytometer in Section 1.2.2). Therefore, for each patient we have access to a sample in $\mathbb{R}^d$, where each point corresponds to a cell and each coordinate corresponds to a given **marker**. The ensemble of markers is called a **panel** (Definition 5.2).

Definition 5.2 (Panel)**.** In flow cytometry, a "panel" refers to a specific combination of fluorescently labelled antibodies that we call **markers** that are used to identify and characterise different cell populations within a sample. Each antibody in the panel is labelled with a unique fluorochrome that emits light at a specific wavelength when excited by a laser in the flow cytometer. By analysing the pattern of fluorescence emitted by cells passing through the flow cytometer, researchers can determine the presence and abundance of specific cell types based on the markers targeted by the antibodies in the panel.

In addition to fluorescently labelled antibodies, flow cytometry also outputs the side scatter (**SSC**) and forward scatter (**FSC**), which provide information about cell size and granularity, contributing to the comprehensive characterization of cell populations within the sample.

The design of a panel therefore involves the selection of the antibodies and the fluorochromes (colours) used for labelling. In particular, the cytometrist must minimise spectral overlap to allow accurate and simultaneous detection of multiple markers in the same sample.

The one used for this analysis is shown in Table 5.1. The morphological markers and the CD markers are used to identify the main lymphocyte subsets and once it is done, the cytometrist can look at the RBD markers.

Patients The number of events per patient ranges from two hundred thousand to sixteen million, as shown in Figure 5.2. Only a small proportion of these events are lymphocytes. Bone marrow also contains other types of white blood cells (granulocytes and monocytes), red blood cells and platelets. The samples also contain *doublets* (two cells that pass in front of the laser at the same time), dead cells or cell fragments. Figure 5.3 shows the development of blood cells.

Among the patients, some are healthy and part of the control group, some have MGUS, the asymptomatic first stage of MM, some have smoldering multiple myeloma (SMM), an intermediate stage between MGUS and MM that is also asymptomatic, and some have multiple myeloma. See Table 5.2.

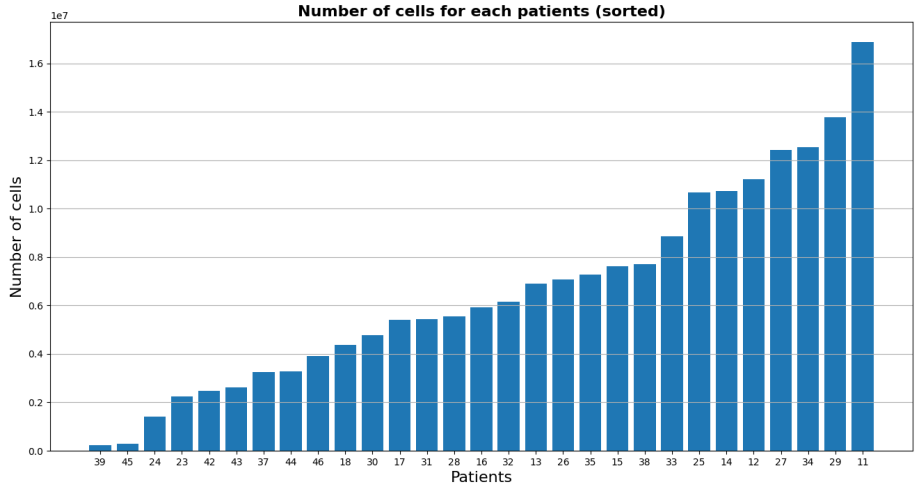


Figure 5.2: Number of cells in each patient. The smallest sample has 217,099 events while the largest has 16,878,172. The lymphocytes represent only a (small) subset of the sample as we will see in Figure 5.5.

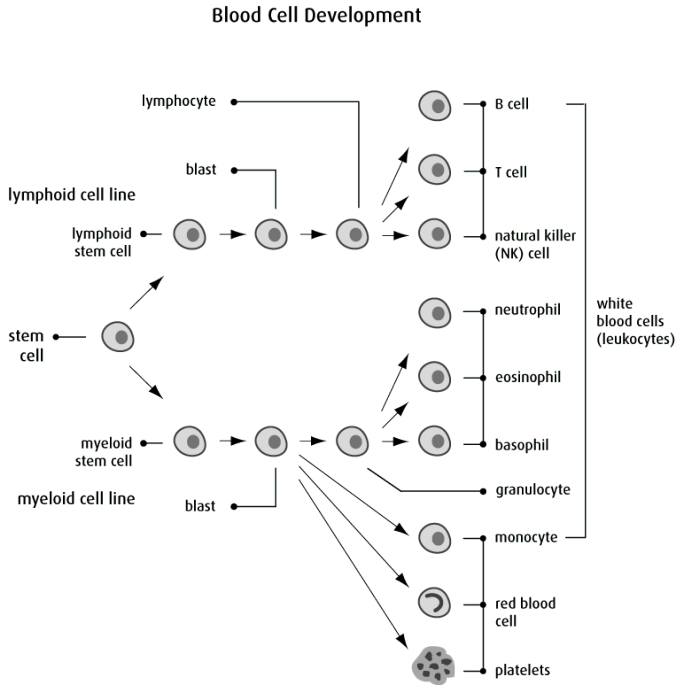


Figure 5.3: Blood cell development diagram from the Canadian Cancer Society [110].

Fluorochrome	**Marker**
Morphological	
FSC	Forward scatter area
SSC	Side scatter area
RBDs	
A color	A RBD
Cluster of differentiation	
PerCP-Cy5.5	CD19
PE-Cy7	CD38
APC-H7	CD44
V500	CD45
V450	CD138
BV786	CD56

Table 5.1: The panel used in our datasets is a combination of the morphological markers (FSC and SSC), which give information about the size and granularity of the cells, the RBDs studied (we have not detailed them as they are confidential) and the CD markers, which are *cell surface* molecules.
Note that in this table, we only show the markers that are common to all patients.

Diagnostic	**Patients**
Control	32
MGUS	13, 16, 27, 28, 33, 35, 43
SMM	11, 24, 46
MM	12, 14, 15, 17, 18, 23, 25, 29, 30, 31, 37, 38, 39, 42, 44, 45

Table 5.2: Breakdown of patients by diagnosis.

Framework The data were analysed using two different frameworks. On the one hand, a classical manual gating strategy was performed on the data (see Figure 1.2 for an example of a manual gating strategy) using the software FlowJo. With this manual gating, we obtain for each patient a set of four manually selected clusters, one for each cell type while the remaining cells are unlabelled. We will refer to these clusters as the **gates**.

On the other hand, a second analysis was performed using METAflow, the software developed at Metafora and presented in the introduction of this manuscript. The software performs a hierarchical clustering on the data using a density estimator based on the algorithm of Chazal et al. [20]. For each patient, the output is a binary tree that can be visualised as a dendrogram, where each node represents a **cluster** and at a given depth we obtain a clustering of the data. In this analysis, all the points belong to several clusters, but we do not have access to the labels, so we do not know which cluster of the tree corresponds to which gate.

In particular, note that the gates are never perfectly recovered by METAflow. We come back to this in Section 5.2.2.

Gating strategy In Figure 5.4 we show the number of *T cells*, *B cells*, *NK cells* and *plasmocytes* for each patient. These FlowJo gates were the one we used in Sections 3.4.2 and 4.5.1 to test DFM and M-DFM procedures.

However, because the colour used conflicted with one of the RBDs, the **CD3** marker is missing from the panel. **CD3** is the marker normally used to identify *T cells*. As the marker is missing from this panel, the cytometrists had to find another gating strategy to select them. They chose to define the *T cells* negatively, i.e. the *T cells* are defined as lymphocytes that are neither *B cells*, *NK cells* nor *plasmocytes*.

Labels Unlike classical machine learning problems where we assume that we have access to a *true* label for each point (at least for the training set), here we do not have access to it. Throughout this section we will assume that the *true* labels are given by the FlowJo gates, but this is a choice and not a truth because the gates are the result of a human process with biases and errors. On the one hand, the gates are drawn by software using square and hard thresholds, which creates "unnatural" gates (in the sense of gates that do not respect the density of the distribution), as shown in Figure 5.12 and on the other hand, the gating strategy and the panel have to be chosen.

For example, since the **CD3** marker is missing from the panel, the *T cells* are defined negatively. This is a choice of gating strategy that is not absolute and another cytometrist could choose differently (in fact, as this approach proved unsuccessful, as we shall see, a new gating strategy was proposed to select the *T cells*, but we will not present this new data in this section).

What's more, even if the gating strategy is perfect and the panel is perfectly selected, flow cytometry can not tell us the *true* genotypic nature of the cells, only the phenotypic information (i.e. the set of observable characteristics or traits of the cells).

5.2 Embedding with Random Fourier Features

We compute the Random Fourier Feature (RFF) embedding of each FlowJo gate and each METAflow cluster. The Random Fourier Features embedding used during the experiments was presented in Section 3.4.1.

5.2.1 FlowJo

Let us start by looking at the FlowJo gates. In Figure 5.4 we show the number of cells in each gate and for each patient and in Figure 5.5 we show the proportions of each cell type in the samples. As we can see, the number of *T cells* far exceeds the number of other cells. For

example, in patient 45 the *T cells* make up more than 80% of the sample. The other cells represent less than 1% of the total data set, except for patient 39 which represents $\sim 5\%$. The overrepresentation of *Plasmocytes* for this patient was already noted by the cytometrists during their analyses and was not considered suspicious. We can therefore assume that this gate represents the patient's plasma cells.

It is highly unlikely that the *T cells* represent this amount in the sample, and it is suspected that the gating strategy of negatively defining the *T cells* is responsible for this overrepresentation. What we can do is look at the RFF embedding of the *T cells* to see if we can detect this anomaly with this tool.

RFF on FlowJo To compute the vectorisation of the gates we have to select a bandwidth.

In Chapter 3 we presented a criterion for choosing it: Δ_{min}. This value represents how far each population is from a mixture of the other populations.

In Figure 5.6 we plot the value of Δ_{min} against the bandwidth σ when the sources are the FlowJo gates. More specifically, for a given cell type, the source is the concatenation of all FlowJo gates from all patients.

Since the number of events per patient is highly unbalanced, as shown in Figures 5.4 and 5.5, this concatenation tends to over-represent large gates at the expense of small ones. This is particularly the case for *T cells*. Therefore, in the figure, we also show in light grey the value of Δ_{min} against the bandwidth σ for the FlowJo gates of each individual patient. As we can see, the optimal bandwidth for the concatenations is $\sigma = 0.32$, while the optimal bandwidth for each individual patient is closer to $\sigma = 0.25$. For the rest of this chapter we will use $\sigma = 0.25$.

Note that the value of the criterion is around 0.2 and 0.5, which is high. This means that the embedding was able to separate the different populations well. However, it is important to remember that in this plot we are only considering 4 cell types. We do not know how the other populations in the samples behave for this bandwidth.

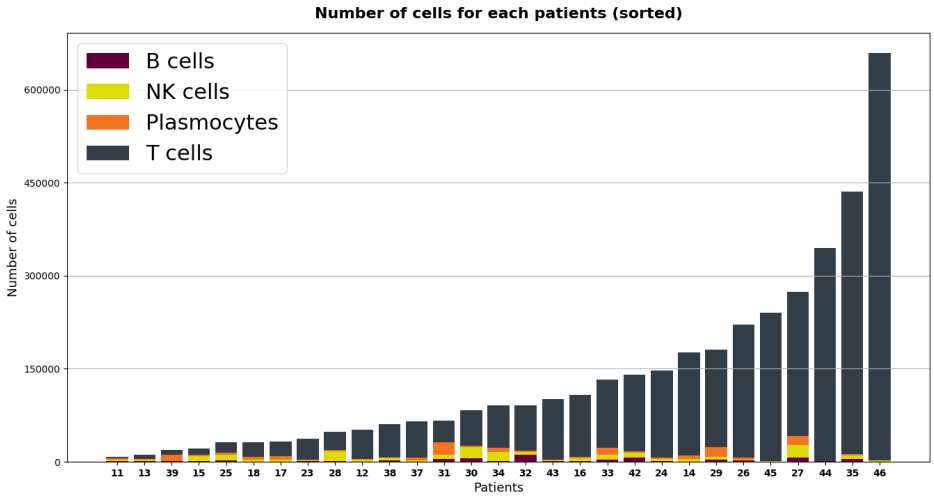


Figure 5.4: Number of cells in each FlowJo gate for each patient. Patients are sorted according to the number of cells of interest.

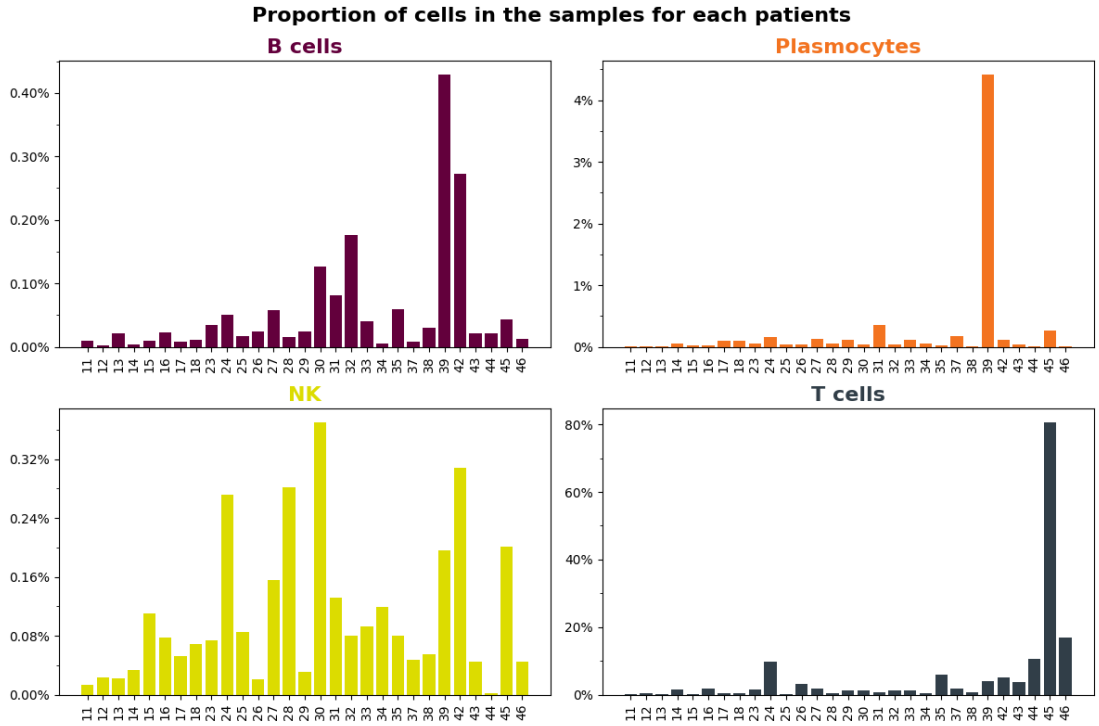


Figure 5.5: Proportion of cells in the samples for each patient.

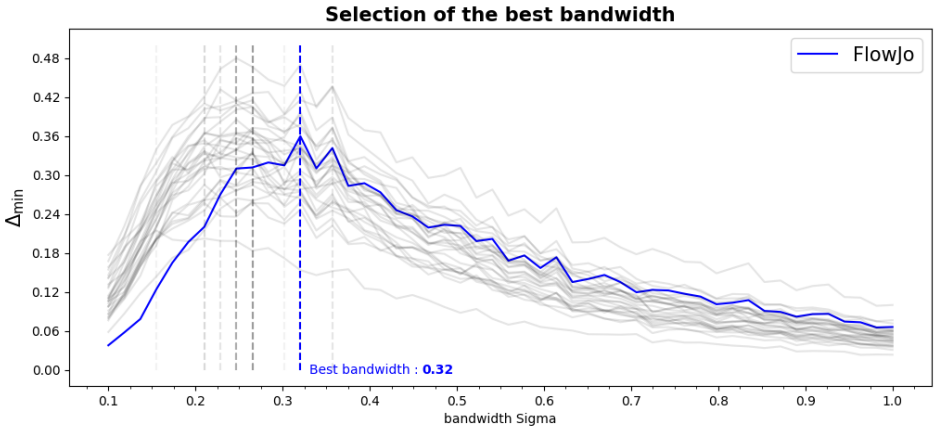


Figure 5.6: Value of the $\Delta_{\min}$ criterion for different values of σ. In blue, the sources are a concatenation of all FlowJo gates. Each grey line corresponds to one patient. The grey dashed vertical lines are the argmax of each grey line and the blue dashed vertical line is the argmax of the blue line.

In Figure 5.7 we show for each type of cell the distance matrix between the embedding of each patient's gate, it highlights several characteristics of the data.

Firstly, looking at the *B cells* and *NK cells*, we can clearly see that there are two groups of patients: patients 11 to 35 and patients 37 to 46. According to the table 5.2, the patients in the second group all have multiple myeloma except patient 46 who has SMM, but some patients in the first group also have multiple myeloma, so the diagnosis is not the reason for this discrepancy.

To understand what is happening in the figure, we need to look directly at the FlowJo gates. In Figure 5.8 we show the histograms of each marginal of the *B cells* and the *NK cells*. As shown in Figure 5.12, once the lymphocytes have been identified the markers used to identify the *B cells* and the *NK cells* are the markers CD19 and CD56. We do not see much difference between the two groups on these coordinates, which explains why the cytometrist did not see the change in distribution during the analysis. Two markers can be highlighted to explain this change: CD138 and CD44. After verification, it appears that the "APC-H7" fluorochrome associated with CD44 was associated with a new marker after patient 37.

Therefore, for the rest of this section, we will remove this marker and use only the other five. In Figure 5.9 we show the distance matrix again, but this time for the new data.

A clustering can still be read from the matrices, especially for the *NK cells*, but the two previous groups have disappeared. A more detailed analysis could be carried out to understand the exact nature of these clusters, but this is beyond the scope of our work. In fact, if we only wanted to differentiate between *B cells* and *NK cells* with a group as homogeneous as possible, we would use only the two markers: CD19 and CD56 as in Figure 5.12. The problem

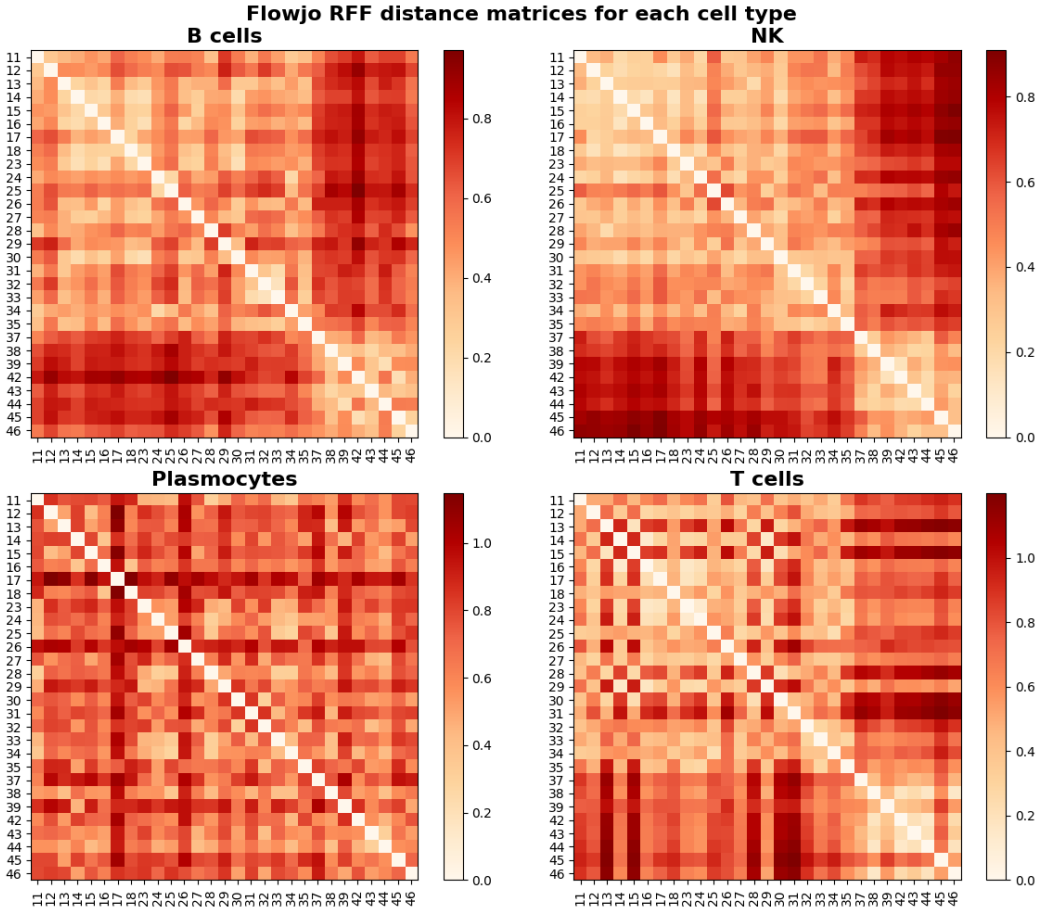


Figure 5.7: Pairwise distance of patients' FlowJo gates for each cell type.

is that with only these two markers we can not differentiate between the lymphocytes and the other cells. Our analysis (as well as METAflows) is multidimensional. The other dimensions explain the *in-group* variability of each cell type.

What is really important to us is firstly that the different FlowJo groups are as separated as possible, we will see that this is indeed the case, and secondly that the cells of interest are separated from the other cells.

In Figure 5.10 we show a two-dimensional representation of the gates using t-SNE [124]. It is important to note that this representation is "*cherry-picked*" because t-SNE depends heavily on one parameter: the **perplexity**. A different value would give a different result. In the figure we use $\text{perplexity} = 6$.

This plot shows that the RFF embedding allows us to distinguish between the different cells. We can also see that one *plasmocytes* gate looks more like a *B cells* than a *plasmocytes*,

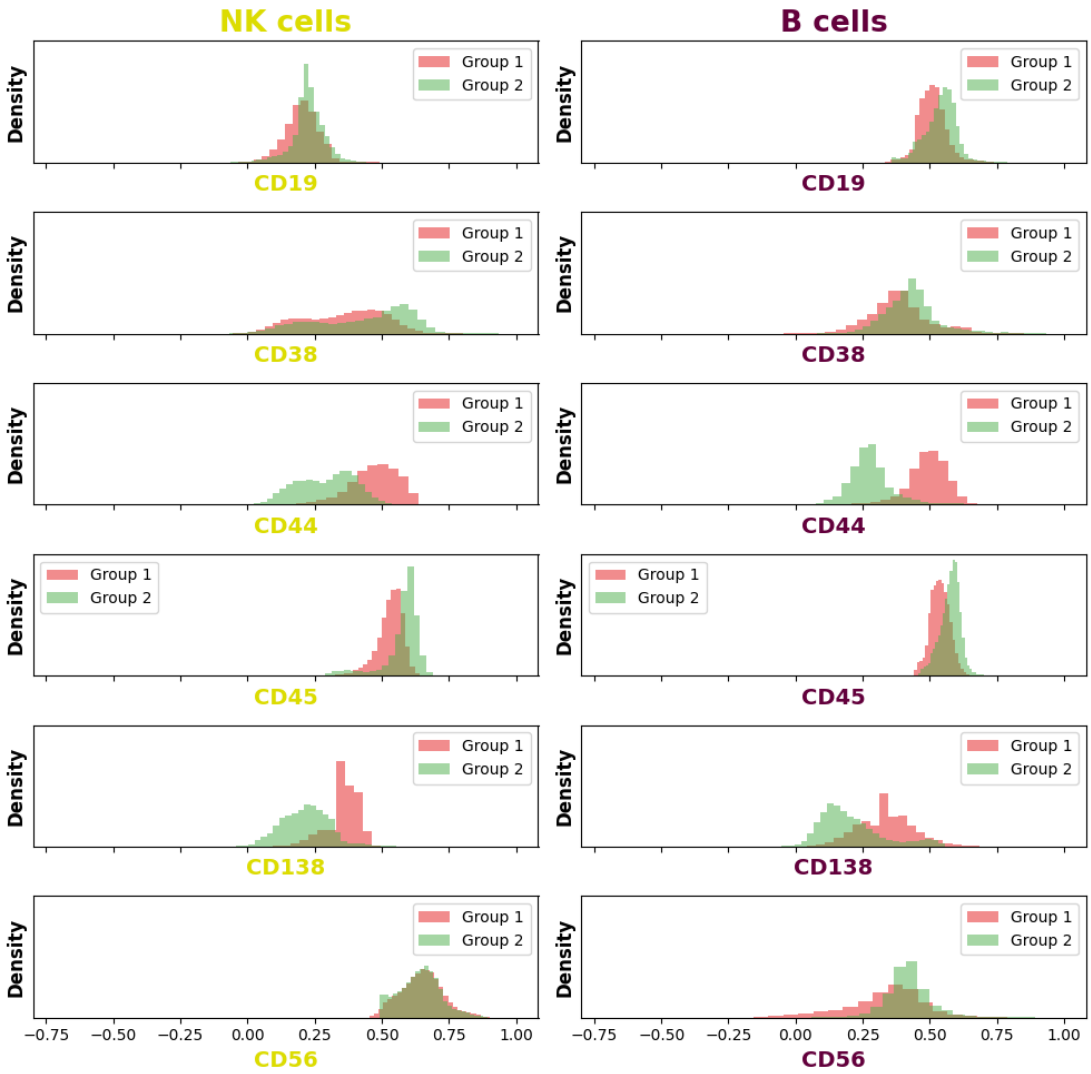


Figure 5.8: Histogram of each marginal for the *B cells* and *NK cells*.

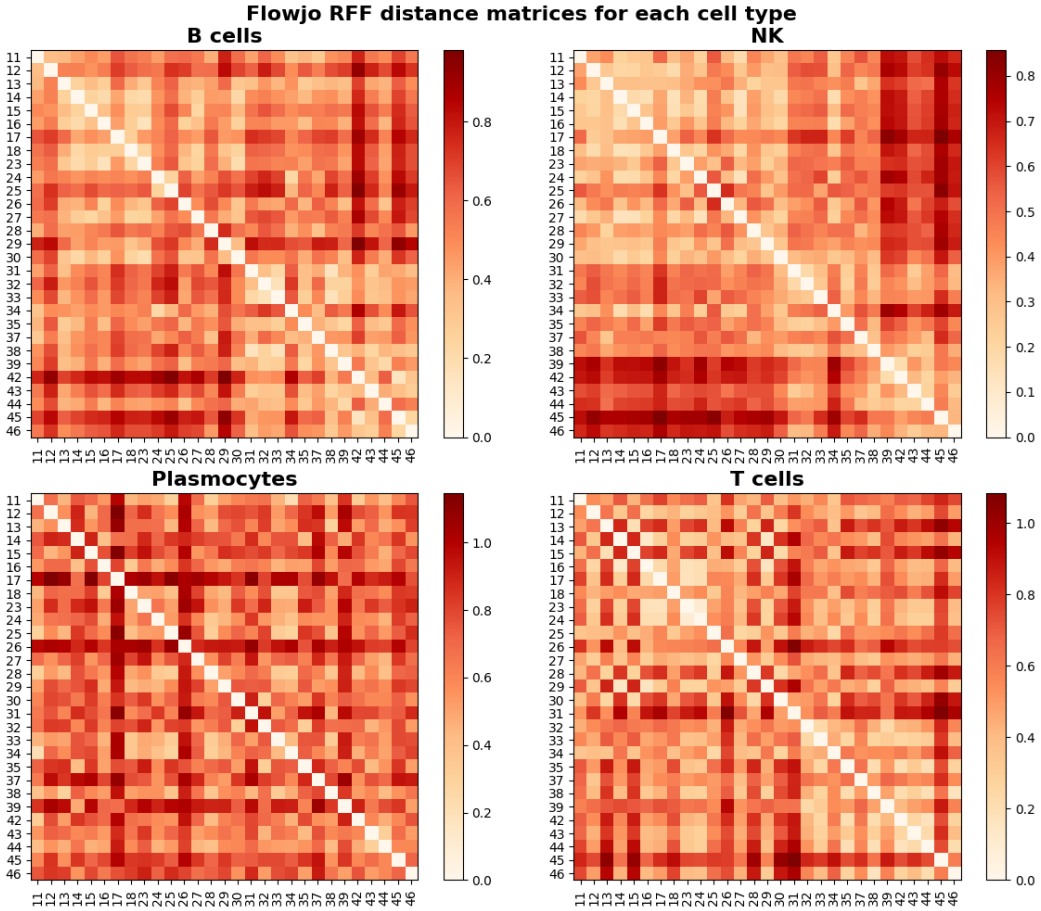


Figure 5.9: Pairwise distance of patients' FlowJo gates for each cell type, excluding the *APC-H7* fluorochrome. The bandwidth used is $\sigma = 0.25$.

after verification this abnormal gate belongs to patient 26.

The reason for this is that the *plasmocytes* were not found in the sample 26, probably because the marker used to differentiate between *plasmocytes* and *B cells* can disappear over time if the sample is not analysed early enough. For this patient, the expert chose a different gating strategy to find them. Using the RFF, we can see that the corresponding gate looks more like a *B cells* than a *plasmocytes*, so this new gating strategy was not successful.

5.2.2 METAflow

METAflow performs a hierarchical clustering on the data using a density estimator. The estimator uses all available markers, including the RBDs or the CD44 marker that we have removed when calculating the RFF, as well as other markers that we have not included in Table 5.1 because they are not common to all patients.

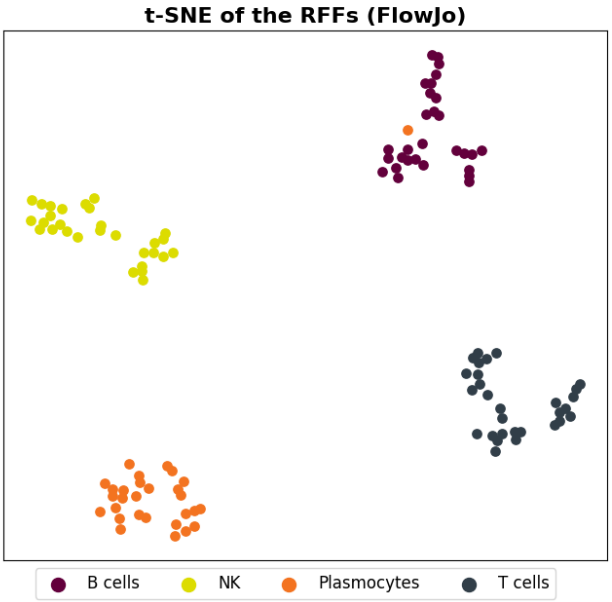


Figure 5.10: Two-dimensional representation of the RFF embeddings of the FlowJo gates using t-SNE [124].

The output of the algorithm is a tree, where each node represent a cluster.

If we have access to the FlowJo gates, we can use the F1 score to find the nodes that best match the gates.

Definition 5.3 (F1 score)**.** The F1 score between two sets is a measure of similarity. It is defined as twice the harmonic mean between precision and recall.

In Figure 5.11 we show the value of the best F1 score for each patient and each cell type. An arbitrary threshold of 0.6 is used at Metafora to distinguish between clusters that were recovered and clusters that were not. As we can see, METAflow is on average able to recover the FlowJo gates. However, the gates are rarely recovered perfectly. In fact this is desirable as we show in Figure 5.12. The FlowJo gates are the result of a manual procedure and are often obtained by "cutting" at a given threshold in a certain direction. Therefore they do not respect the density of the underlying distribution. In a way, the best METAflow gates are *smoothed* versions of the FlowJo gates.

Without delving into the clustering, we can not be sure that METAflow has not found clusters that are more likely to be the cell of interest than the FlowJo gates, as we do not

have access to the *true* labels. Therefore, the F1 score must be seen as a measure of the agreement between the manual and automatic gates, not as a measure of the quality of the clustering, and since we do not have access to the *true* labels, we can not compute such a measure.

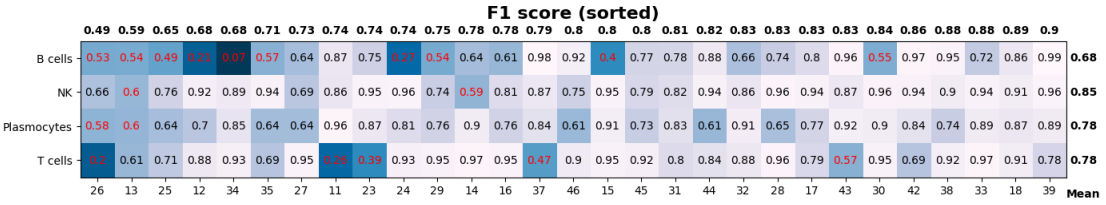


Figure 5.11: This heatmap shows the best F1 score of the tree for each cell type and patient. Patients are sorted by their mean F1 score. The mean F1 score for each cell type is shown on the right of the heatmap. The value of the F1 score is in red if it is below the (arbitrary) threshold of 0.6.

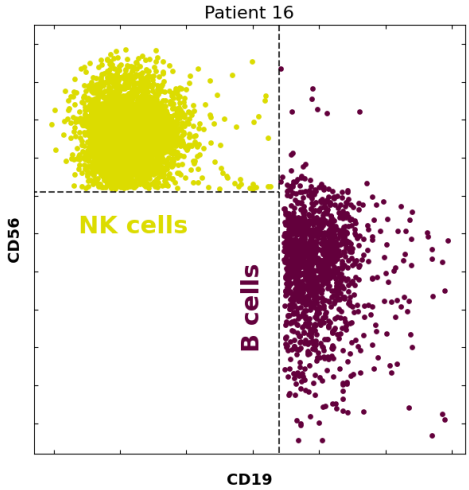


Figure 5.12: Scatter plot of the *B cells* and *NK cells* of patient 16 according to FlowJo for two markers used in the gating strategy to gate the cells: CD19 and CD56. The dashed black lines have been added by us to illustrate that the FlowJo gates are the result of manual selection. This highlights two problems with this gating method. Firstly, the two gates are arbitrary in the sense that if the same cytometrist were to redo the gating, the threshold used (here in black) would not be exactly the same. Secondly, the resulting gates are not *natural* in terms of the density of the distribution. It is therefore not surprising that METAflow can not recover the exact gates.

RFF on METAflow As we explained in the introduction (Section 1.2.3), the advantage of a mean embedding approach is its linearity with respect to the distributions. To compute the Random Fourier Features embedding of a node, we need to compute a weighted average of the embeddings of its children. As a result, the time required to compute the embedding of each node is approximately the same as the time required to compute the embedding of each point in the data set.

Note that this is not necessarily fast, as the number of points is massive : from two hundred thousand to sixteen million, see Figure 5.2.

For each node of a tree, we compute the RFF embedding (with $\sigma = 0.25$) of the cluster using only the 5 markers presented in Table 5.1 (minus CD44 as explained), the seed is fixed so that the embedding of the FlowJo gates can be compared with the embedding of the nodes.

5.3 Labelling the nodes

In this section we want to use RFF to automatically label 4 nodes of the tree with our labels: *B cells*, *T cells*, *Plasmocytes* and *NK*. We will consider two scenarios.

In the first, we assume that the specialists perform the gating using FlowJo, but want to have access to clusters that are more "natural" than those obtained with the first analysis. Therefore, the experts perform a second analysis, this time using METAflow. In this case, the objective is to select the node in the tree that "corresponds" to the FlowJo gate. Previously, we did this using the F1 score because it balanced precision and recall, i.e. the cluster takes as many points as possible from the gate while controlling the number of false positives. Here we explore the use of RFF to select the best cluster.

In the second scenario, we assume that the FlowJo gates are not available. In this case, the specialists use METAflow directly on the data. Again, the goal is to select the clusters that correspond to the cell of interest. This is not an unsupervised problem because we assume that we have access to a reference sample.

5.3.1 When we have access to the FlowJo gates.

For each patient and cell type we calculate the RFF of the FlowJo gate, the selected cluster is the closest cluster in terms of RFF distance. In Figure 5.13 we show the F1 score of the selected cluster compared to the best possible F1 score. Since we are using the F1 score as a measure of quality and no longer as a measure of concordance, as we did for example in Figure 5.11, we will necessarily get clusters with worse or equal F1 scores as if we had taken the best F1 score. Once again, we are faced with the difficulty of not having access to the true label of the data, since without going into the clusters we can not be sure that the selected clusters are not better than those with the best F1 scores.

In this setting, RFF is a good tool for cluster selection, except for *T cells*. In particular,

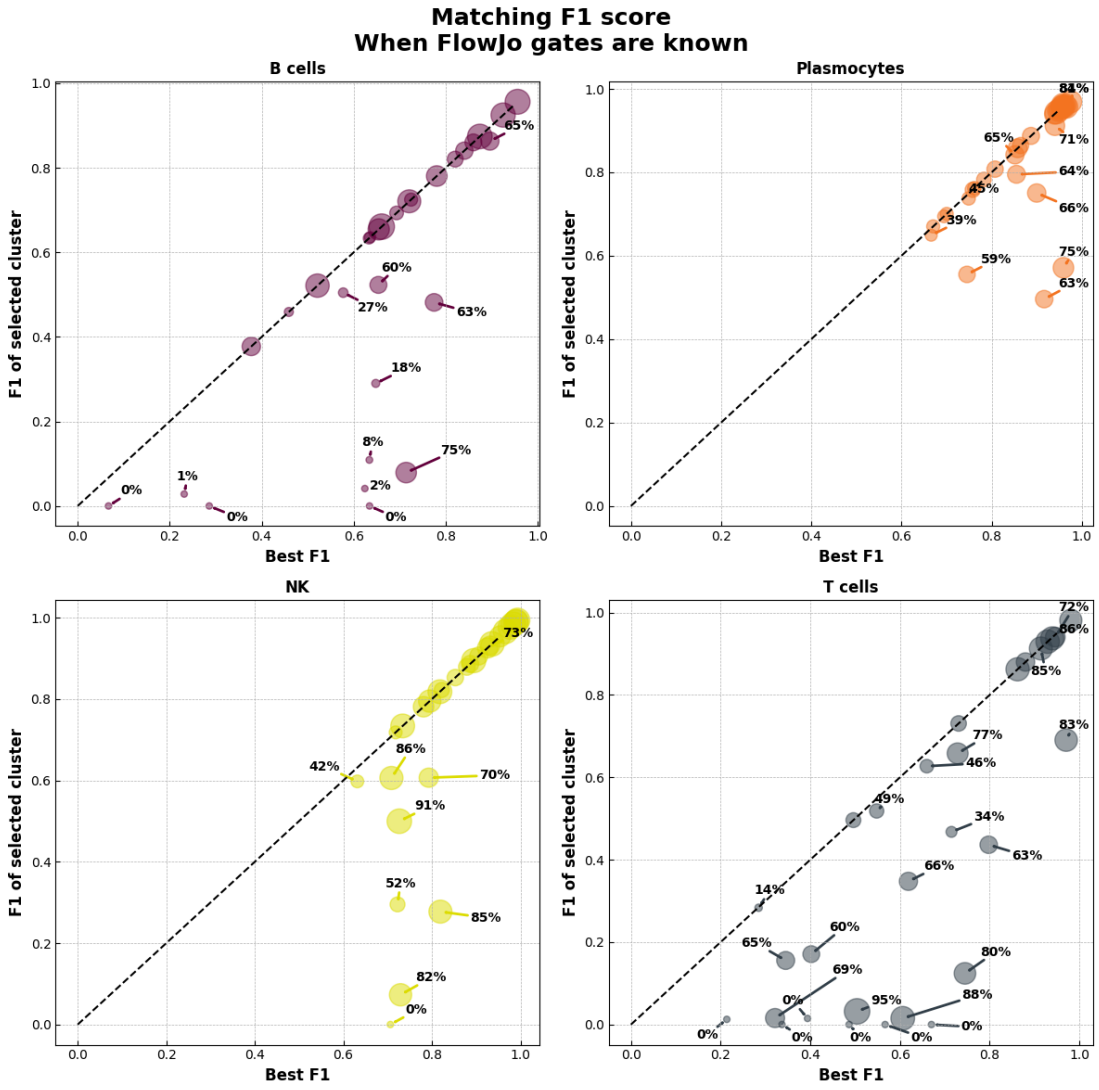


Figure 5.13: For each cell type, we plot the F1 score of the selected cluster against the best F1 score. The size of the points is proportional to the *precision* of the selected cluster. If the method selects a different cluster than the method that directly uses the F1 score, we print the precision of the cluster next to the point.

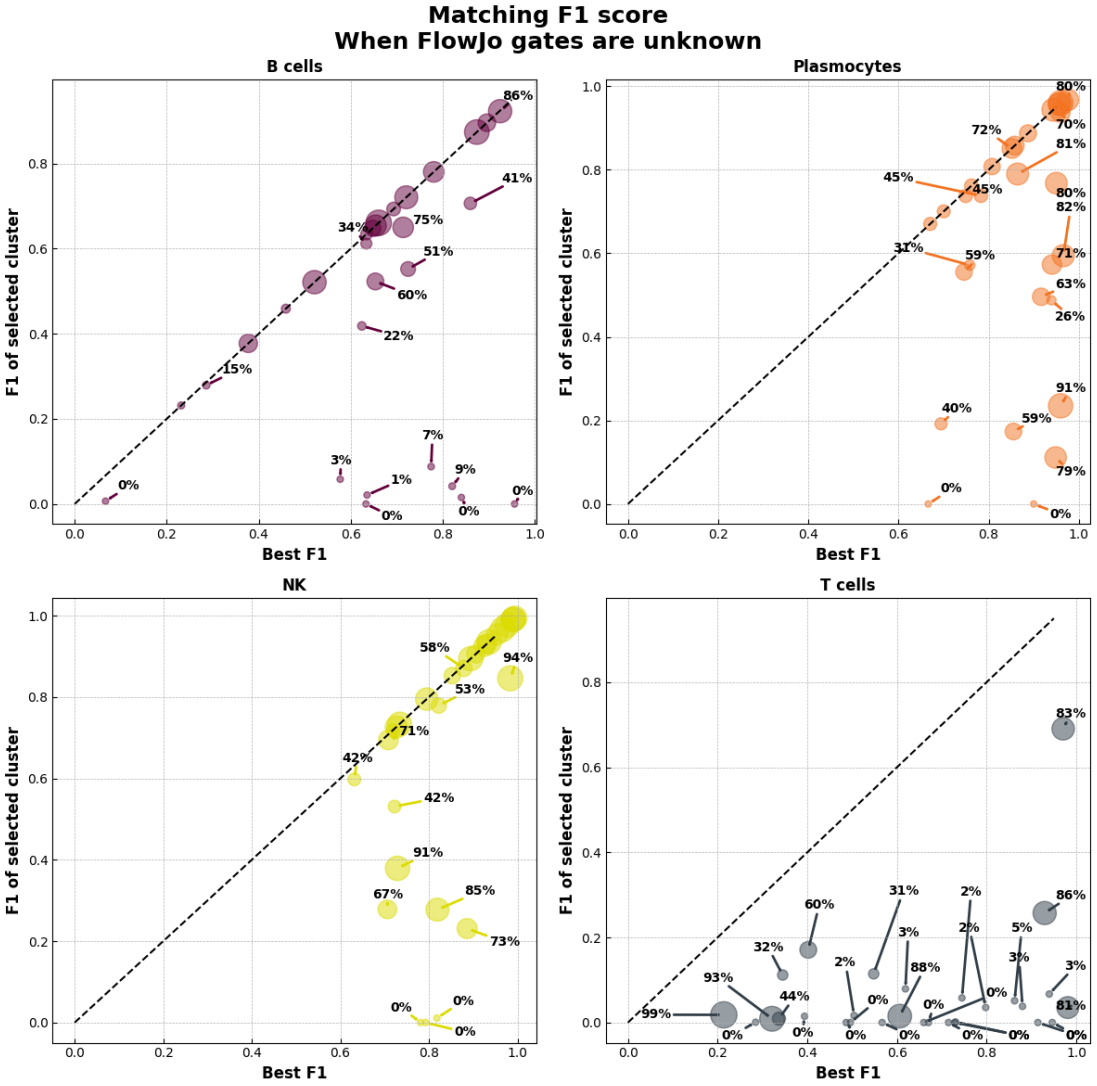


Figure 5.14: For each cell type, we plot the F1 score of the selected cluster against the best F1 score. The size of the points is proportional to the *precision* of the selected cluster. If the method selects a different cluster than the method that directly uses the F1 score, we print the precision of the cluster next to the point.

plasma cells, although heterogeneous as seen in Figure 5.9, are well found. We also note the presence of clusters with poor F1 scores but high precision (and therefore poor recall), that are small subset of the best cluster.

Since in this setting we have access to the F1 score of the clusters, a hybrid approach could be used where we select the clusters using the RFF, but ensuring that the F1 score or precision (depending on the criterion we set ourselves) is not too low, in order to eliminate the few very poor matches we find.

5.3.2 When we do not have access to the FlowJo gates

If we do not have access to the FlowJo gates, we have to use other sources. We assume that the specialist has already gated all but one patient, and we use this information to match the cluster in the tree. Note that this is different from calculating the RFF embedding of the concatenations as we did to select the bandwidth in Figure 5.6. By averaging the RFF embedding directly, we ensure that the small gates are not dominated by the large gates.

In Figure 5.14, we show the F1 score of the selected cluster compared to the best possible F1 score.

In this other setting, the results are way more mitigated. Once again, we note that many of the clusters found, particularly for plasma cells, have a low F1 score but a high precision.

5.4 Quantification

In this last section we want to estimate the proportions of the 4 classes in each node of the tree. The goal is to help the specialist explore the tree by pointing out the "directions" in the tree in which the proportions are higher. Once we have computed the RFF embedding of each node of the tree (the *targets*, to use the terminology used throughout the manuscript), we can quickly solve a QP problem for each node.

As in the previous section, we need to choose our sources. For the same reasons, we consider two choices: when we have the 4 FlowJo gates of the patients and when we have the FlowJo gates of all others patients.

The sources are calculated exactly as in the previous section.

In Figure 5.15 we show the estimated proportion of each selected cluster (selected using the F1 score), against the true proportions when we have access to the FlowJo gates of the patient, while in Figure 5.16 we show the same, but when we use the mean embedding of the other patients as source. In Table 5.3 we show the errors measure using the absolute distance between the estimation and the true proportions.

The error is lower when considering all clusters rather than only the clusters selected by the F1 score. This is because many clusters are far from the cells of interest, indicating the robustness of the method. As a result, numerous clusters have their proportions estimated at zero. As shown in Table 5.3, using the FlowJo gates from the same patient or from other

patients only marginally changes the results, with a slight advantage for the same patient's FlowJo gates.

When only the best clusters are considered, the error is significantly higher. As seen in Figure 5.15, the estimations almost always overestimate the true proportions. Although it is slightly better when using FlowJo gates from other patients, the results remain poor.

Cells	All cluster		Best cluster	
	Same patients	Different patients	Same patients	Different patients
B cells	0.267 ± 0.234	0.3 ± 0.231	0.407 ± 0.224	0.389 ± 0.231
Plasmocytes	0.033 ± 0.09	0.05 ± 0.099	0.33 ± 0.157	0.229 ± 0.204
NK	0.056 ± 0.123	0.078 ± 0.145	0.239 ± 0.151	0.183 ± 0.187
T cells	0.503 ± 0.292	0.623 ± 0.33	0.268 ± 0.294	0.096 ± 0.357

Table 5.3: For each cell type, the table shows the mean error and standard deviation, categorized by whether the data source is from the same patients or different patients. The "All cluster" columns represent the error for all clusters, while the "Best cluster" columns show the error for the cluster selected using the F1 score.

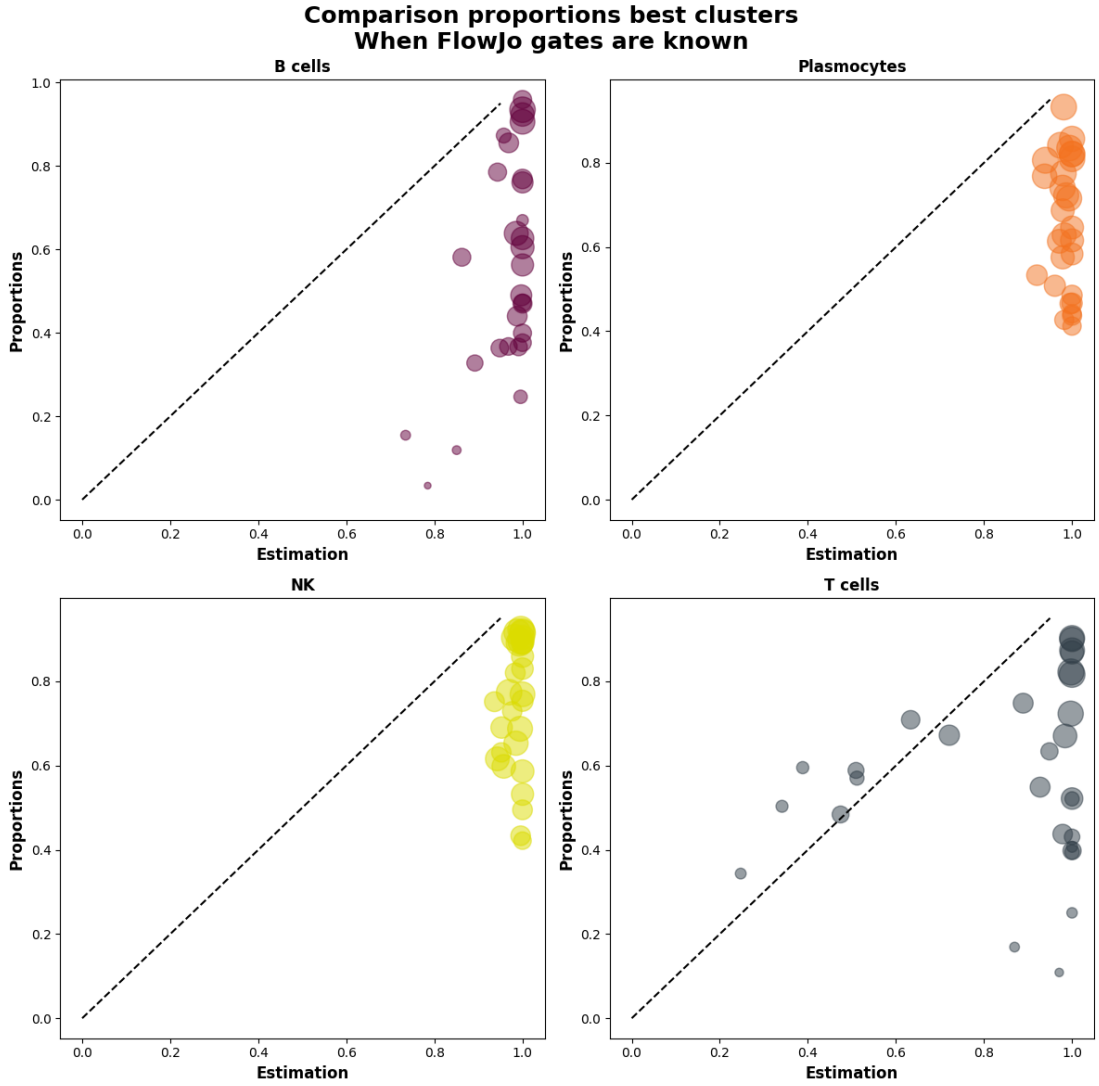


Figure 5.15: Proportions estimated against the true proportions of each selected clusters.

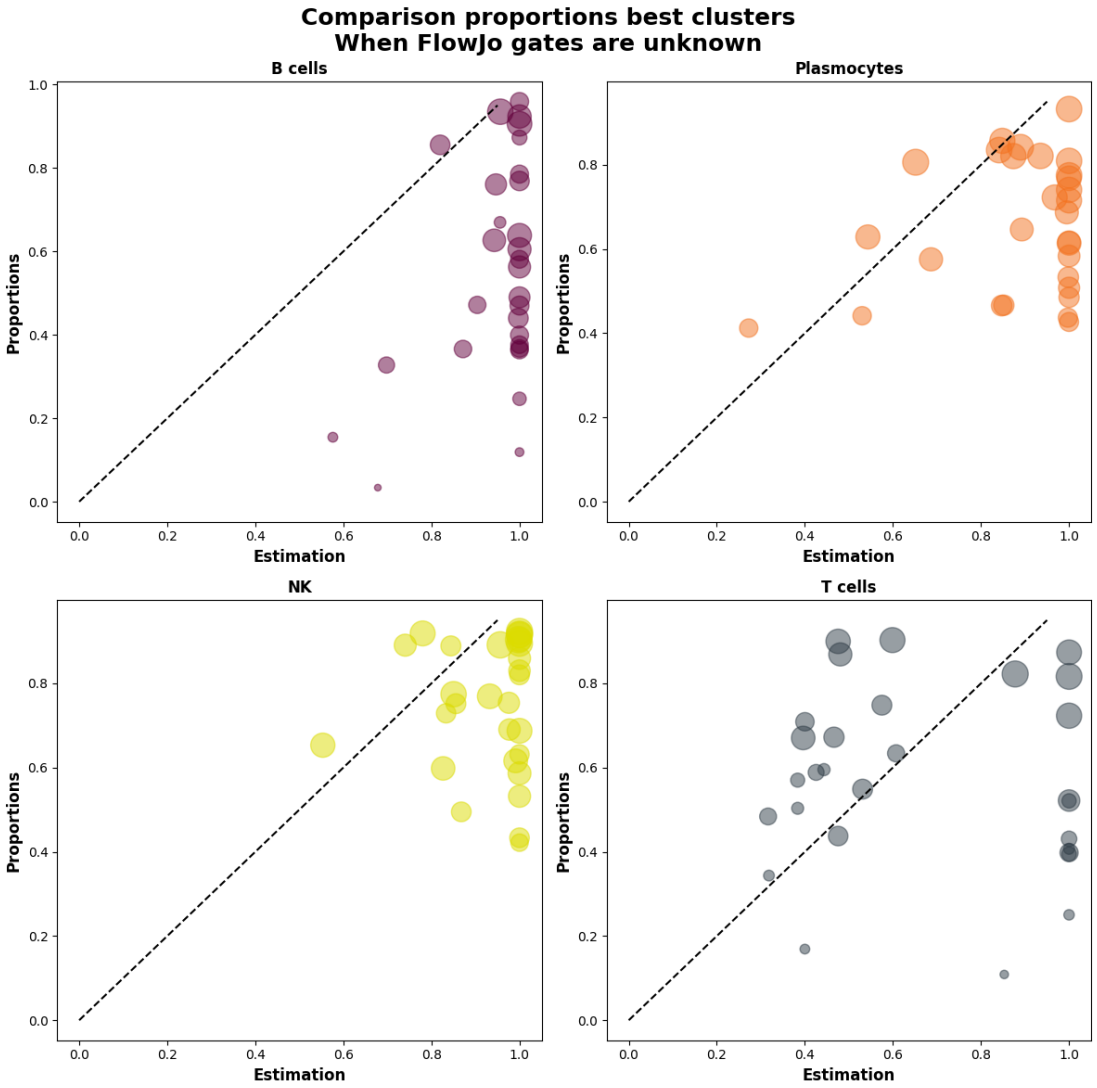


Figure 5.16: Proportions estimated against the true proportions of each selected clusters.

5.5 Conclusion

RFF embeddings can be used as a complementary tool for flow cytometry analysis. They are fast to compute and adapt well to the number of points and tree structure of METAflow. In the use case presented in this chapter, we successfully used the embeddings to detect a problem in the marker used that might have gone unnoticed because the problem was on markers that are not used to differentiate the different lymphocytes. The RFF can also be used to check that a gating strategy has been successful. In this example, we were able to see that one of the patient's plasma cells had not been successfully gated. It would be interesting to look at the selected (using the RFF) plasmocyte gates from this patient to see which cluster was selected.

For the problem of labelling the nodes, the results are complex to analyse because the F1 score is used both as a way to select the cluster and as a way to evaluate the quality of a selection. Therefore, the clusters selected using the RFF can only be worse than those selected using the F1 score. To correctly measure the quality of the selected clusters, we would have to look at them manually. Nevertheless, it is hard to imagine that those with 0% accuracy are great clusters, so in the case where we have the FlowJo gates, a hybrid approach might be more reasonable.

For quantification, however, the results are poor, even when we use the FlowJo gates as sources. Two reasons explain these results. In Corollary 3.2, we highlight the robustness of DFM with RFF embedding to contaminations that are far from the source distributions, and as we saw in Figures 5.6, 5.10 or in the experiments we performed with these data in Chapter 3 and 4, the different lymphocyte populations are well separated. However, this is not the case for all other non-lymphocyte cells in the sample. Moreover, the non-lymphocytes cells of a cluster tend to be close to the lymphocytes because they are part of the same cluster. This explains why the estimated proportions are completely overestimated.

Furthermore, as we can see in Figure 5.9, the references for a given cell type are heterogeneous. This means that the *open set label shift* hypothesis is not verified, i.e. $\mathbb{P}_i \neq \mathbb{Q}_i$. In Theorem 3.3 we show that this was not a problem under an orthogonal condition of the embeddings, which is not verified here.

In flow cytometry, and in molecular biology in general, this inter-sample variability is called the *batch effect*. Due to the experimental nature of the experiments, many non-biological factors, such as laboratory conditions, changes in the instruments used to perform the experiment, or even the exposure of the sample to light, can cause a shift in the distribution. There exists pre-processing steps to mitigate this phenomenon, such as CytoNorm [125], but we have not used them here.

Conclusion and perspectives

In this thesis we have worked on a problem of proportion estimation called "**label shift quantification**", as well as on an extension called **open set label shift quantification**. Introduced as a problem in its own right in a series of articles by Forman [46, 47, 48], **label shift quantification** has since gained increasing attention in the machine learning community, but remains somewhat confidential.

In the first chapter of this thesis, we presented an overview of the methods introduced by the *quantification* community, as well as some methods introduced by researchers outside the community. We also discussed the evaluation protocols and, importantly for the rest of the thesis, the works that have been proposed to unify the different methods of the literature under the same framework.

In this spirit, we proposed our own framework: **Distribution Feature Matching** of **DFM**, a method based on mean vectorisation (in $\mathbb{R}^D$ or, more generally, in any Hilbert space) that includes several methods from the literature, and we proposed a theoretical analysis of the convergence of the framework. To the best of our knowledge, this is the first time that a unifying framework has been proposed to obtain a convergence theorem that applies to all methods. Moreover, the bound we obtained was tighter than those found in the literature. We also investigated the convergence property of DFM under open set label shift and showed robustness to noise for the Gaussian kernel when the noise distribution is far from the source.

In the next chapter, we presented a *covariance-aware* extension called M-**DFM**, where we take into account the covariance information of the embeddings and not only the means as in DFM. Using a vectorial Bernstein inequality, we showed that M-DFM gives better guarantees.

Chapter 3 was published in *Machine Learning and Knowledge Discovery in Databases: Research Track. ECML PKDD 2023. Lecture Notes in Computer Science, vol 14173. Springer, 2023* [30], and a patent has been filed by Metafora, University Paris-Saclay and the CNRS, based on the article, on a "*method for determining proportions of populations in an ensemble of biological objects*" using the Random Fourier Features vectorisation.

In the final chapter, we explore the use of this vectorisation in flow cytometry applications, specifically on three tasks: analysis of manual gates, labelling of nodes, and quantification. The first was successful, we believe that the characterisations obtained through this vectorisation can be used to compare the gates, to assert that the gating strategy was successful and to find irregularities in the dataset, such as mislabelling. The results of the second applica-

tion are more complex to analyse because we do not have access to the true labels and more research needs to be done. Unfortunately, the quantification fails because the assumptions we make about the distribution are not verified.

Several potential avenues for future research and development based on the findings and insights presented in this thesis are outlined below.

Minimax rate We did not address the question of the lower bounds in any chapter, and in particular one can wonder if the DFM procedures are minimax.

This question was not addressed by the authors of the methods presented in the first chapter. However, as we have already pointed out, the literature on quantification is more applied than theoretical. Most of the methods we discuss do not have convergence theorem, so the minimax question does not arise. The methods for which we have theoretical guarantees, such as BBSE by Lipton et al. [77] or MLLS by Garg et al. [52], did not propose minimax rates either.

We found one article in the quantification literature that tackles this problem by Vaz et al. [126]. The main contribution of their paper is to propose the first lower bound on the risk of label shift quantification.

Without going into the details of the assumptions, they showed that no estimator could have a faster convergence rate than $\max\left(n^{-1/2}, m^{-1/2}\right)$, which is the rate we obtained for the DFM procedures. A more in-depth review of this paper should be considered to confidently assert that DFM procedures are indeed minimax optimal.

Since label shift quantification is a parametric estimation, it was expected to obtain a parametric rate of convergence. An extension of the work of Vaz et al. [126] would be to consider optimality in terms of the parameters of the problem, in particular the number of classes and on the "ammount of shift" w.

Hypothesis testing In the introduction, we stated that three problems could be addressed: *detect* if a label shift happened in the target, *correct* a procedure developed on the source, and *quantify* the shift. In this manuscript, we focused on the last problem, but we could explore how to adapt our DFM procedures to perform hypothesis testing.

The problem amounts to test whether the target sample $\hat{\mathbb{Q}}$ and the (possibly reweighed) source sample $\sum_{i=1}^{c} \hat{\beta}_i \hat{\mathbb{P}}_i$ have the same distribution.

Lipton and his co-authors, in the article where they "introduce" BBSE (although BBSE already existed under the name *adjusted classify and count* in the quantification literature), proposed *Black-Box Shift Detection* (BBSD) to detect label shift by running a simple two-sample test (a Kolmogorov-Smirnov test) on the outputs of the classifier. We can extend their approach by using any embedding. For instance, with the kernel mean embedding, we would obtain the maximum mean discrepancy, and if we also use a Mahalanobis distance, we would obtain the maximum kernel Fisher discriminant analysis, two statistics for which we

have a rich and detailed literature.

Closely related settings have been studied in the literature. For instance, the setting where we want to test whether a given class is present in the target, i.e. $\alpha_i > 0$ for a given i, has been studied by Gaucher et al. [53] in the binary case under the name *supervised contamination detection*. Their test, called the *Estimated Density Ratio Test*, was shown to be empirically better (in terms of power) than the MMD approach. Their work is of particular interest to us as they have tested their methods on Flow Cytometry datasets.

Another setting studied in the literature, that can be linked to the open set label shift setting, is the *novelty detection* literature, see Pimentel et al. [91]. However, the settings studied assume that the marginal proportions of the different classes in the target are the same as in the source.[1] A natural extension for our setting would be to perform a test when the marginal proportions change.

In depth numerical study In our experiments, we tested only 5 DFM procedures: BBSE, RFFM, EnergyQuantifier, FourierClassifier, and MeanDFM. Additionally, we included the results of a Classify and Count procedure as a baseline. More embeddings could be explored, such as histograms or the intermediate layers of neural networks. Comparing these results with the numerous methods presented in Section 2.2, especially MLLS, often regarded as the state-of-the-art, could provide further insights.

Better bounds for M-DFM In Theorem 4.2, we presented a convergence bound for M-DFM, which we used to derive a criterion for choosing the optimal M. We discussed the nature of this optimal M in detail in Section 4.4, highlighting the differences between it and the pooled covariance matrix $\Sigma_{\tilde{\alpha}}$. However, the theorem is only true for M that are independent of the data, while the optimal M depends on both the pooled covariance matrix and the empirical source embeddings $\hat{V}$. We can deal with the first dependency by splitting the data as explained, but we can not do so for the dependency on $\hat{V}$. This remains a gap which should be bridged in future developments.

One way to navigate around this issue would be to modify the proofs, so that the Gram matrix that appears in the criterion depends on the source embeddings rather than the empirical ones, i.e. we replace the dependence in $\hat{V}$ by a dependence in V directly.
Alternatively, we could show that the optimal M is directly the pooled covariance matrix. Indeed, the experiments in that chapter showed no difference between the two choices of M, even in cases where the pooled covariance matrix is clearly suboptimal according to the criterion. We think that an alternative proof of Theorem 4.2 might yield a different bound with a criterion that depends not on the variances of $\Phi(\hat{\mathbb{P}}_i)$ but on the variances of $\Pi_V\left(\Phi(\hat{\mathbb{P}}_i)\right)$. In other words, only the variances in the subspace generated by the source embeddings would matter for the choice of M. We believe that the optimal M for this new criterion would be the pooled covariance matrix, thus explaining the empirical results.

[1] With the notations we used throughout this manuscript: $\forall i \in [1, \dots, c]$: $\beta_i = \alpha_i / (\sum_{i=1}^{c} \alpha_i)$.

A final issue to be addressed is the error when $\Sigma_{\tilde{\alpha}}$ is also estimated with practical regularisation.

Shift in Flow Cytometry As we pointed out in the experiments in Chapter 5, the label shift hypothesis is not verified in flow cytometry. As pointed out by Tachet et al. [114], the label shift assumption is too strong: "label shift clearly fails in most practical applications". For this matter, they introduced *generalised label shift* a more general assumption, where they assume that there exists a function $g\colon \mathcal{X} \mapsto \mathcal{Z}$ such that for all classes $i : \mathbb{P}(g(x)|y = i) = \mathbb{Q}(g(x)|y = i)$. In other words, the label shift assumption is verified for at least one embedding function. The authors proposed to train a neural network where the first layers correspond to the embedding g, while simultaneously learning the proportions α. In flow cytometry, an ante-hoc approach such as CytoNorm [125] can be used to mitigate the *batch effect*, this procedure could mimic the function g of the *generalised label shit* assumption. A direct extension of our experiments would be to use such approaches before computing the vectorisations.

We could also investigate the modelisation of the shift with other assumptions than label shift, in particular *covariate shift*, a shift characterised by $\mathbb{P}(y|x) = \mathbb{Q}(y|x)$. This setting is adapted to flow cytometry because the gates are typically (but not always) obtained on one patient once and then apply to all the other patients. See Tasche [118, 119] for some results on quantification under covariate shift.

APPENDIX

Appendix A

Concentration inequality in Hilbert spaces

This appendix is devoted to the presentation of vector norm concentration inequalities in Hilbert spaces used throughout this thesis. The three theorems presented are a Hoeffding based inequality which is agnostic to the variances of the random variables, a Bernstein based inequality and finally a Bennett based inequality which is worse than the Bernstein one, but we present it for completeness. In particular this encompasses concentration inequalities of the approximation of the Kernel Mean Embedding (introduced in Section 1.2.3) of a distribution $\mathbb{P}$ with the Kernel Mean Embedding of an empirical distribution $\hat{\mathbb{P}} := \frac{1}{n}\sum_{k=1}^{n} k(X_i, \cdot)$, where $X_1, \cdots, X_n$ is an iid sample of $\mathbb{P}$. However, the inequalities we seek are more general than this case. One of the main contributions of Chapter 3 is to propose a general framework for quantification that involves embedding the data points $X_1, \cdots, X_n$ in a Hilbert space $\mathcal{H}$, which includes but is not limited to Kernel Mean Embedding. Furthermore, it is assumed that the tuple points/labels: (X_i, Y_i) are independent and identically distributed, but conditionally on the labels the points are independent but not iid because each point X_i comes from the distribution $\mathbb{P}_{Y_i}$. This distinction has its importance in the proofs, and so we look for theorems that assume independence but not equidistribution (i.e. not iid).

Throughout this appendix, we will refer to $Z_1, \ldots, Z_n$ as n independent but not necessarily identically distributed random variables taking values in a Hilbert space $\mathcal{H}$. In our case, $Z_i = \phi(X_i)$ where ϕ is an arbitrary mapping from $\mathcal{X}$ to $\mathcal{H}$. We assume that the random variables Z_i are bounded by a constant C. For instance, if ϕ is the implicit mapping of a kernel k, then $C = \sup_{x \in \mathcal{X}} \sqrt{k(x,x)}$ and if the kernel k is translation invariant on $\mathbb{R}^d$, i.e. $k(x,y) = \kappa(x-y)$ for some function κ, then $C = \kappa(0)$.

With these assumptions in mind, we want to find a real $\mathcal{B} := \mathcal{B}_{\delta,C}(Z_1, \dots, Z_n)$ such that, with probability greater than $1 - \delta$:

$$\left\| \frac{1}{n} \sum_{i=1}^{n} (Z_i - \mathbb{E}[Z_i]) \right\| \leq \mathcal{B}_{\delta,C}(Z_1, \dots, Z_n),$$

and we want both the asymptotic speed and the constants to be as small as possible.

A.1 Hoeffding-based inequality

The first methods used in the kernel literature to bound $\left\|\mu_{\mathbb{P}} - \mu_{\hat{\mathbb{P}}}\right\|_{\mathcal{H}}$ used vectorial variants of Hoeffding's inequality. These results have appeared in various versions in the literature (one of the earliest versions seems to be by Pinelis [92]) and are based on McDiarmid's inequality [80] which controls the difference between a function and its expectation as long as the function satisfies the *bounded differences property*.

In the kernel literature, authors did not apply McDiarmid's inequality directly, but first expressed the dual formulation of the norm $\left\|\mu_{\mathbb{P}} - \mu_{\hat{\mathbb{P}}}\right\|$, which is the sup-norm of an empirical process :

$$\left\|\mu_{\mathbb{P}} - \mu_{\hat{\mathbb{P}}}\right\| = \sup_{\|f\|_{\mathcal{H}} \leq 1} \left\langle f, \mu_{\mathbb{P}} - \mu_{\hat{\mathbb{P}}} \right\rangle = \sup_{\|f\|_{\mathcal{H}} \leq 1} \left(\mathbb{E}_{\mathbb{P}}[f(X)] - \frac{1}{n} \sum_{i=1}^{n} f(x_i) \right),$$

and applied McDiarmid here. To upper bound the expected value of this sup-norm, they used a symmetrization step and bounded it by twice the Rademacher complexity of the class $\{f \in \mathcal{H}, \|f\| \leq 1\}$ and then bounded the Rademacher complexity of this class using a result of Barlett et al. [6].

This kind of approach was proposed by Song [111] in his phd thesis (Theorem 27) and he obtained the bound $\mathcal{B} = \frac{2}{\sqrt{n}} + \sqrt{\log(2/\delta)/2n}$. Note that in his proof he assumed that $C = 1$ which is a standard assumption for kernels. Using, the same technique Lopez et al. [78] obtained a better bound with $\mathcal{B} = C\frac{(2+\sqrt{2\log(1/\delta)})}{\sqrt{n}}$ (see Theorem 1). To the best of our knowledge, this is the best bond achieved using the dual formulation of the norm. For completeness, note that Briot et al. [10] also used this technique to obtain a bound of the form $\mathcal{B} = C\frac{(\sqrt{2}+\sqrt{2\log(1/\delta)})}{\sqrt{n}}$ (see Lemma 1).

Another approach is to apply McDiarmid directly to the norm without using the dual formulation. For KME, this idea was used by Tolstikhin et al. [122] to obtain a bound of the form $\mathcal{B} = C\frac{(1+\sqrt{2\log(1/\delta)})}{\sqrt{n}}$. In their paper they assumed Z_i to be iid in a Hilbert space of real-valued functions but their results can be generalized to independent but not iid variables in any Hilbert space.

In the article derived from Chapter 3 of this phd (see the ArXiv version with the proofs [29]), we used the same argument but we obtained the same constant as in Lopez et al. [78]. We propose here a slightly improved proof that leads to better constants.

Theorem A.1. *Let $Z_1, \dots, Z_n$ be independent (not necessarily identically distributed) random variables taking values in a Hilbert space $\mathcal{H}$. Suppose that $\forall i \in [n] : \|Z_i\| \leq C < \infty$. Then with probability greater than $1 - \delta$:*

$$\left\| \frac{1}{n} \sum_{i=1}^{n} (Z_i - \mathbb{E}[Z_i]) \right\| \leq C \frac{(1 + \sqrt{2 \log(1/\delta)})}{\sqrt{n}},$$

Proof. Since $\|Z_i\| \leq C$, the random variables Z_i take values in $\mathcal{B}_C := \{z \in \mathcal{H} : \|z\| \leq C\}$. Define the function $F : (\mathcal{B}_C)^n \to \mathbb{R}$ as

$$F(z_1, \dots, z_n) := \left\| \frac{1}{n} \sum_{i=1}^{n} (z_i - \mathbb{E}[Z_i]) \right\|.$$

Straightforward computations show that the function F satisfies the *bounded difference condition*. Namely, let us fix all the values $z_1, \cdots, z_n$ in $\mathcal{B}_C$ except for the z_j which will be set to $\bar{z}_j$, we have

$$|F(z_1, \cdots, z_n) - F(z_1, \cdots, \bar{z}_j, \cdots z_n)| = \frac{1}{n} \|z_i - \bar{z}_i\| \leq \frac{2C}{n}.$$

Using McDiarmid's inequality, since Z_i are realisations of independent random variables taking values in $\mathcal{B}_C$, it holds with probability greater than $1 - \delta$:

$$\left\| \frac{1}{n} \sum_{i=1}^{n} (Z_i - \mathbb{E}[Z_i]) \right\| \leq \mathrm{E} \left[\left\| \frac{1}{n} \sum_{i=1}^{n} (Z_i - \mathbb{E}[Z_i]) \right\| \right] + C \sqrt{\frac{2 \log(1/\delta)}{n}}.$$

Let us write $\tilde{Z}_i = Z_i - \mathbb{E}[Z_i]$. The random variables $\tilde{Z}_i$ are independent, bounded in norm and centered.

By Jensen's inequality,

$$\begin{aligned}
\mathbb{E} \left[\left\| \frac{1}{n} \sum \tilde{Z}_i \right\| \right] &\leq \sqrt{\mathbb{E} \left[\left\| \frac{1}{n} \sum \tilde{Z}_i \right\|^2 \right]} \\
&= \frac{1}{n} \left(\sum_{i,j=1}^{n} \mathbb{E}[\langle \tilde{Z}_i, \tilde{Z}_j \rangle] \right)^{\frac{1}{2}} \\
&= \frac{1}{n} \left(\sum_{i=1}^{n} \mathbb{E}[\|\tilde{Z}_i\|^2] \right)^{\frac{1}{2}},
\end{aligned}$$

where we have used that since for $i \neq j$ the variables $\tilde{Z}_i, \tilde{Z}_j$ are independent and centered it holds $\mathbb{E}\left[\langle \tilde{Z}_i, \tilde{Z}_j \rangle\right] = 0$. We could simply bound $\mathbb{E}\Big[\|Z_i - \mathbb{E}[Z_i]\|^2\Big]$ by $2C^2$ but by using a simple argument found for instance in [22] or in [122], we can achieve a better bound:

$$\begin{aligned}
\mathbb{E}\Big[\big\|\tilde{Z}_i\big\|^2\Big] &= \mathbb{E}\Big[\|Z_i - \mathbb{E}[Z_i]\|^2\Big] \\
&= \mathbb{E}\Big[\|Z_i\|^2\Big] + \mathbb{E}\Big[\|\mathbb{E}[Z_i]\|^2\Big] - 2\mathbb{E}[\langle Z_i, \mathbb{E}[Z_i]\rangle] \\
&= \mathbb{E}\Big[\|Z_i\|^2\Big] - \|\mathbb{E}[Z_i]\|^2 \\
&\leq \mathbb{E}\Big[\|Z_i\|^2\Big] \leq C^2,
\end{aligned}$$

so that $\mathbb{E}\left[\left\|\frac{1}{n}\sum \tilde{Z}_i\right\|\right] \leq \frac{C}{\sqrt{n}}$. □

A.2 Bernstein-based inequality

To take into account the covariance matrices of the random variables, Wolfer et al. [127] used a concentration inequality based on a vectorial version of Bernstein. They assumed that Z_i were iid but their result can be extended to our setting and we propose here an alternative version of their work with our assumptions.

Their result is based on a Bernstein inequality in Hilbert space due to Yurinsky [129] (see Theorem 3.3.4). We propose here an alternative version where we use as a building block a theorem due to Pinelis [93] (Theorem 3.3), which is tighter and more general because it applies to martingales in Banach spaces that satisfy a smoothness condition. With this change, the difference between our version and Wolfer's one will be the constant before the term in $\mathcal{O}(1/n)$, we will have $2/3$ while they had $4/3$.

First let us state the theorem of Pinelis.

Theorem A.2 ([93]). *Let $f = (f_0, f_1, \cdots)$ be a martingale in a $(2, D)-$smooth separable Banach space. Let us note $f^* = \sup \|f_j\|$ and $\mathbb{E}_{j-1}$ the conditional expectation given the filtration F_{j-1}.*

Suppose that

$$\left\|\sum_{i=1}^{\infty} \mathbb{E}_{j-1}\|f_j - f_{j-1}\|^p\right\|_{\infty} \leq \frac{p!}{2D^2} B^2 H^{p-2}$$

for some $H > 0$, $B > 0$ and for all $p > 1$. Then, for all $r \geq 0$,

$$\mathbb{P}(f^* \geq r) \leq 2\exp\left(-\frac{r^2}{B^2 + B\sqrt{B^2 + 2Hr}}\right). \tag{A.1}$$

Suppose we have n centered independent random variables $(\xi_i)_{i=1}^n$ in a Hilbert space $\mathcal{H}$. Following the theorem, we propose the next corollary.

Corollary A.1. *Let $\mathcal{H}$ be a Hilbert space, and let (ξ_i) be n independent random variables (not necessarily identically distributed) with values in $\mathcal{H}$. Suppose that*

- $\mathbb{E}[\xi_i] = 0$ *for all i.*
- *There exists constants $B > 0$ and $H > 0$ such that for all $p \geq 2$:*

$$\sum_{i=1}^{n} \mathbb{E}[\|\xi_i\|^p] \leq \frac{p!}{2} B^2 H^{p-2}.$$

Then, for any $r > 0$,

$$\mathbb{P}\left(\max_{1 \leq s \leq n} \left\| \sum_{i=1}^{s} \xi_i \right\| \geq r \right) \leq 2 \exp\left(- \frac{r^2}{2B^2 + rH} \right). \tag{A.2}$$

Proof. We apply Theorem A.2.
In this context, the Banach space is now the Hilbert space $\mathcal{H}$, so that the smoothness constant D is equal to 1. The martingale is defined by $f_j = \sum_{i=1}^{j} \xi_i$ so that $f_j - f_{j-1} = \xi_j$ and $f^* = \max_{1 \leq s \leq n} \|\sum_{i=1}^{s} \xi_i\|$.

Using Equation (A.1):

$$\mathbb{P}\left(\max_{1 \leq s \leq n} \left\| \sum_{i=1}^{s} \xi_i \right\| \geq r \right) \leq 2 \exp\left(- \frac{r^2}{B^2 + B\sqrt{B^2 + 2Hr}} \right),$$

we then use $\sqrt{1+x} \leq 1 + 0.5x$ for $x \geq 0$:

$$\mathbb{P}\left(\max_{1 \leq s \leq n} \left\| \sum_{i=1}^{s} \xi_i \right\| \geq r \right) \leq 2 \exp\left(- \frac{r^2}{2B^2 + rH} \right).$$

□

Note that in Yurinsky [129] they had $2B^2 + 2rH$ instead of $2B^2 + rH$.
We can now state our version of Bernstein's inequality for vector-valued random variables.

Theorem A.3. *Let $Z_1, \ldots, Z_n$ be independent (not necessarily identically distributed) random variables taking values in a Hilbert space $\mathcal{H}$. Suppose that $\forall i \in [n] : \|Z_i\| \leq C < \infty$. Denote $\overline{\Sigma} := \frac{1}{n} \sum \Sigma_{Z_i}$, where Σ_{Z_i} is the covariance matrix of Z_i. Then with probability greater than $1 - \delta$:*

$$\left\| \frac{1}{n} \sum_{i=1}^{n} (Z_i - \mathbb{E}[Z_i]) \right\| \leq \frac{2}{3} \frac{C \log(2/\delta)}{n} + \sqrt{\frac{2 \log(2/\delta)}{n} Tr(\overline{\Sigma})}. \tag{A.3}$$

Proof. We will apply Corollary A.1 to the random variables $\xi_i = Z_i - \mathbb{E}[Z_i]$ who satisfy $\mathbb{E}[\xi_i] = 0$. First note that, $\|\xi_i\| \leq 2C$. For $p = 2$:

$$\sum_{i=1}^{n} \mathbb{E}[\|\xi_i\|^2] = \sum_{i=1}^{n} Tr(\Sigma_{z_i}) = nTr(\overline{\Sigma}).$$

While for $p \geq 3$:

$$\begin{aligned}\sum_{i=1}^{n} \mathbb{E}[\|\xi_i\|^p] &\leq (2C)^{p-2} \sum_{i=1}^{n} \mathbb{E}[\|\xi_i\|^2] \\ &= nTr(\overline{\Sigma})(2C)^{p-2}.\end{aligned}$$

This last inequality has not yet the form $\frac{p!}{2}B^2H^{p-2}$, a simple option would be to bound it by $\frac{p!}{2}nTr(\overline{\Sigma})(2C)^{p-2}$ and in that case H would be equal to $2C$ and we would obtain the same bound as in the Bennett-based inequality that we will present shortly after (Theorem A.5). Wolfer et al. [127] proposed to lower the gap between $nTr(\overline{\Sigma})(2C)^{p-2}$ and $\frac{p!}{2}nTr(\overline{\Sigma})(2C)^{p-2}$ as such:

$$\begin{aligned}\sum_{i=1}^{n} \mathbb{E}[\|\xi_i\|^p] &= nTr(\overline{\Sigma})(2C)^{p-2} \\ &\overset{\dagger}{\leq} \inf_a \frac{p!}{2} nTr(\overline{\Sigma}) \left(\frac{2C}{a}\right)^{p-2} \\ &= \frac{p!}{2} nTr(\overline{\Sigma}) \left(\frac{2C}{3}\right)^{p-2},\end{aligned}$$

where the inf in (†) is taken over all value a such that $1 \leq \frac{p!}{2a^{p-2}}$ for all $p \geq 2$. The minimum is obtained for $a = 3$. Hence, we have $B^2 = nTr(\overline{\Sigma})$ and $H = 2/3C$. A direct application of Corollary A.1 yield for any $r > 0$:

$$\mathbb{P}\left(\max_{1\leq s\leq n} \left\|\sum_{i=1}^{s} \xi_i\right\| \geq r\right) \leq 2\exp\left(-\frac{r^2}{2nTr(\overline{\Sigma}) + 2/3rC)}\right).$$

Hence, with probability greater than $1 - \delta$ we have:

$$\max_{1\leq s\leq n} s\left\|\frac{1}{s}\sum_{i=1}^{s}(Z_i - \mathbb{E}[Z_i])\right\| \leq \frac{2}{3}C\log(2/\delta) + \sqrt{2nTr(\overline{\Sigma})\log(2/\delta)},$$

which yield Equation (A.3) for the special case $s = n$. □

A.3 Bennett-based inequality

Another approach taken by Smale et al. [108] to take into account the covariances of the random variables, consists in using a vector-valued Bennett inequality instead of a Bernstein

one. Their result is based on a concentration result by Pinelis [93] on the norm of vector-valued martingales in certain "smooth" Banach spaces (Theorem 3.4). The resulting bound has the same form as the Bernstein one $(\mathrm{A}.3)$ except for the constant before the term in $\mathcal{O}\left(1/n\right)$. In Bernstein we had $4/3$, while in Bennett we have 4. Note that the reference [93] may not be the oldest one, as there exist previous works on the same subject by Penelis and Sahanenko written in Russian.

Although the bound is not as good, we propose to demonstrate the result here for completeness. Again, their result assumes that the Z_i are independent and identically distributed, but can be easily extended to our setting.

First let us state the theorem of Pinelis.

Theorem A.4 ([93]). *Let $f = (f_0, f_1, \cdots)$ be a martingale in a $(2, D)-$smooth separable Banach space. Let us note $d^* = \sup_j \|f_j - f_{j-1}\|$, $s_2^2 = \sum_{j=1}^{\infty} \mathbb{E}_{j-1}[\|d_j\|^2$ where $\mathbb{E}_{j-1}$ stands for the conditional expectation given the filtration F_{j-1} and $f^* = \sup \|f_j\|$.*

Let us suppose that $\|d^*\|_\infty \leq a$, $\|s_2\|_\infty \leq b/D$ for some $a, b > 0$.
Then for all $r \geq 0$,

$$\mathbb{P}(f^* \geq r) \leq 2\exp\left(\frac{r}{a} - \left(\frac{r}{a} + \frac{b^2}{a^2}\right)\log\left(1 + \frac{ra}{b^2}\right)\right).$$

In our context, the Banach space is a Hilbert space $\mathcal{H}$ so that the smoothness constant D is equal to 1. The martingale is defined by $f_j = \sum_{i=1}^{j}(Z_i - \mathbb{E}[Z_i])$ so that $d^* = \sup_j \|Z_j - \mathbb{E}[Z_j]\| \leq 2C$, and $s_2^2 = \sum_{i=1}^{j} \mathbb{E}\left[\|Z_i - \mathbb{E}[Z_i]\|^2\right] = nTr(\overline{\Sigma})$.

The following theorem is a restatement of Smale's results (Lemmas 1 and 2).

Theorem A.5 ([108]). *Let $Z_1, \ldots, Z_n$ be independent (not necessarily identically distributed) random variables taking values in a Hilbert space $\mathcal{H}$. Suppose that $\forall i \in [n]$: $\|Z_i\| \leq C < \infty$. Denote $\overline{\Sigma} \coloneqq \frac{1}{n}\sum \Sigma_{Z_i}$, where Σ_{Z_i} is the covariance matrix of Z_i. Then,*

$$\mathbb{P}\left\{\sup_{i=1,\cdots,s}\left\|\frac{1}{s}\sum_{i=1}^{s}(Z_i - \mathbb{E}[Z_i])\right\| \geq \varepsilon\right\} \leq 2\exp\left\{-\frac{n\varepsilon}{4C}\log\left(1 + \frac{2C\varepsilon}{Tr(\overline{\Sigma})}\right)\right\}. \tag{A.4}$$

In particular, for any $0 < \delta < 1$, with confidence $1 - \delta$:

$$\left\|\frac{1}{n}\sum_{i=1}^{n}(Z_i - \mathbb{E}[Z_i])\right\| \leq 4\frac{C\log(2/\delta)}{n} + \sqrt{\frac{2\log(2/\delta)}{n}Tr(\overline{\Sigma})}, \tag{A.5}$$

The proof follows the approach of Smale et al. [108] in section 3. The first inequality $(\mathrm{A}.4)$ is derived from Pinelis' theorem and the second inequality $(\mathrm{A}.5)$ is simply a restatement of the first.

Proof. Let us apply Theorem A.4 with $r = n\varepsilon$, $a = 2C$ and $b^2 = nTr(\overline{\Sigma})$.

We have,

$$\begin{aligned}\frac{r}{a} - \left(\frac{r}{a} + \frac{b^2}{a^2}\right)\log\Big(1 + \frac{ra}{b^2}\Big) &= -\frac{b^2}{a^2}\Big[-\frac{ra}{b^2} + (1 + \frac{ra}{b^2})\log\Big(1 + \frac{ra}{b^2}\Big)\Big] \\ &= -\frac{b^2}{a^2}\ h\Big(\frac{ra}{b^2}\Big) \\ &\leq -\frac{r}{2a}\log\Big(1 + \frac{ra}{b^2}\Big)\end{aligned}$$

Where $h(t) = (1+t)\log(1+t) - t$. In the last line, we use $h(t) \geq \frac{t}{2}\log(1+t)$.

The rest of the proof is a direct application of Equation (A.4) and the inequality

$$\forall x \geq 0, \quad \log(1+x) \geq \frac{x}{1+x}.$$

Using both arguments we have:

$$\begin{aligned}\mathbb{P}\left\{\left\|\frac{1}{n}\sum_{i=1}^{n}(\xi_i - \mathbb{E}[\xi])\right\| \geq \varepsilon\right\} &\leq 2\exp\left\{-\frac{n\varepsilon}{4C}\log\left(1 + \frac{2C\varepsilon}{Tr(\overline{\Sigma})}\right)\right\} \\ &\leq 2\exp\left\{-\frac{n\varepsilon^2}{4C\varepsilon + 2Tr(\overline{\Sigma})}\right\}\end{aligned}$$

For a fixed confidence value δ, we search for the ε such that $-n\varepsilon^2 = \log(\delta/2)(4C\varepsilon + 2Tr(\overline{\Sigma}))$. Solving this quadratic equation yield:

$$\varepsilon \geq 4\frac{C\log(2/\delta)}{n} + \sqrt{\frac{2\log(2/\delta)}{n}Tr(\overline{\Sigma})}.$$

□

BIBLIOGRAPHY

Bibliography

[1] Ya I Alber and AI Notik. On some estimates for projection operator in Banach space. *arXiv preprint funct-an/9311003*, 1993.

[2] Amr Alexandari, Anshul Kundaje, and Avanti Shrikumar. Maximum likelihood with bias-corrected calibration is hard-to-beat at label shift adaptation. In *International Conference on Machine Learning*, pages 222–232. PMLR, 2020.

[3] Valentin Amrhein, Sander Greenland, and Blake McShane. Scientists rise up against statistical significance. *Nature*, 567(7748):305–307, 2019.

[4] Kamyar Azizzadenesheli, Anqi Liu, Fanny Yang, and Animashree Anandkumar. Regularized learning for domain adaptation under label shifts. *arXiv preprint arXiv:1903.09734*, 2019.

[5] Jose Barranquero, Jorge Díez, and Juan José del Coz. Quantification-oriented learning based on reliable classifiers. *Pattern Recognition*, 48(2):591–604, 2015.

[6] Peter L Bartlett and Shahar Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[7] Antonio Bella, Cesar Ferri, José Hernández-Orallo, and Maria Jose Ramirez-Quintana. Quantification via probability estimators. In *2010 IEEE International Conference on Data Mining*, pages 737–742. IEEE, 2010.

[8] AAT Bioquest. Fundamentals of flow cytometry.

[9] Javier Borge-Holthoefer, Walid Magdy, Kareem Darwish, and Ingmar Weber. Content and network dynamics behind egyptian political polarization on twitter. In *Proceedings of the 18th ACM conference on computer supported cooperative work & social computing*, pages 700–711, 2015.

[10] Francois-Xavier Briol, Alessandro Barp, Andrew B Duncan, and Mark Girolami. Statistical inference for generative models with maximum mean discrepancy. *arXiv preprint arXiv:1906.05944*, 2019.

[11] AA Buck, JJ Gart, et al. Comparison of a screening test and a reference test in epidemiologic studies. ii. A probabilistic model for the comparison of diagnostic tests. *American Journal of Epidemiology*, 83(3):593–602, 1966.

[12] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural networks*, 106:249–259, 2018.

[13] Mirko Bunse. Unification of algorithms for quantification and unfolding. *INFORMATIK 2022*, 2022.

[14] Mirko Bunse. Qunfold: Composable quantification and unfolding methods in python. In *Proceedings of the 3rd International Workshop on Learning to Quantify (LQ 2023)*, pages 1–7, 2023.

[15] Mirko Bunse, Pablo González, Alejandro Moreo, and Fabrizio Sebastiani. Report on the 3rd international workshop on learning to quantify (LQ 2023. *ACM SIGKDD Explorations Newsletter*, 2023.

[16] Raffaello Camoriano, Tomás Angles, Alessandro Rudi, and Lorenzo Rosasco. Nytro: When subsampling meets early stopping. In *Artificial Intelligence and Statistics*, pages 1403–1411. PMLR, 2016.

[17] Alberto Castaño, Jaime Alonso, Pablo González, and Juan José del Coz. An equivalence analysis of binary quantification methods. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 6944–6952, 2023.

[18] Alberto Castaño, Jaime Alonso, Pablo González, Pablo Pérez, and Juan José del Coz. QuantificationLib: A python library for quantification and prevalence estimation. *SoftwareX*, 26:101728, 2024.

[19] Benjamin Charlier, Jean Feydy, Joan Alexis Glaunès, François-David Collin, and Ghislain Durif. Kernel operations on the GPU, with autodiff, without memory overflows. *Journal of Machine Learning Research*, 22(74):1–6, 2021. `https://www.kernel-operations.io/keops/index.html`.

[20] Frédéric Chazal, Leonidas J Guibas, Steve Y Oudot, and Primoz Skraba. Persistence-based clustering in riemannian manifolds. *Journal of the ACM (JACM)*, 60(6):1–38, 2013.

[21] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM.

[22] Badr-Eddine Chérief-Abdellatif and Pierre Alquier. Finite sample properties of parametric MMD estimation: robustness to misspecification and dependence. *Bernoulli*, 28(1):181–213, 2022.

[23] Melissa Cheung, Jonathan J Campbell, Liam Whitby, Robert J Thomas, Julian Braybrook, and Jon Petzing. Current trends in flow cytometry automated data analysis software. *Cytometry Part A*, 99(10):1007–1021, 2021.

[24] Andrew J Cowan, Damian J Green, Mary Kwok, Sarah Lee, David G Coffey, Leona A Holmberg, Sherilyn Tuazon, Ajay K Gopal, and Edward N Libby. Diagnosis and management of multiple myeloma: A review. *Jama*, 327(5):464–477, 2022.

[25] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

[26] Ashlynn R Daughton and Michael J Paul. Constructing accurate confidence intervals when aggregating social media data for public health monitoring. *Precision Health and Medicine: A Digital Revolution in Healthcare*, pages 9–17, 2020.

[27] Juan José del Coz, Pablo González, Alejandro Moreo, and Fabrizio Sebastiani. Report on the 1st international workshop on learning to quantify (LQ 2021). *ACM SIGKDD Explorations Newsletter*, 24(1):49–51, 2022.

[28] Zahra Donyavi, Adriane BS Serapiao, and Gustavo Batista. MC-SQ and MC-MQ: Ensembles for multi-class quantification. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[29] Bastien Dussap, Gilles Blanchard, and Badr-Eddine Chérief-Abdellatif. Label shift quantification with robustness guarantees via distribution feature matching. *arXiv preprint arXiv:2306.04376*, 2023.

[30] Bastien Dussap, Gilles Blanchard, and Badr-Eddine Chérief-Abdellatif. Label shift quantification with robustness guarantees via distribution feature matching. In *Machine Learning and Knowledge Discovery in Databases: Research Track*, pages 69–85. Springer Nature Switzerland, 2023.

[31] Moulines Eric, Francis Bach, and Zaïd Harchaoui. Testing for homogeneity with kernel Fisher discriminant analysis. *Advances in Neural Information Processing Systems*, 20, 2007.

[32] Andrea Esuli, Alessandro Fabris, Alejandro Moreo, and Fabrizio Sebastiani. *Learning to Quantify*, volume 47. Springer Nature, 2023.

[33] Andrea Esuli, Alessio Molinari, and Fabrizio Sebastiani. A critical reassessment of the Saerens-Latinne-Decaestecker algorithm for posterior probability adjustment. *ACM Transactions on Information Systems (TOIS)*, 39(2):1–34, 2020.

[34] Andrea Esuli, Alejandro Moreo, Fabrizio Sebastiani, and Gianluca Sperduti. A detailed overview of LeQua 2022: Learning to quantify. In *Working Notes of the 13th Conference and Labs of the Evaluation Forum (CLEF 2022), Bologna, IT*, 2022.

[35] Andrea Esuli, Alejandro Moreo, Fabrizio Sebastiani, and Gianluca Sperduti. LeQua 2022: A lab on learning to quantify @ CLEF2022, 2022.

[36] Andrea Esuli, Alejandro Moreo, Fabrizio Sebastiani, and Gianluca Sperduti. LeQua 2024: The 2nd data challenge on learning to quantify, 2024.

[37] Andrea Esuli, Alejandro Moreo Fernández, and Fabrizio Sebastiani. A recurrent neural network for sentiment quantification. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1775–1778, 2018.

[38] Andrea Esuli and Fabrizio Sebastiani. Optimizing text quantifiers for multivariate loss functions. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 9(4):1–27, 2015.

[39] Andrea Esuli, Fabrizio Sebastiani, and Ahmed Abbasi. Sentiment quantification. *IEEE Intell. Syst.*, 25(4):72–75, 2010.

[40] Alessandro Fabris, Andrea Esuli, Alejandro Moreo, and Fabrizio Sebastiani. Measuring fairness under unawareness of sensitive attributes: A quantification-based approach. *Journal of Artificial Intelligence Research*, 76:1117–1180, 2023.

[41] Tom Fawcett and Peter A Flach. A response to Webb and Ting's on the application of ROC analysis to predict classification performance under varying class distributions. *Machine Learning*, 58:33–38, 2005.

[42] Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C Prati, Bartosz Krawczyk, Francisco Herrera, Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C Prati, et al. Cost-sensitive learning. *Learning from imbalanced data sets*, pages 63–78, 2018.

[43] Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouvé, and Gabriel Peyré. Interpolating between optimal transport and MMD using sinkhorn divergences. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2681–2690. PMLR, 2019.

[44] Aykut Firat. Unified framework for quantification. *arXiv preprint arXiv:1606.00868*, 2016.

[45] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[46] George Forman. Counting positives accurately despite inaccurate classification. In *European conference on machine learning*, pages 564–575. Springer, 2005.

[47] George Forman. Quantifying trends accurately despite classifier error and class imbalance. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 157–166, 2006.

[48] George Forman. Quantifying counts and costs via classification. *Data Mining and Knowledge Discovery*, 17:164–206, 2008.

[49] Paul Freulon, Jérémie Bigot, and Boris P Hejblum. CytOpt: Optimal transport with domain adaptation for interpreting flow cytometry data. *The Annals of Applied Statistics*, 17(2):1086–1104, 2023.

[50] Wei Gao and Fabrizio Sebastiani. From classification to quantification in Tweet sentiment analysis. *Social Network Analysis and Mining*, 6:1–22, 2016.

[51] Saurabh Garg, Sivaraman Balakrishnan, and Zachary Lipton. Domain adaptation under open set label shift. *Advances in Neural Information Processing Systems*, 35:22531–22546, 2022.

[52] Saurabh Garg, Yifan Wu, Sivaraman Balakrishnan, and Zachary Lipton. A unified view of label shift estimation. *Advances in Neural Information Processing Systems*, 33:3290–3300, 2020.

[53] Solenne Gaucher, Gilles Blanchard, and Frédéric Chazal. Supervised contamination detection, with flow cytometry application. *arXiv preprint arXiv:2404.06093*, 2024.

[54] Benyamin Ghojogh, Fakhri Karray, and Mark Crowley. Fisher and kernel fisher discriminant analysis: Tutorial. *arXiv preprint arXiv:1906.09436*, 2019.

[55] Cyprien Gilet, Susana Barbosa, and Lionel Fillatre. Discrete box-constrained minimax classifier for uncertain and imbalanced class proportions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6):2923–2937, 2020.

[56] Cyprien Gilet, Marie Guyomard, Sébastien Destercke, and Lionel Fillatre. Softmin discrete minimax classifier for imbalanced classes and prior probability shifts. *Machine Learning*, 113(2):605–645, 2024.

[57] Pablo González, Eva Álvarez, Jorge Díez, Ángel López-Urrutia, and Juan José del Coz. Validation methods for plankton image classification systems. *Limnology and Oceanography: Methods*, 15(3):221–237, 2017.

[58] Pablo González, Alberto Castaño, Nitesh V Chawla, and Juan José Del Coz. A review on quantification learning. *ACM Computing Surveys (CSUR)*, 50(5):1–40, 2017.

[59] Víctor González-Castro, Rocío Alaiz-Rodríguez, and Enrique Alegre. Class distribution estimation based on the Hellinger distance. *Information Sciences*, 218:146–164, 2013.

[60] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. Ieee, 2013.

[61] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample-problem. *Advances in neural information processing systems*, 19, 2006.

[62] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[63] Arthur Gretton, Alex Smola, Jiayuan Huang, Marcel Schmittfull, Karsten Borgwardt, and Bernhard Schölkopf. Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3(4):5, 2009.

[64] Gregory Gundersen. Random Fourier Features. `https://gregorygundersen.com/blog/2019/12/23/random-fourier-features`, 2019.

[65] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.

[66] Omar Hagrass, Bharath K Sriperumbudur, and Bing Li. Spectral regularized kernel two-sample tests. *arXiv preprint arXiv:2212.09201*, 2022.

[67] Zaid Harchaoui, Félicien Vallet, Alexandre Lung-Yut-Fong, and Olivier Cappé. A regularized kernel-based approach to unsupervised audio segmentation. In *2009 IEEE international conference on acoustics, speech and signal processing*, pages 1665–1668. IEEE, 2009.

[68] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

[69] Daniel J Hopkins and Gary King. A method of automated nonparametric content analysis for social science. *American Journal of Political Science*, 54(1):229–247, 2010.

[70] Arun Iyer, Saketha Nath, and Sunita Sarawagi. Maximum mean discrepancy for class ratio estimation: Convergence bounds and kernel selection. In *International Conference on Machine Learning*, pages 530–538. PMLR, 2014.

[71] Del Coz J. J, González P., Moreo A., and Sebastiani F. Proceedings of the 2nd international workshop on learning to quantify (LQ 2022). 2022.

[72] Thorsten Joachims. A support vector method for multivariate performance measures. In *Proceedings of the 22nd international conference on Machine learning*, pages 377–384, 2005.

[73] Hideko Kawakubo, Marthinus Christoffel Du Plessis, and Masashi Sugiyama. Computationally efficient class-prior estimation under class balance change using energy distance. *IEICE TRANSACTIONS on Information and Systems*, 99(1):176–186, 2016.

[74] Robert A Kyle and S Vincent Rajkumar. ASH 50th anniversary review. *Blood*, 111(6):2962–2972, 2008.

[75] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*, pages 3744–3753. PMLR, 2019.

[76] Erich Leo Lehmann, Joseph P Romano, and George Casella. *Testing statistical hypotheses*, volume 3. Springer, 1986.

[77] Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. Detecting and correcting for label shift with black box predictors. In *International conference on machine learning*, pages 3122–3130. PMLR, 2018.

[78] David Lopez-Paz, Krikamol Muandet, Bernhard Schölkopf, and Iliya Tolstikhin. Towards a learning theory of cause-effect inference. In *International Conference on Machine Learning*, pages 1452–1461. PMLR, 2015.

[79] André Maletzke, Denis dos Reis, Everton Cherman, and Gustavo Batista. DyS: A framework for mixture models in quantification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4552–4560, 2019.

[80] Colin McDiarmid et al. On the method of bounded differences. *Surveys in combinatorics*, 141(1):148–188, 1989.

[81] Katherine M McKinnon. Flow cytometry: An overview. *Current protocols in immunology*, 120(1):5–1, 2018.

[82] Letizia Milli, Anna Monreale, Giulio Rossetti, Fosca Giannotti, Dino Pedreschi, and Fabrizio Sebastiani. Quantification trees. In *2013 IEEE 13th International Conference on Data Mining*, pages 528–536. IEEE, 2013.

[83] Alejandro Moreo, Andrea Esuli, and Fabrizio Sebastiani. QuaPy: a python-based framework for quantification. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 4534–4543, 2021.

[84] Alejandro Moreo, Manuel Francisco, and Fabrizio Sebastiani. Multi-label quantification. *ACM Transactions on Knowledge Discovery from Data*, 18(1):1–36, 2023.

[85] Alejandro Moreo, Pablo González, and Juan José del Coz. Kernel density estimation for multiclass quantification. *arXiv preprint arXiv:2401.00490*, 2023.

[86] Alejandro Moreo and Fabrizio Sebastiani. Tweet sentiment quantification: An experimental re-evaluation. *Plos one*, 17(9):e0263449, 2022.

[87] Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, Bernhard Schölkopf, et al. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning*, 10(1-2):1–141, 2017.

[88] Harikrishna Narasimhan, Purushottam Kar, and Prateek Jain. Optimizing non-decomposable performance measures: A tale of two classes. In *International Conference on Machine Learning*, pages 199–208. PMLR, 2015.

[89] Pablo Pérez-Gállego, Alberto Castano, José Ramón Quevedo, and Juan José del Coz. Dynamic ensemble selection for quantification tasks. *Information Fusion*, 45:1–15, 2019.

[90] Pablo Pérez-Gállego, José Ramón Quevedo, and Juan José del Coz. Using ensembles for problems with characterizable changes in data distribution: A case study on quantification. *Information Fusion*, 34:87–100, 2017.

[91] Marco AF Pimentel, David A Clifton, Lei Clifton, and Lionel Tarassenko. A review of novelty detection. *Signal processing*, 99:215–249, 2014.

[92] Iosif Pinelis. An approach to inequalities for the distributions of infinite-dimensional martingales. In *Probability in Banach Spaces, 8: Proceedings of the Eighth International Conference*, pages 128–134. Springer, 1992.

[93] Iosif Pinelis. Optimum bounds for the distributions of martingales in Banach spaces. *The Annals of Probability*, pages 1679–1706, 1994.

[94] Aleksandr Podkopaev and Aaditya Ramdas. Distribution-free uncertainty quantification for classification under label shift. In *Uncertainty in Artificial Intelligence*, pages 844–853. PMLR, 2021.

[95] Olaya Pérez-Mon, Alejandro Moreo, Juan José del Coz, and Pablo González. Quantification using permutation-invariant networks based on histograms, 2024.

[96] Lei Qi, Mohammed Khaleel, Wallapak Tavanapong, Adisak Sukul, and David Peterson. A framework for deep quantification learning. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part I*, pages 232–248. Springer, 2021.

[97] Joaquin Quinonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. Mit Press, 2008.

[98] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007.

[99] Aaditya Ramdas, Nicolás García Trillos, and Marco Cuturi. On Wasserstein two-sample testing and related families of nonparametric tests. *Entropy*, 19(2):47, 2017.

[100] Alessandro Rudi, Raffaello Camoriano, and Lorenzo Rosasco. Less is more: Nyström computational regularization. *Advances in Neural Information Processing Systems*, 28, 2015.

[101] Marco Saerens, Patrice Latinne, and Christine Decaestecker. Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure. *Neural computation*, 14(1):21–41, 2002.

[102] Amartya Sanyal, Pawan Kumar, Purushottam Kar, Sanjay Chawla, and Fabrizio Sebastiani. Optimizing non-decomposable measures with deep networks. *Machine Learning*, 107:1597–1620, 2018.

[103] Tobias Schumacher, Markus Strohmaier, and Florian Lemmerich. A comparative evaluation of quantification methods. *arXiv preprint arXiv:2103.03223*, 2021.

[104] Fabrizio Sebastiani. Evaluation measures for quantification: An axiomatic approach. *Information Retrieval Journal*, 23(3):255–288, 2020.

[105] Dino Sejdinovic, Bharath Sriperumbudur, Arthur Gretton, and Kenji Fukumizu. Equivalence of distance-based and RKHS-based statistics in hypothesis testing. *The annals of statistics*, pages 2263–2291, 2013.

[106] Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3), 2008.

[107] Telmo Silva Filho, Hao Song, Miquel Perello-Nieto, Raul Santos-Rodriguez, Meelis Kull, and Peter Flach. Classifier calibration: a survey on how to assess and improve predicted class probabilities. *Machine Learning*, 112(9):3211–3260, 2023.

[108] Steve Smale and Ding-Xuan Zhou. Learning theory estimates via integral operators and their approximations. *Constructive approximation*, 26(2):153–172, 2007.

[109] Alex Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. A Hilbert space embedding for distributions. In *International conference on algorithmic learning theory*, pages 13–31. Springer, 2007.

[110] Canadian Cancer Society. The plasma cells. `https://cancer.ca/en/cancer-information/cancer-types/multiple-myeloma/what-is-multiple-myeloma/the-plasma-cells`.

[111] Le Song. Learning via hilbert space embedding of distributions. *University of Sydney (2008)*, 17, 2008.

[112] Amos Storkey. When training and test sets are different: characterizing learning transfer. 2008.

[113] Danica J. Sutherland and Jeff Schneider. On the Error of Random Fourier Features. *arXiv:1506.02785 [cs, stat]*, 2015.

[114] Remi Tachet des Combes, Han Zhao, Yu-Xiang Wang, and Geoffrey J Gordon. Domain adaptation with conditional distribution matching and generalized label shift. *Advances in Neural Information Processing Systems*, 33:19276–19289, 2020.

[115] Dirk Tasche. Exact fit of simple finite mixture models. *Journal of Risk and Financial Management*, 7(4):150–164, 2014.

[116] Dirk Tasche. Fisher consistency for prior probability shift. *The Journal of Machine Learning Research*, 18(1):3338–3369, 2017.

[117] Dirk Tasche. Minimising quantifier variance under prior probability shift. *arXiv preprint arXiv:2107.08209*, 2021.

[118] Dirk Tasche. Class prior estimation under covariate shift: No problem. *arXiv preprint arXiv:2206.02449*, 2022.

[119] Dirk Tasche. Invariance assumptions for class distribution estimation. *arXiv preprint arXiv:2311.17225*, 2023.

[120] Junjiao Tian, Yen-Cheng Liu, Nathaniel Glaser, Yen-Chang Hsu, and Zsolt Kira. Posterior re-calibration for imbalanced datasets. *Advances in Neural Information Processing Systems*, 33:8101–8113, 2020.

[121] Ryan J Tibshirani, Rina Foygel Barber, Emmanuel Candes, and Aaditya Ramdas. Conformal prediction under covariate shift. *Advances in neural information processing systems*, 32, 2019.

[122] Ilya Tolstikhin, Bharath K Sriperumbudur, Krikamol Mu, et al. Minimax estimation of kernel mean embeddings. *Journal of Machine Learning Research*, 18(86):1–47, 2017.

[123] Juozas Vaicenavicius, David Widmann, Carl Andersson, Fredrik Lindsten, Jacob Roll, and Thomas Schön. Evaluating model calibration in classification. In *The 22nd international conference on artificial intelligence and statistics*, pages 3459–3467. PMLR, 2019.

[124] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11), 2008.

[125] Sofie Van Gassen, Brice Gaudilliere, Martin S Angst, Yvan Saeys, and Nima Aghaeepour. CytoNorm: a normalization algorithm for cytometry data. *Cytometry Part A*, 97(3):268–278, 2020.

[126] Afonso Fernandes Vaz, Rafael Izbicki, and Rafael Bassi Stern. Quantification under prior probability shift: The ratio estimator and its extensions. *Journal of Machine Learning Research*, 20(79):1–33, 2019.

[127] Geoffrey Wolfer and Pierre Alquier. Variance-aware estimation of kernel mean embedding. *arXiv preprint arXiv:2210.06672*, 2022.

[128] Jack Chongjie Xue and Gary M Weiss. Quantification and semi-supervised classification methods for handling changes in class distribution. In *Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 897–906, 2009.

[129] Vadim Yurinsky. *Sums and Gaussian vectors*. Springer, 2006.

[130] Ibrahim Yusuf, George Igwegbe, and Oluwafemi Azeez. Differentiable histogram with hard-binning. *arXiv preprint arXiv:2012.06311*, 2020.

[131] Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699, 2002.

[132] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

[133] Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. Domain adaptation under target and conditional shift. In *International conference on machine learning*, pages 819–827. PMLR, 2013.

List of Figures

List of Tables

Pour Marilyn.