# Nouvelles approches itératives avec garanties théoriques pour l'adaptation de domaine non supervisée

Jean-Philippe Peyrache

École Doctorale ED488 Sciences, Ingénierie, Santé


# Nouvelles Approches Itératives avec Garanties Théoriques pour l'Adaptation de Domaine Non Supervisée

par

**Jean-Philippe Peyrache**

---

Thèse préparée pour obtenir le grade de :

**Docteur de l'Université Jean Monnet de Saint-Étienne**

Spécialité : **Informatique**

Laboratoire Hubert Curien, UMR CNRS 5516

Faculté des Sciences et Techniques

---

**Soutenance le vendredi 11 juillet 2014 devant le jury composé de :**

| | | |
|---|---|---|
| Antoine Cornuéjols | **Professeur** à AgroParisTech | **Rapporteur** |
| Élisa Fromont | **Maître de Conférences** à l'Université de Saint-Étienne | **Examinateur** |
| Amaury Habrard | **Professeur** à l'Université de Saint-Étienne | **Co-directeur** |
| Jean-Christophe Janodet | **Professeur** à l'Université d'Évry | **Rapporteur** |
| Mikaela Keller | **Maître de Conférences** à l'Université de Lille III | **Examinateur** |
| Marc Sebban | **Professeur** à l'Université de Saint-Étienne | **Directeur** |

# Remerciements

Voici donc venu le moment si particulier des remerciements, qui sonne comme le point final de ces quatre années.

Je tiens en premier lieu à remercier Antoine Cornuéjols, Professeur à AgroParisTech et Jean-Christophe Janodet, Professeur à l'Université d'Évry que je retrouve presque 10 ans après les cours de Caml, d'avoir accepté de rapporter mon travail de thèse. Je leur suis reconnaissant de l'intérêt porté à ce manuscrit, se traduisant par des remarques pertinentes et des pistes évoquées, me permettant une prise de recul plus importante sur ces travaux. Merci également à Élisa Fromont, Maître de Conférences à l'Université de Saint-Étienne et Mikaela Keller, Maître de Conférences à l'Université de Lille III, toutes deux aussi sympathiques que compétentes, d'avoir accepté d'être examinatrices de ce travail.

Dans un deuxième temps, je remercie mes deux co-directeurs de thèse, Amaury Habrard et Marc Sebban pour la qualité de leur suivi et leur compréhensivité. Merci à eux de m'avoir guidé dans le monde de la recherche, j'ai beaucoup appris à leurs côtés. Je suis extrêmement reconnaissant de la confiance qu'ils m'ont accordée et je n'hésiterai pas à me tourner vers eux dans le futur.

Je remercie également tous les collègues du Laboratoire Hubert Curien, passés ou présents, qui ont rendu l'atmosphère plus agréable, aussi studieuse ait-elle été. Merci donc à tous ceux qui ont partagé un jour mon bureau : en premier lieu mes amis du "foyer", Aurélien, avec qui il fut aussi plaisant de partager des discussions au bord de la piscine à Boca Raton que des "Sunset celebrations" à Key West, Fabien, qui m'a notamment prêté ses livres de Paul Jorion et avec qui je me régale à refaire le monde politique et Mattias, que j'apprécie toujours autant de retrouver sur un terrain de football et dont la barbe n'a d'égal que sa réactivité lorsqu'il s'agit de répondre aux mails. Merci à vous trois pour ces tranches de rigolade ou ces échanges divers aussi bien dans le bureau que sur la passerelle. On le savait bien que l'on conquerrait la fac !

Merci aussi à tous les autres occupants de ce bureau : Chahrazed, ma collègue depuis le master à qui je souhaite de terminer rapidement, Christophe M., dont les repas me faisaient régulièrement envie, malgré des quantités bien trop réduites, David C., qui a parfois eu l'air d'avoir du mal à nous supporter (ça peut se comprendre), Émilie M., avec qui j'ai partagé une thématique de recherche et un co-directeur, et que j'ai eu le plaisir de croiser plusieurs fois en conférence (je ne me

permettrai pas d'évoquer l'enregistrement du taxi...), Émilie S., malgré son aversion pour la viande de lapin (c'est que tu n'as jamais goûté !), Jan, le suédois éphémère, Laurent B., que je remercie pour son accueil à Strasbourg (quelle soirée !), Michaël, à qui je souhaite plein de *Metric Learning*, Reda et ses corrections express, Tung, à l'énergie et à la gentillesse débordantes, et Valentina, qui a passé haut la main le test du tournoi de football (qui je l'espère perdurera après mon départ), nous permettant de décrocher cette si attendue deuxième place.

Je tiens aussi à remercier les collègues du couloir ou d'ailleurs : Baptiste J. et sa passion du badminton (j'aurai peut-être une chance quand tu approcheras de la retraite), Catherine et son talent pour la réalisation de l'affiche et du logo de CAp', Christine, toujours pleine d'énergie, Élisa, au sourire permanent et communicatif (et dont la passion pour les gâteaux n'est plus à démontrer), Fabrice M., qui nous a tant tannés pour que nous lui fournissions des questions en *Outils Logiciels*, François, qui j'espère me pardonnera pour mon projet de compilation bâclé en L3, Leo, au rire et à l'accent inimitables, Marc B. et sa passion pour la photographie et le vin, Mathias, "la machine", qui m'a si souvent motivé (parfois forcé) à aller courir (je salue au passage les acolytes de la course à pied Baptiste M., Christophe H., Damien M. et Saïd), Philippe pour le vélo qu'il m'a vendu à prix cassé et avec lequel j'ai connu mes premières émotions de cycliste, Pierre aux réflexes de gardien de but étonnants, Rémi E., dernier arrivé mais déjà initiateur du tournoi de pétanque, Richard, le forcené de la natation et Thierry M., l'homme aux compétences multiples, sans oublier Alain, Christophe D., Éric, Loïc D. et Olivier A., ainsi que Claude dont les histoires auront égayé certaines de mes journées et Fabrice A. avec qui j'ai eu de nombreuses discussions au cours de nos repas de midi.

Merci également à toutes les autres personnes du laboratoire, qu'elles soient personnels administratifs ou techniques, doctorants ou enseignants-chercheurs, que l'on ait partagé un café, une discussion ou un concours de pronostics. Et merci à tous ceux que j'ai croisés pendant une école d'été à Copenhague, au cours d'une conférence à Barcelone, Chambéry, Boca Raton, Nancy ou Prague ou durant un workshop à Cassel, Saint-Victor-sur-Loire, Lille ou Porquerolles, qu'ils aient été doctorants à Marseille (j'ai une pensée particulière pour Guillaume R. avec qui j'ai partagé quelques bières à Prague, Pierre et son langage si fleuri, que j'ai découvert sous son meilleur jour en Californie, Sokol et Thomas P.) ou Paris (comme ce bon vieux Gabriel Dulac-Arnold, sosie et presque anagramme d'Arnaud Delubac que je salue au passage), professeurs ou maîtres de conférences à Lille, Marseille, Paris ou ailleurs. Merci notamment à Liva, Ludovic, Marc T., Patrick, Rémi G. et les autres pour leurs remarques et questions pertinentes à l'occasion de certaines de mes présentations à LAMPADA, ainsi que pour leur bonne humeur (à quand le projet PROSTAT ?). Enfin, merci à ces inconnus d'un soir (ou de plusieurs), recontrés dans un bar ou une auberge de jeunesse, qui font toute la richesse et le plaisir d'un voyage, aussi professionnel soit-il.

Ces remerciements ne seraient pas complets sans une affectueuse pensée pour les amis que j'ai côtoyés tout au long de ces annés et qui ont rendu ma vie plus gaie. Merci donc à tous les amis des "années fac", en premier lieu desquels le "foyer" : Antoine, son air de viking et son goût prononcé pour

le chocolat chaud (j'en profite pour saluer Julie), Arnaud, qui m'a fait découvrir (entre autres) le plaisir de randonner en montagne, Aurélie, devenue une véritable aventurière depuis son expérience canadienne, Benoit M. qui devrait s'acheter des montres plutôt que d'installer des configurations pour CoD, Bertrand, le néo-belge futur médecin, Cécile, qui assiste à toutes les soutenances (un petit coucou à Angela et Roxanne au passage), Clément M., qui nous est revenu épanoui de son aventure néo-zélandaise (comment ne pas l'être ?), Elsa, qui je l'espère réussira à quitter l'église scientologue, Florent T. alias Nœil, mon premier colocataire (ah, ce 40 rue Lassaigne...), qui tenta tant bien que mal d'être assidu en master, Fred, exilé au Canada où il a découvert le plaisir du tatouage, Grégory alias Schnoble, qui a fait des filles aussi mignonnes qu'il est misanthrope, Julien alias Pepesse, l'homme aux multiples (et étranges) expériences, Laurent M. alias Ajantis (c'est bon, on a fini avec les surnoms), qui en compagnie de Delphine M. (aussi appelé "la copine") et à l'aide de titane, nous a fait un beau petit Augustin (que j'espère voir avant sa majorité), Marion C., toujours aussi attachée à son Aveyron natal, Natacha, qui est devenue plus sportive que moi, Odile, néo-guyanaise mais qui sera docteur aprs moi (niark !), Quentin, dont la mandibule est maintenant plus solide que jamais, Sandy, qui j'espère me chantera une chanson pour me féliciter (j'en profite pour transmettre des bises à Chaton), Séb P., avec qui j'ai partagé une année FSGT, Sophia, la guitariste et footeuse, Thomas T., notre quota roux à la crinière audacieuse, Yacine, qui est devenu à la surprise générale un vrai globe-trotter et bien sûr Aurélien, Fabien et Mattias que je re-cite allègrement. Merci à David A., le polyglotte exilé en Allemagne avec qui j'aime parler physique des particules, Nadja, la luxembourgeoise qui sera bientôt docteur à son tour et Steven, le néo-professeur des écoles aux quizz toujours bien sentis. Je remercie aussi tous les grévistes pour les idéologies fièrement défendues, en particulier : Amélie, aujourd'hui à Rome, Céline M., l'ancienne "kiosqueuse", Elsa-Marie, sa langue bien pendue et son palais aiguisé et Maxime, l'ex-président CERISE. Ces années n'auraient pas été les mêmes sans Brigitte et son biberon, que je remercie également. J'ai aussi une pensée pour l'ASUM, qui a tenté de faire vivre le campus de La Métare avec un certain succès et pour Madame Jourda et toute l'équipe de la scolarité, que j'ai eu le plaisir de cotoyer aussi bien en tant qu'étudiant que comme enseignant. Enfin, merci aussi à Guigui, mon compagnon sportif dont les tee-shirts n'auront laissé que peu de gens insensibles et qui sera ravi de ne rentrer dans aucune catégorie.

Parmi les amis, je remercie également toute l'équipe du Tout-Puissant Babet, avec qui j'ai pris tant de plaisir sur les terrains (même les pires des stabilisés) : Christian alias Bob, qui travaille toujours beaucoup trop, Clément G., qui m'a promis de craquer une torche pour la soutenance, Jérémie le futur papa, Loïc B., qui m'a tant appris au niveau du ballon rond, aussi bien à Soulac qu'à Tarentaise, Martin, l'hypocondriaque du groupe, Olivier P., parti retrouver un climat australien proche de celui de sa Normandie natale, Pierrot, notre Loïc Perrin, Séb H., le dernier rempart, Thierry L., qui a filé au son des sirènes montpelliéraines, sans oublier Antoine et Arnaud, déjà cités par ailleurs. J'en profite pour saluer tous les compagnons de soccer, notamment Kevin, Olivier G. et Vincent. Merci aussi à ceux, recontrés plus récemment mais que j'ai toujours beaucoup de plaisir à voir, faisant partie de ce que j'appellerai sobrement le "groupe IUFM" : Émilie R., dont le talent

précoce pour la chanson nous a bluffés, Florent J., mon colocataire au Scott carbone et aux conseils avisés pour le voyage à vélo, Leïla, dont les chutes improbables sont la marque de fabrique, Ludivine, au top lorsqu'il s'agit d'organiser des voyages, Marie F., qui devrait avoir un genou opérationnel sous peu, Marine, que je félicite encore pour sa brillante réussite au concours, Marion B., que j'ai sauvée à *Sushi Ren*, Pauline, qui se souviendra longtemps de son cadeau d'anniversaire et Thomas, avec qui j'ai eu plaisir à partager des sorties VTT. J'ai également une pensée pour Alexis que j'ai bien l'temps de remercier et Hugo, l'agrégé qui ne supporte pas le plat. Merci aussi aux autres : Benoit W., aux idées de course farfelues, Bérengère, l'accordéoniste de la rue Michel Rondet, Clément P., le fou de la slackline, Damien L., le fidèle des années collège, Florian P. et nos parties de squash endiablées, Henri, notre journaliste attitré, les libraires de *Lune et l'autre*, Marie M. et Céline G., avec qui j'ai toujours plaisir à travailler lors de la Fête du Livre, sans oublier Nelly ma sœur de cœur, ainsi que Ludo, Sylvain et tous ceux que je ne peux pas citer, faute de place.

Je remercie aussi chaleureusement tous les membres de ma famille, que je ne pourrai pas citer de façon exhaustive, de m'avoir toujours soutenu, ainsi que ma belle-famille, notamment Agnès, Florian B. et Patrice. Merci à la famille de Charente, de Haute-Loire, de Saint-Régis-du-Coin ou d'ailleurs. Merci également à Christiane C. pour son investissement énorme dans la préparation du buffet. Enfin, je remercie Papy et Mamette, des grands-parents en or, Franck, mon oncle et parrain sur qui je peux toujours compter, et mon père pour sa relecture attentive du manuscrit et nos nombreuses conversations. Merci enfin à toutes les personnes qui ont assisté à la soutenance, votre présence me va droit au cœur.

Je prie tous ceux que j'aurais oublié de ne pas m'en tenir rigueur, les souvenirs sont si nombreux qu'il est difficile de s'en rappeler en intégralité. De manière générale, merci à tous ceux que j'ai croisés, de près ou de loin, au fil des années et qui ont contribué à faire de moi la personne que je suis ajourd'hui. Merci à Stéphane de Groodt de m'avoir inspiré, merci à *Monsieur Madame* de m'avoir offert une chance de monter sur scène, merci à *3 minutes sur mer* pour leur musique et merci au sport qui m'a souvent été d'une aide précieuse pour décompresser. Merci à Stanley Kubrick pour ses films de toute beauté, à Jacques Brel et Renaud pour leur poésie, à la soif d'apprendre qui ne demande qu'à être assouvie, au vélo qui va devenir mon compagnon des prochains mois et à tous ceux qui veulent faire avancer l'humanité dans la bonne direction.

Pour conclure, je tiens évidemment à adresser une très grosse pensée à Estelle et à ma maman (que je félicite pour le courage et la détermination dont elle fait preuve au quotidien) pour leur soutien sans faille, y compris dans les moments plus délicats. Je leur dois beaucoup, c'est pourquoi je tiens à les remercier très sincèrement pour tout ce qu'elles ont fait, depuis l'écoute attentive qu'elles ont montrée lors des répétitions de mes présentations, jusqu'à la réalisation du buffet. Du fond du cœur, merci.

*"Je préfère les hommes qui donnent à ceux qui expliquent."*

– Jacques Brel

*"Le commencement de toutes les sciences, c'est l'étonnement de ce que les choses sont ce qu'elles sont."*

– Aristote

# Table des matières

**Chapitre**

# Liste des tableaux

**Tableau**

# Table des figures

**Figure**

# Chapitre I

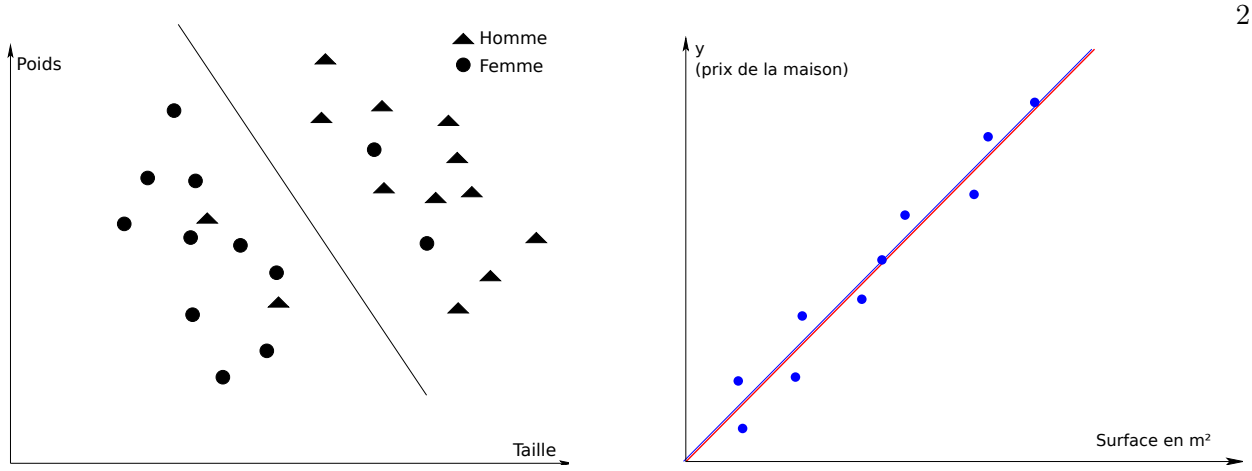
## Introduction Générale

Dans un souci toujours croissant de gain de temps, d'énergie ou d'argent, de plus en plus de tâches du quotidien tendent à être automatisées. Parmi celles-ci, on peut citer la reconnaissance de caractères manuscrits, la détection de spams, la reconnaissance vocale, l'analyse de données médicales, la reconnaissance d'objets dans des images ou des vidéos, etc. Dans ce contexte, l'*apprentissage automatique* (voir [Bishop, 2007, Cornuéjols and Miclet, 2010, Mohri et al., 2012] pour une présentation plus vaste de ce domaine de recherche) a connu un engouement sans cesse croissant. Il offre un cadre méthodologique, permettant de construire des algorithmes dont l'objectif est la conception de modèles réalisant automatiquement de telles tâches. Ces modèles (aussi appelés hypothèses) sont appris à partir de données dites *d'apprentissage* représentant des éléments observés du monde réel.

De manière générale, ces données (ou exemples) d'apprentissage peuvent être décrites selon différentes formes : des vecteurs d'attributs réels (dans le cas de données médicales, un patient pourra être représenté par sa tension, son poids, sa taille, etc) ou des données structurées, comme des chaînes de caractères, des arbres ou des graphes (les chaînes peuvent correspondre à des mots issus d'un texte, les arbres à des documents XML ou HTML, les graphes à des molécules ou des réseaux sociaux).

L'apprentissage automatique regroupe plusieurs cadres, parmi lesquels l'*apprentissage non supervisé*, où l'algorithme d'apprentissage doit identifier un certain nombre d'ensembles (appelés *clusters*) regroupant les exemples similaires au sens d'une métrique. De tels algorithmes peuvent être par exemple utilisés pour la détection de communautés scientifiques, les exemples d'apprentissage étant des auteurs scientifiques, leurs caractéristiques pouvant correspondre aux conférences dans lesquelles leurs travaux ont été publiés, à leurs co-auteurs, etc. Un objectif possible est de faire émerger automatiquement de nouvelles thématiques de recherche, ou encore des sous-communautés de chercheurs.

Un autre cadre est celui de l'*apprentissage supervisé*, où chaque exemple d'apprentissage est couplé avec une étiquette, pouvant prendre sa valeur dans un ensemble discret ou continu, et où le modèle appris a pour objectif de faire le moins d'erreurs possible sur les prédictions des étiquettes de nouveaux exemples, non vus durant l'apprentissage.




(a) (b)

FIGURE I.1 – (a) : Illustration d'une tâche de classification. On dispose d'exemples à deux dimensions (taille, poids) associés à une étiquette (homme ou femme). On cherche à apprendre un classifieur linéaire séparant au mieux les hommes des femmes. (b) : Illustration d'une tâche de régression. On cherche à apprendre une droite de régression minimisant les erreurs de prédiction des étiquettes des exemples.

Lorsque l'ensemble des étiquettes est discret, on se trouve face à une tâche dite de *classification*. Dans le cas de la détection de spams, les données d'apprentissage sont les mails de différents utilisateurs et leur représentation peut par exemple prendre la forme d'un sac de mots, contenant leur fréquence d'apparition. Dans ce cas, les deux étiquettes possibles pour un mail sont "spam" ou "non-spam" en fonction de l'annotation de l'utilisateur. Il s'agit ensuite de construire un modèle (ou classifieur), à l'aide des exemples dont on dispose, capable d'assigner automatiquement la bonne étiquette à un nouveau mail.

Lorsque l'ensemble des étiquettes est continu, la tâche à réaliser est appelée *régression*. Une illustration est la prédiction de température en fonction de divers attributs météorologiques. On peut également imaginer une tâche consistant à prédire le prix d'une maison en fonction du nombre de pièces, de la surface habitable, de la taille du jardin, etc. Une illustration simple en deux dimensions de ces deux sous-domaines de l'apprentissage supervisé est donnée par la Figure I.1. Dans ce travail de thèse, nous nous concentrerons exclusivement sur des tâches de classification supervisée.

Dans un tel contexte, un postulat généralement admis est que les données d'apprentissage et les données sur lesquelles sera déployé le modèle appris sont issues de la même distribution statistique. Si cette hypothèse est plutôt légitime dans certaines applications, elle devient forte pour traiter bon nombre de problèmes du monde réel. Pour revenir à l'exemple de la détection de spams, de nombreuses différences peuvent être observées d'une boîte mail à une autre, par rapport à l'utilisation de celle-ci (à titre personnel ou professionnel) ou encore en fonction des différentes listes de diffusion souscrites. De plus, la notion même de spam peut être différente selon les utilisateurs

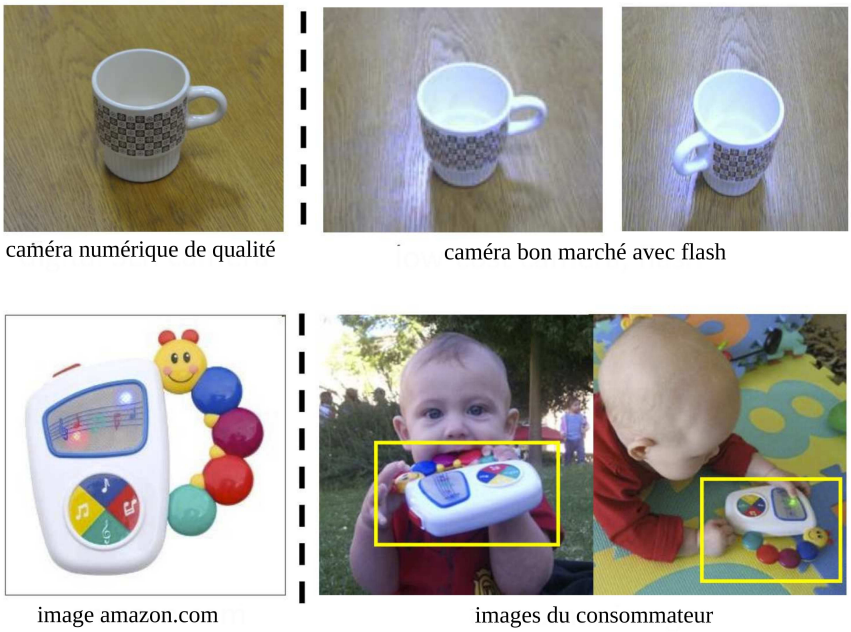



FIGURE I.2 – Exemple d'une tâche d'adaptation de domaine en computer vision : on cherche à reconnaître un objet dans une image. Dans le premier cas (en haut), les conditions d'acquisition varient entre les images d'apprentissage (à gauche) et celles de test (à droite). Dans le deuxième cas (en bas), le contexte dans lequel se situe l'objet est différent entre l'apprentissage et le test [Kulis et al., 2011]

(certains considérant les publicités ou les invitations à signer des pétitions comme des spams, d'autres non). Ainsi, un modèle appris à partir de la boîte mail d'un individu ne sera pas forcément efficace sur celle d'une autre personne.

Le constat du non respect de cette hypothèse de distributions identiques se retrouve également beaucoup en *computer vision* (*e.g.* sur des problématiques de classification d'images) [Kulis et al., 2011], comme illustré par l'exemple de la Figure I.2. En effet, les conditions d'acquisition de l'image ou le modèle de l'appareil photo utilisé peuvent engendrer des différences importantes dans la représentation des données [Torralba and Efros, 2011]. Dans le cas de la reconnaissance de caractères manuscrits, la différence d'écriture entre plusieurs utilisateurs peut conduire à une divergence importante entre les distributions d'apprentissage et de test. Même si les tâches sont liées (reconnaître un type d'écriture nous aide généralement à en reconnaître un autre), l'apprentissage ne peut plus être réalisé sans tenir compte de cette différence de distributions. Pour faire face aux limitations du cadre classique d'apprentissage, un nouveau domaine de recherche a émergé ces dernières années : l'*adaptation de domaine*[1] (AD).

En AD, l'ensemble d'apprentissage est composé de deux sous-parties : la première contient

1. Un domaine sera généralement caractérisé par la distribution associée aux données.

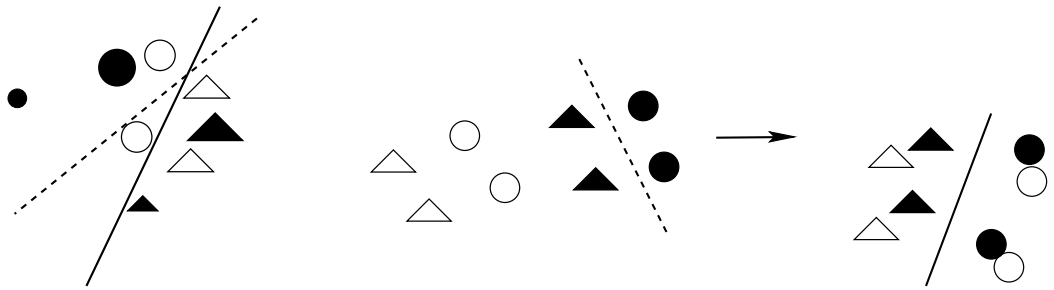



FIGURE I.3 – Illustration de deux des approches algorithmiques d'adaptation de domaine. Dans les figures (a) et (b), les ronds et les triangles représentent deux classes différentes. Les noirs sont ceux du domaine source, les blancs ceux du domaine cible. Le séparateur en pointillés correspond à celui qui serait inféré en cas d'apprentissage sur les exemples sources uniquement. Le classifieur en trait plein correspond à celui que l'on obtient après la procédure d'adaptation. (a) illustre l'idée de repondération des exemples. Les points sources les plus proches des points cibles ont un poids plus important, pour avoir une influence plus grande dans la construction du séparateur. Dans (b) est représentée la notion de reprojection. Le but est de trouver un espace commun dans lequel les exemples des deux domaines seront proches.

des points dits "sources" étiquetés. Ils correspondent à ce que nous appelions jusqu'à présent les exemples d'apprentissage. La seconde comprend des exemples dits "cibles". Ceux-ci sont issus de la distribution statistique qui caractérise les données sur lesquelles le modèle appris sera déployé. Deux situations peuvent alors être distinguées : (i) *l'adaptation de domaine semi-supervisée*, où des exemples cibles étiquetés sont disponibles, mais dans une quantité insuffisante pour apprendre un modèle performant à l'aide de ceux-ci uniquement et (ii) *l'adaptation de domaine non supervisée*, où les données cibles sont présentes en nombre, mais sans information sur leur étiquette. Dans les deux cas, il est donc nécessaire d'utiliser les exemples sources étiquetés dont on dispose pour inférer le modèle qui sera déployé sur les données cibles. Dans cette thèse, nous ne travaillerons que dans le cadre, plus complexe, de l'adaptation de domaine non supervisée.

À partir des premiers travaux théoriques réalisés en AD [Ben-David et al., 2006], un certain nombre de conditions nécessaires ont été établies pour bien adapter. Ces conditions théoriques, qui seront détaillées plus formellement dans le cadre de cette thèse, peuvent être résumées de manière simplifiée comme suit : si un algorithme d'AD est capable de réduire la divergence statistique entre les données sources et cibles, tout en apprenant de manière supervisée un classifieur performant sur les données sources disponibles, ce classifieur devrait être efficace également sur les données cibles.

Ces conditions théoriques ont donné naissance à trois grandes familles d'algorithmes d'AD :

- Les méthodes dites de *repondération* sont utilisées lorsque les données sources ne sont pas assez représentatives des données cibles (à cause d'un biais dans la collecte par exemple). Dans ces cas-là, le décalage entre les deux distributions est généralement faible. Ces ap-

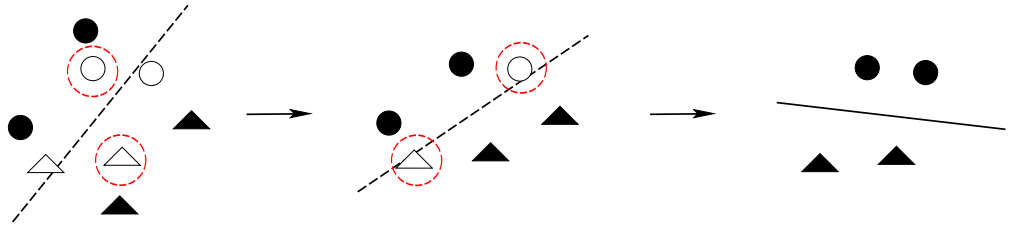


FIGURE I.4 – Illustration du principe des algorithmes d'auto-étiquetage. Les ronds et les triangles représentent deux classes différentes. Les noirs sont ceux du domaine source, les blancs ceux du domaine cible. Sur la gauche, le classifieur est inféré à l'aide des exemples sources uniquement. Ensuite, à chaque étape, un certain nombre de points sources sont supprimés de l'ensemble d'apprentissage, tandis que des points cibles sont étiquetés et insérés dans cet ensemble. Un nouveau classifieur est ensuite appris. Sur la droite, on voit un modèle appris uniquement à l'aide des points cibles, étiquetés itérativement.

proches ont pour but d'apprendre une repondération des exemples sources, afin d'éliminer ce biais. La distribution source repondérée doit donc être la plus similaire possible à la distribution cible (voir Figure I.3(a)). Une méthode d'apprentissage supervisé classique est ensuite utilisée pour inférer un modèle depuis l'ensemble d'apprentissage source repondéré.

- Les méthodes dites de *reprojection* peuvent être utilisées lorsque les deux domaines ne possèdent pas le même espace d'entrée. Celles-ci visent à apprendre un nouvel espace de projection commun aux deux domaines, dans lequel ces derniers seront proches selon une certaine mesure de divergence (voir Figure I.3(b)). Ces espaces peuvent être composés de nouveaux attributs latents ou être issus d'une procédure de sélection d'attributs. Dans tous les cas, l'objectif est de trouver l'espace dans lequel la divergence entre les deux distributions est la plus faible, afin d'utiliser un algorithme d'apprentissage supervisé classique sur les exemples sources étiquetés, inférant un modèle qui sera aussi efficace sur le domaine cible.

- Enfin, la dernière catégorie rassemble les approches dites *d'auto-étiquetage*. Ces dernières, opérant de façon itérative, commencent par apprendre une hypothèse sur les données sources étiquetées. Par la suite, à chaque étape, sont insérées dans l'ensemble d'apprentissage étiqueté des données cibles auxquelles l'hypothèse a attribué une étiquette, en remplacement d'exemples issus de la distribution source. Le but est donc d'adapter itérativement l'hypothèse (voir Figure I.4).

Notons que les méthodes de reprojection et d'auto-étiquetage opèrent généralement, contrairement à celles de repondération, dans des situations où les écarts de distributions sont plus importants. C'est dans ce contexte, plus compliqué, que se positionne cette thèse. Deux contributions principales y sont proposées.

La première se place dans le cadre de la reprojection d'exemples et est basée sur la théorie

du *boosting*[2] [Schapire, 1989]. Elle consiste en un algorithme théoriquement fondé, appelé SLDAB, qui projette les données sources et cibles dans l'espace des sorties des classifieurs faibles appris de manière à ce que l'erreur empirique sur les exemples sources soit minimisée et que les marges sur les points cibles soient maximisées.

Notre seconde contribution se présente sous la forme d'un algorithme d'auto-étiquetage baptisé GESIDA, reposant sur une analyse théorique préalable que nous avons effectuée sur les conditions nécessaires au bon fonctionnement d'une telle approche itérative. GESIDA généralise l'algorithme DASVM, introduit dans [Bruzzone and Marconcini, 2010], aux données structurées. Il tire également parti de la théorie des $(\varepsilon,\gamma,\tau)$-bonnes fonctions de similarité, introduite dans [Balcan and Blum, 2006, Balcan et al., 2008], afin de s'affranchir de certaines contraintes imposées par DASVM, notamment l'utilisation de fonctions de similarité semi-définies positives.

Cette thèse a été effectuée dans l'équipe *Machine Learning* du Laboratoire Hubert Curien UMR CNRS 5516, rattaché à l'Université de Saint-Étienne, membre de l'Université de Lyon. Ce travail s'est inscrit dans le contexte du projet ANR LAMPADA (ANR-09-EMER-007) et de PASCAL2, réseau européen d'excellence, soutenant la recherche en apprentissage automatique, en statistiques et en optimisation.

Le manuscrit est organisé de la manière suivante :

- Le Chapitre II introduit les définitions et notations nécessaires à la bonne compréhension du document et des contributions qui y sont présentées. Le contexte particulier de l'apprentissage supervisé y est détaillé de manière formelle.

- Dans le Chapitre III est proposé, après une présentation du problème de l'AD, un état de l'art des approches existantes. Après avoir introduit les principales contributions théoriques, consistant principalement en des bornes sur l'erreur en généralisation et des mesures de divergence entre les deux domaines, les approches algorithmiques sont présentées, séparées en trois catégories : les techniques de repondération, les méthodes basées sur le changement d'espace de représentation et les approches d'auto-étiquetage.

- Le Chapitre IV introduit la première contribution de cette thèse, approche de reprojection qui prend la forme de l'algorithme SLDAB. Celui-ci, inspiré d'AdaBoost [Freund and Schapire, 1996], vise à apprendre itérativement des séparateurs dits faibles, respectant une condition d'erreur sur la source et une condition de marge sur la cible. Un terme de divergence entre les deux domaines est également pris en compte. Le résultat final de l'algorithme est une combinaison linéaire optimisée de toutes les hypothèses faibles. Dans ce chapitre, nous effectuons une analyse théorique du comportement de SLDAB. Nous introduisons également une nouvelle mesure de divergence, inspirée de la *variation perturbée* [Harel and

2. Le boosting consiste à combiner des hypothèses faibles (*i.e.* meilleures que l'aléatoire), apprises itérativement, afin d'obtenir une hypothèse finale forte.

Mannor, 2012] et terminons par des résultats expérimentaux confirmant le bon comportement de l'algorithme, en comparaison avec plusieurs approches de l'état de l'art.

- Dans le Chapitre V, nous nous intéressons en particulier aux approches d'auto-étiquetage. Nous offrons dans un premier temps une analyse théorique sur les conditions nécessaires permettant la réussite d'un algorithme d'auto-étiquetage dans le cadre de l'adaptation de domaine. Après cette étude, nous introduisons notre algorithme GESIDA, tirant parti de la théorie des $(\varepsilon,\gamma,\tau)$-bonnes fonctions de similarité [Balcan and Blum, 2006, Balcan et al., 2008], afin de nous affranchir des contraintes de semi-définie positivité de l'algorithme DASVM [Bruzzone and Marconcini, 2010], référence dans le domaine des approches d'auto-étiquetage. La théorie de Balcan et al. permet de définir formellement à quel point une fonction de similarité est pertinente pour une tâche donnée et offre des garanties théoriques sur le classifieur linéaire qui exploitera cette fonction. Nous évaluons GESIDA, en comparaison avec d'autres algorithmes, sur un problème de reconnaissance d'images représentées sous la forme de données structurées, en l'occurrence des chaînes de caractères et des arbres.
- Enfin, le Chapitre VI résume les contributions présentées dans cette thèse et introduit de nouvelles perspectives s'inscrivant logiquement dans la suite de ce travail.

Le Tableau I.1 récapitule les notations que nous utiliserons tout au long de cette thèse.

TABLEAU I.1 – Récapitulatif des notations utilisées.

| Notation | Correspondance |
|---|---|
| $\mathbb{R}$ | Ensemble des nombres réels |
| $\mathbb{R}^+$ | Ensemble des nombres réels positifs |
| $\mathbb{R}^d$ | Ensemble des vecteurs réels de dimension $d$ |
| $\mathbb{N}$ | Ensemble des entiers naturels |
| $[n]$ | L'ensemble $1, \ldots, n$ |
| $X$ | Espace d'entrée |
| $Y$ | Espace de sortie (espace des étiquettes) |
| $z = (x, y)$ | Exemple étiqueté arbitraire |
| $\mathbf{x}$ | Vecteur arbitraire |
| $x^i$ | $i^{\text{ème}}$ élément de $\mathbf{x}$ |
| $\mathsf{x}$ | Chaîne de caractères arbitraire |
| $\lvert\mathsf{x}\rvert$ | Longueur de $\mathsf{x}$ |
| $\mathsf{x}^i$ | $i^{\text{ème}}$ symbole de $\mathsf{x}$ |
| $S$ | Ensemble arbitraire |
| $\lvert S\rvert$ | Taille de $S$ |
| $\mathcal{D}_S$ | Distribution dont est issu l'ensemble $S$ |
| $\mathcal{D}_S^X$ | Distribution marginale de $S$ selon $X$ |
| $[\cdot]_+$ | Fonction hinge |
| $\lvert\lvert \cdot \rvert\rvert$ | Norme arbitraire |
| $\lvert\lvert \cdot \rvert\rvert_p$ | Norme $L_p$ |
| $x \sim \mathcal{D}$ | $x$ est distribué i.i.d. selon une distribution $\mathcal{D}$ |
| $Pr[\cdot]$ | Probabilité d'un événement |
| $\mathbb{E}[\cdot]$ | Espérance d'une variable aléatoire |

# Chapitre II

# Préliminaires

## II.1 Introduction

Nous nous intéressons dans cette thèse au cadre de l'apprentissage supervisé. Le but d'un algorithme d'apprentissage supervisé est d'inférer un modèle depuis un ensemble d'exemples d'apprentissage étiquetés, capable d'obtenir une bonne performance sur de nouvelles données, comme illustré dans la Figure II.1. Dans ce chapitre, nous introduisons les notions essentielles de l'apprentissage supervisé, le cadre classique PAC, initialement introduit par [Valiant, 1984], ainsi que les travaux plus récents en stabilité uniforme [Bousquet and Elisseeff, 2002] et en robustesse algorithmique [Xu and Mannor, 2010, Xu and Mannor, 2012]. Nous présentons spécifiquement deux algorithmes d'apprentissage supervisé, sur lesquels se basent nos contributions.

## II.2 Apprentissage supervisé

Comme évoqué précédemment, un algorithme d'apprentissage supervisé infère une hypothèse depuis un ensemble d'exemples d'apprentissage étiquetés. Nous définissons la notion d'ensemble d'apprentissage de la manière suivante.

**Définition II.1** (Ensemble d'apprentissage)**.** *Un ensemble d'apprentissage de taille $n$ est un ensemble $S = \{z_i = (x_i, y_i)\}_{i=1}^n$ de $n$ observations indépendamment et identiquement distribuées (i.i.d.), selon une distribution conjointe inconnue $\mathcal{D}_S$ sur l'espace $Z = X \times Y$, où $X$ représente l'espace d'entrée et $Y$ l'espace de sortie. Pour une observation donnée $z_i \in Z$, $x_i \in X$ est l'exemple et $y_i \in Y$ l'étiquette associée. Quand $Y$ est discret, on se retrouve confronté à une tâche de classification, et $y_i$ est appelé classe de $x_i$, tandis que quand $Y$ est continu, on a affaire à une tâche de régression et $y_i$ est la valeur réelle correspondant à $x_i$.*

Dans ce travail, nous nous focalisons sur des tâches de classification binaire, à savoir $Y = \{-1, +1\}$. Nous utilisons dans cette thèse des exemples d'apprentissage prenant la forme soit de

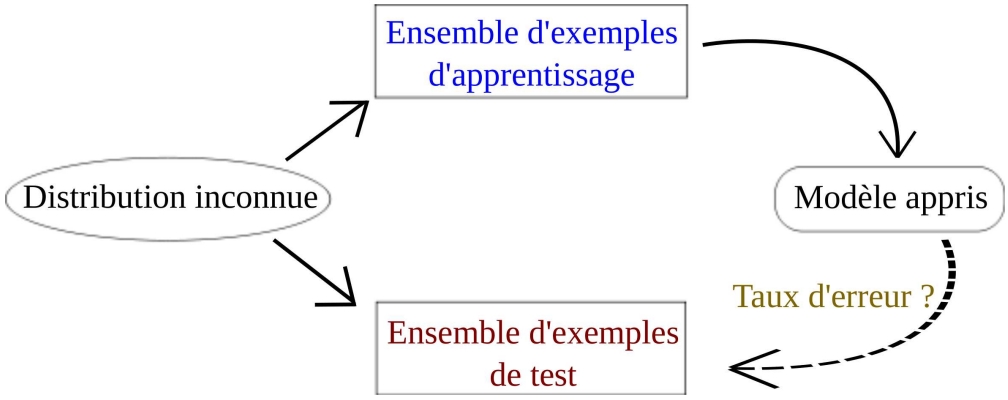



FIGURE II.1 – Illustration simplifiée du problème de l'apprentissage supervisé.

vecteurs d'attributs, pour lesquels nous considérons que $X \subseteq \mathbb{R}^d$, soit de données structurées sous la forme de chaînes de caractères ou d'arbres.

**Définition II.2** (Alphabet et chaîne de caractères)**.** *Un alphabet $\Sigma$ est un ensemble non vide fini de symboles. Une chaîne de caractères $x$ est une suite de symboles de $\Sigma$. La chaîne vide est notée $\$$ et $\Sigma^*$ représente l'ensemble de toutes les chaînes de longueur finie (y compris $\$$) pouvant être générées depuis $\Sigma$. La longueur d'une chaîne $x$ est notée $|x|$.*

**Définition II.3** (Arbre)**.** *On considère un ensemble de valeurs $V$. Un arbre sur cet ensemble est défini de façon récursive comme suit. Étant donné une valeur $v$ de noeud ($v \in V$) et $m$ sous-arbres $a_1, \ldots, a_m$, $(v, (a_1, \ldots, a_m))$ est un nouvel arbre de racine $v$ et possédant $m$ fils : $a_1, \ldots, a_m$. Un élément de $V$ ne possédant aucun fils est une feuille.*

Une illustration d'arbre est donnée dans la Figure II.2.

L'apprentissage supervisé peut être défini de la manière suivante.

**Définition II.4** (Apprentissage supervisé)**.** *L'apprentissage supervisé est la tâche consistant à inférer une fonction $h : X \to L$ (où $L$ est l'espace de décision), appartenant à une classe d'hypothèses $\mathcal{H}$, depuis un ensemble d'apprentissage $S$, prédisant du mieux possible $y \in Y$ à partir de $x \in X$, pour tout $(x, y)$ issu de $\mathcal{D}_S$. L'espace de décision $L$ peut être différent de $Y$.*

Notons que parfois, l'ensemble d'apprentissage peut être composé d'un sous-ensemble contenant des exemples non étiquetés. C'est souvent le cas dans des tâches où l'étiquetage des données est particulièrement coûteux. L'algorithme peut utiliser ces exemples durant la phase d'apprentissage afin d'améliorer la qualité du modèle inféré. On parle alors d'*apprentissage semi-supervisé*.

Une question cruciale en apprentissage supervisé est de définir la qualité d'une hypothèse $h$. On l'évalue généralement à l'aide d'une fonction de perte non-négative $\ell : \mathcal{H} \times Z \to \mathbb{R}^+$, calculant $\forall (x_i, y_i)$ une mesure de désaccord entre la sortie de l'hypothèse $h(x_i)$ et la vraie étiquette $y_i$. Une

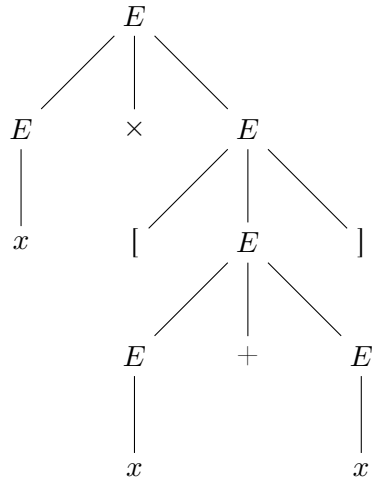



FIGURE II.2 – Représentation graphique de l'arbre $(E, ((E, (x)), (\times), (E, ([), (E, (E, (x)), (+), (E, (x))), (]))))$

fonction de perte utilisée habituellement dans des tâches de classification binaire est la *perte 0/1*, aussi appelée erreur de classification :

$$\ell_{0/1}(h, z) = \begin{cases} 1 & \text{si } yh(x) < 0 \\ 0 & \text{sinon.} \end{cases}$$

L'erreur sur un exemple est donc de 1 lorsque la prédiction de l'hypothèse $h$ est différente de l'étiquette réelle, et de 0 sinon. À partir d'une fonction de perte comme celle-ci, on peut définir la notion d'erreur réelle (ou erreur en généralisation) d'une hypothèse $h$ sur une distribution $\mathcal{D}_S$.

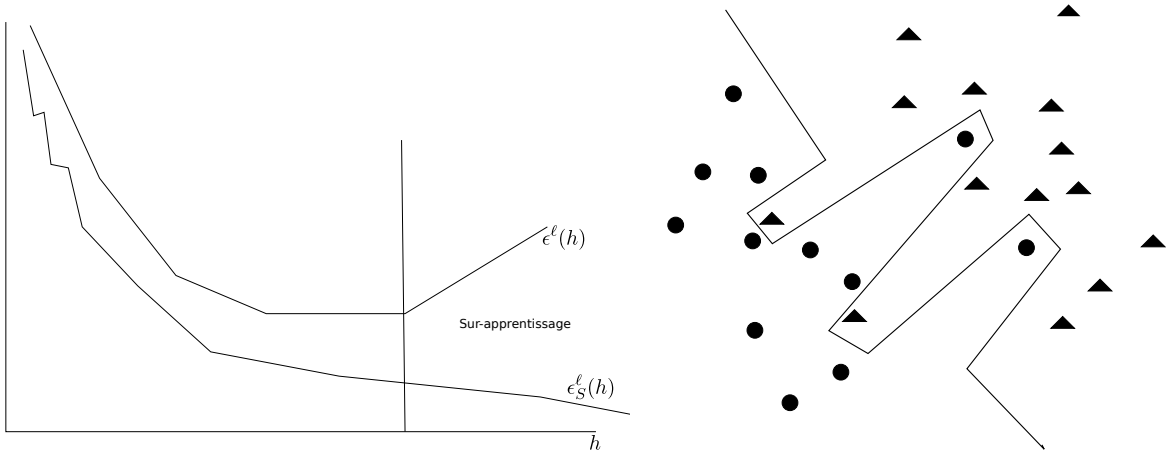
**Définition II.5** (Erreur réelle)**.** *L'erreur réelle (ou risque réel) $\epsilon^{\ell}_{\mathcal{D}_S}(h)$ d'une hypothèse $h$ selon une fonction de perte $\ell$ correspond à l'espérance de la perte de $h$ sur la distribution $\mathcal{D}_S$ :*

$$\epsilon^{\ell}_{\mathcal{D}_S}(h) = \mathbb{E}_{z \sim \mathcal{D}_S}[\ell(h, z)].$$

Le but de l'apprentissage supervisé est de trouver une hypothèse minimisant cette erreur réelle. Cependant, celle-ci ne peut généralement pas être mesurée puisque la distribution $\mathcal{D}_S$ est inconnue. Il est par contre possible de calculer une estimation sur l'ensemble d'apprentissage $S$, appelée erreur empirique.

**Définition II.6** (Erreur empirique)**.** *Soit $S = \{z_i = (x_i, y_i)\}_{i=1}^{n}$ un ensemble d'apprentissage. L'erreur empirique (ou risque empirique) $\epsilon^{\ell}_{S}(h)$ d'une hypothèse $h$ sur $S$, selon une fonction de perte $\ell$, correspond à la perte moyenne de $h$ sur les exemples de $S$ :*

$$\epsilon^{\ell}_{S}(h) = \frac{1}{n} \sum_{i=1}^{n} \ell(h, z_i).$$




(a) (b)

FIGURE II.3 – Illustration du sur-apprentissage. Sur la figure (a), on trouve en abscisse, la complexité de l'hypothèse apprise et en ordonnée, les erreurs empirique et réelle. Une hypothèse trop complexe, comme celle de la figure (b), bien que permettant d'obtenir une erreur empirique très faible, entraînera une forte augmentation de l'erreur en généralisation.

Dans la présentation de nos contributions, l'erreur empirique d'une hypothèse $h$ sur un ensemble $S$ sera notée $\epsilon_S(h)$, et sera calculée en utilisant la fonction de perte $\ell_{0/1}$.

L'inférence d'une bonne hypothèse, c'est-à-dire qui dispose d'une erreur réelle faible, doit être guidée par un paradigme d'apprentissage. Celui visant à minimiser, durant l'apprentissage, l'erreur empirique est le plus souvent utilisé. Cependant, ce principe doit être appliqué sous certaines conditions. En effet, cette stratégie n'est judicieuse que si un nombre d'exemples d'apprentissage suffisamment important est disponible. Si ce n'est pas le cas, il existe toujours une hypothèse $h$, parfois complexe, qui prédit parfaitement les étiquettes des exemples d'apprentissage, *i.e.* $\epsilon^{\ell}_{S}(h) = 0$, sans garantir pour autant que $h$ ait une erreur en généralisation faible. Une situation où l'erreur en généralisation d'une hypothèse est bien plus importante que son erreur empirique caractérise un phénomène de *sur-apprentissage* (voir Figure II.3).

Une solution consiste à essayer de trouver un compromis entre la diminution de l'erreur empirique et la complexité de l'hypothèse, connu sous le nom de *compromis bias-variance*. Plusieurs paradigmes existent pour parvenir à des modèles obtenant une faible erreur en généralisation, parmi lesquels on trouve :

- La *minimisation du risque empirique* (*Empirical Risk Minimization* -ERM-), qui consiste à sélectionner une hypothèse $h$, dans un espace contraint d'hypothèses $\mathcal{H}$, minimisant le risque empirique $\epsilon^{\ell}_{S}(h)$ sur un ensemble d'apprentissage $S$. Une des limitations de cette approche est due à la difficulté du choix de l'espace d'hypothèses. En effet, $\mathcal{H}$ doit être assez grand pour inclure des hypothèses ayant une faible erreur empirique, mais dans le même temps

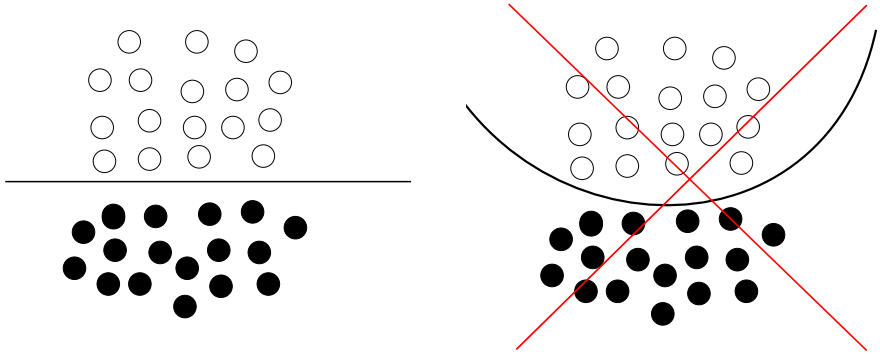


FIGURE II.4 – Illustration du principe du rasoir d'Occam. On privilégie toujours l'hypothèse la plus simple consistante avec les données. Ici, on sélectionnera le séparateur linéaire plutôt que le séparateur convexe.

être assez restreint pour éviter le sur-apprentissage. Sélectionner un espace $\mathcal{H}$ approprié sans connaissance préalable sur la tâche à réaliser est donc difficile.

- La *minimisation du risque structurel* (*Structural Risk Minimization* -SRM-), qui utilise une suite infinie de classes d'hypothèses de taille croissante, *i.e.* $\mathcal{H}_1 \subset \mathcal{H}_2 \subset \ldots$, et sélectionne l'hypothèse qui minimise une version pénalisée du risque empirique $\epsilon_S^\ell(h) + \text{pen}(\mathcal{H}_c)$, tendant à privilégier les classes d'hypothèses les moins complexes. Cette approche est basée sur le principe dit du *rasoir d'Occam*, qui suggère de toujours choisir la solution la plus simple consistante avec les données d'apprentissage (comme on peut le voir sur la Figure II.4, où on privilégiera le séparateur linéaire au séparateur convexe).

- La *minimisation du risque régularisé* (*Regularized Risk Minimization* -RRM-), qui suit elle aussi l'idée du rasoir d'Occam, mais se présente sous une forme différente. Il s'agit en effet de sélectionner un vaste espace d'hypothèses $\mathcal{H}$ et un terme de régularisation (généralement une norme $||h||$) afin de sélectionner l'hypothèse $h$ obtenant le meilleur compromis, via un paramètre $\lambda$, entre la minimisation du risque empirique et celle du terme de régularisation. Le problème de minimisation est alors le suivant : $\min_{h \in \mathcal{H}} \epsilon_S^\ell(h) + \lambda ||h||$.

Toutes ces approches se basent sur la minimisation du risque empirique, dépendant lui-même de la fonction de perte $\ell$. La plus naturelle des fonctions de perte en classification binaire, comme mentionné précédemment, est la perte $0/1$. Cependant, celle-ci n'est ni convexe, ni différentiable en 0, ce qui entraîne des problèmes de complexité quant à sa minimisation pour une tâche donnée. Il est notamment montré dans [Feldman et al., 2009] que ce problème est NP-complet, même dans le cas de classes de fonctions simples comme les classifieurs linéaires. C'est pourquoi des fonctions alternatives convexes (donc optimisables plus efficacement) sont utilisées. Parmi les principales pertes existantes, nous pouvons citer :

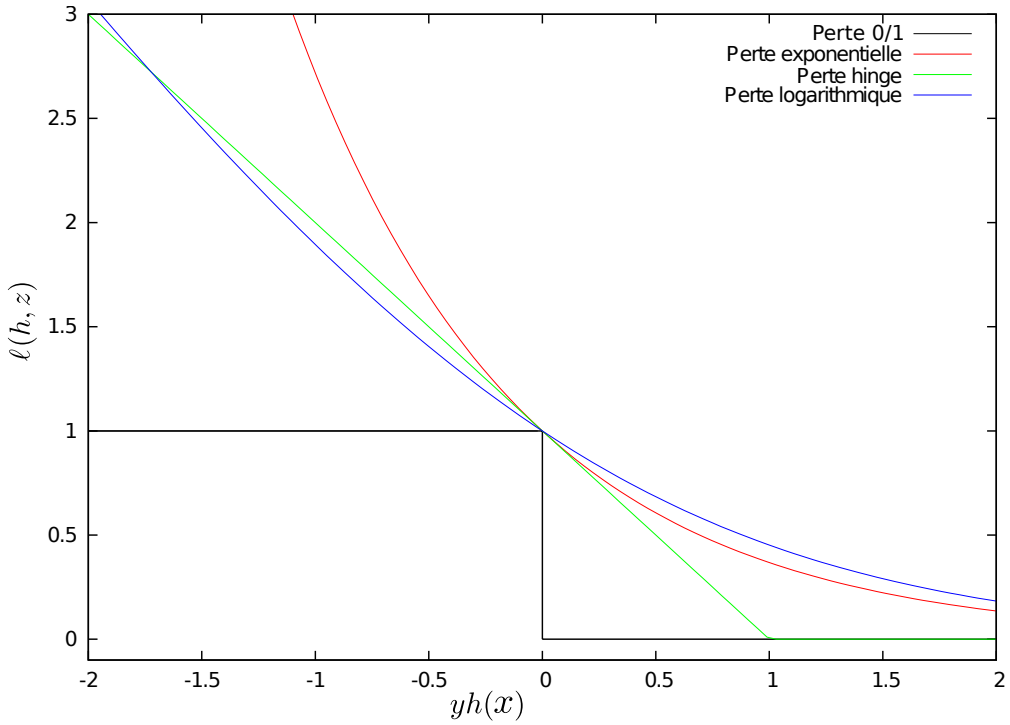



FIGURE II.5 – Comportement de différentes fonctions de perte en classification binaire.

- La *perte exponentielle* : $\ell_{exp}(h,z) = \exp^{-yh(x)}$, utilisée par exemple dans la théorie du boosting, notamment dans ADABOOST [Freund and Schapire, 1996]. Nous y aurons recours dans le Chapitre IV.
- La *perte hinge* : $\ell_{hinge}(h,z) = [1 - yh(x)]_+ = \max(0, 1 - yh(x))$, utilisée par les machines à vecteurs de support (SVM), algorithme introduit dans [Cortes and Vapnik, 1995]. Nous y ferons référence au cours du Chapitre V.
- La *perte logarithmique* : $\ell_{log}(h,z) = \log(1 + \exp^{-yh(x)})$, utilisée entre autres par LOGITBOOST, algorithme présenté dans [Friedman et al., 1998].

Ces trois fonctions de perte, ainsi que la $\ell_{0/1}$, sont illustrées dans la Figure II.5. Le choix de la fonction de perte est en général difficile à effectuer et il dépend principalement du problème à traiter et de la classe d'hypothèses considérée. Il existe quelques résultats sur l'intérêt relatif des différentes fonctions de perte [Ben-David et al., 2012, Rosasco et al., 2004]. Il ressort de ces travaux que la perte hinge est celle qui offre en général les meilleures garanties.

Il est intéressant de noter que ces trois fonctions utilisent l'expression $yh(x)$, qui correspond à la marge d'un exemple $x$ pour une hypothèse $h$ donnée. Celle-ci est positive uniquement dans le cas où $x$ est correctement classé par $h$. On voit donc bien ici que maximiser les marges revient, par approximation de la $\ell_{0/1}$, à minimiser le taux d'erreur de classification.

Les approches introduites dans cette section fournissent un cadre algorithmique pour trouver un modèle $h$ par réduction de l'erreur empirique, tout en évitant le risque de sur-apprentissage. Cependant, ceci ne nous dit rien sur la performance en généralisation de $h$. Le cadre PAC proposé par Valiant [Valiant, 1984] propose une réponse théorique à cette problématique, en permettant notamment de dériver des bornes en généralisation.

## II.3 Le modèle PAC

Le cadre théorique PAC (Probablement Approximativement Correct) [Valiant, 1984] permet de dériver des bornes sur l'erreur en généralisation d'une hypothèse. Une borne PAC se présente sous la forme suivante :

$$Pr[|\epsilon^{\ell}_{\mathcal{D}_S}(h) - \epsilon^{\ell}_S(h)| < \mu] \geq 1 - \delta,$$

avec $\mu \geq 0$ et $\delta \in [0, 1]$. Il s'agit donc d'une borne sur la probabilité d'observer une différence entre l'erreur empirique sur $S$ et l'erreur réelle sur la distribution dont est issu $S$ inférieure à $\mu$. Le terme PAC s'explique par la présence des paramètres $\mu$ (pour le "Approximativement"), qui caractérise la différence entre l'erreur empirique sur $S$ et l'erreur réelle, et $\delta$ (pour le "Probablement"), qui correspond à la probabilité que cette différence soit inférieure à $\mu$.

La dérivation des bornes PAC repose généralement sur des outils statistiques utilisant des inégalités de concentration. Nous présentons ici trois cadres différents.

### II.3.1 La convergence uniforme

La *convergence uniforme* [Vapnik and Chervonenkis, 1971] est peut-être le cadre le plus connu. Il permet de dériver des bornes en généralisation relativement à une famille d'hypothèses. Dans le cas où $\mathcal{H}$ est fini, on obtient des bornes de la forme suivante.

**Théorème II.1** (Borne de convergence uniforme dans le cas fini)**.** *Soient $S$ un ensemble d'apprentissage de taille $n$, i.i.d. selon une distribution $\mathcal{D}_S$, $\mathcal{H}$ un espace d'hypothèses de taille finie et $\delta > 0$. Pour tout $h \in \mathcal{H}$, avec une probabilité de $1 - \delta$ sur l'ensemble aléatoire $S$, on obtient :*

$$\epsilon^{\ell}_{\mathcal{D}_S}(h) \leq \epsilon^{\ell}_S(h) + \sqrt{\frac{\ln |\mathcal{H}| + \ln \frac{1}{\delta}}{2n}}.$$

Ces bornes prennent la forme d'une somme de l'erreur empirique de $h$ sur l'ensemble d'apprentissage $S$ et d'un terme de pénalité dépendant du nombre $n$ d'exemples de $S$, de la complexité de l'espace d'hypothèses $\mathcal{H}$ et de la valeur de $\delta$. Le convergence uniforme traduit le fait qu'à $\mathcal{H}$ fixée, plus $n$ est grand, plus l'erreur empirique converge vers l'erreur réelle, classiquement à une vitesse

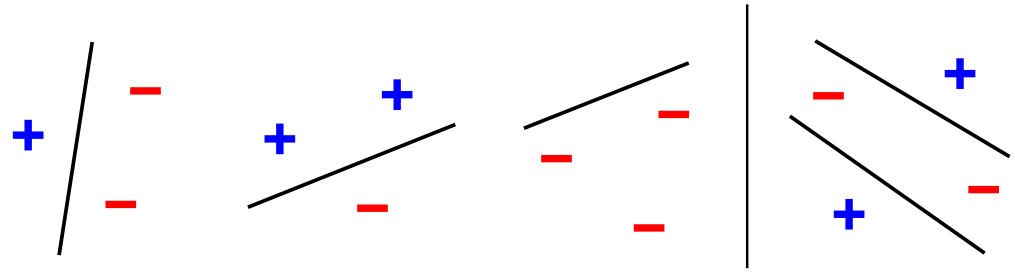


FIGURE II.6 – Illustration de la pulvérisation des points par une droite dans le plan. Une droite est capable de séparer 3 points selon tous les étiquetages possibles (sur la gauche du dessin, 3 des 8 étiquetages sont illustrés). Par contre, une droite n'est pas capable de séparer 4 points selon tous les étiquetages, comme illustré à droite. La VC-dimension de cette famille de séparateurs est donc de 3.

de convergence en $\mathcal{O}(\frac{1}{\sqrt{n}})$. Dans le même temps, plus la classe d'hypothèses $\mathcal{H}$ est complexe, plus le terme de pénalité augmente, traduisant un risque de sur-apprentissage.

Lorsque $|\mathcal{H}|$ est infini, la borne précédente devient inutile. Afin d'exprimer la capacité d'une famille d'hypothèses, on fait en général dans ce cas appel à des mesures de complexité. Par exemple, la VC-dimension [Vapnik and Chervonenkis, 1971] correspond au cardinal du plus grand ensemble de points que l'algorithme peut pulvériser. On dit d'une classe d'hypothèses $\mathcal{H}$ qu'elle pulvérise un ensemble de données $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ si pour tout étiquetage de cet ensemble, il existe $h \in \mathcal{H}$ ne faisant aucune erreur dans l'étiquetage de ces données. Dans le cas des séparateurs linéaires dans le plan par exemple, il est possible de pulvériser 3 points mais pas 4, comme illustré par la Figure II.6. La VC-dimension de cette classe d'hypothèses, ainsi que celle d'un algorithme inférant de tels séparateurs est donc de 3. Dans le cas de classes d'hypothèses infinies, les bornes en généralisation utilisant la VC-dimension prennent la forme suivante.

**Théorème II.2** (Borne de convergence uniforme utilisant la VC-dimension)**.** *Soient $S$ un ensemble d'apprentissage de taille $n$, i.i.d. selon une distribution $\mathcal{D}_S$, $\mathcal{H}$ un espace d'hypothèses de VC-dimension $d$ et $\delta > 0$. Pour tout $h \in \mathcal{H}$, avec une probabilité de $1 - \delta$ sur l'ensemble aléatoire $S$, on a :*

$$\epsilon^{\ell}_{\mathcal{D}_S}(h) \leq \epsilon^{\ell}_{S}(h) + \sqrt{\frac{d\left(\ln \frac{2n}{d} + 1\right) + \ln \frac{4}{\delta}}{n}}.$$

La complexité de Rademacher [Koltchinskii, 2001, Bartlett and Mendelson, 2002] est une autre mesure destinée à estimer la capacité d'une famille d'hypothèses. Plutôt que de considérer, comme la VC-dimension, le pire cas (*i.e.* tous les étiquetages possibles), cette mesure estime la capacité moyenne à l'aide de variables aléatoires représentant différents étiquetages.

**Définition II.7** (Complexité de Rademacher)**.** *Soit $\mathcal{H}$ un ensemble de fonctions réelles définies sur $X$. Étant donné un ensemble $S \in X^m$, la complexité de Rademacher empirique de $\mathcal{H}$ est définie de la manière suivante :*

$$\widehat{\mathfrak{R}}_S(\mathcal{H}) = \frac{2}{m}\mathbb{E}_\sigma \left[ \sup_{h\in\mathcal{H}} |\sum_{i=1}^{m} \sigma_i h(\mathbf{x}_i)|, S = (\mathbf{x}_1, \ldots, \mathbf{x}_m) \right].$$

*L'espérance est calculée sur $\sigma = (\sigma_1, \ldots, \sigma_n)$, où les $\sigma_i$ sont des variables aléatoires uniformes indépendantes prenant leur valeur dans $\{-1, +1\}$. La complexité de Rademacher d'un ensemble d'hypothèses $\mathcal{H}$ est définie comme l'espérance de $\widehat{\mathfrak{R}}_S(\mathcal{H})$ sur tous les ensembles de taille $m$ :*

$$\mathfrak{R}_m(\mathcal{H}) = \mathbb{E}_S[\widehat{\mathfrak{R}}_S(\mathcal{H}), |S| = m].$$

La complexité de Rademacher donne lieu à des bornes en généralisation de la forme suivante.

**Théorème II.3** (Borne en généralisation utilisant la complexité de Rademacher)**.** *Soient $S$ un ensemble d'apprentissage de taille $n$, i.i.d. selon une distribution $\mathcal{D}_S$, $\mathcal{H}$ un espace d'hypothèses et $\delta > 0$. Pour tout $h \in \mathcal{H}$, avec une probabilité de $1 - \delta$ sur l'ensemble aléatoire $S$, on a :*

$$\epsilon^\ell_{\mathcal{D}_S}(h) \leq \epsilon^\ell_S(h) + 2\widehat{\mathfrak{R}}_S(\mathcal{H}) + \sqrt{\frac{2\ln\frac{2}{\delta}}{n}}.$$

Une limitation importante des bornes précédentes est qu'elles dépendent de la famille d'hypothèses considérée, mais nullement de l'algorithme d'apprentissage lui-même. Les deux cadres suivants permettent de s'affranchir de cette restriction.

### II.3.2 La stabilité uniforme

La *stabilité uniforme* [Bousquet and Elisseeff, 2000, Bousquet and Elisseeff, 2002] prend en compte la capacité de résistance de l'algorithme à des variations de l'ensemble d'apprentissage. Plus l'algorithme est stable à la substitution ou la suppression d'un exemple de l'ensemble d'apprentissage, meilleure sera la convergence de l'erreur empirique vers l'erreur en généralisation. Plus formellement, la stabilité uniforme est définie de la manière suivante.

**Définition II.8** (Stabilité uniforme)**.** *Un algorithme $\mathcal{A}$ respecte une stabilité uniforme $\frac{\kappa}{n}$, par rapport à une fonction de perte $\ell$, si la condition suivante est remplie :*

$$\forall S, |S| = n, \forall i \in [n] : \sup_z |\ell(h, z) - \ell(h_i, z)| \leq \frac{\kappa}{n},$$

*où $\kappa$ est une constante positive, $S^i$ est obtenu depuis l'ensemble d'apprentissage $S$, en remplaçant le $i_{ème}$ élément $z_i \in S$ par un autre exemple $z'_i$, distribué i.i.d. selon $\mathcal{D}_S$, $h$ et $h_i$ sont les hypothèses apprises par $\mathcal{A}$ respectivement sur $S$ et $S^i$.*

Cette définition correspond au cas où un exemple d'apprentissage est remplacé par un autre. Une autre définition, que nous ne présenterons pas, existe. Elle correspond au cas où un exemple d'apprentissage est supprimé de $S$.

Il a été montré que de nombreux algorithmes satisfaisaient cette définition. Il a également été prouvé que lorsque la Définition II.8 était respectée, la borne suivante pouvait être dérivée, en $\mathcal{O}(\frac{1}{\sqrt{n}})$.

**Théorème II.4** (Borne de stabilité uniforme)**.** *Soit $S$ un ensemble d'apprentissage de taille $n$, distribué i.i.d. selon une distribution $\mathcal{D}_S$ et $\delta > 0$. Pour n'importe quel algorithme $\mathcal{A}$ de stabilité uniforme $\frac{\kappa}{n}$, par rapport à une fonction de perte $\ell$ bornée par une constante $B$, avec une probabilité $1 - \delta$ sur l'ensemble aléatoire $S$, on a :*

$$\epsilon^{\ell}_{\mathcal{D}_S}(h) \leq \epsilon^{\ell}_{S}(h) + \frac{\kappa}{n} + (2\kappa + B)\sqrt{\frac{\ln(1-\delta)}{2n}},$$

*où $h$ est l'hypothèse apprise par $\mathcal{A}$ sur $S$.*

Il est intéressant de noter que dans ce cadre, le terme de pénalité ne dépend plus directement de la complexité de l'espace d'hypothèses, ce qui permet d'éviter les difficultés rencontrées avec des algorithmes comme les SVMs ou les k-plus proches voisins, ayant pour particularité d'avoir une VC-dimension infinie.

### II.3.3 La robustesse algorithmique

La *robustesse algorithmique* [Xu and Mannor, 2010, Xu and Mannor, 2012] offre d'autres perspectives de généralisation, dépendant cette fois-ci de la capacité de l'algorithme à obtenir une classification similaire sur des exemples géométriquement proches, selon un partitionnement $\mathcal{Z}$ de l'espace. Deux exemples sont considérés comme proches s'ils se trouvent dans la même partition. La notion de robustesse est formalisée par la définition suivante.

**Définition II.9** (Robustesse algorithmique)**.** *Un algorithme $\mathcal{A}$ est $(K, \rho(\cdot))$-robuste, pour $K \in \mathbb{N}$ et $\rho(\cdot) : Z^n \to \mathbb{R}$, si $Z$ peut être partitionné en $K$ ensembles disjoints, notés $\{C_i\}_{i=1}^{K}$, tels que la condition suivante soit valide pour tout $S \in Z^n$ :*

$$\forall z \in S, \forall z' \in Z, \forall i \in [K] : \textit{si } z, z' \in C_i, \textit{alors } |\ell(h, z) - \ell(h, z')| \leq \rho(S),$$

*où $h$ est l'hypothèse apprise par $\mathcal{A}$ à partir de $S$.*

Pour résumer, un algorithme est robuste si pour tout exemple $z'$ se retrouvant dans le même sous-ensemble qu'un exemple d'apprentissage $z$, la différence entre les pertes associées à $z$ et $z'$ est bornée. La Figure II.7 donne une intuition du principe de partitionnement de l'espace. Les travaux

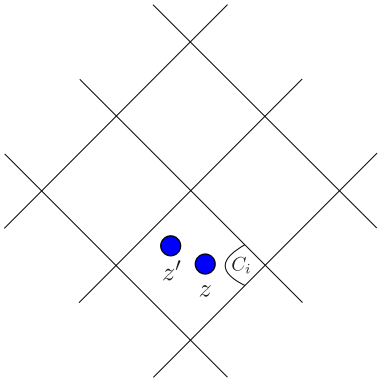



FIGURE II.7 – Illustration du partitionnement de l'espace pour la robustesse algorithmique. L'exemple $z'$, se trouve dans la même partition $C_i$ que l'exemple d'apprentissage $z$. Si l'algorithme est robuste, la différence entre les pertes associées à $z$ et $z'$ est bornée.

de [Xu and Mannor, 2010, Xu and Mannor, 2012] ont montré qu'un algorithme robuste possédait des garanties en généralisation, formalisées par le théorème suivant.

**Théorème II.5** (Borne de robustesse)**.** *Soit $\ell$ une fonction de perte bornée par une constante $B$ et $\delta > 0$. Si un algorithme $\mathcal{A}$ est $(K, \rho(\cdot))$-robuste, alors avec une probabilité $1 - \delta$, on a :*

$$\epsilon^{\ell}_{\mathcal{D}_S}(h) \leq \epsilon^{\ell}_{S}(h) + \rho(S) + B\sqrt{\frac{2K \ln 2 + 2\ln(\frac{1}{\delta})}{n}},$$

*où $h$ est l'hypothèse apprise par $\mathcal{A}$ sur $S$.*

En se basant sur le modèle PAC, de nombreux algorithmes d'apprentissage supervisé ont été introduits. Ceux-ci cherchent à minimiser l'erreur empirique sur un ensemble d'apprentissage $S$, selon certaines conditions et en utilisant une fonction de perte donnée. Comme évoqué précédemment, minimiser l'erreur de classification (la perte 0/1) étant NP-complet, on fait alors appel à des fonctions de perte alternatives, ce qui revient à maximiser la marge $yh(x)$ des exemples d'apprentissage.

Nous présentons dans ce qui suit deux algorithmes, visant explicitement à maximiser les marges des exemples. Le premier est l'algorithme ADABOOST [Freund and Schapire, 1996], issu de la théorie du boosting, sur laquelle repose notre première contribution (voir Chapitre IV). Le second correspond aux machines à vecteurs de support (SVM) [Boser et al., 1992, Cortes and Vapnik, 1995], à l'origine de notre deuxième contribution, présentée dans le Chapitre V.

## II.4 Algorithmes maximisant les marges

### II.4.1 Boosting et AdaBoost

Le boosting [Schapire, 1989] est une méthode ensembliste dont le principe est de combiner itérativement des hypothèses dites faibles pour obtenir un classifieur final fort au sens de la théorie PAC. Deux conditions sont nécessaires à la réussite d'une méthode ensembliste :

- Chacune des hypothèses faibles à combiner doit être meilleure qu'un tirage aléatoire (autrement dit, son erreur empirique doit être inférieure à 0.5 dans le cas binaire).
- Les différentes hypothèses doivent générer de la diversité, c'est-à-dire que leurs erreurs respectives doivent être faites sur des exemples différents.

Partant de ce constat, Freund et Schapire [Freund and Schapire, 1996] ont proposé la définition d'apprenant faible.

**Définition II.10** (Apprenant faible)**.** *Une hypothèse $h_n$ apprise à une itération $n$ est un apprenant faible sur un échantillon d'apprentissage étiqueté $S$, tiré selon $\mathcal{D}_S$, si $h_n$ a un taux de succès au moins un peu meilleur que l'aléatoire, c'est-à-dire $\exists\ \tau_n \in ]0; \frac{1}{2}]$ :*

$$\epsilon_{S^n}(h_n) = \widehat{Pr}_{\mathbf{x}_i \sim D_n^S}[h_n(\mathbf{x}_i) \neq y_i] = \frac{1}{2} - \tau_n,$$

*où $[.]$ est une fonction indicatrice et $D_n^S$ est la distribution empirique correspondant à $S$.*

**Entrée :**

- un ensemble $S$ ($|S| = m$) d'exemples étiquetés,
- un nombre d'itérations $N$.

**Sortie :** un classifieur $H_N$.
Initialisation : $\forall \mathbf{x}_i \in S, D_1(\mathbf{x}_i) = \frac{1}{m}$.
**pour** $n = 1$ **à** $N$ **faire**
  Apprendre $h_n$ satisfaisant les conditions d'apprenant faible (Définition II.10).
  Soit $\alpha_n = \frac{1}{2} ln \frac{1-\epsilon_{S^n}(h_n)}{\epsilon_{S^n}(h_n)}$.
  Règle de mise à jour : $\forall x_i \in S,\ D_{n+1}(\mathbf{x}_i) = D_n(\mathbf{x}_i).\frac{e^{-\alpha_n h_n(\mathbf{x}_i).y_i}}{Z_n}$.
  où $Z_n$ est un coefficient de normalisation.
**fin**
$f_N(\mathbf{x}) = \sum_{n=1}^{N} \alpha_n h_n(\mathbf{x})$.
Classifieur final : $H_N(\mathbf{x}) = signe(f_N(\mathbf{x}))$.

**Algorithme 1** : AdaBoost

L'algorithme de boosting le plus connu, appelé AdaBoost [Freund and Schapire, 1996], est présenté dans l'Algorithme 1.

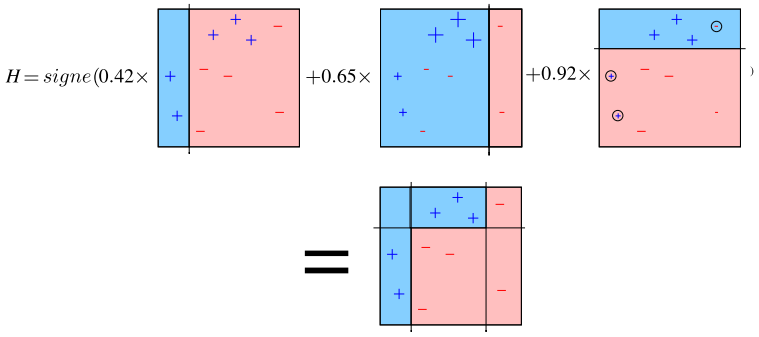



FIGURE II.8 – Illustration du comportement du boosting. En haut, les apprenants faibles obtenus lors des trois itérations et leur poids. En bas, le classifieur final obtenu par la combinaison linéaire des trois séparateurs.

ADABOOST part d'une distribution uniforme sur $S$, puis apprend, à chaque itération, une nouvelle hypothèse $h_n$ satisfaisant la Définition II.10. N'importe quelle classe d'hypothèses peut être utilisée, mais les auteurs ont montré qu'il convenait de choisir une famille de classifieurs peu complexe, afin d'éviter le sur-apprentissage. Il est par exemple courant d'utiliser des arbres de décision à une dimension (aussi appelés stumps). On calcule ensuite le poids $\alpha_n$ qui sera attribué à l'hypothèse $h_n$. Celui-ci dépend de l'erreur empirique de $h_n$ sur $S$ et joue le rôle d'un critère de confiance : plus l'erreur est faible, plus le poids de l'hypothèse (et donc son influence dans la combinaison finale) sera grand. Enfin, les poids des exemples sont mis à jour de la manière suivante à une étape $n$ : les exemples mal étiquetés par $h_n$ voient leur poids augmenter exponentiellement en $n+1$, tandis que les exemples bien classés ont leur poids diminué, ceci dans le but de concentrer les efforts de l'hypothèse suivante sur les exemples mal étiquetés par $h_n$.

Ce processus est répété pour $N$ itérations ($N$ étant un hyperparamètre). Finalement, le classifieur obtenu n'est autre que la combinaison linéaire des $h_n$ appris tout au long des $N$ itérations : $H_N(\mathbf{x}) = signe(\sum_{n=1}^{N} \alpha_n h_n(\mathbf{x}))$, comme illustré par la Figure II.8. Notons que ce classifieur correspond à un hyperplan séparateur dans l'espace de projection des sorties des hypothèses $h_n$.

Au delà de sa simplicité et de son efficacité, une des forces de cet algorithme est qu'il permet la dérivation de nombreuses garanties théoriques, que nous présentons rapidement dans ce qui suit.

**Théorème II.6.** *Soit $\epsilon_S(H_N)$ l'erreur empirique obtenue par le classifieur $H_N$ renvoyé par* ADABOOST *après $N$ itérations sur l'échantillon $S$ des exemples sources étiquetés.*

$$\epsilon_S(H_N) \quad = \quad \widehat{Pr}_{\mathbf{x}_i \sim S}[y_i f_N(\mathbf{x}_i) < 0] \leq \frac{1}{|S|} \sum_{\mathbf{x}_i \sim S} e^{-y_i f_N(\mathbf{x}_i)} \leq \Pi_n Z_n.$$

Ce théorème signifie donc que minimiser l'erreur empirique revient à minimiser le produit des $Z_n$. Schapire montre que cette minimisation est assurée quand le coefficient $\alpha_n = \frac{1}{2}\ln(\frac{1-\epsilon_S(h_n)}{\epsilon_S(h_n)})$

est appliqué à chacun des $h_n$. Le Théorème II.6 montre aussi que l'erreur empirique est approximée par la perte exponentielle vue en Section II.2. Le théorème suivant montre le comportement de l'erreur empirique en fonction du nombre d'itérations.

**Théorème II.7.**

$$\prod_n Z_n = \prod_n (2\sqrt{\epsilon_S(h_n)(1-\epsilon_S(h_n))} = \prod_n \sqrt{1-4\tau_n^2} < \exp(-2\sum_n \tau_n^2),$$

*où* $\epsilon_S(h_n) = \frac{1}{2} - \tau_n$.

Ce théorème signifie que l'erreur empirique décroît exponentiellement vite avec le nombre $N$ d'itérations. Les auteurs ont également dérivé la borne suivante sur l'erreur en généralisation.

**Théorème II.8.** *Soit $\mathcal{H}$ une classe d'hypothèses de VC-dimension $d$. Pour tout $\delta > 0$ et $\theta > 0$, avec une probabilité $1-\delta$, tout ensemble de classifieurs $H_N$, construit depuis $m$ exemples d'apprentissage satisfait :*

$$\epsilon_{\mathcal{D}_S}(H_N) \leq \widehat{Pr}[yf_N(\mathbf{x}) \leq \theta] + \mathcal{O}\left(\sqrt{\frac{d}{m}\frac{\log^2(\frac{m}{d})}{\theta^2} + \log(\frac{1}{\delta})}\right).$$

Il est intéressant de noter que cette borne dépend de la marge des exemples obtenue par $f_N$ et que le terme de complexité implique la VC-dimension de la famille d'hypothèses faibles utilisée (et non pas celle du classifieur final). Le théorème suivant montre que la marge des exemples augmente avec le nombre d'itérations, impliquant donc une diminution de l'erreur en généralisation.

**Théorème II.9.** *Soit $yf_N(\mathbf{x})$, la marge d'un exemple après $N$ itérations d'*AdaBoost.

$$\widehat{Pr}[yf_N(\mathbf{x}) \leq \theta] \leq 2^N \prod_n \sqrt{\epsilon_S(h_n)^{1-\theta}(1-\epsilon_S(h_n)^{1-\theta})}.$$

Si $\epsilon_S(h_n) \leq \frac{1}{2} - \tau_n$ (condition assurée par l'apprenant faible), $\forall \theta < \tau_n$, alors cette borne diminue exponentiellement vite avec le nombre d'itérations $N$. Ceci signifie donc qu'augmenter le nombre d'itérations de l'algorithme n'entraîne pas, comme on aurait pu le craindre, un phénomène de sur-apprentissage mais au contraire une maximisation des marges.

Les avantages de cet algorithme, aussi bien théoriques que pratiques, ont entraîné l'émergence de très nombreux travaux. Sans être exhaustif, on peut citer le boosting pour la classification multi-classes [Li, 2006, Mukherjee and Schapire, 2010, Saberian and Vasconcelos, 2011]. Le travail présenté dans [Janodet et al., 2004] introduit une version du boosting pour l'inférence grammaticale, tandis que celui de [Warmuth et al., 2007] propose une extension de l'algorithme adaptée aux cas où les exemples ne sont pas linéairement séparables. On peut aussi citer [Pardoe and Stone, 2010] qui introduit un algorithme pour la régression dans le cadre de l'apprentissage par transfert, ou encore [Sebban et al., 2002] pour la sélection d'attributs et [Sebban et al., 2001] pour l'optimisation d'algorithmes de type k-plus proches voisins.

### II.4.2 Les machines à vecteurs de support (SVM)

Une des méthodes les plus utilisées, de par ses bonnes performances et une relative adaptabilité à de nombreux cadres (classification multi-classes, régression) est celle dite des machines à vecteurs de support (SVM) [Cortes and Vapnik, 1995]. Nous la présentons ici, pour plus de simplicité, dans le cadre de la classification binaire. L'idée de base derrière cet algorithme est relativement intuitive. Il s'agit de trouver un hyperplan séparateur $h(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + w_0 = \sum_{i=0}^{d} w^i x^i$, dans le cas où $X$ est de dimension $d$. La fonction de décision pour attribuer à un exemple $\mathbf{x}$ sa classe estimée dépend donc simplement du signe de $h(\mathbf{x})$.

Cet hyperplan séparateur est contraint de manière à ce que la marge soit maximale. La marge d'un séparateur est définie par la marge minimale sur tous les exemples.

Chercher l'hyperplan séparateur de marge maximale offre plusieurs avantages. Tout d'abord, celui-ci a toutes les chances d'obtenir une bonne performance en généralisation. Ensuite, il existe un unique hyperplan séparateur de marge maximale, pour un $S$ donné, qui peut être trouvé efficacement par la résolution du problème convexe suivant :

$$\min_{\mathbf{w}, w_0} \frac{1}{2}||\mathbf{w}||^2,$$

$$\text{avec } y_i(\mathbf{w}^T \mathbf{x}_i + w_0) \geq 1; \forall (\mathbf{x}_i, y_i) \in S.$$

Cette résolution devient néanmoins difficilement envisageable dans sa forme primale, dès lors que la dimension d'entrée dépasse un certain seuil. Pour contourner cette limitation, il est possible de recourir à la forme duale du problème, celle-ci permettant en outre une reformulation à base de produits scalaires :

$$\max_{\alpha} [\sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle], \text{ avec } \alpha_i \geq 0, i = 1 \ldots m, \sum_{i=1}^{m} \alpha_i y_i = 0$$

L'hyperplan solution est alors $h^\star(\mathbf{x}) = (\mathbf{w}^\star \mathbf{x}) + w_0^\star = \sum_{i=1}^{m} \alpha_i^\star y_i \langle \mathbf{x}_i, \mathbf{x} \rangle + w_0^\star$.

Ici, on cherche donc les multiplicateurs de Lagrange $\alpha_i$. Les points auxquels sont associés des $\alpha$ non nuls sont appelés **vecteurs de support**. Ce sont les seuls qui déterminent l'hyperplan optimal et ce sont donc les seuls utilisés pour la classification d'un nouvel exemple, qui ne requiert que le calcul de produits scalaires entre des vecteurs de l'espace d'entrée. La solution, en termes de nombre de paramètres, ne dépend donc plus de la dimension de l'espace d'entrée, mais du nombre de vecteurs de support, généralement bien inférieur au nombre d'exemples d'apprentissage. Une illustration du résultat de cet algorithme est présenté dans la Figure II.9.

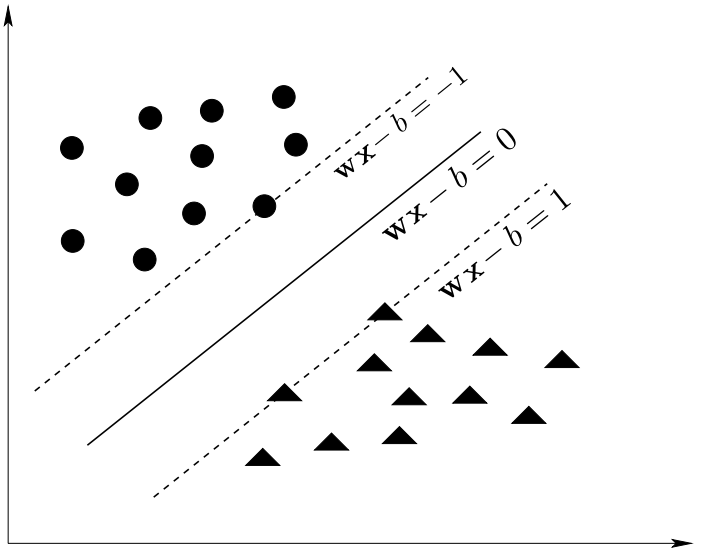



FIGURE II.9 – Illustration d'un hyperplan séparateur obtenu avec les machines à vecteurs de support.

Il est important de noter que la formulation précédente suppose que les exemples des deux classes soient linéairement séparables, ce qui est rarement le cas. Pour remédier à cela, l'approche par fonctions noyaux a été introduite. L'idée est de considérer une fonction $\phi$ non linéaire de l'espace d'entrée $X$, permettant de projeter les exemples dans un nouvel espace, potentiellement infini, où ils seront linéairement séparables.

$$\mathbf{x} = (x^1, \ldots, x^d)^T \mapsto \phi(\mathbf{x})^T = (\phi_1(\mathbf{x}), \ldots, \phi_d(\mathbf{x}), \ldots)^T.$$

Le classifieur, dans sa forme duale, ne s'exprime que sous la forme de produits scalaires entre les exemples. Pour définir un classifieur dans le nouvel espace induit par $\phi$, il suffit donc de connaître le produit scalaire dans cet espace et pas nécessairement la projection elle-même. Les fonctions noyaux correspondent en fait au calcul des produits scalaires dans un nouvel espace de projection. Le problème de détermination de la bonne transformation non linéaire $\phi$ revient donc à celui du choix d'une bonne fonction noyau $K : X \times X \longrightarrow \mathbb{R}$, telle que $K(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$, celle-ci devant être symétrique et positive semi-définie. Dans ce cas, calculer le produit scalaire de deux exemples dans l'espace de reprojection revient à évaluer la valeur de la fonction noyau entre ces deux mêmes exemples. Il en découle le problème d'optimisation suivant :

$$\max_{\alpha} [\sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)], \text{ avec } \alpha_i \geq 0, i = 1 \ldots m, \sum_{i=1}^{m} \alpha_i y_i = 0$$

L'hyperplan solution est donc $h(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i^\star y_i K(\mathbf{x}, \mathbf{x}_i) + w_0^\star$.

Cette astuce permet donc de séparer des exemples qui ne sont pas linéairement séparables dans leur espace d'origine, à l'aide d'une reprojection implicite, induite par la fonction noyau. La classification d'un nouvel exemple $\mathbf{x}'$ ne nécessite donc ensuite que l'évaluation de la valeur de la fonction noyau $K(\mathbf{x}_i, \mathbf{x}')$, pour tous les $\mathbf{x}_i \in S$, simulant ainsi le calcul du produit scalaire dans l'espace de reprojection.

Des variantes de cette formulation ont été introduites dans la littérature. On peut par exemples citer le SVM à norme 1 [Zhu et al., 2003], permettant d'induire des modèles parcimonieux. Le problème de la classification multi-classes a également été abordé dans [Bordes et al., 2007]. Des approches étudiant les garanties théoriques offertes par cet algorithme ont aussi été proposées, comme dans [Xu et al., 2009], qui travaille sur la robustesse des machines à vecteurs de support. D'autre part, [Joachims, 1999] introduit le *Transductive SVM*, utilisant des données tests non étiquetées durant la phase d'apprentissage, afin de minimiser l'erreur sur cet ensemble particulier. Enfin, [Ladicky and Torr, 2011] introduit le SVM localement linéaire.

Bien que les SVMs soient très populaires et performants, le choix de la fonction noyau reste un problème ouvert. Celui-ci dépend fortement de la tâche considérée et influe grandement sur le résultat. De plus, les contraintes de symétrie et de semi-définie positivité de $K$ empêchent parfois l'exploitation de fonctions de similarité assez naturelles pour une application donnée. Nous présentons dans le Chapitre V une généralisation de cette approche qui s'affranchit de ces contraintes, appelée théorie des $(\varepsilon, \gamma, \tau)$-bonnes fonctions de similarité [Balcan and Blum, 2006, Balcan et al., 2008]. Nous nous basons ensuite sur cette théorie pour proposer une des contributions de cette thèse.

Les algorithmes de boosting et les SVMs, comme la majorité des algorithmes d'apprentissage supervisé, se positionnent dans le cadre PAC et suivent l'hypothèse selon laquelle les exemples d'apprentissage sont issus de la même distribution que les exemples tests sur lesquels le modèle est destiné à être appliqué. L'erreur empirique, calculée sur un ensemble d'apprentissage $S$, est donc censée converger vers l'erreur en généralisation avec l'augmentation du nombre d'exemples.

Or, dans de très nombreux cas, cette hypothèse n'est pas vérifiée pour différentes raisons :

- Soit les données ont évolué avec le temps, entraînant une variation des distributions statistiques entre l'apprentissage et le test.
- Soit les conditions d'acquisition des données tests sont différentes de celles d'apprentissage.
- Soit encore, la tâche de classification elle-même a changé entre l'apprentissage et le test.

Pour toutes ces raisons, un nouvel axe de recherche a émergé ces dernières années : l'adaptation de domaine. Des travaux théoriques ont été proposés, présentant un cadre formel permettant le développement d'approches pour remédier à ces problèmes.

Dans le chapitre suivant, après avoir introduit le cadre de l'adaptation de domaine, nous présentons les travaux théoriques majeurs sur le sujet. Par la suite, nous faisons état des principales

approches algorithmiques de la littérature, inspirées de ces cadres théoriques. Celles-ci se répartissent en trois catégories : les approches de repondération, les méthodes de reprojection et les algorithmes d'auto-étiquetage.

# Chapitre III

## État de l'Art en Adaptation de Domaine

**Résumé :** Dans ce chapitre, nous présentons de manière formelle le cadre de l'adaptation de domaine. Nous introduisons dans un premier temps les travaux théoriques, proposant notamment des bornes en généralisation dépendant de l'erreur empirique sur le domaine source et d'une mesure de divergence entre les deux distributions. Par la suite, nous présentons des algorithmes de l'état de l'art, visant à réduire ces bornes et répartis selon trois catégories : les approches de repondération, les méthodes de reprojection et les algorithmes d'auto-étiquetage.

### III.1 Introduction

Après avoir introduit le cadre "classique" de l'apprentissage supervisé et mis en avant ses limitations dans certaines applications de la vie réelle, nous nous focalisons maintenant sur l'adaptation de domaine. Cet axe de recherche est un sous-domaine de *l'apprentissage par transfert* [Pan and Yang, 2010]. Le principe de l'apprentissage par transfert, s'inspirant de l'apprentissage chez l'humain, est d'utiliser des connaissances acquises sur un domaine (ou une tâche) source donné(e) pour apprendre un modèle sur un domaine ou une tâche cible différent(e).

S'il est assez intuitif d'admettre qu'utiliser des connaissances sur la source ne peut que renforcer le nouveau modèle, sous réserve que les deux tâches soient proches, la mise en oeuvre algorithmique de cette idée est difficile à réaliser. L'apprentissage par transfert regroupe plusieurs situations. Selon la catégorisation de [Pan and Yang, 2010] (voir Figure III.1), on peut distinguer :

- L'*apprentissage par transfert inductif*, où l'on considère que les deux domaines source et cible sont les mêmes, mais que les tâches sont différentes. Un exemple pourrait être d'apprendre deux modèles à partir d'images issues d'un seul corpus (domaine unique), un devant définir si une personne est présente sur l'image (première tâche), l'autre devant définir si un animal est présent sur l'image (deuxième tâche). L'idée sous-jacente est que, les tâches étant reliées, le modèle reconnaissant les personnes doit nous permettre de faciliter l'apprentissage du modèle de détection d'animaux.

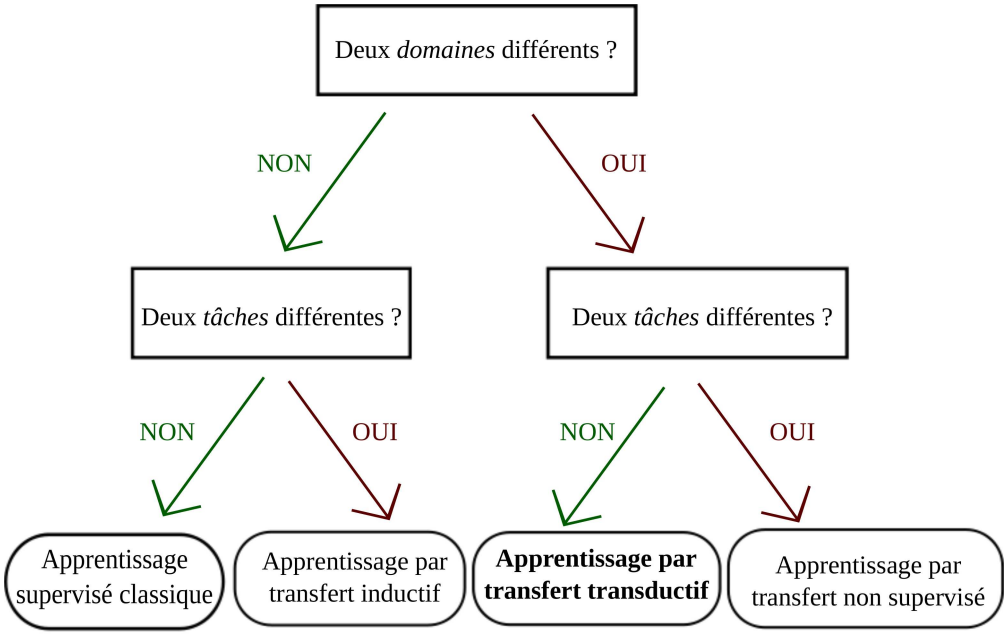
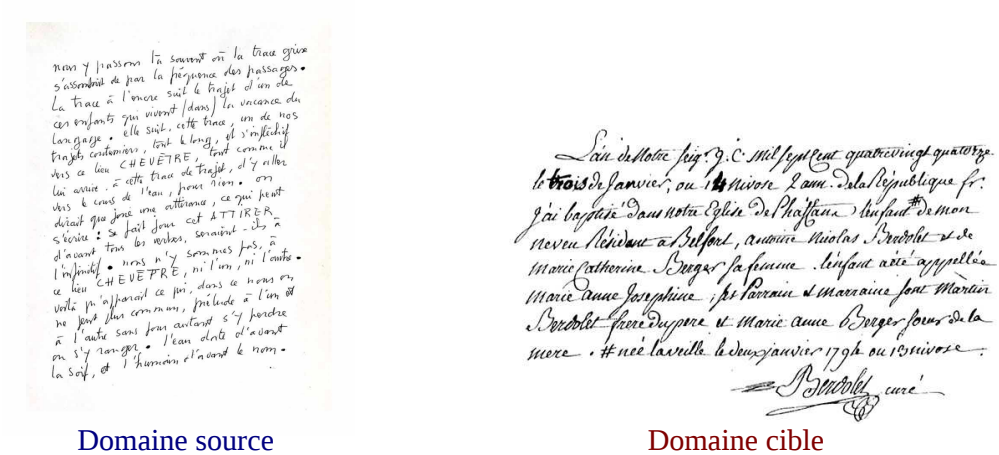



FIGURE III.1 – Représentation des différents cadres en apprentissage par transfert.

- L'*apprentissage par transfert transductif*, où l'on considère que les tâches sont identiques, mais que les domaines sont différents. C'est le cadre de l'adaptation de domaine. Une illustration peut être donnée sur la reconnaissance de caractères manuscrits. On cherche à apprendre un modèle qui reconnaît automatiquement les différents caractères à partir de textes manuscrits écrits à notre époque (domaine source). On souhaite ensuite que ce modèle soit capable de reconnaître des caractères extraits de textes manuscrits du Moyen-âge (domaine cible/test). De nouveau, l'intérêt d'utiliser les connaissances acquises sur la source est assez clair, les domaines étant liés bien que différents. Ce problème est illustré par la Figure III.2
- L'*apprentissage par transfert non supervisé*, où l'on considère qu'à la fois les domaines et les tâches à réaliser sont différents. Pour reprendre l'exemple de détection de personne, on pourrait imaginer tirer parti de la tâche source (apprendre un modèle détectant la présence d'une personne dans une image) pour réaliser la tâche cible (apprendre un modèle détectant la présence d'un animal dans une image). Mais cette fois-ci, les images sources et cibles seraient issues de deux corpus différents.

L'adaptation de domaine est équivalente à l'apprentissage par transfert transductif. Ce cadre considère que l'apprenant dispose de données sources étiquetées et de données cibles peu ou pas étiquetées. Même si certaines données cibles sont labelisées, on considère qu'elles sont en trop petit nombre pour autoriser directement l'apprentissage d'un modèle : l'adaptation est donc nécessaire. Dans cette thèse, nous nous focalisons sur la version la plus complexe de l'adaptation de domaine,



Domaine source Domaine cible

FIGURE III.2 – Illustration de deux distributions différentes dans le cadre de la reconnaissance de caractères manuscrits. Le domaine source contient des caractères issus de textes écrits à notre époque, tandis que le domaine cible contient des caractères issus de textes écrits au Moyen-âge.

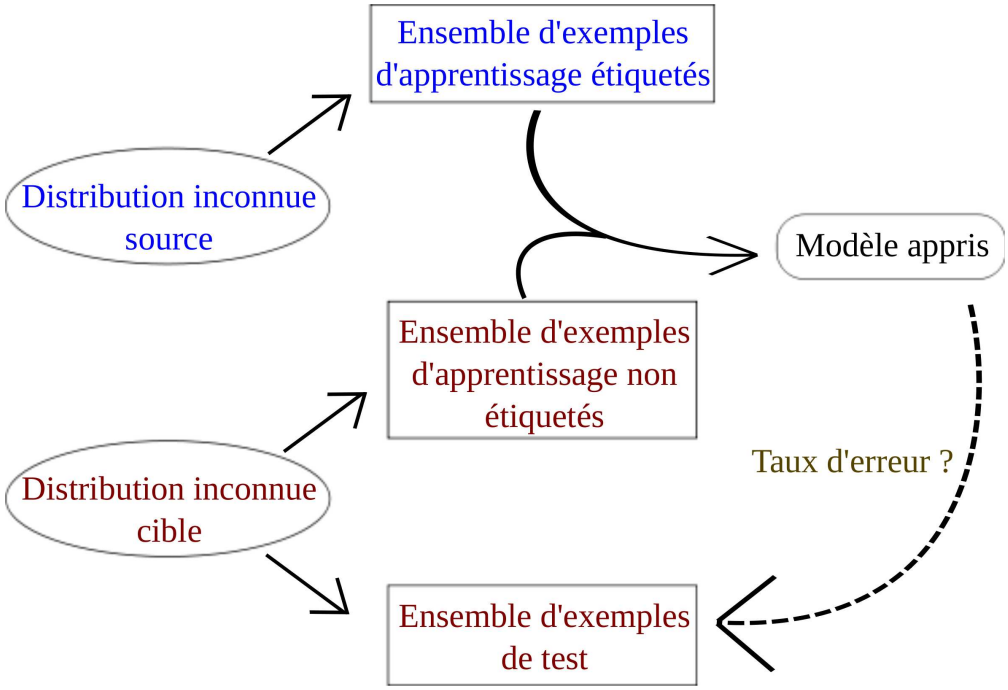



FIGURE III.3 – Illustration simplifiée du problème de l'adaptation de domaine non supervisée.

où les données cibles sont toutes non étiquetées. Ce cadre est habituellement baptisé *adaptation de domaine non supervisée*. Une illustration simple est donnée par la Figure III.3.

Imaginons une tâche consistant à apprendre un séparateur, disposant d'un e-mail en entrée, et lui attribuant une étiquette "spam" ou "non-spam". Le modèle appris à partir des mails d'un individu ne sera pas forcément efficace sur la boîte mail d'un individu différent. En effet, suivant

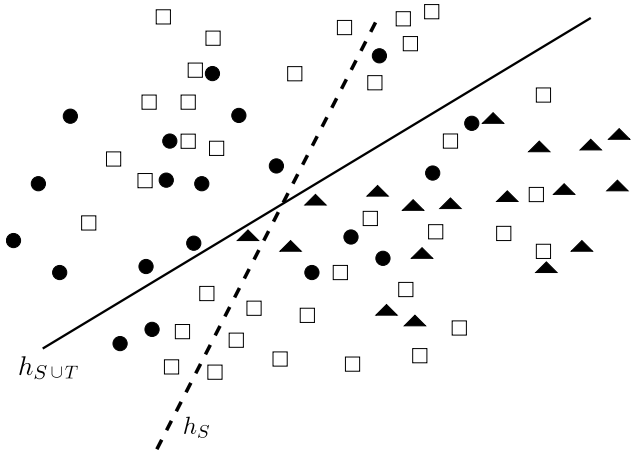



FIGURE III.4 – Illustration simple en deux dimensions d'un problème d'adaptation de domaine. Les points sources, en noir, sont étiquetés cercle ou triangle. $h_S$ correspond au classifieur appris seulement à partir de ces exemples. Les points cibles non étiquetés, en blanc, sont représentés par des carrés et ne suivent pas exactement la même distribution que les points sources. Afin d'être performant sur les carrés, le classifieur doit s'adapter en tenant compte des données non étiquetées, pour obtenir $h_{S\cup T}$.

l'usage de la boîte mail (personnel ou professionnel) et la définition même de "spam" qui peut diverger selon les personnes, il est difficile de s'assurer que les exemples d'apprentissage et les données sur lesquelles on utilisera le modèle sont issues de la même distribution.

En AD, l'ensemble d'apprentissage est défini comme suit.

**Définition III.1** (Ensemble d'apprentissage en AD)**.** *Un ensemble d'apprentissage de taille $n$ dans le cas de l'AD est un ensemble $S \cup T$ composé de deux sous-ensembles $S = \{z_i = (x_i, y_i)\}_{i=1}^{|S|}$ et $T = \{x_j\}_{j=1}^{|T|}$, tels que $|S| + |T| = n$. Les observations sources de $S$ (respectivement cibles de $T$) sont i.i.d. selon une distribution jointe inconnue $\mathcal{D}_S$ (respectivement $\mathcal{D}_T^X$) sur l'espace $Z = X \times Y$ (respectivement $X$), où $X$ représente l'espace d'entrée et $Y$ l'espace de sortie.*

L'erreur réelle est définie de la manière suivante.

**Définition III.2** (Erreur réelle dans le cas de l'AD)**.** *L'erreur réelle $\epsilon^{\ell}_{\mathcal{D}_T}(h)$ d'une hypothèse $h$ selon une fonction de perte $\ell$ correspond à l'espérance de la perte de $h$ sur la distribution $\mathcal{D}_T$ :*

$$\epsilon^{\ell}_{\mathcal{D}_T}(h) = \mathbb{E}_{z\sim\mathcal{D}_T}[\ell(h, z)].$$

Alors que l'objectif est de minimiser l'erreur sur la distribution $\mathcal{D}_T$, rappelons que les exemples distribués selon la cible dont on dispose dans l'ensemble d'apprentissage sont non étiquetés. C'est pourquoi l'adaptation de domaine consiste à s'appuyer sur les exemples étiquetés de $S$ **et dans le même temps les exemples non étiquetés de** $T$, afin d'obtenir le modèle le plus performant possible sur $\mathcal{D}_T$, domaine différent de $\mathcal{D}_S$, comme illustré par la Figure III.4.

Le problème de l'AD se pose dans de nombreux domaines, parmi lesquels le traitement naturel de la langue [Rosenfeld, 1996, Roark and Bacchiani, 2003, Blitzer et al., 2007, Daumé III et al., 2010], la reconnaissance d'images [Martinez, 2002, Kulis et al., 2011] ou encore le multimédia [Duan et al., 2009, Duan et al., 2012, Roy et al., 2012]. De nombreuses contributions ont été proposées, basées sur différentes approches, certaines privilégiant l'aspect théorique, d'autres plus axées sur un point de vue pratique. Dans ce chapitre, nous passons en revue les principales catégories de méthodes et présentons quelques-unes des contributions majeures du domaine. Pour plus de détails, le lecteur pourra se référer aux différents travaux suivants : [Pan and Yang, 2010] présentant la majorité des contributions en apprentissage par transfert et [Jiang, 2008, Margolis, 2011] où l'on peut retrouver bon nombre des approches introduites en adaptation de domaine.

Nous présentons dans un premier temps les contributions théoriques faites dans ce domaine. Celles-ci se présentent sous la forme de bornes sur l'erreur réelle et se basent souvent sur une mesure de divergence. Les contributions algorithmiques, que nous présentons dans un second temps, introduisent des approches tirant parti ce ces études théoriques.

## III.2 Cadres théoriques de l'adaptation de domaine

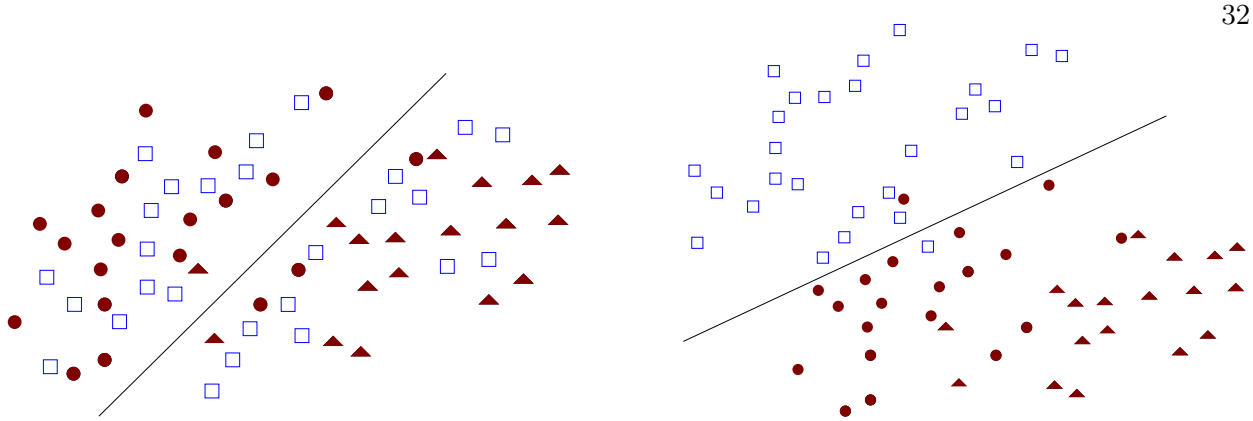
Le modèle PAC classique n'étant plus directement utilisable dans le cadre de l'adaptation de domaine, un certain nombre de travaux ont porté sur la définition de nouveaux cadres théoriques établissant les conditions permettant à un algorithme d'adapter correctement. Les premières contributions, présentées dans [Ben-David et al., 2006, Ben-David et al., 2010], généralisent le cadre PAC en intégrant notamment une notion de mesure de divergence entre distributions. Cette dernière est appelée $\mathcal{H}$-divergence.

**Définition III.3** ($\mathcal{H}$-divergence)**.** *Étant donné un espace d'entrée $X$, et $\mathcal{D}$ et $\mathcal{D}'$ deux distributions de probabilités sur $X$, étant donnée $\mathcal{H}$ une classe d'hypothèses sur $X$. Soit $I(h)$ l'ensemble pour lequel $h \in \mathcal{H}$ est la fonction caractéristique, i.e., $x \in I(h) \Leftrightarrow h(x) = 1$, la $\mathcal{H}$-divergence entre $\mathcal{D}$ et $\mathcal{D}'$ est calculée de la façon suivante :*

$$d_{\mathcal{H}}(\mathcal{D}, \mathcal{D}') = 2 \sup_{h \in \mathcal{H}} |Pr_{\mathcal{D}}[I(h)] - Pr_{\mathcal{D}'}[I(h)]|.$$

Cette mesure de divergence présente deux avantages par rapport aux mesures classiques comme la $L_1$ [1] par exemple. Premièrement, pour les classes d'hypothèses $\mathcal{H}$ de VC-dimension finie, la $\mathcal{H}$-divergence peut être estimée depuis un ensemble fini. Deuxièmement, la $\mathcal{H}$-divergence n'est jamais plus grande que la $L_1$, pour n'importe quelle classe $\mathcal{H}$, et est même généralement plus petite. Enfin, sa valeur dépend de la classe d'hypothèses $\mathcal{H}$ considérée, elle permet donc de savoir à quel

1. La distance $L_1$ entre deux distributions $\mathcal{D}$ et $\mathcal{D}'$ est définie de la manière suivante : $d_{L_1}(\mathcal{D}, \mathcal{D}') = 2 \sup_{B \in \mathcal{B}} |Pr_{\mathcal{D}}[B] - Pr_{\mathcal{D}'}[B]|$, où $\mathcal{B}$ est l'ensemble des sous-ensembles mesurables sur $\mathcal{D}$ et $\mathcal{D}'$



(a) (b)

FIGURE III.5 – Illustration du principe de calcul de la $\mathcal{H}$-divergence empirique. Tous les points sources, quelle que soit leur étiquette, sont en rouge, tandis que les points cibles sont en bleu. Un séparateur est ensuite appris pour discriminer les points sources et cibles. Dans la figure (a), ceux-ci sont difficilement séparables, car géométriquement proches. La $\mathcal{H}$-divergence empirique sera donc faible. À l'inverse, dans la figure (b), les points cibles et sources sont éloignés (ils semblent donc suivre deux distributions très différentes). Dans ce cas, le taux d'erreur du séparateur sera faible, entraînant une valeur élevée de la divergence.

point les hypothèses de $\mathcal{H}$ permettent de différencier les distributions. Le lemme suivant borne la véritable divergence par rapport à sa valeur empirique.

**Lemme III.1.** *Soit $\mathcal{H}$ une classe d'hypothèses sur $X$ de VC-dimension $d$. Si $D$ et $D'$ sont deux ensembles de taille $m$, respectivement distribués selon $\mathcal{D}$ et $\mathcal{D}'$, et si $\hat{d}_{\mathcal{H}}(D, D')$ est la $\mathcal{H}$-divergence empirique entre $D$ et $D'$, alors pour n'importe quel $\delta \in [0, 1]$, avec une probabilité d'au moins $1 - \delta$ :*

$$d_{\mathcal{H}}(\mathcal{D}, \mathcal{D}') \leq \hat{d}_{\mathcal{H}}(D, D') + 4\sqrt{\frac{d \log(2m) + \log \frac{2}{\delta}}{m}}.$$

Ce lemme montre donc que la $\mathcal{H}$-divergence empirique converge vers la véritable $\mathcal{H}$-divergence pour des classes d'hypothèses de VC-dimension finie. Dans ce travail, les auteurs montrent également qu'il est possible de calculer facilement la $\mathcal{H}$-divergence empirique, et ceci sans avoir recours aux étiquettes, ce qui est donc un point essentiel en adaptation de domaine.

L'idée est la suivante. Il s'agit d'étiqueter à $-1$ tous les exemples sources et à $+1$ tous les exemples cibles. Ensuite, un séparateur $h \in \mathcal{H}$ est appris pour discriminer au mieux ces deux classes. Ce principe est illustré par la Figure III.5. La $\mathcal{H}$-divergence empirique peut être directement calculée depuis le taux d'erreur de $h$. Intuitivement, plus l'erreur est petite, plus il est facile de séparer les exemples sources des exemples cibles et donc, plus la divergence entre les deux domaines est grande. Nous pouvons maintenant introduire l'une des principales bornes en généralisation en adaptation de domaine, tirant parti de la notion de divergence empirique $\hat{d}_{\mathcal{H}}$.

**Théorème III.1.** *Soit $\mathcal{H}$ une classe d'hypothèses de VC-dimension $d$. Si $S$ et $T$ sont deux ensembles, chacun de taille $m$, distribués respectivement selon $\mathcal{D}_S$ et $\mathcal{D}_T$, alors pour n'importe quel $\delta \in [0, 1]$, avec une probabilité d'au moins $1 - \delta$, pour tout $h \in \mathcal{H}$ :*

$$\epsilon_{\mathcal{D}_T}(h) \leq \epsilon_{\mathcal{D}_S}(h) + \frac{1}{2}\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(S, T) + 4\sqrt{\frac{2d \log(2m) + \log \frac{2}{\delta}}{m}} + \lambda.$$

Cette borne est définie lorsque la fonction de perte considérée est la $\ell_{0/1}$. Elle permet d'avoir une estimation de l'erreur en généralisation d'une hypothèse sur le domaine cible de manière très simple. Dans cette formule, $\lambda = \epsilon_{\mathcal{D}_S}(h^\star) + \epsilon_{\mathcal{D}_T}(h^\star)$, avec $h^\star = \min_{h\in\mathcal{H}} \epsilon_{\mathcal{D}_S}(h) + \epsilon_{\mathcal{D}_T}(h)$, représente l'erreur combinée de l'hypothèse jointe idéale, qui correspond à l'hypothèse de $\mathcal{H}$ qui minimise l'erreur conjointement aux deux domaines. Cette quantité $\lambda$ apporte une information théorique importante. En effet, elle donne une indication sur les cas où l'adaptation est possible, voire envisageable. Un $\lambda$ élevé rend la borne très lâche, ce qui ne permet pas de garantir que l'adaptation donnera un résultat correct. L'utilisation d'un algorithme d'AD dans ce cas peut même entraîner un transfert négatif. En revanche, si $\lambda$ est faible, signifiant ainsi qu'il existe un classifieur performant sur les deux domaines à la fois, on peut espérer adapter efficacement.

Il est intéressant de noter que si les deux distributions $\mathcal{D}_S$ et $\mathcal{D}_T$ sont les mêmes, la borne est équivalente à une borne PAC standard (voir Section II.3). Lorsque ce n'est pas le cas, et que la valeur de $\lambda$ est faible, réduire cette borne revient donc à réduire algorithmiquement la divergence entre les deux domaines, tout en conservant une hypothèse avec une faible erreur sur la source. Cette idée est à l'origine des algorithmes que nous présentons ultérieurement dans ce chapitre.

À la suite de ces travaux, Mansour et al. [Mansour et al., 2009] ont introduit une nouvelle distance entre distributions, inspirée de celle présentée dans le travail pionnier [Ben-David et al., 2006]. Celle-ci, appelée *discrepancy distance*, peut être utilisée pour des tâches plus générales comme la régression et est définie de la manière suivante.

**Définition III.4** (Discrepancy distance)**.** *Soit $\mathcal{H}$ un ensemble de fonctions de $X$ vers $Y$ et soit $L : Y\times Y \to \mathbb{R}_+$ une fonction de perte sur $Y$, avec $\mathcal{L}_\mathcal{D}(f,g) = \mathbb{E}_{x\sim\mathcal{D}}[L(f(x),g(x))]$, pour toutes fonctions $f, g : X \to Y$ et toute distribution $\mathcal{D}$. La discrepancy distance $disc_L$ entre deux distributions $\mathcal{D}$ et $\mathcal{D}'$ sur $X$ se définit comme :*

$$disc_L(\mathcal{D},\mathcal{D}') = \max_{h,h'\in\mathcal{H}} |\mathcal{L}_\mathcal{D}(h',h) - \mathcal{L}_{\mathcal{D}'}(h',h)|.$$

La discrepancy distance est donc symétrique, elle respecte l'inégalité triangulaire quelle que soit la fonction de perte utilisée, cependant elle ne définit pas une distance[2] , dans la mesure où on peut avoir $disc_L(\mathcal{D},\mathcal{D}') = 0$, avec $\mathcal{D} \neq \mathcal{D}'$. Il est intéressant de noter que dans le cas de la $\ell_{0/1}$, cette distance coïncide avec la $\mathcal{H}$-divergence.

Une fois de plus, les intérêts de cette mesure de divergence sont multiples. À la fois car on peut calculer sa valeur empirique depuis des ensembles finis, lorsque la VC-dimension de la famille d'hypothèses est finie et également parce que celle-ci peut être bornée par rapport à sa valeur empirique pour une fonction de perte bornée, à l'aide de la complexité de Rademacher de la classe de fonctions utilisée.

Le corollaire suivant montre que la discrepancy distance peut être estimée depuis des échantillons finis.

2. Une mesure $d$ est une distance (ou métrique) si elle respecte trois conditions : (i) $d(x,y) = 0$ si et seulement si $x = y$, (ii) $d(x,y) = d(y,x)$ (symétrie) et (iii) $d(x,z) \leq d(x,y) + d(y,z)$ (inégalité triangulaire)

**Corollaire III.1.** *Soit $\mathcal{H}$ un ensemble d'hypothèses borné par $M > 0$ pour une fonction de perte $L$ : $L(h, h') \leq M, \forall h, h' \in \mathcal{H}$. Soit $\mathcal{D}_S$ (respectivement $\mathcal{D}_T$) une distribution sur $X$ et $\widehat{\mathcal{D}_S}$ (respectivement $\widehat{\mathcal{D}_T}$) la distribution empirique correspondante pour un ensemble $S$ (respectivement $T$). Alors, pour tout $\delta > 0$, avec une probabilité d'au moins $1 - \delta$ sur les ensembles $S$ de taille $m$ distribués selon $\mathcal{D}_S$ et les ensembles $T$ de taille $n$ distribués selon $\mathcal{D}_T$ :*

$$disc_L(\mathcal{D}_S, \mathcal{D}_T) \leq disc_L(\widehat{\mathcal{D}_S}, \widehat{\mathcal{D}_T}) + 4\left(\widehat{\mathfrak{R}}_S(\mathcal{H}) + \widehat{\mathfrak{R}}_T(\mathcal{H})\right) + 3M\left(\sqrt{\frac{\log\frac{4}{\delta}}{2m}} + \sqrt{\frac{\log\frac{4}{\delta}}{2n}}\right).$$

Exploitant le corollaire précédent, Mansour et al. dérivent la borne en généralisation suivante.

**Théorème III.2.**

$$\epsilon_{\mathcal{D}_T}(h) \leq \epsilon_{\mathcal{D}_T}(h^\star_{\mathcal{D}_T}) + \mathcal{L}_{\mathcal{D}_S}(h, h^\star_{\mathcal{D}_S}) + disc(\mathcal{D}_S, \mathcal{D}_T) + \lambda',$$

où $h^\star_{\mathcal{D}_S}$ (respectivement $h^\star_{\mathcal{D}_T}$) représente l'hypothèse de $\mathcal{H}$ qui minimise l'erreur sur $\mathcal{D}_S$ (respectivement $\mathcal{D}_T$). $\lambda' = \mathcal{L}_{\mathcal{D}_S}(h^\star_{\mathcal{D}_S}, h^\star_{\mathcal{D}_T})$ est là aussi un terme théorique, qui donne une indication sur les cas où l'adaptation est possible. Le terme $\epsilon_{\mathcal{D}_T}(h^\star_{\mathcal{D}_T})$ étant lui aussi impossible à estimer, les deux termes à minimiser pour réduire la borne sont $\mathcal{L}_{\mathcal{D}_S}(h, h^\star_{\mathcal{D}_S})$ et $disc(\mathcal{D}_S, \mathcal{D}_T)$. Autrement dit, l'idée sous-jacente, une fois de plus, est de minimiser l'erreur de $h$ sur les exemples issus de $\mathcal{D}_S$, tout en diminuant la discrepancy distance entre les deux distributions.

Dans leur travail, les auteurs présentent une comparaison de leur borne avec celle de Ben-David et al. [Ben-David et al., 2010]. Ils y expliquent notamment que sur les quatre termes de leur borne, seulement un implique la fonction d'étiquetage, contre trois dans la borne basée sur la $\mathcal{H}$-divergence. Un cas extrême y est présenté, lorsqu'il n'existe qu'une seule hypothèse $h$ dans $\mathcal{H}$ et une seule fonction d'étiquetage $f$. Dans ce cas, leur borne est égale à $\epsilon_{\mathcal{D}_T}(h) + disc(\mathcal{D}_S, \mathcal{D}_T)$, tandis que celle de Ben-David équivaut à $2\epsilon_{\mathcal{D}_S}(h) + \epsilon_{\mathcal{D}_T}(h) + disc(\mathcal{D}_S, \mathcal{D}_T)$. Même si ce cas est extrême, les auteurs affirment qu'un rapport de 3 peut apparaître entre les deux bornes dans des situations beaucoup plus réalistes, notamment lorsque la distance entre la fonction d'étiquetage et la classe d'hypothèses est significative.

Dans un travail récent, Mansour et al. [Mansour and Schain, 2012] utilisent les notions de robustesse ainsi que de $\lambda$-shift pour introduire une nouvelle borne.

La notion de robustesse garantit que l'écart entre deux exemples géométriquement proches n'est pas trop important (par rapport à $\rho(S)$, voir Définition II.9). Les auteurs introduisent également la notion de $\lambda$-shift à l'aide de la définition suivante.

**Définition III.5** ($\lambda$-shift)**.** *$\mathcal{D}$ est $\lambda$-shift par rapport à $\mathcal{D}'$, noté $\mathcal{D} \in \lambda(\mathcal{D}')$, si $\forall y \in Y$, on a $Pr_{\mathcal{D}}[y] \leq Pr_{\mathcal{D}'}[y] + \lambda(1 - Pr_{\mathcal{D}'}[y])$ et $Pr_{\mathcal{D}}[y] \geq Pr_{\mathcal{D}'}[y](1 - \lambda)$.*

Le $\lambda$-shift correspond donc à la probabilité de changement d'étiquette entre les deux distributions, dans le sens où cette probabilité ne change pas à $\lambda$ près. Les auteurs utilisent ensuite ces notions dans le cadre de l'adaptation de domaine. Ils montrent que la distribution cible est $\lambda$-shift par rapport à la distribution source selon une partition de l'espace d'entrée $X$, si dans chaque région $X_i$, la probabilité conditionnelle cible des étiquettes $Pr_{\mathcal{D}_T}[y|x \in X_i]$ est $\lambda$-shift par rapport à la probabilité conditionnelle source des étiquettes $Pr_{\mathcal{D}_S}[y|x \in X_i]$. Si la distribution source est $\lambda$-shift par rapport à la distribution cible, les probabilités de changement d'étiquette entre les deux domaines sont faibles. Les auteurs définissent par la suite une borne sur la perte moyenne maximale de $h$, appris sur $S \sim \mathcal{D}_S$, dans une région $X_i$, pour une distribution $\mathcal{D}'$, $\lambda$-shift par rapport à $\mathcal{D}_S$, de la manière suivante :

$$l_S^{\lambda}(h, X_i) \leq \max_y \{l_i^y \overline{\lambda}^y(\mathcal{D}_i) + \sum_{y' \neq y} l_i^{y'} \underline{\lambda}^{y'}(\mathcal{D}_i)\},$$

où $\overline{\lambda}^y(\mathcal{D}')$ (respectivement $\underline{\lambda}^y(\mathcal{D}')$) représente la borne supérieure (respectivement inférieure) de la probabilité de l'étiquette $y$ pour $\mathcal{D}'$, et où $l_i^y$ est définie ainsi :

$$l_i^y = \begin{cases} \max_{s \in S \cap X_i \times y} l(h, s) \text{ si } S \cap X_i \times y \neq \emptyset \\ M \text{ sinon} \end{cases}$$

Enfin, les auteurs dérivent une borne sur l'erreur en généralisation sur le domaine cible, pour une hypothèse inférée par un algorithme $(K, \rho)$-robuste, sur un ensemble d'apprentissage $S \cup T$, avec $|T| = n$.

**Théorème III.3.** *Pour un algorithme $\mathcal{A}$, $(K, \rho)$-robuste et la partition liée $Z = X \times Y$, si $\mathcal{D}_T$ est $\lambda$-shift par rapport à $\mathcal{D}_S$, selon la partition de $X$, alors $\forall \delta > 0$, avec une probabilité d'au moins $1 - \delta$, $\forall h \in H$ :*

$$\epsilon_{\mathcal{D}_T} \leq \rho + M \sqrt{\frac{2K \ln 2 + 2 \ln \frac{1}{\delta}}{n}} + \sum_{i=1}^{K_x} T(X_i) l_S^{\lambda}(h, X_i),$$

où $T$ est l'ensemble de taille $n$ d'exemples non étiquetés distribués selon $\mathcal{D}_T$. Un algorithme est également introduit dans ce travail, cherchant à minimiser le cas où le changement d'étiquette entre la source et la cible induit la plus grande perte. En l'occurrence, l'apprentissage du classifieur se fait par la minimisation du dernier terme de la borne précédente, c'est en effet le seul qui dépende de $h$.

D'autres travaux ont traité le problème de l'adaptation de domaine d'un point de vue théorique, notamment dans [Germain et al., 2013], basé sur la théorie PAC-bayésienne. Dans ce travail, une pseudo-distance entre les deux distributions permet aux auteurs de dériver une borne en gé-

néralisation pour le classifieur de Gibbs[3] . Les auteurs présentent ensuite une spécification aux classifieurs linéaires, introduisant notamment un algorithme d'apprentissage obtenant des résultats pratiques intéressants.

Les cadres théoriques présentés dans cette section permettent de définir de manière formelle l'adaptation de domaine et introduisent des bornes sur l'erreur en généralisation. L'idée sous-jacente pour réduire celle-ci est d'obtenir l'erreur empirique source la plus basse possible, tout en diminuant la divergence entre les deux domaines. Se basant sur cette idée, des approches algorithmiques ont vu le jour, proposant divers moyens pour y parvenir. Trois grandes familles peuvent être distinguées. La première regroupe les méthodes dites de repondération, où le but est de trouver une pondération des exemples sources simulant au mieux la distribution cible. La deuxième contient les approches basées sur un changement d'espace de représentation, que ce soit par sélection d'attributs ou création de nouveaux attributs latents. Enfin, la troisième catégorie regroupe les méthodes itératives dites d'auto-étiquetage, où l'algorithme insère à chaque étape des exemples cibles auxquels il a attribué une étiquette.

Rappelons que dans les bornes proposées, un terme dont la valeur ne peut être estimée existe. Il représente en quelque sorte une différence entre les fonctions d'étiquetage de $\mathcal{D}_S$ et $\mathcal{D}_T$. Ceci signifie que si les deux domaines sont trop éloignés, les algorithmes d'AD risquent de réaliser des transferts négatifs. Cette question constitue d'ailleurs toujours un problème ouvert. Parmi les algorithmes qui seront présentés dans la suite de ce chapitre, la majorité d'entre eux considère que l'adaptation est effectivement "possible". Notre contribution du Chapitre IV, pour éviter la dégénérescence du processus de transfert, exploite une mesure de divergence adaptée durant l'apprentissage.

## III.3 Principales familles d'algorithmes

Les contributions algorithmiques en adaptation de domaine peuvent être réparties en trois grandes familles :

- Les **méthodes de repondération**, qui sont particulièrement efficaces dans le cadre spécifique du *covariate shift*[4] . Ce dernier est un cas particulier de l'adaptation de domaine, dans lequel l'hypothèse émise est que la probabilité conditionnelle d'un exemple $x$ d'être étiqueté $y$ est la même pour les deux distributions : $Pr_{\mathcal{D}_S}[y|x] = Pr_{\mathcal{D}_T}[y|x]$, tandis que les distributions marginales sont différentes (à cause d'un biais lors de la collecte des données sources par exemple) : $Pr_{\mathcal{D}_S}[x] \neq Pr_{\mathcal{D}_T}[x]$. Dans ce cas, l'idée est donc de trouver une repondération des exemples sources telle que ceux situés dans des régions de forte densité d'exemples cibles aient un poids important et inversement. Un algorithme d'apprentissage

3. Le classifieur de Gibbs est défini par une distribution $\mathcal{D}$ sur un ensemble d'hypothèses $\mathcal{H}$. Il s'agit d'un vote de majorité pondéré.

4. Notons que certaines personnes considèrent le problème du covariate shift comme indépendant de celui de l'adaptation de domaine.

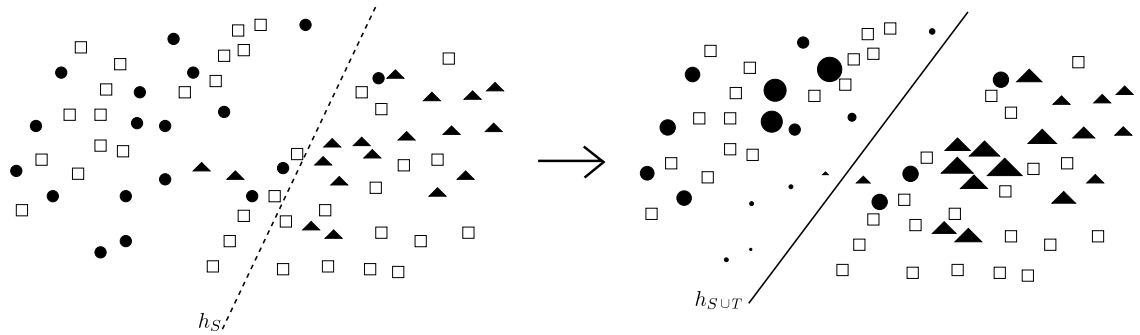



FIGURE III.6 – Illustration simplifiée du principe de repondération pour l'adaptation de domaine. Plus les points sources se situent dans le voisinage proche des points cibles, plus l'importance qu'on leur attribue dans le processus d'apprentissage sera élevée. $h_S$ représente le séparateur appris en se basant sur la distribution source d'origine, tandis que $h_{S\cup T}$ représente le classifieur appris sur la distribution source repondérée.

"classique" peut ensuite être exécuté sur cet ensemble repondéré, supposé comme étant plus proche de la distribution cible que l'ensemble non pondéré.

- Les **méthodes de projection** ont pour objectif de trouver un nouvel espace de représentation pour les exemples afin de décrire au mieux les caractéristiques partagées par les deux domaines $\mathcal{D}_S$ et $\mathcal{D}_T$. Il est possible de distinguer deux stratégies : la première suppose que certains des attributs sont généralisables et que, par conséquent, ils peuvent être partagés par les deux domaines, tandis que d'autres sont spécifiques à un des deux. Dans ce contexte, un algorithme de sélection d'attributs, en pénalisant ou en supprimant certains, peut être utilisé pour trouver un espace de faible dimension des attributs partagés. La seconde stratégie vise à apprendre de nouveaux attributs latents, modélisant les corrélations entre les deux domaines, comme par exemple à l'aide d'une Analyse en Composantes Principales (ACP).

- Enfin, les **approches itératives dites d'auto-étiquetage** visent à étiqueter progressivement les données cibles à l'aide d'une hypothèse apprise sur les données sources, puis à exploiter ces exemples dits *semi-étiquetés* pour apprendre un nouveau classifieur. La difficulté de ce genre d'approches est de définir la façon de choisir les exemples cibles à étiqueter à chaque étape et la proportion adéquate. Plusieurs méthodes ont été proposées, notamment DASVM [Bruzzone and Marconcini, 2010] que nous présenterons en détails dans la Section III.3.3.

### III.3.1 Méthodes de repondération

Dans le cas où les exemples d'apprentissage ne sont pas assez représentatifs, au sens de la distribution (par exemple à cause d'un biais lors de la collecte des données), du domaine cible $\mathcal{D}_T$, on a affaire à un problème connu sous le nom de *covariate shift*. En supposant qu'un certaine partie

des données sources peut être utilisée pour apprendre correctement les exemples cibles, l'adaptation de domaine peut s'effectuer sur la base d'une repondération des exemples d'apprentissage, afin que la version pondérée de $Pr_{\mathcal{D}_S}[x]$ soit la plus proche possible de $Pr_{\mathcal{D}_T}[x]$, comme illustré par la Figure III.6. L'erreur réelle sur le domaine cible peut se réécrire de la façon suivante dans le cas discret :

$$\begin{aligned}
\epsilon^{\ell}_{\mathcal{D}_T}(h) &= \mathbb{E}_{(x,y)\sim\mathcal{D}_T}\ell(h(x),y) \\
&= \mathbb{E}_{(x,y)\sim\mathcal{D}_T}\frac{Pr_{\mathcal{D}_S}[x,y]}{Pr_{\mathcal{D}_S}[x,y]}\ell(h(x),y) \\
&= \sum_{(x,y)\in X\times Y} Pr_{\mathcal{D}_T}[x,y]\frac{Pr_{\mathcal{D}_S}[x,y]}{Pr_{\mathcal{D}_S}[x,y]}\ell(h(x),y) \\
&= \mathbb{E}_{(x,y)\sim\mathcal{D}_S}\frac{Pr_{\mathcal{D}_T}[x,y]}{Pr_{\mathcal{D}_S}[x,y]}\ell(h(x),y)
\end{aligned}$$

Ce qui signifie que le poids à appliquer pour un exemple $(x, y)$ est défini par $\frac{Pr_{\mathcal{D}_T}[x,y]}{Pr_{\mathcal{D}_S}[x,y]}$. Cependant, dans les tâches d'apprentissage classiques, les vraies probabilités marginales sont inconnues. On dispose d'ensembles distribués selon $\mathcal{D}_S$ et $\mathcal{D}_T$, dont les exemples sont parfois de grande dimension, rendant ainsi l'estimation de densité difficile. C'est pourquoi un grand nombre de travaux ont développé des approches pour estimer les poids à appliquer. Dans [Huang et al., 2006], les auteurs proposent une méthode appelée *Kernel Mean Matching* (KMM), pour estimer des poids $\mathbf{w}(x_i)$, pour chaque $x_i \in S$, afin de rendre la distribution pondérée $\mathcal{D}_S$ la plus similaire possible à $\mathcal{D}_T$. La similarité entre les distributions est mesurée comme étant la différence moyenne des exemples (pondérés) projetés dans un espace de Hilbert à noyau reproduisant, connue sous le nom de Maximum Mean Discrepancy (MMD), proposée dans [Gretton et al., 2006]. La fonction objectif est donnée par le terme de discrepancy, entre les deux moyennes empiriques. En prenant $K_{ij} = k(x_i, x_j)$ et $\kappa_i = \frac{m}{m'}\sum_{j=1}^{m'} k(x_i, x'_j)$, $m$ et $m'$ représentant respectivement la dimension d'entrée de la source et de la cible, et si :

$$||\frac{1}{m}\sum_{i=1}^{m} w^i\Phi(x_i) - \frac{1}{m'}\sum_{i=1}^{m'}\Phi(x'_i)||^2 = \frac{1}{m^2}\mathbf{w}^T K\mathbf{w} - \frac{2}{m^2}\kappa^T\mathbf{w} + \text{ constante},$$

alors le problème d'optimisation est le suivant :

$$\min_{\mathbf{w}} \frac{1}{2}\mathbf{w}^T K\mathbf{w} - \kappa^T\mathbf{w},$$

$$\text{avec } w^i \in [0, B] \text{et } \left|\sum_{i=1}^{m} w^i - m\right| \leq m\varepsilon,$$

où $B$ est une constante. Une approche similaire, appelée *Kullback-Leibler Importance Estimation Procedure* (KLIEP) a été introduite dans [Sugiyama et al., 2007, Tsuboi et al., 2008]. Le but est également d'estimer des poids permettant de maximiser la similarité entre la cible et la source pondérée, mais cette fois-ci, en fonction de la divergence de Kullback-Leibler (KL-divergence). Celle-ci est une mesure introduite dans [Kullback and Leibler, 1951], calculée de la manière suivante entre deux distributions $\mathcal{D}$ et $\mathcal{D}'$ :

$$KL(\mathcal{D}||\mathcal{D}') = \sum_i \ln\left(\frac{Pr_{\mathcal{D}}[i]}{Pr_{\mathcal{D}'}[i]}\right) Pr_{\mathcal{D}}(i).$$

Cette divergence, bien que n'étant pas une métrique, est positive et borne la $L_1$. L'approche KLIEP cherche donc des poids $\mathbf{w}(x_i)$, permettant de minimiser $KL(\mathcal{D}_T||\mathbf{w}\mathcal{D}_S)$. L'estimation des poids prend la forme d'une combinaison linéaire de fonctions "de base", comme des gaussiennes centrées sur chaque exemple, afin de minimiser la KL-divergence calculée par rapport aux exemples du domaine cible. Le vecteur de poids $\mathbf{w}$ est en effet modélisé de la manière suivante : $\widehat{\mathbf{w}}(x) = \sum_{l=1}^{b} \alpha_l \phi_l(x)$, où $\{\alpha_l\}_{l=1}^b$ sont les paramètres à apprendre et $\{\phi_l(x)\}_{l=1}^b$ sont des fonctions de base telles que $\phi_l(x) \geq 0$ pour tout $x \in \mathcal{D}_S$ pour $l = 1, 2, \ldots, b$.

D'autres méthodes ont également été développées, par exemple dans [Zadrozny, 2004], où les auteurs proposent un modèle pour le covariate shift dans lequel une variable binaire aléatoire $s \in \{0, 1\}$ détermine si un exemple cible appartient ou non à l'ensemble d'apprentissage source. Les probabilités de sélection $Pr(s = 1|x)$ sont inversement proportionnelles aux poids désirés, de sorte que $\mathbf{w}(x) = \frac{Pr_{\mathcal{D}_T}(x)}{Pr_{\mathcal{D}_T}(x|s=1)} = \frac{Pr(s=1)}{Pr(s=1|x)}$. Dans leur partie expérimentale, les auteurs supposent que la probabilité de sélection est connue, mais suggèrent de l'apprendre automatiquement, évitant ainsi de devoir estimer explicitement les densités (en considérant que les couples $(x, s)$ sont disponibles). Ce serait effectivement le cas si les éléments du domaine source étaient en réalité un sous-ensemble des données du domaine cible. Cependant, ce n'est pas vrai dans le scénario auquel nous faisons face.

Dans [Cortes et al., 2008], une méthode d'estimation de poids est proposée, basée sur un clustering effectué sur toutes les données, puis une estimation de poids spécifique à chacun des clusters, en fonction du nombre d'exemples sources et cibles présents. Un autre travail [Ren et al., 2008] introduit une approche, elle aussi basée sur le clustering, dont le but est de sélectionner les exemples d'apprentissage permettant d'équilibrer la distribution entre les clusters.

Bickel et al., quant à eux, proposent dans [Bickel et al., 2007] d'apprendre un classifieur afin d'estimer les poids. Ils utilisent une variable binaire aléatoire $\sigma$, qui définit si un exemple appartient au domaine source ou cible. Contrairement à la variable de sélection $s$, utilisée dans [Cortes et al., 2008] et [Ren et al., 2008], la valeur de $\sigma_i$ est connue pour les exemples à la fois de $\mathcal{D}_S$ et $\mathcal{D}_T$. La fonction de pondération $w(x)$ peut être écrite comme le ratio $\frac{Pr(\sigma=0|x)}{Pr(\sigma=1|x)}$, qui convient à un modèle de régression logistique $Pr(\sigma|x, \mathbf{\Lambda})$ prédisant le domaine $\mathcal{D}_S$ ou $\mathcal{D}_T$. Par la suite, ils proposent

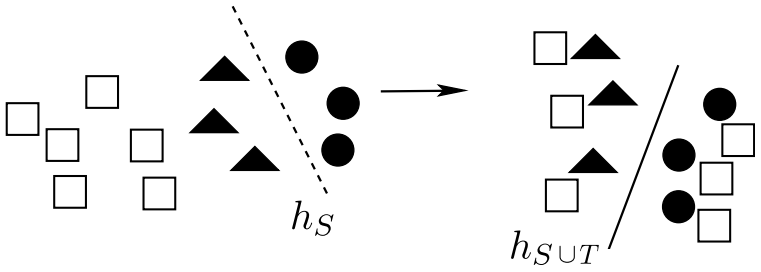



FIGURE III.7 – Illustration simplifiée du principe de changement d'espace de représentation dans le cadre de l'adaptation de domaine. On cherche à trouver un nouvel espace, dans lequel les données sources et cibles seront proches, tout en étant capable de discriminer les deux classes des points sources. $h_S$ représente le séparateur qui serait appris en utilisant uniquement les données sources dans l'espace d'origine. $h_{S\cup T}$ représente le classifieur inféré après reprojection en cherchant à rapprocher les deux domaines.

d'optimiser simultanément les paramètres $\mathbf{\Lambda}$ des poids et ceux $\theta$ du classifieur $Pr(y|x;\theta)$ :

$$\sum_{(x_i,y_i)\in\mathcal{D}_S} w(x;\mathbf{\Lambda})\log(Pr(y_i|x_i;\theta)) + \sum_{x_i\in\mathcal{D}_S,\mathcal{D}_T} \log(Pr(\sigma_i|x_i;\mathbf{\Lambda})) - \alpha||\mathbf{\Lambda}||^2 - \gamma||\theta||^2,$$

où les deux derniers termes correspondent à la régularisation des vecteurs de paramètres. Malgré de bons résultats expérimentaux, il n'apparaît pas évident que cette approche soit meilleure que celles adoptant un comportement séquentiel : d'abord apprendre les poids, puis apprendre un séparateur avec ceux-ci. Cette catégorie de méthodes semble particulièrement adaptée dans le cas d'un mauvais échantillonage, ou lorsque les deux domaines sont relativement proches.

D'autres travaux étudiant ces approches existent, comme par exemple [Dudík et al., 2005], où sont proposées trois méthodes différentes dans le cadre de l'estimation de densité d'entropie maximale et [Jiang and Zhai, 2007a], qui introduit une approche de repondération dans le cadre spécifique du traitement de la langue naturelle.

Enfin, il est intéressant de noter qu'un des travaux théoriques proposant des bornes sur l'erreur en généralisation [Mansour et al., 2009] introduit également une approche de repondération ayant pour but la minimisation de la borne présentée, en diminuant le terme de divergence introduit, entre la distribution cible et la distribution source repondérée.

### III.3.2 Changements d'espace de représentation

Lorsque deux domaines sont similaires, sans être identiques, il est légitime de penser qu'on peut leur trouver des caractéristiques communes. Suivant cette idée, les méthodes basées sur le changement d'espace de représentation cherchent donc à construire un nouvel espace visant à rapprocher

les domaines source et cible. Ce principe est illustré par la Figure III.7. Deux classes d'approches peuvent alors être distinguées :

- La première part de l'hypothèse selon laquelle certains attributs sont spécifiques à un domaine donné, tandis que d'autres sont généralisables. Il s'agit alors de rendre les deux domaines les plus similaires possible en pénalisant ou en supprimant les attributs dont les statistiques varient de façon trop importante entre la source et la cible.
- La seconde vise à apprendre une projection des données sources et cibles, dans un nouvel espace à base d'attributs latents, afin que les exemples soient plus proches dans ce nouvel espace que dans l'espace de départ.

De nombreuses méthodes ont été proposées dans ces deux sous-catégories. Un travail pionnier a été effectué dans [Blitzer et al., 2006], proposant l'algorithme SCL (*Structural Correspondence Learning*), cherchant à définir des attributs discriminants et similaires dans les deux domaines. Les domaines sont alors mis en correspondance par rapport à ces attributs dans un espace de représentation commun dans lequel est apprise l'hypothèse. Une extension de ce travail, gardant la même intuition, a été proposée dans [Blitzer et al., 2011].

Dans [Satpal and Sarawagi, 2007], le but est de sélectionner les attributs permettant de minimiser une distance, de type MMD, entre les moyennes des deux domaines, tout en maximisant la performance en terme de classification sur les données issues du domaine source. Les auteurs utilisent les champs aléatoires conditionnels (CRF) [Lafferty et al., 2001], où l'on cherche à apprendre un vecteur de poids $\mathbf{w}$ sur les "attributs" $f_k(x, y)$ qui sont des fonctions à la fois sur $x$ et $y$. La "distance" mesurée n'est autre que la somme $\sum_{k \in F} d(E^k_{\mathcal{D}_S}, E^k_{\mathcal{D}_T})$ des distances entre les moyennes $E^k_{\mathcal{D}_S}$ et $E^k_{\mathcal{D}_T}$ de chaque attribut $k$ (dans le sous-ensemble $F$ sélectionné) pour les exemples. Cependant, comme les attributs dépendent de $y$, qui est inconnu dans le domaine cible, on utilise la probabilité courante $Pr(y|x, \mathbf{w})$. Il n'est pas commode de chercher tous les sous-ensembles d'attributs, c'est pourquoi les auteurs modifient le problème afin de faire une sélection d'attributs plus souple, en utilisant $\mathbf{w}$ pour pondérer la somme des distances d'attributs. Il en résulte la fonction objectif suivante :

$$\max_{\mathbf{w}} \sum_{i \in \mathcal{D}_S} \sum_k w^k f_k(x_i, y_i) - \log(z_{\mathbf{w}}(x_i)) - \lambda \sum_k |w^k|^\gamma d(E_S\{f_k(x, y)\}, E_T\{f_k(x, y)\}),$$

où le premier terme représente la fonction objectif CRF de base et le second prend en compte la distance entre les deux domaines comme un terme de pénalité. Une idée similaire pour des classifieurs d'entropie maximale est décrite dans [Arnold et al., 2007]. Cependant, plutôt que de pénaliser les attributs avec une large divergence, les auteurs réétalonnent les attributs du domaine source, afin que ceux-ci correspondent aux valeurs attendues dans le domaine cible. Une autre approche, introduite dans [Jiang and Zhai, 2007b], a aussi pour but d'utiliser des attributs généralisables aux

deux domaines. La méthode proposée consiste à apprendre un classifieur utilisant une régression logistique régularisée pour permettre aux attributs généralisables d'être moins régularisés lors de la phase d'apprentissage, en comparaison des attributs spécifiques à un domaine donné.

Un travail récent [Morvant et al., 2011, Morvant et al., 2012] tente de minimiser la $\mathcal{H}$-divergence précédemment présentée, afin de diminuer la borne sur l'erreur en généralisation. Pour ce faire, les auteurs proposent une approche, basée sur les $(\varepsilon, \gamma, \tau)$-bonnes fonctions de similarité [Balcan and Blum, 2006, Balcan et al., 2008], qui vise à projeter les exemples dans un nouvel espace où chaque dimension représente la similarité à un exemple d'apprentissage. Ils le construisent en sélectionnant des couples d'exemples sources et cibles, le but étant que les deux éléments d'un couple soient proches après reprojection. Dans ce nouvel espace, la $\mathcal{H}$-divergence entre les deux domaines est donc plus faible. Enfin, étant donné que la construction des couples d'exemples est un problème difficile, ils proposent une approche itérative, basée sur la sélection d'un nombre de paires limité et un schéma de repondération des similarités conservant les distributions proches l'une de l'autre.

Finalement, d'autres contributions de ce type peuvent être citées, comme [Florian et al., 2004], où est proposé un modèle statistique applicable aux tâches de traitement de la langue naturelle, [Daumé III, 2007], où une approche très simple est introduite, basée sur l'augmentation de la dimension de l'espace de représentation, [Pan et al., 2008], où les auteurs proposent une méthode apprenant un espace de faible dimensionalité, composé d'attributs latents, dans lequel les deux domaines sont proches, ou encore [Ji et al., 2011], introduisant une approche d'analyse en composantes principales multi-vues [5] .

### III.3.3 Approches d'auto-étiquetage

La dernière catégorie de méthodes d'AD est basée sur l'auto-étiquetage. Le principe est assez intuitif : on apprend tout d'abord un classifieur $h$ sur l'ensemble des données sources et on l'utilise pour étiqueter quelques données cibles, dont on se sert ensuite pour apprendre un nouveau modèle. Il s'agit donc d'insérer des exemples cibles étiquetés (on parle de points *semi-étiquetés*) en lieu et place d'exemples sources initiaux, à chaque itération. Dans ce cas, on utilisera comme classifieur final le dernier appris. De manière générale, la procédure peut être décrite ainsi. Soient $S^{(0)} = S$ et $T^{(0)} = T$ :

(1) Apprentissage de $h^{(n)}$ sur $\{(x_i, y_i)\}_{i=1}^{S^{(n)}} \in S^{(n)}$.

(2) $\forall x_j \in T^{(n)}, y_j = h^{(n)}(x_j)$.

(3) Sélection de $S'^{(n)} = \{(x_i, y_i)\}_{i=1}^{n'<|S|}$, exemples sources à retirer de l'ensemble d'apprentissage.

5. Le principe de l'apprentissage multi-vues est de considérer que l'on peut disposer de plusieurs types de données, représentées différemment mais liées entre elles, pour un même problème. Il s'agit donc de combiner les informations recueillies sur chacune des "vues" pour obtenir le classifieur final.

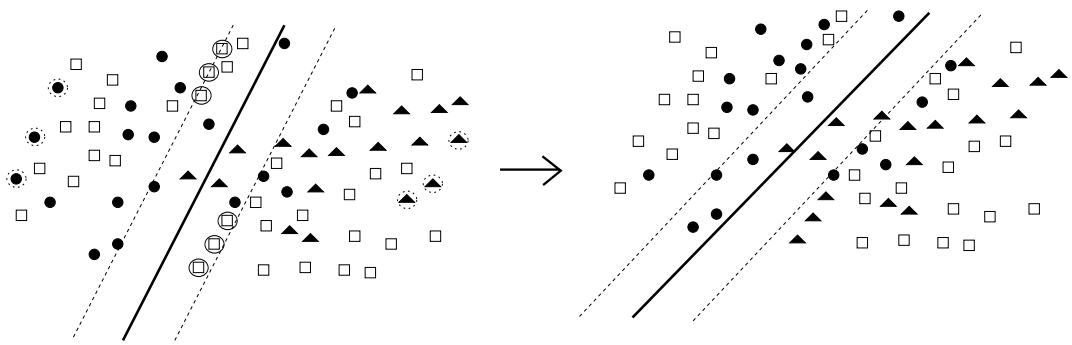


FIGURE III.8 – Illustration du comportement de DASVM. On fixe ici $k = 3$. Trois points sources de chacune des deux classes sont retirés de l'ensemble d'apprentissage, tandis que six points cibles (trois de chaque côté de l'hyperplan) sont étiquetés (devenant ainsi des exemples semi-étiquetés) par le classifieur courant, puis insérés dans l'ensemble d'apprentissage qui servira à apprendre le classifieur suivant.

(4) Sélection de $T'^{(n)} = \{(x_j, y_j)\}_{j=1}^{n'}$, exemples semi-étiquetés à insérer dans l'ensemble d'apprentissage.

(5) $S^{(n+1)} = S^{(n)} \cup T'^{(n)} \setminus S'^{(n)}$

(6) $T^{(n+1)} = T^{(n)} \setminus \{x_j\} \in T'^{(n)}$

(7) Si le critère d'arrêt n'est pas atteint, $n = n + 1$ et on recommence à l'étape (1).

Un des algorithmes les plus connus dans ce domaine, appelé DASVM [Bruzzone and Marconcini, 2010], se base sur la théorie des SVMs et adapte itérativement un séparateur, initialement appris sur les données sources uniquement. À chaque étape, $2k$ exemples sont sélectionnés dans l'ensemble cible, puis insérés, couplés avec les étiquettes attribuées par le séparateur, à la place de $2k$ exemples issus de la distribution source. Plus précisément, dans chaque classe, les $k$ exemples cibles se retrouvant dans une bande de marge (donc dans une zone d'indécision, afin d'entraîner un changement dans le séparateur), mais le plus proche possible de cette marge (dans le but d'avoir tout de même une certaine confiance dans l'étiquette qu'on va leur attribuer), sont sélectionnés. Ces exemples semi-étiquetés remplacent les $k$ points sources de chaque classe, situés le plus loin de l'hyperplan (dans la mesure où ils influent peu sur la construction de ce dernier). Un nouveau SVM est ensuite appris sur l'ensemble d'apprentissage mis à jour et le processus est répété itérativement (la Figure III.8 propose une illustration simplifiée du comportement de l'algorithme entre les itérations $i$ et $i + 1$). Le problème d'optimisation résolu à chaque étape est le suivant :

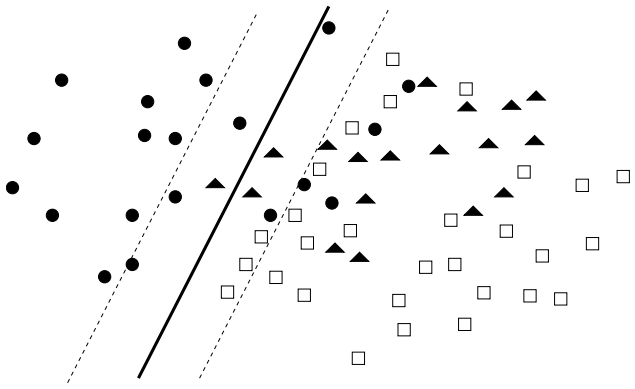


FIGURE III.9 – Illustration d'un cas extrême d'adaptation de domaine. Dans ce cas, tous les exemples cibles se situent du même côté de l'hyperplan. Un algorithme d'auto-étiquetage sera donc dans l'incapacité d'adapter correctement, puisque tous les exemples semi-étiquetés seront de la même classe.

$$\forall l = 1, \ldots, \mu^{(i)}, (\mathbf{x}_S^l, y_S^l) \in S^{(i)}, \forall u = 1, \ldots, \nu^{(i)}, (\mathbf{x}_T^u, \hat{y}_{T^{(i-1)}}^u) \in S^{(i)} :$$

$$\begin{cases} \min\limits_{\mathbf{w},b,\xi_S,\xi_T} \{\frac{1}{2}||\mathbf{w}^{(i)}||^2 + C^{(i)} \sum\limits_l \xi_S^l + \sum\limits_u C_\star^u \xi_T^u\} \\ y_S^l \cdot (\mathbf{w}^{(i)} \cdot x_S^l + b^{(i)}) \geq 1 - \xi_S^l \\ \hat{y}_{T^{(i-1)}}^u \cdot (\mathbf{w}^{(i)} \cdot \mathbf{x}_T^u + b^{(i)}) \geq 1 - \xi_T^u \\ \xi_S^l, \xi_T^u \geq 0, \end{cases}$$

où $h^{(i)}(\mathbf{x}) = \mathbf{w}^{(i)} \cdot \mathbf{x} + b^{(i)}, \mu^{(i)}$ est le nombre d'exemples sources originaux restants, $\nu^{(i)}$ le nombre d'exemples semi-étiquetés déjà insérés dans $S^{(i)}$ (tel que $|S^{(i)}| = \mu^{(i)} + \nu^{(i)}$). Il est important de noter que $C_\star^u$ et $C^{(i)}$ sont des paramètres de régularisation pour les exemples semi-étiquetés et les exemples sources originaux respectivement. Ils visent à contrôler le nombre d'exemples mal classés dans $S^{(i)}$. Augmenter ces valeurs revient à accroître la pénalité associée aux erreurs. De plus, si un exemple semi-étiqueté dans $S_i$ obtient, à l'itération $i$, une étiquette différente de celle qui lui a été précédemment assignée, il est remis dans $T^{(i)}$. Ceci permet d'augmenter le poids des exemples semi-étiquetés qui conservent la même étiquette au fil des itérations. Dans le même temps, le poids des exemples originaux de $S$ diminue, afin de donner plus d'importance aux exemples cibles. Comme expliqué précédemment, on prendra ici en compte le dernier séparateur pour classifier les exemples cibles, celui-ci ayant en effet été appris sur une grande majorité d'exemples issus de la distribution $\mathcal{D}_T$.

Bien qu'étant devenu une référence en matière de méthodes d'auto-étiquetage, DASVM présente plusieurs limitations. La première est due au fait que l'algorithme suppose que les distributions source et cible sont proches. En effet, si ce n'est pas le cas, comme illustré sur la Figure III.9, l'algorithme est incapable d'adapter. Pour contourner ce problème, les auteurs ont proposé une approche visant à estimer la qualité du modèle inféré. Il s'agit de la *validation circulaire*. Celle-ci consiste en

le processus suivant :

(1) Lancer un algorithme d'auto-étiquetage avec deux ensembles $S$ et $T$.

(2) Étiqueter les exemples de $T$ à l'aide du classifieur obtenu.

(3) Lancer un algorithme d'auto-étiquetage, en utilisant comme ensemble source les exemples de $T$, auxquels sont associées les étiquettes obtenues, et comme ensemble cible $S$, les exemples étant privés de leur étiquette.

(4) Mesurer le taux de réussite du classifieur sur les exemples de $S$, à l'aide des étiquettes originales.

Cette approche heuristique permet de valider le modèle inféré par un algorithme d'auto-étiquetage. Intuitivement, si le taux de réussite du classifieur, appris avec les exemples de $T$ auto-étiquetés, sur les exemples de $S$, est important, cela signifie que les étiquettes attribuées aux exemples de $T$ sont majoritairement les bonnes, compte tenu du fait que l'on arrive à effectuer la procédure d'adaptation "dans le sens inverse". A contrario, si nous ne sommes pas capables de bien adapter de $T$ à $S$, on peut légitimement supposer que les étiquettes attribuées aux exemple de $T$, ne permettant pas d'obtenir un classifieur correct sur $S$, sont majoritairement fausses.

Cependant, la validation circulaire est particulièrement coûteuse, dans la mesure où l'algorithme d'auto-étiquetage doit être appliqué deux fois. Nous proposons donc dans le chapitre suivant, pour pallier aux problèmes causés par les situations dans lesquelles les deux domaines sont éloignés, une approche itérative tenant compte d'une mesure de divergence durant l'apprentissage, évitant ainsi un surcoût algorithmique trop important.

Une deuxième limitation de DASVM vient de l'absence de cadre théorique pour de telles méthodes itératives, visant à semi-étiqueter les points cibles. Une des contributions de cette thèse, présentée dans le Chapitre V, vise à combler ce manque en définissant les conditions théoriques pour qu'un tel algorithme adapte bien. Nous proposons également une approche, étendant DASVM au cas des données structurées, tout en relaxant la contrainte, imposée par la théorie des SVMs, d'avoir une fonction de similarité semi-définie positive.

Une autre approche, appelée *validation croisée pour l'apprentissage par transfert* est proposée dans [Zhong et al., 2010]. Celle-ci a pour but de sélectionner l'algorithme, les paramètres et le domaine source (parmi plusieurs disponibles) les plus adaptés pour un domaine cible donné. L'idée est d'utiliser une pondération basée sur le ratio des densités pour réduire la différence des distributions marginales entre les deux domaines, puis d'appliquer une procédure de validation inverse afin d'approximer la différence entre les distributions conditionnelles estimées et réelles sur le domaine cible.

Pour finir, notons que d'autres méthodes itératives existent, n'entrant pas exactement dans le cadre des algorithmes d'auto-étiquetage. Une approche basée sur le co-apprentissage (initialement

introduit dans le cadre de l'apprentissage semi-supervisé par [Blum and Mitchell, 1998]), et inspirée de [Chen et al., 2011b] a été introduite dans [Chen et al., 2011a]. Elle consiste à sélectionner des sous-ensembles d'attributs cibles ainsi que les exemples cibles sur lesquels l'algorithme est le plus confiant, afin de les ajouter itérativement à l'ensemble source. Enfin, [Pérez and Sánchez-Montañés, 2007] propose une procédure itérative basée sur une extension de l'algorithme EM.

## III.4 Conclusion

Nous avons présenté, au cours de cette section, le problème de l'adaptation de domaine. Les théories classiques de l'apprentissage automatique n'étant plus valables dans ce contexte, il a fallu définir de nouveaux cadres théoriques. Certains travaux ont dérivé des bornes sur l'erreur en généralisation sur le domaine cible, pour des classifieurs appris sur le domaine source. Ces résultats reposent généralement sur des mesures de divergence estimant l'éloignement entre la source et la cible. Un grand nombre d'algorithmes d'AD se sont inspirés de ces études théoriques et peuvent être regroupés en trois catégories : les méthodes de repondération, celles de recherche d'un espace de projection commun et les approches itératives d'auto-étiquetage.

Le chapitre suivant présente la première contribution de cette thèse. Celle-ci prend la forme d'un nouvel algorithme de boosting qui projette les exemples sources et cibles dans un nouvel espace, correspondant aux sorties des classifieurs faibles appris itérativement.

# Chapitre IV

## Nouvelle Approche de Boosting pour l'Adaptation de Domaine

**Résumé :** Dans ce chapitre, nous présentons SLDAB, un algorithme d'AD, prenant son origine à la fois dans la théorie du boosting et la théorie de l'adaptation de domaine. Celui-ci vise à traiter les données cibles non étiquetées en minimisant à la fois l'erreur de classification sur les exemples sources et la proportion de violations de marge sur les points cibles. Afin d'éviter la production d'hypothèses dégénérées, nous introduisons une mesure de divergence, dont le but est de pénaliser les hypothèses qui ne permettent pas de réduire l'écart entre les deux domaines. Notre algorithme effectue ensuite une projection des exemples des deux domaines dans l'espace de sortie des hypothèses faibles, dans lequel un hyperplan pour chacune des deux distributions a été inféré. Nous présentons une analyse théorique de notre algorithme et montrons son efficacité en pratique en comparaison avec l'état de l'art.

**Publications dont est issu le travail de ce chapitre :**

Habrard A., Peyrache J-P., Sebban M.
Boosting for Unsupervised Domain Adaptation
ECML/PKDD 2013, Proceedings part II, 433-448, **2013**

Habrard A., Peyrache J-P., Sebban M.
Un Cadre Formel de Boosting pour l'Adaptation de Domaine
14e Conférence francophone sur l'Apprentissage automatique (CAp' 2012), 1-16, **2012**
**Prix du meilleur papier**

## IV.1 Introduction

Cette première contribution repose sur la théorie du boosting, approche ensembliste initialement introduite dans [Schapire, 1989], visant à inférer un modèle fort (au sens PAC), à partir

d'une combinaison linéaire d'hypothèses dites faibles. Le boosting semble effectivement convenir particulièrement bien pour des tâches d'AD pour deux raisons : (i) il est par nature une procédure **adaptative**, dans la mesure où des exemples d'apprentissage sont repondérés afin de créer de la diversité dans les hypothèses induites, (ii) on sait que le boosting maximise la marge [Schapire et al., 1997], caractéristique qui nous apparaît très utile dans le cas où l'ensemble d'apprentissage contient des données non étiquetées, pour lesquelles le taux d'erreur empirique ne peut pas être optimisé.

Notons que le boosting a déjà été utilisé dans des méthodes d'adaptation de domaine, mais principalement dans des situations dans lesquelles l'algorithme d'apprentissage reçoit des données cibles étiquetées. Dans [Dai et al., 2007] par exemple, les auteurs introduisent TrAdaBoost. Cet algorithme utilise la théorie du boosting sur un ensemble d'apprentissage contenant un grand nombre de données sources et peu de données cibles, toutes étiquetées. Ils émettent l'hypothèse que les données cibles sont en quelque sorte une nouvelle version des données obtenues depuis la distribution source et que, du même coup, une partie seulement des exemples sources est assez proche de la cible pour apporter une véritable plus-value lors de l'apprentissage. Ils utilisent donc le schéma de pondération classique d'AdaBoost sur les exemples cibles, tandis que les données sources bien étiquetées par l'hypothèse courante (les auteurs supposant que ces exemples sont assez proches des données cibles) conservent le même poids. A contrario, les exemples sources mal étiquetés voient leur poids diminuer, pour ne pas trop influencer l'hypothèse suivante. Le classifieur final est celui correspondant à la combinaison linéaire des $N/2$ derniers apprenants faibles, les $N/2$ premiers étant encore perturbés par des exemples sources ayant un poids trop important. Les auteurs dérivent la borne en généralisation suivante :

$$\epsilon_{\mathcal{D}_T}(H_T^N) \leq 2^{\lceil \frac{N}{2} \rceil} \prod_{n=\lceil \frac{N}{2} \rceil}^{N} \sqrt{\epsilon_{S^n}(h_n)(1-\epsilon_{S^n}(h_n))},$$

où $\epsilon_{S^n}(h_n) < 0.5$ représente l'erreur de l'apprenant faible $h_n$. L'erreur sur la distribution cible diminue donc avec le nombre $N$ d'itérations. Une généralisation de TrAdaBoost à plusieurs sources est présentée dans [Yao and Doretto, 2010].

Récemment, un autre travail sur le boosting appliqué à l'adaptation de domaine a été proposé dans [Becker et al., 2013]. Les auteurs considèrent plusieurs tâches, représentant autant de domaines différents. Au moins une de ces tâches contient un grand nombre d'exemples, étant ainsi assimilée au domaine source. Tous les exemples sont ici étiquetés. Les auteurs partent de l'hypothèse selon laquelle les différents domaines ont simplement subi des distorsions, et qu'il est donc possible de les projeter, à l'aide d'une transformation non-linéaire, dans un espace commun dans lequel ils sont séparables de façon linéaire.

Le boosting est utilisé ici pour apprendre à la fois la transformation, qui est spécifique à chacun des domaines, et la fonction de séparation dans le nouvel espace de représentation. Il s'agit en conséquence de minimiser la perte exponentielle des données d'apprentissage dans chacune

des tâches. Cette minimisation est complexe, mais le boosting est particulièrement adapté à cette situation. Les résultats expérimentaux semblent confirmer l'intérêt de cette approche, qui est tout de même spécifique à certains problèmes d'analyse d'images présentés dans leur travail.

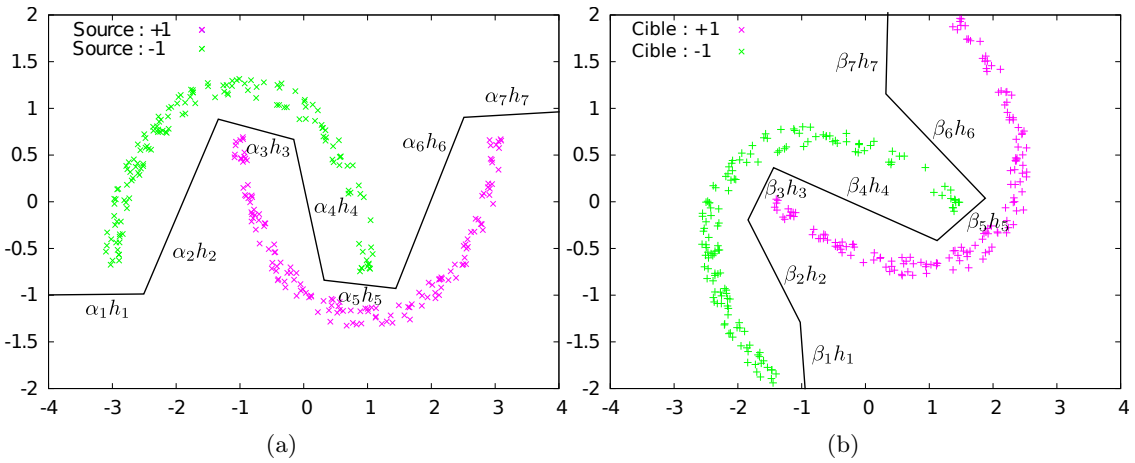
Quelques méthodes basées sur le boosting proposent de s'affranchir de la contrainte d'avoir des exemples cibles étiquetés [d'Alché Buc et al., 2001, Bennett et al., 2002, Mallapragada et al., 2009]. Cependant, elles se situent dans le contexte de l'apprentissage semi-supervisé, c'est-à-dire que l'on suppose que les domaines sources et cibles sont suffisamment similaires pour considérer que directement diminuer l'erreur sur le domaine source diminuera également celle sur la cible.

Nous proposons ici une approche visant à traiter à la fois des problématiques d'apprentissage semi-supervisé et d'adaptation de domaine non supervisée. L'algorithme que nous présentons dans ce chapitre a été conçu pour optimiser simultanément le **taux d'erreur en classification** sur les exemples sources étiquetés (approche classique en boosting), ainsi que la **proportion de violations de marge** (par rapport à une marge $\gamma$) sur les points cibles non étiquetés. Cette approche se base sur l'hypothèse communément acceptée [Bruzzone and Marconcini, 2010] selon laquelle plus la distance d'un exemple non étiqueté au classifieur est grande, plus la probabilité qu'on lui attribue la bonne étiquette est importante. Cependant, ce principe n'est valide que sous deux conditions :

- le classifieur est efficace sur les données sources, afin d'avoir confiance en les marges sur les données cibles,
- la divergence entre les deux distributions doit être réduite, afin d'augmenter la pertinence du point précédent.

Par conséquent, l'objectif de notre algorithme est de (i) réduire itérativement la proportion de points cibles situés à une distance plus petite que $\gamma$ du classifieur, tout en minimisant le taux d'erreur en classification sur les exemples sources et (ii) intégrer la divergence courante dans le processus de repondération des données pour éviter des processus de transfert négatif.

Dans la suite de ce chapitre, nous présentons l'algorithme SLDAB (*Self-Labeling Domain Adaptation Boosting*). Celui-ci se trouve au carrefour des méthodes d'AD itératives et des approches visant à trouver un nouvel espace de projection. Une partie importante est consacrée à l'analyse théorique du comportement de SLDAB. Nous prouvons notamment que l'erreur sur la source et le risque d'avoir de faibles marges sur la cible, tendent vers 0 avec les itérations. Durant cette analyse, nous considérons une mesure de divergence quelconque $g_n \in [0, 1]$. Nous consacrons ensuite une section à la présentation des caractéristiques d'une bonne divergence et introduisons une nouvelle mesure, inspirée de la variation perturbée introduite dans [Harel and Mannor, 2012]. La particularité de celle-ci est qu'elle dépend du classifieur faible construit, lui permettant ainsi d'être prise en compte dans la phase de mise à jour des poids. Le chapitre se termine par une présentation des résultats expérimentaux, en comparaison avec plusieurs approches de l'état de l'art et par une discussion autour des garanties en généralisation, avant de conclure.




(a) (b)

FIGURE IV.1 – Illustration de l'intuition de SLDAB. Les exemples sources et cibles se situent dans le même espace de projection et les mêmes hypothèses faibles sont combinées pour obtenir le classifieur final. La seule différence réside dans les poids qui sont appliqués à ces hypothèses faibles. Dans le cas de la source (figure (a)), il s'agit des poids $\alpha$, tandis que dans le cas du domaine cible (figure (b)), les poids appliqués sont les $\beta$.

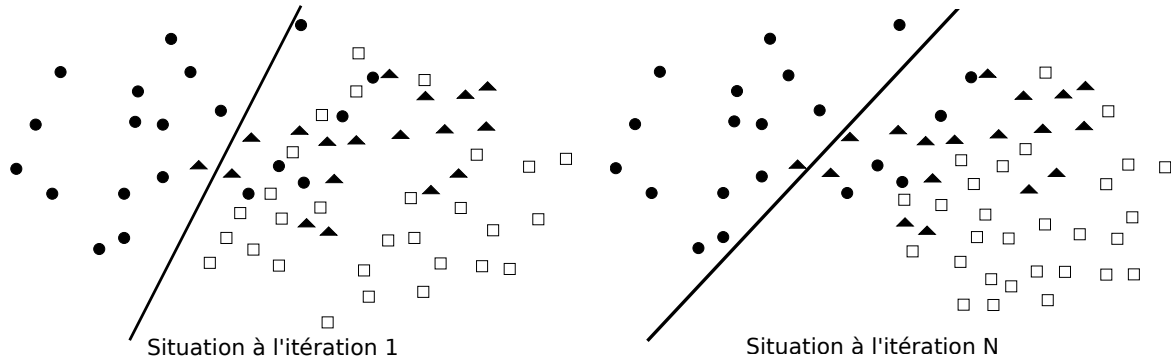
## IV.2 Intuition de SLDAB

Rappelons que le boosting a pour objectif d'apprendre itérativement des classifieurs faibles $h_n$ (typiquement des stumps) et de les combiner linéairement selon leur pertinence. Dans le cas de la classification supervisée classique, cette pertinence dépend de la capacité de $h_n$ à correctement étiqueter les exemples d'apprentissage de $S$ selon la distribution courante $\mathcal{D}_n$. Pour rappel, le classifieur final $F_S^N$, après $N$ itérations, est défini comme suit :

$$F_S^N = \sum_{n=1}^{N} \alpha_n h_n(\mathbf{x}), \text{ où } \alpha_n = \frac{1}{2} \ln \frac{1 - \epsilon_{S^n}(h_n)}{\epsilon_{S^n}(h_n)}.$$

Géométriquement, $F_S^N$ correspond donc à un hyperplan optimisé dans l'espace des sorties des $h_n$.

Dans le cas de l'adaptation de domaine, la situation est différente. L'échantillon d'apprentissage est non seulement composé d'un sous-ensemble $S$, contenant des exemples sources étiquetés, mais aussi d'un sous-ensemble $T$, comprenant des exemples cibles non étiquetés. Dans ce chapitre, nous proposons un algorithme qui, **à l'aide des mêmes hypothèses faibles**, minimise l'erreur empirique sur $S$ tout en maximisant les marges sur $T$. L'idée intuitive derrière cette stratégie est de permettre une projection des données sources et cibles dans le même espace de représentation à $N$ dimensions ($N$ étant le nombre d'itérations de l'algorithme) en réduisant la divergence entre les deux domaines.




(a) (b)

FIGURE IV.2 – Illustration du problème causé par les hypothèses dégénérées. Au départ de l'algorithme, on se trouve dans la situation illustrée par la figure (a). Si un mesure de divergence n'est pas prise en compte durant le processus, on risque de se retrouver dans la situation de la figure (b), dans laquelle les conditions sont respectées (faible erreur de classification sur la source, marges importantes sur la cible), mais dans laquelle tous les exemples cibles obtiennent la même étiquette.

Si les hypothèses apprises $h_1, \ldots, h_n, \ldots, h_N$ sont bien les mêmes pour $S$ et $T$, le coefficient de pondération de chacune d'entre elles devra être différent selon le domaine d'appartenance de l'exemple. Si pour les données de $S$, et conformément à la théorie de l'AD [Ben-David et al., 2010], le critère de qualité de $h_n$ doit rester basé sur l'erreur empirique $\epsilon_{S_n}(h_n)$, il en va différemment pour les données cibles de $T$. La qualité de $h_n$ sur $T$ doit dépendre de sa capacité à minimiser le nombre de violations de marge des exemples cibles. De ce fait, nous avons pour objectif d'optimiser une deuxième combinaison linéaire $F_T^N = \sum_{n=1}^{N} \beta_n h_n(\mathbf{x})$, où $\beta_n$ dépend non seulement de la marge de l'exemple, mais également de la divergence entre $S$ et $T$ induite par $h_n$.

Pour résumer, l'algorithme SLDAB vise à construire 2 hyperplans séparateurs dans l'espace à $N$ dimensions reposant sur l'apprentissage conjoint entre $S$ et $T$ de $N$ hypothèses faibles communes aux exemples sources et cibles. L'orientation géométrique de $F_S^N$ et $F_T^N$ dépendra par contre de leur capacité à minimiser respectivement le risque empirique sur $S$ et le nombre de violations de marge sur $T$. La Figure IV.1 illustre graphiquement le principe de notre méthode.

Comme indiqué précédemment, il est important que les coefficients $\beta_n$ tiennent compte de la capacité de $h_n$ à réduire le nombre de violations de marge, mais aussi à réduire la divergence entre $S$ et $T$. En effet, baser $\beta_n$ uniquement sur les marges ne permettrait pas d'assurer une non-dégénérescence du modèle. La Figure IV.2 illustre cette situation où, au fur et à mesure des itérations, l'algorithme tend à éloigner les données sources et cibles tout en minimisant les erreurs sur la source et les violations de marge sur la cible. De ce fait, nous avons besoin d'une mesure de divergence qui dépende spécifiquement de l'hypothèse $h_n$ construite à l'instant $n$. Dans la section suivante, nous considérons une définition assez générique de cette divergence. Nous y porterons une

attention plus particulière dans la Section IV.6.

## IV.3 Définitions et notations

Soient $\mathcal{H}$ une classe d'hypothèses, $h_n \in \mathcal{H} : X \to [-1, +1]$ une hypothèse apprise depuis $\mathcal{D}_S$ et $\mathcal{D}_T$, et les distributions considérées dans l'algorithme à l'étape $n$, $\mathcal{D}_S^n$ et $\mathcal{D}_T^n$. On note $g_n \in [0, 1]$ une mesure de divergence induite par $h_n$ entre $\mathcal{D}_S$ et $\mathcal{D}_T$. Notre objectif est de tenir compte de $g_n$ dans notre schéma de boosting afin de pénaliser les hypothèses qui ne permettent pas la réduction de la divergence entre les deux distributions. Pour ce faire, nous considérons la fonction $f_{DA} : [-1, +1] \to [-1, +1]$ telle que $f_{DA}(h_n(\mathbf{x})) = |h_n(\mathbf{x})| - \lambda g_n$, où $\lambda \in [0, 1]$. $f_{DA}(h_n(\mathbf{x}))$ exprime la capacité de $h_n$ non seulement à obtenir de grandes marges (c'est-à-dire une grande valeur de $|h_n(\mathbf{x})|$ [1] , mais également à réduire la divergence entre $\mathcal{D}_S$ et $\mathcal{D}_T$ (correspondant à une faible valeur de $g_n$). $\lambda$ joue ici le rôle d'un paramètre de compromis permettant d'ajuster l'importance de la marge et de la divergence.

Soit $T_n^- = \{\mathbf{x} \in T | f_{DA}(h_n(\mathbf{x})) \leq \gamma\}$. Si $\mathbf{x} \in T_n^- \Leftrightarrow |h_n(\mathbf{x})| \leq \gamma + \lambda g_n$. Par conséquent, $T_n^-$ correspond à l'ensemble des points cibles qui soit violent la condition de marge (en effet, si $|h_n(\mathbf{x})| \leq \gamma \Rightarrow |h_n(\mathbf{x})| \leq \gamma + \lambda g_n$), soit ne satisfont pas suffisamment cette marge pour compenser une divergence importante entre les deux distributions (c'est-à-dire que $|h_n(\mathbf{x})| > \gamma$, mais $|h_n(\mathbf{x})| \leq \gamma + \lambda g_n$). De la même manière, on définit $T_n^+ = \{\mathbf{x} \in T | f_{DA}(h_n(\mathbf{x})) > \gamma\}$, de telle sorte que $T = T_n^- \cup T_n^+$. Finalement, nous définissons $W_n^+ = \sum_{\mathbf{x} \in T_n^+} \mathcal{D}_T^n$ et $W_n^- = \sum_{\mathbf{x} \in T_n^-} \mathcal{D}_T^n$, tels que $W_n^+ + W_n^- = 1$. $W_n^+$ (respectivement $W_n^-$) correspond donc à la somme des poids des points satisfaisant (respectivement ne satisfaisant pas) la contrainte $|h_n(\mathbf{x})| > \gamma + \lambda g_n$.

Nous avons vu dans le Chapitre II qu'une hypothèse $h_n$ est considérée comme un apprenant faible sur $\mathcal{D}_S$, par rapport à la distribution courante $\mathcal{D}_S^n$, si sa performance est au moins un peu meilleure que l'aléatoire, c'est-à-dire que $\epsilon_{S^n}(h_n) < \frac{1}{2}$. Nous étendons ici cette définition à celle d'un apprenant faible pour l'adaptation de domaine.

**Définition IV.1** (Apprenant faible pour l'AD)**.** *Une hypothèse $h_n$, apprise à une itération $n$ depuis un ensemble source étiqueté $S$ distribué selon $\mathcal{D}_S$ et un ensemble cible non étiqueté $T$ distribué selon $\mathcal{D}_T$, et induisant une divergence $g_n$ entre $S$ et $T \in [0, 1]$, est un apprenant faible pour l'AD sur $T$ si* $\forall \gamma \leq 1$, $\forall \lambda \in [0, 1]$ *:*

*(1) $h_n$ est un apprenant faible pour $S$, c'est-à-dire $\epsilon_{S^n}(h_n) < \frac{1}{2}$.*

*(2) $\hat{L}_n = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_T^n}[|f_{DA}(h_n(\mathbf{x}))| \leq \gamma] < \frac{\gamma}{\gamma + \max(\gamma, \lambda g_n)}$,*

*où $\hat{L}_n$ représente la proportion empirique d'exemples cibles ne satisfaisant pas la condition de marge.*

1. Normalement, la marge est définie par $yh(\mathbf{x})$ pour tout exemple$(\mathbf{x}, y)$. Cependant, les exemples de $T$ étant non étiquetés, la marge revient ici à calculer $|h_n(\mathbf{x})|$.

La Condition 1 signifie que pour être capable d'adapter depuis $\mathcal{D}_S$ à $\mathcal{D}_T$ en utilisant un schéma de boosting, $h_n$ doit obligatoirement apprendre quelque chose de nouveau à chaque itération par rapport à la fonction d'étiquetage source. Ceci permet de satisfaire la condition théorique (voir Théorème III.1), selon laquelle bien adapter requiert de bien apprendre la source. La Condition 2, qui sera utile lors de l'analyse théorique, tient compte non seulement de la capacité de $h_n$ à satisfaire la marge $\gamma$, mais également de son aptitude à réduire la divergence entre les deux domaines. Ceci permet de satisfaire la deuxième condition théorique, tout en évitant l'inférence de modèles dégénérés. Depuis la Définition IV.1, il apparaît que :

- Si $\max(\gamma, \lambda g_n) = \gamma$, alors $\frac{\gamma}{\gamma+\max(\gamma,\lambda g_n)} = \frac{1}{2}$. La Condition 2 est alors la même que celle de l'apprenant faible sur la source, excepté qu'elle n'opère pas sur la même fonction de perte. $\hat{L} < \frac{1}{2}$ exprime une condition de marge, alors que $\epsilon_{S^n} < \frac{1}{2}$ représente une contrainte de classification. Il est intéressant de noter que si nous sommes dans ce cas pour chacune des hypothèses $h_n$, ceci signifie que la divergence entre la source et la cible est relativement faible ($\max(\gamma, \lambda g_n) = \gamma, \forall n$) et que la tâche que l'on tente de résoudre s'apparente plus à un problème de classification semi-supervisée [2] .
- Si $\max(\gamma, \lambda g_n) = \lambda g_n$, alors la contrainte imposée par la Condition 2 est renforcée par rapport au boosting classique (en effet, $\hat{L}_n < \frac{\gamma}{\gamma+\max(\gamma,\lambda g_n)} < \frac{1}{2}$) afin de compenser une grande divergence entre $\mathcal{D}_S$ et $\mathcal{D}_T$. Dans ce cas, la tâche requiert un processus spécifique d'adaptation de domaine dans le schéma de pondération.

## IV.4 Algorithme SLDAB

Le pseudo-code de SLDAB est présenté dans l'Algorithme 2. A l'instar d'AdaBoost, SLDAB démarre depuis une distribution uniforme sur $S$ et $T$. Il apprend ensuite itérativement une nouvelle hypothèse $h_n$, respectant les contraintes de l'apprenant faible pour l'AD présentées dans la Définition IV.1. Cette tâche n'est pas triviale. En effet, tandis qu'apprendre un stump est suffisant pour satisfaire la contrainte d'apprenant faible d'AdaBoost, trouver une hypothèse remplissant la Condition 1 sur la source et la Condition 2 sur la cible dans le même temps est plus compliqué. Pour contourner ce problème, nous proposons une stratégie simple qui tend effectivement à générer des hypothèses satisfaisant les deux conditions.

Dans un premier temps, nous tirons aléatoirement, dans le but de générer de la diversité (ce qui est, nous le rappelons, une condition essentielle au bon comportement d'une méthode ensembliste), $\frac{k}{2}$ stumps qui satisfont chacun la condition 1 sur la source et $\frac{k}{2}$ stumps satisfaisant la condition 2 sur la cible. Par la suite, nous cherchons une combinaison convexe $h_n = \sum_k \kappa_k h_n^k$ des $k$

2. Rappelons que l'apprentissage semi-supervisé, à l'inverse de l'adaptation de domaine, émet l'hypothèse selon laquelle les données non étiquetées disponibles suivent la même distribution que les exemples étiquetés.

**Entrée :**

- un ensemble $S$ de données étiquetées de taille $l$,
- un ensemble $T$ de données non étiquetées de taille $m$,
- une fonction de divergence $g_n$,
- un nombre d'itérations $N$,
- une marge $\gamma \in [0, 1]$,
- un paramètre de compromis $\lambda \in [0, 1]$.

**Sortie :**

- un classifieur source $H_S^N$,
- un classifieur cible $H_T^N$.

Initialisation : $\forall(\mathbf{x}', y') \in S, D_S^1(\mathbf{x}') = \frac{1}{l}$,
$\forall \mathbf{x} \in T, D_T^1(\mathbf{x}) = \frac{1}{m}$.
**pour** $n = 1$ **à** $N$ **faire**
Apprendre une hypothèse $h_n$ respectant la Définition IV.1.
Calculer la valeur de la divergence $g_n$ (voir la Section IV.6 pour des détails).
$\alpha_n = \frac{1}{2} \ln \frac{1-\epsilon_{S^n}(h_n)}{\epsilon_{S^n}(h_n)}$,
$\beta_n = \frac{1}{\gamma + \max(\gamma, \lambda g_n)} \ln \frac{\gamma W_n^+}{\max(\gamma, \lambda g_n) W_n^-}$.
$\forall(\mathbf{x}', y') \in S,\ D_S^{n+1}(\mathbf{x}') = D_S^n(\mathbf{x}').\frac{e^{-\alpha_n signe(h_n(\mathbf{x}')).y'}}{Z'_n}$.
$\forall \mathbf{x} \in T,\ D_T^{n+1}(\mathbf{x}) = D_T^n(\mathbf{x}).\frac{e^{-\beta_n f_{DA}(h_n(\mathbf{x})).y^n}}{Z_n}$,
où $y^n = signe(f_{DA}(h_n(\mathbf{x})))$ si $|f_{DA}(h_n(\mathbf{x}))| > \gamma$,
$y^n = -signe(f_{DA}(h_n(\mathbf{x})))$ sinon,
et $Z'_n$ et $Z_n$ sont des coefficients de normalisation.
**fin**

$$\forall(\mathbf{x}', y') \in S, F_S^N(\mathbf{x}') = \sum_{n=1}^{N} \alpha_n signe(h_n(\mathbf{x}')),$$

$$\forall \mathbf{x} \in T, F_T^N(\mathbf{x}) = \sum_{n=1}^{N} \beta_n signe(h_n(\mathbf{x})).$$

Classifieur source final : $H_S^N(\mathbf{x}') = signe(F_S^N(\mathbf{x}'))$
Classifieur cible final : $H_T^N(\mathbf{x}) = signe(F_T^N(\mathbf{x}))$.

**Algorithme 2** : SLDAB

stumps $h_n^k$ qui respecte dans le même temps les deux conditions. Pour ce faire, nous proposons la résolution du problème d'optimisation convexe suivant :

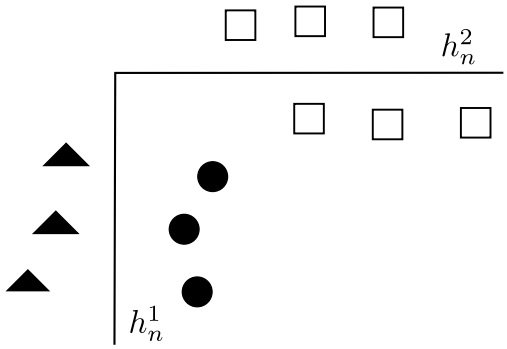



FIGURE IV.3 – Illustration de la combinaison de séparateurs. Un seul séparateur ne suffirait pas pour satisfaire les deux conditions de la Définition IV.1. C'est pourquoi nous proposons d'en combiner plusieurs, comme dans cette figure, où la combinaison de $h_n^1$ et $h_n^2$ permet d'obtenir un apprenant faible pour l'adaptation de domaine, classant correctement la majorité des exemples sources et obtenant dans le même temps des marges satisfaisantes, par rapport à $\gamma$ et $\lambda g_n$ sur le domaine cible.

$$\underset{\kappa}{\operatorname{argmin}} \sum_{(\mathbf{x}',y')\in S} D_n^S(\mathbf{x}') \left[ -y' \sum_k \kappa_k signe(h_n^k(\mathbf{x}')) \right]_+ + \sum_{\mathbf{x}\in T} D_n^T(\mathbf{x}) \left[ 1 - \left( \sum_k \kappa_k marg(f_{DA}(h_n^k(\mathbf{x}))) \right) \right]_+ \tag{IV.1}$$

où $[1-x]_+ = \max(0, 1-x)$ est la perte hinge et $marg(f_{DA}(h_n^k(\mathbf{x})))$ renvoie $-1$ si $f_{DA}(h_n^k(\mathbf{x}))$ est inférieur à $\gamma$ (c'est-à-dire si $h_n$ n'obtient pas une marge suffisante par rapport à $g_n$) et $+1$ dans le cas contraire. Résoudre ce problème d'optimisation tend à satisfaire les contraintes de la Définition IV.1. En effet, minimiser le premier terme de l'Équation IV.1 tend à réduire le risque sur les données sources, tandis que minimiser le second terme tend à diminuer le nombre de violations de marge sur les données cibles. Utiliser la combinaison de plusieurs séparateurs permet donc de satisfaire les deux conditions, comme illustré par la Figure IV.3

Afin de générer l'apprenant faible pour l'AD le plus simple possible, nous suggérons de fixer $k = 2$. Si les conditions ne sont pas satisfaites sur plusieurs tirages de $k = 2$, $k$ est alors incrémenté. Si malgré l'augmentation de $k$ (limité à un certain seuil fixé par l'utilisateur), aucune hypothèse ne satisfait les conditions de l'apprenant faible pour l'AD, l'adaptation est probablement difficile à réaliser et la procédure itérative s'interrompt. Celle-ci est décrite dans l'Algorithme 3.

Une fois que $h_n$ a été appris, les poids des exemples, étiquetés comme non étiquetés, sont modifiés en fonction de deux règles de mise à jour différentes. Ceux des exemples sources sont mis à jour selon la même stratégie que dans ADABOOST. En ce qui concerne les exemples cibles, leur poids est modifié en fonction de leur localisation dans l'espace. Si un exemple cible $\mathbf{x}$ ne satisfait pas la condition $f_{DA}(h_n(\mathbf{x})) > \gamma$, une pseudo-étiquette $y^n = -signe(f_{DA}(h_n(\mathbf{x})))$ lui est attribuée, simulant ainsi une mauvaise classification. Notons qu'une telle décision a une interprétation géo-

**Entrée :**

- un ensemble $S$ de données étiquetées de taille $l$,
- un ensemble $T$ de données non étiquetées de taille $m$,
- un nombre $k$ de stumps à combiner,
- une constante $K_{\text{MAX}}$,
- une constante $I_{\text{MAX}}$.

**Sortie :**

- une combinaison de stumps.

$k = 2$
**tant que** $k < K_{\mathbf{MAX}}$ **faire**
  **pour** $i = 0$ **à** $I_{\mathbf{MAX}}$ **faire**
    **pour** $j = 0$ **à** $k$ **faire**
      **si** $j$ **est impair alors**
        Tirer un stump respectant la Condition 1 de la Définition IV.1.
      **fin**
      **sinon**
        Tirer un stump respectant la Condition 2 de la Définition IV.1.
      **fin**
    **fin**
    Résoudre le Problème IV.1.
    **si Le classifieur appris respecte les conditions de la Définition IV.1 alors**
      Renvoyer le classifieur.
    **fin**
  **fin**
**fin**
Interrompre la procédure itérative.

**Algorithme 3** : Construction du classifieur faible de SLDAB

métrique : cela signifie qu'on augmente exponentiellement le poids des points qui sont situés dans une bande de marge de largeur $\gamma + \lambda g_n$. Si $\mathbf{x}$ est à l'extérieur de cette bande, une pseudo-étiquette $y^n = signe(f_{DA}(h_n(\mathbf{x})))$ lui est attribuée, entraînant ainsi une diminution exponentielle de $D_T^n(\mathbf{x})$ à l'itération suivante.

## IV.5 Analyse théorique

Dans cette section, nous présentons une analyse théorique de notre algorithme SLDAB. Rappelons que la qualité d'une hypothèse $h_n$ est mesurée non seulement par sa capacité à classifier correctement les exemples sources, mais également à pseudo-étiqueter les exemples cibles avec une

marge importante eu égard à la divergence $g_n$ induite par l'hypothèse elle-même. Etant donné que les contraintes d'apprenant faible pour l'AD de la Définition IV.1 sont respectées, les résultats standards d'AdaBoost sont toujours valables sur $\mathcal{D}_S$, à savoir la diminution de l'erreur empirique sur $S$, ainsi que la diminution de l'erreur en généralisation sur $\mathcal{D}_S$ avec les itérations (voir Théorèmes II.7 et II.8). Dans ce qui suit, nous montrons que la perte empirique $\hat{L}_{H_T^N}$, qui représente la proportion de violations de marge sur $T$ après $N$ itérations, décroît avec $N$.

### IV.5.1 Borne sur la perte empirique

**Théorème IV.1.** *Soit $\hat{L}_{H_T^N}$ la proportion des exemples cibles de $T$ avec une marge plus petite que $\gamma$ par rapport aux divergences $g_n (n = 1, \ldots, N)$ après $N$ itérations de SLDAB, on a :*

$$\hat{L}_{H_T^N} = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_T}[\mathbf{y}\mathbf{F}_\mathbf{T}^\mathbf{N}(\mathbf{x}) < 0] \leq \frac{1}{|T|}\sum_{\mathbf{x}\sim\mathcal{D}_T} e^{-\mathbf{y}\mathbf{F}_\mathbf{T}^\mathbf{N}(\mathbf{x})} = \prod_{n=1}^{N} Z_n,$$

*où $\mathbf{y} = (y^1, \ldots, y^n, \ldots, y^N)$ est le vecteur des pseudo-étiquettes et*
$\mathbf{F}_\mathbf{T}^\mathbf{N}(\mathbf{x}) = (\beta_1 f_{DA}(h_1(\mathbf{x})), \ldots, \beta_n f_{DA}(h_n(\mathbf{x})), \ldots, \beta_N f_{DA}(h_N(\mathbf{x})))$.

*Démonstration.* Soit $\hat{L}_{H_T^N} = \hat{Pr}_{\mathbf{x}\sim T}[\mathbf{y}\mathbf{F}_\mathbf{T}^\mathbf{N}(\mathbf{x}) < 0]$ où $[\cdot]$ est la fonction indicatrice, nous avons :

$$\hat{L}_{H_T^N} \quad = \quad \frac{1}{|T|}\sum_{\mathbf{x}\sim T}[-\mathbf{y}\mathbf{F}_\mathbf{T}^\mathbf{N}(\mathbf{x}) \geq 0] \leq \frac{1}{|T|}\sum_{\mathbf{x}\sim T} e^{-\mathbf{y}\mathbf{F}_\mathbf{T}^\mathbf{N}(\mathbf{x})}. \tag{IV.2}$$

Donc, la première inégalité du Théorème IV.1 est vérifiée. De plus, $\forall \mathbf{x} \in T$,

$$D_{N+1}(\mathbf{x}) = \frac{D_N(\mathbf{x}) e^{-\beta_N f_{DA}(h_N(\mathbf{x}))\dot{y}^N}}{Z_n} = \frac{D_1(\mathbf{x})\prod_n e^{-\beta_n f_{DA}(h_n(\mathbf{x}))y^n}}{\prod_n Z_n}.$$

Considérant la distribution uniforme à l'itération $n = 1$, on obtient

$$\sum_{\mathbf{x}\sim T} D_{N+1}(\mathbf{x})\prod_n Z_n = \frac{1}{|T|}\sum_{\mathbf{x}\sim T} e^{-\sum_n \beta_n f_{DA}(h_n(\mathbf{x}))y^n} = \frac{1}{|T|}\sum_{\mathbf{x}\sim T} e^{-\mathbf{y}\mathbf{F}_\mathbf{N}^\mathbf{T}(\mathbf{x})}. \tag{IV.3}$$

Nous déduisons des Équations IV.2 et IV.3 le dernier résultat du théorème :

$$\hat{L}_{H_T^N} \leq \frac{1}{|T|}\sum_{\mathbf{x}\sim T} e^{-\mathbf{y}\mathbf{F}_\mathbf{T}^\mathbf{N}(\mathbf{x})} = \frac{1}{|T|}\sum_{\mathbf{x}\sim T} D_{N+1}(\mathbf{x})\prod_n Z_n \leq \frac{1}{|T|}\prod_n Z_n.$$

□

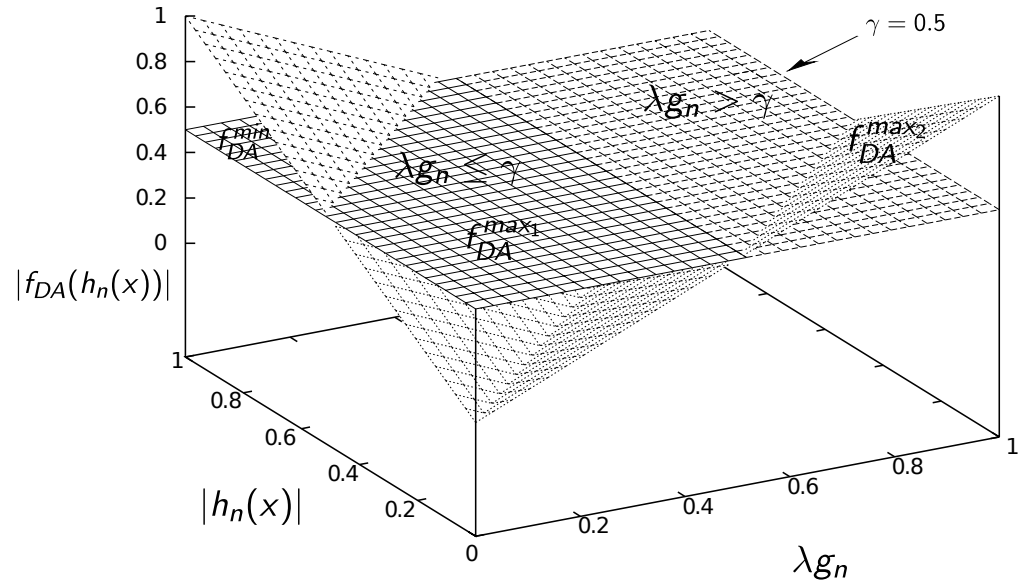



FIGURE IV.4 – Bornes supérieures des différents éléments de $Z_n$ pour une valeur arbitraire $\gamma = 0.5$. Si $\mathbf{x} \in T_n^+$, la borne supérieure correspond à $|f_{DA}| = \gamma$ (voir le plateau $f_{DA}^{min}$). Si $\mathbf{x} \in T_n^-$, on obtient la borne supérieure $\max(\gamma, \lambda g_n)$, qui est soit $\gamma$ quand $\lambda g_n \leq \gamma$ (voir $f_{DA}^{max_1}$) soit $\lambda g_n$ dans le cas inverse (voir $f_{DA}^{max_2}$).

Notons que cette preuve est très similaire à celle de [Freund and Schapire, 1996], à la différence près que $\mathbf{y}$ représente le vecteur des pseudo-étiquettes (il dépend donc de $\lambda g_n$ et $\gamma$) et non le vecteur des véritables étiquettes.

### IV.5.2 Coefficients de confiance optimaux

Le Théorème IV.1 suggère la minimisation de chacun des $Z_n$, afin de réduire la perte empirique $\hat{L}_{H_T^N}$ sur $T$. Dans ce but, nous réécrivons $Z_n$ comme suit :

$$Z_n = \sum_{\mathbf{x} \in T_n^-} D_T^n(\mathbf{x}) e^{-\beta_n f_{DA}(h_n(\mathbf{x}))y^n} + \sum_{\mathbf{x} \in T_n^+} D_T^n(\mathbf{x}) e^{-\beta_n f_{DA}(h_n(\mathbf{x}))y^n}. \quad \text{(IV.4)}$$

Les deux termes de la partie droite de l'Équation IV.4 peuvent être bornés de la manière suivante :

- $\forall \mathbf{x} \in T_n^+$, $D_T^n(\mathbf{x}) e^{-\beta_n f_{DA}(h_n(\mathbf{x}))y^n} \leq D_T^n(\mathbf{x}) e^{-\beta_n \gamma}$.
- $\forall \mathbf{x} \in T_n^-$, $D_T^n(\mathbf{x}) e^{-\beta_n f_{DA}(h_n(\mathbf{x}))y^n} \leq D_T^n(\mathbf{x}) e^{\beta_n max(\gamma, \lambda g_n)}$.

La Figure IV.4 illustre géométriquement ces bornes supérieures. Pour les $\mathbf{x} \in T_n^+$, les poids sont diminués. On obtient donc une borne supérieure en prenant la plus petite baisse possible,

à savoir $f_{DA}(h_n(\mathbf{x}))y^n = |f_{DA}| = \gamma$ (soit $f_{DA}^{min}$ dans la Figure IV.4). D'autre part, si $\mathbf{x} \in T_n^-$, on obtient une borne supérieure en prenant la valeur maximale de $f_{DA}$ (c'est-à-dire la plus grande augmentation possible, puisque dans ce cas on augmente le poids). Deux cas peuvent être distingués : (i) si $\lambda g_n \leq \gamma$, le maximum est $\gamma$ (voir $f_{DA}^{max_1}$), (ii) si $\lambda g_n > \gamma$, la Figure IV.4 montre que l'on peut toujours trouver une configuration où $\gamma < f_{DA} \leq \lambda g_n$. Dans ce cas, $f_{DA}^{max_2} = \lambda g_n$. On obtient donc finalement la borne supérieure suivante : $|f_{DA}| = \max(\gamma, \lambda g_n)$.

En insérant les bornes supérieures précédentes dans l'Équation IV.4, on obtient :

$$Z_n \leq W_n^+ e^{-\beta_n \gamma} + W_n^- e^{\beta_n \max(\gamma, \lambda g_n)} = \tilde{Z}_n. \tag{IV.5}$$

Si on calcule la dérivée partielle de $\tilde{Z}_n$ par rapport à $\beta_n$, nous obtenons la valeur de $\beta$ utilisée dans SLDAB :

$$\begin{aligned} \frac{\partial \tilde{Z}_n}{\partial \beta_n} = 0 \quad &\Rightarrow \quad \max(\gamma, \lambda g_n) W_n^- e^{\beta_n \max(\gamma, \lambda g_n)} = \gamma W_n^+ e^{-\beta_n \gamma} \\ &\Rightarrow \quad \beta_n = \frac{1}{\gamma + \max(\gamma, \lambda g_n)} \ln \frac{\gamma W_n^+}{\max(\gamma, \lambda g_n) W_n^-}. \end{aligned} \tag{IV.6}$$

Il est important de noter que $\beta_n$ est calculable si

$$\frac{\gamma W_n^+}{\max(\gamma, \lambda g_n) W_n^-} \geq 1 \Leftrightarrow \gamma(1 - W_n^-) \geq \max(\gamma, \lambda g_n) W_n^- \Leftrightarrow W_n^- < \frac{\gamma}{\gamma + max(\gamma, \lambda g_n)},$$

ce qui est toujours vrai dans la mesure où $h_n$ est un apprenant faible pour l'AD et qu'il satisfait la Condition 2 de la Définition IV.1. De plus, on remarque que selon l'Équation IV.6, $\beta_n$ tend à être de plus en plus petit à mesure que la divergence augmente. En d'autres termes, une hypothèse $h_n$, présentant des poids $W_n^+$ et $W_n^-$, obtiendra une valeur de confiance plus importante qu'une hypothèse $h_{n'}$ aux poids identiques si $g_n < g_{n'}$.

Afin de simplifier les développements à venir, réécrivons $\max(\gamma, \lambda g_n)$ sous la forme $\max(\gamma, \lambda g_n) = c_n \times \gamma$, où $c_n \geq 1$. Nous pouvons réécrire l'Équation IV.6 comme suit :

$$\beta_n = \frac{1}{\gamma(1 + c_n)} \ln \frac{W_n^+}{c_n W_n^-}, \tag{IV.7}$$

et la Condition 2 de la Définition IV.1 devient :

$$W_n^- < \frac{1}{1 + c_n}. \tag{IV.8}$$

### IV.5.3 Convergence de la perte empirique

Le théorème suivant montre que, si la contrainte sur $T$ de l'apprenant faible pour l'AD est satisfaite (c'est-à-dire que $W_n^- < \frac{1}{1+c_n}$), $Z_n$ est toujours plus petit que 1, ce qui conduit (depuis le Théorème IV.1), à une diminution de la perte empirique $\hat{L}_{H_T^N}$ avec le nombre d'itérations.

**Théorème IV.2.** *Si $H_T^N$ est la combinaison linéaire obtenue par* SLDAB *depuis $N$ apprenants faibles pour l'AD, alors* $\lim_{N\to\infty} \hat{L}_{H_T^N} = 0$.

*Démonstration.* En intégrant l'Équation IV.7 dans l'Équation IV.5 nous obtenons :

$$\begin{aligned} Z_n &\leq W_n^+ \left(\frac{c_n W_n^-}{W_n^+}\right)^{\frac{1}{(1+c_n)}} + W_n^- \left(\frac{W_n^+}{c_n W_n^-}\right)^{\frac{c_n}{(1+c_n)}} && \text{(IV.9)} \\ &= \left(W_n^+\right)^{\frac{c_n}{(1+c_n)}} \left(W_n^-\right)^{\frac{1}{(1+c_n)}} \left(c_n^{\frac{1}{(1+c_n)}} + c_n^{-\frac{c_n}{(1+c_n)}}\right) \\ &= \left(W_n^+\right)^{\frac{c_n}{(1+c_n)}} \left(W_n^-\right)^{\frac{1}{(1+c_n)}} \left(\frac{c_n+1}{c_n^{\frac{c_n}{(1+c_n)}}}\right) \\ &= u_n \times v_n \times w_n, && \text{(IV.10)} \end{aligned}$$

où $u_n = (W_n^+)^{\frac{c_n}{(1+c_n)}}$, $v_n = (W_n^-)^{\frac{1}{(1+c_n)}}$ et $w_n = \left(\frac{c_n+1}{c_n^{\frac{c_n}{(1+c_n)}}}\right)$. En calculant la dérivée partielle de $u_n$, $v_n$ et $w_n$ par rapport à $c_n$, nous obtenons

$$\begin{aligned} \frac{\partial u_n}{\partial c_n} &= \frac{\ln W_n^+}{(c_n+1)^2} \left(W_n^+\right)^{\frac{c_n}{(1+c_n)}}, \\ \frac{\partial v_n}{\partial c_n} &= -\frac{\ln W_n^-}{(c_n+1)^2} \left(W_n^-\right)^{\frac{1}{(1+c_n)}}, \\ \frac{\partial w_n}{\partial c_n} &= -\frac{\ln c_n}{(c_n+1)^2} \frac{c_n+1}{c_n^{\frac{c_n}{(1+c_n)}}}. \end{aligned}$$

Nous en déduisons que

$$\frac{\partial Z_n}{\partial c_n} = (\frac{\partial u_n}{\partial c_n} \times v_n + \frac{\partial v_n}{\partial c_n} \times u_n) \times w_n + \frac{\partial w_n}{\partial c_n} \times u_n \times v_n$$

$$= \left(W_n^+\right)^{\frac{c_n}{(1+c_n)}} \times \left(W_n^-\right)^{\frac{1}{(1+c_n)}} \times \left(\frac{c_n+1}{c_n^{\frac{c_n}{(1+c_n)}}}\right) \times \frac{1}{(c_n+1)^2} \times \left(\ln W_n^+ - \ln W_n^- - \ln c_n\right)$$

$$= \left(W_n^+\right)^{\frac{c_n}{(1+c_n)}} \times \left(W_n^-\right)^{\frac{1}{(1+c_n)}} \times \frac{c_n^{\frac{-c_n}{(1+c_n)}}}{c_n+1} \times \left(\ln W_n^+ - \ln W_n^- - \ln c_n\right).$$

Les trois premiers termes de l'équation précédente sont positifs. De ce fait,

$$\frac{\partial Z_n}{\partial c_n} \quad > 0 \Leftrightarrow \ln W_n^+ - \ln W_n^- - \ln c_n > 0 \Leftrightarrow W_n^- < \frac{1}{c_n + 1},$$

ce qui est toujours vrai en raison de l'hypothèse sur l'apprenant faible pour l'AD (voir Équation IV.8). Donc, $Z_n(c_n)$ est une fonction monotone croissante sur $[1, \frac{W_n^+}{W_n^-}[$, avec :

- $Z_n < 2\sqrt{W_n^+ W_n^-}$ (qui est le résultat classique d'AdaBoost) quand $c_n = 1$,
- et $\lim\limits_{c_n \to \frac{W_n^+}{W_n^-}} Z_n = 1$.

Il en découle que $\forall n,\ Z_n < 1 \Leftrightarrow \lim\limits_{N\to\infty} \hat{L}_{H_T^N} < \lim\limits_{N\to\infty} \prod\limits_{n=1}^{N} Z_n = 0.$ □

Donnons maintenant une idée sur la nature de la convergence de $\hat{L}_{H_T^N}$ avec le nombre d'itérations. Une hypothèse $h_n$ est un apprenant faible pour l'AD si $W_n^- < \frac{1}{1+c_n} \Leftrightarrow c_n < \frac{W_n^+}{W_n^-} \Leftrightarrow c_n = \tau_n \frac{W_n^+}{W_n^-}$ avec $\tau_n \in ]\frac{W_n^-}{W_n^+}; 1[$. $\tau_n$ mesure à quel point $h_n$ est proche de la condition de l'apprenant faible. Notons que $\beta_n$ augmente à mesure que $\tau_n$ diminue. De l'Équation IV.10 et de $c_n = \tau_n \frac{W_n^+}{W_n^-}$, on obtient :

$$\begin{aligned}
Z_n &\leq \left(W_n^+\right)^{\frac{c_n}{(1+c_n)}} \left(W_n^-\right)^{\frac{1}{(1+c_n)}} \left(\frac{c_n+1}{c_n^{\frac{c_n}{(1+c_n)}}}\right) \\
&= \left(1 - \frac{\tau_n}{\tau_n + c_n}\right)^{\frac{c_n}{(1+c_n)}} \left(\frac{\tau_n}{\tau_n + c_n}\right)^{\frac{1}{(1+c_n)}} \left(\frac{c_n+1}{c_n^{\frac{c_n}{(1+c_n)}}}\right) \\
&= \frac{c_n^{\frac{c_n}{(1+c_n)}}}{(\tau_n + c_n)^{\frac{c_n}{(1+c_n)}}} . \frac{\tau_n^{\frac{1}{(1+c_n)}}}{(\tau_n + c_n)^{\frac{1}{(1+c_n)}}} . \frac{c_n+1}{c_n^{\frac{c_n}{(1+c_n)}}} \\
&= \left(\frac{\tau_n^{\frac{1}{1+c_n}}}{\tau_n + c_n}\right)(c_n + 1).
\end{aligned}$$

On en déduit que,

$$\begin{aligned}
\hat{L}_{H_T^N} \leq \prod_{n=1}^{N} Z_n = exp \sum_{n=1}^{N} \ln Z_n &\leq exp \sum_{n=1}^{N} \left(\ln\left(\left(\frac{\tau_n^{\frac{1}{1+c_n}}}{\tau_n + c_n}\right)(c_n+1)\right)\right) \\
&= exp \sum_{n=1}^{N} \left(\frac{1}{1+c_n} \ln \tau_n + \ln(\frac{c_n+1}{\tau_n + c_n})\right). \qquad \text{(IV.11)}
\end{aligned}$$

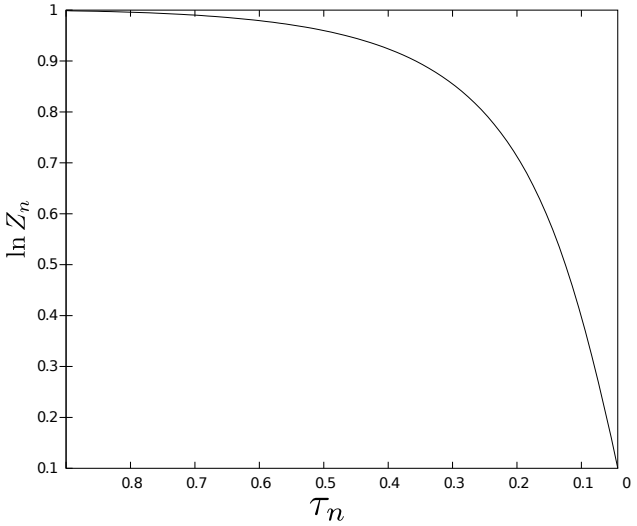



FIGURE IV.5 – Évolution de $\ln Z_n$ par rapport à $\tau_n$.

Le Théorème IV.2 nous indique que le terme entre parenthèses dans l'exponentielle de l'Équation IV.11 est négatif (c'est-à-dire que $\ln Z_n < 0, \forall Z_n$). C'est pourquoi la perte empirique diminue exponentiellement vite vers 0 avec le nombre $N$ d'itérations. De plus, étant donné que $Z_n$ est une fonction monotone croissante de $c_n$ sur $[1; \frac{W_n^+}{W_n^-}[$, c'est également une fonction monotone croissante de $\tau_n$ sur $[\frac{W_n^+}{W_n^-}; 1[$. En d'autres termes, plus $\tau_n$ est faible, plus la convergence de la perte empirique $\hat{L}_{H_T^N}$ est rapide. La Figure IV.5 illustre cette assertion pour une configuration arbitraire de $W_n^+$ et $W_n^-$. Elle montre que $\ln Z_n$, et du même coup $\hat{L}_{H_T^N}$, décroissent exponentiellement vite à mesure que diminue $\tau_n$.

## IV.6 Divergence $g_n$

Nous savons des travaux théoriques sur l'AD [Ben-David et al., 2010, Mansour et al., 2009] qu'une bonne adaptation est possible lorsque la discordance entre les deux distributions est faible et qu'on minimise l'erreur empirique sur le domaine source. Dans notre algorithme SLDAB, cette deuxième condition est assurée par l'utilisation du schéma standard de boosting sur $S$. Pour ce qui est de l'écart de distributions, nous utilisons dans notre algorithme une mesure de divergence $g_n \in [0, 1]$, induite par $h_n$. Dans cette section, nous discutons de la définition de cette dernière.

Une solution pourrait être de calculer une divergence eu égard à la classe d'hypothèses considérée, comme la $\mathcal{H}$-divergence [Ben-David et al., 2010] [3] , introduite au Chapitre III. Cependant, nous pensons qu'une telle mesure n'est pas adaptée à notre cadre spécifique, étant donné que SLDAB

3. Pour rappel, la $\mathcal{H}$-divergence est définie en fonction de la classe d'hypothèses $\mathcal{H}$ par : $\sup_{h,h' \in \mathcal{H}} |\mathbb{E}_{x \sim \mathcal{D}_T}[h(x) \neq h'(x)] - \mathbb{E}_{x' \sim \mathcal{D}_S}[h(x') \neq h'(x')]|$. Elle peut être estimée empiriquement en apprenant un classifieur capable de séparer les exemples sources des exemples cibles, respectivement étiquetés $+1$ et $-1$.

**Entrée :**

- $S = \{x'_1, \ldots, x'_n\}$,
- $T = \{x_1, \ldots, x_m\}$,
- $\varepsilon > 0$,
- une distance $d$.

1. Définir le graphe $\hat{G} = (\hat{V} = (\hat{A}, \hat{B}), \hat{E})$ où $\hat{A} = \{x'_i \in S\}$ et $\hat{B} = \{x_j \in T\}$,
Connecter une arête $e_{ij} \in \hat{E}$ si $d(x'_i, x_j) \leq \varepsilon$.
2. Calculer l'appariement maximal $\hat{G}$.
3. $S_u$ et $T_u$ correspondent au nombre de sommets non appariés dans $S$ et $T$ respectivement.
4. Sortie $\widehat{PV}(S,T) = \frac{1}{2}(\frac{S_u}{n} + \frac{T_u}{m}) \ \in [0,1]$.

**Algorithme 4** : Calcul de $\widehat{PV}(S,T)$ [Harel and Mannor, 2012].

tient compte d'une mesure de divergence induite par **une hypothèse particulière** $h_n$. En effet, le but n'est pas d'évaluer la différence entre les deux domaines de manière globale, mais plutôt d'éviter la sélection d'hypothèses induisant une dégénérescence du modèle. Il est donc essentiel de définir une mesure $g_n$ tenant compte de l'hypothèse elle-même et étant capable (i) d'évaluer l'écart entre la source et la cible et (ii) d'éviter les hypothèses dégénérées.

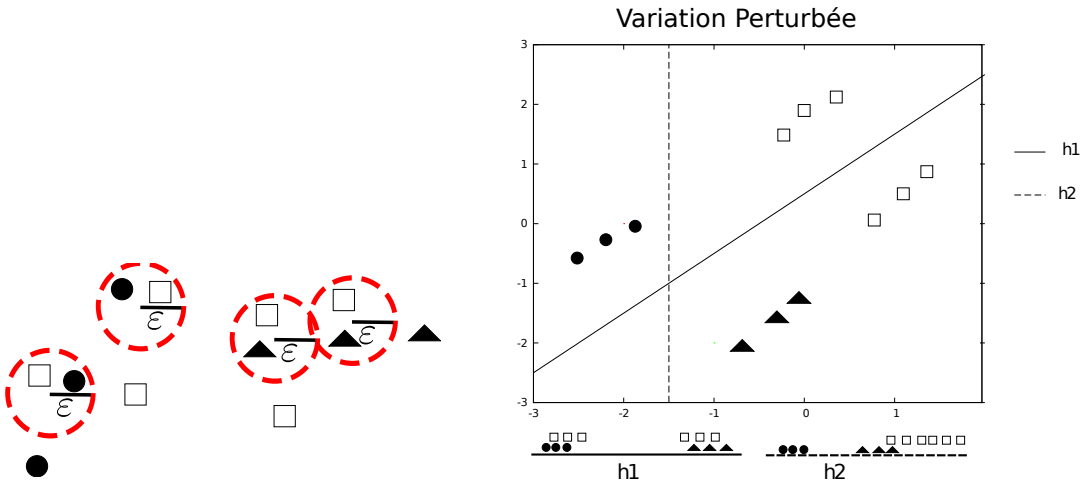
Afin de satisfaire le premier objectif, nous proposons d'utiliser la récente mesure de *variation perturbée*, introduite dans [Harel and Mannor, 2012], qui évalue l'écart entre les deux distributions, tout en autorisant de petites variations, déterminées par un paramètre $\varepsilon > 0$ et une distance $d$.

**Définition IV.2** (Variation perturbée)**.** *Soit $\mathcal{D}_S$ et $\mathcal{D}_T$ deux distributions marginales sur $X$, et $M(\mathcal{D}_S, \mathcal{D}_T)$ l'ensemble de toutes les distributions jointes sur $X \times X$ avec $\mathcal{D}_S$ et $\mathcal{D}_T$. La variation perturbée par rapport à une distance $d : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}$ et $\varepsilon > 0$ est définie par :*

$$PV(\mathcal{D}_S, \mathcal{D}_T) = \inf_{\mu \in M(\mathcal{D}_S, \mathcal{D}_T)} Pr_\mu[d(\mathcal{D}'_S, \mathcal{D}'_T) > \varepsilon]$$

*sur toutes les paires $(\mathcal{D}'_S, \mathcal{D}'_T) \sim \mu$ telles que la distribution marginale de $\mathcal{D}'_S$ (respectivement $\mathcal{D}'_T$) est $\mathcal{D}_S$ (respectivement $\mathcal{D}_T$).*

Un exemple source est donc apparié à un exemple cible, si leur distance est inférieure à $\varepsilon$, selon la mesure de distance $d$, comme illustré par la Figure IV.6(a). Intuitivement, deux ensembles sont similaires si chaque point cible est proche d'un point source en fonction de $d$. Cette mesure est consistante et son estimation empirique $\widehat{PV}(S,T)$ depuis deux ensembles $S \sim \mathcal{D}_S$ et $T \sim \mathcal{D}_T$ peut être calculée efficacement par une procédure d'appariement maximal de graphe résumée dans l'Algorithme 4. Dans notre contexte particulier, nous calculons cette mesure empirique sur les sorties de l'hypothèse courante : $S_{h_n} = \{h_n(\mathbf{x}'_1), \ldots, h_n(\mathbf{x}'_{|S|})\}$, $T_{h_n} = \{h_n(\mathbf{x}_1), \ldots, h_n(\mathbf{x}_{|T|})\}$ en utilisant la




(a) (b)

FIGURE IV.6 – Illustration de l'intérêt de la PV dans notre mesure de divergence. Les exemples sources sont ici représentés par les éléments en noir, tandis que les données cibles sont représentées en blanc. Dans la figure (a), on apparie un exemple source avec un exemple cible, selon la distance euclidienne, lorsqu'ils sont distants de moins de $\varepsilon$. Dans la figure (b), les deux hypothèses $h_1$ et $h_2$ séparent correctement les données sources. Cependant, si on compare les valeurs renvoyées par les hypothèses (comme indiqué en bas de la figure), celles de $h_1$ nous permettent d'apparier tous les exemples sources et cibles, contrairement à celles de $h_2$. Et en effet, $h_1$ sépare correctement les deux classes cibles, tandis que $h_2$ attribue la même classe à tous les exemples cibles. L'utilisation de cette mesure nous permet donc, pour deux hypothèses obtenant la même performance sur les points sources, de sélectionner celle qui est la plus cohérente sur les données cibles.

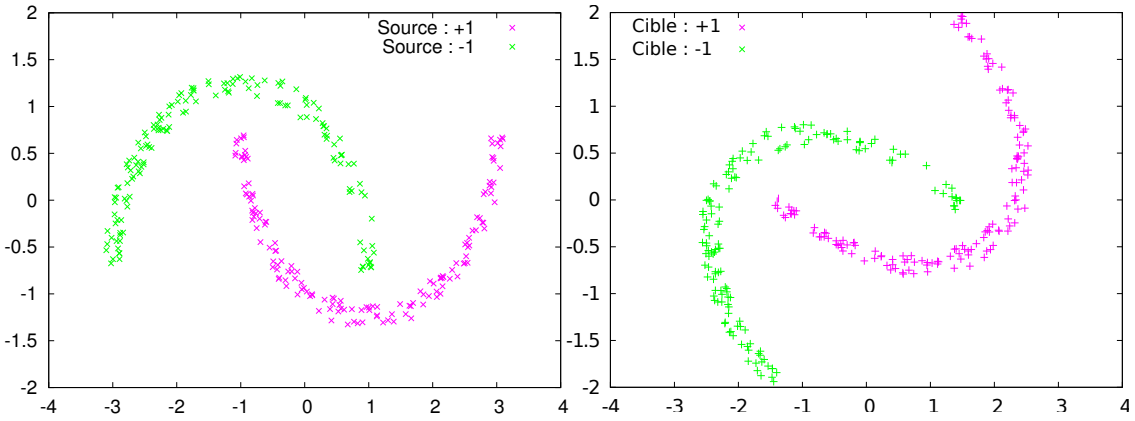
$L_1$ comme mesure de distance $d$ et $1 - \widehat{PV}(S_{h_n}, T_{h_n})$ comme mesure de similarité, comme illustré par la Figure IV.6(b).

Concernant le deuxième objectif visé, nous utilisons la mesure suivante, basée sur la notion d'entropie :

$$ENT(h_n) = 4 \times p_n \times (1 - p_n)$$

où $p_n$ [4] représente la proportion d'exemples cibles recevant une étiquette positive de la part de $h_n$ : $p_n = \dfrac{\sum_{i=1}^{|T|} [h_n(\mathbf{x}_i) \geq 0]}{|T|}$. Cette mesure nous permet d'éviter les hypothèses dégénérées où toutes les données cibles se voient attribuer la même étiquette. En effet, dans ce cas $ENT(h_n)$ est égal à 0. À l'inverse, si les étiquettes sont réparties équitablement entre les deux classes, cette mesure est égale à 1.

4. Nous supposons que les vraies étiquettes sont réparties de façon équivalente, si ce n'est pas le cas, $p_n$ doit être repondéré, en se basant par exemple sur la répartition des étiquettes sources.




(a) (b)

FIGURE IV.7 – Exemples de la base de données MOONS. Dans la figure (a) est représentée la distribution source, tandis que la figure (b) contient les données du domaine cible, obtenues après une rotation de 30°.

Finalement, $g_n \in [0, 1]$ est définie de la manière suivante :

$$g_n = 1 - (1 - \widehat{PV}(S_{h_n}, T_{h_n})) \times ENT(h_n),$$

où $g_n = 1$ si une des deux mesures est nulle.

## IV.7 Résultats expérimentaux

### IV.7.1 Bases de données

Afin de montrer l'intérêt pratique de SLDAB et de confirmer le bien-fondé de l'utilisation du boosting en AD, nous effectuons deux sortes d'expérimentations, respectivement dans un cadre d'adaptation de domaine puis dans un cadre d'apprentissage semi-supervisé. En effet, nous avons pu montrer, via la Définition IV.1, que SLDAB est capable de traiter ces deux types de tâches. Nous utilisons deux bases de données différentes. La première est une base de données synthétique, appelée MOONS [Bruzzone and Marconcini, 2010]. Elle correspond à deux lunes jumelles imbriquées (voir Figure IV.7(a) pour le domaine source et Figure IV.7(b) pour le domaine cible, après une rotation de 30°) dans un espace à deux dimensions où les données suivent une distribution uniforme, chacune des lunes représentant une classe. La seconde est une base de données réelle de SPAMS de l'UCI [5] , contenant 4601 e-mails (2788 d'entre eux étant des "spams" et 1813 "non-spams") dans un espace à 57 dimensions. Sur ces 57 attributs, 48 correspondent à la fréquence d'apparition dans le mail d'un

5. http ://archive.ics.uci.edu/ml/dataset/Spambase

TABLEAU IV.1 – Taux d'erreur sur la base de données MOONS, la colonne "Moyenne" correspond au taux d'erreur moyen sur toutes les rotations, accompagné des écarts-types moyens.

| Angle | 20° | 30° | 40° | 50° | 60° | 70° | 80° | 90° | Moyenne (en %) |
|---|---|---|---|---|---|---|---|---|---|
| **SVM** | 10.3 | 24 | 32.2 | 40 | 43.3 | 55.2 | 67.7 | 80.7 | $44.2 \pm 0.9$ |
| **AdaBoost** | 20.9 | 32.1 | 44.3 | 53.7 | 61.2 | 69.7 | 77.9 | 83.4 | $55.4 \pm 0.4$ |
| **DASVM** | **0.0** | 21.6 | 28.4 | 33.4 | 38.4 | 74.7 | 78.9 | 81.9 | $44.6 \pm 3.2$ |
| **SVM-W** | 6.8 | 12.9 | 9.5 | 26.9 | 48.2 | 59.7 | 66.6 | 67.8 | $37.3 \pm 5.3$ |
| **SLDAB-$\mathcal{H}$** | 6.9 | 11.3 | 18.1 | 32.8 | 37.5 | 45.1 | 55.2 | 59.7 | $33.3 \pm 2.1$ |
| **SLDAB-$g_n$** | 1.2 | **3.6** | **7.9** | **10.8** | **17.2** | **39.7** | **47.1** | **45.5** | $\mathbf{21.6 \pm 1.2}$ |

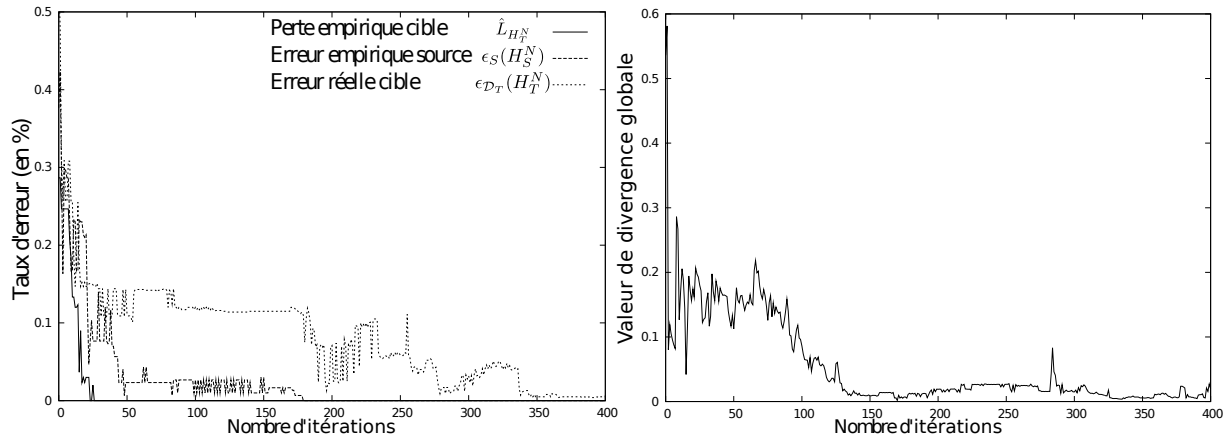
mot donné, 6 représentent la fréquence d'apparition d'un caractère donné, 1 équivaut à la longueur moyenne des chaînes de caractères ne contenant que des majuscules, 1 correspond à la longueur de la plus longue chaîne de caractères en majuscules et le dernier représente le nombre total de majuscules dans le mail.

### IV.7.2 Tâche d'adaptation de domaine

#### IV.7.2.1 Base de données MOONS

Dans cette série d'expérimentations, le domaine cible est obtenu par une rotation anti-horaire du domaine source, correspondant aux données originales. Nous considérons 8 problèmes différents de difficulté croissante, en fonction des angles de rotation, allant de 20 à 90 degrés. Pour chaque domaine, 300 exemples sont générés (150 de chaque classe). Afin d'estimer l'erreur en généralisation, nous utilisons un ensemble indépendant de 1000 exemples, distribués selon la distribution cible. Chacun des problèmes d'adaptation est répété 10 fois et les résultats moyens obtenus sont calculés en faisant abstraction des meilleur et pire tirages.

Nous comparons SLDAB avec quatre approches différentes. Afin de servir de *baseline*, nous utilisons ADABOOST (avec des stumps), et un SVM (avec un noyau gaussien et des hyperparamètres réglés par validation croisée) entraînés uniquement sur le domaine source. Nous comparons également SLDAB avec deux méthodes d'AD : DASVM (basé sur une implémentation LIBSVM) et une approche de repondération pour le problème du covariate shift, présentée dans [Huang et al., 2006]. Cette méthode d'AD non supervisée (que nous nommerons SVM-W), repondère les exemples sources en essayant de rapprocher les distributions source et cible par le processus dit de *Kernel Mean Matching* (voir Section III.3.1), puis infère un classifieur de type SVM en utilisant l'ensemble source repondéré. Enfin, pour confirmer la pertinence de notre mesure de divergence $g_n$, nous lançons SLDAB avec deux divergences différentes : SLDAB-$g_n$ utilise notre nouvelle mesure $g_n$, introduite dans la section précédente, tandis que SLDAB-$\mathcal{H}$ est basé sur la $\mathcal{H}$-divergence, celle-ci étant calculée sur les sorties des hypothèses courantes $h_n$. Nous ajustons les paramètres de SLDAB en sélection-




(a) (b)

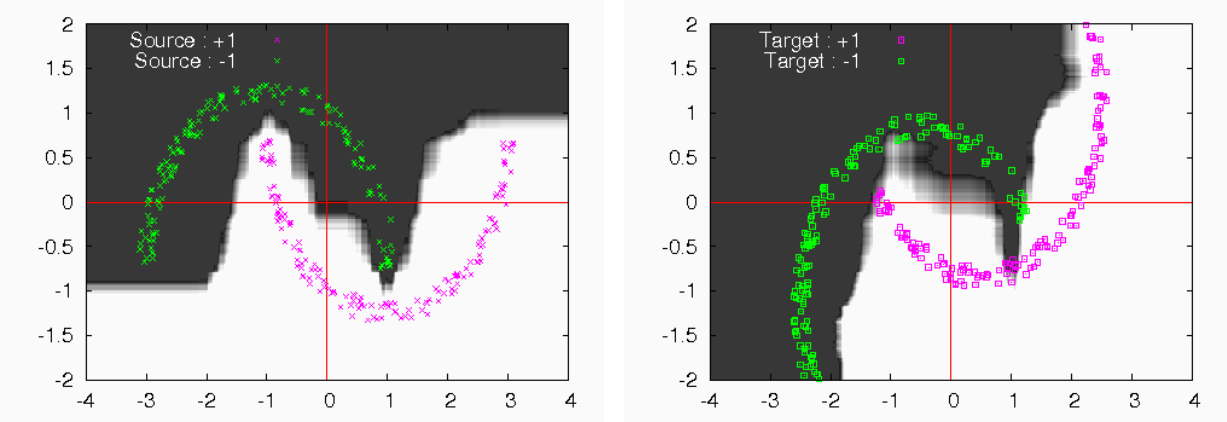
FIGURE IV.8 – (a) : Fonctions de perte sur une tâche à 20° de rotation. (b) : Évolution de la divergence globale.

nant, à l'aide d'une grille de recherche, ceux à même de remplir les conditions de la Définition IV.1 et conduisant à la plus petite divergence par rapport à la combinaison finale $F_T^N$.

Les résultats obtenus sur les différents problèmes d'adaptation sont reportés dans le Tableau IV.1. Nous pouvons remarquer que, excepté pour une rotation de 20 degrés (pour laquelle DASVM est légèrement meilleur), SLDAB-$g_n$ obtient de significativement meilleurs résultats, particulièrement sur des angles de rotation importants. Comme attendu, ADABOOST et les SVMs, entraînés seulement sur les données sources, divergent très vite avec la difficulté de la rotation, montrant la nécessité d'une procédure d'adaptation de domaine. DASVM, quant à lui, n'est pas capable de gérer un trop grand écart entre les deux domaines, entraînant une dégénérescence du modèle, avec des performances moins bonnes que l'aléatoire. Ces résultats montrent que notre approche est plus robuste face à des problèmes d'AD difficiles. Enfin, malgré de bons résultats en comparaison d'autres algorithmes, SLDAB-$\mathcal{H}$ ne fonctionne pas aussi bien que la version utilisant notre divergence $g_n$, montrant ainsi que cette dernière est plus adaptée à notre approche.

La Figure IV.8(a) illustre le comportement de notre algorithme sur un problème correspondant à une rotation de 20 degrés. Premièrement, comme attendu par le Théorème IV.1, la perte empirique sur la cible converge très rapidement vers 0, après environ 25 itérations. L'erreur empirique sur la source $\epsilon_S(H_S^N)$ nécessite plus d'itérations qu'avec ADABOOST standard pour converger, ceci s'expliquant par les contraintes imposées sur les données cibles. Nous pouvons également observer que l'erreur cible $\epsilon_{\mathcal{D}_T}(H_T^N)$ décroît avec $N$ et continue de diminuer même lorsque les deux mesures empiriques ont atteint 0. Ce comportement confirme l'intérêt d'obtenir une faible erreur source ainsi que des marges importantes sur la cible.

La Figure IV.8(b) montre, quant à elle, l'évolution de la divergence globale au fil des ité-




(a) (b)

FIGURE IV.9 – Illustration du comportement de SLDAB sur une tâche à 30° de rotation. (a) : Frontière de décision pour $H_S^N$ sur la source. (b) : Frontière de décision pour $H_T^N$ sur la cible.

rations, calculée sur la combinaison courante $H_T^n = \sum_{k=1}^{n} \beta_k h_k(\mathbf{x})$. Nous pouvons voir que notre schéma de boosting permet une réduction de cette divergence entre les données sources et cibles, ceci expliquant également la diminution de l'erreur en généralisation sur la cible, observée sur la figure.

Enfin, les Figures IV.9(a) et IV.9(b) représentent les zones de décision des modèles inférés sur un problème de rotation à 30 degrés. Les points sont étiquetés négatifs dans la région sombre et positifs dans celle plus claire. En observant la Figure IV.9(a), nous pouvons remarquer que la frontière de décision apprise sur le domaine source (c'est-à-dire $H_S^N = signe(\sum_{n=1}^{N} \alpha_n signe(h_n(\cdot))))$ classifie correctement l'intégralité des exemples de l'ensemble d'apprentissage. Sur la Figure IV.9(b), nous avons reporté la frontière de décision apprise par SLDAB sur l'ensemble cible (c'est-à-dire $H_T^N = signe(\sum_{n=1}^{N} \beta_n signe(h_n(\cdot))))$. Nous pouvons voir que la rotation a été presque parfaitement apprise. Rappelons que cette frontière de décision a été inférée **sans aucune information sur les étiquettes des données cibles**. Les frontières de décision des deux figures montrent bien l'intérêt des deux schémas de pondération différents.

#### IV.7.2.2 Base de données SPAMS

Dans le but de concevoir un problème d'AD depuis des données réelles, issues de cette base de données de l'UCI, nous séparons dans un premier temps la base originale en trois ensembles de même taille. Le premier est utilisé comme ensemble d'apprentissage, représentant la distribution source. Dans les deux autres, un bruit gaussien est ajouté afin de simuler une distribution différente.

Tableau IV.2 – Taux d'erreur sur la base de données Spams.

| Algorithme | Taux d'erreur (en%) |
|---|---|
| **SVM** | 38 |
| **AdaBoost** | 59.4 |
| **DASVM** | 37.5 |
| **SVM-W** | 37.9 |
| **SLDAB-$\mathcal{H}$** | 37.1 |
| **SLDAB-$g_n$** | **35.8** |

Comme tous les attributs sont normalisés dans l'intervalle $[0; 1]$, nous utilisons, pour chaque attribut $n$, une valeur réelle aléatoire dans $[-0.15; 0.15]$ comme moyenne $\mu_n$ et une valeur réelle aléatoire dans $[0; 0.5]$ comme variance $\sigma_n$. Nous générons ensuite le bruit en fonction d'une distribution normale $\mathcal{N}(\mu_n, \sigma_n)$. Après avoir modifié les deux ensembles conjointement en suivant la même procédure, nous en conservons un comme ensemble cible, tandis que l'autre sert d'ensemble test indépendant.

Cette opération est répétée 5 fois. Les résultats moyens des différents algorithmes sont reportés dans le Tableau IV.2. Comme pour le problème sur la base de données Moons, nous comparons notre approche avec AdaBoost standard et un SVM appris uniquement sur la source. Nous la comparons également avec DASVM et SVM-W. Nous pouvons voir que SLDAB obtient de meilleurs résultats que tous les autres algorithmes sur cette base de données réelle. De plus, la version de SLDAB utilisant notre divergence $g_n$ est celle qui mène aux meilleurs résultats.

### IV.7.3 Tâche d'apprentissage semi-supervisé

Notre critère de divergence nous permet de quantifier la distance entre les deux domaines. Si sa valeur est faible durant l'intégralité du processus, cela signifie que nous nous trouvons face à un problème qui ressemble plus à une tâche semi-supervisée qu'à une tâche d'adaptation de domaine. Nous rappelons que dans le cadre de l'apprentissage semi-supervisé, il s'agit d'inférer un modèle depuis un ensemble contenant un faible nombre d'exemples étiquetés et un grand nombre d'exemples non étiquetés, tous **issus de la même distribution**. Dans cette série d'expérimentations, nous étudions le comportement de notre algorithme sur deux variantes semi-supervisées des bases de données Moons et Spams.

#### IV.7.3.1 Base de données Moons

Nous générons aléatoirement un ensemble d'apprentissage de 300 exemples et un ensemble de test indépendant de 1000 exemples issus de la même distribution. Nous sélectionnons ensuite $n$ exemples de l'ensemble d'apprentissage pour lesquels l'étiquette sera disponible, avec $n$ variant de 10 à 50, de telle sorte qu'exactement la moitié des exemples soient étiquetés positifs. Les autres exemples

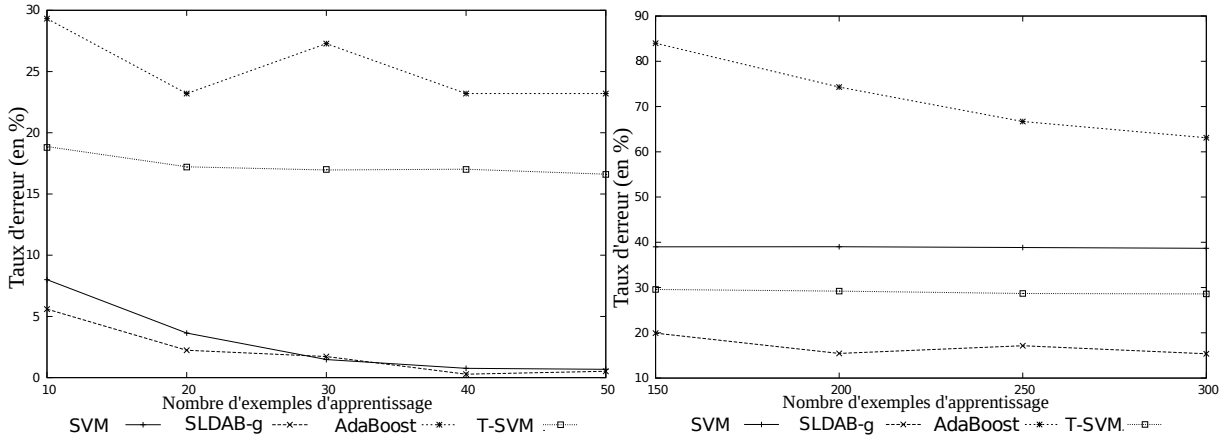



FIGURE IV.10 – (a) : Taux d'erreur des différents algorithmes dans une tâche d'apprentissage semi-supervisé sur la base de données MOONS en fonction du nombre d'exemples étiquetés. (b) : Taux d'erreur des différents algorithmes dans une tâche d'apprentissage semi-supervisé sur la base de données SPAMS en fonction du nombre d'exemples étiquetés.

d'apprentissage constituent les données non étiquetées. Nous évaluons les différentes méthodes en calculant le taux d'erreur sur l'ensemble test. Pour cette expérimentation, nous comparons SLDAB-$g_n$ avec ADABOOST, un SVM et l'algorithme de SVM transductif T-SVM, introduit dans [Joachims, 1999], qui est une méthode destinée à l'apprentissage semi-supervisé, faisant usage de l'information donnée par les exemples non étiquetés pour inférer un classifieur de type SVM. Nous répétons le processus 5 fois pour chacune des valeurs de $n$ et reportons les résultats moyens dans la Figure IV.10.

Notre algorithme obtient de meilleurs résultats que les autres approches sur les ensembles contenant peu de données étiquetées et est compétitif avec le SVM pour les ensembles contenant plus de données labelisées. Nous pouvons également remarquer qu'ADABOOST, utilisant uniquement les exemples sources, n'est pas capable d'obtenir de bons résultats. Ceci peut s'expliquer par un phénomène de sur-apprentissage sur l'ensemble étiqueté de faible taille, entraînant ainsi une performance en généralisation peu satisfaisante. De façon surprenante, T-SVM est peu efficace sur cette tâche. Celui-ci n'arrive donc pas à tirer profit des exemples non étiquetés disponibles.

#### IV.7.3.2 Base de données SPAMS

Nous utilisons ici le même protocole que pour la base de données MOONS dans le cadre semi-supervisé. Nous prenons les 4601 exemples d'origine, issus de la même distribution, et nous les répartissons dans deux ensembles : un tiers des exemples faisant office d'ensemble d'apprentissage, le reste étant utilisé comme ensemble test indépendant afin de calculer le taux d'erreur. Dans l'ensemble d'apprentissage, $n$ exemples sont désignés comme étant les données étiquetées, $n$ variant de 150 à

300, les exemples restants étant utilisés comme données non étiquetées, à l'image de la précédente expérimentation. Cette procédure est répétée 5 fois pour chaque valeur de $n$ et les résultats moyens sont indiqués dans la Figure IV.10.

Nous pouvons remarquer que toutes les approches sont à même de réduire le taux d'erreur à mesure que le nombre d'exemples étiquetés dans l'ensemble d'apprentissage augmente (même si cette diminution n'est pas significative pour SVM et T-SVM), ce qui est un comportement attendu. SVM et encore plus AdaBoost (qui n'utilise pas les données non étiquetées), obtiennent un taux d'erreur important, même avec 300 exemples étiquetés. T-SVM est capable de tirer profit des exemples non étiquetés, obtenant ainsi un taux d'erreur significativement plus faible qu'un SVM classique. Enfin, SLDAB obtient des résultats meilleurs que les autres algorithmes d'au moins 10 points.

Les deux expérimentations sur Moons et Spams confirment que SLDAB est également capable de se comporter de manière adéquate dans un cadre d'apprentissage semi-supervisé. Cet aspect fait de notre approche un outil général et pertinent pour une large classe de problèmes.

## IV.8 Discussion autour des garanties en généralisation

L'étude théorique présentée dans la Section IV.5 nous a permis de dériver plusieurs résultats sur le comportement de SLDAB en apprentissage. Dans cette section, nous discutons des garanties en généralisation. Dans la théorie du boosting [Schapire et al., 1997], rappelons qu'une borne sur l'erreur en généralisation a été introduite, qui a pour principal avantage de ne pas dépendre du nombre d'itérations du processus au niveau du terme de pénalisation.

**Théorème IV.3** (Borne sur l'erreur en généralisation d'AdaBoost)**.** *Soit $\mathcal{H}$ une classe de classifieurs de VC-dimension $d$. $\forall \delta > 0$ et $\gamma > 0$, avec une probabilité $1-\delta$, n'importe quel ensemble de $N$ classifieurs construit depuis un échantillon d'apprentissage $S$ de taille $|S|$ issu d'une distribution $\mathcal{D}_S$ satisfait l'inégalité suivante sur l'erreur en généralisation $\epsilon_{\mathcal{D}_S}(H_S^N)$ :*

$$\epsilon_{\mathcal{D}_S}(H_S^N) \leq \widehat{Pr}_{\mathbf{x}\sim S}[marge(\mathbf{x}) \leq \gamma] + \mathcal{O}\left(\sqrt{\frac{d}{|S|}\frac{log^2(|S|/d)}{\gamma^2} + log(1/\delta)}\right). \tag{IV.12}$$

Ce théorème fait état du fait qu'obtenir une marge importante sur l'ensemble d'apprentissage (le premier terme de la partie droite) permet l'amélioration de la borne sur l'erreur en généralisation. De plus, Schapire et al. ont prouvé qu'avec AdaBoost ce terme décroît exponentiellement vite avec le nombre $N$ d'apprenants faibles. En appliquant le Théorème IV.3 sur l'erreur cible dans le contexte de SLDAB, on peut déduire :

$$\epsilon_{\mathcal{D}_T}(H_T^N) \quad \leq \quad \widehat{Pr}_{\mathbf{x}\sim T}[yF_T^N(\mathbf{x}) \leq \gamma] + \mathcal{O}\left(\sqrt{\frac{d}{|T|}\frac{log^2(|T|/d)}{\gamma^2} + log(1/\delta)}\right), \tag{IV.13}$$

À la différence de $\widehat{Pr}_{\mathbf{x}\sim S}[marge(\mathbf{x}) \leq \gamma]$ dans le Théorème IV.3, nous ne pouvons calculer que la pseudo-valeur $\widehat{Pr}_{\mathbf{x}\sim T}[\mathbf{y}F_T^N(\mathbf{x}) \leq \gamma]$ : en effet, nous utilisons les pseudo-étiquettes pour calculer cette perte durant notre processus d'adaptation, mais la véritable marge d'un exemple nécessiterait l'étiquette réelle. Si le Théorème IV.2 nous a permis de prouver que cette pseudo-perte baisse exponentiellement vite avec $N$, nous ne pouvons pas conclure d'un point de vue théorique sur la décroissance de la perte réelle. Pour contourner ce problème, nous avons tenté d'utiliser certains travaux en AD, bornant l'erreur cible par l'erreur source ajoutée à un terme de divergence. Ceci nous conduirait à une borne sur l'erreur en généralisation de la forme suivante :

$$\epsilon_{\mathcal{D}_T}(H_T^N) \quad \leq \quad \hat{L}_{H_S^N} + div(S,T) + \lambda^\star + \mathcal{O}\left(\sqrt{\frac{d}{|T|}\frac{log^2(|T|/d)}{\gamma^2} + log(1/\delta)}\right).$$

Comme exprimé dans de nombreux travaux sur l'AD, réduire l'erreur en généralisation sur la cible revient à réduire l'erreur empirique sur la source, tout en diminuant la divergence entre les deux distributions. Nous savons que la réduction de l'erreur empirique sur la source est assurée par notre algorithme, par contre nous ne pouvons à nouveau qu'observer la diminution empirique de la divergence globale entre les deux domaines sans pouvoir la prouver. Le problème est donc translaté de la proportion de violations de marge à la divergence entre les deux domaines.

En conséquence, si nous avons pu dériver des garanties de convergence sur la perte empirique, et observer empiriquement une décroissance de la divergence et de l'erreur en généralisation, nous ne sommes pas aujourd'hui en mesure de prouver des garanties en généralisation sur SLDAB.

## IV.9 Conclusion

Dans ce chapitre, nous avons présenté un nouvel algorithme d'AD, basé sur le boosting, appelé SLDAB. Cet algorithme, travaillant dans le cadre difficile de l'AD non supervisée, construit itérativement une combinaison d'apprenants faibles pour l'AD, capable de minimiser dans le même temps l'erreur de classification sur la source et les violations de marge sur les exemples cibles non étiquetés. L'originalité de cette approche dépend principalement de l'introduction d'une nouvelle mesure de divergence entre distributions utilisée durant le processus itératif. Cette divergence donne plus d'importance aux hypothèses capables de rapprocher les distributions source et cible en fonction des sorties des classifieurs. Dans ce contexte, nous avons prouvé théoriquement que notre approche convergeait exponentiellement vite avec le nombre d'itérations. Les expérimentations ont montré le bon comportement pratique de SLDAB dans des problèmes d'AD, à la fois sur des données synthétiques et réelles. De plus, SLDAB est une approche assez généraliste pour lui permettre d'obtenir de bonnes performances dans un cadre d'apprentissage semi-supervisé, en faisant ainsi un algorithme applicable dans de nombreux problèmes.

Même si nos expérimentations ont montré de bons résultats, nous n'avons pas encore prouvé

de garanties sur l'erreur en généralisation sur la cible, même si nous conjecturons que SLDAB permet de la réduire. En effet, la minimisation des violations de marge sur les exemples cibles implique une minimisation de notre divergence dans l'espace induit par les différents classifieurs $h_n$. Les cadres classiques d'AD indiquent que de bonnes capacités en généralisation adviennent quand un algorithme est capable d'assurer à la fois un taux d'erreur faible sur le domaine source, tout en réduisant la discordance entre les deux distributions, qui est en l'occurrence ce que fait SLDAB. Une perspective logique est de montrer que la divergence spécifique que nous avons introduite permet d'obtenir des garanties en généralisation au $\varepsilon$ près utilisé dans la mesure de variation perturbée.

# Chapitre V

## Nouvelle Approche d'Auto-Etiquetage pour l'Adaptation de Domaine sur Données Structurées

**Résumé :** Dans ce chapitre, nous traitons des algorithmes d'auto-étiquetage pour l'adaptation de domaine. Notre contribution comprend plusieurs volets : premièrement, nous proposons une étude théorique visant à définir les conditions nécessaires pour garantir le bon fonctionnement d'un algorithme d'auto-étiquetage. Ensuite, nous appuyant sur ces recommandations théoriques, nous introduisons un nouvel algorithme d'AD, baptisé GESIDA, particulièrement adapté aux données structurées. Cet algorithme tire profit de la récente théorie des $(\varepsilon, \gamma, \tau)$-bonnes fonctions de similarité [Balcan and Blum, 2006, Balcan et al., 2008], qui ne requiert pas l'utilisation d'un noyau valide pour apprendre correctement et engendre des modèles parcimonieux. Finalement, nous appliquons notre algorithme sur des tâches de classification d'images, représentées sous la forme de séquences et d'arbres, et montrons que les résultats obtenus attestent de la pertinence de GESIDA.

**Publications dont est issu le travail de ce chapitre :**

Habrard A., Peyrache J-P., Sebban M.
Iterative Self-Labeling Domain Adaptation for Linear Structured Image Classification
International Journal on Artificial Intelligence Tools (IJAIT), Volume N° 22, Issue N° 5, **2013**

Habrard A., Peyrache J-P., Sebban M.
Domain Adaptation with Good Edit Similarities : a Sparse Way to deal with Rotation and Scaling Problems in Image Classification
Proceedings of the 23rd IEEE International Conference on Tools with Articial Intelligence (ICTAI 2011), 181-188, **2011**
**Prix du meilleur papier**

## V.1 Introduction

Dans ce second chapitre de contribution, nous nous intéressons aux approches d'AD basées sur le principe d'auto-étiquetage. L'algorithme DASVM, présenté dans le Chapitre III, s'inscrit dans ce contexte. Pour rappel, cet algorithme cherche à remplacer itérativement des exemples sources par des exemples cibles semi-étiquetés pour progressivement modifier un classifieur type SVM. À notre connaissance, il n'existe pas de cadre théorique donnant des conditions permettant d'obtenir des garanties de fonctionnement pour des algorithmes d'auto-étiquetage.

Notre première contribution dans ce chapitre a pour objectif d'apporter des réponses à ce sujet. La difficulté ici est de caractériser des conditions à satisfaire lors de chaque itération pour permettre l'adaptation. Les conditions proposées, se basant sur une extension de la notion d'apprenant faible au cadre spécifique de l'auto-étiquetage, tiennent également compte des marges qui définissent une notion de confiance sur l'étiquette attribuée. Leur utilisation est particulièrement adaptée à des approches itératives type DASVM [Bruzzone and Marconcini, 2010], puisque la sélection d'exemples à insérer/enlever à chaque itération dépend de la marge du classifieur courant.

Le second volet de notre contribution s'intéresse au traitement des données structurées, représentées sous forme de chaînes de caractères ou d'arbres, dans un contexte d'AD. La plupart des travaux existant en AD traitent des cas où les données sont d'abord converties sous la forme de données vectorielles. Il est ensuite possible d'utiliser les approches d'AD classiques [Blitzer et al., 2007, Daumé III et al., 2010]. L'inconvénient de ce genre d'approches est d'impliquer une perte d'information, causée par la représentation vectorielle choisie. Nous souhaitons ici traiter directement des chaînes de caractère ou des arbres.

Étendre les approches existantes en AD pour travailler directement sur des données structurées est particulièrement difficile. Ceci peut s'expliquer par le fait qu'il est plus compliqué de (i) définir des mesures de divergence entre des distributions de données structurées ou (ii) concevoir et calculer des mesures de similarité entre des chaînes de caractères, des arbres ou des graphes. Dans ce travail, nous introduisons un nouvel algorithme d'auto-étiquetage pour l'AD qui traite les données structurées en se basant sur le principe de DASVM [Bruzzone and Marconcini, 2010].

Une approche naïve pour traiter les données structurées consisterait simplement à utiliser dans DASVM un noyau **valide** (basé par exemple sur la **distance d'édition** [Wagner and Fischer, 1974]), c'est-à-dire une fonction de similarité qui remplit les conditions de semi-définie positivité (SDP) et de symétrie. Cependant, comme il a été prouvé dans [Cortes et al., 2004], la plupart des fonctions de similarité basées sur la distance d'édition ne sont pas SDP.

Pour contourner ce problème, une approche classique consiste à insérer la distance d'édition $e_d$ dans un noyau Gaussien, de sorte que $K(\mathsf{x}, \mathsf{x}') = e^{-t \times e_d(\mathsf{x}, \mathsf{x}')}$ et à régler le méta-paramètre $t$ par validation croisée afin d'assurer la validité du noyau résultant. En plus du fait qu'une légère modification de $t$ peut entraîner d'énormes différences dans la performance de l'algorithme, l'uti-

lisation des SVMs, ainsi que le paramétrage du noyau, dans un processus itératif d'AD entraînent un coût d'apprentissage important. Pour éviter ces différents problèmes, nous suggérons de relâcher la contrainte du noyau par l'utilisation de la récente théorie des $(\varepsilon, \gamma, \tau)$-bonnes fonctions de similarité, introduite dans [Balcan and Blum, 2006, Balcan et al., 2008], permettant d'apprendre avec des fonctions de similarité non-PSD. Les auteurs prouvent que si une fonction de similarité est $(\varepsilon, \gamma, \tau)$-bonne, alors elle peut être utilisée pour construire un séparateur linéaire [1] de marge $\gamma$ et obtenant un taux d'erreur arbitrairement proche de $\varepsilon$. Il est intéressant de noter que ce séparateur peut être inféré en utilisant un programme linéaire et a de grandes chances d'être parcimonieux, grâce à l'utilisation de la norme $L_1$. Dans ce chapitre, nous exploitons ce cadre afin de concevoir un algorithme d'AD itératif théoriquement fondé.

Nous proposons d'évaluer notre algorithme, baptisé GESIDA (Good Edit Similarity-based Iterative Domain Adaptation) sur une tâche de reconnaissance de chiffres manuscrits, où les chiffres sont représentés par des objets symboliques, encodés sur un alphabet à 8 directions à l'aide de la représentation de Freeman [Freeman, 1974]. Nous considérons deux problèmes d'adaptation, correspondant à des problèmes de changement d'échelle et de rotation.

Ce chapitre est organisé comme suit : dans la section suivante, nous présentons une analyse théorique sur les conditions nécessaires à respecter pour assurer une bonne adaptation de domaine dans le cadre d'un algorithme d'auto-étiquetage. Nous introduisons notre contribution dans la Section V.3, en commençant par présenter la théorie des $(\varepsilon, \gamma, \tau)$-bonnes fonctions de similarité, puis présentons notre algorithme d'AD, tirant parti des bonnes fonctions de similarité, et tendant à satisfaire les exigences théoriques présentées. La Section V.4 nous permet de montrer expérimentalement le bon comportement de notre algorithme, dans une tâche de reconnaissance de caractères manuscrits. Enfin, nous concluons et proposons d'intéressantes perspectives de recherche.

## V.2 Analyse théorique

Un algorithme itératif d'auto-étiquetage pour l'AD incorpore à chaque itération $i$ dans l'ensemble d'apprentissage $S^{(i)}$ des exemples cibles semi-étiquetés, issus de l'ensemble cible $T$, dans le but d'adapter le classifieur courant au concept cible. Etant donné que nous n'avons pas accès aux véritables étiquettes de ces exemples cibles, le choix des points à insérer et la conception de l'algorithme nécessitent une vigilance particulière afin d'éviter des phénomènes de divergence dus à de potentiels mauvais étiquetages successifs. Dans cette section, nous étudions les conditions minimales et nécessaires au succès de tels algorithmes d'auto-étiquetage. Cette étude théorique est effectuée dans le cas basique d'une sélection aléatoire des exemples cibles semi-étiquetés. Par conséquent, le but d'un algorithme d'auto-étiquetage pour l'AD pertinent consistera à satisfaire au minimum ces exigences.

1. Ce séparateur est appris dans un espace explicite où les scores de similarité aux dits **points raisonnables** sont utilisés comme des attributs. $\tau$ représente la proportion minimale de points raisonnables (voir Section V.3.1 ou [Balcan et al., 2008] pour plus de détails).

Nos conditions sont basées sur une notion d'apprenant faible pour l'auto-étiquetage. Avant d'introduire celle-ci, nous commençons par rappeler la définition d'**apprenant faible**, introduite dans [Freund and Schapire, 1996], et précédemment présentée.

**Définition V.1** (Apprenant faible)**.** *Une hypothèse $h_n$ apprise à une itération $n$ est un apprenant faible sur un échantillon d'apprentissage étiqueté $S$, tiré selon $\mathcal{D}_S$, si $h_n$ a un taux de succès au moins un peu meilleur que l'aléatoire, c'est-à-dire $\exists\ \tau_n \in ]0; \frac{1}{2}]$ :*

$$\epsilon_{S^n}(h_n) = \widehat{Pr}_{x_i \sim D_n^S}[h_n(x_i) \neq y_i] = \frac{1}{2} - \tau_n,$$

*où $[.]$ est une fonction indicatrice et $D_n^S$ est la distribution empirique correspondant à $S$.*

Nous étendons cette définition au cadre des algorithmes d'auto-étiquetage, introduisant ainsi l'**apprenant faible pour l'auto-étiquetage**, par rapport à $2k$ exemples cibles semi-étiquetés.

**Définition V.2** (Exemple semi-étiqueté)**.** *Un point cible semi-étiqueté, inséré dans $S^{(i)}$ à une étape $i$, à la place d'un point source est un exemple aléatoirement distribué selon $T$ (sans remplacement), étiqueté par l'hypothèse $h^{(i-1)}$ apprise depuis l'ensemble d'apprentissage précédent $S^{(i-1)}$.*

**Définition V.3** (Apprenant faible pour l'auto-étiquetage)**.** *Un classifieur $h^{(i)}$ appris à une itération $i$ depuis l'ensemble d'apprentissage courant $S^{(i)}$ est un apprenant faible pour l'auto-étiquetage par rapport à un ensemble $SL^j = \{x_1^T \dots x_{2k}^T\}$ de $2k$ exemples cibles semi-étiquetés, insérés à l'étape $j$ si son erreur empirique $\epsilon_{SL^j}^{(j)}(h^{(i)})$ sur ces $2k$ points, par rapport à leur vraie étiquette (inconnue), est plus petite que $0.5$, c'est-à-dire*

$$\epsilon_{SL^j}^{(j)}(h^{(i)}) = \mathbb{E}_{x_l^T \in SL^j}[h^{(i)}(x_l^T) \neq y_l^T] < \frac{1}{2}.$$

Nous donnons maintenant une première condition nécessaire sur l'erreur d'un apprenant faible pour l'auto-étiquetage.

**Théorème V.1.** *Soit $h^{(i)}$ un apprenant faible appris à l'étape $i$ depuis $S^{(i)}$ et $\tilde{\epsilon}_S^{(i)}(h^{(i)}) = \frac{1}{2} - \gamma_S^{(i)}$ son erreur empirique correspondante*[2] *. Soit $\epsilon_T^{(i)}(h^{(i)}) = \frac{1}{2} - \gamma_T^{(i)}$ l'erreur empirique (inconnue) de $h^{(i)}$ sur l'ensemble cible $T$. $h^{(i)}$ est un apprenant faible pour l'auto-étiquetage par rapport à un ensemble $SL^j = \{x_1^T \dots x_{2k}^T\}$ de $2k$ exemples cibles semi-étiquetés insérés à l'itération $j$ ($j \leq i$), si $\gamma_T^{(j-1)} > 0$.*

*Démonstration.* Un exemple semi-étiqueté inséré à l'étape $j$ peut être mal étiqueté par $h^{(i)}$ de deux manières : (i) soit il a obtenu une semi-étiquette différente de son étiquette réelle (attribuée par $h^{(j-1)}$ avec une probabilité de $(\frac{1}{2} - \gamma_T^{(j-1)})$) et celle-ci a été confirmée par $h^{(i)}$ (avec une probabilité de $(\frac{1}{2} + \gamma_S^{(i)})$), (ii) soit il a obtenu une semi-étiquette identique à son étiquette réelle (attribuée par

2. Dans ce contexte semi-étiqueté pour l'AD, nous utilisons le symbole tilde plutôt que l'habituel symbole circonflexe dans la mesure où nous n'avons accès qu'aux semi-étiquettes des exemples de $S^{(i)}$ provenant initialement de $T$, qui peuvent donc être fausses comme exprimé dans la preuve.

$h^{(j-1)}$ avec une probabilité de $(\frac{1}{2}+\gamma_T^{(j-1)})$) et celle-ci a été infirmée par $h^{(i)}$ (avec une probabilité de $(\frac{1}{2}-\gamma_S^{(i)})$). Selon la Définition V.3, $h^{(i)}$ est un apprenant faible pour l'auto-étiquetage par rapport à un ensemble $SL^j = \{x_1^T \dots x_{2k}^T\}$ de $2k$ exemples cibles semi-étiquetés, insérés à une itération $j$ si

$$\begin{aligned} \epsilon_{SL^j}^{(j)}(h^{(i)}) = \mathbb{E}_{x_l^T \in SL^j}[h^{(i)}(x_l^T) \neq y_l^T] &< \frac{1}{2} \\ \Leftrightarrow (\frac{1}{2}+\gamma_S^{(i)})(\frac{1}{2}-\gamma_T^{(j-1)}) + (\frac{1}{2}-\gamma_S^{(i)})(\frac{1}{2}+\gamma_T^{(j-1)}) &< \frac{1}{2} \\ \Leftrightarrow \gamma_S^{(i)}\gamma_T^{(j-1)} &> 0 \\ \Leftrightarrow \gamma_T^{(j-1)} &> 0 \end{aligned}$$

étant donné que $\gamma_S^{(i)} > 0$ selon l'hypothèse de l'apprenant faible.

□

Le Théorème V.1 est simple à interpréter. Il signifie que $h^{(i)}$ sera capable d'étiqueter correctement (par rapport à leur vraie étiquette) plus de $k$ exemples cibles semi-étiquetés parmi $2k$ si au moins la moitié d'entre eux se sont vus attribuer une semi-étiquette correcte.

Par la suite, nous prouvons les conditions nécessaire à un algorithme d'auto-étiquetage pour l'AD, inférant des apprenants faibles pour l'auto-étiquetage, lui permettant d'effectuer une adaptation de domaine efficace.

**Théorème V.2.** *Soient $S^{(0)}$ l'ensemble d'apprentissage d'origine composé de $N$ points source étiquetés et $T$ un ensemble de $|T| \geq N$ exemples cibles non étiquetés. Soit $\mathcal{A}$ un algorithme d'AD itératif qui remplace aléatoirement, à chaque itération $i$, $2k$ points sources étiquetés originaux de $S^{(i)}$ par $2k$ exemples cibles semi-étiquetés distribués aléatoirement depuis $T$ sans remplacement, et infère à chaque étape un apprenant faible (selon la Définition V.3). Soit $h^{(\frac{N}{2k})}$ l'apprenant faible pour l'auto-étiquetage appris par $\mathcal{A}$ après $\frac{N}{2k}$ itérations de ce type, nécessaires pour changer $S^{(0)}$ en un nouvel ensemble d'apprentissage composé uniquement d'exemples cibles. L'algorithme $\mathcal{A}$ effectue une adaptation de domaine efficace avec $h^{(\frac{N}{2k})}$ si*

$$\gamma_S^{(i)} \geq \gamma_T^{(i)}, \forall i = 1, \dots, \frac{N}{2k}, \tag{V.1}$$

$$\gamma_S^{max} > \sqrt{\frac{\gamma_T^{(0)}}{2}}, \tag{V.2}$$

*où $\gamma_S^{max} = max(\gamma_S^{(0)}, \dots, \gamma_S^{(n)})$.*

*Démonstration.* $h^{(\frac{N}{2k})}$ effectue une adaptation de domaine efficace si et seulement si $\epsilon_S^{(\frac{N}{2k})}(h^{(\frac{N}{2k})}) < \epsilon_T^{(0)}(h^{(0)})$. Cette condition signifie que l'hypothèse $h^{(\frac{N}{2k})}$ apprise depuis des exemples cibles semi-étiquetés uniquement doit obtenir un meilleur résultat que $h^{(0)}$, apprise depuis des exemples sources

étiquetés uniquement. Soit $\epsilon^{(j)}_{SL^j}(h^{(i)})$ l'erreur faite par $h^{(i)}$ sur les $2k$ exemples semi-étiquetés insérés dans l'ensemble d'apprentissage à l'itération $j$.

$$\epsilon^{(j)}_{SL^j}(h^{(i)}) = (\frac{1}{2} + \gamma^{(i)}_S)(\frac{1}{2} - \gamma^{(j-1)}_T) + (\frac{1}{2} - \gamma^{(i)}_S)(\frac{1}{2} + \gamma^{(j-1)}_T) = \frac{1}{2} - 2\gamma^{(i)}_S\gamma^{(j-1)}_T.$$

Nous en déduisons que :

$$\begin{aligned}
\epsilon_S^{(\frac{N}{2k})}(h^{(\frac{N}{2k})}) &< \epsilon_T^{(0)}(h^{(0)}) \\
\Leftrightarrow \frac{1}{N}\sum_{j=1}^{\frac{N}{2k}} 2k(\frac{1}{2} - 2\gamma_S^{(\frac{N}{2k})}\gamma_T^{(j-1)}) &< \frac{1}{2} - \gamma_T^{(0)} \\
\Leftrightarrow \frac{4k}{N}\gamma_S^{(\frac{N}{2k})}\sum_{j=1}^{\frac{N}{2k}}\gamma_T^{(j-1)} &> \gamma_T^{(0)} \\
\Leftrightarrow \frac{4k}{N}\gamma_S^{(\frac{N}{2k})}\sum_{j=1}^{\frac{N}{2k}}\gamma_S^{(j-1)} &> \gamma_T^{(0)}, \text{ à cause de la Condition (V.1).} \qquad \text{(V.3)} \\
\Leftrightarrow \frac{4k}{N}\frac{N}{2k}(\gamma_S^{max})^2 &> \gamma_T^{(0)}, \text{ où } \gamma_S^{max} = max(\gamma_S^{(0)}, \ldots, \gamma_S^{(n)}) \qquad \text{(V.4)} \\
\Leftrightarrow \gamma_S^{max} &> \sqrt{\frac{\gamma_T^{(0)}}{2}}.
\end{aligned}$$

□

Avant d'analyser ce théorème, nous interprétons à l'aide du Théorème V.3 la signification de la Condition V.1.

**Théorème V.3.** *Si la Condition V.1 du Théorème V.2 est vérifiée, cela signifie que* $\forall i$,

$$\tilde{\epsilon}_S^{(i+1)}(h^{(i)}) > \tilde{\epsilon}_S^{(i)}(h^{(i)}).$$

*Démonstration.*

$$\begin{aligned}
\gamma_S^{(i)} &\geq \gamma_T^{(i)} \\
\Leftrightarrow \frac{2k}{N}\gamma_S^{(i)} &\geq \frac{2k}{N}\gamma_T^{(i)} \\
\Leftrightarrow (\frac{1}{2} - \gamma_S^{(i)}) + \frac{2k}{N}\gamma_S^{(i)} &\geq (\frac{1}{2} - \gamma_S^{(i)}) + \frac{2k}{N}\gamma_T^{(i)} \\
\Leftrightarrow (\frac{1}{2} - \gamma_S^{(i)}) + \frac{2k}{N}\gamma_S^{(i)} - \frac{2k}{N}\gamma_T^{(i)} &\geq \frac{1}{2} - \gamma_S^{(i)} \\
\Leftrightarrow (1 - \frac{2k}{N})(\frac{1}{2} - \gamma_S^{(i)}) + \frac{2k}{N}(\frac{1}{2} - \gamma_T^{(i)}) &\geq \frac{1}{2} - \gamma_S^{(i)} \\
\Leftrightarrow \tilde{\epsilon}_S^{(i+1)}(h^{(i)}) &\geq \tilde{\epsilon}_S^{(i)}(h^{(i)}),
\end{aligned}$$

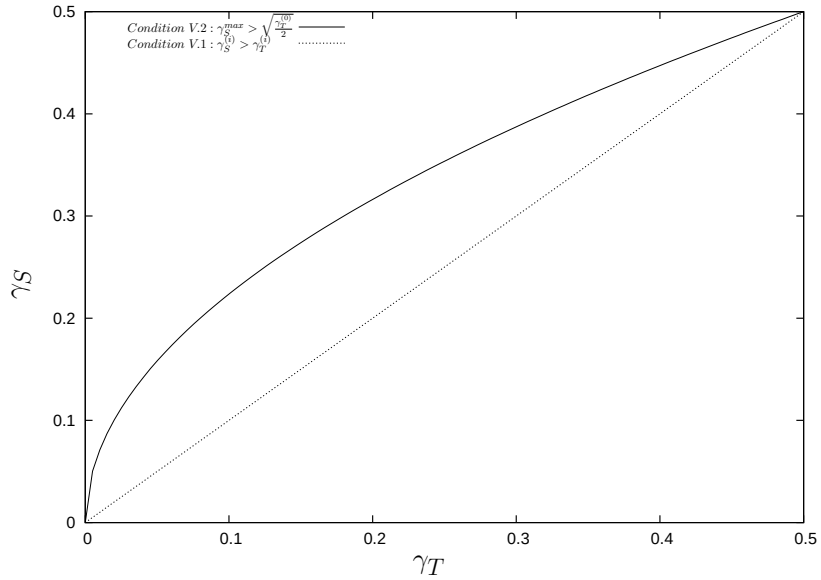



FIGURE V.1 – Conditions nécessaires à l'efficacité d'un algorithme d'auto-étiquetage pour l'AD. La ligne pointillée signifie que, selon la Condition V.1 du Théorème V.2, $\gamma_S^{(i)}$ doit être au moins plus grand que $\gamma_T^{(i)}$ pour apprendre quelque chose de nouveau à chaque itération. La ligne solide exprime la Condition V.2, qui est que $\gamma_S^{max}$ doit aussi être plus grand que $\sqrt{\frac{\gamma_T^{(0)}}{2}}$ (représenté par la zone sous la courbe).

où $\tilde{\epsilon}_S^{(i+1)}(h^{(i)})$ est l'erreur empirique (par rapport aux, potentiellement fausses, étiquettes à la disposition de l'apprenant) de l'ancien classifieur $h^{(i)}$ sur le nouvel ensemble d'apprentissage $S^{(i+1)}$, obtenu après que $2k$ nouveaux exemples cibles semi-étiquetés y aient été insérés. □

Comme dans la théorie du boosting, le Théorème V.3 peut être interprété de la manière suivante. Le classifieur construit depuis $S^{(i+1)}$ doit apprendre quelque chose de nouveau par rapport à la distribution cible $\mathcal{D}_T$. Notons que dans l'algorithme AdaBoost, la contrainte est plutôt $\epsilon_S^{(i+1)}(h^{(i)}) = 0.5$, ce qui est suffisant étant donné qu'AdaBoost construit une combinaison linéaire de tous les apprenants faibles. Comme les algorithmes d'auto-étiquetage ne conservent que la dernière hypothèse inférée, la condition $\epsilon_S^{(i+1)}(h^{(i)}) = 0.5$ n'est pas suffisante. Pour résumer, en contraignant l'ancien classifieur $h^{(i)}$ à fonctionner correctement sur le nouvel ensemble d'apprentissage $S^{(i+1)}$, le Théorème V.3 signifie que l'algorithme d'AD doit inférer une nouvelle hypothèse capable d'apprendre quelque chose de nouveau par rapport à $\mathcal{D}_T$.

Cependant, la Condition V.1 n'est pas suffisante pour effectuer une adaptation efficace. Dans le Théorème V.2, la Condition V.2 exprime l'idée selon laquelle, en plus de la Condition V.1, au moins une hypothèse doit être significativement meilleure que $h^{(0)}$ (appris depuis des données sources uniquement) sur $T$. La Figure V.1 illustre ces deux conditions.

**Discussion**

Avant de présenter notre nouvel algorithme d'AD, nous terminons cette section par une discussion générale.

L'information principale et intuitive donnée par les précédents résultats théoriques est qu'un algorithme d'AD $\mathcal{A}$ adapte correctement si à chaque itération $i$ le classifieur induit $h^{(i)}$ satisfait les conditions suivantes :

- $h^{(i)}$ **doit fonctionner correctement sur** $\boldsymbol{T}$ : l'hypothèse $h^{(i)}$ apprise par $\mathcal{A}$ depuis le domaine source doit avoir une performance raisonnable sur le domaine cible. Cette idée intuitive est exprimée formellement par la définition de l'apprenant faible pour l'auto-étiquetage et le Théorème V.1 ($\gamma_T^{(i)} > 0$).
- $h^{(i)}$ **doit fonctionner correctement sur** $\boldsymbol{S}$ : étant donné que la qualité des exemples cibles insérés dans l'ensemble d'apprentissage dépend essentiellement des semi-étiquettes attribuées par $h^{(i)}$, il est nécessaire d'avoir confiance en $h^{(i)}$. Par conséquent, $h^{(i)}$ doit obtenir une performance suffisamment bonne sur les données depuis lesquelles il a été inféré. Cette idée est exprimée formellement par la Condition V.1 du Théorème V.2 ($\gamma_S^{(i)} > \gamma_T^{(i)}$).
- $\mathcal{A}$ **doit obtenir une meilleure performance qu'un processus non adaptatif** : en effet, le classifieur final $h^{(\frac{N}{2k})}$ appris depuis des données cibles semi-étiquetées uniquement doit obtenir une meilleure performance sur $T$ qu'un classifieur appris depuis des données sources uniquement. Cette idée est décrite par la Condition V.2 du Théorème V.2 ($\gamma_S^{\max} > \sqrt{\frac{\gamma_T^{(0)}}{2}}$) et illustrée par la Figure V.1.

Une seconde remarque par rapport à cette étude théorique est que dans la preuve du Théorème V.2, nous avons utilisé $\gamma_S^{\max}$ pour borner chacun des $\gamma_S^{(i)}$, afin de passer de l'Équation V.3 à l'Équation V.4. Il faut noter que nous aurions pu utiliser une autre stratégie qui aurait mené à un résultat légèrement différent. Revenons à l'Équation V.3, nous obtenons :

$$\begin{aligned}
\frac{4k}{N}\gamma_S^{(\frac{N}{2k})}\sum_{j=1}^{\frac{N}{2k}}\gamma_S^{(j-1)} &> \gamma_T^{(0)}, \text{ à cause de la Condition (V.1).} \\
\Leftrightarrow 2\gamma_S^{(\frac{N}{2k})}\frac{2k}{N}\sum_{j=1}^{\frac{N}{2k}}\gamma_S^{(j-1)} &> \gamma_T^{(0)} \\
\Leftrightarrow 2\gamma_S^{(\frac{N}{2k})}\bar{\gamma}_S &> \gamma_T^{(0)}, \text{ où } \bar{\gamma}_S = \tfrac{2k}{N}\textstyle\sum_{j=1}^{\frac{N}{2k}}\gamma_S^{(j-1)} \\
\Leftrightarrow \bar{\gamma}_S &> \gamma_T^{(0)}, \text{ car } \forall j, \gamma_S^{(j)} < \tfrac{1}{2}.
\end{aligned} \tag{V.5}$$

L'Équation V.5 fournit une condition différente de celle décrite par la Condition V.2. En utilisant la moyenne des différentes valeurs de $\gamma$ plutôt que la valeur maximale, nous obtenons avec

l'Équation V.5 une condition moins contrainte pour une bonne adaptation. En effet, tandis que nous avons utilisé $\frac{N}{2k}$ bornes supérieures pour passer de l'Equation V.3 à l'Équation V.4, nous n'avons recouru qu'à une seule pour arriver à l'Équation V.5. Comme les résultats théoriques présentés dans cette section constituent les exigences minimales pour adapter correctement, nous avons préféré conserver la solution la plus contrainte.

Finalement, notons qu'à travers notre analyse, nous avons utilisé la quantité $\gamma_T$ pour fournir des garanties théoriques sur un algorithme d'auto-étiquetage pour l'AD. Nous pouvons noter qu'en pratique, $\gamma_T^{(i)}$ est inconnu car $T$ est composé à l'origine d'exemples non étiquetés. Clairement, nous affirmons que cela ne remet pas en cause l'intérêt des théorèmes précédents qui donnent plusieurs informations sur la stratégie à adopter pour obtenir une adaptation de domaine effective et sélectionner efficacement les points cibles à insérer. Par exemple, rappelons que DASVM sélectionne de chaque côté du séparateur linéaire les $k$ points situés dans une bande de marge, avec la marge la plus grande. À la lumière de notre étude théorique, cette stratégie tend à satisfaire les conditions requises pour adapter correctement. En effet, d'une part de tels points correspondent à ceux avec la probabilité la plus importante de modifier le SVM suivant (étant donné qu'ils sont probablement associés à des multiplieurs de Lagrange différents de zéro) et ce sont ceux qui **aident à apprendre quelque chose de nouveau sur le domaine cible**. D'autre part, en choisissant ceux qui ont la marge la plus grande, c'est-à-dire le plus éloignés de l'hyperplan, nous augmentons la probabilité que les semi-étiquettes attribuées soient correctes. C'est pourquoi DASVM tend également à satisfaire l'hypothèse de **l'apprenant faible pour l'auto-étiquetage**.

Pour ces raisons, nous nous sommes inspirés de DASVM afin de concevoir un nouvel algorithme d'auto-étiquetage, appelé GESIDA, en mesure de traiter directement des données structurées et qui vise à satisfaire au mieux les exigences théoriques dans le but d'adapter correctement.

## V.3 L'algorithme GESIDA

### V.3.1 Théorie des ($\varepsilon$,$\gamma$,$\tau$)-bonnes similarités

Comme évoqué précédemment, notre contribution s'inspire largement de l'algorithme DASVM, introduit dans [Bruzzone and Marconcini, 2010]. Malgré de bons résultats pratiques, cette méthode présente un certain nombre de limitations. Tout d'abord, aucune étude théorique n'a été menée sur cette approche. Ensuite, la complexité quadratique du problème d'optimisation devant être résolu plusieurs fois entraîne un coût de calcul important. Enfin, notre objectif étant d'exploiter directement des données sous forme de chaînes de caractères ou d'arbres, l'utilisation de DASVM entraînerait deux problèmes majeurs : premièrement, les contraintes algorithmiques sont importantes ; deuxièmement, adapter les SVMs aux fonctions de similarité s'appliquant aux données structurées n'est pas chose aisée. En effet, afin d'être une fonction noyau valide, une fonction de similarité impliquée dans la phase d'apprentissage doit obligatoirement être semi-définie positive et symétrique.

Cependant, comme démontré dans [Cortes et al., 2004], la distance d'édition (qui est la mesure de similarité la plus communément utilisée entre les chaînes et les arbres) n'est pas SDP.

Afin de contourner ce problème, nous suggérons de recourir à la théorie récente de l'apprentissage à base de fonctions de similarité dites $(\varepsilon, \gamma, \tau)$-bonnes, introduite dans [Balcan and Blum, 2006, Balcan et al., 2008]. Ces fonctions présentent comme avantage majeur le fait de ne pas nécessiter d'être SDP pour apprendre correctement et avec des garanties en généralisation. De plus, cette théorie fournit une alternative bien plus parcimonieuse à la théorie des SVMs.

La théorie de Balcan et al. sur l'apprentissage à base de bonnes fonctions de similarité est une généralisation des méthodes à noyau. En effet, un noyau classique est une bonne fonction de similarité. Cependant, cette théorie offre l'avantage d'être applicable également à des fonctions n'étant ni SDP, ni symétriques, nous permettant ainsi de résoudre des problèmes ne pouvant pas être résolus avec l'approche classique des noyaux. L'idée intuitive est qu'une fonction de similarité quelconque est **bonne** si elle a un certain pouvoir discriminant. Plus formellement, les auteurs proposent la définition suivante.

**Définition V.4.** *Une fonction de similarité $K$ est une fonction de similarité $(\varepsilon, \gamma, \tau)$-bonne pour un problème d'apprentissage $P$ s'il existe une fonction (aléatoire) indicatrice $R(x)$, définissant un ensemble (probabiliste) de "points raisonnables", de telle sorte que les conditions suivantes soient respectées :*

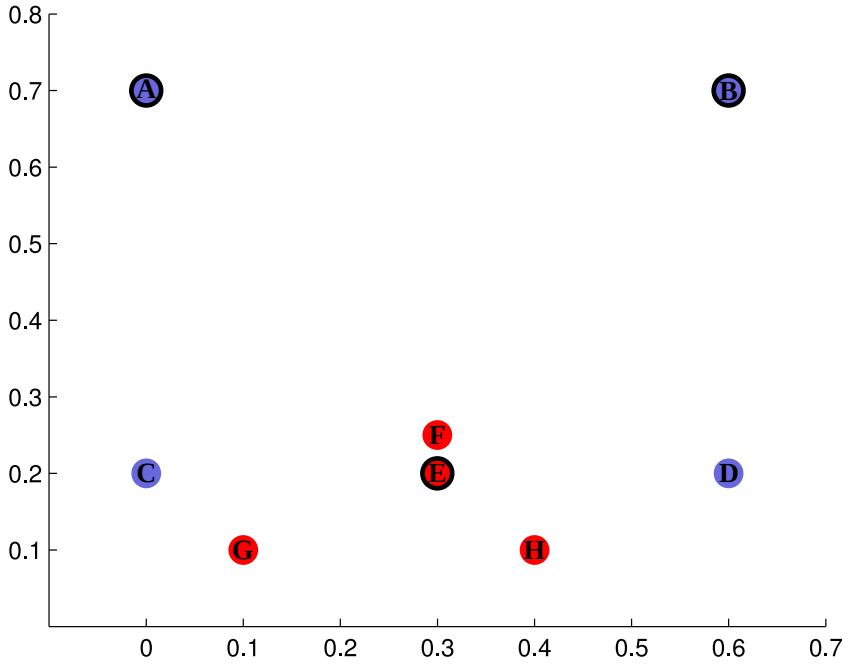
*(1) Une proportion $1 - \varepsilon$ des exemples $(x, y)$ satisfait :*

$$\mathbb{E}_{(x',y')} Pr[yy' K(x, x') | R(x')] \geq \gamma. \tag{V.6}$$

*(2) $Pr_{x'}[R(x')] \geq \tau$.*

La première condition peut être interprétée comme le fait qu'une grande proportion des exemples doit être plus similaire à des points raisonnables de la même classe qu'à des points raisonnables de la classe opposée. La seconde condition indique qu'au moins une proportion $\tau$ des exemples doit appartenir aux points raisonnables. Les auteurs ont prouvé qu'en utilisant des similarités de ce type, il était possible d'apprendre un bon séparateur linéaire dans un espace de projection correspondant aux similarités à l'ensemble des points raisonnables $R$. Ceci est formalisé par le théorème PAC suivant.

**Théorème V.4.** *Soit $K$ une fonction de similarité $(\varepsilon, \gamma, \tau)$-bonne pour un problème d'apprentissage $P$. Soit $S = \{x'_1, x'_2, \ldots, x'_d\}$ un ensemble (potentiellement non étiqueté) de $d = \frac{2}{\tau}\left(\log(2/\delta) + 8\frac{\log(2/\delta)}{\gamma^2}\right)$ points raisonnables de $D$. On considère la reprojection $\phi^S : X \to \mathbb{R}^d$ définie comme suit : $\phi_i^S(x) = K(x, x'_i)$, $i \in \{1, \ldots, d\}$. Alors, avec une probabilité d'au moins $1 - \delta$ sur l'ensemble aléatoire $S$, la distribution induite $\phi^S(P)$ dans $\mathbb{R}^d$ a un séparateur linéaire possédant un taux d'erreur d'au plus $\varepsilon + \delta$ couplé à une marge $L_1$ d'au moins $\gamma/2$.*




| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| A | 1 | 0.4 | 0.5 | 0.22 | 0.42 | 0.46 | 0.39 | 0.28 |
| B | 0.4 | 1 | 0.22 | 0.5 | 0.42 | 0.46 | 0.22 | 0.37 |
| E | 0.42 | 0.42 | 0.7 | 0.7 | 1 | 0.95 | 0.78 | 0.86 |
| **Marge** | 0.3277 | 0.3277 | 0.0063 | 0.0063 | 0.0554 | 0.0106 | 0.0552 | 0.0707 |

FIGURE V.2 – Intuition graphique de la Définition V.4. Soient les 8 points représentés ci-dessus (les points bleus sont ceux de la classe positive, les rouges ceux de la classe négative) et la fonction de similarité $K(\mathbf{x}, \mathbf{x}') = 1 - ||\mathbf{x} - \mathbf{x}'||_2$. Nous avons sélectionné 3 points raisonnables (A,B et E, entourés en noir), ce qui fixe donc $\tau = \frac{3}{8}$. Les scores de similarité aux points raisonnables, ainsi que la marge obtenue par chacun des points (telle que donnée par l'Équation V.6) sont indiqués dans le tableau. Il existe un nombre infini d'instantiations valides de $\varepsilon$ et $\gamma$, étant donné qu'il s'agit d'un compromis entre la marge $\gamma$ et la proportion de violations de marge $\varepsilon$. Par exemple, $K$ est une $(0, 0.006, \frac{3}{8})$-bonne fonction de similarité, car tous les points (équivalant à $\varepsilon = 0$) sont en moyenne plus similaires aux points raisonnables de la même classe qu'aux points raisonnables de la classe opposée, avec une marge $\gamma = 0.006$. Il est également possible de dire que $K$ est une $(\frac{2}{8}, 0.01, \frac{3}{8})$-bonne fonction de similarité ($\varepsilon = \frac{2}{8}$, car les exemples C et V violent la marge $\gamma = 0.01$).

À titre d'exemple (repris depuis [Bellet, 2012]), la Figure V.2 représente 8 points (4 de classe positive, 4 de classe négative) dans un espace à deux dimensions. Ces points ne sont pas linéairement séparables. Trois d'entre eux sont sélectionnés comme étant des points raisonnables, donc les plus représentatifs de leur classe : A et B pour la classe positive, E pour la classe négative. En-dessous de la figure, se trouve le tableau récapitulant les similarités des exemples à chacun des points raisonnables (selon la fonction $K(\mathbf{x}, \mathbf{x}') = 1 - ||\mathbf{x} - \mathbf{x}'||_2$), ainsi que leur marge respective. Sélectionner ces trois points raisonnables nous permet donc de projeter les points dans l'espace à

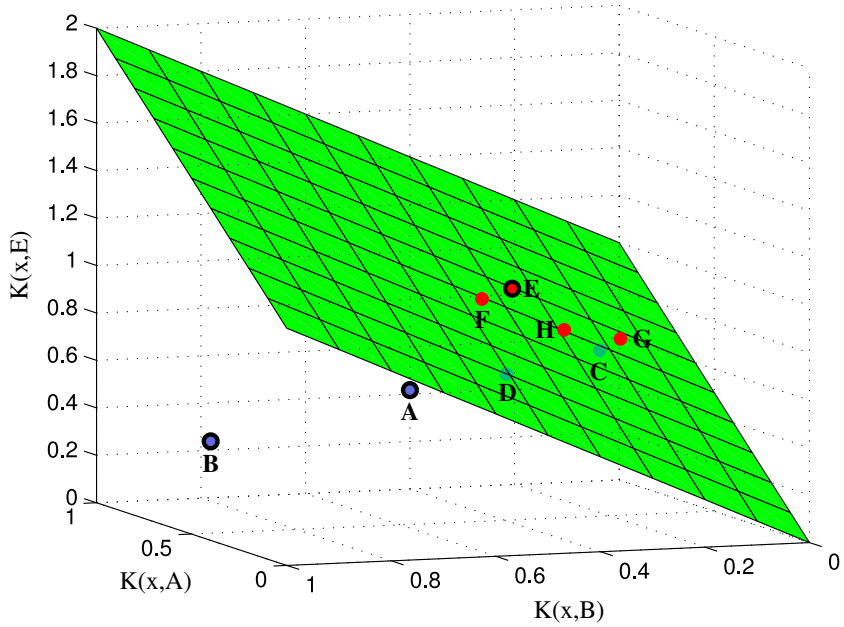



FIGURE V.3 – Espace de projection obtenu sur l'exemple de la Figure V.2 : les scores de similarité aux points raisonnables (A,B et E) sont utilisés comme des nouveaux attributs. Etant donné que $K$ est une fonction de similarité $(0, 0.006, \frac{3}{8})$-bonne pour un $\gamma > 0$, le séparateur linéaire d'équation $K(\mathbf{x}, A) + K(\mathbf{x}, B) - K(\mathbf{x}, E) = 0$ (représenté sur la figure par la grille verte) ne commet aucune erreur de classification, avec une marge $\gamma = 0.006$, bien que les données ne soient pas linéairement séparables dans l'espace d'origine.

trois dimensions représenté dans la Figure V.3, dans lequel il est possible de trouver un séparateur linéaire ne commettant aucune erreur de classification, avec une marge $\gamma = 0.006$.

Donc, étant donnée une fonction de similarité $(\varepsilon, \gamma, \tau)$-bonne, nous sommes capables d'apprendre, avec une forte probabilité, un bon séparateur linéaire dans l'espace explicite induit par $\phi$. Les auteurs ont proposé une formulation, utilisant $d_u$ exemples non étiquetés ainsi que $d_l$ exemples étiquetés, afin d'apprendre efficacement ce séparateur $\alpha \in \mathbb{R}^{d_u}$, correspondant à la résolution du problème d'optimisation suivant :

$$\min_{\alpha} \quad \sum_{i=1}^{d_l} \left[ 1 - \sum_{j=1}^{d_u} \alpha^j y_i K(x_i, x'_j) \right]_+ \quad \text{tel que} \sum_{j=1}^{d_u} |\alpha^j| \leq 1/\gamma, \tag{V.7}$$

où $[1 - z]_+ = \max(0, 1 - z)$ est la perte hinge.

### V.3.2 Distance d'édition

Étant donné que nous souhaitons travailler avec des données structurées, il est nécessaire de définir une similarité adaptée à ce type de données en particulier. Parmi les mesures de similarité

existantes, la distance d'édition [Wagner and Fischer, 1974] est l'une des plus populaires pour les chaînes de caractères.

**Définition V.5** (Distance d'édition)**.** *La distance d'édition $e_d(\mathsf{x}, \mathsf{x}')$ entre deux chaînes de caractères $\mathsf{x}$ et $\mathsf{x}'$ (de longueurs respectives $m$ et $n$) est le nombre minimal d'opérations d'édition à effectuer pour transformer $\mathsf{x}$ en $\mathsf{x}'$. Les opérations autorisées sont l'insertion, la suppression et la substitution d'un symbole.*

$e_d$ peut être calculée en $\mathcal{O}(m \times n)$ en utilisant des techniques de programmation dynamique. Afin de se servir de cette distance en utilisant des SVMs dans des problèmes de classification binaire, une approche classique consiste à insérer $e_d$ dans un noyau gaussien tel que $K(\mathsf{x}, \mathsf{x}') = e^{-t \times e_L(\mathsf{x}, \mathsf{x}')}$, où $t > 0$ est un paramètre. Cependant, il a été prouvé que ce genre de similarité n'était généralement pas un noyau valide [Cortes et al., 2004]. De plus, utiliser de telles approches engendre une perte d'information.

Dans notre contexte, et suivant le cadre théorique de Balcan et al., nous proposons d'utiliser directement la mesure $K(\mathsf{x}, \mathsf{x}') = -e_d(\mathsf{x}, \mathsf{x}')$ comme une bonne fonction de similarité, renormalisée dans l'intervalle $[-1, 1]$. Nous montrons dans la partie expérimentale que cette similarité possède en effet les propriétés d'une fonction $(\varepsilon, \gamma, \tau)$-bonne, ce qui lui permet de s'insérer naturellement dans le cadre théorique de Balcan.

L'idée est donc d'utiliser cette fonction afin d'apprendre des classifieurs linéaires dans un espace de projection explicite (plutôt qu'un espace implicite, comme c'est le cas dans les approches de type SVM), correspondant aux similarités aux points raisonnables. Comme cet ensemble est inconnu à priori, nous utilisons l'intégralité de l'ensemble d'apprentissage $S$ comme points raisonnables. Ceci nous permet d'apprendre un séparateur linéaire de la forme $h(\cdot) = \sum_{\mathsf{x}_i \in S} \alpha_i K(\cdot, \mathsf{x}_i)$ dans le but de réaliser une approche d'adaptation de domaine itérative suivant le principe de DASVM. Nous introduisons maintenant notre nouvel algorithme.

### V.3.3 GESIDA

A l'image de DASVM, GESIDA dispose en entrée d'un ensemble d'apprentissage étiqueté $S$ distribué selon un domaine source $\mathcal{D}_S$ et d'un ensemble cible non étiqueté $T$ distribué selon un domaine cible $\mathcal{D}_T$, les deux étant constitués de données structurées. L'algorithme déplace itérativement des exemples de $T$ dans $S$, dans le but de progressivement modifier un classifieur afin de se rapprocher du concept cible.

Notons $S^{(i)}$ et $T^{(i)}$ respectivement l'ensemble d'apprentissage étiqueté et l'ensemble cible non étiqueté considérés à chaque itération $i$, de telle sorte que $S^{(0)} = S$ et $T^{(0)} = T$. A chaque étape $i$, nous apprenons un classifieur $h^{(i)}$ sur $S^{(i)}$ en résolvant le problème linéaire suivant :

$$\min_{\alpha} \sum_{i=1}^{|S^{(i)}|} \left[ 1 - \sum_{j=1}^{|S^{(i)}|} \alpha^j y_i K(\mathsf{x}_i, \mathsf{x}'_j) \right]_+ \text{, tel que } \sum_{j=1}^{|S^{(i)}|} |\alpha^j| \leq 1/\gamma, \tag{V.8}$$

où $\gamma$ correspond à la marge souhaitée et $K(\mathsf{x}, \mathsf{x}') = -e_d(\mathsf{x}, \mathsf{x}')$ correspond simplement à l'opposé de la distance d'édition, renormalisée dans l'intervalle $[-1, 1]$. Ensuite, nous associons une classe "candidate" $y'^{(i)} = signe(h^{(i)}(\mathsf{x}'))$ à chacun des exemples cibles non étiquetés $x' \in T$. A chaque itération $i, 2k$ exemples semi-étiquetés de l'ensemble courant $T^{(i)}$ sont alors sélectionnés et déplacés dans l'ensemble d'apprentissage $S^{(i)}$, à la place de $2k$ exemples sources originaux. Nous présentons maintenant la stratégie utilisée par GESIDA pour sélectionner les points sources et cibles concernés à chaque itération.

### V.3.4 Sélection des données

Nous avons prouvé précédemment que, pour bien adapter, plus de la moitié des exemples cibles devaient être correctement semi-étiquetés à chaque itération $i$. Etant donné que $\gamma_T^{(i)}$ est inconnu, nous utilisons la marge, qui représente la distance d'un exemple à l'hyperplan séparateur, pour déterminer la confiance que nous avons en l'étiquette qui lui a été attribuée. En effet, plus la distance d'un exemple cible à l'hyperplan séparateur est importante, plus la probabilité que celui-ci soit correctement étiqueté est grande (impliquant probablement $\gamma_T^{(i)} > 0$). D'un autre côté, sélectionner uniquement les exemples cibles avec les marges les plus importantes n'impacterait pas suffisamment le classifieur courant, entraînant ainsi le non respect de la condition $\tilde{\epsilon}_S^{(i+1)}(h^{(i)}) > \tilde{\epsilon}_S^{(i)}(h^{(i)})$, nécessaire pour apprendre quelque chose de nouveau. Dans le but de trouver un bon compromis entre ces deux contraintes, nous proposons une sélection des données en deux étapes :

- Durant la première phase, nous nous concentrons principalement sur le fait de construire un classifieur suffisamment "stable" sur les données cibles. Pour cette raison, nous sélectionnons les $k$ exemples semi-étiquetés de chaque classe possédant les marges les plus importantes, considérant qu'ils sont susceptibles d'être les plus proches des données sources, tout en étant représentatifs de leur classe dans $T$. Comme dans DASVM, les $2k$ points semi-étiquetés ainsi sélectionnés remplacent $2k$ points sources originaux de $S^{(i)}$ ($k$ de chaque classe), en l'occurrence les plus éloignés de l'hyperplan séparateur.

- Durant la seconde phase, nous nous concentrons sur les points plus difficiles, pour lesquels le modèle courant n'est pas suffisamment bon, c'est-à-dire les exemples semi-étiquetés avec une marge inférieure à $\gamma$. Nous insérons tout d'abord les exemples cibles avec une marge légèrement inférieure à $\gamma$ puis nous nous concentrons progressivement sur les données étant situées à une distance de plus en plus petite de l'hyperplan.

Plus formellement, soit $I$ et $k$ deux méta-paramètres de notre algorithme et $T^{(i)} = T_{+1}^{(i)} \cup T_{-1}^{(i)}$ l'ensemble des exemples sélectionnés, défini selon les règles suivantes :

- Première étape, de $i = 0$ à $i = I$ :

  (1) $T_{+1}^{(i)}$ contient les $k$ exemples de $T^{(i)}$ avec les marges $|h^{(i)}(\mathsf{x})|$ les plus grandes, pour $h^{(i)}(\mathsf{x}) > 0$.

  (2) $T_{-1}^{(i)}$ contient les $k$ exemples de $T^{(i)}$ avec les marges $|h^{(i)}(\mathsf{x})|$ les plus grandes, pour $h^{(i)}(\mathsf{x}) < 0$.

- Seconde étape, pour $i > I$ :

  (1) $T_{+1}^{(i)}$ contient les $k$ exemples de $T^{(i)}$ avec les marges $|h^{(i)}(\mathsf{x})|$ les plus grandes, pour $h^{(i)}(\mathsf{x}) > 0$, immédiatement en-dessous de $\gamma - (\frac{k}{|T|+1} \times (i - I))$.

  (2) $T_{-1}^{(i)}$ contient les $k$ exemples de $T^{(i)}$ avec les marges $|h^{(i)}(\mathsf{x})|$ les plus grandes, pour $h^{(i)}(\mathsf{x}) < 0$, immédiatement en-dessous de $\gamma - (\frac{k}{|T|+1} \times (i - I))$.

### V.3.5 Détection d'*outliers* et algorithme

Un risque habituel dans les algorithmes d'auto-étiquetage est l'insertion d'*outliers*[3] (comprendre des exemples semi-étiquetés à qui l'algorithme a attribué la mauvaise étiquette) dans l'ensemble d'apprentissage, particulièrement durant les premières itérations. Dans DASVM, les auteurs remettent les exemples semi-étiquetés de $S^{(i)}$ dans $T^{(i)}$ (sans l'étiquette associée) si leur étiquette change entre deux itérations. Nous proposons de remplacer cette condition forte par une autre, inspirée de la théorie du boosting. Dans l'algorithme BrownBoost, Freund [Freund, 2000] suggère de diminuer l'importance des *outliers* quand il est clair que ceux-ci sont trop difficiles à classifier correctement après un certain nombre d'itérations. Nous appliquons ce principe à notre algorithme, en considérant une valeur de confiance minimale (représentée par la marge $|h^{(i)}(\mathsf{x})|$) devant être atteinte par un exemple semi-étiqueté après un nombre donné d'itérations. Lorsque celle-ci n'est pas obtenue, l'exemple est considéré comme un *outlier* et replacé dans $T^{(i)}$. Notons que pour chaque exemple replacé dans $T^{(i)}$, nous le remplaçons dans l'ensemble d'apprentissage courant par un autre exemple semi-étiqueté, afin de conserver le même nombre d'exemples dans l'ensemble d'apprentissage. Notre processus itératif s'arrête lorsque plus aucun exemple source de l'ensemble d'apprentissage original $S$ n'est présent dans l'ensemble d'apprentissage courant $S^{(i)}$. Le pseudo-code de GESIDA est présenté dans l'Algorithme 5.

3. Le terme français généralement utilisé est celui de "donnée aberrante", mais nous préférerons ici la terminologie anglaise

**Entrée :**

- un ensemble $S$ de $N$ exemples sources étiquetés,
- un ensemble $T$ de $M > N$ exemples cibles non étiquetés,
- deux paramètres $k$ et $I$.

**Sortie :** un classifieur $h$.
Initialisation : $S^{(0)} = S$ ; $T^{(0)} = T$;
**pour** $i = 1$ **à** $\frac{N}{2k}$ **faire**
  Apprendre $h^{(i)}$ sur $S^{(i)}$ en résolvant le Problème V.7
  Calculer la marge des exemples de $S^{(i)}$ et $T^{(i)}$
  Mettre à jour $S^{(i)}$ et $T^{(i)}$ afin de traiter des *outliers* semi-étiquetés dans $S^{(i)}$
  Construire $SL^{(i)}$ et $Sup^{(i)}$ en fonction des paramètres $k$ et $I$
  $S^{(i+1)} \leftarrow SL^{(i)} \cup (S^{(i)} \setminus Sup^{(i)})$
  $T^{(i+1)} \leftarrow T^{(i)} \setminus SL^{(i)}$
**fin**
Renvoyer le classifieur final appris sur $S^{(\frac{N}{2k})}$ en résolvant le Problème V.7;

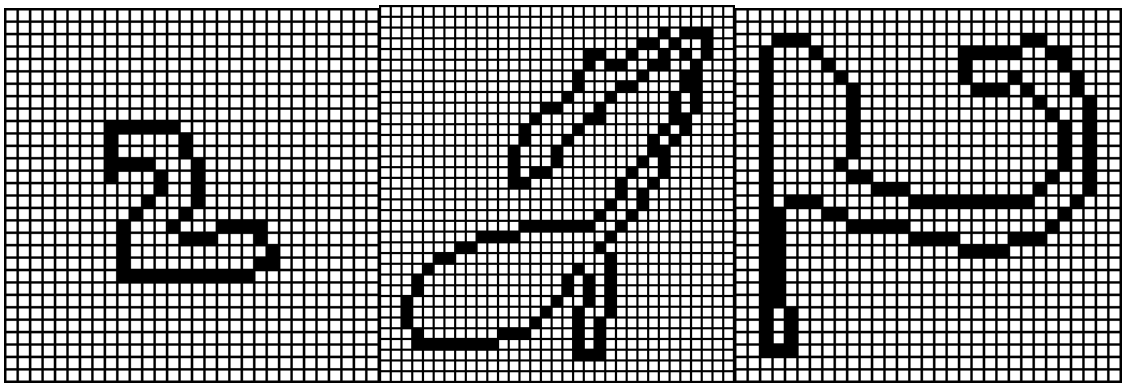
**Algorithme 5** : Pseudo-code de GESIDA.

## V.4 Résultats expérimentaux

### V.4.1 Base de données

Dans cette section, nous fournissons une évaluation expérimentale de notre algorithme d'AD pour les données structurées, sur des représentations à la fois sous forme de chaînes de caractères et d'arbres. Nous proposons de traiter des tâches d'AD originales, en nous attaquant à des problèmes de rotation et de différence d'échelle souvent rencontrés en reconnaissance d'images. Une illustration est donnée par la Figure V.4. Nous utilisons la base classique "NIST Special Database 3" de l'Institut National des Standards et Technologies (NIST), contenant un ensemble de chiffres manuscrits encodés par des images au format bitmap.

Nous menons cinq types d'expérimentations. Premièrement, nous menons une étude préliminaire dont l'objectif est de s'assurer de la qualité des similarités d'édition utilisées dans notre algorithme comme fonctions de similarité $(\varepsilon, \gamma, \tau)$-bonnes. Ensuite, nous fournissons une série d'expérimentations sur des données sous forme de chaînes de caractères, suivie par une autre série utilisant une représentation sous forme d'arbres. Après quoi nous présentons une analyse des points raisonnables (correspondant aux exemples discriminants sur lesquels se base le classifieur) trouvés par l'algorithme tout au long des itérations. Nous terminons cette section par une validation expérimentale de l'étude théorique de la Section V.2. Avant de détailler toutes ces expérimentations, nous commençons par présenter le processus utilisé pour encoder les images en données structurées.



(a) (b) (c)

FIGURE V.4 – Illustration de la base de données des chiffres. (a) représente une instance du chiffre 2 parmi les plus courtes, (b) représente une instance du chiffre 2 parmi les plus longues et (c) correspond à un chiffre 2 ayant subi une rotation.

### V.4.2 Construire des représentations structurées de chiffres manuscrits

La base de données NIST contient des chiffres encodés sous la forme d'images au format bitmap de taille $128 \times 128$. Pour chacun des chiffres (de 0 à 9), sont fournis 1000 exemples différents.

Alors qu'une pratique usuelle en traitement d'images consiste à recourir à des représentations numériques de **sacs de mots visuels** [Salton et al., 1975], de tels vecteurs d'attributs ne permettent pas l'intégration d'informations structurelles additionnelles ou de relations topologiques entre les objets des images. Une alternative efficace peut alors être de représenter les images sous forme de chaînes de caractères.

Différents travaux se sont intéressés à ce problème, proposant diverses approches. Le premier [Freeman, 1974] suggère d'encoder les objets binaires comme des séquences de symboles construites depuis un alphabet à 8 directions. Un autre travail [Daliri and Torre, 2008] introduit une représentation du contour de chacun des objets par une chaîne de caractères dont les composants sont des paires de symboles (l'angle entre un point du contour et son voisin, ainsi que la distance normalisée au centre de gravité). Enfin, dans [Hsieh and Hsu, 2008, Punitha and Guru, 2008] est proposée une représentation sous forme de chaîne de caractères en deux dimensions, encodant la position relative des mots visuels dans l'espace à deux dimensions d'origine, permettant de construire un graphe de similarités codant les relations topologiques entre les objets importants.

Dans ce travail, nous utilisons des chaînes de caractères représentant chaque chiffre sous la forme d'une séquence de codes de Freeman [Freeman, 1974], cette représentation étant à la fois simple à mettre en oeuvre et facilement manipulable. L'idée consiste à suivre le contour du chiffre depuis le pixel le plus en haut à gauche, jusqu'à revenir à ce même pixel. La chaîne de caractères correspondant

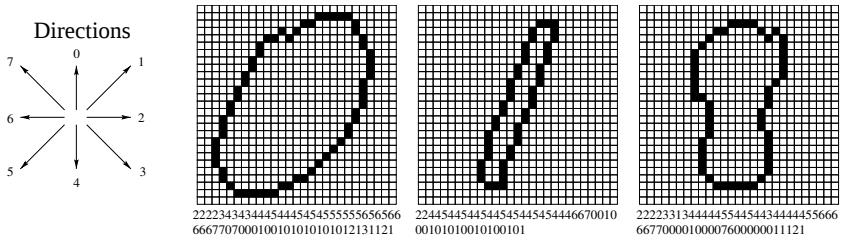



FIGURE V.5 – Images de trois chiffres manuscrits : $0, 1$ et $8$ et leur représentation sous forme de chaîne de caractères.

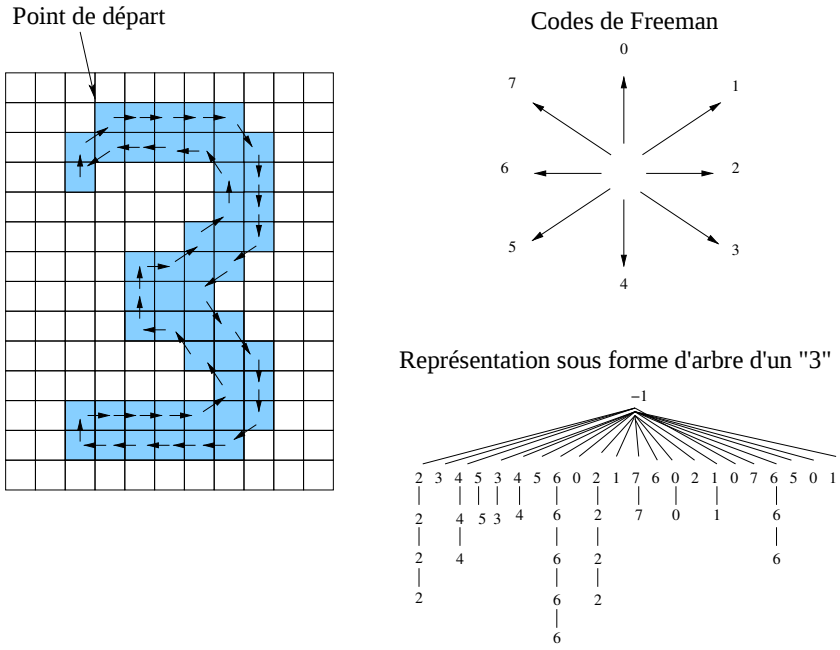



FIGURE V.6 – Image d'un chiffre 3 manuscrit et sa représentation sous forme d'arbre.

au chiffre est donc la séquence de codes de Freeman représentant les directions successives du contour (voir la Figure V.5 pour une illustration).

Pour obtenir des données sous forme d'arbres, nous suivons le principe initialement proposé dans [Bernard et al., 2008]. Afin d'encoder un chiffre donné, l'idée est de premièrement construire une racine, étiquetée par un symbole fictif "-1". L'étape suivante consiste à extraire la représentation sous forme de chaîne de caractères basée sur les codes de Freeman que nous venons de présenter. Cette chaîne de caractères est ensuite parcourue de gauche à droite, chaque nouveau code de Freeman définissant un fils de la racine. Si le même code est trouvé plusieurs fois d'affilée durant l'analyse de la chaîne, chacune des répétitions devient un fils du noeud courant. Un exemple d'une telle représentation est donné dans la Figure V.6.

Pour les chaînes de caractères, nous utilisons la fonction de similarité basée sur la distance d'édition $K(\mathsf{x}, \mathsf{x}') = -e_d(\mathsf{x}, \mathsf{x}')$. Pour les arbres, nous remplaçons $e_d$ par une distance d'édition entre

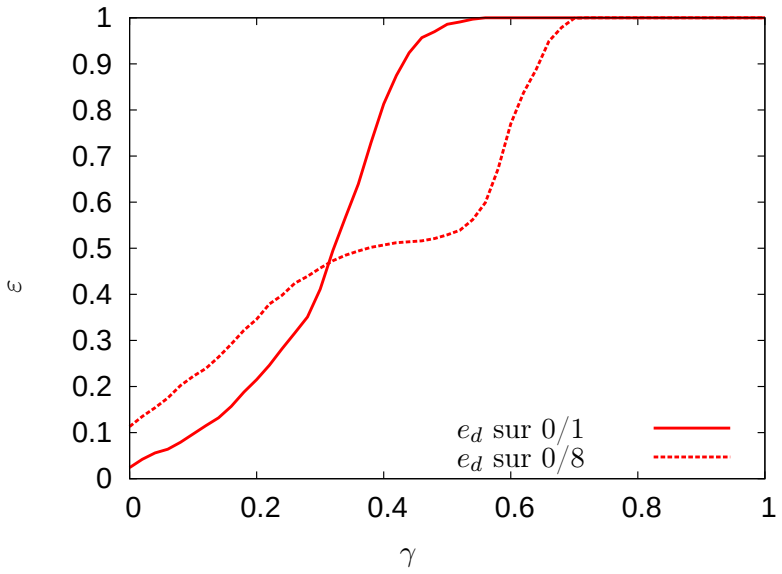



FIGURE V.7 – Estimation de $\varepsilon$ comme une fonction de $\gamma$ sur deux tâches de classification de chiffres manuscrits.

arbres. Dans les expérimentations, nous utilisons l'algorithme de distance d'édition entre arbres de Selkow [Selkow, 1977] (présenté dans l'Annexe A), basé sur les trois opérations d'édition suivantes : substitution de l'étiquette d'un noeud, insertion d'un sous-arbre entier et suppression d'un sous-arbre entier (le lecteur intéressé se référera à [Bille, 2005] pour une présentation des différentes distances d'édition entre les arbres).

### V.4.3 Qualité des similarités d'édition

Au cours de cette étude préliminaire, nous vérifions que la similarité d'édition $K$ possède bien les propriétés des fonctions $(\varepsilon, \gamma, \tau)$-bonnes, eu égard à la Définition V.4 du cadre théorique de Balcan et al. Nous mettons de côté la problématique de l'AD dans cette sous-section et nous estimons la qualité de cette similarité d'édition sur les 45 problèmes binaires de classification supervisée possibles (un pour chaque paire de chiffres), en utilisant 500 exemples pour chaque classe. Nous fixons $\tau$ comme étant égal à 1, signifiant ainsi que nous considérons l'intégralité des points comme pouvant être raisonnables, et estimons $\varepsilon$ comme une fonction de $\gamma$. Pour des raisons de simplicité, nous analysons les résultats obtenus sur deux sous-problèmes uniquement : "les 0 face aux 1" et "les 0 face aux 8", en utilisant une représentation sous forme de chaînes. Les courbes sont reportées dans la Figure V.7. L'interprétation est qu'une marge de $\gamma$ entraîne une proportion $\varepsilon$ des exemples violant cette marge. La Figure V.7 montre que même sans sélectionner un sous-ensemble pertinent de points raisonnables, il y a moins de la moitié des exemples violant la condition de marge pour une valeur de $\gamma$ inférieure à 0.3. Une tendance similaire est observée sur les autres problèmes binaires,

TABLEAU V.1 – Résultats moyens sur les 45 problèmes de classification binaire dans un problème de changement d'échelle, utilisant une représentation sous forme de chaînes de caractères.

| | |
|---|---|
| Taux de bonne classification de DASVM (en %) | $83.3 \pm 3.2$ |
| Nombre final de vecteurs de support | $120 \pm 7.8$ |
| Taux de bonne classification de GESIDA (en %) | $\mathbf{94.7 \pm 2.1}$ |
| Nombre final de points raisonnables | $\mathbf{11 \pm 2.4}$ |
| Taux de bonne classification par sélection aléatoire pour l'AD (en %) | $52.14 \pm 0.6$ |
| Taux de bonne classification sans adaptation (en %) | $50.21 \pm 1.7$ |

ainsi que pour les représentations sous forme d'arbres. De cette expérimentation, nous concluons que la fonction de similarité basée sur la distance d'édition offre des propriétés de qualité raisonnables pour le cadre de Balcan et al., entraînant donc des garanties en généralisation intéressantes.

### V.4.4 Expérimentations sur les données sous forme de chaînes

Nous présentons maintenant les expérimentations réalisées dans le but d'évaluer la capacité de GESIDA à traiter deux importants problèmes d'invariance rencontrés en reconnaissance d'images, à savoir des opérations de rotation et de changement d'échelle. Nous étudions ces problèmes pour chaque tâche binaire possible sur la base de données NIST. Notre objectif est de vérifier si les approches d'AD sont à même de fournir une solution originale à des problèmes de ce type, généralement abordés sous l'angle de la définition d'attributs invariants, comme le fait par exemple SIFT [Lowe, 2004]. Nous comparons quatre approches : (i) DASVM utilisant $K(\mathsf{x}, \mathsf{x}') = e^{-t \times e_d(\mathsf{x},\mathsf{x}')}$ (nous utilisons une simple implémentation basée sur LIBSVM pour mener à bien ces expérimentations), (ii) GESIDA utilisant la similarité d'édition $K(\mathsf{x}, \mathsf{x}') = -e_d(\mathsf{x}, \mathsf{x}')$, (iii) une approche suivant notre principe mais utilisant un processus de sélection aléatoire des exemples à insérer dans et à supprimer de $S^{(i)}$ à chaque itération et (iv) une méthode de base sans adaptation, c'est-à-dire apprenant un modèle uniquement depuis les données sources. Les méta-paramètres de toutes ces approches sont réglés par validation croisée sur un ensemble d'exemples indépendant. Toutes les expérimentations sont effectuées en suivant une procédure de validation croisée en 5 sous-ensembles.

#### V.4.4.1 Problèmes de changement d'échelle

En encodant le contour de chaque image bitmap à l'aide des codes de Freeman, nous obtenons des séquences de symboles dont la longueur représente en quelque sorte la taille du chiffre. Pour traiter ces problèmes de changement d'échelle, nous construisons pour chacun des problèmes binaires un ensemble source $S$ contenant les 100 plus petits exemples de chaque classe, tandis que l'ensemble cible $T$, sur lequel nous voulons adapter notre classifieur, contient 100 chiffres parmi les 200 plus grands exemples. Les 100 restants sont utilisés dans l'ensemble de test.

Dans le Tableau V.1, nous reportons les résultats moyens obtenus sur tous les problèmes

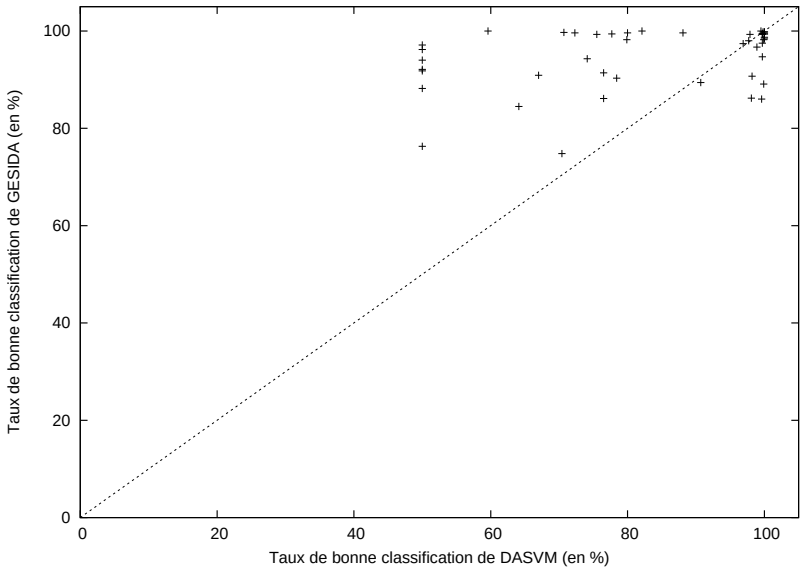



FIGURE V.8 – Comparaison entre GESIDA et DASVM sur les 45 tâches binaires pour les problèmes de changement d'échelle.

binaires, ainsi que l'écart type. Nous pouvons voir que notre approche d'AD basée sur les similarités d'édition est capable d'adapter correctement et est même significativement meilleure que DASVM (par rapport au test de Student). De plus, tirant profit de l'utilisation de la norme $L_1$, GESIDA infère des modèles beaucoup plus parcimonieux, avec un nombre moyen de points raisonnables dix fois plus faible que le nombre moyen de vecteurs de support requis par DASVM. Il est également intéressant de noter que le taux de bonne classification obtenu par un processus de sélection aléatoire est bien moins important que celui obtenu en utilisant notre stratégie de sélection. La dernière ligne du Tableau V.1 montre qu'un modèle appris depuis la source et directement appliqué sur le domaine cible sans adaptation n'est pas meilleur qu'un tirage aléatoire. Ceci confirme expérimentalement l'intérêt de l'adaptation de domaine dans le cas d'un problème de changement d'échelle.

Dans la Figure V.8, nous comparons GESIDA et DASVM sur chacun des 45 problèmes binaires de changement d'échelle. Nous pouvons voir que 28 de ces points sont situés au-dessus de la ligne $y = x$, signifiant ainsi que GESIDA obtient une meilleure performance finale que DASVM pour 28 des problèmes.

Il faut également noter que DASVM obtient un taux de bonne classification d'exactement 50% sur plusieurs problèmes. Ceci s'explique par le fait que l'hypothèse initiale $h^{(0)}$ attribue la même étiquette à tous les points cibles, stoppant ainsi le processus itératif dès le début de l'algorithme.

TABLEAU V.2 – Résultats moyens sur les 45 problèmes de classification binaire dans un problème de rotation, utilisant une représentation sous forme de chaînes de caractères.

| | |
|---|---|
| Taux de bonne classification de DASVM (en %)<br>Nombre final de vecteurs de support | $57.1 \pm 11.7$<br>$113 \pm 9.3$ |
| Taux de bonne classification de GESIDA (en %)<br>Nombre final de points raisonnables | $\mathbf{59.2 \pm 8.1}$<br>$\mathbf{17 \pm 2.7}$ |
| Taux de bonne classification par sélection aléatoire pour l'AD (en %) | $56.63 \pm 7.8$ |
| Taux de bonne classification sans adaptation (en %) | $55.48 \pm 7.7$ |

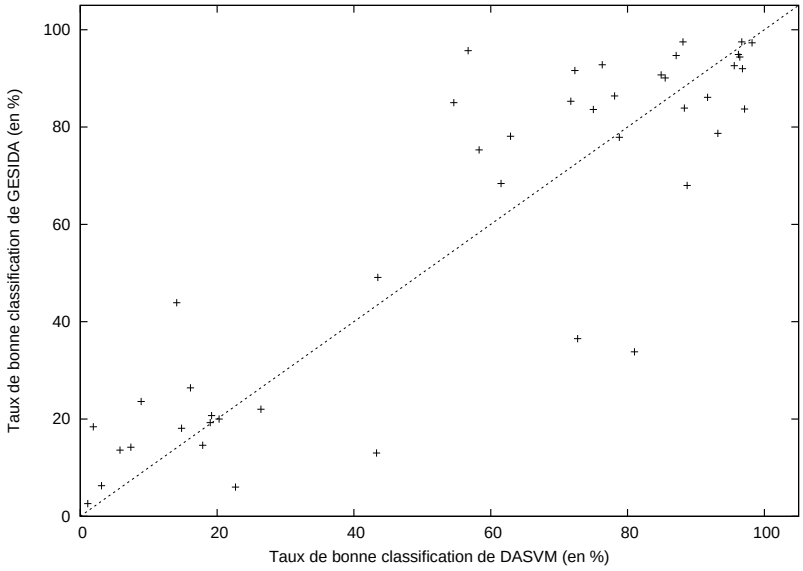



FIGURE V.9 – Comparaison entre GESIDA et DASVM sur les 45 tâches binaires pour les problèmes de rotation.

### V.4.4.2 Problèmes de rotation

Dans cette série d'expérimentations, pour chaque problème binaire, nous choisissons 100 exemples de chaque classe afin de constituer un ensemble d'apprentissage source $S$ et 200 autres exemples de chaque classe sur lesquels nous appliquons une rotation de 90 degrés. Ce protocole nous permet de définir le problème de rotation comme étant une nouvelle tâche d'AD. Comme précédemment, nous conservons 100 exemples ayant subi une rotation pour constituer l'ensemble cible $T$, les 100 autres étant utilisés pour évaluer les différentes méthodes.

Les résultats moyens sur les 45 sous-problèmes de rotation sont indiqués dans le Tableau V.2. Une fois de plus, nous pouvons voir que GESIDA génère des modèles bien plus parcimonieux et obtient en moyenne une performance meilleure que celle obtenue par DASVM. Malgré ce comportement intéressant, il est à noter que les problèmes de rotation ont l'air beaucoup plus difficiles à traiter que ceux de changement d'échelle. Ceci est confirmé par les gains plutôt faibles que l'on peut

TABLEAU V.3 – Résultats moyens sur les 45 problèmes de classification binaire dans le cas du changement d'échelle, utilisant une représentation sous forme d'arbres.

| | |
|---|---|
| Taux de bonne classification de DASVM (en %)<br>Nombre final de vecteurs de support | $66.30 \pm 15.3$<br>$95 \pm 17.6$ |
| Taux de bonne classification de GESIDA (en %)<br>Nombre final de points raisonnables | **67.46 ± 14.9**<br>**16 ± 6.9** |
| Taux de bonne classification par sélection aléatoire pour l'AD (en %) | $53.21 \pm 0.4$ |
| Taux de bonne classification sans adaptation (en %) | $50.04 \pm 0.25$ |

observer en utilisant une méthode d'AD par rapport à une approche n'effectuant aucune adaptation ou utilisant un processus de sélection aléatoire.

La Figure V.9 fournit une comparaison entre GESIDA et DASVM pour les 45 problèmes de rotation. La performance de GESIDA est meilleure que celle de DASVM pour 27 de ces problèmes.

En observant cette figure, nous pouvons voir que sur plusieurs problèmes, l'un des deux, ou parfois les deux, algorithme(s) diverge(nt) complètement, effectuant un transfert négatif et obtenant une performance finale inférieure à un tirage aléatoire. Ce phénomène est dû à la difficulté du problème et explique en partie les faibles moyennes des taux de bonne classification obtenus par les deux algorithmes, les cas menant à une divergence annihilant les bons résultats observés dans d'autres cas.

### V.4.5 Expérimentations sur les données sous forme d'arbres

Dans cette section, nous reportons les résultats expérimentaux obtenus sur les données représentées sous forme d'arbres. Nous restreignons ici notre étude aux problèmes de changement d'échelle, étant donné que les résultats obtenus sur les problèmes de rotation suivent la même tendance que ceux présentés pour les chaînes de caractères. Nous utilisons le même protocole expérimental que celui présenté dans la section précédente, en dehors du fait que nous utilisons une similarité d'édition basée sur la distance de Selkow, calculée entre deux arbres.

Les résultats moyens des 45 problèmes sont indiqués dans le Tableau V.3. Cette expérimentation montre que notre approche est en mesure d'obtenir de meilleurs résultats qu'une sélection aléatoire et qu'une approche n'effectuant aucune adaptation, ce qui confirme que notre algorithme effectue bien une adaptation. GESIDA est, en moyenne, légèrement meilleur que DASVM, mais les taux de précision globaux sont significativement plus faibles que ceux obtenus en utilisant des chaînes de caractères. Cette performance limitée illustre le fait que les représentations sous forme d'arbres utilisées ici ne sont à priori pas totalement pertinentes pour traiter des problèmes de rotation ou de changement d'échelle en reconnaissance d'images [4] . De façon plus intéressante, nous

4. Par exemple, si nous considérons deux séquences de codes de Freeman 22222222 et 22122122, la distance d'édition calculée entre les deux chaînes de caractères est de 2, tandis que la distance obtenue avec l'algorithme de

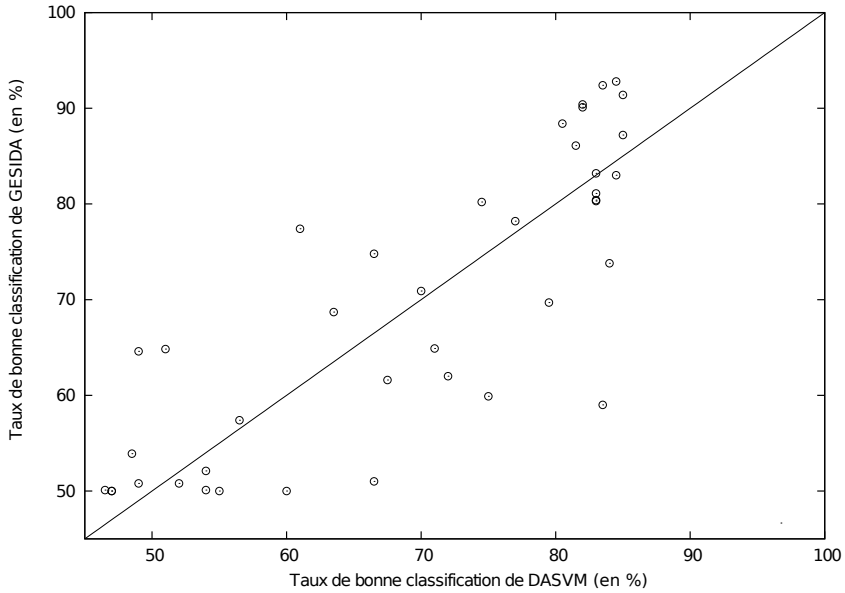



FIGURE V.10 – Comparaison entre GESIDA et DASVM sur les 45 tâches binaires pour les problèmes de changement d'échelle, dans le cadre d'une représentation sous forme d'arbres.

remarquons que les modèles obtenus par GESIDA sont toujours très parcimonieux, entraînant ainsi des coûts de calcul plus faibles.

Dans la Figure V.10, nous comparons les résultats obtenus par GESIDA et DASVM sur chacun des 45 problèmes binaires de changement d'échelle. Nous pouvons voir que 28 des 45 cercles sont au-dessus de la ligne $y = x$, signifiant que GESIDA est meilleur que DASVM pour 28 des sous-problèmes.

### V.4.6 Étude des points raisonnables

Un des aspects intéressants de GESIDA réside dans sa capacité à produire des modèles très parcimonieux, ce qui a été confirmé dans toutes les expérimentations. Il est donc naturel de se demander à quoi ressemblent les exemples sélectionnés en tant que points raisonnables. Intuitivement, ces points devraient être des sortes d'exemples discriminants sur lesquels le classifieur se base. Dans cette section, notre objectif est de les étudier en fonction des différentes itérations du processus d'AD.

Afin d'analyser ces points, nous considérons une tâche visant à séparer les chiffres pairs des chiffres impairs, sur une représentation sous forme de chaînes de caractères. Nous proposons de traiter un problème de rotation en tant que tâche d'AD, étant donné que ce type de problème permet de déduire immédiatement l'origine de chacun des exemples (données sources ou cibles). Nous

Selkow sur leurs représentations sous forme d'arbres est de 12. Une structuration plus complexe semble donc nécessaire dans le cadre de l'utilisation des arbres.

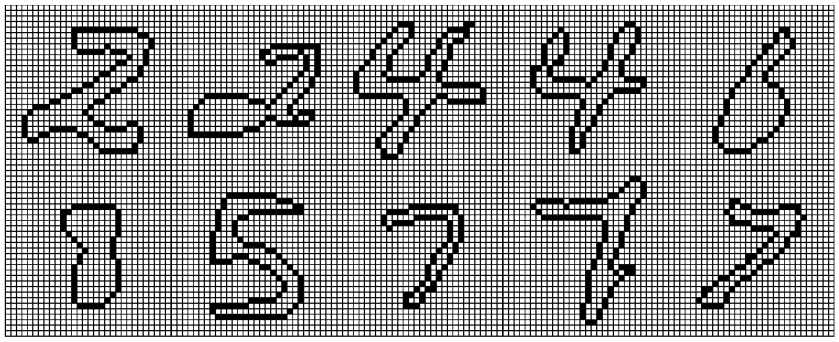


FIGURE V.11 – Ensemble des points raisonnables sélectionnés par le premier classifieur appris par GESIDA pour une tâche visant à séparer les chiffres pairs et impairs.

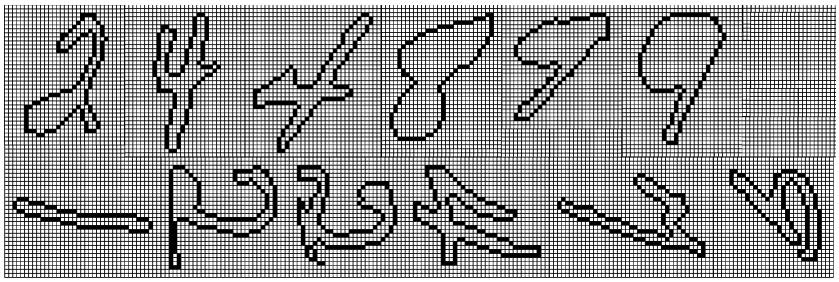


FIGURE V.12 – Ensemble des points raisonnables obtenus au milieu du processus d'AD, traitant un problème de rotation, pour une tâche visant à séparer les chiffres pairs et impairs.

utilisons ensuite le protocole suivant. Tout d'abord, nous construisons un ensemble d'apprentissage contenant 100 chiffres étiquetés "pairs" et 100 étiquetés "impairs" (20 exemples de chaque chiffre). Nous construisons ensuite un ensemble cible selon le même principe, sauf que nous appliquons une rotation de 90 degrés sur chacun des 100 exemples. Finalement, nous lançons notre algorithme GESIDA, en utilisant une valeur de marge élevée pour s'assurer de l'inférence de modèles très parcimonieux.

Dans la Figure V.11, nous montrons un ensemble de points raisonnables, sélectionnés au début du processus, lorsque nous apprenons le premier classifieur sur les données sources uniquement. Cet ensemble de petite taille capture un niveau suffisant de "diversité" sur quelques exemples, lui permettant ainsi d'être capable de discriminer. Les différents 2, 4 et 7 correspondent à plusieurs variations d'un chiffre donné. Nous pouvons noter que certains chiffres ne sont pas représentés, à l'image des 0, des 1 et des 9. Ce peut être justifié par le fait que les contours du 6 et du 8 sont en mesure de couvrir ceux des 0. De façon similaire, la partie basse du 5 peut représenter la partie basse des 9, et les 7, qui ne ressemblent à aucun chiffre pair, peuvent servir à représenter les 1.

Nous considérons maintenant les points raisonnables obtenus par un modèle au milieu du processus d'AD. À cette étape, l'ensemble d'apprentissage contient à la fois des exemples sources et

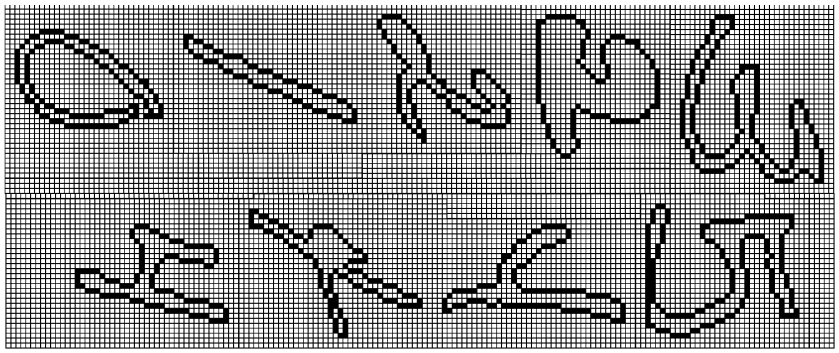


FIGURE V.13 – Ensemble des points raisonnables obtenus à la fin du processus d'AD traitant un problème de rotation, pour une tâche visant à séparer les chiffres pairs et impairs.

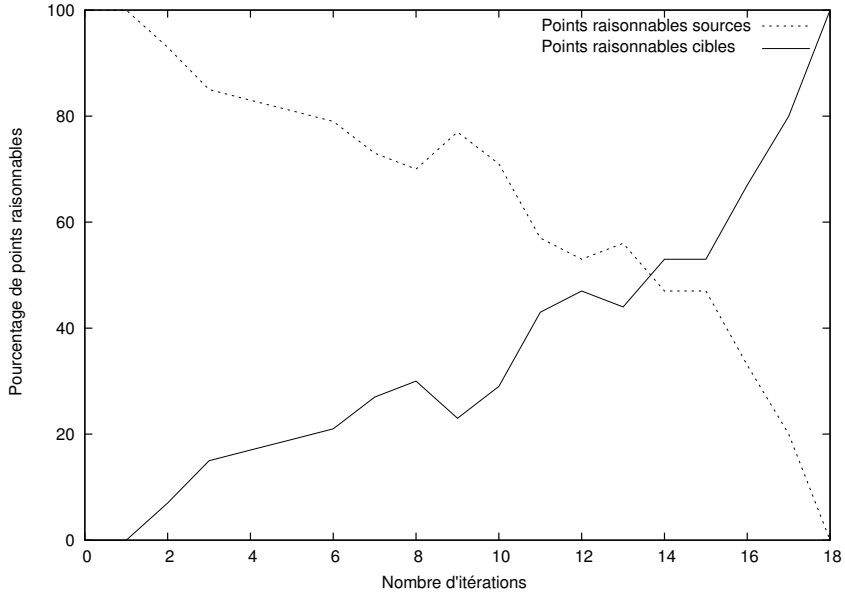



FIGURE V.14 – Proportion de points raisonnables issus des données sources et cibles, en fonction de l'itération du processus d'AD, traitant un problème de rotation, pour une tâche visant à séparer les chiffres pairs et impairs.

cibles, nous amenant à attendre les deux types d'exemples dans l'ensemble des points raisonnables. Les exemples sélectionnés par le modèle sont indiqués dans la Figure V.12. Premièrement, comme attendu, les deux types de points sont présents, mais le nombre total de points sélectionnés est plus important que dans le premier ensemble présenté précédemment. Ceci peut s'expliquer par le fait qu'à cette étape, le modèle doit trouver des exemples discriminants pour les tâches source et cible à la fois, impliquant de considérer plus d'exemples. Les chiffres sont légèrement différents de ceux du premier ensemble et ne couvrent pas la même diversité. Une explication possible est que certains exemples peuvent être mal étiquetés, réduisant ainsi le nombre d'exemples différents que le modèle a à considérer.

TABLEAU V.4 – Expérimentations par rapport à la sélection aléatoire sur deux problèmes de classification.

| Itération | $P_1$ | | | $P_2$ | | |
|---|---|---|---|---|---|---|
| | $\gamma_S^{(i)}$ | $\gamma_T^{(i-1)}$ | 1 - $\epsilon_T^{(i)}$ | $\gamma_S^{(i)}$ | $\gamma_T^{(i-1)}$ | 1 - $\epsilon_T^{(i)}$ |
| 1 | 0.5 | 0 | 0.585 | 0.50 | -0.1 | 0.32 |
| 2 | 0.475 | 0.085 | 0.75 | 0.50 | -0.18 | 0.285 |
| 3 | 0.48 | 0.25 | 0.73 | 0.50 | -0.215 | 0.285 |
| 4 | 0.49 | 0.23 | 0.795 | 0.50 | -0.215 | 0.24 |
| 5 | 0.49 | 0.295 | 0.875 | 0.50 | -0.26 | 0.18 |
| 6 | 0.49 | 0.375 | 0.94 | 0.50 | -0.32 | 0.205 |
| 7 | 0.49 | 0.44 | 0.94 | 0.50 | -0.295 | 0.19 |
| 8 | 0.49 | 0.44 | 0.94 | 0.50 | -0.31 | 0.12 |
| 9 | 0.49 | 0.44 | 0.94 | 0.50 | -0.38 | 0.145 |
| 10 | 0.495 | 0.44 | 0.985 | 0.50 | -0.355 | 0.115 |
| 11 | 0.5 | 0.485 | 0.99 | 0.495 | -0.385 | 0.115 |

Les points raisonnables sélectionnés pour le dernier modèle appris durant le processus d'AD sont représentés dans la Figure V.13. Comme attendu, cet ensemble ne contient que des chiffres issus de l'ensemble cible. Il est également plus petit que celui obtenu au milieu du processus, ce qui peut être justifié par le fait que l'algorithme n'a besoin de s'appuyer que sur un seul type d'exemples (les exemples cibles dans ce cas). Cet ensemble est également quelque peu différent du premier : les 7 ont été remplacés par un 1, le 8 et le 6 par un 0. La diversité est capturée ici par des exemples de 4 et de 2, mais on observe un manque de représentativité pour les chiffres impairs. Ce dernier point explique en partie pourquoi nous n'obtenons pas de bons résultats pour ce problème de rotation.

Finalement, dans la Figure V.14, nous montrons la proportion de points raisonnables issus des données originales sources et cibles en fonction des différentes itérations. Les points cibles deviennent les plus représentés uniquement à partir des trois dernières itérations, montrant ainsi que les exemples sources conservent une grande importance, même après le milieu du processus.

### V.4.7 Évaluation expérimentale de l'approche de sélection aléatoire

Dans cette dernière série d'expérimentations, nous cherchons à vérifier que les conditions théoriques présentées dans la Section V.2 pour adapter correctement sont confirmées empiriquement dans le contexte d'une sélection aléatoire des exemples. Nous supposons ici que nous avons accès à la véritable étiquette des exemples cibles, afin d'être en mesure de calculer $\gamma_T^{(i)}$, habituellement inconnu. Nous considérons deux tâches d'AD : $P_1$ correspond à un problème de classification binaire de changement d'échelle entre des 0 et des 1. $P_2$, quant à lui, correspond à un problème classification binaire de rotation entre des 5 et des 7. Ces deux tâches sont sélectionnées de telle manière que GESIDA fonctionne correctement sur $P_1$ et diverge sur $P_2$. Nous reportons dans le Tableau V.4 les résultats des différentes valeurs pour $\gamma_S^{(i)}$, $\gamma_T^{(i)}$ et $1 - \epsilon_T^{(i)}$.

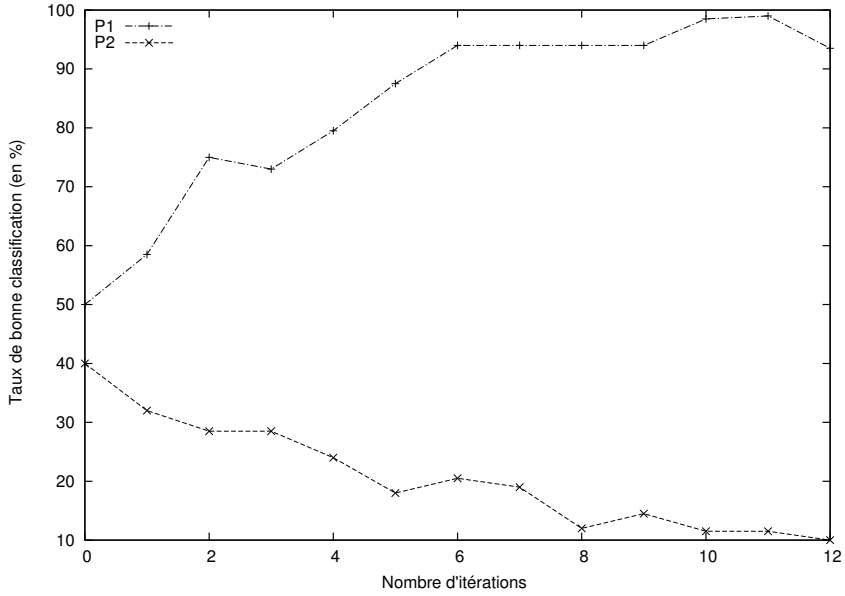



FIGURE V.15 – Évolution de l'erreur en généralisation sur la cible pour les deux problèmes $P_1$ et $P_2$.

Premièrement, nous pouvons remarquer que dans les deux cas, les conditions présentées dans le Théorème V.2 sont remplies durant toutes les itérations, c'est-à-dire que $\forall i, \gamma_S^{(i)} > \gamma_T^{(i)}$ et $\gamma_S^{\max} > \sqrt{\frac{\gamma_T^{(0)}}{2}}$. Cependant, les hypothèses inférées à chaque itération du problème $P_2$ ne sont pas des apprenants faibles pour l'auto-étiquetage, selon la Définition V.3. Ceci explique le phénomène de divergence observé sur ce problème. D'autre part, dans le problème $P_1$, toutes les hypothèses sont des apprenants faibles pour l'auto-étiquetage, justifiant ainsi la convergence de l'algorithme. L'inférence d'apprenants faibles, associée aux deux conditions nécessaires du Théorème V.2, assure une bonne adaptation. L'évolution de l'erreur en généralisation sur la cible au fil des itérations pour $P_1$ et $P_2$ est indiquée dans la Figure V.15. Celle-ci illustre bien que la divergence est immédiate dès le début des itérations, tandis que le taux de classification correcte augmente régulièrement quand une bonne adaptation est possible. Cette expérimentation met donc en lumière le fait que l'étude théorique menée dans la Section V.2 donne bien les conditions nécessaires à une bonne adaptation.

## V.5 Conclusion

Dans ce chapitre, nous avons proposé une nouvelle approche itérative pour l'AD, GESIDA, spécifiquement conçue pour traiter directement des données structurées telles que les chaînes de caractères ou les arbres. À l'inverse des méthodes projetant de telles données dans un espace numérique, nous les manions en utilisant une similarité basée sur la distance d'édition et possédant des capacités de généralisation, selon la théorie des $(\varepsilon, \gamma, \tau)$-bonnes fonctions de similarité, nous permettant ainsi de nous en servir dans le cadre de Balcan et al. Nous avons tiré parti de ce cadre pour

proposer un algorithme d'adaptation de domaine utilisant les similarités d'édition. Notre approche incorpore itérativement des exemples cibles dans l'ensemble d'apprentissage, en les sélectionnant en fonction de leur marge courante.

Nous avons également fourni une analyse théorique abordant le cas général des méthodes d'auto-étiquetage, en donnant les conditions minimales et nécessaires pour réussir à adapter correctement. Ces conditions s'appuient notamment sur la nécessité d'avoir des apprenants faibles pour l'auto-étiquetage et une relation minimale entre l'apprenant faible initial et la meilleure hypothèse trouvée au long des itérations.

Nous avons évalué GESIDA en considérant des tâches d'AD originales, se concentrant sur des problèmes de robustesse à des déformations basées sur le changement d'échelle et la rotation, dans le cadre de la reconnaissance d'images. Notre méthode a montré de bonnes capacités d'adaptation, particulièrement sur les problèmes de changement d'échelle en utilisant des données représentées sous la forme de chaînes de caractères, mais également en se montrant meilleure que DASVM en moyenne. Une caractéristique importante de l'algorithme est le fait que les modèles inférés soient extrêmement parcimonieux. Cette caractéristique est cruciale du point de vue des applications à grande échelle, étant donné que le calcul des similarités d'édition est généralement coûteux.

Une première perspective théorique concerne le mode de sélection des exemples. Les résultats de notre étude ont été établis dans le cas basique d'une sélection aléatoire des exemples cibles. Par conséquent, les résultats sont indépendants de la stratégie de sélection des exemples utilisée dans l'algorithme. Une analyse capable de tenir compte de cette stratégie permettrait donc d'obtenir des résultats plus fins et plus informatifs sur le comportement de l'algorithme. Une autre direction pourrait consister à essayer de dériver des bornes en généralisation, montrant ainsi que notre approche est en mesure de converger itérativement vers l'erreur jointe sur les deux domaines. D'un point de vue applicatif, une perspective possible serait d'étudier l'influence d'autres représentations structurées pour les images. Étudier la capacité des méthodes d'AD à traiter des problèmes d'invariance plus généraux dans des tâches de reconnaissance d'images est aussi un travail futur qui pourrait être intéressant. Finalement, nous visons également à appliquer ce genre d'approche d'AD itérative dans des problèmes de régression ou pour la classification de données temporelles.

# Chapitre VI

## Conclusion Générale et Perspectives

Dans cette thèse, nous avons apporté des contributions théoriques et algorithmiques en adaptation de domaine non supervisée. L'objectif était de contourner un certain nombre de limitations de l'état de l'art, notamment :

- Le manque de cadre théorique pour les méthodes d'auto-étiquetage.
- L'absence de prise en compte explicite des risques de transfert négatif dans les algorithmes d'AD.
- L'insuffisance des méthodes traitant de l'AD sur des données structurées.

Notre première contribution, basée sur la théorie du boosting, visait à traiter de l'AD non supervisée sur des données vectorielles. Dans ce but, nous avons tiré profit du boosting en définissant des apprenants faibles pour l'AD, tenant compte d'une mesure de divergence entre les deux distributions. Ces apprenants faibles sont itérativement combinés afin d'obtenir un classifieur final fort au sens PAC. Nous avons dérivé plusieurs garanties théoriques, notamment sur les violations de marge des exemples cibles (qui sont le seul élément objectif dont on dispose pour évaluer les exemples non étiquetés), dont la proportion converge vers 0 avec le nombre d'itérations. Les garanties théoriques d'AdaBoost sont également conservées, et en particulier la convergence vers 0 de l'erreur empirique sur la distribution source. Enfin, nous avons introduit notre propre mesure de divergence, qui dépend de l'hypothèse elle-même et est calculée entre les deux domaines. Cette mesure nous permet d'éviter les cas dégénérés, lorsque les deux distributions sont trop éloignées, mais également de gérer les tâches d'apprentissage semi-supervisé. Les expérimentations que nous avons menées, à la fois sur une base synthétique et une base réelle, justifient l'intérêt de cette approche, aussi bien dans le cadre de l'adaptation de domaine que dans celui de l'apprentissage semi-supervisé.

La deuxième contribution que nous avons proposée se destine plus particulièrement à traiter les données structurées. Nous avons fait appel à une approche d'auto-étiquetage, inspirée de l'algorithme DASVM, qui se base sur la théorie des SVMs. DASVM possède un certain nombre de

limitations dont nous avons voulu nous affranchir. Au-delà du coût de calcul important, dû au fait que le séparateur doit être appris à chaque itération, la fonction de similarité doit être semi-définie positive. Pour éviter ces restrictions, nous avons introduit une nouvelle approche, basée sur la récente théorie des fonctions de similarité $(\varepsilon, \gamma, \tau)$-bonnes. Ces dernières sont une généralisation de la théorie des SVMs, s'affranchissant des contraintes de SDP. Nous avons tout d'abord proposé une étude sur les conditions nécessaires à la réussite d'un algorithme d'auto-étiquetage. Ces conditions nous ont guidés dans la conception de notre algorithme, GESIDA. Celui-ci utilise des fonctions de similarité $(\varepsilon, \gamma, \tau)$-bonnes calculées entre données structurées, pour construire des séparateurs. À chaque itération, un certain nombre d'exemples de l'ensemble d'apprentissage sont ensuite remplacés par des exemples semi-étiquetés pour l'inférence d'un nouveau classifieur et ainsi de suite. Nous avons également introduit une méthode permettant d'éviter l'insertion d'un trop grand nombre d'*outliers* parmi les exemples semi-étiquetés. Une large série d'expérimentations a été menée sur des chiffres manuscrits représentés sous forme de chaînes de caractères et d'arbres dans le contexte de problèmes de changement d'échelle et de rotation. Cette étude expérimentale a permis de montrer l'intérêt de notre approche, à la fois sur le plan du taux de bonne classification et de la parcimonie des modèles induits. Les résultats montrent aussi la pertinence de notre stratégie de sélection des points semi-étiquetés, par rapport à une sélection aléatoire. Nous avons également mené une autre série d'expérimentations pour étudier les points raisonnables sélectionnés par l'algorithme au long des itérations, notamment pour observer l'évolution de la proportion de points raisonnables issus de la source en comparaison avec la proportion de ceux issus de la cible. Enfin, une dernière étude expérimentale nous a conforté dans le bien-fondé de notre étude théorique sur les conditions nécessaires à respecter pour éviter la divergence d'un algorithme d'auto-étiquetage. En effet, nous avons étudié deux problèmes différents en utilisant une sélection aléatoire des exemples. Il s'est avéré que le problème dans lequel les conditions étaient respectées à chaque itération aboutissait à une adaptation de domaine effective, contrairement à l'autre problème, dans lequel nous avons observé un phénomène de divergence.

À l'issue de ce travail de thèse, deux perspectives principales émergent assez naturellement des résultats obtenus. La première se positionne au niveau théorique et vise à démontrer les garanties de convergence de l'erreur en généralisation sur la cible. Ce problème reste ouvert dans la majorité des approches d'AD. Selon les travaux théoriques menés dans ce domaine, il faut, pour y parvenir, obtenir un modèle commettant peu d'erreurs sur le domaine source et que, dans le même temps, la divergence calculée entre les distributions source et cible soit faible. Cependant, un terme incompressible, représentant la différence entre les deux domaines (le paramètre $\lambda$ de la borne de Ben-David et al.), existe et c'est de celui-ci que provient toute la difficulté de réussir une bonne adaptation. L'objectif est donc de pouvoir estimer correctement ce paramètre afin d'avoir une idée des risques de transfert négatif. Quelques approches existent déjà, comme la validation circulaire proposée dans [Bruzzone and Marconcini, 2010], mais elle reste très coûteuse et assez peu fondée d'un point de vue théorique.

Une solution au problème précédent serait de relâcher la contrainte d'AD non supervisée, en autorisant l'accès à un petit nombre d'exemples étiquetés issus de la distribution cible (ce cadre est celui de l'adaptation de domaine supervisée). Ce contexte se rapproche de l'apprentissage semi-supervisé à la différence que les points étiquetés dont nous disposons sont issus de deux distributions différentes. La dérivation de garanties en généralisation serait plus simple dans ce cas, étant donné que le terme de divergence entre les deux domaines pourrait être estimé de manière plus juste, des exemples cibles étiquetés étant disponibles.

L'extension de ce travail à d'autres domaines de l'AD constitue une autre perspective naturelle. La régression en adaptation de domaine, par exemple, n'a été étudiée que dans quelques travaux [Cortes and Mohri, 2011, Cortes and Mohri, 2014]. Une notion de temporalité peut également être prise en compte et les méthodes itératives paraissent particulièrement adaptées à ce genre de problématique. Plusieurs travaux [Barve and Long, 1997, Mohri and Medina, 2012] peuvent être assimilés à cette notion, se plaçant dans le cadre des *Drifting Distributions* (ou *Concept Drift*), où l'objectif est de sélectionner une hypothèse $h$, dans un ensemble $\mathcal{H}$, minimisant la perte sur la distribution $\mathcal{D}_{T+1}$, en disposant d'exemples distribués selon les $T$ premières distributions jointes.

D'autres perspectives moins directes semblent être des sujets d'étude prometteurs. Des travaux en lien avec la recherche de représentation ont vu le jour, notamment [Gong et al., 2013], où les auteurs, dans le cadre de l'AD, cherchent les exemples issus de la source étant distribués de la façon la plus similaire aux exemples cibles.

Le domaine du *Life-long Learning* est également lié à l'adaptation de domaine. Il s'agit d'utiliser des connaissances précédemment acquises, potentiellement longtemps auparavant, pour apprendre de nouveaux modèles, à l'image de l'apprentissage chez l'humain tout au long d'une vie. La difficulté réside dans le stockage de l'information à chaque nouvelle tâche, ainsi que dans la réutilisation de ces connaissances de manière appropriée pour une tâche donnée.

Enfin, l'estimation de paramètres et la sélection de modèles sont des problèmes récurrents. Plusieurs approches, comme par exemple la validation inverse présentée dans [Zhong et al., 2010], ont été proposées, mais des recherches plus approfondies méritent d'être encore menées.

# Liste des publications

## Journaux internationaux

Habrard A., Peyrache J-P., Sebban M.

Iterative Self-Labeling Domain Adaptation for Linear Structured Image Classification

International Journal on Artificial Intelligence Tools (IJAIT), Volume N° 22, Issue N° 5, **2013**

## Conférences internationales

Habrard A., Peyrache J-P., Sebban M.

Boosting for Unsupervised Domain Adaptation

ECML/PKDD 2013, Proceedings part II, 433-448, **2013**

Habrard A., Peyrache J-P., Sebban M.

Habrard A., Peyrache J-P., Sebban M.

Domain Adaptation with Good Edit Similarities : a Sparse Way to deal with Rotation and Scaling Problems in Image Classification

Proceedings of the 23rd IEEE International Conference on Tools with Articial Intelligence (ICTAI 2011), 181-188, **2011**

**Prix du meilleur papier**

*L'article suivant, bien qu'ayant été publié durant la thèse, n'entre pas directement dans le cadre de celle-ci. Il est l'aboutissement d'un travail dans le cadre d'un stage recherche de Master 2.*

Bernard M., Jeudy B., Peyrache J-P., Sebban M., Thollard F.

Using the $\mathcal{H}$-divergence to Prune Probabilistic Automata

Proceedings of the 23rd IEEE International Conference on Tools with Articial Intelligence (ICTAI 2011), 725-731, **2011**

## Conférences nationales

Habrard A., Peyrache J-P., Sebban M.

Un Cadre Formel de Boosting pour l'Adaptation de Domaine

14e Conférence francophone sur l'Apprentissage automatique (CAp' 2012), 1-16, **2012**

**Prix du meilleur papier**

# Bibliographie


[Arnold et al., 2007] Arnold, A., Nallapati, R., and Cohen, W. W. (2007). A comparative study of methods for transductive transfer learning. In International Conference on Data Mining (ICDM) Workshops, pages 77–82.

[Balcan and Blum, 2006] Balcan, M.-F. and Blum, A. (2006). On a Theory of Learning with Similarity Functions. In International Conference on Machine Learning (ICML), pages 73–80.

[Balcan et al., 2008] Balcan, M.-F., Blum, A., and Srebro, N. (2008). Improved guarantees for learning via similarity functions. In Conference on Learning Theory (COLT), pages 287–298.

[Bartlett and Mendelson, 2002] Bartlett, P. L. and Mendelson, S. (2002). Rademacher and gaussian complexities : Risk bounds and structural results. Journal of Machine Learning Research, 3 :463–482.

[Barve and Long, 1997] Barve, R. D. and Long, P. M. (1997). On the complexity of learning from drifting distributions. Information and Computation, 138(2) :170–193.

[Becker et al., 2013] Becker, C. J., Christoudias, C. M., and Fua, P. (2013). Non-Linear Domain Adaptation with Boosting. In Neural Information Processing Systems (NIPS).

[Bellet, 2012] Bellet, A. (2012). Supervised Metric Learning with Generalization Guarantees. PhD thesis, University of Saint-Etienne.

[Ben-David et al., 2010] Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. (2010). A theory of learning from different domains. Machine Learning, 79(1-2) :151–175.

[Ben-David et al., 2006] Ben-David, S., Blitzer, J., Crammer, K., and Pereira, F. (2006). Analysis of representations for domain adaptation. In Neural Information Processing Systems (NIPS), pages 137–144.

[Ben-David et al., 2012] Ben-David, S., Loker, D., Srebro, N., and Sridharan, K. (2012). Minimizing the misclassification error rate using a surrogate convex loss. In International Conference on Machine Learning (ICML).

[Bennett et al., 2002] Bennett, K. P., Demiriz, A., and for Computational Linguistics (ACL)in, R. M. (2002). Exploiting unlabeled data in ensemble methods. In Conference on Knowledge Discovery and Data Mining (KDD), pages 289–296.

[Bernard et al., 2008] Bernard, M., Boyer, L., Habrard, A., and Sebban, M. (2008). Learning probabilistic models of tree edit distance. Pattern Recognition, 41(8) :2611–2629.

[Bickel et al., 2007] Bickel, S., Brückner, M., and Scheffer, T. (2007). Discriminative learning for differing training and test distributions. In International Conference on Machine Learning (ICML), pages 81–88, New York, NY, USA. ACM.

[Bille, 2005] Bille, P. (2005). A survey on tree edit distance and related problem. Theoretical Computer Science, 337(1-3) :217–239.

[Bishop, 2007] Bishop, C. M. (2007). Pattern Recognition and Machine Learning.

[Blitzer et al., 2007] Blitzer, J., Dredze, M., and Pereira, F. (2007). Biographies, bollywood, boom-boxes and blenders : Domain adaptation for sentiment classification. In Association for Computational Linguistics (ACL).

[Blitzer et al., 2011] Blitzer, J., Kakade, S., and Foster, D. P. (2011). Domain adaptation with coupled subspaces. In International Conference in Artificial Intelligence and Statistics (AISTATS), pages 173–181.

[Blitzer et al., 2006] Blitzer, J., McDonald, R., and Pereira, F. (2006). Domain adaptation with structural correspondence learning. In Conference on Empirical Methods in Natural Language Processings (EMNLP), pages 120–128.

[Blum and Mitchell, 1998] Blum, A. and Mitchell, T. M. (1998). Combining labeled and unlabeled sata with co-training. In Conference on Learning Theory (COLT), pages 92–100.

[Bordes et al., 2007] Bordes, A., Bottou, L., Gallinari, P., and Weston, J. (2007). Solving multiclass support vector machines with larank. In International Conference on Machine Learning (ICML), pages 89–96.

[Boser et al., 1992] Boser, B. E., Guyon, I., and Vapnik, V. (1992). A training algorithm for optimal margin classifiers. In Conference on Learning Theory (COLT), pages 144–152.

[Bousquet and Elisseeff, 2000] Bousquet, O. and Elisseeff, A. (2000). Algorithmic stability and generalization performance. In Neural Information Processing Systems (NIPS), pages 196–202.

[Bousquet and Elisseeff, 2002] Bousquet, O. and Elisseeff, A. (2002). Stability and generalization. Journal of Machine Learning Research, 2 :499–526.

[Bruzzone and Marconcini, 2010] Bruzzone, L. and Marconcini, M. (2010). Domain adaptation problems : A dasvm classification technique and a circular validation strategy. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(5) :770–787.

[Chen et al., 2011a] Chen, M., Weinberger, K. Q., and Blitzer, J. (2011a). Co-training for domain adaptation. In Neural Information Processing Systems (NIPS), pages 2456–2464.

[Chen et al., 2011b] Chen, M., Weinberger, K. Q., and Chen, Y. (2011b). Automatic feature decomposition for single view co-training. In International Conference on Machine Learning (ICML), pages 953–960.

[Cornuéjols and Miclet, 2010] Cornuéjols, A. and Miclet, L. (2010). Apprentissage artificiel : concepts et algorithmes.

[Cortes et al., 2004] Cortes, C., Haffner, P., and Mohri, M. (2004). Rational kernels : Theory and algorithms. Journal of Machine Learning Research, 5 :1035–1062.

[Cortes and Mohri, 2011] Cortes, C. and Mohri, M. (2011). Domain adaptation in regression. In International Conference on Algorithmic Learning Theory (ALT), pages 308–323.

[Cortes and Mohri, 2014] Cortes, C. and Mohri, M. (2014). Domain adaptation and sample bias correction theory and algorithm for regression. Theoretical Computer Science, 519 :103–126.

[Cortes et al., 2008] Cortes, C., Mohri, M., Riley, M., and Rostamizadeh, A. (2008). Sample selection bias correction theory. Computing Research Repository (CoRR), abs/0805.2775.

[Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3) :273–297.

[Dai et al., 2007] Dai, W., Yang, Q., Xue, G.-R., and Yu, Y. (2007). Boosting for transfer learning. In International Conference on Machine Learning (ICML), pages 193–200.

[d'Alché Buc et al., 2001] d'Alché Buc, F., Grandvalet, Y., and Ambroise, C. (2001). Semi-supervised marginboost. In Neural Information Processing Systems (NIPS), pages 553–560.

[Daliri and Torre, 2008] Daliri, M. R. and Torre, V. (2008). Robust symbolic representation for shape recognition and retrieval. Pattern Recognition, 41(5) :1799–1815.

[Daumé III, 2007] Daumé III, H. (2007). Frustratingly easy domain adaptation. In Association for Computational Linguistics (ACL).

[Daumé III et al., 2010] Daumé III, H., Kumar, A., and Saha, A. (2010). Co-regularization based semi-supervised domain adaptation. In Neural Information Processing Systems (NIPS), pages 478–486.

[Duan et al., 2012] Duan, L., Tsang, I. W., and Xu, D. (2012). Domain transfer multiple kernel learning. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(3).

[Duan et al., 2009] Duan, L., Tsang, I. W.-H., Xu, D., and Maybank, S. J. (2009). Domain transfer svm for video concept detection. In Computer Vision and Pattern Recognition Conference (CVPR), pages 1375–1381.

[Dudík et al., 2005] Dudík, M., Schapire, R. E., and Phillips, S. J. (2005). Correcting sample selection bias in maximum entropy density estimation. In Neural Information Processing Systems (NIPS).

[Feldman et al., 2009] Feldman, V., Guruswami, V., Raghavendra, P., and Wu, Y. (2009). Agnostic learning of monomials by halfspaces is hard. In Annual Symposium on Foundations of Computer Science (FOCS), pages 385–394.

[Florian et al., 2004] Florian, R., Hassan, H., Ittycheriah, A., Jing, H., Kambhatla, N., Luo, X., Nicolov, N., and Roukos, S. (2004). A statistical model for multilingual entity detection and tracking. In Conference of the Association for Computational Linguistics and Human Language Technology, pages 1–8.

[Freeman, 1974] Freeman, H. (1974). Computer processing of line-drawing images. Association for Computing Machinery (ACM) Computing Surveys, 6(1) :57–97.

[Freund, 2000] Freund, Y. (2000). An adaptive version of the boost by majority algorithm. In Conference on Learning Theory (COLT), pages 102–113.

[Freund and Schapire, 1996] Freund, Y. and Schapire, R. E. (1996). Experiments with a new boosting algorithm. In International Conference on Machine Learning (ICML), pages 148–156.

[Friedman et al., 1998] Friedman, J., Hastie, T., and Tibshirani, R. (1998). Additive logistic regression : a statistical view of boosting. Annals of Statistics, 28 :2000.

[Germain et al., 2013] Germain, P., Habrard, A., Laviolette, F., and Morvant, E. (2013). A PAC-Bayesian Approach for Domain Adaptation with Specialization to Linear Classifiers. In International Conference on Machine Learning (ICML), pages 738–746.

[Gong et al., 2013] Gong, B., Grauman, K., and Sha, F. (2013). Connecting the dots with landmarks : Discriminatively learning domain-invariant features for unsupervised domain adaptation. In International Conference on Machine Learning (ICML), pages 222–230.

[Gretton et al., 2006] Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. J. (2006). A kernel method for the two-sample-problem. In Neural Information Processing Systems (NIPS), pages 513–520.

[Harel and Mannor, 2012] Harel, M. and Mannor, S. (2012). The perturbed variation. In Neural Information Processing Systems (NIPS), pages 1943–1951.

[Hsieh and Hsu, 2008] Hsieh, S.-M. and Hsu, C.-C. (2008). Retrieval of images by spatial and object similarities. Information Processing and Management, 44(3) :1214–1233.

[Huang et al., 2006] Huang, J., Smola, A. J., Gretton, A., Borgwardt, K. M., and Schölkopf, B. (2006). Correcting sample selection bias by unlabeled data. In Neural Information Processing Systems (NIPS), pages 601–608.

[Janodet et al., 2004] Janodet, J.-C., Nock, R., Sebban, M., and Suchier, H.-M. (2004). Boosting grammatical inference with confidence orassociation for computational linguistics (acl)es. In International Conference on Machine Learning (ICML).

[Ji et al., 2011] Ji, Y., Chen, J., Niu, G., Shang, L., and Dai, X. (2011). Transfer learning via multi-view principal component analysis. Journal of Computer Science and Technology, 26(1) :81–98.

[Jiang, 2008] Jiang, J. (2008). A Literature Survey on Domain Adaptation of Statistical Classifiers.

[Jiang and Zhai, 2007a] Jiang, J. and Zhai, C. (2007a). Instance weighting for domain adaptation in nlp. In Association for Computational Linguistics (ACL).

[Jiang and Zhai, 2007b] Jiang, J. and Zhai, C. (2007b). A two-stage approach to domain adaptation for statistical classifiers. In ACM International Conference on Information and Knowledge Management (CIKM), pages 401–410.

[Joachims, 1999] Joachims, T. (1999). Transductive inference for text classification using support vector machines. In International Conference on Machine Learning (ICML), pages 200–209.

[Koltchinskii, 2001] Koltchinskii, V. (2001). Rademacher penalties and structural risk minimization. IEEE Transactions on Information Theory, 47(5) :1902–1914.

[Kulis et al., 2011] Kulis, B., Saenko, K., and Darrell, T. (2011). What you saw is not what you get : Domain adaptation using asymmetric kernel transforms. In Computer Vision and Pattern Recognition Conference (CVPR), pages 1785–1792.

[Kullback and Leibler, 1951] Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. Annals of Mathematical Statistics, 22(1) :79–86.

[Ladicky and Torr, 2011] Ladicky, L. and Torr, P. H. S. (2011). Locally linear support vector machines. In International Conference on Machine Learning (ICML), pages 985–992.

[Lafferty et al., 2001] Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional random fields : Probabilistic models for segmenting and labeling sequence data. In International Conference on Machine Learning (ICML), pages 282–289.

[Li, 2006] Li, L. (2006). Multiclass boosting with repartitioning. In International Conference on Machine Learning (ICML), pages 569–576.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision (IJCV), 60(2) :91–110.

[Mallapragada et al., 2009] Mallapragada, P. K., Jin, R., Jain, A. K., and Liu, Y. (2009). Semiboost : Boosting for semi-supervised learning. IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(11) :2000–2014.

[Mansour et al., 2009] Mansour, Y., Mohri, M., and Rostamizadeh, A. (2009). Domain adaptation : Learning bounds and algorithms. In Conference on Learning Theory (COLT).

[Mansour and Schain, 2012] Mansour, Y. and Schain, M. (2012). Robust domain adaptation. In International Symposium on Artificial Intelligence and Mathematics (ISAIM).

[Margolis, 2011] Margolis, A. (2011). A literature review of domain adaptation with unlabeled data. Technical Report, pages 1–42.

[Martinez, 2002] Martinez, A. M. (2002). Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24 :748–763.

[Mohri and Medina, 2012] Mohri, M. and Medina, A. M. (2012). New analysis and algorithm for learning with drifting distributions. In International Conference on Algorithmic Learning Theory (ALT), pages 124–138.

[Mohri et al., 2012] Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2012). Foundations of Machine Learning. The MIT Press.

[Morvant et al., 2011] Morvant, E., Habrard, A., and Ayache, S. (2011). Sparse Domain Adaptation in Projection Spaces based on Good Similarity Functions. In International Conference on Data Mining (ICDM), pages 457–466.

[Morvant et al., 2012] Morvant, E., Habrard, A., and Ayache, S. (2012). Parsimonious Unsupervised and Semi-Supervised Domain Adaptation with Good Similarity Functions. Knowledge and Information Systems (KAIS), 33(2) :309–349.

[Mukherjee and Schapire, 2010] Mukherjee, I. and Schapire, R. E. (2010). A theory of multiclass boosting. In Neural Information Processing Systems (NIPS), pages 1714–1722.

[Pan et al., 2008] Pan, S. J., Kwok, J. T., and Yang, Q. (2008). Transfer learning via dimensionality reduction. In AAAI Conference on Artificial Intelligence, pages 677–682.

[Pan and Yang, 2010] Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 22(10) :1345–1359.

[Pardoe and Stone, 2010] Pardoe, D. and Stone, P. (2010). Boosting for regression transfer. In International Conference on Machine Learning (ICML), pages 863–870.

[Pérez and Sánchez-Montañés, 2007] Pérez, Ó. and Sánchez-Montañés, M. A. (2007). A new learning strategy for classification problems with different training and test distributions. In International Work-Conference on Artificial Neural Networks (IWANN), pages 178–185.

[Punitha and Guru, 2008] Punitha, P. and Guru, D. S. (2008). Symbolic image indexing and retrieval by spatial similarity : An approach based on b-tree. Pattern Recognition, 41(6) :2068–2085.

[Ren et al., 2008] Ren, J., Shi, X., Fan, W., and Yu, P. S. (2008). Type-independent correction of sample selection bias via structural discovery and re-balancing. In SIAM Inernational Conference on Data Mining (SDM), pages 565–576. SIAM.

[Roark and Bacchiani, 2003] Roark, B. and Bacchiani, M. (2003). Supervised and unsupervised pcfg adaptation to novel domains. In Association for Computational Linguistics (ACL), pages 126–133.

[Rosasco et al., 2004] Rosasco, L., Vito, E. D., Caponnetto, A., Piana, M., and Verri, A. (2004). Are loss functions all the same ? Neural Computation, 16(5) :1063–107.

[Rosenfeld, 1996] Rosenfeld, R. (1996). A maximum entropy approach to adaptive statistical language modeling. Computer Speech and Language, 10 :187–228.

[Roy et al., 2012] Roy, S. D., Mei, T., Zeng, W., and Li, S. (2012). Socialtransfer : cross-domain transfer learning from social streams for media applications. In Association for Computing Machinery (ACM) Multimedia, pages 649–658.

[Saberian and Vasconcelos, 2011] Saberian, M. J. and Vasconcelos, N. (2011). Multiclass boosting : Theory and algorithms. In Neural Information Processing Systems (NIPS), pages 2124–2132.

[Salton et al., 1975] Salton, G., Wong, A., and Yang, C. S. (1975). A vector space model for automatic indexing. Communications of the Association for Computing Machinery (ACM), 18(11) :613–620.

[Satpal and Sarawagi, 2007] Satpal, S. and Sarawagi, S. (2007). Domain adaptation of conditional probability models via feature subsetting. In Principles and Practice of Knowledge Discovery in Databases (PKDD), pages 224–235.

[Schapire et al., 1997] Schapire, R., Freund, Y., Barlett, P., and Lee, W. (1997). Boosting the margin : A new explanation for the effectiveness of voting methods. In International Conference on Machine Learning (ICML), pages 322–330.

[Schapire, 1989] Schapire, R. E. (1989). The strength of weak learnability (extended abstract). In Annual Symposium on Foundations of Computer Science (FOCS), pages 28–33.

[Sebban et al., 2001] Sebban, M., Nock, R., and Lallich, S. (2001). Boosting neighborhood-based classifiers. In International Conference on Machine Learning (ICML), pages 505–512.

[Sebban et al., 2002] Sebban, M., Nock, R., and Lallich, S. (2002). Stopping criterion for boosting-based data reduction techniques : from binary to multiclass problem. Journal of Machine Learning Research, 3 :863–885.

[Selkow, 1977] Selkow, S. (1977). The tree-to-tree editing problem. Information Processing Letters, 6 :184–186.

[Sugiyama et al., 2007] Sugiyama, M., Nakajima, S., Kashima, H., von Bünau, P., and Kawanabe, M. (2007). Direct importance estimation with model selection and its application to covariate shift adaptation. In Neural Information Processing Systems (NIPS).

[Torralba and Efros, 2011] Torralba, A. and Efros, A. A. (2011). Unbiased look at dataset bias. In Computer Vision and Pattern Recognition Conference (CVPR), pages 1521–1528.

[Tsuboi et al., 2008] Tsuboi, Y., Kashima, H., Hido, S., Bickel, S., and Sugiyama, M. (2008). Direct density ratio estimation for large-scale covariate shift adaptation. In SIAM International Conference on Data Mining (SDM), pages 443–454.

[Valiant, 1984] Valiant, L. G. (1984). A theory of the learnable. Communications of the Association for Computing Machinery (ACM), 27(11) :1134–1142.

[Vapnik and Chervonenkis, 1971] Vapnik, V. N. and Chervonenkis, A. Y. (1971). On the uniform convergence of relative frequencies of events to their probabilities. Theory of Probability and its Applications, XVI(2) :264–280.

[Wagner and Fischer, 1974] Wagner, R. A. and Fischer, M. J. (1974). The string-to-string correction problem. Journal of the Association for Computing Machinery (ACM), 21(1) :168–173.

[Warmuth et al., 2007] Warmuth, M. K., Glocer, K. A., and Rätsch, G. (2007). Boosting algorithms for maximizing the soft margin. In Neural Information Processing Systems (NIPS).

[Xu et al., 2009] Xu, H., Caramanis, C., and Mannor, S. (2009). Robustness and regularization of support vector machines. Journal of Machine Learning Research, 10 :1485–1510.

[Xu and Mannor, 2010] Xu, H. and Mannor, S. (2010). Robustness and generalization. In Conference on Learning Theory (COLT), pages 503–515.

[Xu and Mannor, 2012] Xu, H. and Mannor, S. (2012). Robustness and generalization. Machine Learning, 86(3) :391–423.

[Yao and Doretto, 2010] Yao, Y. and Doretto, G. (2010). Boosting for transfer learning with multiple sources. In Computer Vision and Pattern Recognition Conference (CVPR), pages 1855–1862.

[Zadrozny, 2004] Zadrozny, B. (2004). Learning and evaluating classifiers under sample selection bias. In International Conference on Machine Learning (ICML).

[Zhong et al., 2010] Zhong, E., Fan, W., Yang, Q., Verscheure, O., and Ren, J. (2010). Cross validation framework to choose amongst models and datasets for transfer learning. In European Conference on Machine Learning (ECML), pages 547–562.

[Zhu et al., 2003] Zhu, J., Rosset, S., Hastie, T., and Tibshirani, R. (2003). 1-norm support vector machines. In Neural Information Processing Systems (NIPS).

# Annexe A

## Distance de Selkow

La distance de Selkow, représentant la distance d'édition entre deux arbres, est calculée à l'aide de l'Algorithme 6.

Trois tableaux sont définis de la manière suivante au début de l'algorithme :

- lab$(s_i, s_j)$, qui pour une paire d'étiquettes $(s_i, s_j)$ contient le coût de l'opération de substitution de $s_i$ en $s_j$,
- del$(a_k)$, qui contient le coût de l'opération de suppression de l'arbre $a_k$.
- ins$(a_k)$, qui contient le coût de l'opération d'insertion de l'arbre $a_k$,

Les opérations de substitution, de suppression et de substitution d'un noeud sont illustrées respectivement par les Figures A.1, A.2, A.3.

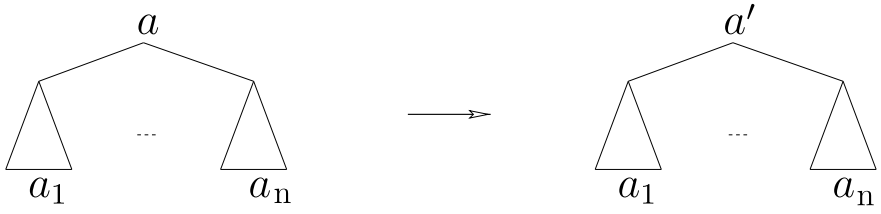



FIGURE A.1 – Substitution du noeud $a$ par le noeud $a'$ dans l'arbre $(a, (a_1, \ldots, a_n))$.

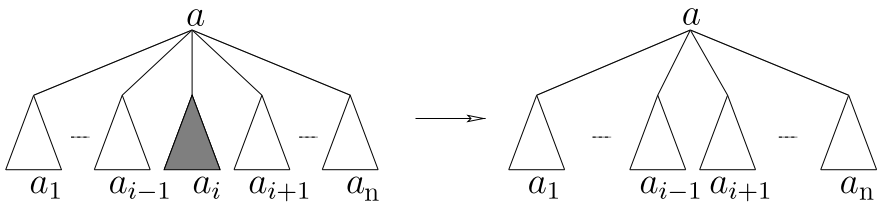



FIGURE A.2 – Suppression du noeud $a_i$ dans l'arbre $(a, (a_1, \ldots, a_n))$.

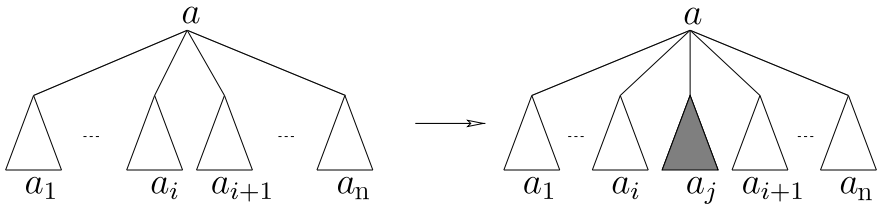



FIGURE A.3 – Insertion du noeud $a_j$ dans l'arbre $(a, (a_1, \ldots, a_n))$.

Selkow$(A, B)$

**Entrée :**

- un arbre $A = (a, (a_1, \ldots, a_m))$,
- un arbre $B = (b, (b_1 \ldots, b_n))$,
- les tableaux lab, ins et del.

**Sortie :** La distance de Selkow entre $A$ et $B$.
Soit une matrice $\delta(0 : m, 0 : n)$
$\delta(0, 0) = \text{lab}(a, b)$
**pour** $k = 1$ **à** $n$ **faire**
| $\delta(0, k) = \delta(0, k-1) + \text{ins}(b_k)$
**fin**
**pour** $k = 1$ **à** $m$ **faire**
| $\delta(k, 0) = \delta(k-1, 0) + \text{del}(a_k)$
**fin**
**pour** $i = 1$ **à** $m$ **faire**
| **pour** $j = 1$ **à** $n$ **faire**
| | $\delta(i, j) = \min(\delta(i-1, j-1) + \text{Selkow}(a_i, b_j), \delta(i, j-1) + \text{ins}(b_j), \delta(i-1, j) + \text{del}(a_i))$
| **fin**
**fin**
Renvoyer $\delta(m, n)$.

**Algorithme 6** : ALGORITHME DE CALCUL DE LA DISTANCE DE SELKOW

**Résumé**

Ces dernières années, l'intérêt pour l'apprentissage automatique n'a cessé d'augmenter dans des domaines aussi variés que la reconnaissance d'images ou l'analyse de données médicales. Cependant, une limitation du cadre classique PAC a récemment été mise en avant. Elle a entraîné l'émergence d'un nouvel axe de recherche : l'Adaptation de Domaine, dans lequel on considère que les données d'apprentissage proviennent d'une distribution (dite source) différente de celle (dite cible) dont sont issues les données de test. Les premiers travaux théoriques effectués ont débouché sur la conclusion selon laquelle une bonne performance sur le test peut s'obtenir en minimisant à la fois l'erreur sur le domaine source et un terme de divergence entre les deux distributions. Trois grandes catégories d'approches s'en inspirent : par repondération, par reprojection et par auto-étiquetage. Dans ce travail de thèse, nous proposons deux contributions. La première est une approche de reprojection basée sur la théorie du boosting et s'appliquant aux données numériques. Celle-ci offre des garanties théoriques intéressantes et semble également en mesure d'obtenir de bonnes performances en généralisation. Notre seconde contribution consiste d'une part en la proposition d'un cadre permettant de combler le manque de résultats théoriques pour les méthodes d'auto-étiquetage en donnant des conditions nécessaires à la réussite de ce type d'algorithme. D'autre part, nous proposons dans ce cadre une nouvelle approche utilisant la théorie des (epsilon,gamma,tau)-bonnes fonctions de similarité afin de contourner les limitations imposées par la théorie des noyaux dans le contexte des données structurées.

**Abstract**

During the past few years, an increasing interest for Machine Learning has been encountered, in various domains like image recognition or medical data analysis. However, a limitation of the classical PAC framework has recently been highlighted. It led to the emergence of a new research axis : Domain Adaptation (DA), in which learning data are considered as coming from a distribution (the source one) different from the one (the target one) from which are generated test data. The first theoretical works concluded that a good performance on the target domain can be obtained by minimizing in the same time the source error and a divergence term between the two distributions. Three main categories of approaches are derived from this idea : by reweighting, by reprojection and by self-labeling. In this thesis work, we propose two contributions. The first one is a reprojection approach based on boosting theory and designed for numerical data. It offers interesting theoretical guarantees and also seems able to obtain good generalization performances. Our second contribution consists first in a framework filling the gap of the lack of theoretical results for self-labeling methods by introducing necessary conditions ensuring the good behavior of this kind of algorithm. On the other hand, we propose in this framework a new approach, using the theory of (epsilon,gamma,tau)-good similarity functions to go around the limitations due to the use of kernel theory in the specific context of structured data.