Apprentissage de sphères maximales d'exclusion avec garanties théoriques

Guillaume Metzler[1], Xavier Badiche[2], Brahim Belkasmi[3], Stéphane Canu[4], Elisa Fromont[5], Amaury Habrard[6], et Marc Sebban[7]

[1,5,6,7]Univ. Lyon, Univ. St-Etienne F-42000, UMR CNRS 5516, Laboratoire Hubert-Curien
[2,3]BLITZ BUSINESS SCEB, 38090 Villefontaine
[4]LITIS EA 4108, Univ. Rouen 76800 St-Etienne du Rouvray

19 avril 2018

Résumé

Dans ce papier, nous proposons une méthode d'apprentissage automatique avec garanties théoriques pour générer des sphères maximales d'exclusion sur des données binaires présentant un fort déséquilibre. Notre objectif est d'apprendre un ensemble de sphères locales, centrées sur les exemples de la classe minoritaire, qui excluent les exemples de la classe majoritaire. Notre contribution est double : 1) le problème est abordé comme un problème d'apprentissage de métrique et 2) nous démontrons des résultats de stabilité uniforme sur le rayon et la métrique apprise par notre algorithme. Nos expériences sur des jeux de données réelles montrent l'intérêt de notre approche.

Mots-clef : Données déséquilibrées, Support Vector Data Description (SVDD), Pb de cercles minimum.

1 Introduction

The study of unbalanced data is an active and important supervised machine learning domain due to its huge economical impact, for example in anomaly or fraud detection in applications related to banks, medicine or industrial processes [AA15, KCP11].

Unbalanced problems are challenging for classic supervised machine learning methods which aim at minimizing an error-based loss function. If one class is rare, the algorithm will struggle to capture any useful information about this class and will obtain a high accuracy by simply predicting all (possibly new) examples as being of the majority class(es).

To tackle the issue, classic methods consist in over or under-sampling the data in order to get more balanced representations of the classes [Agg13]. Then, combination of classifiers (e.g. random forests [KCP11]) that can perform another sampling step or can focus on local subspaces of the data where the minority classes are more represented, are often preferred to standard machine learning techniques. For example, in computer vision, Viola and Jones [VJ01] introduced a cascade learning approach to achieve, through a boosting process, high detection and low false positive rate in an object detection context. A variant of the Support Vector Machines called one-class SVM (OCSVM) [HSKS03] has also been developed for two class problems where the minority class examples are so scarce that they are not taken into account in the learning process. OCSVM is an example of a more general approach called Support Vector Data Description (SVDD) [ALC14]. In general, SVDD methods learn one ball which includes all the training data and excludes all examples lying in the tail of the distribution, which are considered as anomalies. To capture non linearity, SVDD can make use of the kernel trick that can be computationally expensive to store and compute on large datasets.

In this paper, we aim at benefiting from the SVDD setting while taking into account the examples of the minority class. To do so, we learn local models centered at each minority class example which exclude the examples of the majority class(es). This allows us to consider settings where the minority class examples do not necessarily behave as anomalies but may be hidden within one of the mode of the data (e.g fraud data). Similarly to SVDD, we also resort to a change of space by learning, for each local model, a linear projection using a Mahalanobis metric learning-based approach

[BHS13, Kul13]. The data are locally projected in smaller spaces which allow us to capture non linearity in a much cheaper way than SVDD. We show that these projections can be expressed in closed form which ensures (for free) the positive definiteness of the learned metrics. We derive theoretical guarantees both on the metric and on the parameters of the models showing the stability of our algorithm with respect to changes in the training set.

In Section 2, we introduce the classic Support Vector Data Description technique which will be the basis of our method. In Section 3, we present a primal and dual version of our optimization problem as well as a closed form solution of the linear transformation of the data. Section 4 gives a theoretical study of the algorithm. Our experimental results on unbalanced datasets are presented in Section 5. We conclude in Section 6.

2 Support Vector Data Description

The method presented in this article is inspired from the Support Vector Data Description (SVDD) [PA11, TD04] which consists in learning the smallest enclosing ball of the learning data. It can be used to detect anomalies by solving the following constraint optimization problem given a sample of n supposedly non abnormal instances :

$$\begin{array}{ll} \underset{R,\boldsymbol{c},\boldsymbol{\xi}}{min} & R^2 + \frac{\mu}{n}\sum_{i=1}^{n}\xi_i, \\ s.t. & \|\boldsymbol{x}_i - \boldsymbol{c}\|^2 \leq R^2 + \xi_i, \quad \forall i = 1,\dots,n, \\ & \xi_i \geq 0, \end{array} \tag{1}$$

where R and c are respectively the radius and the center of the ball and ξ_i is the slack variable associated to the i^{th} example. μ is tuned in order to control the proportion of data outside the sphere (considered as anomalies). Note that in [PA11], the authors have shown that using the radius instead of the square of the radius in this formulation is often preferable.

Several refinements of the SVDD method can be found. In [LZ06], the SVDD are used in binary classification problems. The authors want to find the minimum enclosing ball of a first class and the maximum excluding ball of the second one. This gives better results than solving the Minimum Enclosing Ball Problem presented in Eq. 1 and makes the parallel between SVDD and SVM where the width of the ring represents the margin of the SVM.

In [LTM13], the aim is to learn a set of hyper-spheres that can describe the entire distribution of the data. In this method, the author also apply a non linear transformation to the data (here, a Gaussian kernel) to improve the overall accuracy in their experiments. The idea of learning several hyper-spheres will be exploited in our formulation. A similar idea is used in [BJZ12] for either binary or multi-class classification problems. They combine the idea of Fuzzy theory with the kernel trick to propose a new decision function.

Even if the previous kernel-based methods are effective, the computation of the kernel is often expensive (according to the number of examples in the dataset) and does not scale well on most real dataset. An interesting approach, which does not suffer from this drawback, is presented in [WGP10]. The authors include a linear transformation of the data, in the form a PSD matrix M in the SVDD optimisation problem. To avoid high computational costs, they set M to the covariance matrix that allows the induction of ellipsoids rather than spheres. Such objects are able to cover a larger volume in the input space compared to the spheres.

In this paper, we exploit the idea of learning multiple local models to capture non linearity at a cheap cost as in [LTM13] and combine it with a metric learning formulation. Unlike [WGP10], we optimize both the shape and the orientation of the ellipsoid by learning a Mahalanobis distance based on a PSD matrix M. A nice property of our approach is that M can be obtained in closed-form solution ensuring directly the positive definiteness of M [SBS14, PH15]. Therefore, we prevent the algorithm from having to check the positiveness of the eigen values of M, which has a cubic complexity in the size of M, as required by many metric learning algorithms [BHS13]. All in all, our approach is **simple** - it learns local models (that can be done in parallel) by solving a simple optimization problem, **theoretically founded** - we derive generalization guarantees, and **efficient** compared to the state of the art.

3 Problem Formulation

Let $S = \{\boldsymbol{x}_i\}_{i=1}^n$ be a sample of n *negative* instances and $P = \{\boldsymbol{c}_j\}_{j=1}^p$ a set of p *positive* examples (our minority class) where each $\boldsymbol{x}_i, \boldsymbol{c}_j$ are feature vectors of $\mathbb{R}^d$. We aim at maximizing ellipsoids centered at each positive $\boldsymbol{c} \in P$ excluding the negative data $\boldsymbol{x}_i$, $i = 1, ..., n$. Learning such ellipsoids boils down to optimizing a Mahalanobis distance, that is finding a positive semi-definite (PSD) $d \times d$ matrix $\mathbf{M}$ projecting the data linearly in a new space and allowing to obtain balls centered at each positive example of maximum radius R. Note that the size of the projection space is equal to the rank of matrix M. Therefore, it can be much smaller that d if the features are strongly correlated. Let B be an upper bound of the possible expected radius, the primal formulation of the problem is defined as follows :

$$\begin{array}{cl} \underset{R,\boldsymbol{M},\boldsymbol{\xi}}{min} & \frac{1}{n}\sum_{i=1}^{n}\xi_i + \mu(B-R)^2 + \lambda\|\mathbf{M}-\mathbf{I}\|_F^2, \\ s.t. & \|\boldsymbol{x}_i - \boldsymbol{c}\|_{\mathbf{M}}^2 \geq R - \xi_i, \quad \forall i = 1,\ldots,n, \\ & \xi_i \geq 0, \\ & B \geq R \geq 0, \end{array} \tag{2}$$

where $\|\boldsymbol{x}_i - \boldsymbol{c}\|_{\mathbf{M}}^2$ is the learned Mahalanobis distance between a negative example x_i and a positive center c; $\boldsymbol{\xi}$ is the vector of the slack variables and $\mu(B-R)^2 + \lambda\|\mathbf{M}-\mathbf{I}\|_F^2$ is a regularization term with $\mu, \lambda > 0$ the corresponding regularization parameters. We choose two different parameters for each part of the regularization term to control the size of the sphere in the transformed space and the complexity of the matrix $\mathbf{M}$ independently. The parameter λ gives the possibility to control the entries of the learned matrix, and therefore the shape of the ellipsoid. In practice, the bigger λ, the closer $\|\boldsymbol{x}_i - \boldsymbol{c}\|_{\mathbf{M}}^2$ to the Euclidean distance (i.e. the ellipsoid looks like a ball). On the other hand, the parameter μ controls the size of the learned ellipsoids.

An illustration of our algorithm, called ME^2 for Maximum Excluding Ellipsoids, based on Problem 2, is given in Figures 1 and 2. We represent the behavior of the solution with and without learning the matrix $\mathbf{M}$. As we can see, by learning local Mahalanobis distances, we are able to cover a larger space which gives us the possibility to capture more examples from the rare class.

Note that the previous problem can also be expressed in its dual form which leads us to a closed form solution. We provide more details about our dual formulation in Appendix A and give here necessary elements to understand the formulation of our approach.

The Lagrangian of Problem 2 is given by :

$$\begin{aligned} \mathcal{L}(\boldsymbol{\alpha},\beta,\delta,\boldsymbol{\gamma},R,\boldsymbol{\xi},\mathbf{M}) = \frac{1}{n}\sum_{i=1}^{n}\xi_i + \mu(B-R)^2 \\ -\sum_{i=1}^{n}\gamma_i\xi_i - \sum_{i=1}^{n}\alpha_i\left(\|\boldsymbol{x}_i - \boldsymbol{c}\|_{\mathbf{M}}^2 - R + \xi_i\right) \\ + \lambda\|\mathbf{M}-\mathbf{I}\|_F^2 - \beta R + \delta(B-R), \end{aligned} \tag{3}$$

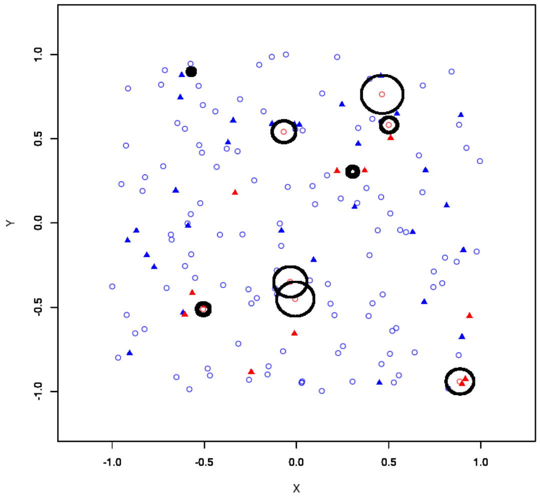

FIGURE 1 – Illustration of the behavior of our algorithm when λ is very small, that is, inducing a (almost not learned) matrix $\mathbf{M}$ close to the identity matrix. Each sphere is centered at a positive example (in red) and is learned using the negative ones (in blue).

where $\boldsymbol{\alpha} = (\alpha_i)_{i=1,\ldots,n}$, $\boldsymbol{\gamma} = (\gamma_i)_{i=1,\ldots,n}$, β and δ are the dual variables. We now write the derivatives of (3) with respect to the primal variables :

$$\nabla_R\mathcal{L} = \sum_{i=1}^{n}\alpha_i + 2\mu R - 2\mu B - \beta, \tag{4}$$

$$\nabla_{\xi_i}\mathcal{L} = \frac{1}{n} - \gamma_i - \alpha_i, \quad \forall i = 1,\ldots,n, \tag{5}$$

The derivative of the Frobenius norm is :

$$\frac{\partial\|\mathbf{M}-\mathbf{I}\|_F^2}{\partial\mathbf{M}} = 2(\mathbf{M}-\mathbf{I}).$$

This last equality implies :

$$\frac{\partial\mathcal{L}}{\partial\mathbf{M}} = -\sum_{k=1}^{n}\alpha_k\left[(\boldsymbol{x}_k - \boldsymbol{c})(\boldsymbol{x}_k^T - \boldsymbol{c}^T)\right] + 2\lambda(\mathbf{M}-\mathbf{I}). \tag{6}$$

Setting all the derivatives equal to zero, we get :

$$(4) \quad \Rightarrow \quad R = \frac{\beta - \delta + 2\mu B - \sum_{i=1}^{n}\alpha_i}{2\mu},$$

$$(5) \quad \Rightarrow \quad 0 \leq \alpha_i \leq \frac{1}{n},$$

$$(6) \quad \Rightarrow \quad \mathbf{M} = I + \frac{1}{2\lambda}\sum_{k=1}^{n}\alpha_k(x_k - c)(x_k - c)^T.$$

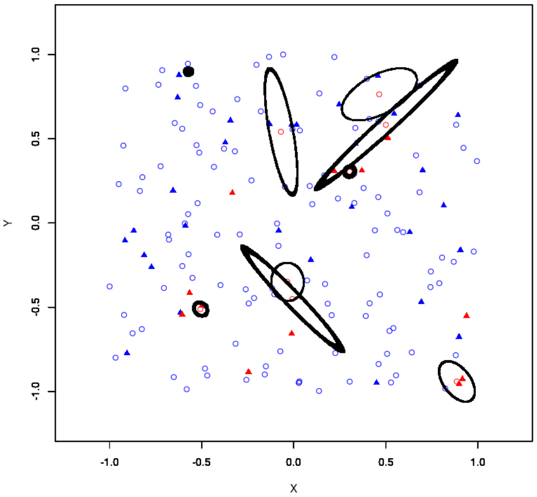

Figure 2 – Illustration of the behavior of our algorithm when the shape and the orientation of the ellipsoids are optimized.

The last equality shows that $\mathbf{M}$ is, by construction, *positive semi definite* as it is a convex combination of *positive semi definite matrices* of rank 1. Furthermore, because of the addition of the Identity matrix to the previous matrices, $\mathbf{M}$ is *positive definite.*

Let us now insert the closed form solution of $\mathbf{M}$ and R in (3) in order to have the dual formulation of Problem 2.

$$\begin{aligned}
\underset{\boldsymbol{\alpha},\beta,\delta}{min} \quad & \boldsymbol{\alpha}^T\left(\frac{1}{4\lambda}\mathbf{G}' + \frac{1}{4\mu}\mathbb{1}_{d\times d}\right)\boldsymbol{\alpha} + \frac{\beta^2}{4\mu} + \frac{\delta^2}{4\mu} + \\
& \boldsymbol{\alpha}^T\left(diag(\mathbf{G}) - \left(B + \frac{\beta}{2\mu} - \frac{\delta}{2\mu}\right)\mathbb{1}_d\right) \\
& +\beta\left(B - \frac{\delta}{2\mu}\right), \\
s.t. \quad & 0 \le \alpha_i \le \frac{1}{n}, \quad \forall i = 1,\dots,n, \\
& \beta,\delta \ge 0,
\end{aligned} \tag{7}$$

where $\mathbf{G}$ is the Gram matrix defined by $G_{ij} = \langle(\boldsymbol{x}_i - \boldsymbol{c}), (\boldsymbol{x}_j - \boldsymbol{c})\rangle$ and $\mathbf{G}'$ is the Hadamard product of $\mathbf{G}$ with itself. $\mathbb{1}_d$ (respectively $\mathbb{1}_{d\times d}$) represents a vector of length d (respectively a matrix of size $d \times d$) where entries are equal to 1.

4 Generalization Guarantees

In this section, we provide generalization guarantees for our approach. We prove a generalization bound on the capacity of our method to exclude negative instances from the learned balls. We propose to derive this bound according to the framework of uniform stability [BE02].

4.1 Uniform Stability

Roughly speaking, an algorithm is *stable* if its output does not change significantly under a small modification of the training sample. This variation must be bounded in $O(1/n)$ in terms of infinite norm where n is the size of the training set S *i.i.d.* from an unknown distribution P.

Definition 1 ([BE02]) *A learning algorithm has a uniform stability in* $\frac{\beta}{n}$ *with respect to a loss function* ℓ *and a parameter set* θ*, with* β *a positive constant if :*

$$\forall S,\ \forall i,\ 1 \le i \le n,\ \sup_{\boldsymbol{x}} |\ell(\theta_S, \boldsymbol{x}) - \ell(\theta_{S^i}, \boldsymbol{x})| \le \frac{\beta}{n},$$

where S *is a learning sample of size* n*,* θ_S *the model parameters learned from* S*,* θ_{S^i} *the model parameters learned from the sample* S^i *obtained by replacing the* i^{th} *example* $\boldsymbol{x}_i$ *from* S *by another example* $\boldsymbol{x}'_i$ *independent from* S *and drawn from* P*.* $\ell(\theta_S, \boldsymbol{x})$ *is the loss suffered at* $\boldsymbol{x}$*.*

In this definition, S^i represents the notion of small modification of the training sample. From Definition 1, one can obtain the following generalization bound [1] :

Theorem 1 (from[BE02], Thm 12) *Let* $\delta > 0$ *and* $n > 1$*. For any algorithm with uniform stability* β/n*, using a loss function bounded by* K*, with probability* $1 - \delta$ *over the random draw of* S *:*

$$L(\theta_S) \le \hat{L}_S(\theta_S) + \frac{2\beta}{n} + (4\beta + K)\sqrt{\frac{\ln 1/\delta}{2n}},$$

where $L(\cdot)$ *is the true risk and* $\hat{L}_S(\cdot)$ *its empirical estimate over* S*.*

4.2 Generalization Bound

Given a centroid c (representing a positive instance) and a learning sample $S = \{\boldsymbol{x}_i\}_{i=1}^n$ of negative instances drawn *i.i.d.* from an unknown probability distribution P_-, the set of parameters to be learned by

[1] This result was proposed in the context of regression and classification tasks. However, one can easily check that it also holds for the setting considered in this section.

ME^2 is the pair $(R, \mathbf{M})$. For convenience, we consider the following optimization problem that is equivalent to Problem 2 :

$$\begin{array}{ll} \underset{R,\mathbf{M}}{min} & \sum_{i=1}^{n} \ell(R,\mathbf{M},\boldsymbol{x}_i) + \mu(B-R)^2 + \lambda\|\mathbf{M}-\mathbf{I}\|_F^2, \\ s.t. & B \geq R \geq 0. \end{array} \tag{8}$$

where $\ell(\cdot)$ represents the loss such that $\ell(R,\mathbf{M},\boldsymbol{x}_i) = \dfrac{1}{n}[R^2 - \|\boldsymbol{x}_i - \boldsymbol{c}\|_M^2]_+$ with $[\cdot]_+$ the hinge loss function : $[a]_+ = \max(a,0)$.

The true risk is defined by $L(\mathbf{M},R) = \mathbb{E}_{\boldsymbol{x}\sim P_-}\ell(\mathbf{M},R,\boldsymbol{x})$ and its empirical estimate over the sample S by $\hat{L}_S(\mathbf{M},R) = \frac{1}{n}\sum_{i=1}^{n}\ell(\mathbf{M},R,\boldsymbol{x}_i)$. We also denote the regularization term as $N(\mathbf{M},R) = \mu(B-R)^2 + \lambda\|\mathbf{M}-\mathbf{I}\|_F^2$. B is set such that $B \geq \max_{\boldsymbol{x}\sim P_-}\|\boldsymbol{x}\|$ and $B \geq \|\boldsymbol{c}\|$. F_S represents the objective function to be minimized, *i.e.* :

$$F_S(\mathbf{M},R) = \hat{L}(\mathbf{M},R) + N(\mathbf{M},R).$$

Note here that it can easily be checked that our loss function ℓ is convex with respect to $\mathbf{M}$ and R.

To prove a generalization bound on our algorithm ME^2, we need to prove that our setting verifies the definition of uniform stability. For this purpose, we first prove that our loss function is actually k-lipschitz in its two first arguments.

Lemma 1 *The loss l is k-lipschitz w.r.t to* $\mathbf{M}$ *and* R *with* $k = \max(1, 4B^2)$*, i.e. : For any* $(\mathbf{M},R)$, $(\mathbf{M}',R')$, $\forall\boldsymbol{x}$ *:*

$$|\ell(\mathbf{M},R,\boldsymbol{x}) - \ell(\mathbf{M}',R',\boldsymbol{x})| \leq k\|(\mathbf{M},R)-(\mathbf{M}',R')\|$$

where $\|(\mathbf{M},R)-(\mathbf{M}',R')\| = |R-R'| + \|\mathbf{M}-\mathbf{M}'\|_F$.

Proof 1

$$\begin{aligned} & |\ell((\mathbf{M},R),\boldsymbol{x}) - \ell((\mathbf{M}',R'),\boldsymbol{x})| \\ = \quad & |\left[R - \|\boldsymbol{x}-\boldsymbol{c}\|_{\mathbf{M}}^2\right]_+ - \left[R' - \|\boldsymbol{x}-\boldsymbol{c}\|_{\mathbf{M}'}^2\right]_+|, \\ \leq \quad & (|R-R'| + |\|\boldsymbol{x}-\boldsymbol{c}\|_{\mathbf{M}}^2 - \|\boldsymbol{x}-\boldsymbol{c}\|_{\mathbf{M}'}^2|), \qquad (9) \\ = \quad & (|R-R'| + |(\boldsymbol{x}-\boldsymbol{c})^T(\mathbf{M}-\mathbf{M}')(\boldsymbol{x}-\boldsymbol{c})|), \\ \leq \quad & (|R-R'| + 4B^2\|\mathbf{M}-\mathbf{M}'\|_F), \qquad (10) \\ \leq \quad & \max(1,4B^2)(|R-R'| + \|\mathbf{M}-\mathbf{M}'\|_F). \end{aligned}$$

Line 9 uses the fact that the hinge loss is 1*-lipschitz and a property of the absolute value. Line 10 can be obtained by the Cauchy-Schwarz inequality and classic properties on norms.* □

We now need a technical lemma on the objective function F_S.

Lemma 2 *Let S be a learning sample, let F_S and F_{S^i} be two objective functions with respect to sample S and S^i and let* $(\mathbf{M},R)$ *and* $(\mathbf{M}^i,R^i)$ *be their respective minimizers. We also define* $\Delta(\mathbf{M},R) = (\mathbf{M}^i,R^i) - (\mathbf{M},R)$ *and recall that* $N(\mathbf{M},R) = \mu(B-R)^2 + \lambda\|\mathbf{M}-\mathbf{I}\|_F^2$*. We have, for all* $t \in [0,1]$ *:*

$$\begin{aligned} & N(\mathbf{M},R) - N((\mathbf{M},R) + t\Delta(\mathbf{M},R)) \\ & \quad + N((\mathbf{M}^i,R^i) - N((\mathbf{M}^i,R^i) - t\Delta(\mathbf{M},R)) \\ & \qquad \leq \frac{2t\max(1,4B^2)}{n}\|\Delta(\mathbf{M},R)\|. \end{aligned}$$

The proof of this Lemma can be found in Section B of the appendix. With this result, we are able to prove the stability of our algorithm.

Proposition 1 *It exists a positive constant κ such that the algorithm ME^2 is uniformly stable with* $\beta = \dfrac{2(\max(1,4B^2))^2}{\kappa\min(\mu,\lambda)}$.

Proof 2 *Setting* $t = \dfrac{1}{2}$ *we have from previous Lemma :*

$$\mu\theta(R) + \lambda\tau(\mathbf{M}) \leq \frac{\max(1,4B^2)}{n}\|\Delta(\mathbf{M},R)\|, \textit{with} \tag{11}$$

$$\begin{aligned} \theta(R) &= (B-R)^2 + (B-R^i)^2 \\ & -(B-(R+\frac{1}{2}(R^i-R)))^2 - (B-(R^i-\frac{1}{2}(R^i-R)))^2, \end{aligned} \tag{12}$$

and

$$\begin{aligned} \tau(\mathbf{M}) &= \|\mathbf{M}-\mathbf{I}\|_F^2 - \|\mathbf{M} + \frac{1}{2}(\mathbf{M}^i-\mathbf{M}) - \mathbf{I}\|_F^2 \\ & + \|\mathbf{M}^i - \mathbf{M}\|_F^2 - \|\mathbf{M}^i - \frac{1}{2}(\mathbf{M}^i-\mathbf{M}) - \mathbf{I}\|_F^2. \end{aligned} \tag{13}$$

By developing Equation (12) we get :

$$\begin{aligned} \theta(R) &= (B-R)^2 + (B-R^i)^2 - 2(B - \frac{1}{2}(R+R^i))^2, \\ &= 2B^2 - 2B(R+R^i) + R^2 + {R^i}^2 \\ & \quad -2\left(B^2 - B(R+R^i) + \frac{1}{4}(R+R^i)^2\right), \\ &= \frac{1}{2}\left(R^2 + {R^i}^2 - 2RR^i\right) = \frac{1}{2}(R-R^i)^2. \end{aligned}$$

Similarly for Equation (13), we have :

$$\begin{aligned}
\tau(\mathbf{M}) &= \|\mathbf{M}-\mathbf{I}\|_F^2 + \|\mathbf{M}^i-\mathbf{I}\|_F^2 \\
&\quad -\|\frac{1}{2}(\mathbf{M}+\mathbf{M}^i)-\mathbf{I}\|_F^2 \\
&\quad -\|\frac{1}{2}(\mathbf{M}+\mathbf{M}^i)-\mathbf{I}\|_F^2, \\
&= \|\mathbf{M}-\mathbf{I}\|_F^2 + \|\mathbf{M}^i-\mathbf{I}\|_F^2 \\
&\quad -\frac{1}{2}\|(\mathbf{M}-\mathbf{I})+(\mathbf{M}^i-\mathbf{I})\|_F^2, \\
&= \frac{1}{2}\left(\|\mathbf{M}-\mathbf{I}\|_F^2 + \|\mathbf{M}^i-\mathbf{I}\|_F^2\right) \\
&\quad +\frac{1}{2}\left(+\sum_{i,j}((\mathbf{M}-\mathbf{I})*(\mathbf{M}^i-\mathbf{I}))_{(i,j)}\right), \\
&= \frac{1}{2}\sum_{k,j}((m_{kj}-\delta_{kj})^2+(m^i_{kj}-\delta_{kj})^2) \\
&\quad +\frac{1}{2}\sum_{k,j}((m_{kj}-\delta_{kj})(m^i_{kj}-\delta_{kj})), \\
&= \frac{1}{2}\|\mathbf{M}-\mathbf{M}^i\|_F^2.
\end{aligned}$$

We can then write the Inequality (11) *as :*

$$\mu(R^i-R)^2+\lambda\|\mathbf{M}^i-\mathbf{M}\|_F^2 \leq \frac{2\max(1,4B^2)}{n}\|\Delta(\mathbf{M},R)\|. \tag{14}$$

Recall that : $\|\Delta(\mathbf{M},R)\| = |R-R^i| + \|\mathbf{M}^i-\mathbf{M}\|_F$*. Because we are working in a finite space, all the norms are equivalent, i.e. there exists a positive constant* κ *such that,* $\forall(R,R^i)\in\mathbb{R}^+$, $\forall(\mathbf{M},\mathbf{M}^i)\in\mathbb{R}^{d\times d}$ *we have :*

$$\kappa(|R-R^i|+\|\mathbf{M}^i-\mathbf{M}\|_F)^2 \leq (R-R^i)^2+\|\mathbf{M}^i-\mathbf{M}\|_F^2.$$

Finally, by combining the previous inequality with Equation (14) and with a reorganization of the terms, we have :

$$\|\Delta(\mathbf{M},R)\| \leq \frac{2\max(1,4B^2)}{n\kappa\min(\mu,\lambda)}.$$

Let $\boldsymbol{x}\in\mathbb{R}$*, starting from the left-hand side of Definition 1 and applying Lemma 2 once and the previous inequality leads to our result :*

$$\begin{aligned}
&|\ell((\mathbf{M},R),\boldsymbol{x})-\ell((\mathbf{M}^i,R^i),\boldsymbol{x})| \\
\leq\ & \max(1,4B^2)\|\Delta(\mathbf{M},R)\|, \\
\leq\ & \frac{2}{n\kappa\min(\mu,\lambda)}(\max(1,4B^2))^2.
\end{aligned}$$

□

To prove the generalization bound, it remains to show that our loss function ℓ is bounded. First we prove a bound on the Frobenius norm of any matrix $\mathbf{M}$ induced by our algorithm.

Lemma 3 *Let* $(\mathbf{M},R)$ *be an optimal solution of Problem 8, we have :*

$$\|\mathbf{M}\|_F \leq \sqrt{\frac{\mu B^2}{\lambda}+d}.$$

Proof 3 *Since* $(\mathbf{M},R)$ *is an optimal solution of Problem 8, we have :*

$$F_S((\mathbf{M},R),\boldsymbol{x}) \leq F_S((I,0),\boldsymbol{x}).$$

Developing the expression of both sides and using the fact that $\left[R-\|\boldsymbol{x}-c\|_{\mathbf{M}}^2\right]_+ + \mu(B-R)^2 \geq 0$*, we have :*

$$\begin{aligned}
\lambda\|\mathbf{M}-\mathbf{I}\|_F^2 &\leq \mu B^2, \\
\Leftrightarrow \|\mathbf{M}-\mathbf{I}\|_F^2 &\leq \frac{\mu B^2}{\lambda}, \\
\Leftrightarrow \|\mathbf{M}\|_F^2 - \|I\|_F^2 &\leq \frac{\mu B^2}{\lambda}, \qquad (15)\\
\Leftrightarrow \|\mathbf{M}\|_F &\leq \sqrt{\frac{\mu B^2}{\lambda}+d}. \qquad (16)
\end{aligned}$$

For line 15, we simply use the triangle inequality, and the last line 16 is based on the fact that $\|I\|_F^2 = d$ *and application of the square root function.* □

Lemma 4 *The loss function* ℓ *is bounded by* $B + 4B^2\sqrt{\frac{\mu B^2}{\lambda}+d}$*.*

Proof 4

$$\begin{aligned}
\ell((\mathbf{M},R),\boldsymbol{x}) &= \left[R-\|\boldsymbol{x}-\boldsymbol{c}\|_{\mathbf{M}}^2\right]_+, \\
&\leq R + B^2\|\mathbf{M}\|_F, \\
&\leq B + 4B^2\sqrt{\frac{\mu B^2}{\lambda}+d}.
\end{aligned}$$

For the first inequality, we use the fact that $[a]_+ \leq |a|$*, we apply the triangle inequality, we use the fact that* $R \leq B$ *by assumption and that* $\|\boldsymbol{x}-\mathbf{c}\|_{\mathbf{M}}^2 \leq (2B)^2\|\mathbf{M}\|_F$*. Then we use Lemma 3 to get the result.* □

Given the stability constant and the fact that the loss is bounded, using Theorem 1, we obtain our final result :

Theorem 2 *Let $\delta > 0$ and $n > 1$. There exists a constant $\kappa > 0$, such that with probability at least $1-\delta$ over the random draw over S, we have for any $(\mathbf{M}, R)$ solution of Problem 8 :*

$$L(\mathbf{M},R) \leq \hat{L}_S(\mathbf{M},R) + \frac{4(\max(1,4B^2))^2}{n\kappa \min(\mu,\lambda)} + \left(\frac{8(\max(1,4B^2))^2}{\kappa \min(\mu,\lambda)} + B + 4B^2\sqrt{\frac{\mu B^2}{\lambda} + d}\right)\sqrt{\frac{\ln 1/\delta}{2n}}$$

Proof 5 *We simply combine Proposition 1, Lemma 4 and Theorem 1.* □

This generalization bound holds for any positive center c. If one has p positive centers, by the union bound, we can extend the previous result for each of the p centers with probability $1-\delta/p$ showing that the models output can control negative instances with high probability. We can notice here that the bound suggests a dependency on the dimensionality of the data d, but this generally holds for any Mahalanobis-based metric learning [VB15].

5 Experiments

We evaluate the behaviour of our approach on seven real datasets coming from the UCI and KEEL databases[2]. We are interested in binary supervised classification tasks where the classes are unbalanced. In these settings, the classic *accuracy* is often irrelevant (as explained in Section 1). We thus evaluate our algorithm ME^2 with a performance measure that is especially dedicated to deal with unbalanced scenarios : the *F-measure* which is the harmonic mean of the *Precision* and *Recall* criteria : $\frac{2 \times Precision \times Recall}{Precision + Recall}$, where : $Precision = \frac{TP}{TP+FP}$ and $Recall = \frac{TP}{TP+FN}$, with TP the number of true positives, FP the number of false positives and TN the number of true negatives.

5.1 Datasets

The characteristics (number of examples, features and the imbalance ratio (I.R.)) of our datasets are described in Table 1. The rate of positive examples in each dataset is between 0.76 (for Abalone 19) and 10.9 (for Yeast3). Because our algorithm does not directly handle categorical variables, such attributes have been replaced by as many binary attributes as the number of modalities of the original feature. For example, in the *Abalone* datasets the following transformation is made for the attribute $V = \{M, I, F\}$: M=(1,0,0), I=(0,1,0) and F=(0,0,1).

Dataset	Nb. ex.	Nb. feat.	I.R.	Rate
Yeast3	1 484	8	8.1	10.9%
Abalone	4 177	8	8.32	10.7%
Wine	1 599	11	29.16	3.3%
Abalone 17	2 338	8	39.31	2.5%
Yeast6	1 484	8	41.4	2.4%
Abalone 20	1 916	8	72.69	1.4%
Abalone19	4 174	8	129.44	0.76%

Table 1 – Number of instances, number of features, Imbalance ratio (i.e. number of negative examples for one positive example) and rate of positives of each real dataset.

5.2 Experimental setup

For each series of experiments, the dataset is separated into a training/validation (S) (80%) set and a test set (20%). We use then a 2-fold cross-validation on the set S while preserving the same I.R in each fold to tune the different methods. Each experiment is repeated 10 times.

Remember that we learn an ellipsoid centered at each positive example of S. This ellipsoid defines in some way the local region of the projection space which is under the influence of the considered positive example. Therefore, this is closely related to the notion of nearest neighborhood. Each ellipsoid depends on two hyperparameters μ and λ as used in our optimization problem (8). To tune these parameters with our 2-fold validation procedure, and according to our previous remark, we associate each example of the validation set to its closest positive example (with respect to the Euclidean distance) in the training set to make use of the specificities of our local models. We use the same strategy at test time. Then, the labeling of new data is predicted as follows (for both validation and test procedure) : it is labelled positive if 1) it belongs to the ellipsoid of its associated positive example in the training set and 2)if it is at most the m^{th} closest nearest neighbour of the ellipsoid's center. Otherwise, the data is classified as negative.Note that the value of m in step 2) allows ME^2 to control the FP rate especially in sce-

[2] These datasets can be found either on the *UCI Repository* (https ://archive.ics.uci.edu/ml/datasets.html) or the *KEEL* website in the "Imbalanced data sets for classification" repository (http ://sci2s.ugr.es/keel/imbalanced.php ?order=ir#sub60).

narios where the imbalance ratio is large. In practice, a small value of m has to be considered, in our experiments we will take the value $m = 3$ for all the datasets for the sake of simplicity.

The hyper-parameters μ and λ are tuned by maximizing the F-Measure for each local model according to the previous classification rule. Note that this is only possible when both positive and negative examples belong to the local validation set. In the case where there are only negative examples, we keep the pair (μ, λ) which minimizes the FP Rate. At training time, μ and λ are tuned respectively in the range $\{0.75, 0.8, 0.85, 0.9, 0.95, 1, 2, 10\}$ and $\{10^{-6:2}\}$.

ME^2 is compared with the following methods :

- A random forest (RF) classifier (with 10 trees).
- A classic decision tree (DT) where we allow the tree to have one example in its leafs.
- To deal with the unbalanced datasets, a decision tree DT_O (resp. DT_U) with an oversampling (resp. undersampling) strategy consisting in multiplying (resp. dividing) the number of positive (resp. negative) examples by a factor of five (resp. two). We also combine the two previous approaches by learning a DT_{OU} decision tree.
- Two Support Vectors Machines, one with a linear kernel (LSVM) and another one with an RBF kernel (RBFSVM). In order to adapt the SVMs to unbalanced datasets, we have given the same global weight to the two classes. The hyperparameters are tuned using the validation set (with both negative and positive examples).

All the classifiers are trained using the corresponding machine learning packages in **R** [3]. All the methods considered are summarized in Table 2 as well as their hyper-parameters and the related **R** package. Note that the optimisation problem of ME^2 is solved using the package *Rsolnp* of **R**.

5.3 Results

The results are reported in Table 3. The datasets are sorted from the least to the most imbalance ratio to see the effect of ME^2 with a decreasing rate of positives. If we look at the first two datasets (i.e. *Yeast3* and *Abalone*) where the rate of positive examples is greater than 10%, our method is less effective than the other baselines. In fact a simple decision tree (with or without sampling) gives the best results (a F-Measure equal to 0.82 for DT while our method outputs 0.60). If ME^2 achieves a similar Recall (0.87) the precision remains too low to get a high F-Measure. This behavior can be easily explained by the fact that many ellipsoids are learned, inducing a large overlapping leading to an important number of FP. ME^2 is more devoted to deal with highly unbalanced scenarios. This is confirmed on the other five datasets. A decision tree in its simple version is no more able to detect any positive examples at test time. Even a combination of trees (RF) cannot detect positive examples when the rate is less than 2%. On the other hand, our ME^2 algorithm gets the best results on those five datasets. By learning both the shape and the orientation of the ellipsoids, our algorithm is able to adapt to the local peculiarities of the space. Said differently, the capacity of the family of hypotheses induced by ME^2 is better than the others, especially the decision tree-based methods which constrain the learning of hypercubes parallel to the axis.

Algorithm	R Package	Hyper-parameter
RF	RandomForest	10 trees
DT	C50	-
DT_O	C50	positive examples ×5
DT_U	C50	negative examples /2
DT_{OU}	C50	pos. ×5 and neg. /2
$LSVM$	e1071	$cost \in \{10^{-1:3}\}$
RBFSVM	e1071	$cost \in \{10^{-1:3}\}$ $\gamma \in \{0.5, 1, 2, 4\}$
ME^2	Rsolnp	$B = 4$ $\lambda \in [10^{-6}, 100]$ $\mu \in [0.75, 10]$

TABLE 2 – Algorithms used in our experimental comparison.

To go deeper in the analysis of ME^2, Figure 3 illustrates the behavior of the 8 methods on the *Yeast6* dataset where we have artificially decreased more and more the rate of positive examples. For each rate, we have evaluated the mean value of the F-Measure over five runs. We can see that ME^2 algorithm (the top curve on both graphs of the Figure) still gives the best results while some most of the other methods (5 over 7) are no more able to detect positive examples while reaching a 98.75% rate of negative examples.

[3] https ://www.r-project.org/

Algorithm	Score	Yeast3	Abalone	Wine	Abalone17	Yeast6	Abalone20	Abalone19
RF	Pre.	0.84 (0.04)	0.76 (0.04)	0.05 (0.15)	0.43 (0.34)	0.10 (0.20)	0 (0)	0 (0)
	Rec.	0.76 (0.06)	0.60 (0.05)	0.009 (0.03)	0.14 (0.10)	0.028 (0.06)	0 (0)	0 (0)
	F-Mea.	0.79 (0.03)	0.67 (0.04)	0.015 (0.05)	0.20 (0.15)	0.044 (0.09)	0 (0)	0 (0)
DT	Prec.	0.76 (0.12)	0.74 (0.04)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
	Rec.	0.88 (0)	0.68 (0.04)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
	F-Mea.	**0.82 (0.12)**	**0.71 (0.04)**	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
DT_O	Prec.	0.67 (0.05)	0.59 (0.03)	0.061 (0.07)	0.34 (0.07)	0.093 (0.08)	0.017 (0.05)	0.019 (0.04)
	Rec.	0.91 (0.02)	0.77 (0.06)	0.063 (0.07)	0.36 (0.09)	0.086 (0.07)	0.020 (0.06)	0.033 (0.067)
	F-Mea.	0.77 (0.03)	0.67 (0.02)	0.061 (0.07)	0.35 (0.08)	0.089 (0.08)	0.018 (0.06)	0.024 (0.05)
DT_U	Prec.	0.75 (0.03)	0.68 (0.08)	0.11 (0.09)	0.28 (0.08)	0.099 (0.10)	0.17 (0.15)	0 (0)
	Rec.	0.90 (0.03)	0.73 (0.09)	0.072 (0.06)	0.43 (0.22)	0.086 (0.07)	0.18 (0.14)	0 (0)
	F-Mea.	**0.82 (0.02)**	0.69 (0.02)	0.082 (0.06)	0.33 (0.13)	0.088 (0.08)	0.18 (0.11)	0 (0)
DT_{OU}	Prec.	0.61 (0.04)	0.50 (0.03)	0.065 (0.04)	0.23 (0.06)	0.14 (0.09)	0.12 (0.13)	0.024 (0.03)
	Rec.	0.94 (0.03)	0.83 (0.05)	0.12 (0.07)	0.47 (0.18)	0.21 (0.13)	0.22 (0.21)	0.083 (0.11)
	F-Mea.	0.74 (0.03)	0.62 (0.03)	0.084 (0.05)	0.31 (0.08)	0.17 (0.10)	0.15 (0.16)	0.037 (0.05)
LSVM	Prec.	0.51 (0)	0.47 (0)	0.049 (0.01)	0.17 (0.012)	0.090 (0)	0.12 (0)	0.020 (0)
	Rec.	0.88 (0)	0.93 (0)	0.49 (0.25)	0.81 (0)	0.43 (0)	0.80 (0)	0.68 (0.12)
	F-Mea.	0.64 (0)	0.62 (0)	0.087 (0.01)	0.29 (0.02)	0.15 (0.01)	**0.21 (0)**	0.038 (0.01)
RBFSVM	Pre.	0.67 (0.07)	0.59 (0.01)	0.78 (0.20)	0.29 (0.12)	0.087 (0.07)	0 (0)	0 (0)
	Rec.	0.78 (0.15)	0.67 (0.07)	0.091 (0)	0.13 (0.08)	0.10 (0.09)	0 (0)	0 (0)
	F-Mea.	0.70 (0.07)	0.63 (0.04)	**0.16 (0.01)**	0.17 (0.10)	0.092 (0.08)	0 (0)	0 (0)
ME^2	Prec.	0.46 (0.02)	0.50 (0.02)	0.11 (0.03)	0.25 (0.03)	0.19 (0.04)	0.14 (0.07)	0.024 (0.02)
	Rec.	0.87 (0.02)	0.83 (0.02)	0.29 (0.08)	0.71 (0.12)	0.43 (0.06)	0.44 (0.14)	0.13 (0.10)
	F-Mea.	0.60 (0.02)	0.62 (0.02)	**0.16 (0.04)**	**0.37 (0.05)**	**0.26 (0.05)**	**0.21 (0.09)**	**0.040 (0.03)**

Table 3 – Comparison of ME^2 with seven methods on the UCI and KEEL datasets described in Table 1. The values represent the mean of each criterion (P : Precision, R : Recall, F : F-Measure) over the 10 runs ; the value between parenthesis is the corresponding standard deviation. A standard deviation equal to zero indicates a value smaller than 10^{-2}. The best results are indicated in bold font.

6 Conclusion

We have presented a method to learn *Maximum Excluding Ellipsoids* in the context of unbalanced binary classification tasks. Our algorithm, called ME^2, is simple, because based on local linear models, and theoretically supported by generalization guarantees that have been derived by using the uniform stability framework. We have shown that our method is particularly efficient and robust when the rate of positive examples is very small. The reason comes from the fact that ME^2 is able to learn decision boundaries in the form of ellipsoids (via a metric learning-based strategy) that are optimized locally to fit the best the specificities of the space.

ME^2 is based on a very simple decision rule looking for the nearest ellipsoid to a test query. We think that this rule may benefit from further investigation, e.g. by considering a combination of ellipsoids to predict the label of a test data. From a theoretical point of view, we have derived a guarantee on the learned matrix M and radius R. Since our decision rule is closed to that of a nearest neighbor classifier, it would be interesting to establish a link between the quality of M and R and the generalization error of such a classifier.

Références

[AA15] Shikha Agrawal and Jitendra Agrawal. Survey on anomaly detection using data mining techniques. In *19th International Conference in Knowledge Based and Intelligent Information and Engineering Systems KES*, volume 60 of *Procedia Computer Science*, pages 708 – 713. Elsevier, 2015.

[Agg13] Charu C. Aggarwal. *Outlier Analysis.* Springer, 2013.

[ALC14] Meriem El Azami, Carole Lartizien, and Stéphane Canu. Robust outlier detection

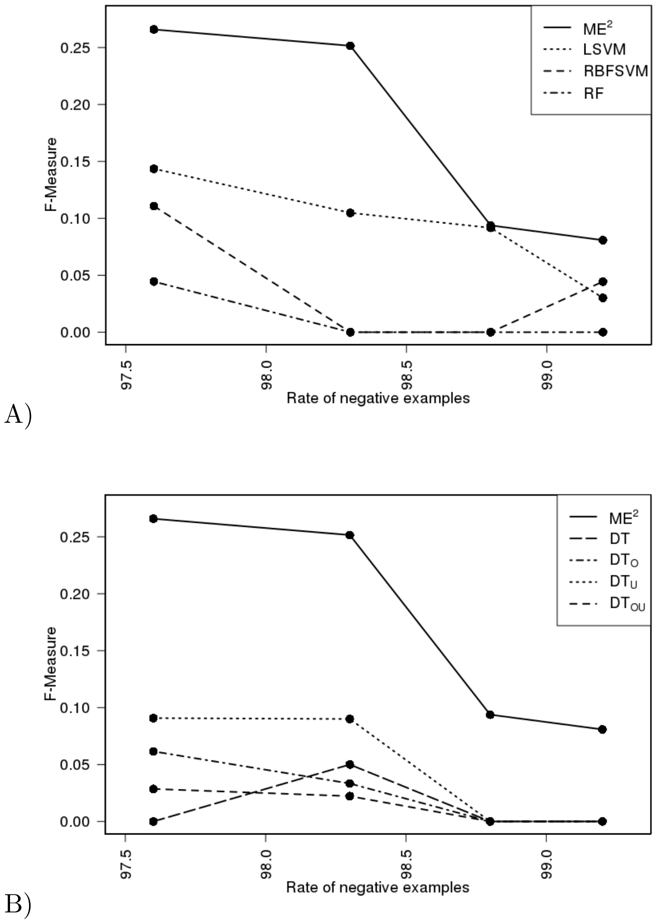

Figure 3 – F-Measure values with respect to the rate of negative examples of the different algorithms in the dataset *Yeast6*. For each rate of negative examples (97.6, 98.3, 98.8 and 99.2) we give the mean value of the F-Measure over five runs for the different algorithms. For the sake of clarity the standard deviation is not represented on both figures. For the same reason, we compare the ME^2 algorithm with *LSVM, RBFSVM* and *RF* on A), and with the different decision trees (with or without sampling methods) on B)

with L0-SVDD. In *22th Eu Symposium on Artificial Neural Net. (ESANN)*, 2014.

[BE02] Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2 :499–526, 2002.

[BHS13] Aurélien Bellet, Amaury Habrard, and Marc Sebban. A survey on metric learning for feature vectors and structured data. *CoRR*, abs/1306.6709, 2013.

[BJZ12] M. El Boujnouni, M. Jedra, and N. Zahid. New decision function for support vector data description. In *Snd Int. Conf. on Innovative Computing Technology (INTECH 2012)*, pages 305–310, 2012.

[HSKS03] Katherine A. Heller, Krysta M. Svore, Angelos D. Keromytis, and Salvatore J. Stolfo. One class support vector machines for detecting anomalous windows registry accesses. In *ICDM work. on Data Min. for Computer Security*, 2003.

[KCP11] Mohammed Khalilia, Sounak Chakraborty, and Mihail Popescu. Predicting disease risks from highly imbalanced data using random forest. *BMC Medical Informatics and Decision Making*, 11(1) :51, 2011.

[Kul13] Brian Kulis. Metric learning : A survey. *Foundations and Trends in Machine Learning*, 5(4) :287–364, 2013.

[LTM13] Trung Le, Dat Tran, and Wanli Ma. Fuzzy multi-sphere support vector data description. In *17th Pacific-Asia Conference (PAKDD), Part II*, pages 570–581. Springer, 2013.

[LZ06] Yi Liu and Y.F. Zheng. Minimum enclosing and maximum excluding machine for pattern description and discrimination. In *18th IEEE International Conference on Pattern Recognition (ICPR06)*, 2006.

[PA11] Eric J. Pauwels and Onkar Ambekar. One class classification for anomaly detection : Support vector data description revisited. In *Industrial Conference on Data Mining*, pages 25–39, 2011.

[PH15] M. Perrot and A. Habrard. Regressive virtual metric learning. In *Proc. of Annual Conference on Neural Information Processing Systems (NIPS)*, 2015.

[SBS14] Yuan Shi, Aurélien Bellet, and Fei Sha. Sparse compositional metric learning. In *Proc. of AAAI Conference on Artificial Intelligence*, pages 2078–2084, 2014.

[TD04] David M. J. Tax and Robert P. W. Duin. Support vector data description. *Machine Learning Journal*, 54(1) :45–66, 2004.

[VB15] Nakul Verma and Kristin Branson. Sample complexity of learning mahalanobis distance metrics. In *NIPS*, 2015.

[VJ01] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 511–518, 2001.

[WGP10] Zhe Wang, Daqi Gao, and Zhisong Pan. An effective support vector data description with relevant metric learning. In *7th International Symposium on Neural Networks (ISNN), Part II*, pages 42–51. Springer, 2010.

A Dual Formulation

We give further details about how to get the dual formulation of Problem 2. Let us first recall the expression of the Lagrangian for the sake of clarity :

$$\begin{aligned}\mathcal{L}(\boldsymbol{\alpha},\beta,\delta,\boldsymbol{\gamma},R,\boldsymbol{\xi},\mathbf{M}) &= \frac{1}{n}\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\gamma_i\xi_i + \mu(B-R)^2 \\ &\quad - \sum_{i=1}^{n}\alpha_i\left(\|x_i-c\|_{\mathbf{M}}^2 - R + \xi_i\right) \\ &\quad + \lambda\|\mathbf{M}-\mathbf{I}\|_F^2 - \beta R + \delta(R-B),\end{aligned}$$

We now set : $A = \sum_{i=1}^{n}\alpha_i\|\boldsymbol{x}_i - c\|_{\mathbf{M}}^2$ and $N = Tr((\mathbf{M}-\mathbf{I})^T(\mathbf{M}-\mathbf{I}))$ and we develop these two expressions.

By developing the expression of A, we get :

$$\begin{aligned}A &= \sum_{i=1}^{n}\alpha_i(\boldsymbol{x}_i-\boldsymbol{c})^T(\boldsymbol{x}_i-\boldsymbol{c}) \\ &+\frac{1}{2\lambda}\sum_{i=1}^{n}\sum_{k=1}^{n}\alpha_i\alpha_k(\boldsymbol{x}_i-\boldsymbol{c})^T(\boldsymbol{x}_k-\boldsymbol{c})(\boldsymbol{x}_k-\boldsymbol{c})^T(\boldsymbol{x}_i-\boldsymbol{c}),\end{aligned}$$

so that, using the previous notations :

$$\begin{aligned}A &= \boldsymbol{\alpha}^T diag(\mathbf{G}) + \frac{1}{2\lambda}\sum_{i=1}^{n}\sum_{k=1}^{n}\alpha_i\alpha_k G_{ik}G_{ki}, \\ A &= \boldsymbol{\alpha}^T diag(\mathbf{G}) + \frac{1}{2\lambda}\boldsymbol{\alpha}^T\mathbf{G}'\boldsymbol{\alpha}\end{aligned}$$

where $\mathbf{G}$ is the Gram matrix defined by $G_{ij} = \langle(\boldsymbol{x}_i - \boldsymbol{c}),(\boldsymbol{x}_j - \boldsymbol{c})\rangle$ and $\mathbf{G}'$ is the Hadamard product of $\mathbf{G}$ with itself. Because $\mathbf{G}$ is *positive semi definite*, so is $\mathbf{G}'$.

From now on, we set $\boldsymbol{y}_k = \boldsymbol{x}_k - \boldsymbol{c}$ for convenience. Let us denote by $y_{ki} = (\boldsymbol{x}_k - \boldsymbol{c})_i$ the i^{th} element of the vector $\boldsymbol{x}_k - \boldsymbol{c}$.

$$\begin{aligned}N &= \frac{1}{4\lambda^2}\sum_{i=1}^{d}\sum_{j=1}^{d}\left(\sum_{k=1}^{n}\alpha_k y_{ki}y_{kj}\right)^2, \\ &= \frac{1}{4\lambda^2}\sum_{k=1}^{n}\alpha_k^2\left(\sum_{i=1}^{d}\sum_{j=1}^{d}y_{ki}^2y_{kj}^2\right) \\ &\quad +\frac{2}{4\lambda^2}\sum_{k>l}^{n}\left(\alpha_k\alpha_l\sum_{i=1}^{d}\sum_{j=1}^{d}y_{ki}y_{kj}y_{li}y_{lj}\right), \\ &= \frac{1}{4\lambda^2}\sum_{k=1}^{n}\alpha_k^2\left(\sum_{i=1}^{d}y_{ki}^2\sum_{j=1}^{d}y_{kj}^2\right) \\ &\quad +\frac{2}{4\lambda^2}\sum_{k>l}^{n}\left(\alpha_k\alpha_l\sum_{i=1}^{d}y_{ki}y_{li}\sum_{j=1}^{d}y_{kj}y_{lj}\right), \\ &= \frac{1}{4\lambda^2}\sum_{k=1}^{n}\alpha_k^2\ G_{kk}^4 + \frac{2}{4\lambda^2}\sum_{k>l}^{n}\left(\alpha_k\alpha_l G_{kl}^2G_{lk}^2\right), \\ N &= \frac{1}{4\lambda^2}\boldsymbol{\alpha}^T\mathbf{G}'\boldsymbol{\alpha}.\end{aligned}$$

We do the same by replacing R by its expression in (3). For the sake of simplicity, we only consider the terms of the Lagrangian where R appears and set $D = \mu(B-R)^2 + R\sum_{k=1}^{n}\alpha_k + \delta(R-B) - \beta R$. We obtain :

$$\begin{aligned}D &= \mu\left(B - \frac{\beta-\delta+2\mu B-\sum_{k=1}^{n}\alpha_k}{2\mu}\right)^2 \\ &\quad +\sum_{k=1}^{n}\alpha_k\left(\frac{\beta-\delta+2\mu B-\sum_{k=1}^{n}\alpha_k}{2\mu}\right) \\ &\quad +\delta\left(\frac{\beta-\delta+2\mu B-\sum_{k=1}^{n}}{2\mu}\right) \\ &\quad -\beta\left(\frac{\beta-\delta+2\mu B-\sum_{k=1}^{n}\alpha_k}{2\mu}\right) \\ &= \frac{1}{4\mu}\left[\left(\sum_{k=1}^{n}\alpha_k\right)^2 + \beta^2 + \delta^2 - 2\delta\beta - 2\beta\sum_{k=1}^{n}\alpha_k\right. \\ &\quad \left. +2\delta\sum_{k=1}^{n}\alpha_k\right] + \frac{1}{2\mu}\left[-\left(\sum_{k=1}^{n}\alpha_k\right)^2 + \beta\sum_{k=1}^{n}\alpha_k\right. \\ &\quad \left. -\delta\sum_{k=1}^{n}\alpha_k + B\sum_{k=1}^{n}\alpha_k\right] + \frac{1}{2\mu}\left[\delta\beta - \delta^2\right. \\ &\quad \left. -\delta\sum_{k=1}^{n}\alpha_k\right] + \frac{1}{2\mu}\left[-\beta^2 + \delta\beta - B\beta + \beta\sum_{k=1}^{n}\alpha_k\right]\end{aligned}$$

The second line is obtained by developing each terms and the last by reducing and factorizing each terms.

We then have :

$$\begin{aligned} D &= -\frac{1}{4\mu}\left[\left(\sum_{k=1}^{n}\alpha_k\right)^2 + \beta^2 + \delta^2\right] \\ &\quad + \sum_{k=1}^{n}\left(B + \frac{\beta}{2\mu} - \frac{\delta}{2\mu}\right) + \beta\left(-B + \frac{\delta}{2\mu}\right) \end{aligned}$$

We can now express our Lagrangien (3) with respect to $\boldsymbol{\alpha}, \beta$ and δ only as :

$$\begin{aligned} \mathcal{L}(\alpha) = -\boldsymbol{\alpha}^T diag(\mathbf{G}) - \frac{1}{2\lambda}\boldsymbol{\alpha}^T\mathbf{G}'\boldsymbol{\alpha} + \frac{1}{4\lambda}\boldsymbol{\alpha}^T\mathbf{G}'\boldsymbol{\alpha} \\ - \frac{1}{4\mu}\left[\left(\sum_{k=1}^{n}\alpha_k\right)^2 + \beta^2 + \delta^2\right] \\ + \left(B + \frac{\beta}{2\mu} - \frac{\delta}{2\mu}\right)\sum_{k=1}^{n}\alpha_k + \beta\left(-B + \frac{\delta}{2\mu}\right). \end{aligned}$$

We then have the dual formulation 7 by minimizing the opposite of the Lagrangian with the associated constraints.

B Proof of Lemma 2

In this section we give the technical proof of Lemma 2.

Proof 6 *Since ℓ (the hinge loss) is convex, so is the empirical risk and thus for all $t \in [0,1]$ we have the two following inequalities :*

$$\begin{aligned} &\hat{L}_{S^i}((\mathbf{M},R) + t\Delta(\mathbf{M},R)) - \hat{L}_{S^i}(\mathbf{M},R) \\ &\qquad \le t\hat{L}_{S^i}((\mathbf{M}^i,R^i) - t\hat{L}_{S^i}(\mathbf{M},R). \end{aligned}$$

and

$$\begin{aligned} &\hat{L}_{S^i}((\mathbf{M}^i,R^i)) - t\Delta(\mathbf{M},R)) - \hat{L}_{S^i}(\mathbf{M}^i,R^i) \\ &\qquad \le t\hat{L}_{S^i}((\mathbf{M},R) - t\hat{L}_{S^i}(\mathbf{M}^i,R^i) \end{aligned}$$

We get the second inequality by switching the role of $(\mathbf{M},R)$ and $(\mathbf{M}^i,R^i)$. If we sum these two inequalities, the right hand side vanishes and we obtain :

$$\begin{aligned} &\hat{L}_{S^i}((\mathbf{M},R) + t\Delta(\mathbf{M},R)) - \hat{L}_{S^i}(\mathbf{M},R) \\ &\quad + \hat{L}_{S^i}((\mathbf{M}^i,R^i)) - t\Delta(\mathbf{M},R)) - \hat{L}_{S^i}(\mathbf{M}^i,R^i) \\ &\qquad \le 0. \quad (17) \end{aligned}$$

By assumption on $(\mathbf{M},R)$ and $(\mathbf{M}^i,R^i)$ we have :

$$\begin{aligned} F_S((\mathbf{M},R)) - F_S((\mathbf{M},R) + t\Delta(\mathbf{M},R)) &\le 0, \\ F_{S^i}((\mathbf{M}^i,R^i)) - F_{S^i}((\mathbf{M}^i,R^i) - t\Delta(\mathbf{M},R)) &\le 0, \end{aligned}$$

then, summing the two previous inequalities and using (17), we get :

$$\begin{aligned} &\hat{L}_{S^i}((\mathbf{M},R) + t\Delta(\mathbf{M},R)) - \hat{L}_S((\mathbf{M},R) + t\Delta(\mathbf{M},R)) \\ &\qquad - \hat{L}_{S^i}(\mathbf{M},R) + \hat{L}_S(\mathbf{M},R) + \\ &\mu[(B-R)^2 + (B-R^i)^2 - (B-(R+t\Delta R))^2 - (B-(R^i-t\Delta R))^2] \\ &+\lambda[\|\mathbf{M}-\mathbf{I}\|_F^2 + \|\mathbf{M}^i-\mathbf{I}\|_F^2 - \|\mathbf{M}+t\Delta\mathbf{M}-\mathbf{I}\|_F^2 - \|\mathbf{M}^i-t\Delta\mathbf{M}-\mathbf{I}\|_F^2] \\ &\qquad \le 0. \quad (18) \end{aligned}$$

Let us set : $H = \hat{L}_{S^i}((\mathbf{M},R) + t\Delta(\mathbf{M},R)) - \hat{L}_{S^i}(\mathbf{M},R) - \hat{L}_S((\mathbf{M},R) + t\Delta(\mathbf{M},R)) + \hat{L}_S(\mathbf{M},R)$. We will use Lemma 1 to bound this term :

$$\begin{aligned} H &\le |\hat{L}_{S^i}((\mathbf{M},R) + t\Delta(\mathbf{M},R)) - \hat{L}_{S^i}(\mathbf{M},R) \\ &\quad -\hat{L}_S((\mathbf{M},R) + t\Delta(\mathbf{M},R)) + \hat{L}_S(\mathbf{M},R)|, \\ &\le |\frac{1}{n}\sum_{\boldsymbol{x}_i \in S^i}\ell((\mathbf{M},R) + t\Delta(\mathbf{M},R), \boldsymbol{x}_i) \\ &\quad -\frac{1}{n}\sum_{\boldsymbol{x}_i \in S}\ell((\mathbf{M},R) + t\Delta(\mathbf{M},R), \boldsymbol{x}_i) \\ &\quad +\frac{1}{n}\sum_{\boldsymbol{x}_i \in S}\ell((\mathbf{M},R), \boldsymbol{x}_i) - \frac{1}{n}\sum_{\boldsymbol{x}_i \in S^i}\ell((\mathbf{M},R), \boldsymbol{x}_i)|, \\ &\le \frac{1}{n}|\ell((\mathbf{M},R) + t\Delta(\mathbf{M},R), \boldsymbol{x}_i) - \ell((\mathbf{M},R), \boldsymbol{x}_i) \\ &\quad -\ell((\mathbf{M},R) + t\Delta(\mathbf{M},R), \boldsymbol{x}_i') + \ell((\mathbf{M},R), \boldsymbol{x}_i')|, \\ &\le \frac{1}{n}|\ell((\mathbf{M},R) + t\Delta(\mathbf{M},R), \boldsymbol{x}_i) - \ell((\mathbf{M},R), \boldsymbol{x}_i)| \\ &\quad +\frac{1}{n}|\ell((\mathbf{M},R) + t\Delta(\mathbf{M},R), \boldsymbol{x}_i') - \ell((\mathbf{M},R), \boldsymbol{x}_i')|, \\ H &\le \frac{2t\max(1, 4B^2)}{n}\|\Delta(\mathbf{M},R)\|. \end{aligned}$$

We successively apply the definition of the empirical risk and triangle inequality to get the previous inequalities. The last one is obtained using Lemma 1. □