Metric Learning from Imbalanced Data with Generalization Guarantees

Léo Gautheron, Amaury Habrard, Emilie Morvant, Marc Sebban

Pattern Recognition Letters
journal homepage: www.elsevier.com

Metric Learning from Imbalanced Data with Generalization Guarantees

Léo Gautheron[a,**], Amaury Habrard[a], Emilie Morvant[a], Marc Sebban[a]

[a] *Univ Lyon, UJM-Saint-Etienne, CNRS, Institut d Optique Graduate School, Laboratoire Hubert Curien UMR 5516, F-42023, SAINT-ETIENNE, France*

ABSTRACT

Since many machine learning algorithms require a distance metric to capture dis/similarities between data points, *metric learning* has received much attention during the past decade. Surprisingly, very few methods have focused on learning a metric in an imbalanced scenario where the number of positive examples is much smaller than the negatives, and even fewer derived theoretical guarantees in this setting. Here, we address this difficult task and design a new Mahalanobis metric learning algorithm (**IML**) which deals with class imbalance. We further prove a generalization bound involving the proportion of positive examples using the uniform stability framework. The empirical study performed on a wide range of datasets shows the efficiency of **IML**.

1. Introduction

Metric learning [2, 19], subfield of representation learning, consists of designing a pairwise function able to measure the dis/similarity between two data points. This issue is key in machine learning where such metrics are at the core of many algorithms, like k-nearest neighbors (kNN), SVMs, k-Means, etc. To construct a dis/similarity measure suitable for a given task, most metric learning algorithms optimize a loss function which aims at bringing closer examples of the same label while pushing apart examples of different labels. In practice, metric learning is usually performed with cannot link/must link constraints—x and x' should be dis/similar [8, 24, 31, 33, 34, 35]—or relative constraints—x should be more similar to x' than to x'' [20, 26, 31, 37].

In this paper, we focus on the family of metric learning algorithms that construct a Mahalanobis distance $d_{\mathbf{M}}(x, x') = \sqrt{(x - x')^\top \mathbf{M}(x - x')}$ parameterized by a positive semidefinite matrix $\mathbf{M}$. Learning a Mahalanobis distance leads to several nice properties: *(i)* $d_{\mathbf{M}}$ is a generalization of the Euclidean distance; *(ii)* it induces a projection such that the distance between two points is equivalent to their Euclidean distance after a linear projection; *(iii)* $\mathbf{M}$ can be low rank implying a projection in a lower dimensional latent space; *(iv)* it involves optimization problems that are often convex and thus easy to solve. Two famous representatives of Mahalanobis distance learning are **LMNN** (Large Margin Nearest Neighbor [31]) and **ITML** (Information-Theoretic Metric Learning [8]), which are both designed to improve the accuracy of the kNN classification rule in the latent space. The principle of **LMNN** is the following: for each training example, its k nearest neighbors of the same class (the *target neighbors*) should be closer than examples of other classes (the *impostors*). **ITML** uses a LogDet regularization and minimizes (*resp.* maximizes) the distance between examples of the same (*resp.* different) class. Another recent Mahalanobis distance learning algorithm is **GMML** (Geometric Mean Metric Learning) [35] where the metric is computed using a closed form solution of an unconstrained optimization problem involving similar and dissimilar pairs. In light of these learning procedures, it is worth noticing that the loss functions optimized in **LMNN**, **ITML** and **GMML** (and in most pairwise metric learning methods) tend to favor the majority class as there is no distinction between the constraints involving examples of the majority class and the constraints on the minority class. This strategy is thus not well suited when dealing with imbalanced datasets. An illustration of this phenomenon on the SPECTFHEART dataset from the UCI repository is shown in Figure 1. We observe that decreasing the proportion of minority examples tends to generate a metric which classifies (with a kNN rule) all the examples as the majority class, thus leading to an accuracy close to 1. On the other hand, the F1-measure, which is much more adapted to imbalanced scenarios (it involves both the false positive and false negative rates), de-

** Corresponding author:
e-mail: leo.gautheron@univ-st-etienne.fr (Léo Gautheron)

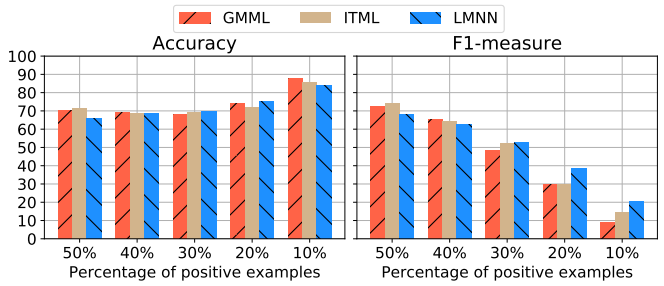

Fig. 1. Illustration on the SPECTFHEART dataset of the negative impact of classic metric learning algorithms when facing an increasing imbalance in the dataset. On the left, as the proportion of minority examples decreases, the metric learning algorithms tend to classify all the examples as the majority class, with an accuracy close to 1. On the right, using the F1-measure, we note that the learned metrics plugged in a kNN actually miss many positives, usually considered as the examples of interest.

creases with the proportion of positives, showing that the classifier missed many positives, usually considered as the examples of interest.

This problem of learning from imbalanced data has been widely tackled in the literature [4, 17]. Classic methods typically make use of over/under-sampling techniques [10, 12, 23, 1] or create synthetic samples in the neighborhood of the minority class—*e.g.,* using SMOTE-like strategies [6, 7, 16] or resorting to adversarial techniques [9]. However, these methods may lead to over or under-fitting and are often subject to an inability to generate enough diversity, especially in a highly imbalanced scenario. Other strategies aim at addressing imbalanced situations directly during the learning process. They include cost-sensitive methods [11, 36] which require prior knowledge on the miss-classification costs, the optimization of imbalance-aware criteria [14, 25, 28] which are often non convex, or ensemble methods based on bagging and boosting strategies [15] that can be computationally expensive.

Unlike the state of the art, we suggest in this paper to address the problem of learning from imbalanced data by optimizing a metric suited to scenarios where the positive data are very scarce. As far as we know, very few methods were designed in this setting. Feng et al. [13] propose to regularize a standard metric learning problem by using the KL-divergence between the classes. Wang et al. [29] proposed **IMLS** that learns a classic metric and then performs a sampling on the training data to account the imbalance. However, as we will see in our experimental study, better performances can be achieved by resorting to a metric dedicated specifically to deal with the imbalance of the application at hand. Note that deep metric learning methods have also received attention by the community to address the problem of imbalanced data [22, 30]. However these methods often require large training datasets, like in visual tasks. However, this requirement is not always fulfilled by the application at hand. Moreover, it is worth noticing that none of the previous approaches come with theoretical guarantees, a gap we will fill in this paper. In order to implicitly control the rates of false positives and false negatives, we propose a new algorithm, called **IML** for Imbalanced Metric Learning, which accounts carefully the nature of the pairwise constraints (by decomposing them with respect to the labels involved in the pairs) and weights their impact in the loss function so as to maximize the F1-measure. Beyond this algorithmic contribution, we further provide a theoretical analysis of **IML** using the uniform stability framework [3]. We derive a generalization bound which has the advantage to involve the proportion of minority examples. This bound provides some insight into the way to tune the weighting parameters to counterbalance the negative impact of imbalanced datasets.

The paper is organized as follows. Section 2 describes our algorithm **IML** which takes the form of a simple regularized convex problem. Section 3 is dedicated to the theoretical analysis. We perform an experimental study of our approach in Section 4 before concluding in Section 5.

2. IML: Imbalanced Metric Learning

2.1. Notations and Setting

In this paper, we deal with binary classification tasks where $\mathcal{X} \subseteq \mathbb{R}^d$ is a d-dimensional input space and $\mathcal{Y} = \{-1, +1\}$ is the binary output space. We further define $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ as the joint space where $z = (x, y) \in \mathcal{Z}$ is a labeled example. In supervised classification, an algorithm is provided with a learning sample $\mathcal{S} = \{z_i{=}(x_i, y_i)\}_{i=1}^n$ of n observations. These observations are drawn *i.i.d.* from a fixed yet unknown distribution $\mathcal{D}$ over $\mathcal{Z}$. We denote this set of n *i.i.d.* observations as $\mathcal{S} \sim \mathcal{D}^n$. We assume that the learning sample is defined as $\mathcal{S} = \mathcal{S}^+ \cup \mathcal{S}^-$, with $\mathcal{S}^+$ the set of positive examples and $\mathcal{S}^-$ the set of negative examples such that the number of positives $n^+ = |\mathcal{S}^+|$ is smaller than the number of negatives $n^- = |\mathcal{S}^-|$ (we say that $+1$ is the minority class and -1 the majority one). The final objective of the learner is to construct a classifier $f : \mathcal{X} \to \mathcal{Y}$ (from a hypothesis space $\mathcal{F}$) which behaves well on unseen data drawn from $\mathcal{D}$. Here, we aim at constructing a Mahalanobis distance which induces a new space in which a kNN will work well on both classes. The Mahalanobis distance is a type of metric parameterized by a positive semidefinite (PSD) matrix $\mathbf{M} \in \mathbb{R}^{d \times d}$ that can be decomposed as $\mathbf{M} = \mathbf{L}^T\mathbf{L}$, where $\mathbf{L} \subseteq \mathbb{R}^{r \times d}$ is a projection induced by $\mathbf{M}$ (where r is the rank of $\mathbf{M}$). A nice property is that the Mahalanobis distance between two points x and x' is equivalent to the Euclidean distance after having projected x and x' in the r-dimensional space, *i.e.*,

$$d^2_{\mathbf{M}}(x, x') = (x - x')^\top \mathbf{M}(x - x') = (\mathbf{L}x - \mathbf{L}x')^\top(\mathbf{L}x - \mathbf{L}x')\,.$$

2.2. Problem Formulation

Classic Mahalanobis metric learning algorithms [2, 5, 18] are usually expressed as follows:

$$\min_{\mathbf{M} \succeq 0} F(\mathbf{M}) = \frac{1}{n^2} \sum_{(z,z') \in \mathcal{S}^2} \ell(\mathbf{M}, z, z') + \lambda Reg(\mathbf{M}), \qquad (1)$$

where one wants to minimize the trade-off between a convex loss ℓ over all pairs of examples and a regularization Reg under the PSD constraint $\mathbf{M} \succeq 0$.

The major drawback of this formulation is that the loss gives the same importance to any pair of examples (z, z') whatever the

labels y and y'. Intuitively, this is not well suited to imbalanced data where the minority class is the set of examples of interest. Some metric learning algorithms [31, 35] allow to weight the role played by the must-link and cannot-link constraints. However, the problem still holds because the labels of the examples are not directly taken into account.

Our simple idea is to decompose further the sets of pairs of examples based on their labels. Each set can then weighted differently during the optimization to reduce the negative effect of the imbalance.

Let us define the loss function of our **IML** algorithm as follows:

$$\ell(\mathbf{M}, z, z') = \begin{cases} a\ell_1(\mathbf{M}, z, z') & \text{if } y{=}+1 \text{ and } y'=+1, \\ (1{-}a)\ell_1(\mathbf{M}, z, z') & \text{if } y{=}-1 \text{ and } y'=-1, \\ b\ell_2(\mathbf{M}, z, z') & \text{if } y{=}+1 \text{ and } y'=-1, \\ (1{-}b)\ell_2(\mathbf{M}, z, z') & \text{if } y{=}-1 \text{ and } y'=+1, \end{cases} \tag{2}$$

with the two functions ℓ_1 and ℓ_2 defined as $\ell_1(\mathbf{M}, z, z') = [d^2_{\mathbf{M}}(x, x') - 1]_+$ and $\ell_2(\mathbf{M}, z, z') = [1{+}m{-}d^2_{\mathbf{M}}(x, x')]_+$ and where $[value]_+ = \max(0, value)$ is the Hinge loss and $m \geq 0$ a margin parameter. The idea of ℓ_1 is to bring examples of the same class at a distance less than 1 while ℓ_2 aims to push far away examples of different classes at a distance larger than 1 plus a given margin m.

Both parameters a and b take values in $[0, 1]$. The parameter a controls the trade-off between bringing closer the minority examples and bringing closer the majority examples. While the second parameter b controls the trade-off between keeping far away the majority examples from the neighborhood of minority ones, and keeping far away minority examples from the neighborhood of majority ones.

Moreover, we set the regularization term of Equation (1) as $Reg(\mathbf{M}) = \|\mathbf{M}{-}\mathbf{I}\|^2_F$ where $\mathbf{I}$ is the identity matrix and $\|.\|_F$ is the Frobenius norm. It aims at avoiding overfitting by enforcing $\mathbf{M}$ to be close to the identity matrix $\mathbf{I}$. Said differently, we aim at being close to the Euclidean distance while satisfying the best the semantic constraints.

All things considered, our **IML** algorithm takes the form of the following convex problem:

$$\min_{\mathbf{M}\geq 0} F(\mathbf{M}) = \frac{1}{n^2}\Bigg(\sum_{(z,z')\in Sim^+} a\ell_1(\mathbf{M}, z, z') + \sum_{(z,z')\in Sim^-} (1{-}a)\ell_1(\mathbf{M}, z, z') + \sum_{(z,z')\in Dis^+} b\ell_2(\mathbf{M}, z, z') + \sum_{(z,z')\in Dis^-} (1{-}b)\ell_2(\mathbf{M}, z, z')\Bigg) + \lambda\|\mathbf{M} - \mathbf{I}\|^2_F, \tag{3}$$

where the four sets Sim^+, Dis^+, Dis^- and Sim^- are defined as subsets of $\mathcal{S} \times \mathcal{S}$ respectively as: $Sim^+ \subseteq \mathcal{S}^+ \times \mathcal{S}^+$, $Dis^+ \subseteq \mathcal{S}^+ \times \mathcal{S}^-$, $Dis^- \subseteq \mathcal{S}^- \times \mathcal{S}^+$ and $Sim^- \subseteq \mathcal{S}^- \times \mathcal{S}^-$.

If we look more closely at the proposed Equation (3), when all pairs from $\mathcal{S} \times \mathcal{S}$ are involved, Sim^+ and Sim^- contain respectively n^+n^+ and n^-n^- pairs while Dis^+ and Dis^- contain respectively n^+n^- and n^-n^+ pairs. This means that the pairs in Dis^+ and Dis^- are symmetric and these two sets might be merged. However, metric learning methods rarely consider all the possible pairs as it becomes quite inefficient in the presence of a large number of examples. Possible strategies to select the pairs include a random subsampling [8, 33, 34, 35] or a selection based on the nearest neighbors rule [24, 31]. For this reason, it might make sense to separate the two sets Dis^+ and Dis^- and allows us to weight them differently as (i) they may not consider the same subsets of pairs, and (ii) may not capture the same geometric information. Another interpretation of such a decomposition in an imbalanced learning setting is the following: if z' is selected as belonging to the neighborhood of z, the minimization of the four terms of Equation (3) can be seen as a nice way to implicitly optimize with a kNN rule the true positive, false negative, false positive and true negative rates respectively.

3. Generalization bound for IML

In this section, we provide a theoretical analysis of our algorithm using the uniform stability framework [3]. This framework can be adapted to any metric learning algorithm A [2, 18] taking the following form:

$$\min_{\mathbf{M}\geq 0} G(\mathbf{M}) = \underbrace{\sum_{(z,z')\in\mathcal{S}^2} \ell_A(\mathbf{M}, z, z')}_{\widehat{R}(\mathbf{M})} + \lambda Reg(\mathbf{M}), \tag{4}$$

where $\widehat{R}(\mathbf{M})$ is the empirical loss of $\mathbf{M}$ on $\mathcal{S}$, and ℓ_A is any loss function that is q-Lipschitz[1] and (σ, p)-admissible[2]. To prove a uniform stability-based generalization bound the algorithm A has to be stable—meaning that its output does not change significantly under a small modification of $\mathcal{S}$— according to the following definition.

Definition 1 ([18] Eq. (5))**.** *A metric learning algorithm A has uniform stability in $\beta \geq 0$ w.r.t. the loss function ℓ_A if $\forall i \in \{1, \ldots, n\}$ the following holds*

$$\forall \mathcal{S} \in \mathcal{Z}^n,\ \sup_{z,z'} \left|\ell_A(\mathbf{M}, z, z') - \ell_A(\mathbf{M}^i, z, z')\right| \leq \beta,$$

where $\mathbf{M}$ is learned from $\mathcal{S}$, and $\mathbf{M}^i$ is learned from $\mathcal{S}^i$, the set obtained by replacing the i^{th} example in $\mathcal{S}$ by another also i.i.d. from $\mathcal{D}$.

If an algorithm has uniform stability, then it is possible to derive an upper bound on its generalization error using the McDiarmid inequality [3] recalled below.

Theorem 1 (McDiarmid Inequality, [3] Th. 2)**.** *Let $G : \mathcal{Z}^n \to \mathbb{R}$ be any function for which there exists constants c_i, $\forall i \in \{1, \ldots, n\}$ s.t. $\sup_{\mathcal{S}\in\mathcal{Z}^n, z'_i\in\mathcal{Z}} |G(\mathcal{S}) - G(\mathcal{S}^i)| \leq c_i$, then*

$$\forall \epsilon{>}0,\ \ \mathbb{P}_{\mathcal{S}}\left[\left|G(\mathcal{S}) - \mathbb{E}_{\mathcal{S}}[G(\mathcal{S})]\right| \geq \epsilon\right] \leq 2\exp\left(\frac{-2\epsilon^2}{\sum_{i=1}^n c_i^2}\right).$$

[1] A function f is q-Lipschitz *w.r.t.* its first argument if for any u, v, $|f(u) - f(v)| \leq q|u - v|$.

[2] A loss ℓ_A is (σ, p)-admissible *w.r.t.* its first argument $\mathbf{M}$ if it is convex in $\mathbf{M}$ and if $\forall z_1, z_2, z_3, z_4$, $|\ell_A(\mathbf{M}, z_1, z_2){-}\ell_A(\mathbf{M}, z_3, z_4)|{\leq}\sigma|y_{12}{-}y_{34}|{+}p$ with $y_{ij}{=}+1$ if $y_i{=}y_j$ and $y_{ij}{=}-1$ otherwise.

where $\mathbb{P}_{\mathcal{S}}$ denotes the probability with respect to the random draw of the sample $\mathcal{S}$ from $\mathcal{D}^n$. Then, one can derive the following theorem.

Theorem 2 ([2] Th. 8.11)**.** *Let $\mathcal{S}$ be a sample of n randomly selected training examples and $\mathbf{M}$ be the PSD matrix learned from an algorithm A with stability β. Assuming that the loss ℓ_A is q-Lipschitz and (σ, p)-admissible, with probability at least $1-\delta$ over the random choice of $\mathcal{S} \sim \mathcal{D}^n$, we have the following bound on the true risk $R(\mathbf{M})$*

$$R(\mathbf{M}) \leq \widehat{R}(\mathbf{M}) + 2\beta + \Big(2n\beta + 2(2\sigma + p)\Big)\sqrt{\frac{\ln 2/\delta}{2n}}.$$

Unlike standard PAC results [27], this kind of generalization bound has two advantages: *(i)* it takes into consideration properties of the algorithm, and *(ii)* it offers tools to deal with the fact that the pairs of examples are usually not drawn *i.i.d.* from $\mathcal{D} \times \mathcal{D}$ [2]. In the rest of this section, we first show that our loss is q-Lipschitz; then, we prove that our algorithm is stable, and finally, we derive a generalization bound on its true risk using the McDiarmid inequality. Note that in the following, we assume that the norm of any example is upper-bounded by a constant, *i.e.*, $\forall x \in \mathbb{R}^d, \|x\| \leq B$.

Lemma 1. *Let $\mathbf{M}$, $\mathbf{M}'$ be any matrices and (z, z') any pair of labeled examples, then the loss ℓ, as defined in Equation* (2)*, is q-Lipschitz w.r.t. its first argument, i.e., we have*

$$\Big|\ell(\mathbf{M}, z, z') - \ell(\mathbf{M}', z, z')\Big| \leq q\|\mathbf{M} - \mathbf{M}'\|_F,$$

with $q = 4B^2$.

Proof. See Appendix A. □

Let $\mathbf{M}$ be the matrix learned from $\mathcal{S}$ and Sim^+, Dis^+, Dis^- and Sim^- be the subsets of pairs coming from $\mathcal{S} \times \mathcal{S}$ as described in Section 2. The true and empirical losses are respectively defined as:

$$R(\mathbf{M}) = \mathop{\mathbb{E}}_{z\sim\mathcal{D}, z'\sim\mathcal{D}} \ell(\mathbf{M}, z, z')$$

$$\text{and } \widehat{R}(\mathbf{M}) = \frac{1}{n^2}\Bigg(\sum_{(z,z')\in Sim^+} a\ell_1(\mathbf{M}, z, z') + \sum_{(z,z')\in Sim^-} (1-a)\ell_1(\mathbf{M}, z, z') + \sum_{(z,z')\in Dis^+} b\ell_2(\mathbf{M}, z, z') + \sum_{(z,z')\in Dis^-} (1-b)\ell_2(\mathbf{M}, z, z')\Bigg).$$

where $\mathop{\mathbb{E}}_{z\sim\mathcal{D}, z'\sim\mathcal{D}}$ denotes the expectation with respect to the random draw of z and z' according to $\mathcal{D}$. Thus Problem (3) can be reformulated as:

$$\min_{\mathbf{M}\geq 0} F(\mathbf{M}) = \widehat{R}(\mathbf{M}) + \lambda\|\mathbf{M} - \mathbf{I}\|_F^2.$$

Uniform Stability of IML. We now proceed to show that our algorithm satisfies Definition 1. For that purpose, we introduce a lemma similar to Lemma 20 in [3] and Lemma 8.6 in [2].

Lemma 2. *Let matrices $\mathbf{M}^*$ and $\mathbf{M}^{*i}$ be the minimizers of F on $\mathcal{S}$ and $\mathcal{S}^i$ respectively. Let $\Delta\mathbf{M}^* = \mathbf{M}^{*i} - \mathbf{M}^*$ and $\rho = \frac{n^+}{n}$ the proportion of minority examples. Then for any $t \in [0, 1]$ we have*

$$\|\mathbf{M}^* - \mathbf{I}\|_F^2 - \|\mathbf{M}^* + t\Delta\mathbf{M}^* - \mathbf{I}\|_F^2 + \|\mathbf{M}^{*i} - \mathbf{I}\|_F^2 - \|\mathbf{M}^{*i} - t\Delta\mathbf{M}^* - \mathbf{I}\|_F^2 \leq \left(\frac{a(2\rho - 1) + 2(1-\rho)}{\lambda n}\right) 2qt\|\mathbf{M}^{*i} - \mathbf{M}^*\|_F.$$

Proof. See Appendix B. □

Note that the parameter b of Problem (3) does not appear in this Lemma. While in the experiments, in order to scale to large datasets, the pairs will be generated by using the k-neighborhood of the examples, we derived the proof in Appendix B by using all the pairs, that allowed us to get rid of the parameter b. This enables us to provide a more general result that does not depend on additional parameters (here the k of the k-nearest-neighbor rule).

Along with the q-Lipschitz property of Lemma 1, Lemma 2 allows us to prove that **IML** is stable.

Lemma 3. IML *has uniform stability with*

$$\beta = \frac{2q^2\Big(a(2\rho - 1) + 2(1-\rho)\Big)}{\lambda n}.$$

Proof. See Appendix C. □

Derivation of the main result. To prove our bound, we follow the derivation of Theorem 8.11 in [2]. We first provide two lemmas, and then we use them in conjunction with the McDiarmid inequality to derive a generalization bound.

First, we introduce a lemma that bounds the difference on the empirical risk over $\mathcal{S}$ and $\mathcal{S}^i$.

Lemma 4. *Let $\mathbf{M}^*$ be the optimal solution of Problem* (3)*. We have*

$$\Big|\widehat{R}(\mathbf{M}^*) - \widehat{R}^i(\mathbf{M}^*)\Big| \leq \frac{2\Big(a(2\rho - 1) + 1 - \rho\Big)4B^2\|\mathbf{M}^*\|_F + 2(1-\rho)(1+m)}{n}.$$

where $\widehat{R}$ and $\widehat{R}^i$ denote respectively the empirical risk over $\mathcal{S}$ and $\mathcal{S}^i$.

Proof. See Appendix D. □

Using Lemma 4 and the stability of Lemma 3, we introduce a lemma similar to Lemma 8.10 in [2].

Lemma 5. *Let matrices $\mathbf{M}^*$ and $\mathbf{M}^{*i}$ be the minimizers of F on $\mathcal{S}$ and $\mathcal{S}^i$ respectively. As* **IML** *has stability β, we have*

$$\Big|R(\mathbf{M}^*) - \widehat{R}(\mathbf{M}^*) - \Big(R(\mathbf{M}^{*i}) - \widehat{R}^i(\mathbf{M}^{*i})\Big)\Big| \leq \frac{2n\beta + D}{n} = c_i,$$

with
$D = 2\Big(a(2\rho - 1) + 1 - \rho\Big)4B^2\|\mathbf{M}^*\|_F + 2(1-\rho)(1+m)$.

Proof. See Appendix E. □

We are now able to state our main result.

Theorem 3 (Generalization bound for **IML**). *Let $\mathcal{S}$ be a sample of $n = n^+ + n^-$ randomly selected training examples with $\rho = \frac{n^+}{n}$ the proportion of minority examples and let $\mathbf{M}^*$ be the optimal solution learned from problem (3) having stability β. With probability at least $1-\delta$ over the random choice of $\mathcal{S} \sim \mathcal{D}^n$, we have*

$$R(\mathbf{M}^*) \leq \widehat{R}(\mathbf{M}^*) + 2\beta + (2n\beta + D)\sqrt{\frac{\ln 2/\delta}{2n}}$$

$$\text{with } \beta = \frac{2q^2\Big(a(2\rho-1) + 2(1-\rho)\Big)}{\lambda n}$$

$$\text{and } D = 2\Big(a(2\rho-1) + 1 - \rho\Big)4B^2\|\mathbf{M}^*\|_F + 2(1-\rho)(1+m)\,.$$

Proof. See Appendix F. □

Discussion. The difference between our Theorem 3 and classic bounds of the form of Theorem 2 is that proportion of minority examples $\rho = \frac{n^+}{n}$ and the weight of the similar minority pairs a appear in the two terms β and D. Classic metric learning bounds are derived in a balanced setting where $\rho = 0.5$ (*i.e.* positives and negatives are balanced) and where the parameters a and b are equal to 0.5. It is worth noting that plugging these values in our bound allows us to retrieve the constant $\beta = \frac{2q^2}{\lambda n}$ as derived in Theorem 8.7 in [2]. This means that our β formulation is a generalization of the standard stability constants in Metric Learning. Regarding the term D, the decomposition into four terms allows us to get a tighter bound by a factor 4 with $D = 4B^2\|\mathbf{M}\|_F + (1+m)$ while $D = 16B^2\|\mathbf{M}\|_F + 4(1+m)$ in [2].

Another interesting interpretation of our bound is that when ρ tends to 0, *i.e.*, the dataset is more and more imbalanced, a classic metric learning method (with $a = b = 0.5$) will converge slower. Indeed, in such a situation, we get a stability constant β which would tend to $\frac{3q^2}{\lambda n} > \frac{2q^2}{\lambda n}$ while by parameterizing by a, we have $\frac{2q^2(-a+2)}{\lambda n}$. In this case, a value of a close to 1 allows us to reduce the negative effect of the imbalance.

4. Experiments

Datasets. In this section, we provide an empirical study of **IML** on 22 datasets. They come mainly from the UCI[3] and Keel[4] repositories. The 'splice' dataset comes from LIBSVM[5]. All datasets are normalized such that each feature has a mean of 0 and a variance of 1.

For the sake of simplicity, we have chosen binary datasets, described in Table 1 where the minority class is given by the columns "Label". Note that **IML** can easily be generalized to multiclass problems by learning one metric per class in a standard "one-versus-all" strategy, and then applying a majority vote.

[3] https://archive.ics.uci.edu/ml/datasets.html

[4] http://sci2s.ugr.es/keel/datasets.php

[5] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html#splice

Optimization Details. Like most Mahalanobis metric learning algorithms, **IML** requires that the learned matrix $\mathbf{M}$ is PSD. There exist different methods to enforce the PSD constraint [19]. A classic solution consists in performing a Projected Gradient Descent where one alternates a gradient descent step and a (costly) projection onto the cone of PSD matrices. The advantage is that the problem remains convex *w.r.t.* $\mathbf{M}$ [34], ensuring that one will attain the optimal solution of the problem by correctly setting the projection step in the gradient descent. Another solution [32] is based on the fact that if $\mathbf{M}$ is PSD, it can be rewritten as $\mathbf{M} = \mathbf{L}^\top\mathbf{L}$. Therefore, instead of learning $\mathbf{M}$, one can enforce $\mathbf{M}$ to be PSD in a cheaper way by directly learning the projection matrix $\mathbf{L} \in \mathbb{R}^{r\times d}$ (where r is the rank of $\mathbf{M}$). This can be done thanks to a gradient descent by computing the gradient of the problem *w.r.t.* $\mathbf{L}$ (instead of $\mathbf{M}$). The implementation[6] we propose is based on this latter approach [32] where we make use of the L-BFGS-B algorithm [38] from the SciPy Python library to optimize our problem: it takes as input our initial point (the identity matrix), the optimization problem of Equation (3), and its gradient, then it performs a gradient descent that returns the projection matrix $\mathbf{L}$ minimizing Problem (3). To prevent us from tuning r and finding the best r-dimensional projection space, we set $r = d$ in the experiments.

As discussed at the end of Section 2, the pairs of examples considered by **IML** in its four terms are chosen using the nearest neighbors rule. Indeed, we noted experimentally that the algorithms using this strategy (**LMNN** [31], **IMLS** [29] and **IML**) perform better than the ones using a random selection strategy (**ITML** [8] and **GMML** [35]).

Experimental setup. All along our experiments, we use a 3NN classifier after projection of the training and test data using the metric learned. The metrics considered in the comparative study are the Euclidean distance and the ones learned by **LMNN** [31], **ITML** [8], **GMML** [35], **IMLS** [29] and **IML**. For each dataset, we generate randomly 20 stratified splits of 70% training examples and 30% test data (same class proportions in training and test) and report the mean results over the 20 splits. The parameters are tuned by 5-fold cross-validation on the training set through a grid search using the following parameter ranges: for **LMNN** and **IMLS**, $\mu \in \{0, 0.05, \ldots, 1\}$ (k is fixed to 3); for **ITML**, $\gamma \in \{2^{-10:10}\}$; for **GMML** $t \in \{0, 0.05, \ldots, 1\}$; and for **IML** we fix $a = b = \frac{n^-}{n}$ and we tune $m \in \{1, 10, 100, 1000, 10000\}$ and $\lambda \in \{0, 0.01, 0.1, 1, 10\}$ (k is also fixed to 3). Note that to study the impact of the hyperparameters a and b and the impact of how the pairs of examples were selected, we performed an experiment where we artificially decreased the proportion of minority examples in the datasets. The results of this experiment are given in Appendix G.

The results of the first experiment are reported in Table 2. On average, the F1-measure of 72.3% obtained by **IML** is the best in comparison to 70.8% for **LMNN** and **IMLS**, 70.1% for **ITML**, 69.3% for **GMML** and 67.3% for the Euclidean distance. Overall, **IML** has also the best average rank of 1.52.

[6] We will make the source code available online in case of acceptance.

Table 1. Description of the datasets (n: number of examples, d: number of features, c: number of classes) and the class chosen as positive (Label), its cardinality (n^+) and its percentage (%).

Name	n	d	c	Label	n^+	%	Name	n	d	c	Label	n^+	%
splice	3175	60	2	-1	1527	48.10%	glass	214	11	6	1	70	32.71%
sonar	208	60	2	R	97	46.64%	newthyroid	215	5	3	2, 3	65	30.23%
balance	625	4	3	L	288	46.08%	german	1000	23	2	2	300	30.00%
australian	690	14	2	1	307	44.49%	vehicle	846	18	4	van	199	23.52%
heart	270	13	2	2	120	44.44%	spectfheart	267	44	2	0	55	20.60%
bupa	345	6	2	1	145	42.03%	hayes	160	4	3	3	31	19.38%
spambase	4597	57	2	1	1812	39.42%	segmentation	2310	19	7	window	330	14.29%
wdbc	569	30	2	M	212	37.26%	abalone	4177	10	28	8	568	13.60%
iono	351	34	2	b	126	35.90%	yeast	1484	8	10	ME3	163	10.98%
pima	768	8	2	1	268	34.90%	libras	360	90	15	1	24	6.66%
wine	178	13	3	1	59	33.15%	pageblocks	5473	10	5	3, 4, 5	231	4.22%

Table 2. Average F1-measure ± standard deviation over 20 splits using different metric learning algorithms.

Dataset	Euclidean	LMNN	ITML	GMML	IMLS	IML
hayes	44.9 ± 13.2	**57.2** ± 12.5	55.4 ± 8.7	52.7 ± 10.8	57.2 ± 12.5	54.9 ± 9.2
wine	94.9 ± 2.2	96.0 ± 2.9	96.3 ± 3.3	95.3 ± 3.1	96.0 ± 2.9	**96.6** ± 2.1
sonar	69.2 ± 5.3	70.6 ± 6.5	70.6 ± 5.9	69.1 ± 5.0	71.1 ± 6.7	**74.6** ± 3.7
glass	66.0 ± 3.4	63.6 ± 5.2	62.6 ± 5.2	**67.2** ± 3.6	63.6 ± 5.2	66.6 ± 4.3
newthyroid	83.4 ± 4.2	88.1 ± 5.2	89.8 ± 5.2	91.1 ± 2.5	88.1 ± 5.2	**91.3** ± 2.6
spectfheart	34.8 ± 12.3	39.1 ± 8.4	34.4 ± 7.9	29.1 ± 11.4	38.6 ± 8.7	**42.4** ± 8.7
heart	76.8 ± 2.1	74.8 ± 3.2	76.8 ± 2.9	76.9 ± 3.6	74.6 ± 3.1	**77.1** ± 3.1
bupa	49.8 ± 4.4	50.1 ± 5.0	51.3 ± 4.8	52.0 ± 5.3	50.1 ± 5.0	**52.5** ± 5.1
iono	67.8 ± 6.7	70.8 ± 3.9	73.4 ± 5.4	72.0 ± 5.4	71.0 ± 4.0	**76.1** ± 2.9
libras	48.4 ± 15.1	**68.3** ± 12.2	65.5 ± 15.3	56.1 ± 16.3	66.6 ± 10.3	67.9 ± 12.1
wdbc	94.2 ± 1.3	93.5 ± 1.7	94.3 ± 1.1	94.4 ± 1.3	93.4 ± 2.2	**95.2** ± 1.1
balance	87.4 ± 1.8	89.8 ± 1.3	**93.0** ± 1.4	90.3 ± 1.3	89.8 ± 1.3	90.6 ± 1.2
australian	79.9 ± 1.7	81.7 ± 2.0	**82.0** ± 1.9	81.0 ± 2.6	81.4 ± 2.0	81.9 ± 1.8
pima	56.2 ± 1.9	55.9 ± 3.3	**57.5** ± 3.0	56.7 ± 3.0	55.9 ± 3.3	57.2 ± 2.7
vehicle	80.5 ± 2.4	**92.6** ± 1.0	90.2 ± 2.4	90.1 ± 1.7	92.5 ± 1.2	91.8 ± 1.9
german	35.3 ± 2.8	37.3 ± 3.9	37.4 ± 3.3	37.1 ± 3.3	37.8 ± 3.7	**38.4** ± 3.5
yeast	73.2 ± 2.3	74.9 ± 2.8	74.2 ± 3.1	73.5 ± 2.6	74.5 ± 2.7	**75.4** ± 2.4
segmentation	81.8 ± 2.4	85.3 ± 2.1	79.6 ± 3.0	80.8 ± 3.1	85.6 ± 2.3	**86.0** ± 2.5
splice	76.3 ± 0.7	86.5 ± 0.8	79.7 ± 1.4	76.3 ± 1.3	**88.0** ± 0.9	87.4 ± 0.6
abalone	22.6 ± 2.1	22.1 ± 2.1	21.2 ± 3.0	21.6 ± 1.7	22.1 ± 2.1	**23.0** ± 1.9
spambase	85.3 ± 0.9	88.4 ± 0.8	87.8 ± 1.0	86.8 ± 0.8	88.7 ± 0.5	**89.3** ± 0.8
pageblocks	71.9 ± 3.0	71.8 ± 3.2	69.7 ± 5.1	**73.7** ± 2.9	71.8 ± 3.1	73.4 ± 2.6
Mean	67.3 ± 4.2	70.8 ± 4.1	70.1 ± 4.3	69.3 ± 4.2	70.8 ± 4.0	**72.3** ± 3.5
Average Rank	5.00	3.57	3.57	4.00	3.35	**1.52**

Table 3. Same experiment as in Table 2 after having applied the SMOTE algorithm [6] until $n^+ = n^-$.

Dataset	Euclidean	LMNN	ITML	GMML	IMLS	IML
hayes	68.0 ± 6.8	64.4 ± 7.6	67.8 ± 7.8	**69.5** ± 7.2	64.2 ± 7.6	68.6 ± 7.2
wine	92.7 ± 2.8	95.3 ± 3.0	96.3 ± 2.6	94.4 ± 3.2	95.4 ± 2.9	**96.7** ± 2.2
sonar	72.6 ± 4.2	73.0 ± 6.4	72.6 ± 4.5	71.2 ± 4.2	72.7 ± 6.0	**75.2** ± 4.8
glass	66.6 ± 2.9	66.1 ± 4.0	64.6 ± 3.2	**67.4** ± 3.8	66.1 ± 4.2	66.2 ± 3.5
newthyroid	87.6 ± 3.5	88.7 ± 4.0	**91.6** ± 3.2	89.9 ± 4.3	88.7 ± 4.0	90.7 ± 2.2
spectfheart	47.4 ± 2.3	41.9 ± 8.5	46.7 ± 6.9	**49.1** ± 4.4	40.9 ± 7.3	46.1 ± 7.8
heart	**77.3** ± 2.0	75.0 ± 2.6	75.7 ± 4.1	77.2 ± 3.9	74.4 ± 3.0	76.9 ± 2.3
bupa	54.1 ± 3.1	55.4 ± 3.4	53.9 ± 3.7	**55.9** ± 4.1	55.4 ± 3.4	54.8 ± 3.5
iono	78.4 ± 2.6	77.7 ± 3.7	77.5 ± 3.8	78.2 ± 3.9	76.9 ± 3.9	**79.9** ± 3.9
libras	68.3 ± 8.1	76.7 ± 8.5	69.7 ± 13.8	69.2 ± 11.0	69.0 ± 14.7	**78.1** ± 9.4
wdbc	93.4 ± 1.3	93.5 ± 2.2	94.0 ± 1.4	93.6 ± 1.5	93.0 ± 2.2	**94.8** ± 1.3
balance	87.4 ± 1.9	89.6 ± 1.5	**92.1** ± 1.4	89.8 ± 1.9	89.5 ± 1.5	90.5 ± 1.4
australian	80.3 ± 1.6	80.7 ± 3.2	**82.4** ± 1.5	81.1 ± 1.8	81.1 ± 3.2	82.1 ± 1.6
pima	60.1 ± 2.6	60.1 ± 2.1	60.8 ± 2.2	60.3 ± 2.8	60.1 ± 2.1	**61.2** ± 2.0
vehicle	80.6 ± 2.1	92.0 ± 1.6	89.9 ± 3.1	89.5 ± 2.1	**92.3** ± 1.5	91.1 ± 1.4
german	46.3 ± 2.2	45.4 ± 3.5	46.4 ± 1.8	46.0 ± 2.3	45.1 ± 3.4	**47.1** ± 2.0
yeast	65.9 ± 2.9	67.1 ± 3.7	**70.4** ± 2.8	68.4 ± 2.3	68.2 ± 3.7	70.4 ± 3.0
segmentation	82.0 ± 1.9	83.8 ± 2.9	81.6 ± 2.2	81.5 ± 1.8	**84.8** ± 3.2	84.8 ± 2.2
splice	74.9 ± 0.9	86.3 ± 0.8	79.6 ± 1.2	76.4 ± 1.3	**87.9** ± 1.1	87.2 ± 0.6
abalone	**32.3** ± 0.7	31.4 ± 1.7	31.9 ± 0.8	31.7 ± 1.1	31.5 ± 1.2	32.3 ± 1.0
spambase	85.9 ± 0.7	88.5 ± 0.6	87.4 ± 0.9	87.2 ± 0.8	88.8 ± 0.8	**89.4** ± 0.8
pageblocks	62.0 ± 2.9	61.5 ± 4.1	55.5 ± 4.0	61.5 ± 3.5	61.0 ± 4.1	**62.5** ± 3.5
Mean	71.1 ± 2.7	72.5 ± 3.6	72.2 ± 3.5	72.2 ± 3.3	72.1 ± 3.9	**73.9** ± 3.1
Average Rank	4.26	3.91	3.39	3.39	4.17	**1.87**

We note that **IML** generally gives better performances on the datasets considered no matter how much they are balanced or not. This means that our reweighting scheme of the pairs can not only improve the performances in an imbalanced setting but is also competitive in more classic scenarios.

Second experiment. To address imbalanced data issues, classic machine learning algorithms typically resort to over/under-sampling techniques [1] or create synthetic samples in the neighborhood of the minority class—*e.g.*, using SMOTE-like strategies [6]. We now aim at studying the behavior of those methods when used as a pre-process of the metric learning procedures. We consider the results of Table 2 as baselines. We compare them to the performances obtained after performing prior to metric learning an over-sampling using SMOTE and a Random Under Sampling (RUS) strategy of the negative data. We use the implementations of these methods from the Python library imbalanced-learn [21].

The results obtained are reported in Table 3 for SMOTE and in Table 4 for RUS. Note that the results from Tables 2, 3 and 4 where computed using the same training/test splits and the same validation folds and are thus comparable. In each of the three settings considered, **IML** obtains the best results showing that it is more appropriate for improving the F1-measure. We also note that SMOTE allows one to increase significantly the performances of all methods, while there is no gain with RUS in comparison with an approach without sampling.

This increase of performance tells us that SMOTE and **IML** are more complementary than competitors with different objectives (re-balancing for the former and representation learning for the latter).

5. Conclusion and perspectives

In this paper, we propose to revisit the classic formulation of metric learning algorithms that learn a Mahalanobis metric in the light of imbalanced data issues. Unlike the state of the art methods that do not make any distinction between the labels of similar pairs, we propose to decompose the usual loss with respect to the different possible labels involved in the pairs. This decomposition allows us to give specific weights to each type of pairs in order to improve the performance on the minority

Table 4. Same experiment as in Table 2 after having applied a Random Under Sampling of the negative examples until $n^- = n^+$.

Dataset	Euclidean	LMNN	ITML	GMML	IMLS	IML
hayes	63.4 ± 9.0	**67.7** ± 7.4	64.7 ± 6.7	66.4 ± 8.1	67.7 ± 7.4	66.0 ± 7.6
wine	91.2 ± 2.6	94.1 ± 2.8	**94.2** ± 3.8	92.7 ± 3.7	94.1 ± 2.9	93.9 ± 3.2
sonar	70.4 ± 5.2	73.1 ± 6.3	70.2 ± 5.7	69.9 ± 5.5	71.1 ± 9.2	**74.4** ± 4.8
glass	64.6 ± 3.5	63.2 ± 4.5	61.1 ± 4.9	**64.6** ± 3.1	62.7 ± 5.0	64.5 ± 4.7
newthyroid	86.6 ± 4.6	91.4 ± 5.0	91.1 ± 4.9	90.6 ± 3.3	91.4 ± 5.0	**92.3** ± 2.6
spectfheart	44.2 ± 3.9	42.6 ± 8.0	46.7 ± 4.6	45.9 ± 5.6	42.6 ± 8.0	**48.8** ± 6.3
heart	**77.4** ± 2.0	75.8 ± 3.3	76.6 ± 2.5	77.3 ± 1.9	75.5 ± 3.3	77.1 ± 2.1
bupa	53.8 ± 4.1	54.1 ± 4.8	54.6 ± 4.1	**55.7** ± 3.6	54.1 ± 4.8	55.0 ± 3.2
iono	73.1 ± 5.2	73.3 ± 4.1	75.4 ± 3.4	74.7 ± 3.2	73.3 ± 4.1	**77.3** ± 3.0
libras	34.3 ± 10.6	35.6 ± 10.9	38.2 ± 12.2	36.4 ± 12.6	35.6 ± 10.9	**39.3** ± 13.8
wdbc	93.7 ± 1.2	92.9 ± 1.6	93.6 ± 1.8	93.2 ± 2.1	92.3 ± 2.5	**94.7** ± 1.4
balance	87.5 ± 1.5	90.1 ± 1.3	**92.8** ± 1.5	90.1 ± 1.7	90.2 ± 1.3	90.7 ± 1.4
australian	80.4 ± 1.7	81.7 ± 2.2	82.2 ± 1.6	81.5 ± 2.3	81.7 ± 2.2	**82.5** ± 2.2
pima	60.8 ± 2.7	60.5 ± 2.1	**62.2** ± 1.7	60.8 ± 2.4	60.4 ± 2.1	61.2 ± 2.5
vehicle	74.0 ± 3.1	89.7 ± 1.6	87.7 ± 2.6	85.5 ± 3.1	**89.7** ± 2.0	89.3 ± 1.9
german	46.7 ± 1.6	46.9 ± 2.5	47.3 ± 2.3	47.5 ± 1.6	46.2 ± 2.0	**48.0** ± 1.8
yeast	57.2 ± 4.5	60.8 ± 4.6	60.9 ± 3.8	59.7 ± 3.8	61.2 ± 4.9	**61.9** ± 3.9
segmentation	64.6 ± 3.1	70.4 ± 2.4	65.7 ± 3.4	64.3 ± 2.9	72.4 ± 2.9	**74.2** ± 1.8
splice	75.9 ± 0.7	86.5 ± 0.6	79.5 ± 1.6	76.2 ± 1.2	**87.9** ± 0.9	87.2 ± 0.6
abalone	**32.8** ± 1.1	32.5 ± 1.4	32.5 ± 1.3	31.6 ± 1.0	32.2 ± 1.3	32.6 ± 1.4
spambase	85.0 ± 1.0	88.1 ± 1.1	86.8 ± 1.2	86.2 ± 1.2	88.4 ± 0.7	**88.7** ± 0.8
pageblocks	46.8 ± 3.7	**50.2** ± 4.8	43.0 ± 5.2	48.3 ± 4.5	49.6 ± 5.6	49.1 ± 4.0
Mean	66.6 ± 3.5	69.1 ± 3.8	68.5 ± 3.7	68.1 ± 3.6	69.1 ± 4.0	**70.4** ± 3.4
Average Rank	4.83	3.65	3.30	3.96	3.43	**1.83**

class. We derive a generalization bound specific to the imbalanced setting showing a convergence term depending on the class imbalance and illustrating the hardness of learning from imbalanced data. Our experimental evaluation shows that we are able to outperform state of the art metric learning algorithms in terms of F1-measure over balanced and imbalanced datasets.

We believe that our work gives rise to exciting perspectives when facing imbalanced data. Among them, we want to study how our algorithm could be adapted to learn non-linear metrics. From an algorithmic point of view, we would like to extend our method by deriving a closed form solution in a similar way as done by Zadeh et al. [35] to drastically reduce the computation time while maintaining good performances.

References

[1] Aggarwal, C., 2013. Outlier Analysis. Springer.

[2] Bellet, A., Habrard, A., Sebban, M., 2015. Metric learning. Synthesis Lectures on Artificial Intelligence and Machine Learning 9.

[3] Bousquet, O., Elisseeff, A., 2002. Stability and generalization. JMLR 2.

[4] Branco, P., Torgo, L., Ribeiro, R., 2016. A survey of predictive modeling on imbalanced domains. ACM Computing Surveys (CSUR) 49, 31.

[5] Cao, Q., Guo, Z., Ying, Y., 2016. Generalization bounds for metric and similarity learning. Machine Learning 102, 115–132.

[6] Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, P., 2002. Smote: Synthetic minority over-sampling technique. J. Artif. Int. Res. 16, 321–357.

[7] Chawla, N., Lazarevic, A., Hall, L., Bowyer, K., 2003. Smoteboost: Improving prediction of the minority class in boosting, in: European conference on principles of data mining and knowledge discovery, Springer. pp. 107–119.

[8] Davis, J., Kulis, B., Jain, P., Sra, S., Dhillon, I., 2007. Information-theoretic metric learning, in: ICML.

[9] Douzas, G., Bacao, F., 2018. Effective data generation for imbalanced learning using conditional generative adversarial networks. Expert Systems with applications 91, 464–471.

[10] Drummond, C., Holte, R., 2003. C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling, in: Workshop on learning from imbalanced datasets II, Citeseer. pp. 1–8.

[11] Elkan, C., 2001. The foundations of cost-sensitive learning, in: International joint conference on artificial intelligence, Lawrence Erlbaum Associates Ltd. pp. 973–978.

[12] Estabrooks, A., Jo, T., Japkowicz, N., 2004. A multiple resampling method for learning from imbalanced data sets. Computational intelligence 20, 18–36.

[13] Feng, L., Wang, H., Jin, B., Li, H., Xue, M., Wang, L., 2018. Learning a distance metric by balancing kl-divergence for imbalanced datasets. IEEE Transactions on Systems, Man, and Cybernetics: Systems .

[14] Frery, J., Habrard, A., Sebban, M., Caelen, O., He-Guelton, L., 2017. Efficient top rank optimization with gradient boosting for supervised anomaly detection, in: ECML-PKDD.

[15] Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., Herrera, F., 2012. A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. IEEE Transactions on Systems, Man, and Cybernetics 42, 463–484.

[16] Han, H., Wang, W., Mao, B., 2005. Borderline-smote: a new over-sampling method in imbalanced data sets learning, in: International Conference on Intelligent Computing, Springer. pp. 878–887.

[17] He, H., Garcia, E., 2009. Learning from imbalanced data. IEEE TKDE 21.

[18] Jin, R., Wang, S., Zhou, Y., 2009. Regularized distance metric learning: Theory and algorithm, in: NIPS.

[19] Kulis, B., 2013. Metric learning: A survey. Foundations and Trends in ML 5, 287–364.

[20] Lee, J., Jin, R., Jain, A., 2008. Rank-based distance metric learning: An application to image retrieval, in: CVPR.

[21] Lemaître, G., Nogueira, F., Aridas, C.K., 2017. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. The Journal of Machine Learning Research 18, 559–563.

[22] Liu, K., Han, J., Chen, H., Yan, H., Yang, P., 2019. Defect detection on el images based on deep feature optimized by metric learning for imbalanced data, in: 2019 25th International Conference on Automation and Computing (ICAC), IEEE. pp. 1–5.

[23] Liu, X., Wu, J., Zhou, Z., 2009. Exploratory undersampling for class-imbalance learning. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 39, 539–550.

[24] Lu, J., Zhou, X., Tan, Y., Shang, Y., Zhou, J., 2014. Neighborhood repulsed metric learning for kinship verification. IEEE transactions on pattern analysis and machine intelligence 36, 331–345.

[25] McFee, B., Lanckriet, G., 2010. Metric learning to rank, in: ICML.

[26] Schultz, M., Joachims, T., 2004. Learning a distance metric from relative comparisons, in: NIPS.

[27] Valiant, L.G., 1984. A theory of the learnable, in: Proceedings of the sixteenth annual ACM symposium on Theory of computing, ACM. pp. 436–445.

[28] Vogel, R., Bellet, A., Clémençon, S., 2018. A probabilistic theory of supervised similarity learning for pointwise roc curve optimization. ICML .

[29] Wang, N., Zhao, X., Jiang, Y., Gao, Y., 2018. Iterative metric learning for imbalance data classification., in: IJCAI.

[30] Wang, Y., Gan, W., Yang, J., Wu, W., Yan, J., 2019. Dynamic curriculum learning for imbalanced data classification, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 5017–5026.

[31] Weinberger, K., Saul, L., 2009. Distance metric learning for large margin nearest neighbor classification. JMLR 10, 207–244.

[32] Weinberger, K.Q., Saul, L.K., 2008. Fast solvers and efficient implementations for distance metric learning, in: Proceedings of the 25th international conference on Machine learning, ACM. pp. 1160–1167.

[33] Xiang, S., Nie, F., Zhang, C., 2008. Learning a mahalanobis distance metric for data clustering and classification. Pattern Recognition 41, 3600–3612.

[34] Xing, E., Jordan, M., Russell, S., Ng, A., 2003. Distance metric learning with application to clustering with side-information, in: NIPS.

[35] Zadeh, P., Hosseini, R., Sra, S., 2016. Geometric mean metric learning, in: ICML.

[36] Zadrozny, B., Langford, J., Abe, N., 2003. Cost-sensitive learning by cost-proportionate example weighting, in: ICDM, IEEE.

[37] Zheng, W., Gong, S., Xiang, T., 2011. Person re-identification by probabilistic relative distance comparison .

[38] Zhu, C., Byrd, R., Lu, P., Nocedal, J., 1997. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. ACM TOMS 23, 550–560.