ALGORITHMIC ROBUSTNESS FOR SEMI-SUPERVISED $(\epsilon, \gamma, \tau)$-GOOD METRIC LEARNING

Maria-Irina Nicolae, Marc Sebban & Amaury Habrard
Hubert Curien Laboratory
Jean Monnet University, Saint-Etienne, France
{Maria.Irina.Nicolae,Marc.Sebban,Amaury.Habrard}@univ-st-etienne.fr

Éric Gaussier & Massih-Reza Amini
Laboratoire d’Informatique de Grenoble
Joseph Fourier University, Grenoble, France
{Eric.Gaussier,Massih-Reza.Amini}@imag.fr

ABSTRACT

The importance of metrics in machine learning has attracted a growing interest for distance and similarity learning, and especially the Mahalanobis distance. However, it is worth noting that this research field lacks theoretical guarantees that can be expected on the generalization capacity of the classifier associated to a learned metric. The theoretical framework of $(\epsilon, \gamma, \tau)$-good similarity functions (Balcan et al., 2008) has been one of the first attempts to draw a link between the properties of a similarity function and those of a linear classifier making use of it. In this paper, we extend this theory to a method where the metric and the separator are jointly learned in a semi-supervised way, setting that has not been explored before. We furthermore provide a generalization bound for the associated classifier based on the algorithmic robustness framework. The behavior of our method is illustrated via some experimental results.

1 INTRODUCTION

Many researchers have used the underlying geometry of the data to improve classification algorithms, e.g. by learning Mahalanobis distances instead of the standard Euclidean distance, thus paving the way for a new research area termed metric learning (Bellet et al., 2013). Most of these studies have based their approaches on distance learning (Baoli et al., 2004; Davis et al., 2007; Diligenti et al., 2003; Shalev-Shwartz et al., 2004; Weinberger & Saul, 2009), even though similarity learning has also attracted a growing interest (Bao et al., 2003; Grabowski & Szalas, 2005; Hust (2004), Qamar & Gaussier (2009)), the rationale being that the cosine similarity should in some cases be preferred over the Euclidean distance. More recently, Balcan et al. (2008) have proposed the first framework that allows one to relate similarities with a classification algorithm making use of them. This general framework, that can be used with any bounded similarity function (potentially derived from a distance), provides generalization guarantees on a linear classifier learned from the similarity. Their algorithm, whose formulation is equivalent to a relaxed $L_1$-norm SVM (Zhu et al., 2003), does not enforce the positive definiteness constraint of the similarity. However, to enjoy such generalization guarantees, the similarity function is assumed to be known beforehand and to satisfy $(\epsilon, \gamma, \tau)$-goodness properties. Unfortunately, Balcan et al. (2008) do not provide any algorithm for learning such similarities.

In order to overcome these limitations, Bellet et al. (2012) have explored the possibility of independently learning an $(\epsilon, \gamma, \tau)$-good similarity that they plug into the initial algorithm (Balcan et al., 2008) to learn the linear separator. Generalization bounds for the learned similarity (a bilinear similarity) were derived via uniform stability arguments (Bousquet & Elisseeff, 2002). However, despite good results in practice, one limitation of this framework is that it imposes to deal with strongly convex objective functions. More importantly, the similarity learning step is done in a completely supervised way while Balcan et al. (2008)’s framework opens the door to the use of unlabeled data.
In this paper, we aim at better exploiting the semi-supervised setting underlying the theoretical framework of Balcan et al. [2008], which is based on similarities between labeled data and unlabeled reasonable points (roughly speaking, the reasonable points play the same role as that of support vectors in SVMs). Furthermore, and unlike Bellet et al. [2012], we propose here to jointly learn the metric and the classifier, so that both the metric and the separator are learned in a semi-supervised way. To our knowledge, this approach has not been explored before in the context of semi-supervised metric learning. Enforcing $(\epsilon, \gamma, \tau)$-goodness allows us to preserve Balcan et al. [2008]'s theoretical guarantees. Lastly, proving the algorithmic robustness (Xu & Mannor, 2010; 2012) of our method leads to consistency bounds for different kinds of similarity functions.

The remainder of this paper is organized as follows: Section 2 reviews some previous results in metric and similarity learning and presents the theory of $(\epsilon, \gamma, \tau)$-good similarities. Section 3 introduces our method that jointly learns the metric and the linear classifier, followed by generalization guarantees for our formulation. We show how to integrate different similarity functions in our setting. Finally, Section 4 features an experimental study on various standard datasets.

2 Notations and Related Work

We denote vectors by lower-case bold symbols ($x$) and matrices by upper-case bold symbols ($A$). Consider the following learning problem: we are given access to labeled examples $z = (x, l(x))$ drawn from some unknown distribution $P$ over $X \times Y$, where $X \subseteq \mathbb{R}^d$ and $Y = \{-1, 1\}$ are respectively the instance and the output spaces. A pairwise similarity function over $X$ is defined as $K : X \times X \rightarrow [-1, 1]$, and the hinge loss as $[c]_+ = \max(0, 1 - c)$. We denote the $L_1$ norm by $\| \cdot \|_1$, the $L_2$ norm by $\| \cdot \|_2$ and the Frobenius norm by $\| \cdot \|_F$.

Metric learning aims at finding the parameters of a distance or similarity function that best satisfy the underlying geometry of the data. This information is usually expressed as pair-based ($x$ and $x'$ should be (dis)similar) or triplet-based constraints ($x$ should be more similar to $x'$ than to $x''$). Typically, the learned metric takes the form of a matrix and is the result of solving an optimization problem.

The approaches that have received the most attention in this field involve learning a Mahalanobis distance, defined as $d_A(x, x') = \sqrt{(x - x')^T A (x - x')}$. The distance is parameterized by the symmetric and positive semi-definite (PSD) matrix $A \in \mathbb{R}^{d \times d}$. This method has attracted a lot of interest due to its interpretability: the Mahalanobis distance implicitly corresponds to computing the Euclidean distance after linearly projecting the data to a different (possibly lower) feature space. The PSD constraint on $A$ ensures $d_A$ is a proper metric. Note that setting $A$ as the identity matrix gives the Euclidean distance. In this context, Large Margin Nearest Neighbors (LMNN) (Weinberger & Saul, 2008; 2009) is one of the most widely-used Mahalanobis distance learning methods. The constraints they use are pair- and triplet-based, derived from each instance’s nearest neighbors. The optimization problem they solve is convex and has a special-purpose solver. The algorithm works well in practice, but is sometimes prone to overfitting due to the absence of regularization, especially when dealing with high dimensional data. Another limitation is that enforcing the PSD constraint on $A$ is computationally expensive. One can partly get around this latter shortcoming by making use of specific solvers or using information-theoretic approaches, such as ITML (Davis et al., 2007). This work was the first one to use LogDet divergence for regularization, and thus provides an easy and cheap way for ensuring that $A$ is a PSD matrix. However, the learned metric $A$ strongly depends on the initial value $A_0$, which is an important shortcoming, as $A_0$ is handpicked.

More generally, Mahalanobis distance learning faces two main limitations. The first one is that enforcing the PSD and symmetry constraints on $A$, beyond the cost it induces, often rules out natural similarity functions for the problem at hand. Secondly, although one can experimentally notice that the Mahalanobis distance learning methods give better accuracy than using the Euclidean distance, no theoretical guarantees are provided to establish a link between the quality of the metric and the behavior of the classifier that makes use of it. In this context, Balcan et al. [2008] introduced a theory for learning with so-called $(\epsilon, \gamma, \tau)$-good similarity functions based on non PSD matrices. This was the first stone to establish generalization guarantees for a linear classifier that would be learned by making use of such similarities. They derived all their results based on the following definition of a good metric.
Definition 1. \cite{Balcan:2008} \(K\) is a \((\epsilon, \gamma, \tau)\)-good similarity function in hinge loss for a learning problem \(P\) if there exists a random indicator function \(R(x)\) defining a probabilistic set of "reasonable points" such that the following conditions hold:

1. We have
   \[
   \Pr_{(x, l(x)) \sim P} \left[1 - l(x)g(x) + \gamma \right] \leq \epsilon, \tag{1}
   \]
   where \(g(x) = \mathbb{E}_{(x', l(x'), R(x'))} [l(x')K(x, x')|R(x')]\).

2. \(\Pr_{x} (R(x')) \geq \tau\).

This definition imposes a constraint on the mass of reasonable points one must consider (greater than \(\tau\)). It also expresses the tolerated margin violations in an averaged way: a \((1 - \epsilon)\) proportion of examples \(x\) are on average \(2\gamma\) more similar to random reasonable examples \(x'\) of their own label than to random reasonable examples \(x'\) of the other label. This allows for more flexibility than pair- or triplet-based constraints. Notice that no constraint is imposed on the form of the similarity function. Definition 1 can then be used to learn well:

Theorem 1. \cite{Balcan:2008} Let \(K\) be an \((\epsilon, \gamma, \tau)\)-good similarity function in hinge loss for a learning problem \(P\). For any \(\epsilon > 0\) and \(\delta < \gamma\epsilon/4\) let \(S = \{x_1', x_2', \ldots, x_d'\}\) be a sample of landmarks drawn from \(P\). Consider the mapping \(\phi^S : \mathcal{X} \rightarrow \mathbb{R}^{d_u}\) defined as follows: \(\phi^S(x) = K(x, x_i')\), \(i \in \{1, \ldots, d_u\}\). With probability \(1 - \delta\) over the random sample \(S\), the induced distribution \(\phi^S(P)\) in \(\mathbb{R}^{d_u}\), has a separator achieving hinge loss at most \(\epsilon + \epsilon_1\) at margin \(\gamma\).

In other words, if \(K\) is \((\epsilon, \gamma, \tau)\)-good according to Definition 1 and enough points are available, there exists a linear separator \(\alpha\) with error arbitrarily close to \(\epsilon\) in the space \(\phi^S\). The procedure for finding the separator involves two steps: first using \(d_u\) potentially unlabeled examples as landmarks to construct the feature space, then using a new labeled set of size \(d_l\) to estimate \(\alpha \in \mathbb{R}^{d_u}\). This is done by solving the following optimization problem:

\[
\min_{\alpha} \left\{ \sum_{i=1}^{d_u} \left[ 1 - \sum_{j=1}^{d_u} \alpha_j l(x_i)K(x_i, x_j) \right] + \sum_{j=1}^{d_u} |\alpha_j| \leq 1/\gamma \right\}. \tag{2}
\]

Note that this problem can be solved efficiently by linear programming. Also, as the problem is \(L_1\)-constrained, tuning the value of \(\gamma\) will produce a sparse solution. The main limitation of this approach is that the similarity function \(K\) is predefined and \cite{Balcan:2008} did not provide any learning algorithm to design \((\epsilon, \gamma, \tau)\)-good similarities.

This problem has been fixed by \cite{Bellet:2012} who optimized the \((\epsilon, \gamma, \tau)\)-goodness of a bilinear similarity function under Frobenius norm regularization. The learned metric is then used to build a global linear classifier for which Theorem 1 holds. Moreover, their algorithm comes with a uniform stability proof which allows them to derive a bound on the generalization error of the associated classifier. Their formulation is strongly convex w.r.t. the metric and takes the following form:

\[
\min_{A} \frac{1}{d_l} \sum_{i=1}^{d_l} \left[ 1 - \frac{1}{\gamma d_u} \sum_{k=1}^{d_u} l(x_i)l(x_k)K_A(x_i, x_k) \right] + \beta||A||_F^2, \tag{3}
\]

where \(K_A(x, x') = x^T A x'\) is the bilinear similarity. It is worth noticing that Equation 3 optimizes an empirical version of the notion of \((\epsilon, \gamma, \tau)\)-goodness of \cite{Balcan:2008}. Moreover, since the random indicator function \(R(x)\) defining the probabilistic set of reasonable points (as defined in Equation 1) is unknown, \cite{Bellet:2012} resort to an additional set of \(d_u\) labeled points that are also used during the classifier learning step. Said differently, they do not exploit the unsupervised setting offered by Problem 2 that does not require to have access to the labels of the \(d_u\) landmarks.

More recently, \cite{Guo:2013} extended the theoretical results of \cite{Bellet:2012}. Using the Rademacher complexity (instead of the uniform stability) and Khinchin-type inequalities, they derive generalization bounds for similarity learning formulations that are regularized w.r.t. more...
general matrix-norms including the $L_1$ and the mixed $L_{(2,1)}$-norms. Moreover, they show that such bounds for the learned similarities can be used to upper bound the true risk of a linear SVM. There are three main distinctions between this approach and our work. Firstly, we propose a method that **jointly learns the metric and the linear separator** at the same time. This allows us to make use of the semi-supervised setting presented by Balcan et al. (2008) to learn well with only a small amount of labeled data. Secondly, our setting uses the algorithmic robustness to establish bounds. This way to proceed enables us to characterize some properties of our algorithm by exploiting the geometry of the (possibly unlabeled) data that is not the case with the Rademacher complexity. Lastly, our setting does not regularize the metric for reasons that will be explained in the following sections. We thus obtain a simpler formulation with less hyper-parameters.

3 LEARNING CONSISTENT GOOD SIMILARITY FUNCTIONS

In this section, we present our semi-supervised framework for jointly learning a similarity function and a linear separator from data. We also provide a generalization bound for our approach based on the recent algorithmic robustness framework proposed by Xu & Mannor (2010; 2012). We end this section by presenting some particular similarity functions that can be used in our setting.

3.1 OPTIMIZATION PROBLEM

Let $\mathcal{S}$ be a sample set of $d_l$ labeled examples $(x, l(x)) \in \mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ and $d_u$ unlabeled examples. We assume that $||x||_2 \leq 1$. All the unlabeled samples are considered reasonable ($\tau = 1$). Let $K_A(x, x')$ be a generic $(\epsilon, \gamma, \tau)$-good similarity function, parameterized by the matrix $A \in \mathbb{R}^{d \times d}$. We want to optimize the goodness of $K_A$ w.r.t. the empirical loss of a finite sample. To this end, we must find the matrix $A$ and the global separator $\alpha \in \mathbb{R}^{d_u}$ that minimize the loss function (in our case, the hinge loss) over the training set $\mathcal{S}$. Our learning algorithm takes the form of the following constrained optimization problem.

$$\min_{\alpha, A} \quad \frac{1}{d_l} \sum_{i=1}^{d_l} \left[ 1 - \sum_{j=1}^{d_u} \alpha_j l(x_i) K_A(x_i, x_j) \right] +$$

s.t. $\sum_{j=1}^{d_u} |\alpha_j| \leq 1/\gamma$ (4)

$A$ diagonal

$|A_{kk}| \leq 1, \quad 1 \leq k \leq d$ (6)

The novelty of this algorithm is the joint optimization over $A$ and $\alpha$: by solving problem (3), we are learning the metric and the separator at the same time. A significant advantage of this formulation is that it extends the semi-supervised setting from the separator learning step to the metric learning, and the two problems are solved using the same data. This method can naturally be used in situations where one has access to few labeled examples and many unlabeled ones: the labeled examples are used in this case to select the unlabeled examples that will serve to classify new points. Another important advantage of our technique is that the constraints on the pair of points do not need to be satisfied entirely, as the loss is averaged on all the reasonable points. In other words, this formulation is less restrictive than pair or triplet-based settings. Constraint (4) takes into account the desired margin $\gamma$ and is the same as in Balcan et al. (2008). Constraints (5) and (6) come to restrict the similarity $K_A$, as it is a generic form and its bounds are not known. This setting allows to preserve the $(\epsilon, \gamma, \tau)$-goodness of $K_A$. Constraining the values in the matrix $A$ limits the risk of overfitting, and thus plays a similar role to regularization. Another argument in favor of this formulation is that regularizing the metric through standard $L_1$ or $L_{(1,2)}$ norms would slowly push the values in the matrix towards zero, while sparsity is not necessarily a wanted feature. Indeed, let $f(x) = \sum_{j=1}^{d_u} \alpha_j K_A(x, x_j)$ be the output of the linear separator w.r.t. $x$. For some linear similarities $K_A(x, x')$, such as the bilinear form $K_A(x, x') = x^T A x'$, computing $f(x)$ boils down to calculating the similarity between $x$ and the barycenter of the (weighted) unlabeled points, making sparsity superfluous.
3.2 Consistency Guarantees

We now present a theoretical analysis of our approach. For the purpose of discussing the algorithmic robustness of the method, let us rewrite the minimization problem (3) with a more generalized notation of the loss function:

$$\min \frac{1}{d_i} \sum_{i=1}^{d_i} \ell(A, \alpha, z_i = (x_i, l(x_i))),$$

where $\ell(A, \alpha, z_i = (x_i, l(x_i))) = \left[ 1 - \sum_{j=1}^{d_i} \alpha_j l(x_i) K_{A}(x_i, x_j) \right]_+$ is the instantaneous loss estimated at point $(x_i, l(x_i))$. Therefore, the optimization problem (3) under constraints (4), (5), and (6) reduces to minimizing the empirical loss $\hat{R}_\ell = \frac{1}{\ell} \sum_{i=1}^{d_i} \ell(A, \alpha, z_i)$ over the training set $S$. To begin with, let us recall the notion of robustness of an algorithm $A$.

**Definition 2** (Algorithmic Robustness (Xu & Mannor 2010; 2012)). Algorithm $A$ is $(M, \epsilon(\cdot))$-robust, for $M \in \mathbb{N}$ and $\epsilon(\cdot) : \mathbb{Z}^{d_i} \to \mathbb{R}$, if $Z$ can be partitioned into $M$ disjoint sets, denoted by $\{C_i\}_{i=1}^{M}$, such that the following holds for all $S \in \mathbb{Z}^{d_i}$:

1. $\forall z = (x, l(x)) \in S, \forall z' = (x', l(x')) \in Z, \forall i \in [M] : \exists z \in C_i, \text{then } |\ell(A, \alpha, z) - \ell(A, \alpha, z')| \leq \epsilon(S)$.

This notion is a desired property of a learning algorithm, as it implies a lack of sensitivity to small perturbations in the training data in the same sense as robust optimization. Roughly speaking, an algorithm is robust if for any example $z'$ falling in the same subset as a training example $z$, the gap between the losses associated with $z$ and $z'$ is bounded (see Figure 2 in Appendix). In other words, robustness characterizes the capability of an algorithm to perform similarly on close train and test instances.

The closeness of the instances is based on the notion of covering number (Kolmogorov & Tikhomirov 1961) allowing one to cover them is no more than a fixed quantity $\rho$. The covering is built as follows: first we consider a $\rho$-cover over the instance $\mathcal{X}$, then we partition $\mathcal{Z}$ by considering one $\rho$-cover over $\mathcal{X}$ for the positive instances and another $\rho$-cover over $\mathcal{X}$ for the negative instances ensuring that two examples in the same region belong to the same class and the distance between them is no more than $\rho$ (Xu & Mannor 2010; 2012 for details). Note that with this construction the $\rho$-covers verify the cluster assumption used in semi-supervised learning (Chapelle et al. 2006).

Now we can state the first theoretical contribution of this paper.

**Theorem 2.** Given a partition of $\mathcal{Z}$ into $M$ subsets $\{C_i\}$ such that $z = (x, l(x))$ and $z' = (x', l(x')) \in C_i$ and $l(x') = l(x)$, and provided that $K_{A}(x, x')$ is $l$-lipschitz w.r.t. its first argument, the optimization problem (3) with constraints (4), (5), and (6) is $(M, \epsilon(S))$-robust with $\epsilon(S) = \frac{1}{\gamma} l \rho$, where $\rho = \sup_{x, x' \in C_i} ||x - x'||$.

The proof of Theorem 2 is given in Appendix. We now give a PAC generalization bound on the true loss making use of the previous robustness result. Let $\hat{R}_\ell = \mathbb{E}_{x \sim \mathcal{Z}} \ell(A, \alpha, z)$ be the true loss w.r.t. the unknown distribution $\mathcal{Z}$ and $\hat{R}_\ell = \frac{1}{\ell} \sum_{i=1}^{d_i} \ell(A, \alpha, z_i)$ be the empirical loss over the training set $S$. Based on the results of Xu & Mannor (2010; 2012), the proof requires the use of the following concentration inequality over multinomial random variables allowing one to capture statistical information coming from the different regions of the partition of $\mathcal{Z}$.

**Proposition 3.** (van der Vaart & Wellner 1996)

Let $\{\lceil N_i \rceil, \ldots, \lceil N_M \rceil\}$ an i.i.d. multinomial random variable with parameters $d_i = \sum_{i=1}^{M} \lceil N_i \rceil$ and $(p(C_1), \ldots, p(C_M))$. By the Bretagnolle-Huber-Carol inequality we have:

$$\Pr \left\{ \sum_{i=1}^{M} \left| \frac{\lceil N_i \rceil}{d_i} - p(C_i) \right| \geq \lambda \right\} \leq 2^M \exp \left( -\frac{d_i \lambda^2}{2} \right),$$

hence with probability at least $1 - \delta$,

$$\sum_{i=1}^{M} \left| \frac{N_i}{d_i} - p(C_i) \right| \leq \sqrt{\frac{2M \ln 2 + 2 \ln(1/\delta)}{d_i}}.$$

We are now able to present our generalization bound thanks to the following theorem.
Theorem 4. Considering that problem (3) is \((M, \epsilon(S))\)-robust, and that \(K_A\) is \(l\)-lipschitz w.r.t. to its first argument, for any \(\delta > 0\) with probability at least \(1 - \delta\), we have:

\[
|\mathcal{R}^\ell - \hat{\mathcal{R}}^\ell| \leq \frac{1}{\gamma} l \rho + B \sqrt{\frac{2M \ln 2 + 2 \ln(1/\delta)}{d_l}},
\]

where \(B = 1 + \frac{1}{\gamma}\) is an upper bound of the loss \(\ell\).

The proof of Theorem 4 follows the one described in [Xu & Mannor, 2010; 2012] and is given in the Appendix. Note that in robustness bounds, the cover radius \(\rho\) can be made arbitrarily small at the expense of larger values of \(M\). As \(M\) appears in the second term, which decreases to 0 when \(d_l\) tends to infinity, this bound provides a standard \(O(1/\sqrt{d_l})\) asymptotic convergence.

3.3 Robustness Analysis for Different Similarity Functions

Our main theorems strongly depend on the \(l\)-lipschitzness of the similarity function. In this section, we focus on some particular similarities that can be used in our setting. The proofs for the following functions are detailed in the Appendix.

Similarity function 1. Let \(K_A^1\) be the bilinear form \(K_A^1(x, x') = x^T A x'\). \(K_A^1(x, x')\) is \(1\)-lipschitz w.r.t. its first argument.

Similarity function 2. We define \(K_A^2(x, x') = 1 - (x - x')^T A (x - x')\), a similarity derived from the Mahalanobis distance. \(K_A^2(x, x')\) is \(4\)-lipschitz w.r.t. its first argument.

Similarity function 3. Let \(K_A^3(x, x') = \exp(-\frac{(x-x')^T A (x-x')}{2\sigma^2})\). \(K_A^3(x, x')\) is \(l\)-lipschitz w.r.t. its first argument with \(l = \frac{2}{2\sigma} (\exp\left(\frac{1}{2\sigma^2}\right) - \exp\left(-\frac{1}{2\sigma^2}\right))\).

As both \(K_A^1\) and \(K_A^2\) are linear w.r.t. their arguments, they have the main advantage to keep problem (3) convex. \(K_A^3\) is also based on the Mahalanobis distance, but this time it is a non linear function, resembling more a gaussian kernel. Plugging \(l = 1\) (resp. \(l = 4\) and \(l = \frac{2}{2\sigma} (\exp\left(\frac{1}{2\sigma^2}\right) - \exp\left(-\frac{1}{2\sigma^2}\right))\)) in Theorem 4 we obtain consistency results for problem (3) using \(K_A^3\) (resp. \(K_A^2\) and \(K_A^1\)). As the gap between empirical and true loss presented in Theorem 4 is proportional with \(l\) for the \(l\)-lipschitzness of each similarity function, we would like to keep this parameter as small as possible. We notice that the generalization bound is tighter for \(K_A^3\) than for \(K_A^2\). The bound for \(K_A^3\) depends on the additional parameter \(\sigma\), that adjusts the influence of the similarity value w.r.t. the distance to the landmarks. The value of \(l\) goes to 0 as \(\sigma\) augments, so larger values of \(\sigma\) are preferable in order to obtain a tight bound for the generalization error. However, note that when \(\sigma\) is large, the exponential behaves almost linearly, i.e. the projection loses its non-linear power.

4 Experiments

The state of the art in metric learning is dominated by algorithms designed to work in a purely supervised setting. Furthermore, most of them optimize a metric adapted to \(k\)-NN classification (e.g. LMNN, ITML), while our work is designed for finding a global linear separator. It is for these reasons that it is difficult to propose a fair comparative study. Therefore, we propose to compare our method to the algorithms that have similar settings: [Balcan et al., 2008]’s method and [Bellet et al., 2012]’s algorithm. We conduct the experimental study on 7 classic datasets taken from the UCI Machine Learning Repository, both binary and multi-class. Their characteristics are presented in Table 1. These datasets are widely used for metric learning evaluation.

4.1 Experimental Setup

We compare the following methods: the algorithm from [Balcan et al., 2008] as a baseline (BBS from now on), SLLC [Bellet et al., 2012] and problem (3) with constraints (4), (5) and (6) (JSL, for Joint Similarity Learning). We plug into BBS and JSL the three similarity functions studied.

We used the code provided by the authors.
under review as a conference paper at ICLR 2015

Table 1: Properties of the datasets used in the experimental study

|        | Balance | Ionosphere | Iris | Liver | Pima | Sonar | Wine |
|--------|---------|------------|------|-------|------|-------|------|
| # Instances | 625    | 351        | 150  | 345   | 768  | 208   | 178  |
| # Dimensions | 4      | 34         | 4    | 6     | 8    | 60    | 13   |
| # Classes   | 3      | 2          | 3    | 2     | 2    | 2     | 3    |

Table 2: Average accuracy (%) with confidence interval at 95%, best result per method

| Lmks.     | Method | Balance | Ionosphere | Iris | Liver | Pima | Sonar | Wine | Avg. |
|-----------|--------|---------|------------|------|-------|------|-------|------|------|
| all pts.  | BBS    | 86.6±2.7| 89.8±3.0   | 74.5±5.4| 67.6±4.5| 73.7±4.2| 69.4±2.9| 87.7±5.9| 78.5±4.1|
|           | SLLC   | 85.9±3.0| 87.1±4.4   | 74.5±5.4| 64.1±7.3| 74.8±4.6| 76.5±5.2| 95.0±2.3| 79.7±4.6|
|           | JSL    | 87.1±2.5| 91.0±2.0   | 74.5±4.4| 69.2±3.2| 72.9±3.9| 72.3±4.1| 87.7±5.0| 79.2±3.6|
| 15 pts.   | BBS    | 86.0±3.0| 86.7±2.5   | 78.2±4.5| 58.6±3.8| 74.3±3.9| 73.2±6.0| 91.5±7.5| 78.4±4.5|
|           | SLLC   | 75.3±5.8| 86.7±3.7   | 75.5±6.4| 64.1±6.6| 70.5±3.7| 68.7±7.5| 95.4±2.5| 76.6±5.2|
|           | JSL    | 87.5±2.7| 86.7±1.6   | 75.5±2.3| 67.3±4.3| 74.3±4.1| 77.4±6.3| 88.5±5.0| 79.6±3.8|

previously (see Section 3.3), thus obtaining six settings. For SLLC, we use the same setting as the authors. As BBS’s setting contains no metric learning step, the metric A is set to the identity matrix, which is equivalent to using standard, non parametered similarity functions. All attributes are centered around zero and scaled to ensure \( |x|_2 \leq 1 \), as this constraint is necessary for all algorithms. We randomly choose 15% of the data for validation purposes, and another 15% as a test set. The training set and the unlabeled data are chosen from the remaining 70% of examples not employed in the previous sets. As SLLC is a fully supervised method, we give it access to the labels of the data otherwise used as unlabeled. In order to illustrate the classification using a restricted quantity of labeled data, the number of labeled points is limited to 5 examples per class, as this is usually a reasonable minimum amount of annotation to rely on. The number of landmarks is either equal to the size of the training set, either set to 15 points, to parallel the fact that we consider 5 labeled points per class. In the latter case, the unlabeled data is chosen from the data as the points closest to the centroids determined by applying k-means++ clustering. When all the available data is used as landmarks, the \( L_1 \) constraint on \( \alpha \) forces the algorithm to choose the most valuable of them by adapting their respective weights. All of the experimental results are averaged over 10 runs, for which we compute a 95% confidence interval. For BBS and JSL, we tune the following parameters by cross-validation: \( \gamma, \beta \in \{10^{-2}, \ldots, 10^{-1}\} \) (for \( K^A_\lambda \)), choosing the value yielding the best accuracy. For SLLC, we tune \( \gamma, \beta \in \{10^{-2}, \ldots, 10^{-2}\} \), \( \lambda \in \{10^{-3}, \ldots, 10^2\} \), as done by the authors. We solve BBS and JSL using projected gradient descent. In JSL, we alternate the optimization variable from \( \alpha \) to \( A \).

4.2 Results

Classification accuracy We report the results obtained for linear classification using the sparse linear classifier proposed by Balcan et al. (2008) in Table 2. JSL obtains the best performance on 6 of the 14 configurations, and a tie on 3 others. Its improvement over SLLC is significant in some of the cases, even though SLLC uses more labeled data. It is, however, outperformed by SLLC in datasets like Wine, or in cases where SLLC uses much more labeled data than JSL (Pima and Sonar, all points used as landmarks). For the latter setting, this advantage is reflected also on the averaged results, while on average JSL outperforms the other methods when using few landmarks. JSL is also more stable when passing from all points used as landmarks to only 15 points.

Choice of similarity Tables 3 and 4 present the impact of the form of the similarity function for BBS and JSL. The best result for pairwise comparison of the two methods using the same similarity function is marked in bold. When using all the available data as (unlabeled) landmarks (Table 3), our method achieves the best results on 15 out of 21 settings, and a tie on one configuration. Table 4 summarizes the results obtained when the unlabeled dataset contains only 15 points. In this case, our technique yields the best accuracy on 11 out of the 21 settings, and similar results in two other cases. Should one wish to compare the similarity functions in the same settings, we notice that \( K^{A^2}_\lambda \)
Table 3: Average accuracy (%) with confidence interval at 95%, all points used as landmarks.

| Method | Balance | Ionosphere | Iris | Liver | Pima | Sonar | Wine |
|--------|---------|------------|------|-------|------|-------|------|
| BBS, \(K^A_1\) | 85.6±3.4 | 88.1±3.2 | 73.6±4.3 | **64.5±3.4** | 71.0±3.0 | 69.4±2.9 | **87.7±5.9** |
| JSL, \(K^A_1\) | 85.7±3.5 | **88.5±2.6** | **74.5±4.4** | 63.9±5.3 | **71.1±3.8** | **72.3±4.1** | **87.7±5.0** |
| BBS, \(K^A_2\) | 86.6±2.7 | 89.8±3.0 | **74.5±5.4** | 67.6±4.5 | **73.7±4.2** | 67.7±4.5 | **86.9±5.4** |
| JSL, \(K^A_2\) | 87.1±2.5 | **91.0±2.0** | 71.4±5.9 | **69.2±3.2** | **72.9±3.9** | 71.9±4.2 | 84.2±6.9 |
| BBS, \(K^A_3\) | 62.9±15.7 | 82.7±7.0 | 62.3±12.9 | 54.1±8.5 | 69.8±4.6 | **64.5±8.1** | 77.3±13.5 |
| JSL, \(K^A_3\) | **81.1±8.5** | **86.2±2.8** | **68.2±8.5** | **58.6±6.3** | **71.1±4.3** | 63.9±10.0 | **83.5±6.2** |

Table 4: Average accuracy (%) with confidence interval at 95%, 15 points used as landmarks.

| Method | Balance | Ionosphere | Iris | Liver | Pima | Sonar | Wine |
|--------|---------|------------|------|-------|------|-------|------|
| BBS, \(K^A_1\) | 84.6±3.2 | **86.7±2.5** | **78.2±4.5** | 58.6±3.8 | 69.7±2.4 | 72.6±5.6 | **89.6±3.9** |
| JSL, \(K^A_1\) | **84.9±2.6** | **86.7±1.6** | 75.5±2.3 | **63.1±5.9** | **71.1±4.1** | **72.9±4.6** | 87.3±5.5 |
| BBS, \(K^A_2\) | 86.0±3.0 | 84.8±3.6 | **75.9±5.5** | 58.2±3.9 | **74.3±3.9** | 73.2±6.0 | **86.9±11.2** |
| JSL, \(K^A_2\) | **87.5±2.7** | **85.0±3.8** | 74.1±6.3 | **67.3±4.3** | **74.3±4.1** | **77.4±6.3** | 76.9±10.5 |
| BBS, \(K^A_3\) | **81.6±11.4** | **77.5±8.3** | 72.3±5.0 | 53.3±6.1 | **69.6±4.6** | 59.7±13.3 | **91.5±7.5** |
| JSL, \(K^A_3\) | 79.6±10.0 | 76.3±7.4 | **72.7±6.3** | **59.6±6.0** | 69.0±8.6 | **68.7±10.0** | 88.5±5.0 |

is the one yielding the best results in most of the cases for both tables. This is surprising, considering that the cosine similarity \(K^A_1\) is 1-lipschitz, thus resulting in a tighter generalization bound, as \(K^A_2\) is 4-lipschitz. We explain this fact by the geometric characteristics of the involved datasets, which makes the Mahalanobis distance more suitable for classification than the generalized cosine. When comparing Tables 3 and 4 we notice that there are only a few cases when the best accuracy is attained with less unlabeled points, but that when this happens the improvement is significant. This phenomenon is due to the fact that the 15 unlabeled points are not chosen randomly, but contain relevant information w.r.t. data topology. In the case of \(K^A_3\), there is a trade-off between the tightness of the bound in Theorem 4 and the stability of the results. Large values of \(\sigma\) will lead to tighter bounds (as \(l\) is smaller), but the resulting similarity function becomes linear and less discriminative. As a consequence, the results vary more for this similarity function, leading to larger confidence intervals, as can be seen in Tables 3 and 4 on almost all the collections.

**Quantity of labeled data** As an illustration of the methods’ behavior when the level of supervision varies, Figure 1 presents the accuracies on Ionosphere with an increasing number of labeled examples. The best results are obtained by JSL with \(K^A_1\) and SLLC. Notice that the accuracy of JSL improves when adding more labeled data, while SLLC obtains the best results with 50%-70% of the available data, followed by a drop when the whole dataset is used.

![Figure 1: Average accuracy w.r.t. the number of labeled points on Ionosphere with 15 landmarks.](image-url)
5 CONCLUSIONS

In this paper, we present a semi-supervised framework for learning an \((\epsilon, \gamma, \tau)\)-good similarity function and a classifier at the same time. We show that our joint approach is theoretically founded using results from Balcan et al. (2008) and new results based on algorithmic robustness. This setting is designed to learn well with a limited amount of labeled data and performs well in practice on various UCI datasets. Future work could cover a kernelized version of our technique to learn more efficient similarities and classifiers in a non-linear feature space, as well as learning local metrics and a combination of them in a coherent framework.

REFERENCES

Balcan, M.-F., Blum, A., and Srebro, N. Improved guarantees for learning via similarity functions. In Servedio, R. A. and Zhang, T. (eds.), COLT, pp. 287–298. Omnipress, 2008.

Bao, J.-P., Shen, J.-Y., Liu, X.-D., and Liu, H.-Y. Quick asymmetric text similarity measures. ICMLC, 2003.

Baoli, L., Qin, L., and Shiwen, Y. An adaptive k-nearest neighbor text categorization strategy. ACM TALIP, 2004.

Bellet, A., Habrard, A., and Sebban, M. Similarity learning for provably accurate sparse linear classification. In ICML, pp. 1871–1878, 2012.

Bellet, A., Habrard, A., and Sebban, M. A survey on metric learning for feature vectors and structured data. arXiv preprint arXiv:1306.6709, 2013.

Bousquet, O. and Elisseeff, A. Stability and generalization. JMLR, 2:499–526, March 2002.

Chapelle, O., Schölkopf, B., and Zien, A. (eds.). Semi-Supervised Learning. MIT Press, Cambridge, MA, 2006.

Davis, J. V., Kulis, B., Jain, P., Sra, S., and Dhillon, I. S. Information-theoretic metric learning. In ICML, pp. 209–216, New York, NY, USA, 2007. ACM.

Diligenti, M., Maggini, M., and Rigutini, L. Learning similarities for text documents using neural networks. In ANNPR, 2003.

Grabowski, M. and Szałas, A. A technique for learning similarities on complex structures with applications to extracting ontologies. In AWIC, LNAI. Springer Verlag, 2005.

Guo, Z.-C. and Ying, Y. Guaranteed classification via regularized similarity learning. CoRR, abs/1306.3108, 2013.

Hust, A. Learning Similarities for Collaborative Information Retrieval. In Proceedings of KI 2004 workshop “Machine Learning and Interaction for Text-Based Information Retrieval”, TIR-04, pp. 43–54, 2004.

Kolmogorov, A. and Tikhomirov, V. \(\epsilon\)-entropy and \(\epsilon\)-capacity of sets in functional spaces. American Mathematical Society Translations, 2(17):277–364, 1961.

Qamar, A. M. and Gaussier, É. Online and batch learning of generalized cosine similarities. In ICDM, pp. 926–931, 2009.

Shalev-Shwartz, S., Singer, Y. and Ng, A. Y. Online and batch learning of pseudo-metrics. In ICML, New York, NY, USA, 2004. ACM.

van der Vaart, A. and Wellner, J. Weak Convergence and Empirical Processes. Springer series in statistics. Springer, 1996.

Weinberger, K. and Saul, L. Fast solvers and efficient implementations for distance metric learning. In ICML, pp. 1160–1167. ACM, 2008.

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

Xu, H. and Mannor, S. Robustness and generalization. In COLT, pp. 503–515, 2010.

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

Zhu, J., Rosset, S., Hastie, T., and Tibshirani, R. 1-norm support vector machines. In NIPS, pp. 16. MIT Press, 2003.
APPENDIX

This appendix details the notion of robustness (Figure 2), the proofs of Theorems 2 and 4, and the \ell\text{-lipschitzness of similarity functions [1][3].

![Figure 2: Illustration of the property of robustness, using a cover based on the L_1 norm.](Image 256x565 to 356x652)

**Proof of Theorem 2**

\[\left|\ell(A, \alpha, z) - \ell(A, \alpha, z')\right| \leq \left| \sum_{j=1}^{d_u} \alpha_j l(x'_j) K_A(x'_j, x_j) - \sum_{j=1}^{d_u} \alpha_j l(x) K_A(x, x_j) \right| \leq \sum_{j=1}^{d_u} |\alpha_j| \cdot |K_A(x'_j, x_j) - K_A(x, x_j)| \leq \sum_{j=1}^{d_u} |\alpha_j| \cdot l|x - x'| | \leq \frac{l}{\gamma} \rho \]

Setting \(\rho = \sup_{x, x' \in C_i} |x - x'|_1\), we get the Theorem. We get Inequality (8) from the 1-lipschitzness of the hinge loss; Inequality (9) comes from the classical triangle inequality; The first inequality on line (10) is due to the \ell\text{-lipschitzness of } K_A(x, x_j) \text{ w.r.t. its first argument, and the result follows from Condition 4.}\]

**Proof of Theorem 4** From Xu & Mannor [2010, 2012]

\[\left| R^\ell - \hat{R}^\ell \right| = \left| \sum_{i=1}^{M} E_{z \sim C_i} \ell(A, \alpha, z) - \frac{1}{d_t} \sum_{i=1}^{d_t} \ell(A, \alpha, z_i) \right| \]

\[\leq \sum_{i=1}^{M} E_{z \sim C_i} (\ell(A, \alpha, z)|z \in C_i) p(C_i) - \frac{1}{d_t} \sum_{i=1}^{d_t} \ell(A, \alpha, z_i) \]

\[\leq \sum_{i=1}^{M} E_{z \sim C_i} (\ell(A, \alpha, z)|z \in C_i) p(C_i) - \sum_{j=1}^{M} E_{z \sim C_i} (\ell(A, \alpha, z)|z \in C_i) \frac{|N_i|}{d_t} \]

\[+ \sum_{i=1}^{M} E_{z \sim C_i} (\ell(A, \alpha, z)|z \in C_i) \frac{|N_i|}{d_t} - \frac{1}{d_t} \sum_{i=1}^{d_t} \ell(A, \alpha, z_i) \]

\[= \sum_{i=1}^{M} E_{z \sim C_i} (\ell(A, \alpha, z)|z \in C_i) p(C_i) - \frac{|N_i|}{d_t} \]

\[+ \frac{1}{d_t} \sum_{i=1}^{M} \sum_{j \in C_i} E_{z \sim C_i} (\ell(A, \alpha, z)|z \in C_i) - \frac{1}{d_t} \sum_{i=1}^{M} \sum_{j \in C_i} \ell(A, \alpha, z_j) \]
\[
\begin{align*}
\frac{1}{\gamma} & + B \sqrt{\frac{2M \ln 2 + 2 \ln(1/\delta)}{dt}}.
\end{align*}
\]

Inequality \[11\] is due to the triangle inequality. Inequality \[12\] comes from the application of Proposition 1 and Theorem 1.

**Proof for similarity function** \[7\] In order to prove that \( K_1^\alpha \) is \( l \)-lipschitz, we need to bound the following difference.

\[
|K_1^\alpha(x, x') - K_1^\alpha(x'', x'')| = |(x''^T A x''') - (x'''^T A x''')|
\]

\[
= |(x - x')^T A (x - x'')| + |(x - x')^T A (x - x'')|
\]

\[
\leq |x - x'| \cdot |A x' - A x''|_2 + |x - x'| \cdot |A x - A x''|_2
\]

\[
\leq 2|x - x'| \cdot \max \left| \frac{A x - A x''}{2} \right|_2.
\]

Inequalities \[13\] comes from the Cauchy-Schwarz inequality and some classical norm properties; Inequality \[14\] is due to Conditions \[5\] and \[6\]. Finally, Inequality \[15\] comes from the assumption \( |x|_2 \leq 1 \).

**Proof for similarity function** \[2\] This proof is similar to the one for similarity \[2\].

\[
\begin{align*}
|K_1^\alpha(x, x'') - K_3^\alpha(x', x'')| &= \exp \left( -\frac{(x - x')^T A (x - x'')}{2\sigma^2} \right) - \exp \left( -\frac{(x - x')^T A (x - x'')}{2\sigma^2} \right) \\
&= \frac{1}{2\sigma^2} \left( \exp \left( \frac{1}{2\sigma^2} \right) - \exp \left( \frac{-1}{2\sigma^2} \right) \right) \\
&\cdot \left| (x - x')^T A (x - x'') + (x - x')^T A (x - x'') - (x - x')^T A (x - x'') \right|
\end{align*}
\]

\[
\leq \frac{1}{2\sigma^2} \left( \exp \left( \frac{1}{2\sigma^2} \right) - \exp \left( \frac{-1}{2\sigma^2} \right) \right) \cdot \left| (x - x')^T A (x - x') + (x - x')^T A (x - x') \right|
\]

\[
\leq \frac{1}{2\sigma^2} \left( \exp \left( \frac{1}{2\sigma^2} \right) - \exp \left( \frac{-1}{2\sigma^2} \right) \right) \cdot \left| (x - x')^T A (x - x') \right|
\]

\[
\leq \frac{1}{2\sigma^2} \left( \exp \left( \frac{1}{2\sigma^2} \right) - \exp \left( \frac{-1}{2\sigma^2} \right) \right) \cdot \left| (x - x')^T A (x' - x') \right|.
\]
\[ \leq \frac{1}{2\pi^2} \left( \exp \left( \frac{1}{2\sigma^2} \right) - \exp \left( -\frac{1}{2\sigma^2} \right) \right) 4||x' - x|| \tag{23} \]
\[ = \frac{2}{\pi^2} \left( \exp \left( \frac{1}{2\sigma^2} \right) - \exp \left( -\frac{1}{2\sigma^2} \right) \right) ||x - x'||. \]

Inequality (19) is due to the \( l \)-lipschitzness of the exponential function on the range \([-\frac{1}{\sqrt{2}\sigma^2}, \frac{1}{\sqrt{2}\sigma^2}]\). Inequalities (20) and (21) come from the Cauchy-Schwarz inequality and some classical norm properties. Inequality (22) is due to Conditions (5) and (6) and inequality (23) is due to \( ||x||_2 \leq 1 \). \qed