Learning Mahalanobis Distance Metric: Considering Instance Disturbance Helps*

Han-Jia Ye, De-Chuan Zhan, Xue-Min Si and **Yuan Jiang**
National Key Laboratory for Novel Software Technology
Collaborative Innovation Center of Novel Software Technology and Industrialization
Nanjing University, Nanjing, 210023, China
{yehj, zhandc, sixm, jiangy}@lamda.nju.edu.cn

Abstract

Mahalanobis distance metric takes feature weights and correlation into account in the distance computation, which can improve the performance of many similarity/dissimilarity based methods, such as kNN. Most existing distance metric learning methods obtain metric based on the raw features and side information but neglect the reliability of them. Noises or disturbances on instances will make changes on their relationships, so as to affect the learned metric. In this paper, we claim that considering disturbance of instances may help the metric learning approach get a robust metric, and propose the Distance metRIc learning Facilitated by disTurbances (DRIFT) approach. In DRIFT, the noise or the disturbance of each instance is *learned*. Therefore, the distance between each pair of (noisy) instances can be better estimated, which facilitates side information utilization and metric learning. Experiments on prediction and visualization clearly indicate the effectiveness of DRIFT.

1 Introduction

Similarity and dissimilarity are widely used in machine learning area, such as classification [Bian and Tao, 2011, Luo *et al.*, 2016], clustering [Xing *et al.*, 2003, Xiang *et al.*, 2008, Law *et al.*, 2016b] and retrieval [McFee and Lanckriet, 2010]. The goal of Distance Metric Learning (DML) is to find a better distance computation which can perform better than the Euclidean one. Given a positive semi-definite matrix M, the (squared) Mahalanobis distance between two instances $\mathbf{x}_i$ and $\mathbf{x}_j$ can be defined as:

$$\mathrm{dist}^2_M(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^\top M(\mathbf{x}_i - \mathbf{x}_j) \ .$$

Since it considers the relationship between different types of features [Lim *et al.*, 2013, Ye *et al.*, 2016b], its advantages have been discovered and validated from various perspectives [Kulis, 2012, Bellet *et al.*, 2015].

To train a Mahalanobis distance metric, various types of side information [Law *et al.*, 2016a] should be collected to provide a direction for distance relative comparisons. After searching a metric decreasing the violation of these constraints, similar instances become close to each other while dissimilar ones are far away. Although ground-truth side information leads to a well-learned Mahalanobis metric [Verma and Branson, 2015, Cao *et al.*, 2016], it is in fact unknown during the training process. Therefore, side information is often generated based on reliable raw features from various sources. For example, random choice [Davis *et al.*, 2007], Euclidean nearest neighbor selection [Weinberger *et al.*, 2006], and all-pair enumeration [Xing *et al.*, 2003, Mao *et al.*, 2016].

To reduce the uncertainty in side information, [Huang *et al.*, 2010] and [Wang *et al.*, 2012] try a selection strategy among all target neighbors. While it is more reasonable to assume that there are inaccuracies in feature value collection, since the feature inaccuracies or noises will damage the structure of neighbors, and consequently affect the reliableness of side information. From the aspect of this generative process, we tackle the unreliability in metric learning and propose the Distance metRIc Facilitated by disTurbances (DRIFT) approach, using which a robust distance metric is achieved based on the explicit consideration of instance disturbances.

In DRIFT, *all possible* variations of disturbances on instances are involved in the expected distance, so as to form different side information constraints as well as assign reasonable weights on them. Specifically, when a pair of noisy instances meets the requirement of the provided side information, the DRIFT's learned metric should tolerate perturbations by enlarging the similarity region. As such the robustness of metric will increase and the generalization ability can be improved. On the contrary, if the side information was hard to satisfy for the concerned pair, assigning obvious perturbations can be risky. Hence, the tolerance level of disturbances on instance pairs reflects the reliableness of side information to some extent. Moreover, perturbation distribution modeled noises make DRIFT have the ability to represent instances distribution quantitatively [Van Der Maaten *et al.*, 2013], and help reduce the effects of incorrect guidance in training. Therefore, it is expected that DRIFT can provide a robust distance metric with better discriminative ability.

DRIFT learns metric and disturbance of instances jointly. Benefited from metric decomposition, we get a simplified objective and acceleration variants with sub-problems further reducing to scalar group optimization. Our empirical investigations provide visualization effects demonstrating the interpretability of DRIFT. Real-world tasks validate DRIFT's superiorities on generalization and robustness, especially in

*This work was supported by NSFC (61632004, 61673201).

the case of unreliable instances/side information.

The rest of this paper starts with discussions about related methods. Then the DRIFT approach is presented in detail. The last are experiments and conclusion.

2 Related Work

Mahalanobis distance is widely researched in distance metric learning. It is originally used in the clustering task [Xing *et al.*, 2003] considering all pair comparisons. ITML [Davis *et al.*, 2007] utilizes the randomly chosen pairwise side information and information based regularizer. While triplets constraints are considered in LMNN [Weinberger *et al.*, 2006] to form a large margin objective. To find a better description of side information, a multi-stage strategy is proposed in [Weinberger and Saul, 2009, Zhan *et al.*, 2009], where the metric learned in the previous stage is used to find nearest neighbors in the current one. [Huang *et al.*, 2010] and [Wang *et al.*, 2012] traverse all target neighbors to find best candidates. In DRIFT, we propose a new perspective on the refinement of side information by considering the disturbances over instances. Different metric learning methods and the ways they use side information can be found in [Bellet *et al.*, 2015].

Perturbations modeling can be regarded as a type of regularization [Wager *et al.*, 2013] to train a robust model [Chen *et al.*, 2014, Wangni and Chen, 2016] or get better feature representations [Van Der Maaten *et al.*, 2013, Chen *et al.*, 2015, Li *et al.*, 2016]. Qian *et al.* [2014] first consider noises in metric learning, but only *fixed* covariance perturbation is used to get a low rank solution. In DRIFT, we *learn* the perturbation distribution to directly model the noises for a robust metric.

The disturbance distribution is also closely related to the instance distributions, and consequently correlated with the instance generation mechanism. Different from [Ye *et al.*, 2016a], where distributions are considered to model the multiple metrics and indirectly infer the metric for unseen instance, DRIFT explicitly models the distribution related to instances and side information. Mao *et al.* [2016] study robust manifold learning. Nevertheless, they focus on the instance distribution towards preserving their Euclidean distances. On the contrary, DRIFT approach considers the disturbance distribution directly for better discriminative ability.

3 Learning Distance Metric Considering Instance Disturbance

The Distance metRIc learning Facilitated by disTurbances (DRIFT) approach learns instance disturbances and distance metric jointly. In this section, we introduce notations first, then give a description of the distribution perturbed distance computation. After that, detailed DRIFT formulation and its optimization strategy are presented. Acceleration strategies are described at last.

3.1 Notations

Given a training set $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^N$, each instance $\mathbf{x}_i \in \mathbb{R}^d$ has a label $y_i \in 1, 2, \ldots, C$. We focus on the input side information in the form of T triplets.[1] In the t-th triplet $\{\mathbf{x}_i^t, \mathbf{x}_j^t, \mathbf{x}_k^t\}$, $\mathbf{x}_j^t$ is the target neighbor of instance $\mathbf{x}_i^t$ and they should be close to each other using learned distance. While $\mathbf{x}_k^t$ is the imposter, i.e., a different class instance that needed to be pushed away. The learned Mahalanobis distance metric M lies in the set of positive semi-definite matrix $\mathcal{S}_d^+$. $\|M\|_F^2 = \langle M, M\rangle = \mathrm{Tr}(M^\top M)$ is the Frobenius norm of a matrix. I is the identity matrix. $[\cdot]_+$ is a scalar input function which only preserves the non-negative part of input value.

We use $\mathcal{P}$ to denote the set of valid probability distributions (nonnegative and sum to one over random variable space). Denote $p_i(\epsilon) \in \mathcal{P}$ as the perturbation distribution for instance $\mathbf{x}_i$ and $\mathbf{p} = \{p_i(\epsilon)\}_{i=1}^N$ is the set of all these distributions. For random variable $\epsilon \in \mathbb{R}^d$, the KL-divergence can produce a non-negative inconsistency measurement between two distributions $p(\epsilon)$ and $p_0(\epsilon)$, which is defined as $\mathrm{KL}(p\|p_0) = \int p(\epsilon)\log\frac{p(\epsilon)}{p_0(\epsilon)}\mathrm{d}\epsilon$.

3.2 Instance Disturbances in Metric Learning

Instance disturbance affects its neighborhood structures, inducing unreliability in training, which can be used for facilitating the utilization of side information. Taking perturbations into account in the distance computation, variants of instances should be used to explain the guidance of side information. In DRIFT, We focus on the *expected* Mahalanobis distance with metric M between two instances $\mathbf{x}_i$ and $\mathbf{x}_j$, which is equivalent to covering *all* the instances $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{x}}_j$ sampled from instance distribution $p(\mathbf{x}_i)$ and $p(\mathbf{x}_j)$, respectively [Li *et al.*, 2016, Mao *et al.*, 2016]:

$$\begin{aligned}
&\mathbb{E}_{\hat{\mathbf{x}}_i,\hat{\mathbf{x}}_j}[\mathrm{dist}_M^2(\hat{\mathbf{x}}_i,\hat{\mathbf{x}}_j)] = \mathbb{E}_{\hat{\mathbf{x}}_i,\hat{\mathbf{x}}_j}\left[(\hat{\mathbf{x}}_i-\hat{\mathbf{x}}_j)^\top M(\hat{\mathbf{x}}_i-\hat{\mathbf{x}}_j)\right]\\
&= \iint \hat{\mathbf{x}}_i^\top M\hat{\mathbf{x}}_i + \hat{\mathbf{x}}_j^\top M\hat{\mathbf{x}}_j - 2\hat{\mathbf{x}}_j^\top M\hat{\mathbf{x}}_i\; p(\hat{\mathbf{x}}_i)p(\hat{\mathbf{x}}_j)\mathrm{d}\hat{\mathbf{x}}_i\mathrm{d}\hat{\mathbf{x}}_j\\
&= \mathbb{E}_{\hat{\mathbf{x}}_i}[\hat{\mathbf{x}}_i^\top M\hat{\mathbf{x}}_i] + \mathbb{E}_{\hat{\mathbf{x}}_j}[\hat{\mathbf{x}}_j^\top M\hat{\mathbf{x}}_j] - 2\mathbb{E}_{\hat{\mathbf{x}}_i}[\hat{\mathbf{x}}_i]^\top M\mathbb{E}_{\hat{\mathbf{x}}_j}[\hat{\mathbf{x}}_j]\ . \quad (1)
\end{aligned}$$

Last step in Eq. 1 comes from the independent assumption between instances $\mathbf{x}_i$ and $\mathbf{x}_j$. Since it is a general assumption that the disturbances on instances are centralized, i.e., $\mathbb{E}_{\hat{\mathbf{x}}_i}[\hat{\mathbf{x}}_i] = \mathbf{x}_i$, the above expected distance can be further transformed into:

$$\mathbb{E}_{\hat{\mathbf{x}}_i,\hat{\mathbf{x}}_j}[\mathrm{dist}_M^2(\hat{\mathbf{x}}_i,\hat{\mathbf{x}}_j)] = \mathrm{dist}_M^2(\mathbf{x}_i,\mathbf{x}_j) + \langle M, \mathrm{Cov}[\mathbf{x}_i] + \mathrm{Cov}[\mathbf{x}_j]\rangle\ . \quad (2)$$

$\mathrm{Cov}[\mathbf{x}_i] \in \mathcal{S}_d^+$ is the covariance matrix of distribution $p(\mathbf{x}_i)$. Hence, the expected Mahalanobis distance between two instances is appended with a term of covariances. Moreover, we can model the disturbance of instance based on Eq. 2 by introducing an *unbiased* random perturbation $\epsilon \in \mathbb{R}^d$, sampled from a distribution $p(\epsilon)$, to the expected distance computation. Therefore, the difference of two perturbed instances sampled from $p(\mathbf{x}_i)$ and $p(\mathbf{x}_j)$ can be denoted as $\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j = \mathbf{x}_i - \mathbf{x}_j + \epsilon$. Thus disturbance over distance also considers the variations over instances, as revealed by the distance transformation:

$$\mathbb{E}_{\hat{\mathbf{x}},\hat{\mathbf{y}}}[\mathrm{dist}_M^2(\hat{\mathbf{x}},\hat{\mathbf{y}})] = \mathrm{dist}_M^2(\mathbf{x},\mathbf{y}) + \mathbb{E}_\epsilon[\epsilon^\top M\epsilon]\ . \quad (3)$$

For the PSD property of metric M, the last term in Eq. 3 is a quadratic form which is positive no matter what value of ϵ takes. So the expected distance in Eq. 3 has the *expansion property* that enlarges original Mahalanobis distance value. In our DRIFT method, we consider the expected distance and learn the distribution over ϵ.

[1] Learning with perturbed distance in the pairwise form can also be formulated in a similar way.

Given a triplet $\{\mathbf{x}_i^t, \mathbf{x}_j^t, \mathbf{x}_k^t\}$, the metric M should make the distance between $\mathbf{x}_i^t$ and imposter $\mathbf{x}_k^t$ larger than the distance between $\mathbf{x}_i^t$ and target neighbor $\mathbf{x}_j^t$ beyond a margin. Due to the expansion property, there is no additional effect in considering expected distance for imposter comparisons. Together with the fact that target neighbor relationship possesses more uncertainty [Wang *et al.*, 2012], it's better to invoke the expected distance only in measuring the target neighbor pair. Therefore, we can formulate our Distance metRIc Facilitated by disTribution (DRIFT) approach as follows:

$$\begin{aligned}
&\min_{M,\,\mathbf{p}} \frac{1}{2}\|M\|_F^2 + \lambda_1 \sum_{i=1}^{N} \mathrm{KL}\big(p_i(\epsilon)\|p_0(\epsilon)\big) + \lambda_2 \sum_{t=1}^{T} \xi_t \ , \\
&s.t.\ \forall t, \mathrm{dist}_M^2(\mathbf{x}_i^t, \mathbf{x}_k^t) - \mathbb{E}\big[\mathrm{dist}_M^2(\mathbf{x}_i^t, \hat{\mathbf{x}}_j^t)\big] \geq 1 - \xi_t,\ \xi_t \geq 0 \ , \\
&\quad M \in \mathcal{S}_d^+ \ ,\ \ \forall i,\ p_i(\epsilon) \in \mathcal{P}, \qquad (4)
\end{aligned}$$

where the first part in the objective is a Frobenius norm regularizer on metric M. The second term is a distribution regularizer, i.e., KL-divergence is used to make the learned perturbation distribution $p_i(\epsilon)$ close to a specified prior $p_0(\epsilon)$. As in the general setting [Mao *et al.*, 2016], we choose prior as a zero-mean multivariate normal distribution: $p_0 \sim \mathcal{N}(0, \Sigma_0)$ for all instances, which satisfies the unbiased requirement. It is notable that in the Eq. 4, we do not constrain the form of $p_i(\epsilon)$ but only require it as a valid distribution. Instances' different perturbations give rise to various impacts when computing distance with others, which considers local properties of instances. The third term minimizes the large margin violation. For each instance, the distance between imposters should be larger than (beyond a margin value) the *expected distance* between its target neighbors.

In our solution, we optimize over target neighbor instance $\mathbf{x}_j^t$ around its neighborhood to find the best disturbance, i.e., only the target neighbor $\hat{\mathbf{x}}_j^t = \mathbf{x}_j^t + \epsilon, \epsilon \sim p_j(\epsilon)$ is perturbed. This simplification gets the same results as considering the distribution on the perturbation of instance differences. Since most existing distance metric learning methods use Euclidean nearest neighbors as target neighbors [Weinberger and Saul, 2009], the perturbation of target neighbors relieves the problem of initial target selection and sets the target neighbor having the right distance with others as well. In addition, it is obvious that learning the parameters of the noise distribution can be regarded as measuring the tolerance of perturbation on target neighbors. Pairs satisfying constraint easily can tolerate perturbations more to some extent, and expand the similar range w.r.t. a center instance, which is shown in Fig. 1. On the other hand, these pairs attract more weights in the training, thus a robust metric is expected to be obtained.

3.3 Optimization for DRIFT

The objective formulation of DRIFT can be transformed to:

$$\begin{aligned}
\min_{M\in\mathcal{S}_d^+,\ p_i\in\mathcal{P}} &\frac{1}{2}\|M\|_F^2 + \lambda_1 \sum_{i=1}^{N} \mathrm{KL}\left(p_i(\epsilon)\|p_0(\epsilon)\right) \\
&+\lambda_2 \sum_{t=1}^{T} \ell\Big(\mathrm{dist}_M^2(\mathbf{x}_i^t, \mathbf{x}_k^t) - \mathbb{E}[\mathrm{dist}_M^2(\mathbf{x}_i^t, \hat{\mathbf{x}}_j^t)]\Big) \ ,
\end{aligned}$$

where $\ell(x) = [1-x]_+$ is the hinge loss. Mahalanobis metric and instance disturbances are learned in an alternative manner.

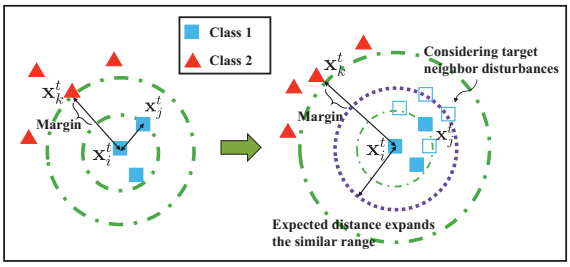


Figure 1: Illustration of DRIFT approach. The left plot shows the large margin requirement to optimize the metric: the distance between imposters should be larger than that between target neighbor with a margin. The right plot demonstrates the scenario when we consider the distribution/perturbation for a target neighbor, which expands the similar range if needed. Hollow blue squares are the perturbed target neighbor samples.

When M is fixed, the third part of the optimization problem is a linear optimization over distribution p_i, which tunes the perturbations under the guide of the current metric. When perturbation distributions $\mathbf{p}$ is fixed, the objective considers the influence of target neighbor by expected distance, and finds a global distance metric to push imposters farther away than target neighbors.

Fix metric M and solve distribution $\mathbf{p}$: We can write the sub-problem in the constraints form:

$$\begin{aligned}
&\min_{p_i(\epsilon)\in\mathcal{P}} \lambda_1 \sum_{i=1}^{N} \mathrm{KL}\left(p_i(\epsilon)\|p_0(\epsilon)\right) + \lambda_2 \sum_{t=1}^{T} \xi_t \ , \qquad (5) \\
&s.t.\ \forall t, \mathrm{dist}_M^2(\mathbf{x}_i^t, \mathbf{x}_k^t) - \mathbb{E}\left[\mathrm{dist}_M^2(\mathbf{x}_i^t, \hat{\mathbf{x}}_j^t)\right] \geq 1 - \xi_t,\ \xi_t \geq 0 \ .
\end{aligned}$$

With the convex property of KL-divergence, we can optimize Eq. 5 from dual. With non-negative multipliers $\boldsymbol{\alpha} = \{\alpha_t\}_{t=1}^T$ and $\boldsymbol{\beta} = \{\beta_t\}_{t=1}^T$, the dual problem can be written as:

$$\begin{aligned}
\max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \min_{p_i,\xi_t} &\lambda_1 \sum_{i=1}^{N} \mathrm{KL}(p_i(\epsilon)\|p_0(\epsilon)) + \lambda_2 \sum_{t=1}^{T} \xi_t - \sum_{t=1}^{T} \beta_t \xi_t \\
&- \sum_{t=1}^{T} \alpha_t \left(c_t - \mathbb{E}_{p_j^t}[\epsilon^\top M \epsilon] - 1 + \xi_t\right) \ , \\
&s.t.\ \ \forall i, p_i(\epsilon) \in \mathcal{P},\ \forall t, \alpha_t \geq 0,\ \beta_t \geq 0 \ . \qquad (6)
\end{aligned}$$

$c_t = \mathrm{dist}_M^2(\mathbf{x}_i^t, \mathbf{x}_k^t) - \mathrm{dist}_M^2(\mathbf{x}_i^t, \mathbf{x}_j^t) = \langle M, A_t \rangle$ is the difference of Mahalanobis distance with metric M between imposters and target neighbors, where $A_t = (\mathbf{x}_i^t, \mathbf{x}_k^t)(\mathbf{x}_i^t, \mathbf{x}_k^t)^\top - (\mathbf{x}_i^t, \mathbf{x}_j^t)(\mathbf{x}_i^t, \mathbf{x}_j^t)^\top$. The expectation $\mathbb{E}_{p_j^t}[\cdot]$ in Eq. 6 is taken over the distribution of the disturbance on $\mathbf{x}_j^t$. After applying stationarity property of KKT condition [Boyd and Vandenberghe, 2004] in Eq. 6, we can get $\lambda_2 - \alpha_t - \beta_t = 0$, thus $0 \leq \alpha_t \leq \lambda_2$. When taking derivative w.r.t. $p_i(\epsilon)$, we have

$$p_i(\epsilon) \propto \exp\left(-\frac{1}{2}\epsilon^\top (\Sigma_0^{-1} + \frac{2}{\lambda_1} \sum_{t=1}^{T} I_j^t \alpha_t M)\epsilon\right) \ . \qquad (7)$$

$I_j^t = I_j^t(\mathbf{x}_i)$ is the indicator whether perturbation distribution of target neighbor j in t-th triplet belongs to $\mathbf{x}_i$. Since p_i is a

valid distribution, we can get the normalization constant from its exponential form, which achieves a multivariate normal distribution. By defining $\Sigma_i^{-1} = \Sigma_0^{-1} + \frac{2}{\lambda_1}\sum_{t=1}^{T} I_j^t \alpha_t M$, we have $p_i(\epsilon) \sim \mathcal{N}(0, \Sigma_i)$. Because M is PSD, the updated covariance matrix is also PSD, meeting the requirement of a normal distribution. It is notable that distributions for perturbations on different instances differ in their ways combining dual variables. Due to complementary slackness, the value of α_t should be zero if a large margin is preserved with the expected distance, then the disturbance will be close to the prior. Otherwise, the distribution will adapt to the current measurement so as to change the weights on different constraints.

Substituting the distribution, we can simplify the dual problem to an optimization of f_1 on variable $\boldsymbol{\alpha}$:

$$\max_{\boldsymbol{\alpha}} f_1(\alpha) = \frac{\lambda_1}{2}\sum_{i=1}^{N}\log\det(\Sigma_i^{-1}) + \sum_{t=1}^{T}\alpha_t(1-c_t)\ , \quad s.t.\ 0 \le \alpha_t \le \lambda_2\ , \tag{8}$$

where $\det(\cdot)$ is the determinant of a matrix. From the smooth concave property of the $\log\det(\cdot)$ term, we can optimize the sub-problem for disturbance using accelerated projected gradient descent method [Nesterov, 2004, Li *et al.*, 2014]. The gradient w.r.t. α_t can be calculated as follows:

$$\frac{\partial f_1}{\partial \alpha_t} = \sum_{i=1}^{N} I_j^t \mathrm{Tr}\left((\Sigma_0^{-1} + \frac{2}{\lambda_1}\sum_{t=1}^{T} I_j^t \alpha_t M)^{-1} M\right) + (1 - c_t)\ . \tag{9}$$

Although there is a matrix inverse operation in Eq. 9, it can be further simplified as shown in the next sub-section.

Fix distribution $\mathbf{p}$ and solve metric M: The sub-problem for distance metric M can be formulated as:

$$\min_{M\in\mathcal{S}_d^+} \frac{1}{2}\|M\|_F^2 + \lambda_2 \sum_{t=1}^{T} \ell\left(\mathrm{dist}_M^2(\mathbf{x}_i^t, \mathbf{x}_k^t) - \mathbb{E}[\mathrm{dist}_M^2(\mathbf{x}_i^t, \hat{\mathbf{x}}_j^t)]\right)\ . \tag{10}$$

Since hinge loss is non-smooth, directly optimizing with sub-gradient descent will have slow convergence rate [Beck and Teboulle, 2009]. So we use a smooth approximation of hinge loss to accelerate the training of M:

$$\ell_s(x) = \frac{1}{L}\log\left(1 + \exp\left(-L(x-1)\right)\right)\ . \tag{11}$$

The larger the parameter L in Eq. 11, the more $\ell_s(x)$ close to the hinge loss [Zhang *et al.*, 2003, Qian *et al.*, 2015a]. With this smoothed loss, the above sub-problem over metric M is a convex smooth one, which can also be optimized with accelerated projected gradient descent method.

Given the learned perturbation distribution p_i in Eq. 7, we can compute the expectation over the quadratic form in Eq. 10 analytically. For a triplet $\{\mathbf{x}_i^t, \mathbf{x}_j^t, \mathbf{x}_k^t\}$, the expected term $\mathbb{E}_{p_j^t}[\epsilon^\top M \epsilon] = \langle \mathbb{E}_{p_j^t}[\epsilon\epsilon^\top], M\rangle = \langle \Sigma_j^t, M\rangle$. The covariance matrix Σ_j^t corresponds to the target neighbor j in triplet t, which can be estimated with learned $\boldsymbol{\alpha}$. If we denote the objective over M with smoothed loss as f_2, we can get the gradient w.r.t. metric M as:

$$\frac{\partial f_2}{\partial M} = M + \lambda_2 \sum_{t=1}^{T} \ell_s'(a_t)(A_t - \sum_{i=1}^{N} I_j^t \Sigma_i)\ ,$$

where $a_t = \mathrm{dist}_M^2(\mathbf{x}_i^t, \mathbf{x}_k^t) - \mathrm{dist}_M^2(\mathbf{x}_i^t, \mathbf{x}_j^t) - \mathbb{E}_{p_j^t}[\epsilon^\top M\epsilon]$ is the input distance value. $\ell_s'(a_t) = \frac{1}{1+\exp(-L(a_t-1))} - 1$ is the derivative value of the smoothed hinge loss.

3.4 Acceleration for DRIFT

Since M should be projected to the PSD cone to preserve a valid metric, we can use its eigen-decomposition $M = UDU^\top$ to further accelerate the optimization process over $\boldsymbol{\alpha}$. For M's symmetric property, its eigen vector U is an orthogonal matrix, and $D = \mathrm{diag}\{D_1, D_2, \ldots, D_d\}$ is a diagonal matrix containing its eigenvalues. In the following discussion, we set prior covariance $\Sigma_0 = \lambda I$.

To get the objective of the dual problem in Eq. 8 when solving the perturbation distribution, we need to compute

$$\mathcal{O}_1 = \log\det(\Sigma_0^{-1} + \frac{2}{\lambda_1}\sum_{t=1}^{T} I_j^t \alpha_t M)\ . \tag{12}$$

Since the determinant of a matrix equals to the product of its eigen-values, we can get $\mathcal{O}_1 = \sum_{d=1}^{D}\log(\frac{1}{\lambda} + q_t D_d)$ with $q_t = \frac{2}{\lambda_1}\sum_{t=1}^{T} I_j^t \alpha_t$ as the accumulated coefficient for each instance. Therefore, the computation of the matrix in Eq. 8 degrades to a scalar group computation problem.

To compute the gradients w.r.t. α_t, we need to get $\mathcal{O}_2 = \mathrm{Tr}((\Sigma_0^{-1} + q_t M)^{-1} M)$. Directly computing the trace term needs the inverse and multiplication of a $d \times d$ matrix. However, we can transform $\mathcal{O}_2$ as:

$$\mathcal{O}_2 = \mathrm{Tr}\left((\Sigma_0^{-1} + q_t M)^{-1} M\right) = \mathrm{Tr}\left(\frac{1}{q_t}(\frac{1}{q_t}\Sigma_o^{-1} + M)^{-1} M\right)$$
$$= \frac{1}{q_t}\mathrm{Tr}\left((I + \frac{1}{q_t}\Sigma_0^{-1} M^{-1})^{-1}\right) = \sum_{d=1}^{D}\frac{1}{q_t + \frac{1}{\lambda D_d}}\ .$$

The last equation comes from the fact that the trace of a matrix equals to the sum of its eigen-values. In summary, we transform the distribution optimization over $\boldsymbol{\alpha}$ to a problem only consisting of group of scalars computation with little computational cost.

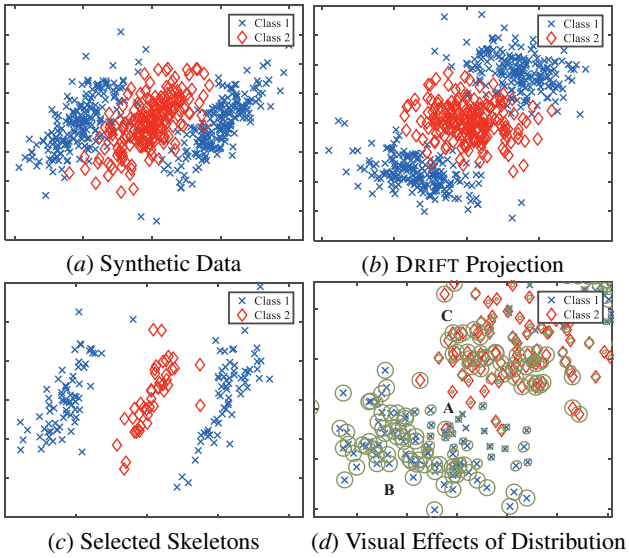
After the distribution $\mathbf{p}$ is known, we need to find $\Sigma_i = (\Sigma_0^{-1} + q_t M)^{-1}$ to complete the gradient computation when optimizing the metric M. We can rewrite the covariance matrix computation as:

$$\Sigma_i = \left(U(\mathrm{diag}(\frac{1}{\lambda}) + q_t D)U^\top\right)^{-1} = U\mathrm{diag}\left(\frac{\lambda}{1 + q_t D_d \lambda}\right)U^\top\ ,$$

which avoids the inverse computation. Operator $\mathrm{diag}(D_d)$ forms the variables over index d to a diagonal matrix.

The number of triplets increases when we meet large-scale datasets, and it is difficult to enumerate all triplets in a single gradient computation of M. Stochastic gradient descent can be a rescue [Qian *et al.*, 2015a], which can be used to reduce the computational burden in the metric sub-problem. In this case, we can consider an upper bound of the loss over the expected distance, where the disturbance of instances can be seamlessly imbedded in the stochastic gradient of metric. For the t-th triplet, using Jensen's inequality, we have $\ell(\mathrm{dist}_M^2(\mathbf{x}_i^t, \mathbf{x}_k^t) - \mathbb{E}[\mathrm{dist}_M^2(\mathbf{x}_i^t, \hat{\mathbf{x}}_j^t)]) = [1 - \mathrm{dist}_M^2(\mathbf{x}_i^t, \mathbf{x}_k^t) + \mathbb{E}[\mathrm{dist}_M^2(\mathbf{x}_i^t, \hat{\mathbf{x}}_j^t)]]_+ \le \mathbb{E}([1 - \mathrm{dist}_M^2(\mathbf{x}_i^t, \mathbf{x}_k^t) + \mathrm{dist}_M^2(\mathbf{x}_i^t, \hat{\mathbf{x}}_j^t)]_+)$. Therefore, we can optimize following objective upper bound for metric M:

$$\min_M \frac{1}{2}\|M\|_F^2 + \frac{\lambda_2}{T}\sum_{t=1}^{T}\mathbb{E}_{p_j^t}\left[[1 - \mathrm{dist}_M^2(\mathbf{x}_i^t, \mathbf{x}_k^t) + \mathrm{dist}_M^2(\mathbf{x}_i^t, \hat{\mathbf{x}}_j^t)]_+\right]\ ,$$



(*a*) Synthetic Data (*b*) DRIFT Projection

(*c*) Selected Skeletons (*d*) Visual Effects of Distribution

Figure 2: Visualization of DRIFT's property on synthetic data. Plots (a)-(d) show the original instances, projected instances, selected skeleton and the learned distribution (for left bottom instances), respectively.

where unbiased gradient can be computed by randomly choosing a triplet and disturbing the target neighbor with its known perturbation distribution.

4 Experiments

In this section, empirical investigations are conducted to validate the effectiveness of DRIFT. In detail, we first show the interpretability of the process of DRIFT on synthetic data, then DRIFT is compared with state-of-the-art methods on the real datasets. At last, we demonstrate the robustness of DRIFT given perturbed side information and instances.

4.1 Visualization on Synthetic Set

We first demonstrate the property of DRIFT on a 2D synthetic dataset. There are totally 600 instances with 2 classes. Class 1 is distributed in two different areas as in plot (a) of Fig. 2. We set the prior of DRIFT to $0.01I$. Using only a single metric, DRIFT clusters the same cluster instances together (plot (b)).

Plot (c) shows the instances in the original space who have zero dual variable sums. As in Eq. 7, when the sum of dual variables related to a particular instance is larger than zero, a difficult constraint is identified and the instance perturbation covariance will be compressed. Therefore, we can use the dual variable α_t to reflect the reliableness of side information to some extent and select the skeleton of data. We also give a visualization of learned distribution. Sizes of ellipsoids in the plot (d) are proportional to their covariances. The larger an ellipsoid, the wider the range of position the corresponding instance can drift. Instances in area "B" and "C" have large neighborhood range, which can satisfy the side information and enlarge the class boundary simultaneously. While instances near the class boundary (area "A") are hard to deal with. So compared with previous instances, they have smaller expected distance with others when they are selected as target neighbors, and therefore impose smaller weights on their related constraints.

4.2 Comparisons on Real-World Benchmarks

To test the classification ability of the learned metric for DRIFT, we compare the proposed DRIFT with state-of-the-art metric learning methods on 15 real datasets over 30 random trials. In each trial, 70% of training data is randomly selected, and the rest is used for test. Parameters are tuned for each method ranging from $\{10^{-2}, 10^{-1}, \ldots, 10^{2}\}$.

We compare with three parts of methods. First is the state-of-the-art metric learning methods, namely LMNN [Weinberger *et al.*, 2006], DNE [Zhang *et al.*, 2007], ITML [Davis *et al.*, 2007], GMML [Zadeh *et al.*, 2016] and RVML [Perrot and Habrard, 2015]. Second group including the ones weighting the side information in the training process, i.e., MSLMNN [Weinberger and Saul, 2009], LNML [Wang *et al.*, 2012] and MSML [Qian *et al.*, 2015b]. The last two methods consider noise/distribution in the distance computation: SGDD [Qian *et al.*, 2014] and MPME [Mao *et al.*, 2016]. The learned metric is validated using 3NN. The results with Euclidean distance is denoted as EUCLID. For our DRIFT approach, we test both performance of the batch and stochastic solver, which are shown as DRIFT_B and DRIFT_S respectively. In the implementation, we initialize metric $M = I$ and $\boldsymbol{\alpha}$ as zero vector. Triplets are initialized the same way as LMNN.

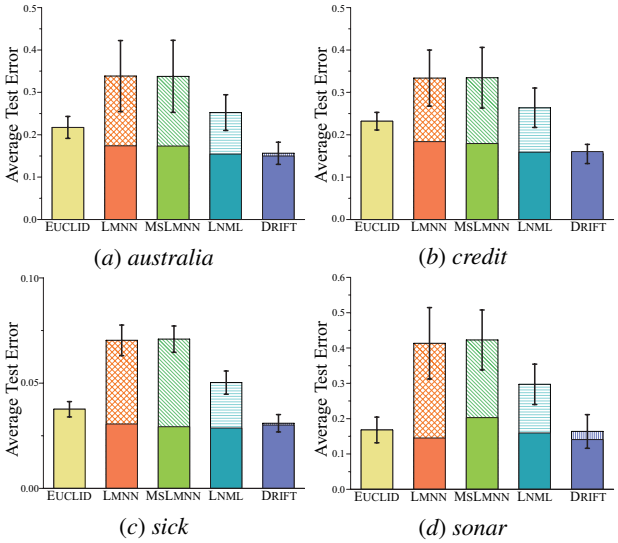
Average test errors of all methods are listed in Table 1. From the results, it can be found that the classification results for kNN can be improved with learned metrics, which shows the necessity and effectiveness of the metric learning. In addition, the methods considering the reliability of provided side information can give better results. For example, triplet selection method LNML gets better results than the non-selection counterpart LMNN. Although MPME considers instance distribution in the training process, it only uses the Euclidean distance as a learning guidance, so cannot perform well when the Euclidean one is not suitable. Our DRIFT approach can perform best on 9 of 15 datasets. Since it considers the instance disturbance, it identifies and takes advantages of useful side information constraints during the training. Compared with LNML, it can give even better results. Effectiveness of DRIFT can also be validated by its t-test comparison with others.

4.3 Investigations on Robustness

To test the robustness of DRIFT approach when dealing with noisy side information, we test DRIFT on above datasets with perturbed triplets constraints. The same partition as last subsection is used and parameters of all methods are fixed before training. For a triplet set $\{\mathbf{x}_i^t, \mathbf{x}_j^t, \mathbf{x}_k^t\}_{t=1}^T$ generated with 3 target neighbors and 10 imposters based on Euclidean nearest neighbor, we construct a noisy version by sampling 20% of them on which positions of $\mathbf{x}_j^t$ and $\mathbf{x}_k^t$ are exchanged. We compare our DRIFT method (batch solver) with LMNN, MSLMNN and LNML, since they obtain a metric from given fixed triplets. For the multi-stage method MSLMNN, we also corrupt its newly generated side information. The results of compared methods are listed in Fig. 3. Due to the page limit, only 4 of datasets are shown. Euclidean distance results are

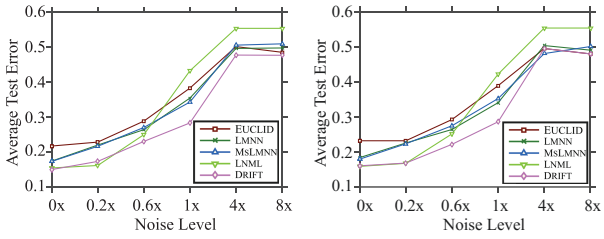
Table 1: Comparisons of classification performance (test errors, mean ± std.) based on 3NN. DRIFT$_B$ and DRIFT$_S$ are compared. The best performance on each dataset is in bold. Last two rows list the Win/Tie/Lose counts of DRIFT$_{B/S}$ against other methods on all datasets with t-test at significance level 95%.

Name	DRIFT$_B$	DRIFT$_S$	LMNN	DNE	ITML	GMML	RVML	LNML	MSLMNN	MSML	MPME	SGDD	EUCLID
australia	**.150±.022**	.174±.028	.174±.020	.217±.026	.175±.021	.203±.048	.157±.020	.155±.023	.173±.028	.162±.020	.249±.025	.233±.073	.217±.026
autompg	**.239±.032**	.255±.035	.259±.037	.272±.033	.266±.032	.259±.034	.294±.027	.262±.040	.243±.033	.334±.058	.295±.028	.276±.052	.260±.036
balance	**.068±.021**	.095±.028	.146±.028	.199±.019	.093±.022	.181±.018	.106±.021	.099±.016	.075±.017	.469±.105	.201±.016	.139±.026	.188±.022
credita	.160±.022	.181±.027	.184±.023	.232±.021	.178±.024	.212±.041	.162±.031	**.159±.019**	.179±.022	.167±.022	.251±.021	.205±.042	.232±.021
german	.278±.026	.281±.021	.292±.021	.296±.020	.295±.021	.284±.020	.280±.020	.284±.020	.297±.019	**.275±.018**	.317±.018	.299±.000	.296±.021
haberma	.293±.031	**.292±.034**	.300±.030	**.292±.032**	.311±.033	.313±.028	.316±.029	.316±.032	.296±.033	.316±.030	.314±.035	.608±.128	.304±.029
hayes-r	**.270±.051**	.278±.049	.314±.072	.411±.041	.315±.063	.385±.074	.330±.048	.275±.046	.278±.058	.397±.059	.373±.079	.421±.088	.398±.046
heart	.191±.026	.194±.027	.200±.031	.190±.034	**.187±.032**	.191±.034	.193±.036	.194±.042	.199±.036	.230±.045	.202±.031	.209±.046	.190±.034
heart-s	**.184±.030**	.190±.032	.195±.026	.188±.030	.187±.030	.191±.036	.193±.033	**.184±.036**	.207±.039	.233±.057	.219±.035	.212±.042	.188±.030
house-v	**.057±.017**	.065±.019	.060±.017	.083±.025	.058±.019	.078±.024	.069±.016	**.057±.020**	.066±.018	.058±.022	.063±.018	.064±.022	.083±.025
Live-di	.370±.042	.373±.038	.373±.045	.384±.040	.391±.052	.398±.041	.386±.040	.371±.043	.382±.044	.424±.043	.455±.048	**.368±.049**	.384±.040
promote	.106±.057	.136±.077	**.105±.037**	.249±.063	.147±.063	.394±.093	.121±.043	.107±.047	.122±.041	.107±.046	.375±.044	.169±.073	.249±.063
segment	**.032±.007**	.035±.007	.039±.006	.053±.008	.035±.006	.054±.008	.035±.006	**.032±.006**	.033±.007	.051±.011	.106±.008	.103±.056	.050±.007
sick	.030±.003	.031±.003	.031±.003	.038±.004	.038±.004	.056±.006	.050±.005	**.029±.005**	**.029±.004**	.033±.004	.048±.004	.083±.045	.038±.004
sonar	.141±.035	**.137±.042**	.145±.032	.168±.036	.170±.035	.210±.040	.236±.056	.160±.038	.203±.045	.200±.050	.183±.047	.162±.056	.168±.036
W / T / L	DRIFT$_B$ vs. others		8 / 7 / 0	12 / 3 / 0	12 / 3 / 0	12 / 3 / 0	8 / 7 / 0	4 / 11 / 0	6 / 9 / 0	11 / 4 / 0	14 / 1 / 0	14 / 1 / 0	13 / 2 / 0
W / T / L	DRIFT$_S$ vs. others		4 / 9 / 2	11 / 4 / 0	6 / 8 / 1	11 / 4 / 0	5 / 8 / 2	2 / 8 / 5	3 / 10 / 2	10 / 1 / 4	13 / 2 / 0	12 / 3 / 0	11 / 4 / 0



(*a*) *australia* (*b*) *credit* (*c*) *sick* (*d*) *sonar*

Figure 3: Investigation of corrupted side information. The results on noise-free datasets are filled with color, and the increases of error rates when training with noisy counterpart are denoted using shadow. Error bars in plots represent the 30 trials std. on corrupted datasets.



(*a*) *australia* (*b*) *credit*

Figure 4: Averaged test errors when different times of basic noises are added on these two datasets, where numerical value before "x" represents the multiplication of basic noise vector added.

also compared. Due to the fact that the corrupted information does not influence kNN, it can often get better results than others. It can be found clearly in Fig. 3 that the corrupted side information has a huge impact on the metric learning process. Both LMNN and MSLMNN get worse results than the Euclidean one, i.e., they learn a poor metric with corrupted constraints. LNML can relieve the negative variation, but DRIFT is almost not affected by this side information and can even train a good metric. It maybe in DRIFT the perturbations of instances takes different types of side information into consideration hence improve its robustness on average.

Performances with the change of noise level are also investigated. The maximum absolute values of each feature construct the basic noise vector, and times of it are added to the original datasets. Averaged test errors of different noise levels are recorded in Fig. 4. From the results, it is notable that DRIFT performs better than others under various noisy environment. In summary, these two performance comparisons validate the robustness of DRIFT, which strengthens the advantage of DRIFT in an unknown scenario.

5 Conclusion

We claim one of the prominent side information noise comes from the inaccuracies of feature values, which will damage the neighbor structure and seriously degenerate the robustness of metric learning approaches. Aiming at the noisy instance issues, Distance metRIc learning Facilitated by disTurbances (DRIFT) approach is proposed in this paper, which considers the perturbations on instance target pairs, to learn a robust metric. It is notable that expected distance for noisy instances is not only used for modeling types of feature value perturbations but also takes account of the constraints weights. Acceleration of DRIFT is also provided. Experiments on real datasets validate the effectiveness of DRIFT on classification performance. Results under noisy environments also highlight the DRIFT's superiorities. DRIFT can also be used to reveal the instance relationship for graph construction, which can be an interesting future work.

References

[Beck and Teboulle, 2009] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[Bellet *et al.*, 2015] A. Bellet, A. Habrard, and M. Sebban. *Metric Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2015.

[Bian and Tao, 2011] W. Bian and D. Tao. Learning a distance metric by empirical loss minimization. In *IJCAI*, pages 1186–1191, Barcelona, Spain, 2011.

[Boyd and Vandenberghe, 2004] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[Cao *et al.*, 2016] Q. Cao, Z.-C. Guo, and Y. Ying. Generalization bounds for metric and similarity learning. *MLJ*, 102(1):115–132, 2016.

[Chen *et al.*, 2014] N. Chen, J. Zhu, J. Chen, and B. Zhang. Dropout training for support vector machines. In *AAAI*, pages 1752–1759, Quebec, Canada, 2014.

[Chen *et al.*, 2015] M. Chen, K. Q Weinberger, Z. E. Xu, and F. Sha. Marginalizing stacked linear denoising autoencoders. *JMLR*, 16:3849–3875, 2015.

[Davis *et al.*, 2007] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *ICML*, pages 209–216, Corvalis, OR., 2007.

[Huang *et al.*, 2010] K. Huang, R. Jin, Z. Xu, and C.-L. Liu. Robust metric learning by smooth optimization. In *UAI*, pages 244–251, Catalina Island, CA., 2010.

[Kulis, 2012] B. Kulis. Metric learning: A survey. *Foundations and Trends® in Machine Learning*, 5(4):287–364, 2012.

[Law *et al.*, 2016a] M. T Law, N. Thome, and M. Cord. Learning a distance metric from relative comparisons between quadruplets of images. *International Journal of Computer Vision*, pages 1–30, 2016.

[Law *et al.*, 2016b] M. T Law, Y. Yu, M. Cord, and E. P Xing. Closed-form training of mahalanobis distance for supervised clustering. In *CVPR*, pages 3909–3917, Las Vegas, NV., 2016.

[Li *et al.*, 2014] N. Li, R. Jin, and Z.-H. Zhou. Top rank optimization in linear time. In *NIPS*, pages 1502–1510. Cambridge, MA.: MIT Press, 2014.

[Li *et al.*, 2016] Y. Li, M. Yang, Z. Xu, and Z. Zhang. Learning with marginalized corrupted features and labels together. In *AAAI*, pages 1251–1257, Phoenix, AZ., 2016.

[Lim *et al.*, 2013] D. Lim, G. Lanckriet, and B. McFee. Robust structural metric learning. In *ICML*, pages 615–623, Atlanta, GA., 2013.

[Luo *et al.*, 2016] Y. Luo, Y. Wen, and D. Tao. On combining side information and unlabeled data for heterogeneous multi-task metric learning. In *IJCAI*, pages 1809–1815, New York, NY., 2016.

[Mao *et al.*, 2016] Q. Mao, L. Wang, and I. W Tsang. A unified probabilistic framework for robust manifold learning and embedding. *MLJ*, pages 1–24, 2016.

[McFee and Lanckriet, 2010] B. McFee and G. R Lanckriet. Metric learning to rank. In *ICML*, pages 775–782, Haifa, Israel, 2010.

[Nesterov, 2004] Y. Nesterov. *Introductory lectures on convex optimization*, volume 87. Springer Science & Business Media, 2004.

[Perrot and Habrard, 2015] M. Perrot and A. Habrard. Regressive virtual metric learning. In *NIPS*, pages 1810–1818. Cambridge, MA.: MIT Press, 2015.

[Qian *et al.*, 2014] Q. Qian, J. Hu, R. Jin, J. Pei, and S. Zhu. Distance metric learning using dropout: a structured regularization approach. In *ACM SIGKDD*, pages 323–332, New York, NY., 2014.

[Qian *et al.*, 2015a] Q. Qian, R. Jin, J. Yi, L. Zhang, and S. Zhu. Efficient distance metric learning by adaptive sampling and mini-batch stochastic gradient descent (sgd). *MLJ*, 99(3):353–372, 2015.

[Qian *et al.*, 2015b] Q. Qian, R. Jin, S. Zhu, and Y. Lin. Fine-grained visual categorization via multi-stage metric learning. In *CVPR*, pages 3716–3724, Boston, MA., 2015.

[Van Der Maaten *et al.*, 2013] L. Van Der Maaten, M. Chen, S. Tyree, and K. Q Weinberger. Learning with marginalized corrupted features. In *ICML*, pages 410–418, Atlanta, GA., 2013.

[Verma and Branson, 2015] N. Verma and K. Branson. Sample complexity of learning mahalanobis distance metrics. In *NIPS*, pages 2584–2592. Cambridge, MA.: MIT Press, 2015.

[Wager *et al.*, 2013] S. Wager, S. Wang, and P. S Liang. Dropout training as adaptive regularization. In *NIPS*, pages 351–359. Cambridge, MA.: MIT Press, 2013.

[Wang *et al.*, 2012] J. Wang, A. Woznica, and A. Kalousis. Learning neighborhoods for metric learning. In *ECML/PKDD*, pages 223–236, Bristol, UK., 2012.

[Wangni and Chen, 2016] J. Wangni and N. Chen. Nonlinear feature extraction with max-margin data shifting. In *AAAI*, pages 2208–2214, Phoenix, AZ., 2016.

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

[Weinberger *et al.*, 2006] K. Q Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *NIPS*, pages 1473–1480. MIT Press, Cambridge, MA.: MIT Press, 2006.

[Xiang *et al.*, 2008] S. Xiang, F. Nie, and C. Zhang. Learning a mahalanobis distance metric for data clustering and classification. *Pattern Recognition*, 41(12):3600–3612, 2008.

[Xing *et al.*, 2003] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. In *NIPS*, pages 505–512. Cambridge, MA.: MIT Press, 2003.

[Ye *et al.*, 2016a] H.-J. Ye, D.-C. Zhan, and Y. Jiang. Instance specific metric subspace learning: A bayesian approach. In *AAAI*, pages 2272–2278, Phoenix, AZ., 2016.

[Ye *et al.*, 2016b] H.-J. Ye, D.-C. Zhan, X.-M. Si, and Y. Jiang. Learning feature aware metric. In *ACML*, pages 286–301, Hamilton, New Zealand, 2016.

[Zadeh *et al.*, 2016] P. H. Zadeh, R. Hosseini, and S. Sra. Geometric mean metric learning. In *ICML*, pages 2464–2471, New York, NY., 2016.

[Zhan *et al.*, 2009] D.-C. Zhan, M. Li, Y.-F. Li, and Z.-H. Zhou. Learning instance specific distances using metric propagation. In *ICML*, pages 1225–1232, Montreal, Canada, 2009.

[Zhang *et al.*, 2003] J. Zhang, R. Jin, Y. Yang, and A. G Hauptmann. Modified logistic regression: An approximation to svm and its applications in large-scale text categorization. In *ICML*, pages 888–895, Washington, D.C., 2003.

[Zhang *et al.*, 2007] W. Zhang, X. Xue, Z. Sun, Y.-F. Guo, and H. Lu. Optimal dimensionality of metric space for classification. In *ICML*, pages 1135–1142, Corvallis, OR., 2007.