Distance Metric Learning Using Dropout: A Structured Regularization Approach

Qi Qian[†], Juhua Hu[‡], Rong Jin[†], Jian Pei[‡] and Shenghuo Zhu[⋆]
[†]Department of Computer Science and Engineering
Michigan State University, East Lansing, MI, 48824, USA
[‡]School of Computing Science
Simon Fraser University, 8888 University Drive, Burnaby, BC, V5A 1S6
[⋆]NEC Laboratories America, Cupertino, CA, 95014, USA
{qianqi, rongjin}@cse.msu.edu, juhuah@sfu.ca, jpei@cs.sfu.ca, zsh@nec-labs.com

ABSTRACT

Distance metric learning (DML) aims to learn a distance metric better than Euclidean distance. It has been successfully applied to various tasks, e.g., classification, clustering and information retrieval. Many DML algorithms suffer from the over-fitting problem because of a large number of parameters to be determined in DML. In this paper, we exploit the dropout technique, which has been successfully applied in deep learning to alleviate the over-fitting problem, for DML. Different from the previous studies that only apply dropout to training data, we apply dropout to both the learned metrics and the training data. We illustrate that application of dropout to DML is essentially equivalent to matrix norm based regularization. Compared with the standard regularization scheme in DML, dropout is advantageous in simulating the structured regularizers which have shown consistently better performance than non structured regularizers. We verify, both empirically and theoretically, that dropout is effective in regulating the learned metric to avoid the over-fitting problem. Last, we examine the idea of wrapping the dropout technique in the state-of-art DML methods and observe that the dropout technique can significantly improve the performance of the original DML methods.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data Mining*; I.2.6 [**Artificial Intelligence**]: Learning

General Terms

Algorithms, Experimentation

Keywords

Distance Metric Learning, Dropout

KDD'14, August 24–27, 2014, New York, NY, USA.

http://dx.doi.org/10.1145/2623330.2623678.

1. INTRODUCTION

Learning a good distance metric is essential to distance based algorithms and applications, e.g., k-nearest neighbors, k-means clustering, content-based image retrieval (CBIR), etc. Distance metric learning (DML) aims to learn a linear mapping such that in the mapped space, examples from the same class are closer to each other than those from different classes. Many methods have been developed for DML [6, 7, 16, 26, 27] in the past, and DML has been successfully applied in various domains, including information retrieval [13], ranking [6], supervised classification [26], clustering [27], semi-supervised clustering [5] and domain adaptation [20].

One problem with DML is that since the number of parameters to be determined in DML is quadratic in the dimension, it may overfit the training data [26], and lead to a suboptimal solution. Although several heuristics, such as early stopping, have been developed to alleviate the over-fitting problem [26], their performance is usually sensitive to the setting of parameters (e.g. stopping criterion in early stopping), making it difficult for practitioners. Another problem with many existing DML methods is their high computational cost since they have to project intermediate solutions onto the Positive Semi-Definite (PSD) cone at every iteration to ensure that the learned metric is PSD. In [6], the authors show that it is possible to avoid the high cost of PSD projection by an one-projection paradigm that only needs to project the learned metric onto the PSD cone once at the end of the optimization algorithm. We adopt the one-projection paradigm in this paper to alleviate the high computational cost.

Recently, dropout has been found to be helpful in alleviating the over-fitting problem in training deep neural networks [14]. It randomly omits half of the feature detectors to prevent the complex co-adaptation between those neurons. The main intuition behind dropout is that it improves the robustness of individual neurons in order to avoid the over-fitting problem. Theoretical analysis about the regularization role of dropout in neural network can be found in [1]. Besides the success in deep learning, dropout has been applied to regression [24] in order to obtain robust feature representations. Features with artificial noises has been a classic topic in machine learning and data mining, and has been examined by many studies [17, 19] with the focus on additive noise that usually leads to a L_2 regularizer [2]. Wager *et al.* [25] analyze dropout within the framework of

regression and find that dropout is first-order equivalent to a L_2 regularizer for the dataset scaled by the inverse diagonal Fisher information matrix. Although dropout has received a lot of interests in machine learning and data mining community, to the best of our knowledge, this is the first study that exploits dropout for alleviating the over-fitting problem in DML.

In this paper, we, for the first time, introduce dropout to DML. Unlike previous studies on dropout that only apply dropout to training data, we apply dropout to both the learned metrics and the training data. By applying appropriate dropout probabilities to the learned metric, we show that dropout can be equivalent to Frobenius norm and L_p in expectation. In addition, we develop a structured regularizer using the dropout technique. Unlike the conventional regularizer that treats diagonal and off diagonal elements equivalently, the structured regularizer introduces different dropout probabilities for diagonal elements and off diagonal elements. To verify the effect of dropout in DML, we have conducted a comprehensive study that compares the dropout technique to other regularization techniques used in DML. Experimental results show that dropout significantly improves the classification performance on most datasets. In addition, we observe that dropout behaves like a trace norm based regularizer in DML when applied to training data: it controls the rank of the learned metric and leads to a skewed distribution of eigenvalues. This is in contrast to the previous studies that view dropout as a L_2 regularizer. Finally, we show that the dropout technique can be easily incorporated into the state-of-art DML methods and significantly improve their performance for most cases. The main contribution of this work is summarized as follows.

- We, for the first time, introduce dropout to DML and apply it to both the learned metric and the training data.
- We show that it is possible to construct structured regularizers using the dropout technique, and verify its performance, both theoretically and empirically.
- We apply the dropout to the state-of-art DML methods and show it can significantly enhance their performance.

The rest of the paper is organized as follows. Section 2 introduces the related DML methods and the analysis for dropout. Section 3 describes applying dropout to the learned metric and training data, respectively. Section 4 summarizes the results of the empirical study. Section 5 concludes this work and discusses the future directions.

2. RELATED WORK

Distance metric learning has been studied sufficiently during the past years [6, 7, 22, 26, 27] and detailed investigations could be found in the survey papers [15, 28]. The early works focus on optimizing pair-wise constraints [7, 27] to make sure the distance between examples from the same class is less than a pre-defined threshold while that from the different classes is larger than the other threshold. Nowadays, triplet constraints, where the distance of examples from the same class should be marginally smaller than that of examples from different classes, are preferred [6, 26] due to their superior performance compared with pair-wise constraints. More analysis shows that triplet constraints have the large margin property and could be explained as learning a set of local SVMs [8]. However, either taking pair-wise constraints or triplet constraints increases the number of training data exponentially, which significantly promotes the risk of over-fitting for DML. In fact, over-fitting phenomenon is reported by many DML methods [6, 26], and most of them try to alleviate it by PSD constraint. PSD constraint, on one hand, is the feasible set where the optimal metric should live in, and on the other hand, restricts the learned metric in the PSD cone to reduce the complexity of the model. Given a huge number of constraints, stochastic gradient descent (SGD) is widely used to learn the metric and PSD projection occurs at every iteration [23]. Unfortunately, the computational cost of PSD projection is cubic to the dimension of data, which significantly limits the application of DML in high dimensional datasets. Additionally, recent empirical study [6] demonstrates that one-projection paradigm, which only performs PSD projection once at the end of the algorithm, has the similar performance as projecting at every iteration. In this paper, we bring dropout to overcome over-fitting in DML. We adopt triplet constraints and one-projection paradigm setting, and show that dropout significantly improves the performance of existing DML methods.

Dropout is a technique developed for training deep neural networks [14]. Since deep neural network is a very complex model and is easy to overfit a small size of training data, dropout randomly omits half of neurons during training to limit the co-adaptation between neurons. Dropout makes sure that each learned neuron is robust. Besides deep learning, dropout has been introduced to more general regression task these years [24, 25]. Maaten *et al.* [24] use dropout to learn a robust feature representation for bag-of-words features, which significantly improves the performance over the original features. Although dropout belongs to the classic artificially corrupted features, it is better than other kinds of noises as reported in the work [24]. Recently, Wager *et al.* [25] analyze dropout within the framework of general regression task, and find that dropout is first-order equivalent to the L_2 regularizer on the rescaled data. The experiments in the study [25] demonstrate that dropout, as a data-dependent L_2 regularizer, outperforms the standard L_2 norm significantly. In this paper, we introduce dropout for DML. Unlike the previous works that focus on dropout in features [24, 25], we also apply dropout for the learned metric. We observe that dropout is equivalent to Frobenius norm, L_1 norm and trace norm with different dropout strategies. Empirical study validates the effectiveness of dropout in DML.

3. DML USING DROPOUT

Given the dataset $X = [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$, distance metric learning is to learn a good Mahalanobis distance metric M, so that for each triplet constraint $(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k)$, where x_i and x_j are in the same class and x_k is from a different class, we have

$$\text{dist}(\mathbf{x}_i, \mathbf{x}_k) - \text{dist}(\mathbf{x}_i, \mathbf{x}_j) > 1$$

where $\text{dist}(\mathbf{x}_i, \mathbf{x}_j)$ is the squared Mahalanobis distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ and is measured by

$$\text{dist}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^\top M(\mathbf{x}_i - \mathbf{x}_j)$$

Algorithm 1 SGD for DML

1: **Input:** Dataset $X \in \mathbb{R}^{d\times n}$, #Iterations T, Stepsize η
2: Initial M_0 as an identity matrix
3: **for** $t = 1, \cdots, T$ **do**
4: Randomly sample a triplet $\{\mathbf{x}_i^t, \mathbf{x}_j^t, \mathbf{x}_k^t\}$
5: $\boxed{M_t = M_{t-1} - \eta\nabla\ell}$
6: **end for**
7: **return** $\Pi_{psd}(\bar{M})$

Therefore, the objective optimization problem based on minimizing empirical risk could be written as

$$\min_{M\in S_+^d} \sum_t \ell(\langle A_t, M\rangle)$$

where S_+^d is the $d\times d$ symmetric PSD cone, $\ell(\cdot)$ is the convex loss function, and $A_t = (\mathbf{x}_i^t - \mathbf{x}_j^t)(\mathbf{x}_i^t - \mathbf{x}_j^t)^\top - (\mathbf{x}_i^t - \mathbf{x}_k^t)(\mathbf{x}_i^t - \mathbf{x}_k^t)^\top$. $\langle\cdot,\cdot\rangle$ denotes the dot product for matrix.

Since the number of triplets is very large (it can be as high as $\mathcal{O}(n^3)$), the optimization problem is usually solved by stochastic gradient descent (SGD) [7, 23]. Instead of projecting the learned metric onto PSD cone at every iteration, which can be an expensive operation, we adopt one-projection paradigm [6], which only projects the learned metric onto the PSD cone once at the end of iterations. Empirical studies have shown that one-projection-paradigm is significantly more efficient and yields the similar performance as SGD with PSD projection performed at every iteration. Therefore, in this work, we will focus on the following optimization problem without the PSD constraint.

$$\min_{M\in S^d} \sum_t \ell(\langle A_t, M\rangle)$$

Algorithm 1 gives the standard SGD algorithm for DML and dropout will be applied to perturb Step 5. We will discuss the details in the next section within this framework. In particular, we will discuss two different applications of dropout, i.e. application of dropout to the learned metric and application of dropout to the training data, in the following two subsections.

3.1 Applying Dropout to Distance Metric

In this section, we focus on applying dropout to the learned metric. Let M be the metric learned from the previous iteration. To apply the dropout technique, we introduce a Bernoulli random matrix $\delta = [\delta_{i,j}]_{i,j=1}^d$, where each $\delta_{i,j}$ is a Bernoulli random variable with $\delta_{i,j} = \delta_{j,i}$. Using the random matrix δ, we compute the dropped out distance metric, denoted by $\hat{M}$ as

$$\hat{M}_{i,j} = \delta_{i,j} M_{i,j}, \quad i,j = 1,\ldots,d$$

Note that by enforcing $\delta_{i,j} = \delta_{j,i}$, $\hat{M}$ is ensured to be a symmetric matrix. Below, we will discuss how to design the dropout probabilities for the Bernoulli random matrix δ to simulate the effect of Frobenius norm based regularization and L_1 norm based regularization, respectively.

3.1.1 Frobenius norm

Frobenius norm is the most widely used regularizer in DML [23, 26], and the standard DML problem with Frobenius norm is given by

$$\min_{M\in S^d} \frac{1}{T}\sum_{t=1}^T \ell(\langle A_t, M\rangle) + \frac{q}{2\eta}\|M\|_F^2 \tag{1}$$

The updating rule in SGD for Frobenius norm based regularization is

$$M_t = M_{t-1} - qM_{t-1} - \eta\nabla\ell_t(M_{t-1})$$

where $\ell_t(M) = \ell(\langle A_t, M\rangle)$.

Instead of using the regularizer directly, we could simulate the effect of Frobenius norm based regularization by applying dropout to the learned metric M_{t-1}. In particular, the Bernoulli random matrix δ is constructed by sampling each $\delta_{i,j:i\le j}$ independently from a Bernoulli distribution with $\Pr[\delta = 0] = q$ and setting $\delta_{j,i} = \delta_{i,j}$ to ensure that δ is symmetric. It is easy to verify that

$$E[\hat{M}_{t-1}] = (1-q)M_{t-1}$$

and the updating rule becomes

$$M_t = \hat{M}_{t-1} - \eta\nabla\ell_t(M_{t-1})$$

THEOREM 1. *Let M_* be the optimal solution output by Algorithm 1. Let $\bar{M}$ be the solution output by Algorithm 1 with dropout in Step 5 and q be the probability that dropout occurs in each item of the learned metric. Assume $|\mathbf{x}|_2 \le r$ and $q = 1/T$, we have*

$$E[\ell(\bar{M})] - \ell(M_*) \le \frac{1}{\eta T}\|M_*\|_F^2 + 8\eta r^2(1 + \frac{1}{T})$$

The detailed proof can be found in appendix. By setting the stepsize η as

$$\eta = \frac{\|M_*\|_F}{r\sqrt{8+8T}}$$

we have

$$E[\ell(\bar{M})] - \ell(M_*) \le \frac{4r\sqrt{2+2T}}{T}\|M_*\|_F = \mathcal{O}(1/\sqrt{T})$$

It is well known that $\mathcal{O}(1/\sqrt{T})$ is the minimax convergence rate for SGD, when the loss function is Lipschitz continuous [11, 12, 29]. As a result, with the appropriate choice of dropout probabilities, dropout will maintain the same convergence rate as the standard SGD method. We also notice that q is suggested to be set as $1/T$ in order to achieve $O(1/\sqrt{T})$ convergence. This result implies that dropout should not be taken too frequently, which is consistent with the analysis of other corrupted feature methods [2, 24]. Finally, since the derivation of convergence rates keep the same regardless of the sampling probabilities used in dropout, we thus will omit the analysis on convergence for the following cases.

3.1.2 L_1 norm

Besides Frobenius norm, L_1 norm also could be used in DML as

$$\min_{M\in S^d} \frac{1}{T}\sum_t \ell(\langle A_t, M\rangle) + \frac{q}{\eta}\|M\|_1$$

It is known as the composite optimization problem [18] and could be solved by iterative thresholding method

$$\begin{cases} M'_t = M_{t-1} - \eta\nabla\ell_t(M_{t-1}) \\ M_{t:i,j} = \text{sign}(M'_{t:i,j})\max\{0, |M'_{t:i,j}| - q\} \end{cases}$$

With different design of sampling probabilities, we can apply dropout to the learned metric to simulate the effect of L_1 regularization. In particular, we introduce a data-dependent dropout probability as

$$\Pr[\delta_{i,j} = 0] = \min\{1, \frac{q}{|M_{i,j}|}\}$$

Now, instead of perturbing M_{t-1}, we apply dropout to M'_t, i.e. the matrix after the gradient mapping. It is easy to verify that the expectation of the perturbed matrix $\hat{M}'$ is given by

$$E\left[[\hat{M'}_t]_{i,j}\right] = \begin{cases} [M'_t]_{i,j} - \text{sign}([M'_t]_{i,j})q & : \quad q \leq |[M'_t]_{i,j}| \\ 0 & : \quad q > |[M'_t]_{i,j}| \end{cases}$$

which is equivalent to the thresholding method stated above.

It is straightforward to extend the method to L_p norm

$$L_p(M) = (\sum_{i,j} |M_{i,j}|^p)^{1/p}$$

by setting the probability as

$$\Pr[\delta_{i,j} = 0] = \min\{1, \frac{q|M_{i,j}|^{p-2}}{(\sum_{i,j} M_{i,j}^p)^{1-1/p}}\}$$

Note that when $p = 1$, it is equivalent to the probability for L_1 norm.

3.1.3 Structured regularizer

Although these conventional regularizers have been applied for DML, they cannot exploit the structure of metrics sufficiently. Given a distance metric, the diagonal elements are more important than those from off diagonal. It is due to the fact that diagonal elements represent the importance of each feature (e.g., linear classifier) rather than the interactions between different features, and they also control the trace of the learned metric. Therefore, different regularizers should be assigned for diagonal and off diagonal elements, respectively. Fortunately, dropout can serve this purpose conveniently.

Given Q, which is a random matrix with each element from a uniform distribution in $[0, 1]$, we investigate the matrix

$$R = (Q + Q^\top)/2 \tag{2}$$

It is obvious that the diagonal elements of R are still from the same uniform distribution, while elements in off diagonal are from a triangle distribution with cumulative distribution function as

$$F(q) = \begin{cases} 2q^2 & : \quad 0 \leq q < 0.5 \\ 1 - 2(1-q)^2 & : \quad 0.5 \leq q \leq 1 \end{cases}$$

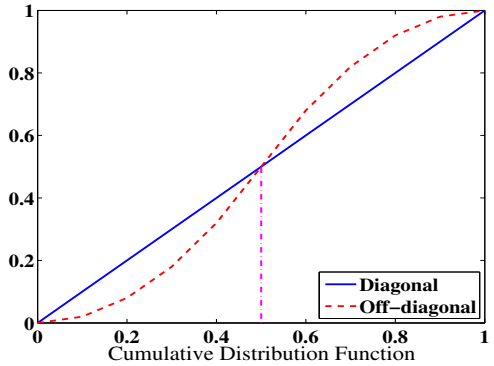
Figure 1 illustrates the cumulative distribution function for the diagonal elements of R and those living in off diagonal. The dropout probability based on the random matrix R is defined as

$$\Pr[\delta_{i,j} = 0] = \Pr[R_{i,j} \leq q]$$

First, we consider dropout with the same probability for each item of the metric as for Frobenius norm. Then, the probability of δ is

$$\Pr[\delta_{i,j} = 0] = \Pr[R_{i,j} \leq q] = \begin{cases} q & : \quad i = j \\ 2q^2 & : \quad i \neq j \end{cases} \tag{3}$$

Figure 1: Cumulative distribution function for diagonal elements and those in off diagonal.



Algorithm 2 Dropout as Structured Frobenius Norm (SGD-M1)

Input: Dataset $X \in \mathbb{R}^{d \times n}$, #Iterations T, Stepsize η
Initial M_0 as an identity matrix
for $t = 1, \cdots, T$ **do**
  Randomly sample a triplet $(\mathbf{x}_i^t, \mathbf{x}_j^t, \mathbf{x}_k^t)$
  Generate random matrix R as in Eqn. 2
  Generate dropout parameters δ by Eqn. 3
  Dropout: $\hat{M}_{t-1} = \delta M_{t-1}$
  $M_t = \hat{M}_{t-1} - \eta \nabla \ell$
end for
return $\Pi_{psd}(\bar{M})$

since $q = 1/T \ll 0.5$ as indicated in Theorem 1.

Therefore, the expectation of $\hat{M}_{t-1}$ is

$$\hat{M}_{t-1} = \begin{cases} M_{i,j}^{t-1} - qM_{i,j}^{t-1} & : \quad i = j \\ M_{i,j}^{t-1} - 2q^2 M_{i,j}^{t-1} & : \quad i \neq j \end{cases}$$

which is equivalent to solving the following problem

$$\min_{M \in S^d} \frac{1}{T} \sum_t \ell(\langle A_t, M \rangle) + \frac{q}{2\eta} \sum_i^d M_{i,i}^2 + \frac{q^2}{\eta} \sum_{i,j:i \neq j} M_{i,j}^2$$

It is obvious that the L_2 norm of diagonal elements in the metric is penalized quadratically more than those from off diagonal. This regularizer seems complex but the implementation by dropout is quite straightforward and Algorithm 2 summarizes the method.

Then, we consider dropout with the probability based on the elements as

$$\Pr[\delta_{i,j} = 0] = \Pr[R_{i,j} \leq \min\{1, \frac{q}{|M_{i,j}|}\}] \tag{4}$$

$$= \begin{cases} \min\{1, q/|M_{i,j}|\} & : \quad i = j \\ 2(q/|M_{i,j}|)^2 & : \quad i \neq j, q < 0.5|M_{i,j}| \\ 1 - 2(1 - q/|M_{i,j}|)^2 & : \quad i \neq j, 0.5 \leq q/|M_{i,j}| \leq 1 \\ 1 & : \quad q > |M_{i,j}| \end{cases}$$

It seems too complicated to analyze at the first glance, but Figure 1 could help us to understand the dropout strategy clearly. For the diagonal elements, they are actually shrunk by q as the L_1 regularizer. For the off diagonal elements, if $|M_{i,j}| > 2q$, the red dashed curve is under the blue solid one, which means the shrinkage is less than q. When $q \leq |M_{i,j}| \leq 2q$, the red dashed curve stands above the blue solid

Algorithm 3 Dropout as Structured L_1 Norm (SGD-M2)

Input: Dataset $X \in \mathbb{R}^{d\times n}$, #Iterations T, Stepsize η
Initial M_0 as an identity matrix
for $t = 1, \cdots, T$ **do**
  Randomly sample a triplet $(\mathbf{x}_i^t, \mathbf{x}_j^t, \mathbf{x}_k^t)$
  $M_t' = M_{t-1} - \eta\nabla\ell$
  Generate random matrix R as in Eqn. 2
  Generate dropout parameters δ by Eqn. 4
  Dropout: $M_t = \delta M_t'$
end for
return $\Pi_{psd}(\bar{M})$

one and the shrinkage on these elements is much faster than the standard L_1 norm. Since q is very small, most of the off diagonal elements have relatively larger values and will be shrunk slower than those with extremely small values. Algorithm 3 summarizes this method. Unlike Algorithm 2, dropout in Algorithm 3 is performed after updating with the current gradient.

3.2 Applying Dropout to Training Data

Besides dropout within the learned metric, in this section we apply dropout to the training data as many pervious studies [24, 25]. Since the analysis for data highly depends on the loss function, we take the hinge loss, which is the most widely used loss function in DML [6, 7, 23, 26], as an example.

Hinge loss is defined as $\ell(z) = [1+z]_+$, where $z = \langle A_t, M\rangle$ and

$$A_t = (\mathbf{x}_i^t - \mathbf{x}_j^t)(\mathbf{x}_i^t - \mathbf{x}_j^t)^\top - (\mathbf{x}_i^t - \mathbf{x}_k^t)(\mathbf{x}_i^t - \mathbf{x}_k^t)^\top$$

Obviously, the loss function penalizes A_t rather than the individual example, so dropout is taken according to the structure of A_t. To avoid affecting the decision of hinge loss, we perturb A_t after calculating the hinge loss.

We begin with additive noise as

$$\hat{A}_t = (\mathbf{x}_i^t - \mathbf{x}_j^t + \epsilon)(\mathbf{x}_i^t - \mathbf{x}_j^t + \epsilon)^\top - (\mathbf{x}_i^t - \mathbf{x}_k^t)(\mathbf{x}_i^t - \mathbf{x}_k^t)^\top \quad (5)$$

where $\epsilon \sim \mathcal{N}(0, qI_{d\times d})$. So the expectation of $\hat{A}_t$ is

$$\begin{aligned} E[\hat{A}_t] &= E[(\mathbf{x}_i^t - \mathbf{x}_j^t + \epsilon)(\mathbf{x}_i^t - \mathbf{x}_j^t + \epsilon)^\top] \\ &\quad -(\mathbf{x}_i^t - \mathbf{x}_k^t)(\mathbf{x}_i^t - \mathbf{x}_k^t)^\top \\ &= A_t + qI \end{aligned}$$

By replacing A_t in Prob. 1 with $\hat{A}_t$, the expectation of the problem becomes

$$\min_{M\in S^d} \sum_t \ell(\langle A_t, M\rangle) + \sum_{t:\ell_t>0} q\|M\|_{tr} \quad (6)$$

Note that the trace norm stands outside of the hinge loss, since the noise is added only after computing the hinge loss and only active constraints will contribute to the trace norm. We use the trace norm rather than the trace of the metric, because the final metric will be projected onto the PSD cone, where the trace of metric is equivalent to the trace norm. Algorithm 4 describes the details of the method.

Although the Guassian noise could perform as the trace norm, the external noise may affect the solution. Therefore, we consider dropout as

$$\hat{\mathbf{x}}_i^t = \delta_t \mathbf{x}_i^t, \quad \hat{\mathbf{x}}_j^t = \delta_t \mathbf{x}_j^t$$

Algorithm 4 Additive Noise as Trace Norm (SGD-D1)

Input: Dataset $X \in \mathbb{R}^{d\times n}$, #Iterations T, Stepsize η
Initial M_0 as an identity matrix
for $t = 1, \cdots, T$ **do**
  Randomly sample a triplet $(\mathbf{x}_i^t, \mathbf{x}_j^t, \mathbf{x}_k^t)$
  if $\ell(A_t, M_{t-1}) > 0$ **then**
    Generate a Guassian noise vector ϵ
    Add noise as in Eqn. 5
    $M_t = M_{t-1} - \eta\hat{A}_t$
  end if
end for
return $\Pi_{psd}(\bar{M})$

Algorithm 5 Dropout as Trace Norm (SGD-D2)

Input: Dataset $X \in \mathbb{R}^{d\times n}$, #Iterations T, Stepsize η
Initial M_0 as an identity matrix
for $t = 1, \cdots, T$ **do**
  Randomly sample a triplet $(\mathbf{x}_i^t, \mathbf{x}_j^t, \mathbf{x}_k^t)$
  if $\ell(A_t, M_{t-1}) > 0$ **then**
    Dropout as in Eqn. 7
    $M_t = M_{t-1} - \eta\hat{A}_t$
  end if
end for
return $\Pi_{psd}(\bar{M})$

where δ_t is a binary value random variable and

$$\Pr[\delta_t = 1 + q/(\mathbf{x}_i^t - \mathbf{x}_j^t)^2] = 1/(1 + q/(\mathbf{x}_i^t - \mathbf{x}_j^t)^2)$$

It is obvious that $E[\delta_t] = 1$ and $V[\delta_t] = 1 + q/(\mathbf{x}_i^t - \mathbf{x}_j^t)^2$. Similar to the additive noise, we only apply dropout for the first item of A_t as

$$\hat{A}_t = (\hat{\mathbf{x}}_i^t - \hat{\mathbf{x}}_j^t)(\hat{\mathbf{x}}_i^t - \hat{\mathbf{x}}_j^t)^\top - (\mathbf{x}_i^t - \mathbf{x}_k^t)(\mathbf{x}_i^t - \mathbf{x}_k^t)^\top$$

Note that when we perform dropout to the training data according to this strategy, we actually drop the rows and the corresponding columns in the first component $(\mathbf{x}_i^t - \mathbf{x}_j^t)(\mathbf{x}_i^t - \mathbf{x}_j^t)^\top$ of A_t. Since the expectation of random variables in diagonal is the variance and it is 1 in off diagonal, the expectation of $\hat{A}$ is

$$\begin{aligned} E[\hat{A}_t] &= E[(\hat{\mathbf{x}}_i^t - \hat{\mathbf{x}}_j^t)(\hat{\mathbf{x}}_i^t - \hat{\mathbf{x}}_j^t)^\top] - (\mathbf{x}_i^t - \mathbf{x}_k^t)(\mathbf{x}_i^t - \mathbf{x}_k^t)^\top \\ &= A_t + qI \end{aligned}$$

By taking $\hat{A}_t$ back to Prob. 1, we obtain the same problem as Prob. 6. Algorithm 5 summarizes this dropout strategy for training data.

THEOREM 2. *Let M_* be the optimal solution output by Algorithm 1. Let $\bar{M}$ be the solution output by Algorithm 5 and q be the probability that dropout occurs in each feature of the dataset. Assume $|\mathbf{x}|_2 \le r$ and $q = 1/T$, we have*

$$E[\ell(\bar{M})] - \ell(M_*) \le \frac{\|M_*\|_F^2}{2\eta T} + 8\eta r^2 + \frac{4\eta r^2}{T}(2d+1) + \frac{\|M_*\|_{tr}}{T}$$

The detailed proof is referred in appendix. If we set the stepsize η as

$$\eta = \frac{\|M_*\|_F}{2r\sqrt{4T + 4d + 2}}$$

we have

$$\ell(\bar{M}) - \ell(M_*) \leq \frac{1}{T}\left(2r\sqrt{(4T+4d+2)}\|M_*\|_F + \|M_*\|_{tr}\right)$$

where $\mathcal{O}(1/\sqrt{T})$ convergence rate, the well known result for standard SGD, is also observed as in Theorem 1.

According to Theorem 2, applying dropout to training data with the appropriate component and dropout probability does not hurt the convergence performance of standard SGD method too. Furthermore, q is required to be sufficiently small to avoid the suboptimal solution, which is also consistent with the analysis in Theorem 1.

4. EXPERIMENTS

4.1 Experiments Setting

Six datasets from different application scenarios are used to verify the effectiveness of the proposed method. Table 1 summarizes the information of these datasets. *ta* is a social network dataset with 6 different categories of terrorist attacks [21]. *semeion* is a handwritten digit dataset downloaded directly from the UCI repository [9]. *caltech10* is a subset of Caltech256 image dataset [10] with 10 most popular categories and we use the version pre-processed by the study [6], where each image is represented by an $1,000$-dimension vector. The other datasets are directly downloaded from LIBSVM database [4]. For *dna*, *protein* and *sensit*, we use the standard training/testing split provided by the original dataset. For the rest datasets, we randomly select 70% of data for training and use the remaining 30% for testing. For each dataset, we randomly select $T = 100,000$ active triplets (e.g., incur the positive hinge loss by Euclidean distance) within the range of 3-nearest same class neighbors as suggested by the study [26]. K-Nearest Neighbor (k=3) classifier is applied after obtaining the metric, since we optimize the triplets from 3-nearest neighbors. All experiments are repeated by 5 trials on different randomly generated triplet sets and the average result with standard deviation is reported.

4.2 Comparison with SGD Methods

In the first experiment, we compare the standard SGD for DML to five SGD variants including our proposed methods (i.e., SGD-M1, SGD-M2, SGD-D1, and SGD-D2). The methods are summarized as follows.

- **SGD**: stochastic gradient descent method as described in Algorithm 1.
- **SGD-PSD**: SGD with PSD projection at every iteration.
- **SGD-M1**: SGD with dropout for the learned metric as structured Frobenius norm (Algorithm 2).
- **SGD-M2**: SGD with dropout for the learned metric as structured L_1 norm (Algorithm 3).
- **SGD-D1**: SGD with additive Guassian noise in training data as trace norm (Algorithm 4).
- **SGD-D2**: SGD with dropout for training data as trace norm (Algorithm 5).

Euclidean distance is also included as the baseline method and denoted as "**Euclid**".

All of these SGD methods are applied on the same triplet set and take one-projection paradigm except SGD-PSD that projects the learned metric onto the PSD cone at every iteration. We search the stepsize η in $\{0.1, 1, 10\}$ by cross validation and $\eta = 1$ shows the best performance, so we fix it for all experiments. The dropout probability parameter q for the proposed methods is searched in $\{10^{-i} : i = 1, \cdots, 5\}$. All SGD methods are started with an identity matrix in the experiment.

Table 1: Statistics for the datasets used in the empirical study. #C is the number of classes. #F is the number of features. #Train and #Test represent the number of training data and test data, respectively.

	# C	# F	#Train	#Test
ta	6	106	902	391
semeion	10	256	1,115	478
dna	3	180	2,000	1,186
caltech10	10	1,000	3,151	1,338
protein	3	357	17,766	6,621
sensit	3	100	78,823	19,705

Table 2 shows the classification accuracy of different SGD methods. First, it is not surprising to observe that all DML algorithms improve the performance compared to the Euclidean distance. Second, for all datasets, we observe that the proposed SGD methods with dropout (i.e., SGD-M1, SGD-M2, SGD-D1, and SGD-D2) significantly outperform the baseline SGD methods (i.e., SGD and SGD-PSD), which is also demonstrated by the statistical significance examined via pairwise t-tests at the 5% significance level. Concretely, on most datasets, the accuracy of SGD with dropout is about 2% improved compared with that of SGD and it is even 4% on *protein*.

Furthermore, we observe that SGD-M1 shows the best performance on *semeion*, *caltech10* and *protein*, while SGD-M2 outperforms other methods on *ta* and *dna*, and SGD-D2 is the best on *sensit*. It is because dropout in the learned metric and dropout in the training data represent different regularizers, and different dataset prefers different regularizer. SGD-D1 and SGD-D2 have the similar performance because they optimize the same trace norm. However, SGD-D2 is a little bit better than SGD-D1 due to the reason that no additional Guassian noise is introduced by SGD-D2. Finally, SGD-PSD performs same as if not worse than SGD, which is consistent with the observation in the study [6].

Then, we investigate if dropout can perform as the regularizers as we expected. Figure 2 compares the effect of different norms with different weights to that of SGD, where only the parameter q of dropout varies and the others are kept the same. First, since SGD-M1 puts more aggressive penalty on the diagonal, Figure 2(a) shows how L_2 norm of the diagonal elements in the metric learned by SGD-M1 varies as q decreases. We observe that the smaller the parameter is, the larger the L_2 norm is, and even when $q = 10^{-5}$, the L_2 norm is still less than that of SGD. It demonstrates that dropout as structured Frobenius norm restricts the size of diagonal elements well. Second, Figure 2(b) compares the sparsity of the learned metric, where sparsity is defined as

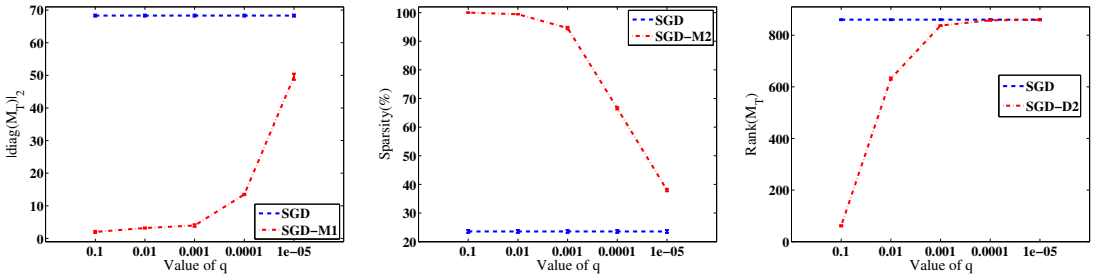
$$\text{Sparsity} = \frac{\#(M_{i,j} = 0)}{d^2}.$$

Without the constraint of L_1 norm, the sparsity of the metric learned by SGD is small as shown by the blue dashed line.

Table 2: Comparison of classification accuracy (%) for different SGD methods, and the best result is bolded (statistical significance examined via pairwise t-tests at the 5% significance level between baselines and the proposed methods).

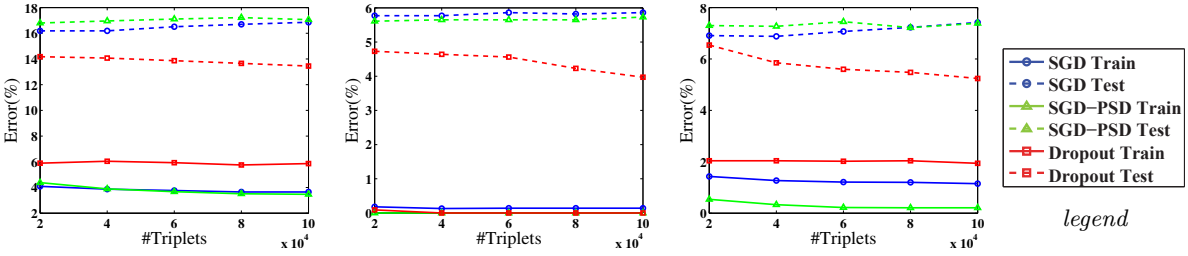
	ta	*semeion*	*dna*	*caltech10*	*protein*	*sensit*
Euclid	80.15	91.63	80.10	62.36	50.05	72.72
SGD	83.14±0.39	94.14±0.36	92.58±0.39	65.29±0.29	60.19±0.16	74.92±0.09
SGD-PSD	82.94±0.56	94.27±0.11	92.61±0.35	64.88±0.39	58.15±0.47	74.54±0.08
SGD-M1	85.77±0.42	**96.03±0.39**	93.86±0.33	**67.32±0.29**	**64.05±0.31**	76.33±0.10
SGD-M2	**86.55±0.22**	95.61±0.42	**94.76±0.59**	66.04±0.24	62.76±0.47	76.10±0.08
SGD-D1	84.33±0.69	95.31±0.55	93.36±0.41	65.85±0.46	61.36±0.57	76.82±0.05
SGD-D2	84.74±0.50	95.40±0.39	93.66±0.39	66.07±0.37	62.86±0.46	**76.89±0.07**

Figure 2: Trend of effect for dropout as regularizers on dataset *caltech*. Fig.(a) is the L_2 norm of the diagonal elements in the metric learned by SGD-M1. Fig.(b) is the sparsity of the metric learned by SGD-M2. Fig.(c) is the rank of the metric learned by SGD-D2.



(a) *SGD-M1* (b) *SGD-M2* (c) *SGD-D2*

Figure 3: Comparison of training error and test error of different SGD methods with different size of triplets. There is no over-fitting observed for SGD method with dropout.



(a) *ta* (b) *semeion* (c) *dna*

However, in SGD-M2 as plotted by the red dash dotted line, with the increasing of the probability of dropout as the structured L_1 norm, the learned metric becomes more sparse, which confirms the effectiveness of SGD-M2. Finally, trace norm constraint usually leads to a low-rank metric [3], so we study the rank of the learned metric by SGD-D2 in Figure 2(c). As expected, when q becomes larger, more stress is put on the trace norm and the lower-rank metric is induced.

Since dropout is found to be helpful to overcome over-fitting in deep learning [10], we empirically study the role of dropout for alleviating over-fitting problem in DML. We fix all parameters as above except the number of sampled triplets, to study the changes of training error and test error on the training and test set, with the increasing of the number of triplets. Figure 3 shows the training error and test error of SGD, SGD-PSD and SGD with best dropout strategy on three small datasets (i.e., SGD-M1 on *semeion* and SGD-M2 on the others), while the number of sampled triplets increases from $20,000$ to $100,000$. First, we observe that the training error of SGD with dropout is similar to that of conventional SGD methods as we indicate in Theorem 1. However, over-fitting is observed for SGD and SGD-PSD when the number of triplets is up to $40,000$, while there is no over-fitting phenomenon for SGD with dropout. It further demonstrates the overwhelming performance of dropout strategies in Table 2 and confirms that dropout is also helpful to overcome the over-fitting problem in DML.

4.3 Comparison with State-of-Art Methods

Besides the comparison with various SGD methods, we also compare our proposed dropout methods to three state-of-art DML algorithms as follows.

- **SPML** [23]: a mini-batch stochastic gradient descent algorithm for DML, which optimizes the hinge loss with Frobenius norm as the regularizer.

Table 3: Comparison of classification accuracy (%) with state-of-art DML methods. "Dropout" refers to the best result of dropout from Table 2. Note that LMNN is a batch learning algorithm, and there is no limitation for the triplets it uses and the number of PSD projections. The best result is bolded (statistical significance examined via pairwise t-tests at the 5% significance level).

	ta	*semeion*	*dna*	*caltech10*	*protein*	*sensit*
SPML	83.56±0.50	94.60±0.27	92.85±0.37	65.46±0.50	59.15±0.37	74.99±0.10
OASIS	83.20±0.95	94.06±0.19	88.57±0.28	65.06±0.40	58.83±0.38	73.50±0.15
LMNN	84.79±0.65	93.77±0.48	**95.13±0.26**	**67.17±0.49**	60.73±0.12	76.47±0.04
Dropout	**86.55±0.22**	**96.03±0.39**	**94.76±0.59**	**67.32±0.29**	**64.05±0.31**	**76.89±0.07**

Table 4: Comparison of classification accuracy (%) for SPML and its variants by wrapping different dropout strategies in. The best result is bolded.

	ta	*semeion*	*dna*	*caltech10*	*protein*	*sensit*
SPML	83.56±0.50	94.60±0.27	92.85±0.37	65.46±0.50	59.15±0.37	74.99±0.10
SPML-M1	84.59±0.38	**95.48±0.41**	93.27±0.16	**66.68±0.70**	61.38±0.36	75.63±0.08
SPML-M2	84.46±0.44	95.27±0.35	**93.91±0.32**	66.15±0.40	61.03±0.48	75.90±0.23
SPML-D	**84.74±0.56**	95.44±0.27	93.41±0.23	66.27±0.36	**62.14±0.68**	**76.84±0.13**

Table 5: Comparison of classification accuracy (%) for OASIS and its variants by wrapping different dropout strategies in. The best result is bolded.

	ta	*semeion*	*dna*	*caltech10*	*protein*	*sensit*
OASIS	83.20±0.95	94.06±0.19	88.57±0.28	65.06±0.40	58.83±0.38	73.50±0.15
OASIS-M1	84.74±0.86	**95.69±0.43**	93.35±0.33	**67.11±0.31**	62.44±0.55	75.47±0.15
OASIS-M2	**84.79±0.79**	94.73±0.38	**94.33±0.54**	66.12±0.49	62.61±0.60	75.47±0.12
OASIS-D	84.38±0.59	95.23±0.48	93.15±0.28	66.71±0.62	**63.06±0.55**	**76.56±0.15**

- **OASIS** [6]: an online learning approach for DML and the symmetric version is adopted in the comparison.
- **LMNN** [26]: a batch learning method with Frobenius norm for DML.

SPML and OASIS use the same triplet set as the proposed methods and also adopt one-projection paradigm. LMNN is a batch learning method, and thus there is no limitation for the type and the number of triplets that it could use for each iteration. Specifically, LMNN is not restricted to the set of triplets used by other methods. There is also no constraint for PSD projection in LMNN and it can perform PSD projection whenever it requires. All codes for these methods are from the authors and the recommended parameters are used. Since SPML is a stochastic method, it shares the same setting as the proposed methods, where the parameter for Frobenius norm is searched within the same range as q to obtain the best performance. SPML and OASIS are both initialized with an identity matrix, while LMNN starts with the matrix from PCA without dimension reduction, which usually has a better performance than the identity matrix for the method [26].

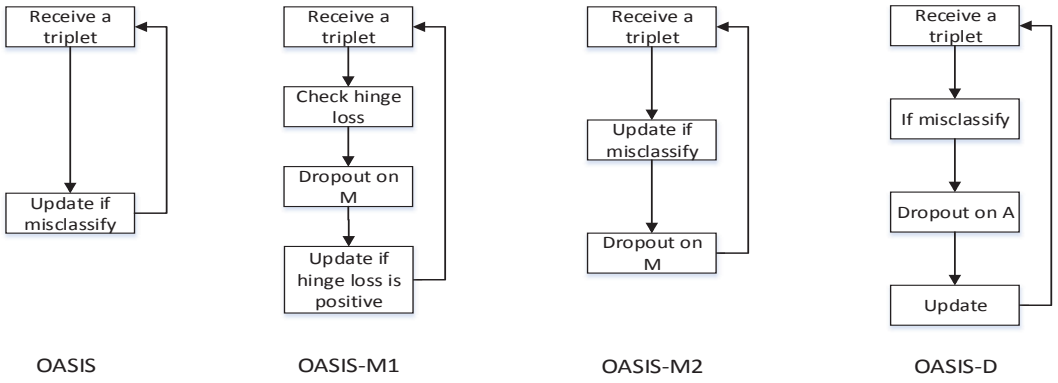
Table 3 summarizes the classification accuracy of different DML algorithms. "**Dropout**" denotes the best result of dropout methods adopted from Table 2. It can be easily observed that, although LMNN is a batch learning method and could utilize much more information than our methods, LMNN only has the similar performance on *dna* and *caltech10*, while SGD method with dropout significantly outperforms on all other datasets. It further demonstrates the effectiveness of the proposed methods. SPML and OASIS are slightly better than the standard SGD method, but significantly worse than SGD method with dropout technique. The performance of OASIS could be explained by the fact that it does not include any conventional regularizer and over-fitting could be easily induced. Although SPML combines the Frobenius norm as the regularizer, it is worse than SGD-M1 and SGD-M2 shown in Table 2, which implies that the proposed structured norm by dropout is more effective than the standard norm.

4.4 Wrap Dropout in Existing DML Methods

In this section, we demonstrate that dropout can be easily wrapped in the existing DML methods and help improve the performance. First, we wrap dropout in SPML, which is a state-of-art mini-batch SGD method for DML. Note that SPML has the Frobenius norm as the regularizer, so we drop it first to make sure that there is only one regularizer at one time. Since it is a SGD method, the dropout on M is the same as Algorithm 2 and Algorithm 3. We denote dropout as the structured Frobenius norm on SPML as "SPML-M1" and dropout as the structured L_1 norm as "SPML-M2". Instead of randomly selecting one triplet at each iteration, SPML samples b triplets at one time and updates according to the batch of the gradient. Therefore, when applying dropout to the training data, we simply perform the dropout on different matrix A in the mini-batch as in Algorithm 5 and the method is denoted as "SPML-D".

Table 4 summarizes the results for wrapped SPML methods. First, it is not surprising to observe that all dropout strategies improve the performance of SPML. On almost all datasets, the improvement on accuracy is more than 1% and it is even about 3% on *protein*, which is also consistent with the observation in the comparison for various SGD methods. Although SPML applies the standard Frobenius norm as the regularizer, SPML with different dropout strategies outperforms it significantly according to the statistical sig-

Figure 4: Procedure of wrapping dropout in OASIS.



nificance examined via pairwise t-tests at the 5% significance level, which shows the superior performance of the proposed structured regularizers.

Then, we wrap dropout in OASIS, which is a state-of-art online learning method for DML. Since online learning has the similar process as stochastic gradient descent method, wrapping dropout in is pretty straightforward. Figure 4 illustrates the procedures of wrapping different dropout strategies in OASIS. Let "OASIS-M1", "OASIS-M2","OASIS-D" denote dropout as the structured Frobeniuse norm, the structured L_1 norm and the trace norm in OASIS, respectively. The comparison of classification accuracy applied by 3-NN is summarized in Table 5. The similar phenomenon as for SPML is observed, that dropout always helps to improve the performance of OASIS significantly according to pairwise t-test at the 5% significance level.

In summary, wrapping dropout in existing DML methods is not only convenient but also very helpful for performance improvement.

5. CONCLUSION

In this paper, we propose two strategies to perform dropout for DML, i.e., dropout in the learned metric and dropout in the training data. For dropout in the metric, we propose the structured regularizer, which is simulated with dropout by assigning different dropout probabilities for the diagonal elements and those living in off diagonal. For dropout in the training data, the data-dependent dropout probability is adopted to mimic the trace norm. We develop the theoretical guarantees for both dropout scenarios to show that dropout will not affect the convergence rate of SGD with the appropriate dropout probability. Furthermore, we demonstrate that the proposed strategies are very convenient to wrap in the existing DML methods. Our empirical study confirms that the proposed methods have the overwhelming performance compared with the baseline methods, and could significantly improve the classification accuracy for the state-of-art DML methods. Since we currently apply dropout to the learned metric and the training data separately, we plan to examine the performance of the combination in the near future.

Acknowledgment

Qian and Jin are supported by ARO (W911NF-11-1-0383), NSF (IIS-1251031) and ONR Award (N000141410631). Hu and Pei's research is supported in part by a NSERC Discovery Grant and a BCIC NRAS Team project. All opinions, findings, conclusions and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

6. REFERENCES

[1] P. Baldi and P. J. Sadowski. Understanding dropout. In *NIPS*, pages 2814–2822, 2013.

[2] C. M. Bishop. Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1):108–116, 1995.

[3] E. J. Candès and T. Tao. The power of convex relaxation: near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

[4] C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. *ACM TIST*, 2(3):27, 2011.

[5] H. Chang and D.-Y. Yeung. Locally linear metric adaptation for semi-supervised clustering. In *ICML*, pages 153–160, 2004.

[6] G. Chechik, V. Sharma, U. Shalit, and S. Bengio. Large scale online learning of image similarity through ranking. *JMLR*, 11:1109–1135, 2010.

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

[8] H. Do, A. Kalousis, J. Wang, and A. Woznica. A metric learning perspective of svm: on the relation of lmnn and svm. In *AISTATS*, pages 308–317, 2012.

[9] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[10] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset, 2007.

[11] E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

[12] E. Hazan and S. Kale. Beyond the regret minimization barrier: an optimal algorithm for stochastic strongly-convex optimization. In *COLT*, pages 421–436, 2011.

[13] X. He, W.-Y. Ma, and H. Zhang. Learning an image manifold for retrieval. In *ACM Multimedia*, pages 17–23, 2004.

[14] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012.

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

[16] D. K. H. Lim, B. McFee, and G. Lanckriet. Robust structural metric learning. In *ICML*, 2013.

[17] K. Matsuoka. Noise injection into inputs in back-propagation learning. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(3):436–440, 1992.

[18] Y. Nesterov et al. Gradient methods for minimizing composite objective function, 2007.

[19] S. Rifai, X. Glorot, Y. Bengio, and P. Vincent. Adding noise to the input of a model trained with a regularized objective. *CoRR*, abs/1104.3250, 2011.

[20] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *ECCV*, pages 213–226. Springer, 2010.

[21] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

[22] S. Shalev-Shwartz, Y. Singer, and A. Y. Ng. Online and batch learning of pseudo-metrics. In *ICML*, 2004.

[23] B. Shaw, B. C. Huang, and T. Jebara. Learning a distance metric from a network. In *NIPS*, pages 1899–1907, 2011.

[24] L. van der Maaten, M. Chen, S. Tyree, and K. Q. Weinberger. Learning with marginalized corrupted features. In *ICML*, pages 410–418, 2013.

[25] S. Wager, S. Wang, and P. Liang. Dropout training as adaptive regularization. In *NIPS*, pages 351–359, 2013.

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

[27] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. J. Russell. Distance metric learning with application to clustering with side-information. In *NIPS*, pages 505–512, 2002.

[28] L. Yang and R. Jin. Distance metric learning: a comprehensive survery. 2006.

[29] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, pages 928–936, 2003.

APPENDIX

A. PROOF OF THEOREM 1

PROOF.

$$\begin{aligned}
&\|M_t - M_*\|_F^2 = \|\hat{M}_{t-1} - \eta A_t - M_*\|_F^2 \\
&= \|\delta M_{t-1} - (1-q)M_{t-1} + M_{t-1} - \eta A_t - M_* - qM_{t-1}\|_F^2 \\
&= (\delta-(1-q))^2\|M_{t-1}\|_F^2 + \|M_{t-1} - M_*\|_F^2 + \eta^2\|A_t\|_F^2 \\
&+q^2\|M_{t-1}\|_F^2 - 2\eta\langle A_t, M_{t-1} - M_*\rangle \\
&+2(\delta-(1-q))\langle M_{t-1}, M_{t-1} - \eta A_t - M_* - qM_{t-1}\rangle \\
&-2q\langle M_{t-1}, M_{t-1} - \eta A_t - M_*\rangle
\end{aligned}$$

Since the loss function is convex, we have

$$\begin{aligned}
&\ell(M_{t-1}) - \ell(M_*) \le \frac{1}{2\eta}(\|M_{t-1} - M_*\|_F^2 - \|M_t - M_*\|_F^2) \\
&+\frac{\eta}{2}\|A_t\|_F^2 + \frac{1}{2\eta}\left((\delta-(1-q))^2\|M_{t-1}\|_F^2 + q^2\|M_{t-1}\|_F^2\right. \\
&+2(\delta-(1-q))\langle M_{t-1}, M_{t-1} - \eta A_t - M_* - qM_{t-1}\rangle \\
&\left.-2q\langle M_{t-1}, M_{t-1} - \eta A_t - M_*\rangle\right)
\end{aligned}$$

Taking expectation on δ, we have

$$\begin{aligned}
&E[\ell(M_{t-1})] - \ell(M_*) \le \frac{1}{2\eta}(\|M_{t-1} - M_*\|_F^2 - \|M_t - M_*\|_F^2) \\
&+\frac{\eta}{2}\|A_t\|_F^2 + \frac{1}{2\eta}(q\|M_{t-1}\|_F^2 - 2q\langle M_{t-1}, M_{t-1} - \eta A_t - M_*\rangle) \\
&\le \frac{1}{2\eta}(\|M_{t-1} - M_*\|_F^2 - \|M_t - M_*\|_F^2) + \frac{\eta}{2}\|A_t\|_F^2 \\
&+\frac{q}{2\eta}(2\langle M_{t-1}, \eta A_t + M_*\rangle - \|M_{t-1}\|_F^2) \\
&\le \frac{1}{2\eta}(\|M_{t-1} - M_*\|_F^2 - \|M_t - M_*\|_F^2) + \frac{\eta}{2}\|A_t\|_F^2 \\
&+\frac{q}{2\eta}(\|\eta A_t + M_*\|_F^2) \\
&\le \frac{1}{2\eta}(\|M_{t-1} - M_*\|_F^2 - \|M_t - M_*\|_F^2) \\
&+\frac{(1+q)\eta}{2}\|A_t\|_F^2 + \frac{q}{2\eta}\|M_*\|_F^2 + q\ell(M_*)
\end{aligned}$$

Since $|\mathbf{x}| \le r$, $\|A_t\|_F \le 4r$. Adding iterations from 1 to T and setting $q = 1/T$, we have

$$E[\ell(\bar{M})] - (1-1/T)\ell(M_*) \le \frac{1}{\eta T}\|M_*\|_F^2 + 8r^2(1+1/T)\eta$$

□

B. PROOF OF THEOREM 2

PROOF.

$$\begin{aligned}
&\|M_t - M_*\|_F^2 = \|M_{t-1} - \eta\hat{A}_t - M_*\|_F^2 \\
&= \|M_{t-1} - M_*\|_F^2 + \eta^2\|\hat{A}_t\|_F^2 - 2\eta\langle\hat{A}_t, M_{t-1} - M_*\rangle
\end{aligned}$$

Taking expectation on $\hat{A}_t$, we have

$$\begin{aligned}
&\|M_t - M_*\|_F^2 \\
&= \|M_{t-1} - M_*\|_F^2 + \eta^2 E[\|\hat{A}_t\|_F^2] - 2\eta\langle A_t + qI, M_{t-1} - M_*\rangle
\end{aligned}$$

Since the loss function is convex, it is

$$\begin{aligned}
&E[\ell(M_{t-1})] - \ell(M_*) \le \frac{1}{2\eta}(\|M_{t-1} - M_*\|_F^2 - \|M_t - M_*\|_F^2) \\
&+\frac{\eta}{2}E[\|\hat{A}_t\|_F^2] + q(\|M_*\|_{tr} - \|M_{t-1}\|_{tr}) \\
&\le \frac{1}{2\eta}(\|M_{t-1} - M_*\|_F^2 - \|M_t - M_*\|_F^2) \\
&+\frac{\eta r^2}{2}(16 + 16dq + 8q) + q\|M_*\|_{tr}
\end{aligned}$$

where q's high order items are omitted since q is a small number. Sum over the iteration from 1 to T, we have

$$E[\ell(\bar{M})] - \ell(M_*) \le \frac{\|M_*\|_F^2}{2\eta T} + \eta r^2(8 + 8dq + 4q) + q\|M_*\|_{tr}$$

The proof is finished by setting $q = 1/T$. □