High-dimensional Similarity Learning via Dual-sparse Random Projection

Dezhong Yao[1], **Peilin Zhao**[2], **Tuan-Anh Nguyen Pham**[1] and **Gao Cong**[3]
[1] Rolls-Royce@NTU Corporate Lab, Nanyang Technological University, Singapore
[2] South China University of Technology; Tencent AI Lab, China
[3] School of Computer Science and Engineering, Nanyang Technological University, Singapore
dzyao@ntu.edu.sg, peilinzhao@hotmail.com, {gaocong, ntapham}@ntu.edu.sg

Abstract

We investigate how to adopt dual random projection for high-dimensional similarity learning. For a high-dimensional similarity learning problem, projection is usually adopted to map high-dimensional features into low-dimensional space, in order to reduce the computational cost. However, dimensionality reduction method sometimes results in unstable performance due to the suboptimal solution in original space. In this paper, we propose a dual random projection framework for similarity learning to recover the original optimal solution from subspace optimal solution. Previous dual random projection methods usually make strong assumptions about the data, which need to be low rank or have a large margin. Those assumptions limit dual random projection applications in similarity learning. Thus, we adopt a dual-sparse regularized random projection method that introduces a sparse regularizer into the reduced dual problem. As the original dual solution is a sparse one, applying a sparse regularizer in the reduced space relaxes the low-rank assumption. Experimental results show that our method enjoys higher effectiveness and efficiency than state-of-the-art solutions.

1 Introduction

Pairwise similarity learning has been widely used in classification, information retrieval, and recommendation systems [Chechik *et al.*, 2010; Bellet *et al.*, 2012; Cheng, 2013]. The continuous increase of data scales and dimensions in many applications (e.g. multimedia, finance, bioinformatics, and healthcare) raises two main challenges for similarity learning. First, high dimensionality poses computational and memory challenges. Second, large-scale data may be sparse and noisy, making it difficult to find any structure in the data [Fern and Brodley, 2003]. As a solution to these problems, random reduction techniques have received much attention recently [Paul *et al.*, 2013; Yang *et al.*, 2015; Xu *et al.*, 2017].

Random projection (RP) is a simple but powerful feature dimensionality reduction method that projects a high-dimensional sample (d dimensions) into a low-dimensional space (m dimensions) by a randomly generated matrix [Bingham and Mannila, 2001]. In contrast to other feature reduction methods, such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and undercomplete Independent Component Analysis (ICA), RP can more efficiently generate the projection matrix, so that it can avoid computing the eigenvalues of samples (with time complexity of $\mathcal{O}(d^3)$), which is too heavy as a data pre-processing step for large-scale datasets. However, RP does not consider the intrinsic structure of the data, so that it may lead to relatively high distortion [Fern and Brodley, 2003]. Specifically, the optimal solution solved in subspace can be used to recover a suboptimal solution in the original space, however it may significantly differ from the optimal solution of the original problem. Recently, **Du**al **R**andom **P**rojection (DuRP) algorithm is studied to recover the optimal solution in the original space [Zhang *et al.*, 2013; 2014; Yang *et al.*, 2015]. Specifically, by combining the Fenchel's duality theorem and random projection, Zhang [2013] proposed a stochastic DuRP solution, which can effectively restore the optimal solution of the original optimization problem. Currently, only a few studies examined the application of DuRP to similarity learning.

In this paper, we proposed an efficient ***dual random projection*** framework for high-dimensional similarity learning task. Our solution avoids the computational cost of $\mathcal{O}(d^3)$. This framework solves the optimization problem in two steps using recovery method. In step 1: RP is applied to the original data to reduce the dimensionality; then we need only solve a low-dimensional optimization problem. In step 2: the dual solution of the low-dimensional problem is constructed from its primal solution, and then we use the dual solution to recover the optimal solution of the original space. The empirical results show that the proposed framework achieves better and more robust performance.

It is notable that previous dual random projection and recovery methods rely on strong assumptions about the data—low rank [Paul *et al.*, 2013; Zhang *et al.*, 2014] or large separable margin [Balcan *et al.*, 2006; Shi *et al.*, 2012]—which may limit their application scenarios for similarity learning. To address this issue, we adopt a ***dual-sparse regularized random projection*** approach for high-dimensional similarity learning. In particular, a dual-sparse regularizer is added to the reduced optimization problem. Experiments on a set of real datasets demonstrate that a suitable

sparse regularizer can reduce the recovery error.

This paper is organized as follows. Related work is reviewed in the next section. Section 3 presents our dual-sparse random regularized random projection framework. Section 4 reports experimental results, and Section 5 concludes our work.

2 Related Work

Similarity/distance metric learning has been intensively studied [Bian and Tao, 2011; Kulis, 2013; Qian *et al.*, 2015]. Distance metric learning (DML) methods focus on learning a symmetric distance between two objects x_1 and x_2 by $(x_1 - x_2)^T M(x_1 - x_2)$, where the parametric matrix M must be positive semi-definite (PSD) [Shalev-Shwartz *et al.*, 2004; Jin *et al.*, 2009]. Another relative similarity learning (RSL) approach learns a similarity score of two objects by $\mathcal{S}_M(x_1, x_2) = x_1^\top M x_2$, where the similarity matrix M in the bilinear model can be non-symmetric [Chechik *et al.*, 2010; Crammer and Chechik, 2012]. Without the PSD constraint, the cost of optimization decreases from $\mathcal{O}(d^3)$ (PSD projection) to $\mathcal{O}(d^2)$. However, when solving the large-scale optimization problem, these two methods both have extremely high computational cost.

Random projection is one popular and efficient techniques to project high-dimensional data into a low-dimensional space and learn a metric in the subspace. It has been successfully applied in many applications, such as classification task [Zhang *et al.*, 2014], regression [Maillard and Munos, 2012; Zhang *et al.*, 2016], and information retrieval [Venna *et al.*, 2010]. However, this dimensionality reduction method often leads to suboptimal performance. To overcome this shortcoming, Davis [2008] and Weinberger [2009] place a strong assumption on the learned metric to be a low-rank matrix. This assumption significantly limits the applications.

Recently, an original space optimal problem recovering method is studied by using dual random projection [Qian *et al.*, 2015; Yang *et al.*, 2015]. Same as random projection method, DuRP reduces the dimensionality of the data and solves a subspace optimization problem. Then, DuRP constructs the dual solution of the subspace and uses it to recover the optimal solution of the original space problem. The recovered optimal solution achieves a small error by using $\Omega(r \log r)$ projections, where r is the rank of the data matrix [Zhang *et al.*, 2013].

Our work is related to the DuRP work [Qian *et al.*, 2015] for DML (DuRPDML). However, DuRPDML still needs to solve the PSD problem in the original high-dimensional space. The time complexity of PSD solution would be $\mathcal{O}(d^3)$, which can cause expensive computational cost for the high-dimensional dataset. The similarity learning method is more efficient and consumes less memory than DML in solving the high-dimensional optimization problem because it does not need the PSD step. Also, to recover the primal solution, DuRPDML assumes the data is low-rank. In contrast, our algorithm makes use of the sparsity of the dual solution to relax this assumption.

3 Dual-sparse Random Projection for Similarity Learning

Let $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_n) \in \mathbb{R}^{d\times n}$ denote a set of training examples. Our goal is to learn a similarity function $\mathcal{S} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ based on a sequence of training triplets $\mathcal{D} = \{(\mathbf{x}_t, \mathbf{x}_t^+, \mathbf{x}_t^-) \in \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^d | t \in [T]\}$ with $[T] = \{1, \cdots, T\}$, where the similarity score between $\mathbf{x}_t$ and $\mathbf{x}_t^+$ is greater than that between $\mathbf{x}_t$ and $\mathbf{x}_t^-$. Specifically, we would like to learn a similarity function $\mathcal{S}(\mathbf{x}, \mathbf{x}')$ that assigns higher similarity scores to more relevant instances, $\mathcal{S}(\mathbf{x}_t, \mathbf{x}_t^+) > \mathcal{S}(\mathbf{x}_t, \mathbf{x}_t^-), \quad \forall t \in [T]$.

For the similarity function, we adopt a parametric similarity function that has a *bi-linear* form: $\mathcal{S}_M(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top M \mathbf{x}'$, where $M \in \mathbb{R}^{d\times d}$. In order to learn the optimal parameter M, we introduce a loss function $\ell(M; (\mathbf{x}_t, \mathbf{x}_t^+, \mathbf{x}_t^-))$ that measures its performance on the t-th triplet. One popular loss function is the hinge loss: $\ell(M; (\mathbf{x}_t, \mathbf{x}_t^+, x_t^-)) = [1 - \mathcal{S}_M(\mathbf{x}_t, \mathbf{x}_t^+) + \mathcal{S}_M(\mathbf{x}_t, \mathbf{x}_t^-)]_+$, where $[\cdot]_+ = \max(0, \cdot)$. The above loss measures how much the violation of the desired constraint $\mathcal{S}(\mathbf{x}_t, \mathbf{x}_t^+) > \mathcal{S}(\mathbf{x}_t, \mathbf{x}_t^-)$ is by the similarity function defined by M.

A set of T triplet constraints is derived from the training examples in $\mathbf{X}$. During each learning step t, a triplet $(\mathbf{x}_t, \mathbf{x}_t^+, \mathbf{x}_t^-)$ will be presented to the algorithm. The algorithm can update the model from M_t to M_{t+1}, based on the current triplet. We denote the model's estimation after the *(t-1)*-th round by M_t. Following the empirical risk minimization framework, the optimal similarity function is learned by solving the following optimization problem:

$$\min_{M\in\mathbb{R}^{d\times d}} \tfrac{\lambda}{2}\|M\|_F^2 + \tfrac{1}{T}\sum_{t=1}^T \ell(M; (\mathbf{x}_t, \mathbf{x}_t^+, \mathbf{x}_t^-)), \tag{1}$$

where $\lambda > 0$ is the regularization parameter and $\ell(\cdot)$ is a convex loss function. We define the following hinge loss function for the triplet: $\ell(M; (\mathbf{x}_t, \mathbf{x}_t^+, \mathbf{x}_t^-)) = \max\{0, 1 - \mathcal{S}(\mathbf{x}_t, \mathbf{x}_t^+) + \mathcal{S}(\mathbf{x}_t, \mathbf{x}_t^-)\} = \max\{0, 1 - \langle M, A_t\rangle\}$, where $A_t = \mathbf{x}_t(\mathbf{x}_t^+ - \mathbf{x}_t^-)^\top$ and $\langle M, A_t\rangle = trace(M^\top A_t)$. Hence, the optimization problem can be described as solving:

$$M_* = \arg\min_{M\in\mathbb{R}^{d\times d}} \tfrac{\lambda}{2}\|M\|_F^2 + \tfrac{1}{T}\sum_{t=1}^T \ell(\langle M, A_t\rangle), \tag{2}$$

on triplet constraints $\{A_t\}_{t=1}^T$. By writing $\ell(\cdot)$ in its convex conjugate form $\ell_*(\cdot)$, we can turn the primal problem (2) into a dual problem:

$$\max_{\alpha_1,\cdots,\alpha_T} \tfrac{1}{T}\sum_{t=1}^T -\ell_*(-\alpha_t) - \tfrac{1}{2\lambda T^2}\left\|\sum_{t=1}^T \alpha_t A_t\right\|_F^2, \tag{3}$$

which is equivalent to

$$\boldsymbol{\alpha}_* = \arg\max_{\boldsymbol{\alpha}\in[0,1]^T} \tfrac{1}{T}\sum_{t=1}^T -\ell_*(-\alpha_t) - \tfrac{1}{2\lambda T^2}\boldsymbol{\alpha}^\top G\boldsymbol{\alpha}, \tag{4}$$

where $\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_T)^\top$ and $G = [G_{a,b}]_{T\times T}$ is a matrix in $\mathbb{R}^{T\times T}$ with $G_{a,b} = \langle A_a, A_b\rangle$. We denote $M_* \in \mathbb{R}^{d\times d}$ as the optimal primal solution to (2), and $\boldsymbol{\alpha}_* \in \mathbb{R}^T$ as the optimal dual solution to (4). Using the first order condition for optimality, we have:

$$M_* = \frac{1}{\lambda T}\sum_{t=1}^T \alpha_*^t A_t. \tag{5}$$

3.1 Dual Random Projection for Similarity Learning

It is a computationally challenge to solve either the primal problem in (2) or the dual problem in (4), when the dimensionality d is high and the number of training triplets T is

Algorithm 1 Dual **R**andom **P**rojection Method for **R**elative **S**imilarity **L**earning (**DuRPRSL**)

Input: the triplet constraints $\mathcal{D} = \{(\mathbf{x}_t, \mathbf{x}_t^+, \mathbf{x}_t^-)\}_{t=1}^T$ and the number of random projection m

1: Initialize a Gaussian random matrix $R \in \mathbb{R}^{d\times m}$
2: Project each triplet as $\widehat{A}_t = R^\top A_t R$
3: Solve the optimization problem (7) using SDCA and obtain the optimal solution $\widehat{\boldsymbol{\alpha}}_*$ (*)
4: Recover the original solution $\tilde{M}_*$ according to (9): $\tilde{M}_* = \frac{1}{\lambda T}\sum_{t=1}^T \widehat{\alpha}_*^t A_t$

Output:
the recovered solution $\tilde{M}_*$

large. We try to overcome this challenge by using the dual random projection technique. Let $R \in \mathbb{R}^{d\times m}$ be a Gaussian random matrix, where $m \ll d$ and $R_{i,j} \sim \mathcal{N}(0, 1/m)$. For each triplet constraint, we project its representation A_t into the low-dimensional space using a randomized matrix: $\widehat{A}_t = R^\top A_t R$. In the low-dimensional space, the primal goal is solving the following optimization problem with the randomly projected triplets $\{\widehat{A}_t\}_{t=1}^T$:

$$\widehat{M}_* = \arg\min_{\widehat{M}\in\mathbb{R}^{m\times m}} \frac{1}{T}\sum_{t=1}^T \ell(\langle \widehat{M}, \widehat{A}_t\rangle) + \frac{\lambda}{2}\|\widehat{M}\|_F^2. \quad (6)$$

The corresponding dual problem is written as

$$\widehat{\boldsymbol{\alpha}}_* = \arg\max_{\widehat{\boldsymbol{\alpha}}\in[0,1]^T} \frac{1}{T}\sum_{t=1}^T -\ell_*(-\widehat{\alpha}_t) - \frac{1}{2\lambda T^2}\widehat{\boldsymbol{\alpha}}^\top \widehat{G}\widehat{\boldsymbol{\alpha}}, \quad (7)$$

where $\widehat{G}_{a,b} = \langle R^\top A_a R, R^\top A_b R\rangle$. We denote $\widehat{\boldsymbol{\alpha}}_* \in \mathbb{R}^m$ as the optimal solution to (7) and the primal solution is

$$\widehat{M}_* = \frac{1}{\lambda T}\sum_{t=1}^T \widehat{\alpha}_*^t \widehat{A}_t. \quad (8)$$

Comparing the dual problem (4) in original space and problem (7) in subspace, the only difference is that the matrix $G_{a,b} = \langle A_a, A_b\rangle$ in (4) is replaced by $\widehat{G}_{a,b} = \langle R^\top A_a R, R^\top A_b R\rangle$ in (7). As $E(\langle R^\top A_a R, R^\top A_b R\rangle) \approx E(\langle A_a, A_b\rangle)$ when the reduced dimension m is sufficiently large, $\widehat{G}$ will be close to G, and $\widehat{\boldsymbol{\alpha}}_*$ is also expected to be close to $\boldsymbol{\alpha}_*$. As the result, we can use $\widehat{\boldsymbol{\alpha}}_*$ to approximate $\boldsymbol{\alpha}_*$ in (4). So the original optimal model can be recovered as:

$$M_* \approx \tilde{M}_* = \frac{1}{\lambda T}\sum_{t=1}^T \widehat{\alpha}_*^t A_t. \quad (9)$$

It is important to note that the recovered metric $\widehat{M}_*$ is the optimal solution in the projected subspace, while $\tilde{M}_*$ is computed directly in the original space using the approximate dual solution $\widehat{\boldsymbol{\alpha}}_*$. Compared to the original space optimization solving method in (3), the time complexity of the proposed method will drop to $\mathcal{O}(md)(\ll \mathcal{O}(d^2))$.

Solving the dual problem (7) is the key step in our framework. Here we choose the Stochastic Dual Coordinate Ascent (SDCA) [Shalev-Shwartz and Zhang, 2013] method to find the optimal $\widehat{\boldsymbol{\alpha}}_*$. It is shown to be empirically faster than other stochastic methods, especially for solving large-scale optimization learning problems [Shalev-Shwartz and Zhang, 2013]. Now the whole framework to obtain $\tilde{M}_*$ can be described in Algorithm 1. The SDCA solution in the projected space is described in Algorithm 2.

Algorithm 2 SDCA: Stochastic Dual Coordinate Ascent for Relative Similarity Learning

Input: $\lambda > 0$, the projected triplet constraints $\{\widehat{A}_t\}_{t=1}^T$

1: Initialize $\widehat{M}^0 = \widehat{M}(\widehat{\alpha}^{(0)}) = 0 \in \mathbb{R}^{m\times m}$, $\widehat{\alpha}^{(0)} = (\widehat{\alpha}_1, \cdots, \widehat{\alpha}_T)^\top = 0$
2: **for** $t = 1, 2, \ldots, T$ **do**
3: Uniformly pick a triplet $\widehat{A}_i$
4: Compute $\Delta\widehat{\alpha}_i$ to increase dual: $\Delta\widehat{\alpha}_i = \max\Big(0, \min\Big(1, \frac{1-\langle \widehat{M}^{(t-1)}, \widehat{A}_i\rangle}{\|\widehat{A}_i\|_F^2/(\lambda T)} + \widehat{\alpha}_i^{(t-1)}\Big)\Big) - \widehat{\alpha}_i^{(t-1)}$
5: $\widehat{\alpha}^{(t)} \leftarrow \widehat{\alpha}^{(t-1)} + \Delta\widehat{\alpha}_i e_i$
6: $\widehat{M}^{(t)} \leftarrow \widehat{M}^{(t-1)} + (\lambda T)^{-1}\Delta\widehat{\alpha}_i \widehat{A}_i$
7: **end for**

Output (Average option):
Let $\bar{\widehat{\alpha}} = \frac{1}{T-T_0}\sum_{i=T_0+1}^T \widehat{\alpha}^{(t-1)}$
Let $\bar{\widehat{M}} = \widehat{M}(\bar{\widehat{\alpha}}) = \frac{1}{T-T_0}\sum_{i=T_0+1}^T \widehat{M}^{(t-1)}$
return $\bar{\widehat{\alpha}}$

Output (Random option):
Let $\bar{\widehat{\alpha}} = \widehat{\alpha}^{(t)}$ and $\bar{\widehat{M}} = \widehat{M}^{(t)}$ for uniformly random $t \in \{T_0+1, \cdots, T\}$
return $\bar{\widehat{\alpha}}$

For the hinge loss, step 3(*) in SDCA has a closed form solution as

$$\Delta\alpha_i = \max\Big(0, \min\Big(1, \frac{1-\langle M^{(t-1)}, A_i\rangle}{\|A_i\|_F^2/(\lambda T)} + \alpha_i^{(t-1)}\Big)\Big) - \alpha_i^{(t-1)}. \quad (10)$$

The dual random projection and recovery approach has one deficiency: some non-support samples in the original optimization problem will become support samples, due to the feature reduction. This could result in the corruption of recover error. As a result, the dual recover methods rely on some strong assumptions of data: low rank [Paul *et al.*, 2013; Zhang *et al.*, 2014] or large separable margin [Balcan *et al.*, 2006; Shi *et al.*, 2012]. To address this limitation, we plan to use dual-sparse randomized reduction to relax the assumptions.

3.2 Dual-sparse Regularized Random Projection (DuSRPRSL) for Similarity Learning

To reduce the number of training instances that are non-support samples in the original optimization problem and transformed into support samples due to the reduction of the feature space, we add a dual-sparse regularization to the reduced dual problem (7). The optimal problem is written as

$$\widetilde{\boldsymbol{\alpha}}_* = \arg\max_{\boldsymbol{\alpha}\in[0,1]^T} \frac{1}{T}\sum_{t=1}^T -\ell_*(-\alpha_t) - \frac{1}{2\lambda T^2}\boldsymbol{\alpha}^\top \widehat{G}\boldsymbol{\alpha} - \frac{1}{T}\tau\|\boldsymbol{\alpha}\|_1, \quad (11)$$

where the regularizer is $\|\boldsymbol{\alpha}\|_1$ with $\tau > 0$.

To understand the effectiveness of dual-sparse regularizer in random projection for similarity learning, we evaluate the performance on a non-smooth hinge loss function $\ell(u) = \max(0, 1-u)$ and also give a solution for a smooth squared hinge loss function $\ell(u) = \max(0, 1-u)^2$. In this paper, the hinge loss function is chosen for illustration. Given $\ell_*(-\alpha_i) = -\alpha_i$ for $\alpha_i \in [0, 1]$, the new dual problem is written as:

$$\widetilde{\boldsymbol{\alpha}}_* = \arg\max_{\boldsymbol{\alpha}\in[0,1]^T} \frac{1}{T}\sum_{t=1}^T \alpha_t - \frac{1}{2\lambda T^2}\boldsymbol{\alpha}^\top \widehat{G}\boldsymbol{\alpha} - \frac{\tau}{T}\|\boldsymbol{\alpha}\|_1. \quad (12)$$

Using variable transformation $-\alpha_t \rightarrow \beta_t$, the above problem is written as

$$\max_{\beta\in[-1,0]^T} \frac{1}{T}\sum_{t=1}^{T}\beta_t(-1-\tau)-\frac{1}{2\lambda T^2}\beta^\top \widehat{G}\beta. \quad (13)$$

Changing into the primal form, we have

$$\max_{M\in\mathbb{R}^{m\times m}} \frac{1}{T}\sum_{t=1}^{T}\ell_{1-\tau}(\langle M^\top, A_t\rangle)+\frac{\lambda}{2}\|M\|_F^2, \quad (14)$$

where $\ell_\gamma(z) = max(0, \gamma - z)$ is a max-margin loss with margin given by γ.

For the hinge loss, step 3(*) in SDCA has a closed form solution as

$$\Delta\alpha_i=\max\left(0,\min\left(1,\frac{(1-\tau)-\langle M^{(t-1)},A_i\rangle}{\|A_i\|_F^2/(\lambda T)}+\alpha_i^{(t-1)}\right)\right)-\alpha_i^{(t-1)}. \quad (15)$$

For the squared loss, we have $\ell(u) = \max(0, 1 - u)^2$, a closed form solution is

$$\Delta\alpha_i=\frac{(1-\tau)-0.5\alpha_i^{(t-1)}-\langle M^{(t-1)},A_i\rangle}{0.5+\|A_i\|_F^2/(\lambda T)}. \quad (16)$$

The proposed dual-sparse formulation provides a bounding on the dual recovery error $\|\widetilde{\boldsymbol{\alpha}}_* - \boldsymbol{\alpha}_*\|$ to overcome the low-rank limitation, which is another contribution of this paper. As the margin becomes small in the reduction feature space, samples become difficult to separate after dimension reduced. A suitable sparse regularizer can help to improve the recovery accuracy. This is because adding the ℓ_1 regularization in the reduced problem of similarity learning is equivalent to using a max-margin loss with a smaller margin. The experiment results demonstrate that the recovery error can be reduced by a suitable sparse regularizer.

4 Experiments

In this section, we first present our study on DuRPRSL for ranking and classification tasks. Then we give a case study on the support of dual-sparse regularized DuRPRSL.

4.1 Experiment Datasets and Settings

To examine the effectiveness of the proposed method, we tested on six public datasets: *Protein*, *Gisette*, *RCV1*, *URL*, *Caltech256*, and *BBC* from LIBSVM, Caltech, and UCD, as shown in Table 1.

Data Set	Source	#Class	#Feature	#Train	#Test
Protein	LIBSVM	3	357	17,766	6,621
Caltech30	Caltech	30	1,000	20,623	8,838
Gisette	LIBSVM	2	5,000	6,000	1,000
BBC	UCD	5	9,636	1,558	667
RCV1	LIBSVM	2	47,236	20,242	677,399
URL	LIBSVM	2	3,231,961	1,677,291	718,839

Table 1: Statistics of standard datasets.

Caltech30 is a subset of the Caltech256 image dataset. We filtered out the 30 most popular categories. The *BBC* news article dataset was gathered from BBC news website. The *Protein* is a bioinformatics dataset, which is used to predict the local conformation of the polypeptide chain. The *Gisette* dataset is used for a handwritten recognition problem, released from NIPS 2003 Feature Selection Challenge. The *RCV1* is a binary text classification dataset. The *URL* dataset is used for malicious URL detection, consisting of 2.4 million URLs and 3.2 million features collected in 120-day. For evaluation, we used the standard training and testing split given by the providers, except for *Caltch30* and *BBC*. For these two datasets, we randomly split them into a training set (70%) and a test set (30%). To generate a triplet $(\mathbf{x}_t, \mathbf{x}_t^+, \mathbf{x}_t^-)$, $\mathbf{x}_t$ is firstly randomly selected from the whole training set, then $\mathbf{x}_t^+$ is randomly selected from the subset of training set, which consists of the examples with the same class of $\mathbf{x}_t$, at last, $\mathbf{x}_t^-$ is randomly selected from the rest of training set, which consists of the examples with different classes of $\mathbf{x}_t$.

To make a fair comparison, all methods adopted the same experimental setup. We randomly selected T=$50,000$ triplets as training instances and set the number of epochs to be 5 for all stochastic methods. The average results over five trials were reported finally. Cross-validation was used to select the values of hyperparameters for all algorithms. Specifically, the parameters set by cross-validation included: the aggressiveness parameter C for OASIS ($C \in \{1, 0.1, 0.08, ..., 0.01\}$) and $\lambda \in \{$5e-2, 5e-1, ..., 5e+6$\}$. Moreover, the hinge loss was used in the implementation of the proposed algorithms.

To evaluate the quality of learned metrics, we first used mean-average-precision (MAP) to evaluate the accuracy of the retrieval performance. Second, we evaluated the classification performance using k-nearest neighbor classifier. More specifically, we applied the proposed algorithms to learn a measurement metric. Then for each test instance $\mathbf{q}$, we used the learned metric to find the top k-nearest training examples and predicted the class assignment by choosing the majority class among the k-nearest neighbors. We also recorded the time cost of the proposed algorithms.

4.2 Comparison Algorithms

We compared eight approaches. **Euclidean**: The baseline measurement method using the standard Euclidean distance in feature space. **OASIS**: A state-of-the-art algorithm learns a bilinear similarity, which is based on online passive-aggressive algorithm using triplet instances. [Chechik *et al.*, 2010]. **DuDML**: This algorithm applies Stochastic Dual Coordinate Ascent (SDCA) [Shalev-Shwartz and Zhang, 2013] to learn the distance metric. **DuRSL**: This algorithm applies SDCA to solve the dual problem Eq.(4) and recover the similarity metric by Eq.(5). **SRP**: Apply random projection to reduce the dimensionality and then use SDCA to learn the similarity metric in subspace. **SPCA**: Apply PCA to project original data into lower dimensional space and then use SDCA to learn the similarity metric in subspace. **DuRPDML**: Dual Random Projected Distance Metric Learning proposed by [Qian *et al.*, 2015]. **DuRPRSL**: The proposed algorithm (Algorithm 1).

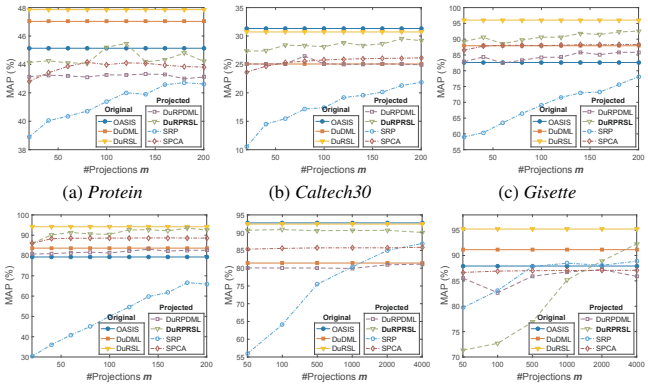
OASIS, DuDML, and DuRSL are the algorithms that solve the optimization problem in the original space. DuRPDML, DuRPRSL, SRP, and SPCA are the methods that apply RP or PCA and solve the optimization problem in subspace. Moreover, DuRPDML and DuRPRSL will recover the solution in the original space by using the suboptimal results.

4.3 Evaluation by Ranking

To demonstrate the effectiveness of the proposed method, we first compared the MAP performance for different approaches, where the dimension of the subspace projection is set as 100. The results are summarized in Table 2. Then,

	Metric in Original Space						Metric in SubSpace	
	Euclidean	OASIS	DuDML	DuRSL	DuRPDML	DuRPRSL	SRP	SPCA
Protein	38.78	45.18±0.12	47.05±0.31	49.30±0.24	43.35±0.16	45.47±0.08	41.38±0.29	43.98±0.15
Caltech30	17.45	31.37±0.03	28.39±0.37	30.71±0.40	25.05±0.18	28.43±0.13	17.37±0.43	25.78±0.11
Gisette	62.05	82.62±0.22	88.92±0.53	95.10±0.61	84.26±0.67	91.01±0.66	69.10±2.35	87.95±0.09
BBC	31.75	79.36±0.42	83.39±0.35	94.02±0.70	81.54±0.31	90.45±0.78	50.08±1.52	88.58±0.08
RCV1	60.50	92.75±0.03	81.44±0.78	92.41±0.53	80.05±0.82	90.62±0.54	60.41±2.33	85.57±0.17
URL	67.21	87.92±0.24	91.15±0.02	94.04±0.03	82.67±0.42	72.68±0.25	83.09±0.20	86.92±0.03

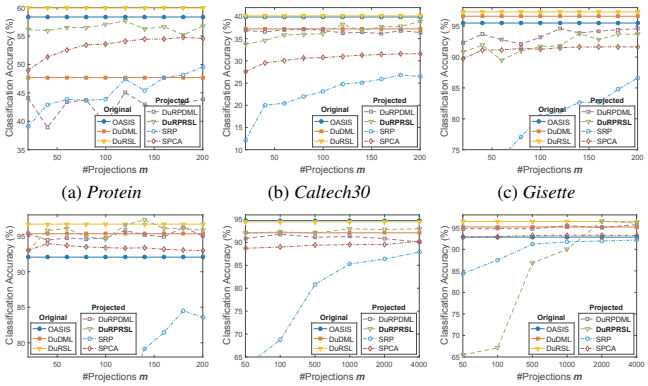
Table 2: Comparison of ranking results by MAP (%) (dim=100).



(a) *Protein* (b) *Caltech30* (c) *Gisette*

(d) *BBC* (e) *RCV1* (f) *URL*

Figure 1: Performance of ranking algorithms with different number of projection dimensions in terms of MAP.



(a) *Protein* (b) *Caltech30* (c) *Gisette*

(d) *BBC* (e) *RCV1* (f) *URL*

Figure 2: Performance of classification algorithms with different number of projection dimensions in terms of k-NN accuracy.

in order to observe the effect of different dimensions on the MAP performance, we varied the dimension of the subspace from 20 to 200 on each dataset. Fig. 1 shows how the MAP results are affected by the projection dimension. In average, DuRPRSL provides at least 5% improvement than DuRPDML. The proposed dual optimal solution DuRPRSL is also quite close to the primal optimal solutions.

For the first experiment, Table 2 shows the MAP results of different learning approaches. First, we observe that RSL-based methods perform better than DML-based methods, and DuRSL achieves the best performance among all the methods. Second, compared with DuRSL, DuRPRSL achieves similar performance on all datasets except *URL*, which resulted from an extremely dimension reduction, from 3 million to 100. When the dimension increases to 2000, we can see that DuRPRSL works best among all algorithms in Fig. 1(f). With the growth of dimensions, the MAP is increasing.

In the subspace, SPCA performs significantly better than SRP, while SRP is slightly better than the baseline method. That is because SPCA keeps the maximal data variance when making projections. Compared with methods in original space, SPCA achieves similar performance as DuRPDML.

For the second experiment, we compared the MAP performance changes with different projection dimensions in Fig. 1. We set the number of random projection dimensions from 20 to 200 for datasets *Protein*, *Caltech30*, *Gisette*, and *BBC*. For *RCV1* and *URL*, we changed the number of random projection dimensions from 50 to 4000 as their feature space are larger. Note that the performance of OASIS, DuDML, and DuRSL remain unchanged with the varied number of dimensions because they do not apply reduction. We can observe that DuRSL almost achieves the best performance of all the datasets.

Of the methods that use random projection, DuRPRSL almost performs the best for different projection dimensions and datasets except for the *URL* dataset. DuRPRSL becomes better than other random projection methods after the number of random projections reaches 2000, while SRP performs better than DuRPRSL when the dimension is less than 2000. That is because SRP projects all the data into a low-dimensional space which maximizes the variance of the low-dimension datasets and solves the optimization problem in the space. DuRPRSL needs to solve the original space optimization problem. Although RP is more computationally efficient than PCA, it often yields worse performance than PCA unless the number of random projections is sufficiently large [Fradkin and Madigan, 2003].

In addition, when comparing SRP and SPCA, the performance of SRP improves with an increasing number of projections, while the performance of SPCA is not significantly affected by the projection dimensions. That is because the principal components are already captured in the subspace.

4.4 Evaluation by Classification

In this experiment, we evaluate the learned metric by its classification application accuracy with k-NN (k=5) classifier. The k-NN accuracy changes in different projection dimensions, which is shown in Fig. 2. From the results, we can see that DuRSL has the best performance and DuRPRSL is similar. Totally, we essentially have the same observation as that for the ranking experiments reported in Section 4.3. DuRPRSL provides at least 10% accuracy improvement than the PR methods for the high-dimensional datasets. DuRPRSL is also quite close to the primal solution DuRSL, which indicates that the recovery error is small.

Comparing DuRPRSL and DuRPDML, DuRPRSL also achieves 5% accuracy improvement except for the dataset

			Protein	Caltech30	Gisette	BBC	RCV1	URL
Metric in Original Space	OASIS		163.18±12.62	844.74±57.54	11142.05±306.66	6952.13±560.81	22767.31±1303.77	16086.49±2253.54
	DuDML		134.18±26.24	739.08±71.54	33994.66±505.87	341260.11±810.73	383310.98±2851.23	560486.51±3053.87
	DuRSL		65.62±1.62	541.55±45.05	29443.66±155.07	246310.02±959.40	158217.06±821.51	537627.78±493.84
	DuRPDML	learning	13.12±0.62	11.59±1.87	12.11±1.43	11.69±0.76	12.98±0.13	11.70±0.83
		recovering	58.76±0.46	166.15±24.33	2644.12±193.92	13106.39±434.72	2718.18±87.93	2185.98±116.92
	DuRPRSL	learning	10.28±0.58	9.48±0.28	9.84±0.52	9.95±0.41	10.55±0.13	9.61±0.04
		recovering	30.92±0.13	110.41±1.50	1123.22±37.69	5817.32±719.52	1093.59±141.88	1155.52±217.48
Metric in SubSpace	SRP		21.41±28.05	21.85±6.65	20.33±2.18	22.09±4.04	11.27±1.61	9.95±2.33
	SPCA		15.33±3.53	10.02±0.07	8.84±0.12	8.84±0.12	10.15±0.98	10.99±0.52

Table 3: Learning time (seconds) used by different approaches.

Gisette. The dual recovery framework we proposed is mainly concerned with time efficiency. For the dataset *Gisette*, DuRPRSL achieves similar performance as DuRPDML, but DuRPRSL has small computational overhead than DuRPDML as shown in Table 3.

4.5 Learning Efficiency of DuRPRSL

In this section, we compare the computational cost of all the learning algorithms, which is recorded in Section 4.3. The dimension of the subspace projection was set to 100 and we choose the optimal performance results for each algorithm. The learning time is recorded by CPU time (in seconds).

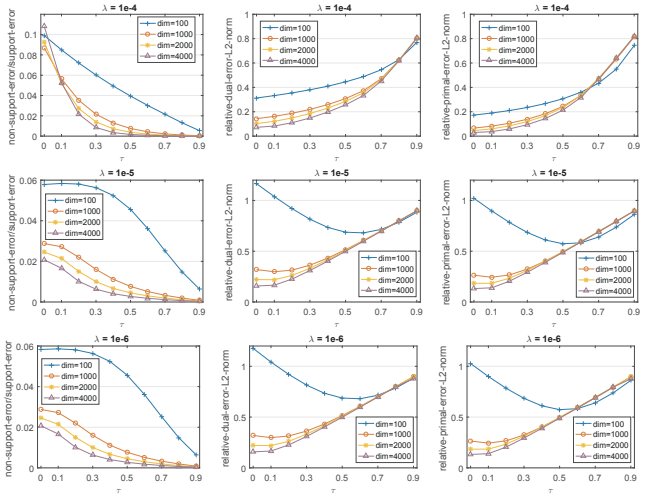
The learning time for the different methods is summarized in Table 3. DuRPDML and DuRPRSL learning time are recorded as two steps: subspace learning step and original space recovering step. It is not surprising to observe that the DuRPDML and DuRPRSL have similar subspace learning time to SRP and SPCA, which are significantly more efficient than other methods. Combining subspace learning time and original space recovering time, DuRPDML and DuRPRSL are 10 times faster than OASIS and 100 times faster for *RCV1* and *URL* datasets. RSL-based methods are always faster than DML-based methods due to avoiding PSD constraints. This PSD constraint makes the recovery time of the DuRPRSL method 2 times faster than DuRPDML.

Comparing the methods (DuRPDML, DuRPRSL) that apply random projection with those (OASIS, DuDML, DuRSL) that do not, we see that random projection methods significantly reduce the computational cost in Table 3. DuRPRSL can save at least 80% computational time than the original space optimization methods in these test datasets.

4.6 Dual-sparse Study

In this experiment, a case study in support of DuSRPRSL is presented. We used the RCV1 data in Table 1 to conduct a case study. We aimed to answer two questions related to our motivation: (i) How is the recovery error affected by the value of τ in Eq. (11)? (ii) How does the number of random projection dimension m affect the recovery error?

We varied the value of $\tau \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$, the value of $m \in \{100, 1000, 2000, 4000\}$, and the value of $\lambda \in \{1\text{e-}4, 1\text{e-}5, 1\text{e-}6\}$. $\tau = 0$ corresponds to the random reduction approach without the sparse regularizer. Three evaluation metrics were used: $e_1 = \frac{\|[\widetilde{\boldsymbol{\alpha}}_*]_{\mathcal{S}^c}\|_1}{\|[\widetilde{\boldsymbol{\alpha}}_*]_{\mathcal{S}} - [\boldsymbol{\alpha}_*]_{\mathcal{S}}\|_1}$, $e_2 = \frac{\|\widetilde{\boldsymbol{\alpha}}_* - \boldsymbol{\alpha}_*\|_2}{\|\boldsymbol{\alpha}_*\|_2}$, and $e_3 = \frac{\|\widetilde{M}_* - M_*\|_2}{\|M_*\|_2}$, where $\widetilde{\boldsymbol{\alpha}}_*$ is the optimal dual solution to (11) and $\boldsymbol{\alpha}_*$ is the optimal dual solution to (4) solving in the original space with the support set $\mathcal{S}$. The set $\mathcal{S}^c$ is the complement of $\mathcal{S}$. e_1 tells the error ratio of non-support sample and support sample. e_2 and e_3 tell the dual recovery error and primal recovery error.



(a) e_1 vs τ (b) e_2 vs τ (c) e_3 vs τ

Figure 3: Recovery error for non-smooth hinge loss.

The recovery performance is shown in Fig. 3. In the left column, the ration of e_1 decreases when τ increases. This indicates that the magnitude of dual variables for the original non-support samples decreases. The recovery errors of the dual solution and the primal solution are illustrated in the middle column and right column in Fig. 3. Generally, the larger τ will lead to a larger dual recovery error. For the case λ=1e-5 and λ=1e-6, the recovery error first decreases and then increases. This indicates that, when τ is less than a threshold, the dual recovery error will decrease as τ increases. This can help to relax the low-rank assumption issue. Comparing case λ=1e-4, λ=1e-5 and case λ=1e-6, the difference is that smaller λ will lead $\|M_*\|_2$ larger, which makes the support samples more sparse.

5 Conclusions

This paper presented a dual random projection method for similarity learning, which is suitable for large-scale high-dimensional datasets. The main idea is to first solve the dual problem in the subspace spanned by the random projection matrix, and then use these dual variables to recover the similarity function in the original space. This method can accurately and efficiently recover the original optimal solution with a small error. In addition, a dual-sparse regularized randomized reduction method is proposed to relax the low-rank assumption. The numerical experiments demonstrated the efficiency and effectiveness of our proposed reduction and recovery methods.

Acknowledgments

This work is conducted within the Rolls-Royce@NTU Corporate Lab with support from the National Research Foundation (NRF) Singapore under the Corp Lab@University Scheme.

References

[Balcan *et al.*, 2006] Maria-Florina Balcan, Avrim Blum, and Santosh Vempala. Kernels as features: On kernels, margins, and low-dimensional mappings. *Machine Learning*, 65(1):79–94, 2006.

[Bellet *et al.*, 2012] Aurélien Bellet, Amaury Habrard, and Marc Sebban. Similarity learning for provably accurate sparse linear classification. In *Proc. ICML*, pages 1491–1498, Edinburgh, Scotland, UK, June 2012.

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

[Bingham and Mannila, 2001] Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Proc. SIGKDD*, pages 245–250, San Francisco, CA, USA, August 2001.

[Chechik *et al.*, 2010] Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 11:1109–1135, 2010.

[Cheng, 2013] Li Cheng. Riemannian similarity learning. In *Proc. ICML*, pages 540–548, Atlanta, GA, USA, June 2013.

[Crammer and Chechik, 2012] Koby Crammer and Gal Chechik. Adaptive regularization for weight matrices. In *Proc. ICML*, pages 425–432, Edinburgh, Scotland, UK, June 2012.

[Davis and Dhillon, 2008] Jason V. Davis and Inderjit S. Dhillon. Structured metric learning for high dimensional problems. In *Proc. SIGKDD*, pages 195–203, Las Vegas, NV, USA, August 2008.

[Fern and Brodley, 2003] Xiaoli Z Fern and Carla E Brodley. Random projection for high dimensional data clustering: A cluster ensemble approach. In *Proc. ICML*, pages 186–193, Washington, DC, USA, August 2003.

[Fradkin and Madigan, 2003] Dmitriy Fradkin and David Madigan. Experiments with random projections for machine learning. In *Proc. SIGKDD*, pages 517–522, Washington, DC, USA, August 2003.

[Jin *et al.*, 2009] Rong Jin, Shijun Wang, and Yang Zhou. Regularized distance metric learning: Theory and algorithm. In *Proc. NIPS*, pages 862–870, Vancouver, British Columbia, Canada, December 2009.

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

[Maillard and Munos, 2012] Odalric-Ambrym Maillard and Rémi Munos. Linear regression with random projections. *Journal of Machine Learning Research*, 13:2735–2772, 2012.

[Paul *et al.*, 2013] Saurabh Paul, Christos Boutsidis, Malik Magdon-Ismail, and Petros Drineas. Random projections for support vector machines. In *Proc. AISTATS*, pages 498–506, Scottsdale, AZ, USA, April 2013.

[Qian *et al.*, 2015] Qi Qian, Rong Jin, Shenghuo Zhu, and Yuanqing Lin. Fine-grained visual categorization via multi-stage metric learning. In *Proc. CVPR*, pages 3716–3724, Boston, MA, USA, June 2015.

[Shalev-Shwartz and Zhang, 2013] Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14(1):567–599, 2013.

[Shalev-Shwartz *et al.*, 2004] Shai Shalev-Shwartz, Yoram Singer, and Andrew Y. Ng. Online and batch learning of pseudo-metrics. In *Proc. ICML*, page 94, Banff, Alberta, Canada, July 2004.

[Shi *et al.*, 2012] Qinfeng Shi, Chunhua Shen, Rhys Hill, and Anton van den Hengel. Is margin preserved after random projection? In *Proc. ICML*, pages 643–650, Edinburgh, Scotland, UK, June 2012.

[Venna *et al.*, 2010] Jarkko Venna, Jaakko Peltonen, Kristian Nybo, Helena Aidos, and Samuel Kaski. Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *Journal of Machine Learning Research*, 11:451–490, 2010.

[Weinberger and Saul, 2009] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, 2009.

[Xu *et al.*, 2017] Yi Xu, Haiqin Yang, Lijun Zhang, and Tianbao Yang. Efficient non-oblivious randomized reduction for risk minimization with improved excess risk guarantee. In *Proc. AAAI*, pages 2796–2802, San Francisco, CA, USA, February 2017.

[Yang *et al.*, 2015] Tianbao Yang, Lijun Zhang, Rong Jin, and Shenghuo Zhu. Theory of dual-sparse regularized randomized reduction. In *Proc. ICML*, pages 305–314, Lille, France, July 2015.

[Zhang *et al.*, 2013] Lijun Zhang, Mehrdad Mahdavi, Rong Jin, Tianbao Yang, and Shenghuo Zhu. Recovering the optimal solution by dual random projection. In *Proc. COLT*, pages 135–157, Princeton University, NJ, USA, June 2013.

[Zhang *et al.*, 2014] Lijun Zhang, Mehrdad Mahdavi, Rong Jin, Tianbao Yang, and Shenghuo Zhu. Random projections for classification: A recovery approach. *IEEE Trans. Information Theory*, 60(11):7300–7316, 2014.

[Zhang *et al.*, 2016] Weizhong Zhang, Lijun Zhang, Rong Jin, Deng Cai, and Xiaofei He. Accelerated sparse linear regression via random projection. In *Proc. AAAI*, pages 2337–2343, Phoenix, AZ, USA, February 2016.