Learning Better Data Representation using Inference-Driven Metric Learning

Paramveer S. Dhillon
CIS Deptt., Univ. of Penn.
Philadelphia, PA, U.S.A
dhillon@cis.upenn.edu

Partha Pratim Talukdar*
Search Labs, Microsoft Research
Mountain View, CA, USA
partha@talukdar.net

Koby Crammer
Deptt. of Electrical Engg.
The Technion, Haifa, Israel
koby@ee.technion.ac.il

Abstract

We initiate a study comparing effectiveness of the transformed spaces learned by recently proposed *supervised*, and *semi-supervised* metric learning algorithms to those generated by previously proposed *unsupervised* dimensionality reduction methods (e.g., PCA). Through a variety of experiments on different real-world datasets, we find IDML-IT, a *semi-supervised* metric learning algorithm to be the most effective.

1 Introduction

Because of the high-dimensional nature of NLP datasets, estimating a large number of parameters (a parameter for each dimension), often from a limited amount of labeled data, is a challenging task for statistical learners. Faced with this challenge, various *unsupervised* dimensionality reduction methods have been developed over the years, e.g., Principal Components Analysis (PCA).

Recently, several supervised metric learning algorithms have been proposed (Davis et al., 2007; Weinberger and Saul, 2009). IDML-IT (Dhillon et al., 2010) is another such method which exploits labeled as well as unlabeled data during metric learning. These methods learn a Mahalanobis distance metric to compute distance between a pair of data instances, which can also be interpreted as learning a transformation of the input data, as we shall see in Section 2.1.

In this paper, we make the following contributions:

Even though different supervised and semi-supervised metric learning algorithms have recently been proposed, effectiveness of the transformed spaces learned by them in NLP datasets has not been studied before. In this paper, we address that gap: we compare effectiveness of classifiers trained on the transformed spaces learned by metric learning methods to those generated by previously proposed *unsupervised* dimensionality reduction methods. We find IDML-IT, a *semi-supervised* metric learning algorithm to be the most effective.

2 Metric Learning

2.1 Relationship between Metric Learning and Linear Projection

We first establish the well-known equivalence between learning a Mahalanobis distance measure and Euclidean distance in a linearly transformed space of the data (Weinberger and Saul, 2009). Let A be a $d \times d$ positive definite matrix which parameterizes the Mahalanobis distance, $d_A(x_i, x_j)$, between instances x_i and x_j, as shown in Equation 1. Since A is positive definite, we can decompose it as $A = P^\top P$, where P is another matrix of size $d \times d$.

$$\begin{aligned} d_A(x_i, x_j) &= (x_i - x_j)^\top A (x_i - x_j) \\ &= (Px_i - Px_j)^\top (Px_i - Px_j) \\ &= d_{\text{Euclidean}}(Px_i, Px_j) \end{aligned} \quad (1)$$

Hence, computing Mahalanobis distance parameterized by A is equivalent to first projecting the instances into a new space using an appropriate transformation matrix P and then computing Euclidean distance in the linearly transformed space. In this paper, we are interested in learning a better representation of the data (i.e., projection matrix P), and we shall achieve that goal by learning the corresponding Mahalanobis distance parameter A.

We shall now review two recently proposed metric learning algorithms.

* Research carried out while at the University of Pennsylvania, Philadelphia, PA, USA.

Proceedings of the ACL 2010 Conference Short Papers, pages 377–381,
Uppsala, Sweden, 11-16 July 2010.

2.2 Information-Theoretic Metric Learning (ITML): Supervised

Information-Theoretic Metric Learning (ITML) (Davis et al., 2007) assumes the availability of prior knowledge about inter-instance distances. In this scheme, two instances are considered similar if the Mahalanobis distance between them is upper bounded, i.e., $d_A(x_i, x_j) \leq u$, where u is a non-trivial upper bound. Similarly, two instances are considered dissimilar if the distance between them is larger than certain threshold l, i.e., $d_A(x_i, x_j) \geq l$. Similar instances are represented by set S, while dissimilar instances are represented by set D.

In addition to prior knowledge about inter-instance distances, sometimes prior information about the matrix A, denoted by A_0, itself may also be available. For example, Euclidean distance (i.e., $A_0 = I$) may work well in some domains. In such cases, we would like the learned matrix A to be as close as possible to the prior matrix A_0. ITML combines these two types of prior information, i.e., knowledge about inter-instance distances, and prior matrix A_0, in order to learn the matrix A by solving the optimization problem shown in (2).

$$\begin{aligned} \min_{A \succeq 0} \quad & D_{\text{ld}}(A, A_0) \qquad (2) \\ \text{s.t.} \quad & \text{tr}\{A(x_i - x_j)(x_i - x_j)^\top\} \leq u, \\ & \qquad \forall (i,j) \in S \\ & \text{tr}\{A(x_i - x_j)(x_i - x_j)^\top\} \geq l, \\ & \qquad \forall (i,j) \in D \end{aligned}$$

where $D_{\text{ld}}(A, A_0) = \text{tr}(AA_0^{-1}) - \log\det(AA_0^{-1}) - n$, is the LogDet divergence.

To handle situations where exactly solving the problem in (2) is not possible, slack variables may be introduced to the ITML objective. To solve this optimization problem, an algorithm involving repeated Bregman projections is presented in (Davis et al., 2007), which we use for the experiments reported in this paper.

2.3 Inference-Driven Metric Learning (IDML): Semi-Supervised

Notations: We first define the necessary notations. Let X be the $d \times n$ matrix of n instances in a d-dimensional space. Out of the n instances, n_l instances are labeled, while the remaining n_u instances are unlabeled, with $n = n_l + n_u$. Let S be a $n \times n$ diagonal matrix with $S_{ii} = 1$ iff instance x_i is labeled. m is the total number of labels. Y is the $n \times m$ matrix storing training label information, if any. $\hat{Y}$ is the $n \times m$ matrix of estimated label information, i.e., output of any classifier, with $\hat{Y}_{il}$ denoting score of label l at node i. .

The ITML metric learning algorithm, which we reviewed in Section 2.2, is supervised in nature, and hence it does not exploit widely available unlabeled data. In this section, we review Inference Driven Metric Learning (IDML) (Algorithm 1) (Dhillon et al., 2010), a recently proposed metric learning framework which combines an existing *supervised* metric learning algorithm (such as ITML) along with *transductive* graph-based label inference to learn a new distance metric from labeled as well as unlabeled data combined. In self-training styled iterations, IDML alternates between metric learning and label inference; with output of label inference used during next round of metric learning, and so on.

IDML starts out with the assumption that existing supervised metric learning algorithms, such as ITML, can learn a better metric if the number of available labeled instances is increased. Since we are focusing on the semi-supervised learning (SSL) setting with n_l labeled and n_u unlabeled instances, the idea is to automatically label the unlabeled instances using a graph based SSL algorithm, and then include instances with low assigned label entropy (i.e., high confidence label assignments) in the next round of metric learning. The number of instances added in each iteration depends on the threshold β[1]. This process is continued until no new instances can be added to the set of labeled instances, which can happen when either all the instances are already exhausted, or when none of the remaining unlabeled instances can be assigned labels with high confidence.

The IDML framework is presented in Algorithm 1. In Line 3, any supervised metric learner, such as ITML, may be used as the METRICLEARNER. Using the distance metric learned in Line 3, a new k-NN graph is constructed in Line 4 , whose edge weight matrix is stored in W. In Line 5 , GRAPHLABELINF optimizes over the newly constructed graph, the GRF objective (Zhu et al., 2003) shown in (3).

$$\min_{\hat{Y}'} \ \text{tr}\{\hat{Y}'^\top L \hat{Y}'\}, \ \text{s.t.} \ \hat{S}\hat{Y} = \hat{S}\hat{Y}' \qquad (3)$$

where $L = D - W$ is the (unnormalized) Lapla-

[1]During the experiments in Section 3, we set $\beta = 0.05$

Algorithm 1: Inference Driven Metric Learning (IDML)
Input: instances X, training labels Y, training instance indicator S, label entropy threshold β, neighborhood size k
Output: Mahalanobis distance parameter A

1: $\hat{Y} \leftarrow Y$, $\hat{S} \leftarrow S$
2: **repeat**
3: $A \leftarrow \text{METRICLEARNER}(X, \hat{S}, \hat{Y})$
4: $W \leftarrow \text{CONSTRUCTKNNGRAPH}(X, A, k)$
5: $\hat{Y}' \leftarrow \text{GRAPHLABELINF}(W, \hat{S}, \hat{Y})$
6: $U \leftarrow \text{SELECTLOWENTINST}(\hat{Y}', \hat{S}, \beta)$
7: $\hat{Y} \leftarrow \hat{Y} + U\hat{Y}'$
8: $\hat{S} \leftarrow \hat{S} + U$
9: **until** convergence (i.e., $U_{ii} = 0,\ \forall i$)
10: return A

cian, and D is a diagonal matrix with $D_{ii} = \sum_j W_{ij}$. The constraint, $\hat{S}\hat{Y} = \hat{S}\hat{Y}'$, in (3) makes sure that labels on training instances are not changed during inference. In Line 6, a currently unlabeled instance x_i (i.e., $\hat{S}_{ii} = 0$) is considered a new labeled training instance, i.e., $U_{ii} = 1$, for next round of metric learning if the instance has been assigned labels with high confidence in the current iteration, i.e., if its label distribution has low entropy (i.e., $\text{ENTROPY}(\hat{Y}'_{i:}) \leq \beta$). Finally in Line 7, training instance label information is updated. This iterative process is continued till no new labeled instance can be added, i.e., when $U_{ii} = 0\ \forall i$. IDML returns the learned matrix A which can be used to compute Mahalanobis distance using Equation 1.

3 Experiments

3.1 Setup

Dataset	Dimension	Balanced
Electronics	84816	Yes
Books	139535	Yes
Kitchen	73539	Yes
DVDs	155465	Yes
WebKB	44261	Yes

Table 1: Description of the datasets used in Section 3. All datasets are binary with 1500 total instances in each.

Description of the datasets used during experiments in Section 3 are presented in Table 1. The first four datasets – Electronics, Books, Kitchen, and DVDs – are from the sentiment domain and previously used in (Blitzer et al., 2007). WebKB is a text classification dataset derived from (Subramanya and Bilmes, 2008). For details regarding features and data pre-processing, we refer the reader to the origin of these datasets cited above. One extra preprocessing that we did was that we only considered features which occurred more 20 times in the entire dataset to make the problem more computationally tractable and also since the infrequently occurring features usually contribute noise. We use classification error (lower is better) as the evaluation metric. We experiment with the following ways of estimating transformation matrix P:

Original[2]: We set $P = I$, where I is the $d \times d$ identity matrix. Hence, the data is not transformed in this case.

RP: The data is first projected into a lower dimensional space using the Random Projection (RP) method (Bingham and Mannila, 2001). Dimensionality of the target space was set at $d' = \frac{\log n}{\epsilon^2 \log \frac{1}{\epsilon}}$, as prescribed in (Bingham and Mannila, 2001). We use the projection matrix constructed by RP as P. ϵ was set to 0.25 for the experiments in Section 3, which has the effect of projecting the data into a much lower dimensional space (84 for the experiments in this section). This presents an interesting evaluation setting as we already run evaluations in much higher dimensional space (e.g., Original).

PCA: Data instances are first projected into a lower dimensional space using Principal Components Analysis (PCA) (Jolliffe, 2002) . Following (Weinberger and Saul, 2009), dimensionality of the projected space was set at 250 for all experiments. In this case, we used the projection matrix generated by PCA as P.

ITML: A is learned by applying ITML (see Section 2.2) on the Original space (above), and then we decompose A as $A = P^\top P$ to obtain P.

[2]Note that "Original" in the results tables refers to original space with features occurring more than 20 times. We also ran experiments with original set of features (without any thresholding) and the results were worse or comparable to the ones reported in the tables.

Datasets	Original $\mu \pm \sigma$	RP $\mu \pm \sigma$	PCA $\mu \pm \sigma$	ITML $\mu \pm \sigma$	IDML-IT $\mu \pm \sigma$
Electronics	31.3 ± 0.9	42.5 ± 1.0	46.4 ± 2.0	33.0 ± 1.0	**30.7±0.7**
Books	37.5 ± 1.1	45.0 ± 1.1	34.8 ± 1.4	35.0 ± 1.1	**32.0±0.9**
Kitchen	33.7 ± 1.0	43.0 ± 1.1	34.0 ± 1.6	30.9 ± 0.7	**29.0±1.0**
DVDs	39.0 ± 1.2	47.7 ± 1.2	36.2 ± 1.6	37.0 ± 0.8	**33.9±1.0**
WebKB	31.4 ± 0.9	33.0 ± 1.0	27.9 ± 1.3	28.9 ± 1.0	**25.5±1.0**

Table 2: Comparison of SVM % classification errors (lower is better), with 50 labeled instances (Sec. 3.2). n_l=50. and $n_u = 1450$. All results are averaged over ten trials. All hyperparameters are tuned on a separate random split.

Datasets	Original $\mu \pm \sigma$	RP $\mu \pm \sigma$	PCA $\mu \pm \sigma$	ITML $\mu \pm \sigma$	IDML-IT $\mu \pm \sigma$
Electronics	27.0 ± 0.9	40.0 ± 1.0	41.2 ± 1.0	27.5 ± 0.8	**25.3±0.8**
Books	31.0 ± 0.7	42.9 ± 0.6	31.3 ± 0.7	29.9 ± 0.5	**27.7±0.7**
Kitchen	26.3 ± 0.5	41.9 ± 0.7	27.0 ± 0.9	26.1 ± 0.8	**24.8±0.9**
DVDs	34.7 ± 0.4	46.8 ± 0.6	32.9 ± 0.8	34.0 ± 0.8	**31.8±0.9**
WebKB	25.7 ± 0.5	31.1 ± 0.5	24.9 ± 0.6	25.6 ± 0.4	**23.9±0.4**

Table 3: Comparison of SVM % classification errors (lower is better), with 100 labeled instances (Sec. 3.2). n_l=100. and $n_u = 1400$. All results are averaged over ten trials. All hyperparameters are tuned on a separate random split.

IDML-IT: A is learned by applying IDML (Algorithm 1) (see Section 2.3) on the Original space (above); with ITML used as METRICLEARNER in IDML (Line 3 in Algorithm 1). In this case, we treat the set of test instances (without their gold labels) as the unlabeled data. In other words, we essentially work in the transductive setting (Vapnik, 2000). Once again, we decompose A as $A = P^\top P$ to obtain P.

We also experimented with the supervised large-margin metric learning algorithm (LMNN) presented in (Weinberger and Saul, 2009). We found ITML to be more effective in practice than LMNN, and hence we report results based on ITML only. Each input instance, x, is now projected into the transformed space as Px. We now train different classifiers on this transformed space. All results are averaged over ten random trials.

3.2 Supervised Classification

We train a SVM classifier, with an RBF kernel, on the transformed space generated by the projection matrix P. SVM hyperparameter, C and RBF kernel bandwidth, were tuned on a separate development split. Experimental results with 50 and 100 labeled instances are shown in Table 2, and Table 3, respectively. From these results, we observe that IDML-IT consistently achieves the best performance across all experimental settings. We also note that in Table 3, performance difference between ITML and IDML-IT in the Electronics and Kitchen domains are statistically significant.

3.3 Semi-Supervised Classification

In this section, we trained the GRF classifier (see Equation 3), a graph-based semi-supervised learning (SSL) algorithm (Zhu et al., 2003), using Gaussian kernel parameterized by $A = P^\top P$ to set edge weights. During graph construction, each node was connected to its k nearest neighbors, with k treated as a hyperparameter and tuned on a separate development set. Experimental results with 50 and 100 labeled instances are shown in Table 4, and Table 5, respectively. As before, we experimented with $n_l = 50$ and $n_l = 100$. Once again, we observe that IDML-IT is the most effective method, with the GRF classifier trained on the data representation learned by IDML-IT achieving best performance in all settings. Here also, we observe that IDML-IT achieves the best performance across all experimental settings.

Datasets	Original $\mu \pm \sigma$	RP $\mu \pm \sigma$	PCA $\mu \pm \sigma$	ITML $\mu \pm \sigma$	IDML-IT $\mu \pm \sigma$
Electronics	47.9 ± 1.1	49.0 ± 1.2	43.2 ± 0.9	34.9 ± 0.5	**34.0±0.5**
Books	50.0 ± 1.0	49.4 ± 1.0	47.9 ± 0.7	42.1 ± 0.7	**40.6±0.7**
Kitchen	49.8 ± 1.1	49.6 ± 0.9	48.6 ± 0.8	31.1 ± 0.5	**30.0±0.5**
DVDs	50.1 ± 0.5	49.9 ± 0.7	49.4 ± 0.6	42.1 ± 0.4	**41.2±0.5**
WebKB	33.1 ± 0.4	33.1 ± 0.3	33.1 ± 0.3	30.0 ± 0.4	**28.7±0.5**

Table 4: Comparison of transductive % classification errors (lower is better) over graphs constructed using different methods (see Section 3.3), with $n_l = 50$ and $n_u = 1450$. All results are averaged over ten trials. All hyperparameters are tuned on a separate random split.

Datasets	Original $\mu \pm \sigma$	RP $\mu \pm \sigma$	PCA $\mu \pm \sigma$	ITML $\mu \pm \sigma$	IDML-IT $\mu \pm \sigma$
Electronics	43.5 ± 0.7	47.2 ± 0.8	39.1 ± 0.7	31.3 ± 0.2	**30.8±0.3**
Books	48.3 ± 0.5	48.9 ± 0.3	43.3 ± 0.4	35.2 ± 0.5	**33.3±0.6**
Kitchen	45.3 ± 0.6	48.2 ± 0.5	41.0 ± 0.7	30.7 ± 0.6	**29.9±0.3**
DVDs	48.6 ± 0.3	49.3 ± 0.5	45.9 ± 0.5	42.6 ± 0.4	**41.7±0.3**
WebKB	33.4 ± 0.4	33.4 ± 0.4	33.4 ± 0.3	30.4 ± 0.5	**28.6±0.7**

Table 5: Comparison of transductive % classification errors (lower is better) over graphs constructed using different methods (see Section 3.3), with $n_l = 100$ and $n_u = 1400$. All results are averaged over ten trials. All hyperparameters are tuned on a separate random split.

4 Conclusion

In this paper, we compared the effectiveness of the transformed spaces learned by recently proposed *supervised*, and *semi-supervised* metric learning algorithms to those generated by previously proposed *unsupervised* dimensionality reduction methods (e.g., PCA). To the best of our knowledge, this is the first study of its kind involving NLP datasets. Through a variety of experiments on different real-world NLP datasets, we demonstrated that supervised as well as semi-supervised classifiers trained on the space learned by IDML-IT consistently result in the lowest classification errors. Encouraged by these early results, we plan to explore further the applicability of IDML-IT in other NLP tasks (e.g., entity classification, word sense disambiguation, polarity lexicon induction, etc.) where better representation of the data is a pre-requisite for effective learning.

Acknowledgments

Thanks to Kuzman Ganchev for providing detailed feedback on a draft of this paper. This work was supported in part by NSF IIS-0447972 and DARPA HRO1107-1-0029.

References

E. Bingham and H. Mannila. 2001. Random projection in dimensionality reduction: applications to image and text data. In *ACM SIGKDD*.

J. Blitzer, M. Dredze, and F. Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*.

J.V. Davis, B. Kulis, P. Jain, S. Sra, and I.S. Dhillon. 2007. Information-theoretic metric learning. In *ICML*.

P. S. Dhillon, P. P. Talukdar, and K. Crammer. 2010. Inference-driven metric learning for graph construction. Technical Report MS-CIS-10-18, CIS Department, University of Pennsylvania, May.

IT Jolliffe. 2002. *Principal component analysis*. Springer verlag.

A. Subramanya and J. Bilmes. 2008. Soft-Supervised Learning for Text Classification. In *EMNLP*.

V.N. Vapnik. 2000. *The nature of statistical learning theory*. Springer Verlag.

K.Q. Weinberger and L.K. Saul. 2009. Distance metric learning for large margin nearest neighbor classification. *The Journal of Machine Learning Research*.

X. Zhu, Z. Ghahramani, and J. Lafferty. 2003. Semi-supervised learning using Gaussian fields and harmonic functions. In *ICML*.