View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Ghent University Academic Bibliography

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1.
Available from http://www.i6doc.com/en/.

Distance metric learning: A two-phase approach

Bac Nguyen[1], Carlos Morell[2], and Bernard De Baets[1]

1- Ghent University - Department of Mathematical Modelling, Statistics and Bioinformatics, Coupure links 653, 9000 Ghent - Belgium

2- Universidad Central de Las Villas - Department of Computer Science Santa Clara, CP 54830 Villa Clara - Cuba

Abstract. Distance metric learning has been successfully incorporated in many machine learning applications. The main challenge arises from the positive semidefiniteness constraint on the Mahalanobis matrix, which results in a high computational cost. In this paper, we develop a novel approach to reduce this computational burden. We first map each training example into a new space by an orthonormal transformation. Then, in the transformed space, we simply learn a diagonal matrix. This two-phase approach is thus much easier and less costly than learning a full Mahalanobis matrix in one phase as is commonly done.

1 Introduction

Learning a good distance metric has become an established topic in machine learning during the past decade. The performance of many fundamental metric-based algorithms, such as k-nearest-neighbor (k-NN) classification and k-mean clustering, can be significantly improved when using an appropriate distance metric to measure the closeness between examples [1, 2]. For this reason, a number of distance metric learning approaches have been proposed (see [3] for a recent survey). Essentially, distance metric learning consists in adjusting a distance metric using the information contained in the input data. The resulting distance metric should satisfy some constraints of the application in question. These constraints are often specified in the form of pairwise constraints $(\mathbf{x}_i, \mathbf{x}_j)$, which means that $\mathbf{x}_i$ and $\mathbf{x}_j$ should be similar (i.e., must-link constraints) or dissimilar (i.e., cannot-link constraints), or in the form of triplet constraints $(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_l)$, which means that $\mathbf{x}_i$ should be more similar to $\mathbf{x}_j$ than to $\mathbf{x}_l$.

We focus on learning a Mahalanobis distance metric, where the squared distance between two examples $\mathbf{x}_i$ and $\mathbf{x}_j$ in $\mathbb{R}^D$ is computed as $d^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{M}(\mathbf{x}_i - \mathbf{x}_j)$ where $\mathbf{M} \succcurlyeq 0$ is a positive semidefinite (PSD) matrix. By factorizing $\mathbf{M} = \mathbf{L}\mathbf{L}^\top$, the Mahalanobis distance can be viewed as the Euclidean distance after applying a linear transformation $\mathbf{x}' = \mathbf{L}^\top \mathbf{x}$, i.e. $d^2_{\mathbf{L}}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{L}\mathbf{L}^\top (\mathbf{x}_i - \mathbf{x}_j) = \|\mathbf{L}^\top(\mathbf{x}_i - \mathbf{x}_j)\|^2$. This implies that learning a Mahalanobis distance metric corresponds to learning a linear transformation. Methods that learn a linear transformation, including Linear Discriminant Analysis (LDA) [4], Neighborhood Component Analysis (NCA) [5] and Distance Metric Learning through Maximization of the Jeffrey divergence (DMLMJ) [6], are mostly formulated as nonconvex optimization problems, which can be solved by gradient descent or eigenvalue optimization techniques. Taking the positive semidefiniteness constraint into account, methods that learn

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1.
Available from http://www.i6doc.com/en/.

the Mahalanobis matrix, including Large Margin Nearest neighbor (LMNN) [1], Information-Theoretic Metric Learning (ITML) [2], and Maximally Collapsing Metric Learning (MCML) [7], are mostly formulated as convex semidefinite programs, which can be solved by standard semidefinite programming, boosting, or Frank-Wolfe algorithms.

Learning a Mahalanobis distance metric becomes a very challenging problem for machines, especially in high-dimensional settings. This limitation arises from the positive semidefiniteness constraint on the Mahalanobis matrix, which requires for most approaches a computational complexity of $O(D^3)$ to make a projection onto the PSD cone. To reduce this computational burden, we propose a novel distance metric learning approach consisting of two phases: (1) we first find an orthonormal transformation to diagonalize the Mahalanobis matrix in the new coordinate system, (2) based on the preceding results, learning a full Mahalanobis matrix turns into learning a nonnegative diagonal matrix. Next, we introduce in more detail the problem formulation and its optimization algorithm.

2 Proposed approach

In this section, we will show how to learn a distance metric for k-NN classification. Our approach aims at finding a distance metric such that for each training example, its nearest neighbors of the same class are pulled as close as possible, while its nearest neighbors of different classes are pushed away as far as possible. Given a set of n labeled training examples $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, the learned distance metric should guarantee that the distance of $\mathbf{x}_i$ to any example of its *positive neighborhood* $\mathcal{N}^+(\mathbf{x}_i)$, which is a set of its nearest neighbors of the same class, should be smaller than the distance to any example of its *negative neighborhood* $\mathcal{N}^-(\mathbf{x}_i)$, which is a set of its nearest neighbors of different classes. The above statement is translated into the following set of triplet constraints

$$\mathcal{T} = \left\{ (\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_l) \mid \mathbf{x}_j \in \mathcal{N}^+(\mathbf{x}_i) \text{ and } \mathbf{x}_l \in \mathcal{N}^-(\mathbf{x}_i) \right\}.$$

By keeping a safety margin between the positive and negative neighborhoods of each training example, we ensure that the performance of k-NN will be improved.

We start by introducing some notations that are necessary for developing our proposal. Let m denote the rank of $\mathbf{M}$. Since $\mathbf{M} \succcurlyeq 0$, it can be represented by a nonnegative weighted sum of m rank-one matrices[1] as $\mathbf{M} = \sum_{i=1}^m w_i \mathbf{a}_i \mathbf{a}_i^\top = \mathbf{A}\mathbf{W}\mathbf{A}^\top$, where $\mathbf{A}$ is a real matrix, whose columns are the m orthonormal vectors $\mathbf{a}_i$, and $\mathbf{W}$ is a diagonal matrix, whose diagonal elements are the m nonnegative values w_i. In other words, $\mathbf{A}$ can be seen as an orthonormal transformation that performs a rotation and a reduction of dimensionality of the input space, while $\mathbf{W}$ can be seen as a matrix of scaling factors, which are given by $\sqrt{w_i}$, along the direction of each axis in the transformed space induced by $\mathbf{A}$. In this paper, we cast the problem of learning a Mahalanobis distance metric as learning an orthonormal transformation $\mathbf{A}$ and a diagonal matrix $\mathbf{W}$. If we can find an appropriate orthonormal transformation that eliminates the correlation

[1]This factorization can be performed using, for instance, eigen-decomposition.

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1.
Available from http://www.i6doc.com/en/.

between features, then learning a full Mahalanobis matrix becomes learning a simple diagonal matrix. In short, our proposed method consists of two phases that are described next.

In the first phase, we learn an orthonormal transformation $\mathbf{A}$ that satisfies as many triplet constraints $(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_l) \in \mathcal{T}$ as possible. For this purpose, we define the following loss function that penalizes the large distances between examples $(\mathbf{x}_i, \mathbf{x}_j)$ of the same class and the small distances between examples $(\mathbf{x}_i, \mathbf{x}_l)$ of different classes,

$$\begin{aligned} f(\mathbf{A}) &= \sum_{(\mathbf{x}_i,\mathbf{x}_j,\mathbf{x}_l)\in\mathcal{T}} d^2_{\mathbf{A}}(\mathbf{x}_i,\mathbf{x}_j) - d^2_{\mathbf{A}}(\mathbf{x}_i,\mathbf{x}_l) \\ &= \operatorname{tr}\Big(\sum_{(\mathbf{x}_i,\mathbf{x}_j,\mathbf{x}_l)\in\mathcal{T}} \big(\mathbf{A}^\top(\mathbf{x}_i-\mathbf{x}_j)\big)^\top\big(\mathbf{A}^\top(\mathbf{x}_i-\mathbf{x}_j)\big) - \big(\mathbf{A}^\top(\mathbf{x}_i-\mathbf{x}_l)\big)^\top\big(\mathbf{A}^\top(\mathbf{x}_i-\mathbf{x}_l)\big)\Big) \\ &= \operatorname{tr}\Big(\sum_{(\mathbf{x}_i,\mathbf{x}_j,\mathbf{x}_l)\in\mathcal{T}} \big(\mathbf{A}^\top(\mathbf{x}_i-\mathbf{x}_j)\big)\big(\mathbf{A}^\top(\mathbf{x}_i-\mathbf{x}_j)\big)^\top - \big(\mathbf{A}^\top(\mathbf{x}_i-\mathbf{x}_l)\big)\big(\mathbf{A}^\top(\mathbf{x}_i-\mathbf{x}_l)\big)^\top\Big) \\ &= \operatorname{tr}\Big(\mathbf{A}^\top \sum_{(\mathbf{x}_i,\mathbf{x}_j,\mathbf{x}_l)\in\mathcal{T}} \Big((\mathbf{x}_i-\mathbf{x}_j)(\mathbf{x}_i-\mathbf{x}_j)^\top - (\mathbf{x}_i-\mathbf{x}_l)(\mathbf{x}_i-\mathbf{x}_l)^\top\Big)\mathbf{A}\Big)\,. \end{aligned}$$

Hence, learning an orthonormal transformation $\mathbf{A}$ amounts to solving the following optimization problem

$$\max_{\mathbf{A}^\top\mathbf{A}=\mathbf{I}} \quad \operatorname{tr}\left(\mathbf{A}^\top\boldsymbol{\Sigma}\mathbf{A}\right)\,,$$

where $\boldsymbol{\Sigma} = \sum_{(\mathbf{x}_i,\mathbf{x}_j,\mathbf{x}_l)\in\mathcal{T}} (\mathbf{x}_i-\mathbf{x}_l)(\mathbf{x}_i-\mathbf{x}_l)^\top - (\mathbf{x}_i-\mathbf{x}_j)(\mathbf{x}_i-\mathbf{x}_j)^\top$. Following [8], the aforementioned problem can be solved using standard eigen-decomposition. That is, the optimal solution is the matrix whose columns are the m linearly independent eigenvectors corresponding to the m largest positive eigenvalues of $\boldsymbol{\Sigma}$. The parameter m corresponds to the number of features in the transformed space induced by $\mathbf{A}$.

In the second phase, we learn a diagonal matrix $\mathbf{W}$ in the transformed space induced by $\mathbf{A}$. After applying the orthonormal transformation, the training examples become $\{(\mathbf{A}^\top\mathbf{x}_i, y_i)\}_{i=1}^n$. In order to satisfy as many triplet constraints $(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_l) \in \mathcal{T}$ as possible, we formulate the problem of learning $\mathbf{W}$ as follows

$$\min_{\mathbf{W}\succcurlyeq 0} \quad \frac{1}{2}\|\mathbf{W}\|_F^2 + C\sum_{(\mathbf{x}_i,\mathbf{x}_j,\mathbf{x}_l)\in\mathcal{T}} \Big[1 + d^2_{\mathbf{W}}(\mathbf{A}^\top\mathbf{x}_i, \mathbf{A}^\top\mathbf{x}_j) - d^2_{\mathbf{W}}(\mathbf{A}^\top\mathbf{x}_i, \mathbf{A}^\top\mathbf{x}_l)\Big]_+\,, \quad (1)$$

where $C > 0$ is the regularization hyper-parameter, and $[.]_+$ is the function that returns the positive part of its argument. The first term in (1) is the squared Frobenius norm regularization of $\mathbf{W}$ and the second term in (1) is the hinge loss function with margin one that penalizes the violations of triplet constraints in $\mathcal{T}$. Since $\mathbf{W}$ is diagonal, the squared Mahalanobis distance metric can be rewritten as

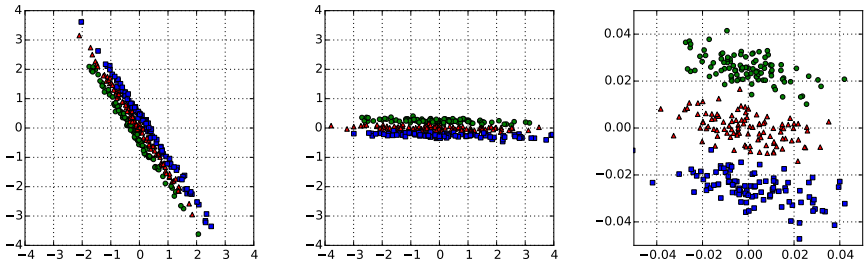
$$d^2_{\mathbf{W}}(\mathbf{A}^\top\mathbf{x}_i, \mathbf{A}^\top\mathbf{x}_j) = \mathbf{w}^\top\left[(\mathbf{A}^\top\mathbf{x}_i - \mathbf{A}^\top\mathbf{x}_j) \circ (\mathbf{A}^\top\mathbf{x}_i - \mathbf{A}^\top\mathbf{x}_j)\right] = \langle \mathbf{w}, \mathbf{a}_{ij}\rangle\,,$$

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1.
Available from http://www.i6doc.com/en/.

where $\mathbf{w}$ is the vector containing all diagonal elements of $\mathbf{W}$, the operator $\circ$ denotes the element-wise product, and $\mathbf{a}_{ij} = (\mathbf{A}^\top \mathbf{x}_i - \mathbf{A}^\top \mathbf{x}_j) \circ (\mathbf{A}^\top \mathbf{x}_i - \mathbf{A}^\top \mathbf{x}_j)$. Let $\xi_{ijl} \geq 0$ be slack variables corresponding to each triplet constraint $(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_l) \in \mathcal{T}$, then problem (1) can be rewritten as

$$\begin{aligned} \min \quad & \frac{1}{2}\mathbf{w}^\top \mathbf{w} + C \sum_{(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_l) \in \mathcal{T}} \xi_{ijl} \\ \text{s.t.} \quad & \forall (\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_l) \in \mathcal{T}: \quad \langle \mathbf{w}, \mathbf{a}_{il} - \mathbf{a}_{ij} \rangle \geq 1 - \xi_{ijl}, \quad \xi_{ijl} \geq 0\,, \\ & \forall i \in \{1, \ldots, m\}: \quad w_i \geq 0\,. \end{aligned} \tag{2}$$

Problem (2) is a convex quadratic program. Hence, it can be solved by a standard quadratic programming solver. However, general-purpose solvers tend to scale poorly in a large number of triplet constraints. Since our optimization problem is very close to that introduced in [9], we employ the Lagrangian dual method via coordinate descent as proposed by Nguyen et al. [9] to solve (2). The idea is to adopt a coordinate descent method to solve the dual problem. In each iteration, it requires to solve only one-variable subproblem while keeping track of the primal variables during the optimization procedure.



(a) (b) (c)

Fig. 1: A synthetic data set illustrating the idea behind TPDML. Examples of the same class are shown in the same color and style: (a) Data set in the original space, (b) data set in the rotated space after the first phase, and (c) data set in the transformed space after the second phase.

We refer to the proposed method as *two-phase distance metric learning* (TPDML). Figure 1 illustrates the main idea behind TPDML. In the original space, it is clear that the classification accuracy of k-NN is very poor since examples of different classes are very close (see Figure 1(a)). After learning the orthonormal transformation $\mathbf{A}$, all examples are rotated along two axes (see Figure 1(b)). Finally, they are properly separated by "shrinking" along the horizontal axis and "stretching" along the vertical axis (see Figure 1(c)) after learning the weights $\mathbf{w}$.

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1. Available from http://www.i6doc.com/en/.

3 Experiments

In this section, we compare the performance of **TPDML** with that of state-of-the-art distance metric learning methods, including **ITML** [2], **LMNN** [1], **DML-eig** [10], **SCML** [11] and the **Euclidean** distance metric. Thus, to make the comparison of these methods as fair as possible, all experiments are carried out in the context of 5-NN. The hyper-parameters of the competing methods are tuned using cross-validation to get the best results. For TPDML, we tune the hyper-parameter C considering as set of values $\{0.001, 0.01, ..., 1000\}$. The source codes implemented in Matlab of all methods have been supplied by the respective authors.

We evaluate the performance of the competing methods on twelve standard benchmark data sets with different sizes. Except *isolet*[2] and *usps*[3], all data sets are downloaded from the KEEL repository[4]. Table 1 presents a summary description of these data sets. All features are normalized into the interval $[0, 1]$. In the experiments, we use 10-fold cross-validation to estimate the test accuracy of k-NN classification.

Data set	#features	#examples	Data set	#features	#examples
appendicitis	7	106	monk-2	6	432
balance	4	625	movement	90	360
banana	2	5300	optdigits	64	5620
isolet	617	7797	sonar	60	208
letter	16	20000	usps	256	9298
magic	10	19020	wine	13	178

Table 1: Description of data sets used in our experiments.

Table 2 shows the classification accuracy of the competing methods on the selected data sets. The last two rows of this table are the average ranks and the training time (in seconds) of each method. On each data set, we assign rank 1 to the method with the highest accuracy, rank 2 to the method with the second highest accuracy, and so on. From the results, we observe that all distance metric learning methods improve the performance of the standard k-NN classification using the Euclidean distance metric. Moreover, our method performs competitively with other competing methods with regard to classification accuracy, while it is an order of magnitude faster in training time.

4 Conclusion

In this paper, we have proposed a novel distance metric learning approach (TPDML) consisting of learning an orthonormal transformation and a diagonal matrix. The learned distance metric aims at reducing the number of local triplet constraints in order to improve the performance of k-NN classification.

[2]https://archive.ics.uci.edu/ml/datasets/ISOLET
[3]http://www-i6.informatik.rwth-aachen.de/ keysers/usps.html
[4]http://sci2s.ugr.es/keel/datasets.php

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1. Available from http://www.i6doc.com/en/.

Data set	Euclidean	ITML	LMNN	DML-eig	SCML	TPDML
appendicitis	85.00	86.00	**88.82**	87.00	86.91	**88.82**
balance	86.24	91.84	84.64	87.52	94.25	**95.35**
banana	89.28	89.34	89.34	89.17	89.36	**89.49**
isolet	91.28	93.07	95.89	89.20	91.70	94.23
letter	95.55	95.37	96.72	84.42	96.54	**96.91**
magic	83.60	83.73	83.74	83.15	84.79	**83.83**
monk-2	94.75	89.43	97.04	**100.00**	99.55	98.86
movement	75.28	74.72	**82.50**	67.22	63.33	79.17
optdigits	98.75	98.70	**99.04**	97.44	97.21	98.83
sonar	84.52	81.69	84.05	**85.05**	80.19	83.62
usps	94.47	94.37	94.57	88.00	85.70	**94.67**
wine	95.49	96.67	97.78	96.63	**98.86**	96.60
Rank	4.33	4.13	2.50	4.33	3.58	**2.13**
Time		7866.82	3398.98	8396.48	2259.24	**567.52**

Table 2: Classification accuracy of the competing methods in our experiments.

Experimental results on real data sets confirmed the efficiency and efficacy of our method. Future work will concentrate on extending this approach into a kernelized version which can be more applicable to non-linear classification problems.

References

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

[2] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *Proceedings of the Twenty-Fourth International Conference on Machine Learning*, pages 209–216, 2007.

[3] A. Bellet, A. Habrard, and M. Sebban. *Metric Learning*. Morgan & Claypool Publishers, 2015.

[4] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

[5] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. In *Advances in Neural Information Processing Systems 17*, pages 513–520, 2005.

[6] B. Nguyen, C. Morell, and B. De Baets. Supervised distance metric learning through maximization of the Jeffrey divergence. *Pattern Recognition*, 64:215–225, 2017.

[7] A. Globerson and S. T. Roweis. Metric learning by collapsing classes. In *Advances in Neural Information Processing Systems 18*, pages 451–458, 2006.

[8] I. Jolliffe. *Principal Component Analysis*. Wiley Online Library, 2005.

[9] B. Nguyen, C. Morell, and B. De Baets. Large-scale distance metric learning for k-nearest neighbors regression. *Neurocomputing*, 214:805–814, 2016.

[10] Y. Ying and P. Li. Distance metric learning with eigenvalue optimization. *The Journal of Machine Learning Research*, 13(1):1–26, 2012.

[11] Y. Shi, A. Bellet, and F. Sha. Sparse compositional metric learning. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pages 2078–2084, 2014.