Information-Theoretic Metric Learning

Jason Davis, Brian Kulis, Suvrit Sra and Inderjit Dhillon
Dept. of Computer Science
University of Texas at Austin
Austin, TX 78712

Abstract

We formulate the metric learning problem as that of minimizing the differential relative entropy between two multivariate Gaussians under constraints on the Mahalanobis distance function. Via a surprising equivalence, we show that this problem can be solved as a low-rank kernel learning problem. Specifically, we minimize the Burg divergence of a low-rank kernel to an input kernel, subject to pairwise distance constraints. Our approach has several advantages over existing methods. First, we present a natural information-theoretic formulation for the problem. Second, the algorithm utilizes the methods developed by Kulis et al. [6], which do not involve any eigenvector computation; in particular, the running time of our method is faster than most existing techniques. Third, the formulation offers insights into connections between metric learning and kernel learning.

1 Introduction

We propose a new formulation for learning a Mahalanobis distance under constraints. We model the problem in an information-theoretic setting by leveraging an equivalence between the multivariate Gaussian distribution and the Mahalanobis distance. We show that the problem of learning an optimal Mahalanobis distance translates to learning the optimal Gaussian with respect to an entropic objective. Thus, our problem can be thought of as maximizing the entropy of a multivariate Gaussian subject to pairwise constraints on the associated Mahalanobis distance.

To solve our problem, we show an interesting connection to a recently proposed low-rank kernel learning problem [6]. Here, a low-rank kernel K is learned that satisfies a set of given distance constraints as well as minimizes the Burg matrix divergence to the given kernel K_0. It was shown that this problem can be optimized using an iterative optimization procedure with cost $O(cd^2)$ per iteration, where c is the number of distance constraints, and d is the dimensionality of the data. In particular, this method does not require costly eigenvalue computations, unlike many other metric learning algorithms [4, 10, 11].

2 Problem Formulation

Given a set of n points $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$ in $\Re^d$, we seek a positive definite matrix A which parameterizes the Mahalanobis distance:

$$d_A(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i - \boldsymbol{x}_j)^T A(\boldsymbol{x}_i - \boldsymbol{x}_j).$$

We assume that some prior knowledge about the distances between these points is known. Specifically, we consider relationships constraining the similarity or dissimilarity between pairs of points. Two points are similar if the Mahalanobis distance between them is smaller than a given upper bound, $d_A(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq u$ for a relatively small value of u. Similarly, two points are dissimilar if $d_A(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq l$ for sufficiently large l.

In particular, for a classification setting where class labels are known for each instance (as in Globerson and Roweis [4]), distances between points in the same class can be constrained to be small, and distances between two points in different classes can be constrained to be large.

Our problem is to learn a matrix A which parameterizes a Mahalanobis distance that satisifies a given set of constraints. Typically, this learned distance function is used for k-nearest neighbor search, k-means clustering, etc. We note that, in the absence of prior knowledge, these algorithms typically use the standard squared Euclidean distance, or equivalently, the Mahalanobis distance parameterized by the identity matrix I.

In general, the set of distance functions in our feasible set will be infinite (we discuss later how to re-formulate the problem for the case when the feasible set is empty). Therefore, we regularize the problem by choosing the Mahalanobis matrix A that is as close as possible to the identity matrix I (which parameterizes the baseline Euclidean distance function). To quantify this more formally, we propose the following information-theoretic framework.

There exists a simple bijection between the set of Mahalanobis distances and the set of multivariate Gaussians with fixed mean $\boldsymbol{m}$. Given a Mahalanobis distance parameterized by A, we express its corresponding multivariate Gaussian as $p(\boldsymbol{x}; \boldsymbol{m}, A) = \frac{1}{Z} \exp\left(-\frac{1}{2} d_A(\boldsymbol{x}, \boldsymbol{m})\right)$, where Z is a normalizing constant. Using this bijection, we define the distance between two Mahalanobis distance functions parametrized by A_1 and A_2 as the (differential) relative entropy between their corresponding multivariate Gaussians:

$$\mathrm{KL}(p(\boldsymbol{x}; \boldsymbol{m}, A_1) \| p(\boldsymbol{x}; \boldsymbol{m}, A_2)) = \int p(\boldsymbol{x}; \boldsymbol{m}, A_1) \log \frac{p(\boldsymbol{x}; \boldsymbol{m}, A_1)}{p(\boldsymbol{x}; \boldsymbol{m}, A_2)} \, d\boldsymbol{x}. \tag{1}$$

Given a set of pairs of similar points S and pairs of dissimilar points D, our distance metric learning problem is

$$\begin{array}{rll} \min & \mathrm{KL}(p(\boldsymbol{x}; \boldsymbol{m}, A) \| p(\boldsymbol{x}; \boldsymbol{m}, I)) & \\ \text{subject to} & d_A(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq u & (i,j) \in S, \\ & d_A(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq l & (i,j) \in D. \end{array} \tag{2}$$

Note that $\boldsymbol{m}$ is an arbitrary fixed vector.

3 Algorithm

In this section, we demonstrate how to solve the information-theoretic metric learning problem (2) by proving its equivalence to a low-rank kernel learning problem. Using this equivalence, we appeal to the algorithm developed in [6] to solve our problem.

3.1 Equivalence to Low-Rank Kernel Learning

Let $X = [\boldsymbol{x}_1 \ \boldsymbol{x}_2 \ ... \ \boldsymbol{x}_n]$, and the Gram matrix over the input points be $K_0 = X^T X$. Consider the following kernel learning problem, to be solved for K:

$$\begin{array}{rll} \min & D_{\text{Burg}}(K, K_0) & \\ \text{subject to} & K_{ii} + K_{jj} - 2K_{ij} \leq u & (i,j) \in S, \\ & K_{ii} + K_{jj} - 2K_{ij} \geq l & (i,j) \in D, \\ & K \succeq 0. & \end{array} \tag{3}$$

The Burg matrix divergence is a Bregman matrix divergence generated by the convex function $\phi(X) = -\log\det X$ over the cone of semi-definite matrices, and it is defined as

$$D_{\text{Burg}}(K, K_0) = \mathrm{Tr}(KK_0^{-1}) - \log\det(KK_0^{-1}) - n. \tag{4}$$

Formulation (3) attempts to find the nearest kernel matrix in Burg-divergence to the input Gram matrix, subject to linear inequality constraints. It can be shown that the Burg divergence between two matrices is finite if and only if their range spaces are the same [6]. This fact allows us to

conclude that the range spaces of K and K_0 are the same if the problem has a feasible solution. Furthermore, the learned matrix K can be written as a rank-d kernel $K = X^T W^T W X$, for some $(d \times d)$ full-rank matrix W.

We now state a surprising equivalence between problems (2) and (3). By solving (3) for $K = X^T W^T W X$, the optimal A for (2) can be easily constructed via $A = W^T W$. We will not provide a detailed proof of this result; however, we present the two key lemmas.

Lemma 1: $D_{\text{Burg}}(K, K_0) = 2\text{KL}(p(\boldsymbol{x}; \boldsymbol{m}, A) \| p(\boldsymbol{x}; \boldsymbol{m}, I)) + c$, *where c is a constant.*

Lemma 1 establishes that the objectives for information-theoretic metric learning and low-rank kernel learning are essentially the same. It was recently shown [3] that the differential relative entropy between two multivariate Gaussians can be expressed as the convex combination of a Mahalanobis distance between mean vectors and the Burg matrix divergence between the covariance matrices. Here, the two mean vectors are the same, so their Mahalanobis distance is zero. Thus, the relative entropy, $\text{KL}(p(\boldsymbol{x}; \boldsymbol{m}, A) \| p(\boldsymbol{x}; \boldsymbol{m}, I))$, is proportional to the Burg matrix divergence from A to I.

Therefore, the proof of the Lemma 1 reduces to showing that $D_{\text{Burg}}(K, K_0)$ and $D_{\text{Burg}}(A, I)$ differ by only a constant. Interestingly, the *dimensions* of the matrices in these two divergences are different: K and K_0 are $(n \times n)$, while A and I are $(d \times d)$.

Lemma 2: *Given $K = X^T A X$, A is feasible for (2) if and only if K is feasible for (3).*

This lemma confirms that if we have a feasible kernel matrix K satisfying the constraints of (3), the corresponding Mahalanobis distance parameterized by A satisfies the constraints of (2). Note that by associating the kernel matrix with the Mahalanobis distance, we can generalize to unseen data points, thus circumventing a problem often associated with kernel learning.

3.2 Metric Learning Algorithm

Given the connection stated above, we can use the methods in [6] to solve (3). Since the output of the low-rank kernel learning algorithm is W, and we prefer A in its factored form $W^T W$ for most applications, no additional work is required beyond running the low-rank kernel learning algorithm.

Our metric learning algorithm is given as Algorithm 1; each constraint projection costs $O(d^2)$ per iteration and requires no eigendecomposition. Thus, an iteration of the algorithm (i.e., looping through all c constraints) requires $O(cd^2)$ time. Note that a naive implementation would cost $O(cd^3)$ time per iteration (because of the multiplication $L^T W$), but the Cholesky factorization can be combined with the matrix multiplication into a single $O(d^2)$ routine, leading to the more efficient $O(cd^2)$ per iteration running time.

The low-rank kernel learning algorithm which forms the basis for Algorithm 1 repeatedly computes Bregman projections, which project the current solution onto a single constraint. By employing the Sherman-Morrison-Woodbury inverse formula appropriately, this projection—which generally has no closed-form solution—can be computed analytically. Furthermore, it can be computed efficiently on a low-rank factorization of the kernel matrix.

4 Discussion

In this work we formulate the Mahalanobis metric learning problem in an information-theoretic setting and provide an explicit connection to low-rank kernel learning. We now briefly discuss extensions to the basic framework, and we contrast our approach with other work on metric learning.

We consider finding the Mahalanobis distance closest to the baseline Euclidean distance as measured by differential relative entropy. In some applications, it may be more appropriate to consider finding a Mahalanobis distance closest to some other baseline; for example, one could use the Mahalanobis distance parametrized by the inverse of the sample covariance matrix S as a baseline, in which case the resulting Burg divergence problem becomes a minimization of $D_{\text{Burg}}(A, S^{-1})$. We note that extensions of this sort can be solved by variants of our proposed framework.

ALGORITHM 1: Algorithm for information-theoretic metric learning

ITMETRICLEARN(X, S, D, u, l)
Input: X: input $d \times n$ matrix, S: set of similar pairs, D: set of dissimilar pairs, u, l: distance thresholds
Output: W: output factor matrix, where $W^T W = A$
1. Set $W = I_d$ and $\lambda_{ij} = 0 \; \forall \; i, j$
2. Repeat until convergence:

- Pick a constraint $(i, j) \in S$ or $(i, j) \in D$
- Let $\boldsymbol{v}^T$ be row i of X minus row j of X
- Set the following variables:
  1. $\boldsymbol{w} = W\boldsymbol{v}$
  2. **if** (similarity constraint)
     $\gamma = \min\left(\lambda_{ij}, \frac{1}{\|\boldsymbol{w}\|_2^2} - \frac{1}{u}\right)$
     $\beta = \gamma / (1 - \gamma \|\boldsymbol{w}\|_2^2)$
     else if (dissimilarity constraint)
     $\gamma = \min\left(\lambda_{ij}, \frac{1}{l} - \frac{1}{\|\boldsymbol{w}\|_2^2}\right)$
     $\beta = -\gamma / (1 + \gamma \|\boldsymbol{w}\|_2^2)$
  3. $\lambda_{ij} = \lambda_{ij} - \gamma$
- Compute the Cholesky factorization $LL^T = I + \beta \boldsymbol{w}\boldsymbol{w}^T$
- Set $W \leftarrow L^T W$

3. Return W

We consider simple distance constraints for similar and dissimilar points, though it is straightforward to incorporate other constraints. For example, Schutz and Joachims [8] consider a formulation where the distance metric is learned subject to relative nearness constraints on the input points (as in, the distance between i and j is closer than the distance between i and k). Our approach can be adapted to handle this setting. In fact, it is possible to incorporate arbitrary linear constraints into our framework.

Finally, our basic formulation assumes that there exists a feasible point that satisfies all of the distance constraints, but in practice, this may fail to hold. A simple extension to our framework can incorporate slack variables on the distance constraints to handle such infeasible cases.

4.1 Related Work

Xing et al. [11] use a semidefinite programming formulation for learning a Mahalanobis distance metric. Their algorithm aims to minimize the sum of squared distances between input points that are "similar", while at the same time aiming to separate the "dissimilar" points by a specified minimum amount. Our formulation differs from theirs in two respects. First, we minimize a Burg-divergence, and second, instead of considering the sum of distortions over dissimilar points, we consider pairs of constrained points.

Weinberger et al. [10] formulate the metric learning problem in a large margin setting, with a focus on kNN classification. They formulate the problem as a semidefinite programming problem and consequently solve it using a combination of sub-gradient descent and alternating projections. Our formulation does not solely have kNN as a focal point, and differs significantly in the algorithmic machinery used.

The paper of Globerson and Roweis [4] proceeds to learn a Mahalanobis metric by essentially shrinking the distance between similar points to zero, and expanding the distance between dissimilar points to infinity. They formulate a convex optimization problem which they propose to solve by a projected-gradient method. Our approach allows more refined interpoint constraints than just a zero/one approach.

Chopra et al. [1] presented a discriminative method based on pairs of convolutional neural networks. Their method aims to learn a distance metric, wherein the interpoint constraints are approximately enforced by penalizing large distances between similar points or small distances between dissimilar points. Our method is solved more efficiently, and the constraints are enforced incrementally. Furthermore, as discussed above, by including slacks on our constraints, we can accommodate "soft-margin" constraints.

Shalev-Shwartz et al. [9] consider an online metric learning setting, where the interpoint constraints are similar to ours. They also provide a margin interpretation, similar to that of [10]. Their formulation considers distances between all pairs of similar and dissimilar points, whereas we consider only a fixed set of input pairwise constrained points.

Other notable work includes the articles [2, 5, 7, 8]. Crammer et al. [2] applies boosting to kernel learning, for a connection of our method kernel learning see Section 3. Lanckriet et al. [7] study the problem of kernel learning via semidefinite programming. Goldberger et al. [5] proposed neighborhood component analysis to explicitly aid kNN; however, the formulation is non-convex and can lead to local optima.

Acknowledgements This research was supported by NSF grant CCF-0431257, NSF Career Award ACI-0093404, and NSF-ITR award IIS-0325116.

References

[1] S. Chopra, R. Hadsell, and Y. LeCun. Learning a Similarity Metric Discriminatively, withapplication to Face Verification.

[2] K. Crammer, J. Keshet, and Y. Singer. Kernel Design using Boosting. In *NIPS*, 2002.

[3] J. V. Davis and I. S. Dhillon. INSERT TITLE HERE. In *NIPS*, 2006.

[4] A. Globerson and S. Roweis. Metric Learning by Collapsing Classes. In *NIPS*.

[5] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood Component Analysis. In *NIPS*, 2004.

[6] B. Kulis, M. Sustik, and I. S. Dhillon. Learning low-rank Kernels. In *ICML*, 2006.

[7] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan. Learning the Kernel Matrix with Semidefinite Programming. In *JMLR*, 2004.

[8] M. Schutz and T. Joachims. Learning a Distance Metric from Relative Comparisons. In *NIPS*, 2003.

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

[10] K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance Metric Learning for Large Margin Nearest Neighbor Classification. In *NIPS*.

[11] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. In *NIPS*, volume 14, 2002.