A Discriminative Metric Learning Algorithm for Face Recognition

TSUYOSHI KATO¹,a) WATARU TAKEI¹,b) SHINICHIRO OMACHI²,c)

Received: March 11, 2013, Accepted: April 24, 2013, Released: July 29, 2013

Abstract: Face recognition is a multi-class classification problem that has long attracted many researchers in the community of image analysis. We consider using the Mahalanobis distance for the task. Classically, the inverse of a covariance matrix has been chosen as the Mahalanobis matrix, a parameter of the Mahalanobis distance. Modern studies often employ machine learning algorithms called metric learning to determine the Mahalanobis matrix so that the distance is more discriminative, although they resort to eigen-decomposition requiring heavy computation. This paper presents a new metric learning algorithm that finds discriminative Mahalanobis matrices efficiently without eigen-decomposition, and shows promising experimental results on real-world face-image datasets.

Keywords: face recognition, metric learning, Mahalanobis distance, optimization, nearest neighbor classifier

1. Introduction

The problem of face recognition has continued to be tackled by many researchers in the field of image analysis, pattern recognition, and psychology, supported by a variety of applications such as biometric verification, surveillance, and database investigation [1], [5]. A face recognition system is an integration of various technologies including sensing devices, image processing, and pattern recognition. Among those technologies, pattern recognition is the most necessary technique and controls the performance of the overall face recognition system. This paper contributes to the stage of pattern recognition that identifies an unknown person from a still cropped image of the frontal face.

Many face recognition methods linearize a face image into a vector, to pose a statistical multiclass classification problem. Several attempts including PCA [1], [5], FDA [1], Mahalanobis distance [7] and their variants have been performed for the face recognition task. Our work focuses on the nearest neighbor approach which not only performs classification, but also provides useful information about how the input image is classified by showing its nearest neighbors. This property is suitable for interactive systems such as image search. This study employs Mahalanobis distance to improve classification accuracy (See Fig.1). The Mahalanobis distance is expressed as

\[ D_{\text{maha}}(x, m; W) = \frac{1}{2} \text{tr}(W(x - m)(x - m)^\top) \]

which computes a deviation of an input vector \( x \) from a specified point \( m \), with a Mahalanobis matrix \( W \).

Classically, the parameter \( m \) is set to the mean vector in the class, and \( W \) is to the inverse of the covariance matrix modified to address the small sample size problem. A single Mahalanobis matrix common to all classes is sometimes used by taking the inverse of the covariance matrix averaged over classes. In both the ways, the parameters are determined only with positive data, with negative data, or data in the other classes discarded. Such approaches are called generative learning, which is a contrasting manner of so-called discriminative learning exploiting negative data to improve the classification boundaries. A classical statistical analysis, Fisher discriminant analysis, is a well-known example of discriminative learning, that yields different prediction results except for the binary classification case. Due to the enormous number of classes and the limitation of computational resources, generative learning approaches were often employed in the 1990s.

The advent of binary classifiers, such as the support vector machine (SVM), renewed the understanding of the importance of exploiting negative data, and bore a sequence of studies that developed learning machines to find the Mahalanobis matrices in a discriminative learning fashion [2], [3], [6], [7]. The learning of Mahalanobis matrices became metric learning. The learned Mahalanobis matrices are often presumed to be used subsequently in the nearest neighbor classifier. In this setting, metric learning methods determine \( W \) with many triplets, \( R_1, \ldots, R_K \), each of which is given by \( R_k = (x_{i_k}, x_{j_k}, x_{l_k}) \in \mathbb{R}^{n \times 3}, \forall k = 1, \ldots, K \), and \( x_{i_k} \) and \( x_{l_k} \) belong to different classes, but \( x_{j_k} \) belongs to the same class as that of \( x_{i_k} \), so that

\[ D_{\text{maha}}(x_{i_k}, x_{j_k}; W) \geq D_{\text{maha}}(x_{i_k}, x_{l_k}; W) + \epsilon. \]

\( \forall k = 1, \ldots, K \), where \( \epsilon \) is a small positive constant. In other words, differences below \( \epsilon \) between two distances, \( D_{\text{maha}}(x_{i_k}, x_{j_k}; W) \) and \( D_{\text{maha}}(x_{i_k}, x_{l_k}; W) \), are penalized as a loss function in their learning algorithms. This approach is called the relative distance penalization.

In this paper, we present a new metric learning algorithm aimed at tuning a nearest neighbor classifier for face recognition. Our algorithm is based on an implementation of the popular software liblinear developed by the libsvm team. The liblinear

---

¹ Graduate School of Engineering, Gunma University, Kiryu, Gunma 376-8515, Japan  
² Graduate School of Engineering, Tohoku University, Sendai, Miyagi 980-8579, Japan  
a) katotsu@cs.gunma-u.ac.jp  
b) takei-wataru@kato-lab.cs.gunma-u.ac.jp  
c) machi@ecei.tohoku.ac.jp
the square of the is the k
stant called the regularization parameter, and \( O[n] \). That property theoretically guarantees to cut down, to decomposition, which is computationally expensive but nec-
et al.’s algorithm \( [4] \), and so does our metric learning algorithm.
of BDRM inherits the property of the linear convergence of Hsieh
ric learning algorithm as an instance of BDRM. The framework
gence regularized machine
optimization algorithm for the generalized regularization func-
regularization function to Bregman divergences, and extend their
accurate solution*1 of the optimal Mahalanobis matrix.

\*1 The two variables, \( \epsilon \) and \( \epsilon \), are distinct in this paper.

2. Related Work

Importantly, Mahalanobis matrices must be positive definite. Metric learning algorithms typically follow the modern machine learning techniques by finding a Mahalanobis matrix that minimizes the sum of a regularization function and a loss function over the positive definite cone \( [2], [6], [7] \). To ensure this constraint, many methods \( [6], [7] \) project, in every iteration, the \( n \times n \) symmetric matrix onto the positive definite cone, which requires the eigen-decomposition to be computationally intensive, \( O(n^3) \), where \( n \) is the number of features.

Davis et al. developed a break-through algorithm, ITML \( [2] \), that addresses the issue of the expensive computational cost for keeping the positive definiteness. ITML employs the Bregman divergences for a regularization function as well as a loss function. They discovered a surprising fact that each iteration of the successive projection algorithm can be performed in \( O(n^2) \) computation, if LogDet divergence is chosen as a Bregman divergence, yet the positive definiteness of \( W \) and the linear convergence of the iterative algorithm are still guaranteed.

A major difference of ITML from the other metric learning methods \( [6], [7] \) is that users have to give, in advance, two constant parameters that represent the lower bound \( b_l \) for distances between examples in different classes, and the upper bound \( b_u \) for distances between examples in same classes. The loss function is formulated with the LogDet divergence to evaluate how the pre-defined two bounds are violated, instead of using the relative distance penalization.

A contribution of this paper is to demonstrate that a metric learning algorithm using the relative distance penalization can be constructed, yet possessing theoretical guarantee of keeping the computational cost of each iteration \( O(n^2) \) and still ensuring the positive definiteness of the Mahalanobis matrix \( W \) without requiring two user-defined bounds, \( b_l \) and \( b_u \).

3. Bregman Divergence Regularized Machine

In this section, we present a new framework for learning machines named BDRM. Bregman divergence is a class of a large number of functions including the squared Euclidean distance, Itakura-Saito distance, KL-divergence etc. Bregman divergence is defined with a seed function \( \varphi \) which is assumed to be continuously-differentiable, real-valued and strictly convex. A Bregman divergence \( D_\varphi : \text{dom}\varphi \times \text{ri}(\text{dom}\varphi) \to [0, +\infty) \) is constructed with \( \varphi \) as

\[
D_\varphi(x; y) = \varphi(x) - \varphi(y) - \langle x - y, \nabla \varphi(y) \rangle
\]

where \( \text{dom}\varphi \) is \( \varphi \)'s domain, and \( \text{ri}(\text{dom}\varphi) \) is the relative interior of \( \text{dom}\varphi \). We consider the following class of learning machines:

\[
\min_{w} D_\varphi(w; w_0) + \frac{1}{2} \sum_{k=1}^{K} c_k \xi_k^2, \quad w \in \text{dom}\varphi
\]

subject to \( \forall k \langle a_k, w \rangle \geq \xi_k - \hat{\xi}_k, \quad \hat{\xi}_k \geq 0 \),

where \( w_0 \in \text{ri}(\text{dom}\varphi), \quad c = [c_1, \ldots, c_K]^\top > 0_K, \quad \epsilon = [\epsilon_1, \ldots, \epsilon_K]^\top \geq 0_K, \quad \forall k, a_k \in \mathbb{R}^n \setminus \{0_k\}. \)

L2-SVM is shown to be an instance of these learning machines by
Algorithm 1 General BDRM.

1: \( v := \nabla_{\phi}(w_0); a := 0; \)
2: for \( t = 1, 2, \ldots \) do
3: for \( k = 1, \ldots, K \) do
4: \( u_k = v - a_k a_k; \)
5: if \( (a_k, \nabla_{\phi}(u_k)) \geq \epsilon \) then
6: \( a_k = 0; \)
7: else
8: Find positive \( a_k \) satisfying a nonlinear equation
\[ (a_k, \nabla_{\phi}(u_k - a_k a_k)) = \epsilon_k - a_k / c_k. \] (2)
9: end if
10: \( v := u_k + a_k a_k; \)
11: end for
12: end for

setting \( \phi(w) = \frac{1}{2}||w||^2 \), \( w_0 = 0_k, a_k = y_k x_k, \) \( c = C1_K \), and \( \epsilon = 1_K \). Hsieh et al. [4] dualize the optimization problem to solve the primal problem by maximizing the dual objective function with respect to dual variables. Introducing dual variables, \( \alpha \in \mathbb{R}^K \), the dual function of Eq. (1) is given by:

\[ q(\alpha) = \inf_{w \in \mathcal{X}} L(w, \alpha) \]

where \( L(w, \alpha) \) is the Lagrange function written as:

\[ L(w, \alpha) = D_{\alpha}(w; w_0) + \frac{1}{2} \sum_{k=1}^{K} c_k \alpha_k^2 + \sum_{k=1}^{K} \alpha_k(\epsilon_k - (a_k, w) - \xi_k). \]

Note that the non-negativeness of slack variables \( \xi_k \) is ensured even without Lagrange multipliers for the constraints. Following their approach, we employ the coordinate ascent method for maximizing the dual problem, and we obtain Algorithm 1.

Here, \( \phi(\cdot) \) denotes the convex conjugate of \( \phi(\cdot) \).

This algorithm inherits a favorable property of Hsieh et al.’s algorithm, linear convergence, because Algorithm 1 still exactly performs a linearly converged coordinate ascent method exactly.

If \( g(a) + \epsilon \geq g(a^*) \), where \( a^* \) is the optimal solution, the solution \( a \) is said to be \( \epsilon \)-accurate. Thanks to the line convergence, an \( \epsilon \)-accurate solution is obtained in \( O(\log(1/\epsilon)) \) iterations. Another point that determines the total time complexity is how fast each iteration works. Since Step 4 and Step 10 need only \( O(n) \) computation, the time complexity of each iteration depends on the computation of \( \nabla_{\phi} \), for Step 5 and Step 8. Hence, if the Bregman divergence is chosen so \( \nabla_{\phi} \) can be computed quickly, the algorithm works efficiently even in a large scale scenario.

4. Metric Learning as a BDRM

This section employs the framework of BDRM to devise a new metric learning algorithm without a time-consuming step of projection onto the positive definite cone. We here consider a multi-class classification setting. First of all, we pick many triplets \( R_1, \ldots, R_K \) from training data where each triplet contains three examples \( R_k = (x_i, x_j, x_l) \) where \( x_i \) and \( x_j \) are in different classes, but \( x_l \) is in the same class as that of \( x_l \). We have denoted the number of triplets by \( K \). The acquired Mahalanobis distance is used in the prediction stage using a nearest neighbor classifier. Ideally, we wish to obtain a Mahalanobis matrix \( W \) such that \( D_{\text{maha}}(x_i, x_j; W) \geq D_{\text{maha}}(x_i, x_k; W) + \epsilon, \) where \( \epsilon \) is a small positive constant. In our experiments described later, we form the \( R_k \)’s as follows: For each class, take every pair of examples, \( i \) and \( l \), in the class; then, for each pair \((i, l)\), find five nearest neighbors of \( i \) according to the Euclidean distance; finally, form a triplet as \( R_k = (i, j, l) \) for each pair \((i, l)\) and each of its nearest neighbors \( j \).

We use a regularization function to avoid over-fitting. Following ITML, we employ LogDet divergence defined by

\[ D_{\text{ld}}(W; W_0) = \text{tr} (W W_0^{-1}) - \log \det W W_0^{-1} - n \]
as a regularization function. The LogDet divergence is constructed by defining the seed function as \( \phi_{\text{ld}}(W) = -\log \det W \) whose derivative is given by \( \nabla_{\phi_{\text{ld}}}(W) = -W^{-1} \). Usually, we set \( W_0 = I_n \) to restrain a Mahalanobis metric far from the Euclidean metric, but different choices of \( W_0 \) are also useful for some special settings such as transfer learning. Furthermore, to ensure the feasibility of the learning problem, we introduce slack variables \( \xi_k \).

Then, we obtain the following optimization problem:

\[ \min_{W} D_{\text{ld}}(W; W_0) + \frac{C}{2} \sum_{k=1}^{K} \xi_k^2, \]

wrt \( W \in \mathbb{S}^+_n \), and \( \xi \in \mathbb{R}^K \) subject to \( \forall k, D_{\text{maha}}(x_i, x_j; W) \geq D_{\text{maha}}(x_i, x_k; W) + \epsilon - \xi_k \)

where \( W_0 \geq O_n, C > 0 \), and \( \epsilon > 0 \) are constant, and \( \mathbb{S}^+_n \) denotes a set of \( n \times n \) strictly positive definite matrices. It can be verified that this formulation is a BDRM with

\[ w = \text{vec}(W), \quad w_0 = \text{vec}(W_0), \quad c = C1_K, \quad \epsilon = \epsilon1_K, \quad \text{and} \quad a_k = \text{vec}(A_k) \]

where \( A_k \equiv (x_j - x_l)(x_j - x_l)^\top - (x_i - x_l)(x_i - x_l)^\top \).

Letting \( Z_k \equiv (x_j - x_l)x_i^\top, \) Algorithm 1 for this setting is re-written in Algorithm 2. Derivation is given in Appendix A.1. The single-variable nonlinear system in Step 10 can be solved by the Newton-Raphson method, and a sufficiently precise solution is obtained with around ten iterations.

In case of \( \alpha_k = 0 \), Step 4 and Step 12 in Algorithm 2 can be done in time \( O(n^2) \) just by copying matrices: \( Y_k = W \) and \( W = Y_k \), respectively. Even if \( \alpha_k > 0 \), only \( O(n^2) \) computations accomplish the two steps as

\[ Y_k = W - W Z_k (Q + \alpha_k^{-1} E)^\top Z_k W, \]
\[ W = Y_k - Y_k Z_k (P - \alpha_k^{-1} E)^\top Z_k Y_k, \]

which are derived with the well-known matrix inversion lemma, where we have defined \( Q \equiv Z_k^\top W Z_k \) and \( E = \text{diag}(+1, -1) \).

As a consequence, we are ready to show the main result:

**Theorem 1.** Algorithm 2 achieves an \( \epsilon \)-accurate solution for problem Eq. (3) in computational time \( O(n^2K \log(1/\epsilon)) \). \( \square \)

5. Experiments

We tested the proposed method, BDRM, on two face databases,
In this paper, we presented a new metric learning algorithm that finds discriminative positive definite Mahalanobis matrices efficiently without eigen-decomposition, and showed promising experimental results on real-world face-image datasets.

ITML[2] introduces a loss function based on LogDet divergence that has seldom been employed in other studies in order to update W in O(n^2) computation. Another important fact utilized by ITML to achieve O(n^2) computational time for an update is the n × n coefficient matrix in a constraint — which corresponds to A_k in BDRM — derived from a doublet is one-rank. Our study employs triplets R_k to form constraints that make A_k two-rank, which disables use of the ITML’s approach. We tackled this is-
suę by employing Newton method instead of deriving a closed-form update rule, and found out that the computational time is still $O(n^2)$ even by using Newton method. Thus, this study disclosed that the Mahalanobis matrix $W$ can be updated in $O(n^2)$ and achieves an excellent generalization performance, even when using $\ell_2$-loss and relative distance penalization, both of which are accepted widely in many machine learning studies (e.g., Ref. [7]).

Acknowledgments TK was supported by Grant-in-Aid for Scientific Research (C) 23500373.

References


(Appended by Koichi Shinoda)

Appendix

A.1 Derivation of Algorithm 2

Here we derive Algorithm 2 from Algorithm 1. To this end, we first introduce two symmetric matrices $U_k$ and $V$ such that $\text{vec}(U_k) = u_k$ and $\text{vec}(V) = v$, respectively. We then show the following claims by induction:

- $Y_k = \nabla \varphi(\alpha_k U_k)$,
- $P_{1,1} - P_{2,2} = (A_k, \nabla \varphi(\alpha_k U_k))$,
- Equivalence between Eq. (2) and Eq. (5), and
- $W = \nabla \varphi(V)$.

We assume the four claims have been maintained until $(t, k-1)$-th iteration. Then, we have

\[ Y_k = (W^{-1} + \alpha_k A_k)^{-1} = (-V + \alpha_k A_k)^{-1} = (-U_k)^{-1} = \nabla \varphi(A_k U_k). \]

Since $A_k = Z_k E Z_k^\top$, we get

\[ (A_k, \nabla \varphi(U_k)) = (Z_k E Z_k^\top Y_k) = (E, Z_k^\top Y_k Z_k) = (E, P) = P_{1,1} - P_{2,2}. \]

Equivalence between Eq. (2) and Eq. (5) can be shown by the equality between the lefthand sides of the two equations as

\[ (A_k, \nabla \varphi(U_k - \alpha_k A_k)) = \frac{p_{2,2} - p_{1,1} + 2\alpha_k}{(p_{1,1} - \alpha_k)(p_{2,2} + \alpha_k) - p_{1,2} p_{2,1}}. \]

The last claim is derived as

\[ W = (Y_k^{-1} + \alpha_k A_k)^{-1} = (-U_k + \alpha_k A_k)^{-1} = (-V)^{-1} = \nabla \varphi(V). \]

Thus, all the four claims have been shown, establishing the equivalence between Algorithm 1 and Algorithm 2 in the setting of Eq. (4).