METRIC BASED GAUSSIAN KERNEL LEARNING FOR CLASSIFICATION

Zhenyu Guo and Z. Jane Wang, Senior Member, IEEE

University of British Columbia
Department of Electrical and Computer Engineering
Vancouver, B.C. Canada
{zhenyug, zjanew}@ece.ubc.ca

ABSTRACT

Metric learning for KNN has attracted increasing attentions in the field of machine learning (e.g., based on the parametric form of Mahalanobis distance). A good distance metric is also the foundation for other machine learning models, for example, a Gaussian RBF kernel is constructed upon distance metric defined in the feature vector space. However, besides the KNN classifier, there is little research work on learning a good distance metric for distance-based models. In this paper, we propose a novel algorithm to learn a Mahalanobis-distance type metric for Gaussian RBF kernels. We conduct experiments on 5 data sets from the UCI Machine Learning Repository database and two face recognition data sets. The classification results show that the proposed algorithm can outperform other state-of-arts on most of the data sets and achieve comparable results on the rest of data sets.

Index Terms— Metric Learning, Gaussian Kernel, Multiple Kernel Learning, Riemannian Manifold

1. INTRODUCTION

Distance metric defined on the data samples plays an important role in classification, clustering, ranking and other machine learning topics. For example, a k-nearest neighbor (KNN) classifier decides the class label of a testing data point directly based on the distances between this testing point to other training points. Therefore, learning a good distance metric for KNN classifiers, referred as metric learning, has become an active research direction recently [1–3]. Metric learning algorithms aim at learning a distance metric such that data points from the same class are closer under this metric and data points from different classes are farther. Such metric learning methods usually involve Semi-Definite Programming (SDP) to optimize over the cone of Positive Definite (PD) matrices. It is worth noting that distance metric is also important for other models in machine learning, besides KNN classifiers. For example, a Gaussian radial basis function (RBF) kernel is constructed based on a distance metric defined in the feature space.

Gaussian RBF kernels have been successfully applied with kernel machines like SVM and Multiple Kernel Learning in computer vision, signal processing and natural language processing. Traditionally, a Gaussian Kernel is defined by a single normalization parameter (a scaler) based on Euclidean distance. And most previous kernel learning methods [4–6] try to optimize over that single parameter to minimize the structural risk for classification. Among these methods, one category tries to learn the PD metric matrix directly from the training data [4], and the other category tries to learn a convex combination of basis kernel functions [5, 6], referred as Multiple Kernel Learning (MKL), to achieve the goal. It is obvious that the fixed Euclidean distance metric limits the degree of freedom for the learning methods in terms of pursuing a good RBF kernel function. In this paper, we propose learning a distance metric for Gaussian RBF kernels used in kernel machines for classification tasks, and we refer it as Metric based Kernel Learning (MetricKL). It is related to metric learning since the objective of our MetricKL is to find a good distance metric matrix; It is also related to kernel learning since the final output of MetricKL is an optimal Gaussian RBF kernel which can be used for kernel machines, like SVM, to perform classification. However the proposed MetricKL is substantially different from both conventional metric learning and kernel learning, and it can be seen as a combination of them. Inspired by multiple kernel learning, our MetricML is to learn the optimal metric matrix by finding a convex combination of base metric matrices, instead of learning the metric matrix directly as in conventional metric learning [1–3]. We summarize the contributions of this paper as follows:

- We propose a parametric form of a Gaussian kernel based on Mahalanobis-distance type metric.
- We propose a sampling scheme to generate base metric matrices along Riemannian geodesic.
- We adopt a multiple kernel learning optimization method to learn the optimal metric matrix for the proposed Gaussian kernel.

The paper is organized as follows. Section 2 will describe the Gaussian RBF kernel and its parametric form based on Mahalanobis-distance type metric. Section 3 will present an approach for generating base metric matrices for the proposed MetricML, and Section 4 will describe the MetricKL algorithm. The classification results on real data sets will be shown in Section 5.

2. METRIC PARAMETRIC FORMULATION FOR GAUSSIAN RBF KERNEL

In this section, we will provide a parametric formulation of the Gaussian RBF kernel based on Mahalanobis-distance type metric, which is a convex combination of base metric matrices. A Gaussian RBF kernel function is traditionally defined on Euclidean distance as:

$$k(x_i, x_j) = \exp\left(-\gamma||x_i - x_j||_2^2\right), \tag{1}$$

where x_i and x_j are two vectors from the feature space and γ means the scaler normalization parameter. This RBF function arises from the statistical assumption that data points follow an underlying i.i.d. Gaussian distribution in the original feature space. Now let's rewrite the above kernel function with a Mahalanobis metric as

$$k(x_i, x_j) = \exp\left(-\gamma(x_i - x_j)^T I(x_i - x_j)\right),$$

which is clearly equivalent to Eq.(1), where the metric is the identity matrix I.

In practice, often the data points don't follow an i.i.d. Gaussian distribution in the original feature space, but may follow a Gaussian distribution in another space where the data is projected by a linear projection operation L from the original feature space. With applying the linear projection L, we can obtain a kernel function as

$$\begin{aligned} k(x_i, x_j) &= \exp\left(-\gamma(Lx_i - Lx_j)^T(Lx_i - Lx_j)\right) \\ &= \exp\left(-\gamma(x_i - x_j)^T L^T L(x_i - x_j)\right) \\ &= \exp\left(-\gamma(x_i - x_j)^T M(x_i - x_j)\right), \end{aligned}$$

where M, with $M = L^T L$, can be considered as a metric matrix. To learn a good kernel function, both the normalization parameter γ and the metric M are important. Traditional Gaussian kernel learning methods are limited to learning only γ, which makes them unable to find the correct similarity between data points in practical applications. Now we propose a convex parametric form on M to learn a better kernel function k.

Inspired by the success of MKL, we propose using a convex combination of N base metric matrices M_s's to represent the metric M and the kernel is now defined as:

$$k(x_i, x_j) = \exp\left(-\gamma(x_i - x_j)^T \sum_{s=1}^{N} d_s M_s (x_i - x_j)\right) \tag{2}$$

$$= \exp\left(-\gamma \sum_{s=1}^{N} d_s (x_i - x_j)^T M_s (x_i - x_j)\right) \tag{3}$$

$$= \prod_{s=1}^{N} \exp\left(-\gamma d_s (x_i - x_j)^T M_s (x_i - x_j)\right), \tag{4}$$

where M_s is the s_{th} metric matrix from the set of N base matrices, and $\mathbf{d} = (d_1, d_2, ..., d_s, ..., d_N)$ means the weight coefficient vector, which is constrained by $\mathbf{d} \succeq \mathbf{0}$. Each M_s is a PD matrix, and thus the convex combination $M = \sum_{s=1}^{N}(d_s M_s)$ is also a PD matrix and represents a valid metric matrix. By simply replacing γd_s jointly by a new d_s in the equation above, we can simplify the parametric kernel function to be only on the parameter vector $\mathbf{d}$. With the following definition

$$k_s(x_i, x_j) = \exp\left(-(x_i - x_j)^T M_s (x_i - x_j)\right), \tag{5}$$

Eq.(2) can be rewritten as

$$k_{\mathbf{d}}(x_i, x_j) = \prod_{s=1}^{N} k_s(x_i, x_j)^{d_s}, \tag{6}$$

which can be viewed as a weighted product of base kernels for a fixed set of $\{M_s\}$. The problem of learning an optimal metric M is now reduced to the problem of learning an optimal weight coefficient vector $\mathbf{d}$ for this product kernel. In Section 4, we will describe how to find the optimal $\mathbf{d}$ based on Generalized Multiple Kernel Learning (GMKL) [6].

3. BASE METRIC MATRICES GENERATION BASED ON COVARIANCE MATRIX

In the previous section, we successfully formulate the metric learning problem for a Gaussian RBF kernel into a product kernel learning problem. The next question to ask is how to generate the base metric matrices to form the convex combination. Although the multiple kernel learning methods became more and more mature, most of the existing algorithms neither can reach the true global optimum or reach the optimum for the exact objective function. Therefore, good base metric matrices for the combination is very important for our learning task.

Recall that the Mahalanobis distance was originally proposed to make use of the precision matrix S (the inverse of the covariance matrix) of the training data, it is natural to consider the precision matrix as a good base metric, which indeed shows good performances and has a strong relationship with the underlying statistical characteristics of the data

set. Research on the estimation of covariance matrix and precision matrix [7, 8] suggests that the combination of a sample precision matrix and a target matrix often works well for the estimation purpose. The identity matrix I (or its weighted version) is often chosen as the target matrix [7, 8]. Inspired by this observation, we include the identity matrix as one base metric in our learning problem.

In the conventional multiple kernel learning, base kernels are usually generated by selecting parameters that span the parameter space on a grid. Since the metric matrix is a PD matrix which lies on a Riemannian manifold, it is too expensive to select the matrices that span the Riemannian manifold for our learning purpose. However, the geodesic connecting the identity matrix I and the precision matrix S actually contains matrices which are geometrically "between" these two matrices. Since both I and S are two candidate base metric matrices, it is reasonable to believe that the matrices along their geodesic have the similar good properties for being a candidate metric. Therefore we propose sampling matrices along the geodesic between I and the precision matrix S to form the basis set $\{M_s\}$. The geodesic $\gamma(t)$ between two PD matrices is given as [9],

$$\gamma(t) = \mathbf{P_1}^{\frac{1}{2}}(\mathbf{P_1}^{-\frac{1}{2}}\mathbf{P_2}\mathbf{P_1}^{-\frac{1}{2}})^t\mathbf{P_1}^{\frac{1}{2}}, \tag{7}$$

where $\mathbf{P1}$ and $\mathbf{P_2}$ are two PD matrices on the Riemannian manifold, and t is "velocity" and $t \in [0, 1]$. Here t being from 0 to can be seen as moving from $\mathbf{P_1}$ to $\mathbf{P_2}$ along the geodesic line. By uniformly sampling $t \in [0, 1]$, we can uniformly sample matrices along the geodesic. The calculation of $\gamma(t)$ can be done efficiently by Eigen decomposition.

4. METRIC BASED KERNEL LEARNING

In Section 2 and Section 3, we have formulated our proposed Metric based Kernel Learning (MetricKL) problem into a product kernel learning task, and we have proposed an approach to generate good base metric matrices base on the Riemannian geometry and precision matrix. In this section, we will describe an optimization algorithm to learn the optimal coefficient vector $\mathbf{d}$ for the metric combination.

To use the parametric Gaussian RBF kernel for classification, we consider the popular SVM with multiple kernels, which can be described as an optimization problem [6],

$$\max_{\alpha} \mathbf{1}^T\alpha - \frac{1}{2}\alpha^T\mathbf{Y}\mathbf{K_d}\mathbf{Y}\alpha + r(\mathbf{d}), \tag{8}$$

$$s.t. \mathbf{1}^T\mathbf{Y}\alpha = 0, 0 \leq \alpha \leq C, \tag{9}$$

$$\mathbf{d} \succeq \mathbf{0}, \tag{10}$$

where $\mathbf{K_d}$ is the kernel matrix computed by $k_\mathbf{d}$ in Eq.(6). Please refer to [6] for details about the above optimization problem. [6] provided an efficient gradient descent based algorithm to solve Eq.(8), which is referred as GMKL and is adopted here by us to solve our MetricKL problem. To better summary the proposed MetricKL algorithm, we describe the steps in Table 1.

Algorithm 1- MetricKL
1: Compute the sample covariance matrix Σ from the training data
2: Compute the sample precision matrix S by inversing Σ
3: Sample N PD matrices M_s by Eq.(7)
4: Construct N base kernel matrices by Eq.(5)
5: Find the optimal $\mathbf{d}^*$ by solving Eq.(8) by GMKL
6: The optimal metric M^* is obtained as $\sum_{s=1}^{N} d_s^* M_s$

Table 1. The MetricKL algorithm for learning the optimal metric for Gaussian RBF kernels.

For the implementation of MetricKL, as described in Table 1, the inverse operation in **step 2** can be done by Eigen decomposition and clipping the spectrum to make sure the resulting inverse matrix is still PD. **Step 1** can be usually performed by calculating the conventional sample covariance matrix as $\Sigma = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})^T$, where n is the sample size, x_i is the i_{th} data point and $\bar{x}$ is the sample mean. However, when the sample size n is much smaller than the data dimension p, such sample covariance matrix cannot yield a satisfactory estimation of the true covariance matrix, which is called the high-dimensional $n << p$ problem. This problem is an active research area itself, methods like [8, 10] can solve this problem by assuming the sparsity assumption. Here due to space limit, we don't include the details on estimating covariance and precision matrices. Please refer to the related references.

5. EXPERIMENT

In this section, we will evaluate the proposed MetricKL algorithm for classification tasks on 5 UCI machine learning data sets, the ORLFace data set and the YaleFace data set. For comparison, we also include the classification results from several state-of-art methods in metric learning and kernel learning: LMNN [1], GMKL [6]. The KNN classifier and the standard SVM are also considered as baseline methods.

5.1. UCI Machine Learning Repository

We first test on 5 data sets from UCI Machine Learning Repository: Ionosphere, Sonar, Iris, Wdbc, Wine. We use the data files provided by LIBSVM [11] which are scaled to $[-1, 1]$. For LMNN, the parameters are set according to [1]. For SVM, GMKL and MetricKL, the weight for slack variables C is set to 100. In GMKL, the 11 standard Gaussian RBF base kernels are combined by taking their product. And in MetricKL, we sample 11 metric matrices by taking $t \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$, and then the 11 base kernels are computed by Eq.(5). For all data sets, we run 10-fold cross-validation to evaluate the classification

data set	KNN	**MetricKL**	GMKL [6]	LMNN [1]	SVM
ionosphere	36.11 ± 3.60	**5.56** ± 3.46	6.67 ± 3.26	13.89 ± 6.42	7.5 ± 3.48
wine	5.26 ± 3.51	**1.58** ± 2.54	1.58 ± 2.54	5.26 ± 3.51	4.21 ± 5.44
iris	4.00 ± 4.66	4.67 ± 4.50	6.00 ± 4.92	4.00 ± 4.66	5.33 ± 4.22
wdbc	8.45 ± 1.90	**7.93** ± 4.39	11.21 ± 3.28	8.45 ± 1.90	10.17 ± 2.63
sonar	53.13 ± 2.30	14.09 ± 5.85	12.27 ± 6.79	15.00 ± 7.44	12.27 ± 6.45
ORLFace	12.50 ± 3.24	**2.25** ± 0.79	3.08 ± 1.11	9.33 ± 2.18	3.17 ± 1.17
YaleFace	44.5 ± 6.67	**24.67** ± 6.93	26.83 ± 5.58	39.83 ± 5.90	29.17 ± 4.86

Table 2. The classification results for different methods on all seven data sets. The average classification errors and standard deviations are shown in percentage(%). We highlight the results of the proposed MetricKL when it performs the best on the specific data set.

Fig. 1. Sample images from ORLFace data set for one subject.

Fig. 2. Sample images from YaleFace data set for one subject.

performances of different methods. The classification error results are reported in Table 2 in percentage.

In Table 2, we can see that the proposed MetricKL outperforms other methods on 3 data sets out of 5, and it achieves comparable results on the left 2 data sets. The proposed MetricKL outperforms GMKL and SVM in most of the cases, which demonstrates that the proposed metric parametric formulation indeed provides more degree-of-freedom for kernel learning and it can learn a better kernel function than state-of-art kernel learning algorithms which mainly focus on learning a few kernel parameters. Also, it is worth noting that the proposed MetricKL outperforms KNN and the metric learning method LMNN consistently. This demonstrates the advantages of learning a distance metric in the kernel space.

5.2. Face Recognition

We also test the proposed MetricKL algorithm for face recognition tasks on two data sets. ORLFace data set, shown in Fig. 1, contains 400 gray scale images of 40 subjects in 10 different poses. we down-sampled the images to 32×32 pixels and then used PCA to further reduce the dimensionality to 50. In the experiment, training set is constructed by randomly sampling 7 images per subject and the rest 3 images are used for testing. Therefore the face recognition problem is a 40-way classification task. The parameters of different models are the same as in the previous section. We run 50 random trials for all methods and report the average classification errors in Table 2.

YaleFace data set, shown in Fig. 2, contains 165 gray scale images of 15 individuals. There are 11 images per subject, one per different facial expression or configuration: center-light, w/glass, happy, left-light, w/no glass, normal, right-light, sad, sleepy, surprised, and wink. For the training/testing splitting, we follow the scheme used for ORLFace by randomly sampling 7 images for training and the rest 4 images for testing. We run 50 random trials on YaleFace to report the average performance for all methods in Table 2.

From Table 2, we can see that the proposed MetricKL achieves the best performance on both face recognition data sets. Comparing to SVM and GMKL, the significant improvement of MetricKL further illustrates that the proposed algorithm is able to learn a good distance metric for Gaussian RBF kernels efficiently.

6. CONCLUSION

In this paper, we present a MetricKL algorithm to learn a Mahalanobis-distance type metric for Gaussian RBF kernels based on the convex combination of base metric matrices. To generate good base metric matrices, we propose a sampling scheme along the Riemannian geodesic between the identity matrix and the precision matrix. The classification results on 5 UCI machine learning data sets and on 2 face recognition data sets show that the proposed MetricKL algorithm generally improves the Gaussian RBF kernel comparing to other kennel learning methods. The results also show that MetricKL could achieve better performances on most of the data sets when compared with several state-of-art metric learning methods. The promising results support the concept of learning a metric for general distance based models. Since Gaussian functions are also widely used in Conditional Random Field, Semi-supervised Learning and other machine learning areas, it is possible to learn a good distance metric for those applications too in the future.

7. REFERENCES

[1] Kilian Q. Weinberger and Lawrence K. Saul, "Distance metric learning for large margin nearest neighbor classification," *Journal of Machine Learning Research (JMLR)*, 2009.

[2] Jun Wang, Huyen Do, Adam Woznica, and Alexandros Kalousis, "Metric learning with multiple kernels," in *Advances in Neural Information Processing Systems (NIPS)*, 211.

[3] Jason V. Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S. Dhillon, "Information-theoretic metric learning," in *International Conference on Machine Learning (ICML)*, 2007.

[4] Gert R.G. Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I. Jordan, "Learning the kernel matrix with semidefinite programming," *Journal of Machine Learning Research (JMLR)*, 2004.

[5] Alain Rakotomamonjy, Francis R. Bach, Stéphane Canu, and Yves Grandvalet, "Simplemkl," *Journal of Machine Learning Research (JMLR)*, 2008.

[6] Manik Varma and Bodla Rakesh Babu, "More generality in efficient multiple kernel learning," in *International Conference on Machine Learning (ICML)*, 2009.

[7] Thomas J. Fisher and Xiaoqian Sun, "Improved stein-type shrinkage estimators for the high-dimensional multivariate normal covariance matrix," *Computational Statistics and Data Analysis*, 2011.

[8] Xiaohui Chen, Z. Jane Wang, and Martin J. McKeown, "Shrinkage-to-tapering estimation of large covariance matrices," *IEEE Transactions on Signal Processing*, 2012.

[9] Maher Moakher, "A differential geometric approach to the geometric mean of symmetric positive-definite matrices," *SIAM Journal on Matrix Analysis and Applications*, 2005.

[10] Jerome Friedman, Trevor Hastie, and Rob Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, 2007.

[11] Chih-Chung Chang and Chih-Jen Lin, "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, 2011.