Multiple Closed-Form Local Metric Learning for K-Nearest Neighbor Classifier

Jianbo Ye

Abstract—Many researches have been devoted to learn a Mahalanobis distance metric, which can effectively improve the performance of kNN classification. Most approaches are iterative and computational expensive and linear rigidity still critically limits metric learning algorithm to perform better. We proposed a computational economical framework to learn multiple metrics in closed-form.

Index Terms—Mahalanobis distance, multiple metric learning, kNN classifier

I. Introduction

NEAREST neighbor algorithm(kNN) [1] is the oldest and simplest methods for classifying objects based on closest training examples in the feature space. Despite its simplicity, the kNN rule often yields competitive results. As a type of instance-based learning, kNN rules in effect compute the decision boundary in an implicit manner.

Most implementation of kNN compute simple Euclidean distances(assuming the examples are represented as vector inputs), when no prior knowledge is available. Unfortunately, Euclidean distance ignores any statistical regularities that might be estimated from a large training set of labeled examples. Motivated by these issues, a number of researchers (such as Xing [2], LDA [3] [4], RCA [5], NCA [6], LMNN [7] [8], MCML [9], ITML [10], BoostMetric [11]) have demonstrated that kNN classification can be greatly improved by learning a Mahalanobis distance(quadratic metric) over the input space, which later was formally called distance metric learning in literature. Most approaches in metric learning convert the learning process to a semi-definite programming(SDP). Some are solved by iterative numerical solvers, while others (were later proven to) like Xing, RCA and LDA, have closed-form solutions [12].

There are two major concerns in previous approaches of metric learning. Firstly, iterative solver computationally converge a metric learning process to a local optimal, such as NCA [6] and its derivatives like LDM [13]. Meanwhile even problem are formulated into a convex optimization and solved by SDP like MCML, ITML and LMNN, there exists a target draft between the objective function and kNN classifier's accuracy. Secondly, whatever formulation of objective in metric learning, the overall performance is restricted to linearity of Mahalanobis distance metric inherently. Hence for data sets which have intrinsic non-linear structures, nonlinear approaches like SVM [14] or kernel learning are expected to perform better at a computational expense.

Jianbo Ye is with the Department of Computer Science, The University of Hong Kong, e-mail: jbye@cs.hku.hk.

In this paper, we proposed a novel computationally economical framework to learning multiple metrics, which benefits from the closed-form method like Linear Discriminant Analysis(LDA [3]) in their efficiency, explores the margin discriminative power like Large Margin Nearest Neighbor(LMNN [7]), and leverage the restriction of linearity and instance-based query efficiency by learning multiple metrics.

Our paper is organized as follows: section II introduces the overall framework of our approach, named *multiple closed-form local metric learning*(CFLML); section III runs over all necessary technical details in CFLML; section IV gives a brief note on efficient implementation; section V provides experimental results of CFLML and some other approaches in literature; we concludes our paper with a discussion on future works in section VI.

II. FRAMEWORK OVERVIEW

A. K-nearest neighbor classification

In the classification phase, k is a user-defined constant, which is chosen heuristically to achieve optimal. The best k basically is determined by the statistical properties of labeled instances, where large values of k reduce the effect of noise on the classification, but make boundaries between classes less distinct. Hence it is somewhat wise to select a larger k if different classes in training set are widely separated. As a comment, some techniques like LMNN are implicitly designed to be applicable for small k(k=1,3).

B. Mahalanobis metric

In short, the Mahalanobis distance of multivariate column vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and a covariance matrix $A_{n \times n}$ (positive semi-definite) is defined as

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T S(\mathbf{x} - \mathbf{y})},$$

where square matrix A is guaranteed to be positive semi-definite, hence it has eigendecomposition

$$A = U^T \Lambda U = L^T L,$$

where $\Lambda_{m \times m}$ is a diagonal matrix formed by non-zero eigenvalues of A, and the columns of U are the corresponding eigenvectors. Set $L_{m \times n} = \Lambda^{1/2}U$, as we see the Mahalanobis distance metric is equivalent to apply a linear transform L over the original vector space.

C. Multiple local metrics

A major limitation of Mahalanobis metric learning is that, it would preserve linear rigidity of the data set. Previous approaches are mainly supposed to make a trade-off when forming the problem to optimize a certain objective function, while enlarge the distance between different labeled pairs and shrink or preserve the distance between pairs of same labels. However, due to the linear rigidity, the trade-off is crucial to the performance of derived kNN classifier.

Note that as an extension to LMNN [7], a multiple metric approaches(MM-LMNN) is proposed in the same paper. Its general idea is to divide the train set into multiple clusters and learn multiple LMNNs individually. However in most cases, this approach does not significantly improve the overall performance.

In stead of dividing train set into clusters, which can be regarded a complementary separation in feature domain, we define scalar valued functions (in terms of metrics) to describe labelling ambiguity over feature domain, and associate instances with different metrics by selecting one with least ambiguity.

D. Overview

The key idea in our framework is that by providing a group of metrics(includes at least one metric) for instances, which we called parents, we could produced a child metric in complementary to the performance of parents. The offspring procedure is a closed-form solution of metric learning process, which is effective to improve the performance of kNN with combination of its parents and much computationally cheaper comparing to other iterative solver based methods.

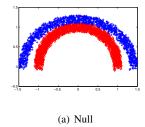
In this paper, we use a simple stochastic local search as follows:

- 1) Set train set and validation set, and an initial metric L_0 , target group of metrics $G = \{L_0\}$. Start iterations.
- 2) In *i*-th step, produce child L_{i+1} from parent $\{L_i\}$.
- 3) If $\{L_{i+1}\} \cup G$ perform better than G, add L_{i+1} into G, set backtrace_count:=0; else backtrace_count++;
- 4) The iteration stops when backtrace_count reaches its maximum, output G.

The above algorithm is supposed to be a *radical* strategy. As a *conservative* alternative, in second step we could produce the child L_{i+1} from parents $\{L_i\} \cup G$. Note that in fact, we could formulate our problem as a standard evolutionary computation, while preserving the group size of G.

III. LEARNING BOUNDARY-BASED DISCRIMINANT KNN CLASSIFIER

The idea of learning locally linear distance metrics for kNN classifier is at least 15 years old(DANN [15]), where linear discriminant analysis is extended to local adaption of the nearest neighbor metric. However, we find few proposals along these lines in literature. How to further justisfy the application of DANN in extension to learn a boundary-based global metrics automatically(maybe in some iterative manner) is still unclear and nontrivial. Figure 1 depicts the motivation



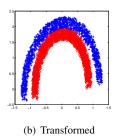


Fig. 1. Boundary-based local metric: notice that the critical boundary of two labeled data lies on the top region in left figure, and the right figure depicts the transformed data driven by critical boundary.

of local linear discriminant analysis, where classical LDA does not work.

In our framework, only the number k of nearest neighbors in kNN is user defined. We only make assumption that the training set reflects the sampling probability density and different sets of class are locally separated, which means in a neighbor, convex hull of points with label of one class has no other points with label of another. Hence the input instance in training set does not necessarily share the same label with its k-nearest neighbors, in other words, the different sets of classes are not necessarily widely separated. Furthermore due to our assumption, the training set should provide information near class boundaries, for the reason that our learning algorithm is supposed to enlarge the margin between different classes.

A. Neighbor estimation

We estimate the k-nearest neighbor distribution of each instance within the same class. For simplicity, we assume it as isotropic Gaussian distribution and obtain the neighbor radius by averaging the distances from its k-nearest neighbors within the same class. We denote the neighbor radius of instance \mathbf{x}_i as $\sigma_i^{(A)}$, which depends on the metric A.

B. Offspring model in closed-form convex optimization

With the estimation of k-nearest neighbor within the same class, we expect its neighbors have the same label. Otherwise, we would give a relatively higher penalty weight for neighbor instance with different label by Gaussian filter or Butter-worth filter,

$$p_i^{(A)}(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_i\|_A^2}{2(\sigma_i^{(A)})^2}\right),$$

$$p_i^{(A)}(\mathbf{x}) = 1/\left(1 + \left(\left\|\mathbf{x} - \mathbf{x}_i\right\|_A / \sigma_i^{(A)}\right)^4\right).$$

We expect to optimize the metric \boldsymbol{A} by diagnosing the non-linear objective

$$\sum_{(i,j) \in K} p_i^{(A)}(\mathbf{x}_j) \left\| \mathbf{x}_i^{(c)} - \mathbf{x}_j \right\|_A^2$$

with some volume preserving constraints, where template K is a subset of pairs(to be determined later) and $\mathbf{x}_i^{(c)}$ is a "center" of "within class" neighbor of instance i.

Let $A = L^T L$, then above objective can be written as

$$\operatorname{Tr}\left(LM_{K}L^{T}\right),$$

where

$$M_K = \sum_{(i,j) \in K} p_i^{(A)}(\mathbf{x}_j) (\mathbf{x}_i^{(c)} - \mathbf{x}_j) ((\mathbf{x}_i^{(c)})^T - \mathbf{x}_j^T),$$

here we set weight function $p_K^{(i)} = \sum_{(i,j) \in K} p_i^{(A)}(\mathbf{x}_j)$ and normalize M_K by $\overline{M}_K = M_K/p_K^{(i)}$.

Considering $D_i = \{(i,j), j \notin C(i)\}$ and $S_i = \{(i,j), j \in C(i)\}$, where C(i) is the set of instances with same label of \mathbf{x}_i . Let $N_i = D_i \cup S_i$. We derive another objective sclar function

$$\mathcal{E}(L) = \operatorname{Tr}\left(L\left(\sum_{i} w_{i} \left(\overline{M}_{D_{i}} - \overline{M}_{S_{i}}\right)\right) L^{T}\right),$$

where $w_i = p_{D_i}^{(i)}/p_{N_i}^{(i)}$. In our implementation, we select $\mathbf{x}_i^{(c)} = (\sum_{j \in S_i} p_i^{(A)}(\mathbf{x}_j)\mathbf{x}_j)/p_{S_i}^{(A)}$ (neighbor-based weighted sum of S_i) or $\mathbf{x}_i^{(c)} = \mathbf{x}_i$.

The optimization is then to maximize $\mathcal{E}(L)$ with a constraint

$$L\left(\sum_{i} w_{i} \overline{M}_{N_{i}}\right) L^{T} = I_{m}$$

where m is the projection dimension. The above optimization problem in fact do have closed form solution by solving a generalized eigenvalue problem (derived from KKT condition):

$$\left(\sum_{i} w_{i} \left(\overline{M}_{D_{i}} - \overline{M}_{S_{i}}\right)\right) y_{k} = \lambda_{k} \left(\sum_{i} w_{i} \overline{M}_{N_{i}}\right) y_{k},$$

where $k = 1, ..., m \le n$ and $\lambda_1 \ge ... \ge \lambda_m > 0$.

Later we will show the solution is effectively to reduce the amount of $\sum_{i} \log(w_i)$, which could be regarded as a generalization of objective in Neighborhood Component Analysis(NCA [6])

In fact, it is not easy to infer the intrinsic dimension m we should used to project. In empirical experiments, we set $L = [\lambda_1 y_1, \dots, \lambda_m y_m]^T$, hence λ_k which approximates zero will diminish the contribution of y_k .

C. Multiple metric

Instead of deriving multiple metric geometrically locally for data cluster, we use metric registration approach, which for each point in training set we assign it a link to the final metric sets. For a group of metrics $G = L_0, \ldots, L_s$, set $A_k = L_k^T L_k$, we derived the offspring process of multiple metrics by modifying

$$w_i^{(G)} = \min_k \{w_i^{(A_k)}\},$$

and the metric associated with single instance i is expected to be the $\operatorname{argmin}_k\{w_i^{(A_k)}\}$.

With metric association, the kNN classifier could be extended to multiple metrics intuitively. For a new instance in

test set, we count reference instances within the k-nearest neighbor in terms of each metric which have the same metric association, and then select the one which corresponds to the largest count as the test instance's metric association.

IV. IMPLEMENTATION

There are some key notes in implementation of our overall framework.

In the initial, we pre-compute the a large nearest neighbor Ω_i for each instance, which in effect assumes $M_{K_i} \approx M_{K_i \cap \Omega_i}$ and $p_{K_i} \approx p_{K_i \cap \Omega_i}$ for $K_i = D_i, S_i, N_i$, and all kNN of \mathbf{x}_i in any metric would fall into Ω_i . Hence our computation complexity is linear during evolution.

During each iteration, we prescribe an active label for each instance when $w_G \leq \theta$, for some threshold θ , where in experiment we set it equal to 0.1. Hence inner instances which does not contribute to the critical boundaries will be labeled inactive in the first few evolutionary steps, which in effect improve overall efficiency dramatically.¹

V. EXPERIMENTAL RESULTS

A. Efficiency

The main computations is kNN classification of validation set, pre-computation of large nearest neighbor Ω and matrix assembly in closed-form solution maximizing objective \mathcal{E} . The former two almost dominate 80% of the overall computations due to that our framework does not implemented in its most efficient manner in our experiments.

For data set in size smaller than 1000, our implementation works out within seconds, while BoostMetric and LMNN needs 1-2 minutes, and NCA need several minutes. For larger data set(1k-10k), our implementation still only spend no more than 5 minutes, while LMNN and BoostMetric averagely need half an hour or more and NCA is running out of time.

B. UCI dataset

We select several data sets from UCI Machine Learning Repository [16], and compare three of our approach(CFLML-1, closed-form metric learning without evolution; CFLML-3, evolution of at most 3 metrics; EM-CFLML, auto-evolution; CFLML-best is the best run of the three in each independent experiment.) in the (multiple metric) kNN classification performance with null(Euclidean distance), Principle Component Analysis(PCA), Linear Discriminant Analysis(LDA [3]), Neighborhood Component Analysis(NCA [6]), BoostMetric [11], Large Margin Nearest Neighbor(LMNN [7] [8]).²

For data set of relevant small size, we run experiment 10 times. Each entry in table I represents error means(standard variance) accordingly and N/A denotes running out of time/memory. In each experiment, we randomly divide data set into 80% for training and 20% for testing. If validation set

¹Details of CPU time for each method w.r.t dataset should have been provided in terms of chart/table in the final version report.

²The implementation of PCA, LDA and NCA is from Matlab Toolbox for Dimensionality Reduction(http://homepage.tudelft.nl/19j49/Matlab_Toolbox_for_Dimensionality_Reduction.html), and the code for BoostMetric and LMNN(version 2) is the author's implementation.

 $TABLE\ I$ UCI dataset error rate(%) of KNN classification w.r.t different metrics

Data Set	Euclidean	PCA	LDA	NCA	Boost-best	LMNN	CFLML-1	CFLML-3	EM-CFLML	CFLML-best
iris	2.33(2.25)	2.67(2.63)	2.00 (2.81)	2.67(3.06)	3.33(3.14)	2.00 (2.81)	3.00(2.46)	3.67(2.92)	3.00(3.31)	2.00 (1.72)
wine	29.19(7.30)	29.46(7.03)	1.08 (1.40)	14.32(8.26)	1.62(2.91)	5.41(4.23)	2.43(2.69)	2.43(2.97)	2.70(2.85)	1.62(2.28)
balance	16.51(1.44)	14.60(1.51)	8.02(1.85)	5.79(3.67)	8.17(1.63)	13.49(5.46)	6.35(2.21)	5.72(2.21)	5.79(2.08)	4.84 (2.03)
wdbc	7.74(1.71)	7.74(1.71)	4.43(0.86)	6.61(2.29)	4.35 (1.83)	8.35(2.02)	6.09(1.59)	5.48(1.69)	6.87(2.07)	5.30(1.45)
vehicle	32.87(2.66)	32.92(2.80)	23.04(3.01)	25.73(3.23)	18.95(2.12)	21.40(3.15)	19.24(2.00)	19.77(3.03)	19.88(2.67)	18.19 (2.03)
wine quality	5.95(0.75)	5.95(0.75)	0.50 (0.23)	N/A	1.07(0.30)	1.77(0.47)	1.75(1.08)	1.73(0.63)	1.55(0.57)	1.33(0.50)
spambase	19.65(1.23)	19.97(0.96)	9.33(0.99)	N/A	18.43(4.37)	10.72(2.70)	7.97(0.75)	8.45(0.62)	7.98(0.59)	7.74 (0.54)
letters	4.29	3.91	3.86	N/A	2.82	3.14	3.19	2.99	3.19	2.99
isolet	11.67	12.76	4.74	N/A	4.68	5.38	5.32	5.32	4.87	4.87

isolet are reduced to 100 dimension by PCA

is needed, we further cut 15% out of training set as validation set.

Noting that for LMNN and BoostMetric, we found k=1,3 achieves best in their classification performance. But for some data sets, a larger k performs better in null, LDA, and our approach. (For example, we set k=9 for wine data set.) In experiments, we select different k individually for each methods to achieve their potentially best performance.

VI. CONCLUSION AND FUTURE WORKS

In this paper, we have introduce a new framework to learn multiple closed-form local metrics(CFLML) for nearest neighbor classification. Given a labeled training set, we have shown how to derived a child metric from a group of parents metrics by solving a closed-form optimization problem. The child metric in some way is supposed and proven to be complementary to parent metrics in their performance of kNN classification. Our framework makes no parametric assumptions about distribution of data and scales naturally, but need to provide a neighbor size k as a trade-off between noise and boundary blurring. By adopting a simple search strategy, multiple metrics and training instances' association are then computed in our framework, and experimental results show that our framework challenges previous single Mahalanobis metric methods in its computational efficiency and classification performance.

In future works, we are interested to refine our closed-form formulation in statistically sound way, optimize the implementation in its efficiency, and design effective evolutionary algorithms.

REFERENCES

- T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. IT-13, no. 1, pp. 21–7, Jan. 1967.
- [2] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. J. Russell, "Distance metric learning with application to clustering with side-information," in *NIPS*, S. Becker, S. Thrun, and K. Obermayer, Eds. MIT Press, 2002, pp. 505–512.
- [3] G. Fisher, "A discriminant analysis of reporting errors in health interviews," *Applied Statistics*, vol. 11, no. 3, pp. 148–163, Nov. 1962.
- [4] S. C. H. Hoi, W. Liu, M. R. Lyu, and W. Y. Ma, "Learning distance metrics with contextual constraints for image retrieval," in CVPR, 2006, pp. II: 2072–2078.

- [5] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall, "Learning distance functions using equivalence relations," in *ICML*, August 21-24, 2003, Washington, DC, USA, T. Fawcett and N. Mishra, Eds. AAAI Press, 2003, pp. 11–18.
- [6] J. Goldberger, S. T. Roweis, G. E. Hinton, and R. Salakhutdinov, "Neighbourhood components analysis," in NIPS, 2004.
- [7] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," in NIPS, 2005.
- [8] K. Q. Weinberger and L. K. Saul, "Fast solvers and efficient implementations for distance metric learning," in *ICML*, *Helsinki*, *Finland*, *June 5-9*, 2008, ser. ACM International Conference Proceeding Series, W. W. Cohen, A. McCallum, and S. T. Roweis, Eds., vol. 307. ACM, 2008, pp. 1160–1167.
- [9] A. Globerson and S. Roweis, "Metric learning by collapsing classes," Advances in neural information processing systems, vol. 18, p. 451, 2006.
- [10] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *ICML, Corvallis, Oregon, USA, June 20-24, 2007*, ser. ACM International Conference Proceeding Series, Z. Ghahramani, Ed., vol. 227. ACM, 2007, pp. 209–216.
- [11] C. Shen, J. Kim, L. Wang, V. D. Hengel, and A. John, "Positive semidefinite metric learning with boosting," in NIPS, 2009, pp. 1651– 1660.
- [12] B. Alipanahi, M. Biggs, and A. Ghodsi, "Distance metric learning vs. fisher discriminant analysis," in *Proceedings of the 23rd national* conference on Artificial intelligence, 2008, pp. 598–603.
- [13] L. Yang, R. Jin, R. Sukthankar, and Y. Liu, "An efficient algorithm for local distance metric learning," in *Proceedings of the National Conference on Artificial Intelligence*, vol. 21, no. 1. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006, p. 543.
- [14] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [15] T. Hastie and R. Tibshirani, "Discriminant adaptive nearest neighbor classification," *Pattern Analysis and Machine Intelligence, IEEE Trans*actions on, vol. 18, no. 6, pp. 607–616, 1996.
- [16] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml



JIANBO YE received B.S. degree in Mathematics from University of Science and Technology of China, July 2011. His researches are focused on computer graphics, visualization and machine learning.