南京航空航天大学

论文集

（二〇〇九年） 第23期

信息科学与技术学院

（第4分册）

南京航空航天大学科技部编

二〇一〇年三月

信息科学与技术学院

043

信息科学与技术学院2009年发表论文明细表

序号	姓名	职称	单位	论文题目	刊物、会议名称	年、卷、期
1	张道强	教授	043	An Efficient Nonnegative Matrix Factorization Approach in Flexible Kernel Space	2009年 the 21th International Joint Conference on Artificial Intelligence 会议上交流	
2	孙丹 张道强	硕士生 教授	043 043	A New Discriminant Principal Component Analysis Method with Partial Supervision	Neural Processing Letters	2009年 第30卷第2期
3	朱凤梅 张道强	硕士生 教授	043 043	张量图像上的半监督降维算法	模式识别与人工智能	2009年 第22卷第4期
4	庄传志 张道强	硕士生 教授	043 043	多视角判别聚类算法	南京理工大学学报	2009年 第33卷增刊
5	卜德云 张道强	硕士生 教授	043 043	自适应谱聚类算法研究	上东大学学报	2009年 第35卷第5期
6	蔡维玲 陈松灿 张道强	博士生 教授 教授	043 043 043	A Simultaneous learning framework for clustering and classification	Pattern Recognition	2009年 第42卷第7期
7	薛晖 陈松灿 杨强	博士生 教授 教授	043 043 043	Discriminatively regularized least-squares classification	Pattern Recognition	2009年 第42卷第1期
8	薛晖 朱玉莲 陈松灿	博士生 讲师 教授	043 043 043	Local ridge regression for face recognition	Neurocomputing	2009年第72卷 第4-6期
9	王哲 陈松灿	博士生 教授	043 043	Multi-view kernel machine on single-view data	Neurocomputing	2009年第72卷 第10-12期
10	杨绪兵 陈松灿 潘志松	博士生 教授 博士生	043 043 043	Proximal support vector machine using local information	Neurocomputing	2009年第72卷 第1-3期
11	朱玉莲 刘俊 陈松灿	博士生 副教授 教授	043 043 043	Semi-random subspace method for face recognition	Image Vision Comput	2009年 第27卷第9期
12	王波 陈松灿	博士生 教授	043 043	Heterogeneous Cross Domain Ranking in Latent Space	2009年18th Conference in Information and Knowledge Management(CIKM) 会议上交流	
13	杨绪兵 潘志松 陈松灿	博士生 副教授 教授	043 043 043	半监督型广义特征值最接近支持向量机	模式识别与人工智能	2009年 第22卷第3期
14	陈斌 李斌 潘志松 陈松灿	博士生 教授 副教授 教授	043 043 043 043	流形嵌入的支持向量数据描述	模式识别与人工智能	2009年 第22卷第4期
15	阮松 陈松灿	硕士生 教授	043 043	采用多尺度多级组合分类器快速定位乳腺X片中的感兴趣区域	中国生物医学工程学报	2009年 第28卷第5期
16	管致锦 秦小麟	博士生 教授	043 043	Reversible Network Construct Based on Orthomorphics permutation	Journal of Information and Computation Science	2009年1月 第6卷第1期
17	蒋鹏 秦小麟	博士生 教授	043 043	基于视觉注意模型的自适应视频关键帧提取	中国图像图形学报	2009年8月 第14卷第8期
18	蒋鹏 秦小麟	博士生 教授	043 043	基于时空模型的快速视频显著区域检测	南京航空航天大学学报	2009年2月 第41卷第1期

19	蒋鹏 秦小麟	博士生 教授	043 043	Keyframe-based Video Summarization using Visual Attention Clue	IEEE Multimedia	2009年9月
20	戴华 秦小麟	博士生 教授	043 043	基于OCAR挖掘的数据库异常检测模型	通信学报	2009年9月 第30卷第9期
21	戴华 秦小麟	博士生 教授	043 043	一种基于事务模板的恶意事务检测方法	计算机科学与技术	2009年10月
22	刘亮 秦小麟	博士生 教授	043 043	基于环扇区的无线传感器网络K近邻查询处理算法	电子学报	2009年
23	吴浩 秦小麟	硕士生 教授	043 043	网格数据库连接查询自适应处理算法研究	2009年 中国计算机大会 会议上交流	
24	李博涵 秦小麟	博士后 教授	043 043	Research on Reverse Nearest Neighbor Queries Using Ranked Voronoi Diagtam	2009年 信息科学与工程国际会议（iCISE2009）在ORAL SESSIONS 部分进行了会议上交流	
25	胡彩平 秦小麟	讲师 教授	043 043	融合空间自相关的空间数据预测模型	吉林大学学报 （信息科学版）	2009年11月 第27卷第6期
26	刘宇雷 秦小麟	博士生 教授	043 043	流数据复杂聚类查询处理算法	南京航空航天大学学报	2009年12月 第41卷第6期
27	司海棠 秦小麟	硕士生 教授	043 043	基于Voronoik阶邻近的目标预警报警方法	计算机应用	2009年2月 第29卷第2期
28	张骏 秦小麟	博士生 教授	043 043	3维空间线与体对象间的拓扑关系完备性研究	中国图像图形学报	2009年3月 第14卷第3期
29	廉成洋 毛宇光 黄玉明	硕士生 副教授 硕士生	043 043 043	基于启发式规则的Web信息抽取技术研究	计算机技术与发展	2009年 19卷第8期
30	席风磊 毛宇光 廉成洋	硕士生 副教授 硕士生	043 043 043	Xquery中FLWOR式的查询重写研究	计算机技术与发展	2009年 第19卷第6期
31	张礼 刘学军	硕士生 副教授	043 043	An improved probabilistic model for finding differential gene expression	2009年 the 2nd International Conference on Biomedical Engineering and Informatics会议上交流	
32	谭文安	教授	043	A Methodology towards the implementation of performance management for virtual enterprise	2010年 The 13th Inter.Conf.on CSCW in Design 会议上交流	
33	周良	副教授	043	Relevant feedback in content-based engineering drawing retrieval	International Conference on Computer Science and Software Engineering, CSSE 2008	2009年v4
34	董珊珊 周良	硕士生 副教授	043 043	基于xUML的模型驱动架构研究	中国制造业信息化	2009年第17期
35	仲伟炜 周良	硕士生 副教授	043 043	基于改进Apriori算法的客户满意度评测研究及应用	舰船电子工程	2009年第2期
36	黄中文 周良	硕士生 副教授	043 043	基于手写签名的电子公文安全认证方案设计	中国制造业信息化	2009年第9期
37	张必欢 谢强 周良	硕士生 副教授 副教授	043 043 043	三维模型检索中手绘草图轮廓特征的应用研究	中国制造业信息化	2009年第1期
38	吴丽 周良	硕士生 副教授	043 043	数据挖掘方法在中医药领域的应用浅析	医学信息	2009年第5期
39	糜德吉 周良 丁秋林	硕士生 副教授 副教授	043 043 043	一种基于命名空间过滤的服务语义匹配算法	中国制造业信息化	2009年第12期

40	陈黄焱 郑洪源	硕士生 副教授	043 043	基于WebGIS高速公路应急联动系统设计	计算机技术与发展	2009年第8期
41	马治国 郑洪源 丁秋林	硕士生 副教授 教授	043 043 043	基于工作仓库的OLAM缓存替换算法	计算机应用	2009年第1期
42	孔祥瑞 郑洪源	硕士生 副教授	043 043	基于企业服务总线的业务集成方法	计算机工程	2009年 第35卷16期
43	陈树蜂 郑洪源	硕士生 副教授	043 043	面向对象软件的依赖性分析与回归测试	计算机应用	2009年第11期
44	赵永金 郑洪源 丁秋林	硕士生 副教授 教授	043 043 043	一种基于本体的语义相似度算法研究	计算机应用	2009年第11期
45	陈阳平 谢强 丁秋林	博士生 副教授 教授	043 043 043	基于AABB层次树的数字样机空间区域计算与搜索方法	南京航空航天大学学报	2009年第4期
46	窦丹丹 谢强 丁秋林	硕士生 副教授 教授	043 043 043	基于Ontology的故障诊断方法	计算机应用	2009年第2期
47	汪芳琴 谢强 丁秋林	硕士生 副教授 教授	043 043 043	基于REST的Web服务研究	中国制造业信息化	2009年第23期
48	张晓伟 谢强	硕士生 副教授	043 043	基于划分和孤立点检测的审计证据获取研究	计算机应用研究	2009年第7期
49	李广 谢强 丁秋林	硕士生 副教授 教授	043 043 043	基于遗传禁忌算法的Ontology划分	计算机工程	2009年第期
50	钱巨	讲师	043	Dependence Analysis for C Programs with Combinability of Dataflow Facts under Consideration	Wuhan University Journal of Natural Sciences	2009年 第14卷第4期
51	钱巨	讲师	043	Improving side-effect analysis with lazy access path resolving	2009年 9th IEEE International Working Conference on Source Code Analysis and Manipulation 会议上交流	
52	钱巨	讲师	043	Contribution-Based Call Stack Abstraction and Its Application in Pointer Analysis of AspectJ Programs	2009年 16th Asia-Pacific Software Engineering Conference (APSEC 2009) 会议上交流	
53	刘欣 章勇	硕士生 副教授	043 043	增量学习的TFIDF_NB协同训练分类算法	中国电子学会第十六届信息论学术年会论文集	
54	焦芬芬 章勇	硕士生 副教授	043 043	基于聚类分析的过滤算法在RSS信息服务中的研究	中国电子学会第十六届信息论学术年会论文集	
55	柏桂荣 章勇	硕士生 副教授	043 043	基于RSS的用户兴趣模型研究	中国通信学会学术会议文集	
56	杨丽 章勇	硕士生 副教授	043 043	P2P系统研究	通信市场	
57	周宾 章勇	硕士生 副教授	043 043	CAN控制器在多支点触发系统中的应用	中国通信学会学术会议文集	
58	周宾 章勇	硕士生 副教授	043 043	嵌入式WEB访问时内存丢失问题研究	单片机与嵌入式系统应用	2009年第10期
59	陆雨花 章勇	硕士生 副教授	043 043	基于距离无关的WSN节点定位技术研究	电脑知识与技术	2009年 第5卷第6期
60	潘志芳 章勇	硕士生 副教授	043 043	一种链式分簇的无线传感器网络路由协议	传感器与微系统	2009年 第28卷第6期

61	彭秀芬 章勇	硕士生 副教授	043 043	移动Ad hoc网络安全技术研究	电脑知识与技术	2009年 第5卷第8期
62	彭秀芬 章勇	硕士生 副教授	043 043	校园无线网络动态组织和覆盖的可行性研究	电脑知识与技术	2009年 第5卷第6期
63	杜国平 朱梧槚	博士生 教授	043 043	集合论-范逻辑悖论	北京航空航天大学学报	2009年 第35卷第3期
64	杜国平	博士生	043	The Completeness and Decidability of Intuitive Implication Logic System	2008年 The 4th International Conference on Natural Computation 会议上交流并收入论文集	
65	杜国平	博士生	043	Intuitive Implication Logic System	2009年 The 5th International Conference on Natural Computation (ICNC'09)会议上交流并收入论文集	
66	陈海燕 王建东 顾彬	讲师 教授 博士生	043 043 043	基于融合先验知识SVM的航班延误预警模型	南航学报	2009年 第41卷第2期
67	戴群 陈松灿	讲师 教授	043 043	一个基于自组织特征映射网络的混合神经网络结构	软件学报	2009年 第20卷第5期
68	戴群	讲师	048	算法设计与分析一本科课程教学研究	计算机教育	2009年第18期
69	高静 曹子宁	硕士生 教授	043 043	基于空间逻辑和计算树逻辑的模型检测	2009年中国高校通信类院系学术研讨会论文集 会议上交流	
70	许梅 曹子宁	硕士生 教授	043 043	基于谓词mu演算和空间逻辑的模型检测算法研究	2009年中国高校通信类院系学术研讨会论文集 会议上交流	
71	曹子宁	教授	043	Branching Bisimulations for Higher Order pi-Calculus	International Conference on Information Technology and Computer Science 会议上交流	
72	曹子宁	教授	043	Distributed Viewpoint Equivalences for Higher Order Processes	2009年 2009 WRI World Congress on Computer Science and Information Engineering 会议上交流	
73	曹子宁	教授	043	A True Concurrent Process Calculus and Its Conflict Bisimulation	2009年 10th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Paraller/Distributed Computing 会议上交流	
74	冯爱民 刘学军	副教授 副教授	043 043	嵌入数据结构信息的单类支持向量机及其线性规划算法	中山大学学报 （自然科学版）	2009年 第48卷第6期
75	冯爱民 刘学军	副教授 副教授	043 043	Structured Learning from Data for Novelty Detection by Linear Programming	2009年 International MultiConference of Engineers and Computer Scientists 会议上交流	
76	胡彩平	讲师	043	《数据库系统概论》课程的研究性教学探讨	时代教育	2009年第5期
77	谭晓阳 陈松灿 周志华 刘俊	教授 教授 教授 副教授	041 041 041 041	Face Recognition under Occlusions and Variant Expressions with Parital Similarity	IEEE Transactions on Information Forensics & Security	2009年 第4卷第2期

IJCAI'09

An Efficient Nonnegative Matrix Factorization Approach in Flexible Kernel Space *

Daoqiang Zhang[1] **Wanquan Liu**[2]
[1]Dept. of CSE, Nanjing University of Aeronautics & Astronautics, China
[2]Dept. of Computing, Curtin University of Technology, Australia
dqzhang@nuaa.edu.cn w.liu@curtin.edu.au

Abstract

In this paper, we propose a general formulation for kernel nonnegative matrix factorization with flexible kernels. Specifically, we propose the Gaussian nonnegative matrix factorization (GNMF) algorithm by using the Gaussian kernel in the framework. Different from a recently developed polynomial NMF (PNMF), GNMF finds basis vectors in the kernel-induced feature space and the computational cost is independent of input dimensions. Furthermore, we prove the convergence and nonnegativity of decomposition of our method. Extensive experiments compared with PNMF and other NMF algorithms on several face databases, validate the effectiveness of the proposed method.

1 Introduction

Nonnegative matrix factorization (NMF) is a recent linear method for finding low-dimensional representation of nonnegative high-dimensional data such as images and texts. It imposes the nonnegativity constraints in both its basis vectors (bases) and coefficients. Due to its part-based representation property [Lee and Seung, 1999], NMF and its variations have been applied to a variety of applications, such as image classification, face expression recognition, face and object recognition, document clustering, *etc* [Berry *et al.*, 2007].

Over the last decade, many variants on NMF have been proposed to improve original NMF from different perspectives. To our knowledge, most works focus on one or several of the following aspects: 1) enhancing the sparseness of representation [Li *et al.*, 2001][Hoyer, 2004][Pascual-Montano *et al.*, 2006]; 2) investigating alternative computational solutions [Berry *et al.*, 2007][Lin, 2007]; 3) introducing discriminative information to improve classification power [Zafeiriou *et al.*, 2006][Yang *et al.*, 2008]. For example, to enhance the sparseness, Li *et al.* [2001] and Hoyer [2004] imposed different extra constraints. As for alternative computational solutions, besides the well-known multiplicative update algorithms, Lin [2007] recently proposed the projected gradient methods for NMF based on bound-constrained optimization. At last, Zafeiriou *et al.* [2006] and Yang *et al.* [2008] both introduced discriminant information into NMF for better classification power.

On the other hand, NMF and its many variants are linear models, i.e. data are decomposed as a linear mixture of basis vectors. Recently, kernel methods [Shawe-Taylor and Cristianini, 2004] have been used in NMF to deal with nonlinear correlation in data. Buciu *et al.* [2008] proposed the polynomial nonnegative matrix factorization (PNMF) method, where the original data as well as the unknown basis vectors are first transformed by a nonlinear polynomial kernel mapping into a higher feature space and then a nonnegative decomposition is accomplished in the feature space. Although PNMF shows improved classification power over conventional NMF algorithms, there remain several problems unresolved yet. Firstly, only the polynomial kernel function can be used in PNMF to keep the nonnegativity constraint. Other kernel functions such as the well-known Gaussian kernel may not be adopted because of the negative solution resulting from the derivative associated from the Gaussian kernel. Secondly, although the decomposition is performed in feature space, PNMF still seeks basis vectors in the original input space and then transform them into feature space. It remains unknown how to find basis vectors directly in the feature space. Finally, at each iteration step of PNMF, the kernel matrices have to be recomputed, and thus a great deal of computational cost are required. In our previous work, we ever proposed performing NMF directly on kernel matrices, but rigorous derivation and analysis in theory was not given [Zhang *et al.*, 2006].

In this paper, we propose an alternative way for using kernel method in NMF. A general framework for kernel based NMF is presented which can efficiently use flexible kernel functions. Besides, unlike in PNMF where basis vectors are still found in original input space, our method directly seeks bases in transformed feature space, which can be further changed into a much easier kernel decomposition problem by using kernel functions. Furthermore, there is no need to repeatedly compute the kernel matrices in each iteration, and the computational cost is low. Algorithmic convergence and nonnegativity property are guaranteed by theoretical proof. Specifically, we use the Gaussian kernel in our framework and present the Gaussian nonnegative matrix factorization (GNMF) algorithm. The effectiveness of the proposed method is validated by extensive experiments on sev-

*This work is partially supported by NSFC (60875030), Doctoral Fund of MOE (200802870003) and Australia ARC Linkage grant.

eral face databases compared with PNMF and conventional NMF algorithms.

The rest of this paper is organized as follows. Section 2 reviews the standard NMF algorithm and the recently proposed PNMF algorithm. Then in Section 3, we present the flexible kernel based NMF framework and give the proposed GNMF algorithm in detail. Experimental results on several benchmark face databases are reported in Section 4. And finally, we conclude this paper and indicate some issues for future research in Section 5.

Notations: Throughout this paper, we use lowercase bold letters to denote vectors and uppercase bold letters to denote matrices, if not stated specially. The operator $\langle \cdot \rangle$ means the inner product, and $\| \cdot \|$ denotes the Frobenius norm. $\boldsymbol{A}^T$ denotes the transpose of a matrix $\boldsymbol{A}$, $\boldsymbol{A}^+$ indicates the Moore-Penrose pseudo-inverse of matrix $\boldsymbol{A}$, and $tr(\boldsymbol{A})$ means the trace operator of the corresponding matrix $\boldsymbol{A}$. The symbol $\boldsymbol{A}_{i\cdot}$ denotes the ith row vector of matrix $\boldsymbol{A}$, and $\boldsymbol{A}_{\cdot i}$ means the ith column vector of matrix $\boldsymbol{A}$. $X \geq 0$ represents the matrix is nonnegative.

2 NMF and PNMF

2.1 NMF

The key ingredient of NMF is the non-negativity constraints imposed on the two matrix factors. Assume that the observed data of the objects are represented as an $n \times m$ matrix $\boldsymbol{X}$, each column of which contains n non-negative attribute values in one of the m objects. In order to represent data or reduce the dimensionality, NMF finds two non-negative matrix factors $\boldsymbol{W}$ and $\boldsymbol{H}$ such that $\boldsymbol{X} \approx \boldsymbol{WH}$. In general, the standard NMF problem can be formally expressed as follows [Lee and Seung, 2001]:

Problem 1 *(The NMF problem) Given an $n \times m$ nonnegative matrix $\boldsymbol{X}$ and a positive integer $r < \min\{n, m\}$, find nonnegative matrices $\boldsymbol{W}$ and $\boldsymbol{H}$ to minimize the following objective function*

$$\begin{aligned} J_1(\boldsymbol{W}, \boldsymbol{H}) &= \min_{\boldsymbol{W},\boldsymbol{H}} \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{WH}\|^2 \\ s.t. &\quad \boldsymbol{W} \geq 0, \boldsymbol{H} \geq 0. \end{aligned} \tag{1}$$

In order to obtain $\boldsymbol{W}$ and $\boldsymbol{H}$, a multiplicative update rule is given in [Lee and Seung, 2001].

2.2 PNMF

The standard NMF is a linear model, and thus it only allows linear correlation. To handle the nonlinear correlation, the polynomial NMF (PNMF) algorithm was recently proposed. The main idea of PNMF is to first transform data into higher dimensional feature space by using a polynomial kernel-induced nonlinear mapping and then perform decomposition in that feature space. Let ϕ denote the nonlinear mapping corresponding to the polynomial kernel, i.e. $k(\boldsymbol{x}, \boldsymbol{z}) = \langle \boldsymbol{x}, \boldsymbol{z} \rangle^{\{d\}} = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{z}) \rangle$, then the PNMF problem can be formally expressed as follows [Buciu *et al.*, 2008]:

Problem 2 *(The PNMF problem) Given the nonnegative input data $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_m]$ and the corresponding transformed input data in polynomial feature space $\Phi(\boldsymbol{X}) = [\phi(\boldsymbol{x}_1), \phi(\boldsymbol{x}_2), ..., \phi(\boldsymbol{x}_m)]$, and a positive integer r, find nonnegative matrices $\boldsymbol{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_r]$ and $\boldsymbol{H}$ to minimize the following objective function*

$$\begin{aligned} J_2(\boldsymbol{W}, \boldsymbol{H}) &= \min_{\boldsymbol{W},\boldsymbol{H}} \frac{1}{2}\|\Phi(\boldsymbol{X}) - \boldsymbol{YH}\|^2 \\ s.t. &\quad \boldsymbol{W} \geq 0, \boldsymbol{H} \geq 0, \end{aligned} \tag{2}$$

where $\boldsymbol{Y} = [\phi(\boldsymbol{w}_1), \phi(\boldsymbol{w}_2), ..., \phi(\boldsymbol{w}_r)]$.

It is easy to see that if we expand the objective function in Eq. 2, the PNMF problem can be solved by invoking only the kernel function. In order to obtain $\boldsymbol{W}$ and $\boldsymbol{H}$, a multiplicative update rule is given as follows [Buciu *et al.*, 2008]:

$$\boldsymbol{H}_{a\mu} = \boldsymbol{H}_{a\mu} \frac{(\boldsymbol{K}_{wx})_{a\mu}}{(\boldsymbol{K}_{ww}\boldsymbol{H})_{a\mu}} \tag{3}$$

$$\boldsymbol{W}_{ia} = \boldsymbol{W}_{ia} \frac{(\boldsymbol{X}\boldsymbol{K}'_{xw})_{ia}}{(\boldsymbol{W}\boldsymbol{\Lambda}\boldsymbol{K}'_{ww})_{ia}} \tag{4}$$

where $(\boldsymbol{K}_{wx})_{a\mu} = k(\boldsymbol{w}_a, \boldsymbol{x}_\mu)$, $(\boldsymbol{K}_{ww})_{ab} = k(\boldsymbol{w}_a, \boldsymbol{w}_b)$ are kernel matrices of dimensions $r \times m$ and $r \times r$, respectively. $(\boldsymbol{K}'_{xw})_{ia} = k'(\boldsymbol{x}_i, \boldsymbol{w}_a)$, $(\boldsymbol{K}'_{ww})_{ab} = k'(\boldsymbol{w}_a, \boldsymbol{w}_b)$ are kernel matrices of dimensions $m \times r$ and $r \times r$ respectively, wherek' is the derivative of the polynomial kernel k, i.e. $k'(\boldsymbol{x}, \boldsymbol{z}) = d\langle \boldsymbol{x}, \boldsymbol{z} \rangle^{\{d-1\}}$. Λ is a diagonal matrix whose diagonal elements are $\lambda_{aa} = \sum_{j=1}^{m} \boldsymbol{H}_{aj}$.

It is noteworthy that although PNMF has used the kernel method to handle nonlinear correlations, it is restricted with the polynomialhkn kernel functions. That is because the iteration updating rule (Eq. 4) needs to compute the derivative of a kernel, while most kernel functions such as the Gaussian kernel may have negative derivatives and thus cannot remain the nonnegativity property in the decomposition. This motivates us to find alternative ways to allow more flexible kernel functions in NMF.

3 The Proposed Method

To overcome the limitations of PNMF, in this section, we propose an alternative kernel NMF framework with flexible kernels. In the following, we first give the new problem formulation, and then derive the iterative update rules and prove the convergence. Specifically, we use the Gaussian kernel in the framework and give the Gaussian NMF (GNMF) algorithm in detail at the end of this section.

3.1 Problem Formulation

Assume that the observed data of the objects are represented as an $n \times m$ matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_m]$. Let ϕ be an implicit nonlinear mapping from the original input space to a high-dimensional feature space, where the inner product is defined as a kernel $k(\boldsymbol{x}, \boldsymbol{y}) = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{y}) \rangle$ in the original input space. Denote $\Phi(\boldsymbol{X}) = [\phi(\boldsymbol{x}_1), \phi(\boldsymbol{x}_2), ..., \phi(\boldsymbol{x}_m)]$. Like in NMF, we want to find two non-negative matrix factors $\boldsymbol{W}$ and $\boldsymbol{H}$ such that $\Phi(\boldsymbol{X}) \approx \boldsymbol{WH}$. However, because the explicit form of ϕ is unknown and $\phi(\boldsymbol{x}_i)$ may lie in very high or even infinite dimensional space, it is unpractical to directly decompose $\Phi(\boldsymbol{X})$ in the feature space.

Fortunately, we can solve that problem by representing the basis vectors $\boldsymbol{w}_i$ with linear combinations of transformed

data $\phi(\boldsymbol{x}_1), \phi(\boldsymbol{x}_2), ..., \phi(\boldsymbol{x}_m)$, i.e. $\boldsymbol{w}_i = \sum_{j=1}^{m} \boldsymbol{A}_{ji}\phi(\boldsymbol{x}_j) = \Phi(\boldsymbol{X})\boldsymbol{A}_{\cdot i}, i = 1, ..., r$. Denote $\boldsymbol{W} = \Phi(\boldsymbol{X})\boldsymbol{A}$, we have

$$\frac{1}{2}\|\Phi(\boldsymbol{X}) - \boldsymbol{W}\boldsymbol{H}\|^2 = \frac{1}{2}\|\Phi(\boldsymbol{X}) - \Phi(\boldsymbol{X})\boldsymbol{A}\boldsymbol{H}\|^2$$
$$= \frac{1}{2}tr(\boldsymbol{K} - 2\boldsymbol{K}\boldsymbol{A}\boldsymbol{H} + \boldsymbol{H}^T\boldsymbol{A}^T\boldsymbol{K}\boldsymbol{A}\boldsymbol{H}) \quad (5)$$

where $\boldsymbol{K} = \Phi^T(\boldsymbol{X})\Phi(\boldsymbol{X})$ is the kernel matrix. Note that each column vector of $\boldsymbol{W}$ lies in the kernel-induced feature space, and thus we cannot constrain it explicitly. Instead, we approximately constrain $\boldsymbol{A}^T\boldsymbol{K}\boldsymbol{A} \geq 0$ due to $\boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{A}^T\Phi^T(\boldsymbol{X})\Phi(\boldsymbol{X})\boldsymbol{A} = \boldsymbol{A}^T\boldsymbol{K}\boldsymbol{A} \geq 0$. From Eq. 5, the flexible-kernel NMF problem can be expressed as follows:

Problem 3 *(The Flexible-Kernel NMF problem) Given the nonnegative input data $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_m]$ and the corresponding kernel matrix $\boldsymbol{K} = \Phi^T(\boldsymbol{X})\Phi(\boldsymbol{X})$, and a positive integer r, find $\boldsymbol{A}$ and nonnegative matrix $\boldsymbol{H}$ to minimize the following objective function*

$$J_3(\boldsymbol{A}, \boldsymbol{H}) = \min_{\boldsymbol{A},\boldsymbol{H}} \frac{1}{2}tr(\boldsymbol{K} - 2\boldsymbol{K}\boldsymbol{A}\boldsymbol{H} + \boldsymbol{H}^T\boldsymbol{A}^T\boldsymbol{K}\boldsymbol{A}\boldsymbol{H})$$
$$s.t. \quad \boldsymbol{A}^T\boldsymbol{K}\boldsymbol{A} \geq 0, \boldsymbol{H} \geq 0. \quad (6)$$

It is easy to see that in the flexible-kernel NMF problem, basis vectors are sought in the transformed feature space, which is apparently different from the PNMF problem. On the other hand, the objective function in Eq. 6 is biquadratic, and generally there is no closed-form solution for it. In the next subsection, we will present an alternately iterative procedure for computing the nonnegative solution.

3.2 Iterative Update Procedure

Before formally describing the derivations of the iterative update rule, we first introduce some preliminary concepts and lemmas which will be used later.

Definition 1 *(Auxiliary function) Function $G(\boldsymbol{A}, \boldsymbol{A}')$ is an auxiliary function for function $F(\boldsymbol{A})$ if the conditions*

$$G(\boldsymbol{A}, \boldsymbol{A}') \geq F(\boldsymbol{A}), G(\boldsymbol{A}, \boldsymbol{A}) = F(\boldsymbol{A}) \quad (7)$$

are satisfied.

Lemma 1 *[Lee and Seung, 2001] If G is an auxiliary function, then F is nonincreasing under the update*

$$\boldsymbol{A}^{t+1} = \arg\min_{\boldsymbol{A}} G(\boldsymbol{A}, \boldsymbol{A}^t), \quad (8)$$

where t denotes the t-th iteration.

Solution of $\boldsymbol{H}$ for given A

When $\boldsymbol{A}$ is fixed, the objective function in Eq. 6 with respect to the coefficient matrix $\boldsymbol{H} = [\boldsymbol{H}_{\cdot 1}, \boldsymbol{H}_{\cdot 2}, ..., \boldsymbol{H}_{\cdot m}]$ can be rewritten as

$$F(\boldsymbol{H}) = \frac{1}{2}tr(\boldsymbol{K} - 2\boldsymbol{K}\boldsymbol{A}\boldsymbol{H} + \boldsymbol{H}^T\boldsymbol{A}^T\boldsymbol{K}\boldsymbol{A}\boldsymbol{H})$$
$$= \frac{1}{2}tr(\boldsymbol{K}) - \sum_{i=1}^{m}\boldsymbol{K}_{i\cdot}\boldsymbol{A}\boldsymbol{H}_{\cdot i} + \frac{1}{2}\sum_{i=1}^{m}\boldsymbol{H}_{\cdot i}^T\boldsymbol{A}^T\boldsymbol{K}\boldsymbol{A}\boldsymbol{H}_{\cdot i} \quad (9)$$

From Eq. 9, it is easy to notice that different column vectors of $\boldsymbol{H}$ are independent to each other for optimization, and thus the objective function can be further simplified into column-wise form as

$$F(\boldsymbol{H}_{\cdot i}) = \frac{1}{2}tr(\boldsymbol{K}) - \boldsymbol{K}_{i\cdot}\boldsymbol{A}\boldsymbol{H}_{\cdot i} + \frac{1}{2}\boldsymbol{H}_{\cdot i}^T\boldsymbol{A}^T\boldsymbol{K}\boldsymbol{A}\boldsymbol{H}_{\cdot i} \quad (10)$$

Following [Lee and Seung, 2001], we can construct an auxiliary function of $F(\boldsymbol{H}_{\cdot i})$ in Eq. 10 as below.

Lemma 2 *If $\boldsymbol{L}(\boldsymbol{H}_{\cdot i}^t)$ is the diagonal matrix*

$$\boldsymbol{L}_{ab}(\boldsymbol{H}_{\cdot i}^t) = \delta_{ab}(\boldsymbol{A}^T\boldsymbol{K}\boldsymbol{A}\boldsymbol{H}_{\cdot i}^t)_a / \boldsymbol{H}_{ai}^t, \quad (11)$$

where δ_{ab} is the indicator function, then

$$G(\boldsymbol{H}_{\cdot i}, \boldsymbol{H}_{\cdot i}^t) = F(\boldsymbol{H}_{\cdot i}^t) + \nabla F(\boldsymbol{H}_{\cdot i}^t)(\boldsymbol{H}_{\cdot i} - \boldsymbol{H}_{\cdot i}^t)$$
$$+ \frac{1}{2}(\boldsymbol{H}_{\cdot i} - \boldsymbol{H}_{\cdot i}^t)^T\boldsymbol{L}(\boldsymbol{H}_{\cdot i}^t)(\boldsymbol{H}_{\cdot i} - \boldsymbol{H}_{\cdot i}^t) \quad (12)$$

is an auxiliary function of $F(\boldsymbol{H}_{\cdot i})$ in Eq. 10.

The proof for Lemma 2 is similar as that in [Lee and Seung, 2001] and we omit it due to space limit. Then, according to Lemma 1, $\boldsymbol{H}_{\cdot i}^{t+1}$ can be computed by minimizing $G(\boldsymbol{H}_{\cdot i}, \boldsymbol{H}_{\cdot i}^t)$.

By setting $\frac{\partial G(\boldsymbol{H}_{\cdot i}, \boldsymbol{H}_{\cdot i}^t)}{\partial \boldsymbol{H}_{\cdot i}} = 0$, we have

$$\boldsymbol{H}_{\cdot i}^{t+1} = \boldsymbol{H}_{\cdot i}^t - [\boldsymbol{L}(\boldsymbol{H}_{\cdot i}^t)]^{-1}\nabla F(\boldsymbol{H}_{\cdot i}^t) \quad (13)$$

From Eqs. 10 and 11, and after some algebra operations, we obtain the update rule for $\boldsymbol{H}_{ai}$ as

$$\boldsymbol{H}_{ai}^{t+1} = \frac{\boldsymbol{H}_{ai}^t(\boldsymbol{A}^T\boldsymbol{K})_{ai}}{(\boldsymbol{A}^T\boldsymbol{K}\boldsymbol{A}\boldsymbol{H}^t)_{ai}} \quad (14)$$

Solution of $\boldsymbol{A}$ for given $\boldsymbol{H}$

When $\boldsymbol{H}$ is fixed, we want to optimize $\boldsymbol{A}$ according to the objective function in Eq. 6. For that purpose, we introduce an auxiliary matrix $\boldsymbol{B} = \boldsymbol{K}^{\frac{1}{2}}\boldsymbol{A}$, where $\boldsymbol{K}$ is the kernel matrix. However, there may be a few negative components in matrix $\boldsymbol{K}^{\frac{1}{2}}$. To keep the nonnegativity property, in this paper we project those negative values to the nearest nonnegative value, i.e. 0, and obtain both symmetric and nonnegative matrix $\boldsymbol{K}^{\frac{1}{2}}$. In our experiments, we found that only very few components of $\boldsymbol{K}^{\frac{1}{2}}$ are of negative values and the projection method works very well in practice.

From $\boldsymbol{B} = \boldsymbol{K}^{\frac{1}{2}}\boldsymbol{A}$, we have $\boldsymbol{A} = (K^{\frac{1}{2}})^{-1}\boldsymbol{B}$, then the objective function in Eq. 6 with with respect to the matrix $\boldsymbol{B} = [\boldsymbol{B}_{1\cdot}^T, \boldsymbol{B}_{2\cdot}^T, ..., \boldsymbol{B}_{m\cdot}^T]^T$ can be rewritten as

$$F(\boldsymbol{B}) = \frac{1}{2}tr(\boldsymbol{K} - 2\boldsymbol{K}^{\frac{1}{2}}\boldsymbol{B}\boldsymbol{H} + \boldsymbol{H}^T\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{H})$$
$$= \frac{1}{2}tr(\boldsymbol{K}) - \sum_{i=1}^{m}\boldsymbol{B}_{i\cdot}\boldsymbol{H}\boldsymbol{K}_{\cdot i}^{\frac{1}{2}} + \frac{1}{2}\sum_{i=1}^{m}\boldsymbol{B}_{i\cdot}\boldsymbol{H}\boldsymbol{H}^T\boldsymbol{B}_{i\cdot}^T \quad (15)$$

From Eq. 15, it is obvious that different row vectors of $\boldsymbol{B}$ are independent to each other for optimization, and thus

the objective function can be further simplified into row-wise form as

$$F(\boldsymbol{B}_{i\cdot}) = \frac{1}{2}tr(\boldsymbol{K}) - \boldsymbol{B}_{i\cdot}\boldsymbol{H}\boldsymbol{K}_{\cdot i}^{\frac{1}{2}} + \frac{1}{2}\boldsymbol{B}_{i\cdot}\boldsymbol{H}\boldsymbol{H}^T\boldsymbol{B}_{i\cdot}^T \quad (16)$$

Similarly, we can construct an auxiliary function of $F(\boldsymbol{B}_{i\cdot})$ in Eq. 16 as below.

Lemma 3 *If $\boldsymbol{L}(\boldsymbol{B}_{i\cdot}^t)$ is the diagonal matrix*

$$\boldsymbol{L}_{ab}(\boldsymbol{B}_{i\cdot}^t) = \delta_{ab}(\boldsymbol{B}_{i\cdot}^T\boldsymbol{H}\boldsymbol{H}^T)_a/\boldsymbol{B}_{ia}^t, \quad (17)$$

where δ_{ab} is the indicator function, then

$$G(\boldsymbol{B}_{i\cdot}, \boldsymbol{B}_{i\cdot}^t) = F(\boldsymbol{B}_{i\cdot}^t) + \nabla F(\boldsymbol{B}_{i\cdot}^t)(\boldsymbol{B}_{i\cdot} - \boldsymbol{B}_{i\cdot}^t) + \frac{1}{2}(\boldsymbol{B}_{i\cdot} - \boldsymbol{B}_{i\cdot}^t)^T\boldsymbol{L}(\boldsymbol{B}_{i\cdot}^t)(\boldsymbol{B}_{i\cdot} - \boldsymbol{B}_{i\cdot}^t) \quad (18)$$

is an auxiliary function of $F(\boldsymbol{B}_{i\cdot})$ in Eq. 16.

Also, it is easy to prove Lemma 3 following [Lee and Seung, 2001]. Similarly, according to Lemma 1, $\boldsymbol{B}_{i\cdot}^{t+1}$ can be computed by minimizing $G(\boldsymbol{B}_{i\cdot}, \boldsymbol{B}_{i\cdot}^t)$.

By setting $\frac{\partial G(\boldsymbol{B}_{i\cdot}, \boldsymbol{B}_{i\cdot}^t)}{\partial \boldsymbol{B}_{i\cdot}} = 0$, we have

$$\boldsymbol{B}_{i\cdot}^{t+1} = \boldsymbol{B}_{i\cdot}^t - [\boldsymbol{L}(\boldsymbol{B}_{i\cdot}^t)]^{-1}\nabla F(\boldsymbol{B}_{i\cdot}^t) \quad (19)$$

From Eq. 16 and Eq. 17, and after some algebra operations, we obtain the update rule for $\boldsymbol{B}_{ia}$ as

$$\boldsymbol{B}_{ia}^{t+1} = \frac{\boldsymbol{B}_{ia}^t(\boldsymbol{K}^{\frac{1}{2}}\boldsymbol{H}^T)_{ia}}{(\boldsymbol{B}^T\boldsymbol{H}\boldsymbol{H}^T)_{ia}} \quad (20)$$

It is obvious that $\boldsymbol{B}^{t+1}$ is nonnegative if the matrices $\boldsymbol{H}$ and $\boldsymbol{B}^t$ are nonnegative. After $\boldsymbol{B}$ is obtained, we update the matrix $\boldsymbol{A}$ as

$$\boldsymbol{A} = (K^{\frac{1}{2}})^{-1}\boldsymbol{B} \quad (21)$$

Equations 20, 21 and 14 constitute the iterative update procedure, which optimizes the matrices $\boldsymbol{H}$ and $\boldsymbol{A}$ alternatively. In the next subsection, we will prove the iterative update procedure can converge to a local optimum.

3.3 Convergence Proof

In this section, we prove the convergence of the iterative update procedure proposed in last subsection.

The iterative update procedure between $\boldsymbol{H}$ and $\boldsymbol{A}$ can be further transformed the iterative update between $\boldsymbol{H}$ and $\boldsymbol{B}$. Substituting Eq. 21 into Eq. 14, we have

$$\boldsymbol{H}_{ai}^{t+1} = \frac{\boldsymbol{H}_{ai}^t(\boldsymbol{B}^T\boldsymbol{K}^{\frac{1}{2}})_{ai}}{(\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{H}^t)_{ai}} \quad (22)$$

Now the iterative update procedure consist of Eqs. 22 and 20. From Eq. 22, the updated matrix $\boldsymbol{H}^{t+1}$ is still nonnegative if the matrices $\boldsymbol{B}$ and $\boldsymbol{H}^t$ are nonnegative.

Theorem 1 *The alternative iterative update procedure*

$$\boldsymbol{H}_{ai}^{t+1} = \frac{\boldsymbol{H}_{ai}^t(\boldsymbol{B}^T\boldsymbol{K}^{\frac{1}{2}})_{ai}}{(\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{H}^t)_{ai}}, \boldsymbol{B}_{ia}^{t+1} = \frac{\boldsymbol{B}_{ia}^t(\boldsymbol{K}^{\frac{1}{2}}\boldsymbol{H}^T)_{ia}}{(\boldsymbol{B}^T\boldsymbol{H}\boldsymbol{H}^T)_{ia}}$$

converges to a local optimum.

Proof. Following [Lee and Seung, 2001] and [Yang *et al.*, 2008], we define

$$F(\boldsymbol{B}, \boldsymbol{H}) = \frac{1}{2}tr(\boldsymbol{K} - 2\boldsymbol{K}^{\frac{1}{2}}\boldsymbol{B}\boldsymbol{H} + \boldsymbol{H}^T\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{H})$$

From the update rule for $\boldsymbol{B}$, we have

$$F(\boldsymbol{B}^{t+1}, \boldsymbol{H}^t) \leq G(\boldsymbol{B}^{t+1}, \boldsymbol{B}^t) \leq F(\boldsymbol{B}^t, \boldsymbol{H}^t)$$

Similarly, from the update rule for $\boldsymbol{H}$, we have

$$F(\boldsymbol{B}^{t+1}, \boldsymbol{H}^{t+1}) \leq G(\boldsymbol{H}^{t+1}, \boldsymbol{H}^t) \leq F(\boldsymbol{B}^{t+1}, \boldsymbol{H}^t)$$

So $F(\boldsymbol{B}^{t+1}, \boldsymbol{H}^{t+1}) \leq F(\boldsymbol{B}^t, \boldsymbol{H}^t)$.

On the other hand, from Eq. 5, it is easy to notice that $F(\boldsymbol{B}^t, \boldsymbol{H}^t) \geq 0$. Then, $F(\boldsymbol{B}^t, \boldsymbol{H}^t)$ decreases monotonically and has lower bound, and hence $F(\boldsymbol{B}^t, \boldsymbol{H}^t)$ will converge to a local optimum. □

3.4 The GNMF Algorithm

In this section, we summarize the above analysis by presenting a specific Gaussian NMF (GNMF) algorithm by using the Gaussian kernel in the flexible-kernel NMF framework. However, it is noteworthy that our method is not confined to the Gaussian kernel, and any kind of kernels can be used. Algorithm 1 lists the GNMF algorithm in detail.

Algorithm 1: The GNMF algorithm

Input:

Kernel matrix $\{\boldsymbol{K}_{ij} = \exp(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2})\}_{i,j=1}^m$
A positive integer $r < m$
A small threshold $\varepsilon > 0$.

Initialize:

Perform SVD decomposition $\boldsymbol{K} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^T$
Compute $\boldsymbol{K}^{\frac{1}{2}} = \max(\boldsymbol{U}\boldsymbol{S}^{\frac{1}{2}}\boldsymbol{U}^T, 0)$
Generate initial nonnegative matrices $\boldsymbol{B}^0$ and $\boldsymbol{H}^0$ with dimensions $m \times r$ and $r \times m$ respectively.

For $t = 1, ..., t_{max}$

1. For given $\boldsymbol{H} = \boldsymbol{H}^t$, update the matrix $\boldsymbol{B}$ as
$$\boldsymbol{B}_{ia}^{t+1} = \boldsymbol{B}_{ia}^t\frac{(\boldsymbol{K}^{\frac{1}{2}}\boldsymbol{H}^T)_{ia}}{(\boldsymbol{B}^T\boldsymbol{H}\boldsymbol{H}^T)_{ia}}$$
2. For given $\boldsymbol{B} = \boldsymbol{B}^t$, update the matrix $\boldsymbol{H}$ as
$$\boldsymbol{H}_{ai}^{t+1} = \boldsymbol{H}_{ai}^t\frac{(\boldsymbol{B}^T\boldsymbol{K}^{\frac{1}{2}})_{ai}}{(\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{H}^t)_{ai}}$$
3. If $\frac{\|\boldsymbol{B}^{t+1} - \boldsymbol{B}^t\|}{\sqrt{mr}} < \varepsilon$ and $\frac{\|\boldsymbol{H}^{t+1} - \boldsymbol{H}^t\|}{\sqrt{mr}} < \varepsilon$, then break.

Output:

$\boldsymbol{A} = (\boldsymbol{K}^{\frac{1}{2}})^{-1}\boldsymbol{B}^t$ and $\boldsymbol{H} = \boldsymbol{H}^t$.

4 Experiments

In this section, we test the performance of the proposed flexible-kernel NMF method. We first compare the GNMF algorithm with standard NMF, Localized NMF (LNMF) and PNMF. Also, we replace the Gaussian kernel in our GNMF with polynomial kernel (pKNMF) and compare its performance. All the NMF algorithms use the same stopping condition (see Step 3 in Algorithm 1) and ε is set to 10^{-4}, and the maximum iteration steps t_{max} is set to 500 in all experiments. For completeness, we also report results of kernel principal component analysis with both the Gaussian kernel (gKPCA) and polynomial kernel (pKPCA).

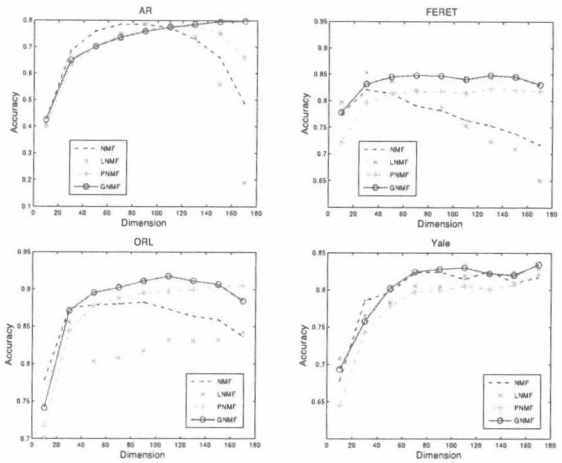

Figure 1: Classification accuracies (%) vs. different number of dimensions on AR-16x12, FERET-16x16, ORL-16x16 and Yale-16x16 databases.

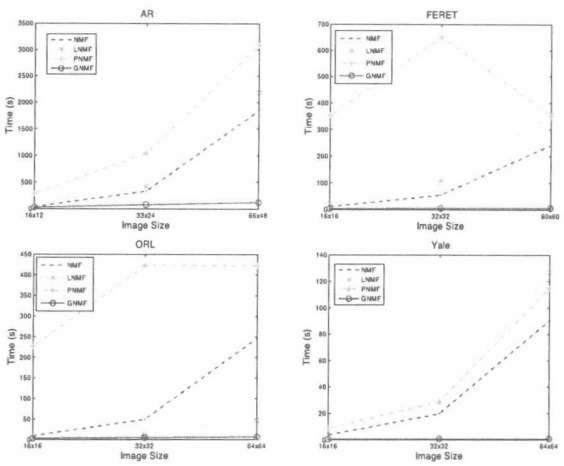

Figure 2: Running time (second) of the four algorithms under different image sizes.

4.1 Data Sets and Experimental Config

In our experiments, we use 4 benchmark face databases, i.e. AR, FERET, ORL and Yale. The AR database contains 1400 frontal facial images from 100 persons, each of which has 14 images at 2 different stages. The FERET database used in our experiments contains 200 persons, each with 2 images. The ORL database consists of 40 persons, each with 10 images. The Yale database contains 165 images from 11 persons. For each database, we resize images in 3 different scales, i.e. 66x48, 33x24 and 16x12 for AR, 60x60, 32x32 and 16x16 for FERET, 64x64, 32x32 and 16x16 for both ORL and Yale. So, there are totally 12 databases for experiments.

We evaluate performances of different algorithms using recognition accuracy. For each database, the first half of the images from each person are used for training and and the rest for testing. The Nearest Neighborhood classifier is adopted for classification after dimensionality reduction, where the number of reduced dimensions is set as $mn/(m+n)$, if without extra explanations. For PNMF, pKNMF and GNMF as well as pKPCA and gKPCA, cross-validation is used for selecting the kernel parameters d and σ respectively. For pKNMF and GNMF, features of a test image $\boldsymbol{x}_{te}$ are extracted as $(\Phi(\boldsymbol{X})\boldsymbol{A})^{+}\phi(\boldsymbol{x}_{te}) \approx \boldsymbol{A}^{+}(\Phi(\boldsymbol{X}))^{+}(\Phi^{T}(\boldsymbol{X}))^{+}\Phi^{T}(\boldsymbol{X})\phi(\boldsymbol{x}_{te}) = \boldsymbol{A}^{+}\boldsymbol{K}^{-1}\boldsymbol{K}_{te}$, where $\boldsymbol{K}_{te} = \Phi^{T}(\boldsymbol{X})\phi(\boldsymbol{x}_{te})$. All experiments are carried out on a PC with 2.7GHz CPU and 1GB RAM.

4.2 Experimental Results

We first compare GNMF with NMF, LNMF, PNMF, and Table 1 gives the classification accuracies of under fixed dimensions ($r = mn/(m+n)$) on the 12 databases. It can be seen from Table 1 that GNMF outperforms the other three algorithms in most cases and is consistently superior to PNMF. Table 1 also indicates that in most cases (except on AR) the four algorithms achieve better performances on small image size than large ones. Furthermore, Fig. 1 gives the classification accuracies of the four algorithms when different number of dimensions are used. We can see from Fig. 1 that GNMF outperforms other algorithms in most cases and is more robust to variations on dimensions.

Table 1: Classification accuracies (%) of NMF, LNMF, PNMF and GNMF on the 12 databases.

Data sets	NMF	LNMF	PNMF	GNMF
AR-16x12	67.16	60.29	75.46	**79.26**
AR-33x24	79.51	**85.69**	82.51	84.67
AR-66x48	77.69	**87.73**	80.31	81.17
FERET-16x16	76.7	75.2	82.6	**84.95**
FERET-32x32	72.55	73.6	80.4	**83.25**
FERET-60x60	65.65	73.5	79.65	**82.5**
ORL-16x16	88.25	82.75	90.65	**91.7**
ORL-32x32	86.2	81.45	87.75	**89.15**
ORL-64x64	81.7	80.3	83.75	**85.25**
Yale-16x16	82.44	82.44	81.22	**83.11**
Yale-32x32	80.78	81.78	81.78	**83.0**
Yale-64x64	77.89	81.89	80.67	**82.56**
Average	78.04	78.89	82.23	**84.21**

We also investigate the computational costs of four algorithms. It is easy to derive that the computational complexity for the iterative procedure of GNMF is $O(m^2rt)$, where m is the data size, r is the reduced dimensions and t is the iteration numbers. In comparison, the complexities of NMF and PNMF are $O(nmrt)$ and $O(nmrdt)$, where n is data dimensions and d is the order of polynomial kernel. Figure 2 plots the curves of running time vs. different image sizes for the four algorithms. As we expected, the curves of GNMF is nearly horizontal on all databases because its computational complexity is not dependent on the image size, i.e. n. Figure 2 shows that GNMF is much efficient than the other algorithms, especially for high-dimensional cases.

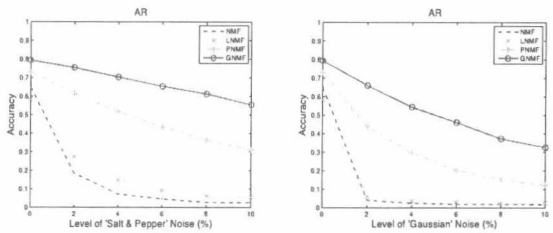

Figure 3: Classification accuracies (%) vs. different levels of noises on AR-16x12.

On the other hand, due to the use of kernel functions both PNMF and GNMF can deal with nonlinear correlations between basis vectors and thus are potentially more robust to image noises than their linear competitors. We carry out experiments on AR database when the test images are corrupted by different levels of 'Salt & Pepper' and 'Gaussian' noises respectively. Figure 3 gives the classification accuracies of four algorithms when test images are corrupted by different levels of noises. Figure 3 validate that nonlinear methods PNMF and GNMF are more advantageous for enhancing robustness to image noises than both NMF and LNMF, and GNMF consistently outperforms PNMF in both cases.

Finally, we make comparisons between kernel NMF (including pKNMF and GNMF) and kernel PCA (including pKPCA and gKPCA), and the results are given in Table 2. Table 2 indicates that pKNMF and GNMF achieve better averaged accuracies than pKPCA and gKPCA respectively across 12 databases, which validates the usefulness of kernel NMF. Furthermore, contrasting Table 2 with Table 1, it can be seen that our pKNMF outperforms PNMF in most cases and achieves better averaged accuracy.

5 Conclusion

In this paper, we proposed a general flexible-kernel based framework for nonnegative matrix factorization. We derived an alternative iteration update procedure and proved its convergence. Specifically, we proposed the Gaussian NMF (GNMF) algorithm with the Gaussian kernel and evaluated its performances on several face databases. One extra advantage of our method is that its computational complexity is independent on data dimensions and thus is potential for high-dimensional data decomposition. Moreover, GNMF can be used for negative data decomposition due to the Gaussian kernel transform and we will investigate that issue in future. Another future work is exploiting supervision information in GNMF to further enhance the discriminant power.

Acknowledgments We thank the the anonymous reviewers for their helpful comments and suggestions.

Table 2: Classification accuracies (%) of pKPCA, gKPCA, pKNMF and GNMF on the 12 databases.

Data sets	pKPCA	gKPCA	pKNMF	GNMF
AR-16x12	68.57	68.14	**79.77**	79.26
AR-33x24	74.29	74.14	**86.26**	84.67
AR-66x48	78.0	77.86	**83.57**	81.17
FERET-16x16	**85.5**	**85.5**	82.9	84.95
FERET-32x32	**85.0**	84.0	77.8	83.25
FERET-60x60	**84.5**	84.0	77.1	82.5
ORL-16x16	87.5	87.0	91.4	**91.7**
ORL-32x32	88.5	88.5	87.8	**89.15**
ORL-64x64	**87.5**	**87.5**	84.7	85.25
Yale-16x16	81.11	**85.56**	82.67	83.11
Yale-32x32	81.11	**84.44**	83.11	83.0
Yale-64x64	81.11	**84.44**	82.22	82.56
Average	81.89	82.59	83.26	**84.21**

References

[Berry *et al.*, 2007] M. Berry, M. Browne, A. Langville, V. Pauca, and R. Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics & Data Analysis*, 52:155–173, 2007.

[Buciu *et al.*, 2008] I. Buciu, N. Nikolaidis, and I. Pitas. Nonnegative matrix factorization in polynomial feature space. *IEEE Trans. on NN*, 19(6):1090–695, 2008.

[Hoyer, 2004] P.O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004.

[Lee and Seung, 1999] D.D. Lee and H.S. Seung. Learning the parts of objects by nonnegative matrix factorization. *Nature*, 401:788–791, 1999.

[Lee and Seung, 2001] D.D. Lee and H.S. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, volume 13, pages 629–634, 2001.

[Li *et al.*, 2001] S. Li, X. Hou, H. Zhang, and Q. Cheng. Learning spatially localized, parts-based representation. In *CVPR*, pages 207–212, 2001.

[Lin, 2007] C.J. Lin. Projected gradient methods for non-negative matrix factorization. *Neural Computation*, 19(10):2756–2779, 2007.

[Pascual-Montano *et al.*, 2006] A. Pascual-Montano, J.M. Carazo, K. Kochi, D. Lehmann, and R.D. Pascual-Marqui. Non-smooth non-negative matrix factorization (nsnmf). *IEEE Trans. on PAMI*, 28(3):403–415, 2006.

[Shawe-Taylor and Cristianini, 2004] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[Yang *et al.*, 2008] J. Yang, S. Yan, Y. Fu, X. Li, and T. Huang. Non-negative graph embedding. In *CVPR*, pages 1–8, 2008.

[Zafeiriou *et al.*, 2006] S. Zafeiriou, A. Tefas, I. Buciu, and I. Pitas. Exploiting discriminant information in nonnegative matrix factorization with application to frontal face verification. *IEEE Trans. on NN*, 17(3):683–695, 2006.

[Zhang *et al.*, 2006] D. Zhang, Z.H. Zhou, and S. Chen. Non-negative matrix factorization on kernels. In *PRICAI*, pages 404–412, 2006.

Neural Processing Letters 2009. 30(2)

Neural Process Lett
DOI 10.1007/s11063-009-9112-6

A New Discriminant Principal Component Analysis Method with Partial Supervision

Dan Sun · Daoqiang Zhang

Abstract Principal component analysis (PCA) is one of the most widely used unsupervised dimensionality reduction methods in pattern recognition. It preserves the global covariance structure of data when labels of data are not available. However, in many practical applications, besides the large amount of unlabeled data, it is also possible to obtain partial supervision such as a few labeled data and pairwise constraints, which contain much more valuable information for discrimination than unlabeled data. Unfortunately, PCA cannot utilize that useful discriminant information effectively. On the other hand, traditional supervised dimensionality reduction methods such as linear discriminant analysis perform on only labeled data. When labeled data are insufficient, their performances will deteriorate. In this paper, we propose a novel discriminant PCA (DPCA) model to boost the discriminant power of PCA when both unlabeled and labeled data as well as pairwise constraints are available. The derived DPCA algorithm is efficient and has a closed form solution. Experimental results on several UCI and face data sets show that DPCA is superior to several established dimensionality reduction methods.

Keywords Principal component analysis (PCA) · Discriminant PCA · Dimensionality reduction · Semi-supervised dimensionality reduction · Partial supervision

1 Introduction

With the rapid accumulation of high-dimensional data such as digital images, web documents and gene expression microarrays, dimensionality reduction has been a fundamental tool for many pattern recognition tasks. According to whether supervised information is available

D. Sun · D. Zhang (✉)
Department of Computer Science and Engineering, Nanjing University of Aeronautics and Astronautics, 210016 Nanjing, China
e-mail: dqzhang@nuaa.edu.cn

D. Sun
e-mail: dansun@nuaa.edu.cn

Published online: 10 July 2009

or not, existing dimensionality reduction methods can be roughly categorized into supervised ones and unsupervised ones. Linear discriminant analysis (LDA) [1] and principal component analysis (PCA) [2] may be the most well-known supervised and unsupervised dimensionality reduction methods respectively. The former extracts the optimal discriminant vectors when class labels are available, while the latter seeks projective vectors to preserve the global covariance structure when class labels are not available. In this paper, we consider the following interesting problem, i.e. when both labeled and unlabeled data are available, how should we perform dimensionality reduction? That problem arises naturally in many practical pattern recognition applications, where unlabeled training data are readily available but labeled ones are fairly expensive to obtain [3,4]. That is, we are often confronted with problems with large amount of unlabeled data but only a few labeled data. Typically, those labeled data contain much more valuable information for discrimination than unlabeled data.

Unfortunately, neither traditional unsupervised dimensionality reduction methods such as PCA nor supervised dimensionality reduction methods such as LDA can well deal with the above dimensionality reduction problems. On one hand, PCA is unsupervised, and it can not use the useful discriminant information in those labeled data. On the other hand, LDA performs on only labeled data. When labeled data are sufficient enough, LDA will nearly always outperform PCA. In contrast, when the number of labeled data per class is so small that labeled data can not reflect the underlying distribution, the generalization performances of LDA on unseen samples will not be guaranteed and PCA might outperform LDA. To overcome the disadvantages of both PCA and LDA, a natural idea is to simultaneously use both unlabeled data and discriminant information in labeled data for dimensionality reduction. More specifically, we can either introduce unlabeled data into LDA, or introduce discriminant information in labeled data into PCA. In this paper, we focus on the latter case.

In this paper, we propose the discriminant PCA model (DPCA), which exploits both labeled and unlabeled data for dimensionality reduction. DPCA inherits from PCA the characteristic of structure preserving on unlabeled data, and has the new discriminant power by using the discriminant information in labeled data. The derived DPCA algorithm is efficient and has a closed form solution. Moreover, DPCA algorithm has the capability to use external knowledge provided by the user, such as pairwise constraints which specify whether a pair of instances belong to the same class (*must-link* constraint) or different classes (*cannot-link* constraint) [5,6]. Experimental results on several UCI and face data sets show that DPCA outperforms several established dimensionality reduction methods. The rest of this paper is organized as follows: Sect. 2 presents some related work in semi-supervised dimensionality reduction. The detailed DPCA algorithm is introduced in Sect. 3. Section 4 reports on the experimental results. Finally, Sect. 5 concludes this paper with some future work.

2 Related Works

In fact, the idea of using both labeled and unlabeled data for learning is not novel in machine learning. There has appeared a new branch in machine learning called semi-supervised learning whose main concern is to learn from a combination of both labeled and unlabeled data [3–5,7]. Because of its success in many practical applications such as text categorization [3], semi-supervised learning has received much attention in recent years. Current researches on semi-supervised learning could be roughly categorized into three classes, i.e. semi-supervised classification [3], semi-supervised regression [4] and semi-supervised clustering [5]. Research advances of semi-supervised learning can be found in an excellent recent survey [7].

Recently, some research works which utilize both labeled and unlabeled data for semi-supervised dimensionality reduction have appeared. For example, Yu et al. [8] proposed a supervised probabilistic PCA model and a semi-supervised probabilistic PCA model, and the latter can incorporate both labeled and unlabeled data for dimensionality reduction. However, their method is based on probabilistic PCA which is a generative model. Also, their algorithm needs iteration and has no closed form solution. Lu et al. [9] proposed a novel hybrid dimension reduction scheme to merge LDA and PCA in a unified framework. In addition, many subspace learning algorithms such as spectral regression discriminant analysis method [10,11] and semi-supervised discriminant analysis method [12] have been proposed. Specifically, Cai et al. [12] proposed the semi-supervised discriminant analysis method called SDA which utilized local neighborhood information of labeled data for dimensionality reduction. However, the number of neighborhood in SDA is still hard to set. Besides SDA, $\text{SSDA}_{\text{CCCP}}$ is a diverse semi-supervised discriminant analysis algorithm proposed by Zhang et al. [13]. It uses the constrained concave–convex procedure (cccp) to maximize an optimality criterion of LDA which leads to estimation of the class labels for the unlabeled data. In one of our recent work [14], we proposed the semi-supervised dimensionality reduction model which uses the pairwise constraints together with unlabeled data for dimensionality reduction. However, in that paper, we didn't discuss using both labeled and unlabeled data for dimensionality reduction.

3 Discriminant Principal Component Analysis

PCA only preserves the global covariance structure of unlabeled data which can not utilize discriminant information in labeled data. In this section, we present the DPCA algorithm which introduces a new discriminant criterion into the original objective function of PCA.

3.1 The DPCA Algorithm

Given a set of n D-dimensional data samples $X = \{x_1, x_2, ..., x_n\}$, suppose that there exist l labeled data $L = \left\{x_{i_1}, x_{i_2}, ..., x_{i_l}\right\} \subseteq X$, $i_r|_{r=1}^{l} \in \{1, 2, ..., n\}$, with the corresponding labels $y_{i_r} \in \{1, 2, ..., c\}$, our task is to find a set of projective vectors $W = [w_1, w_2, ..., w_d]$, such that the transformed low-dimensional representations $z_i = W^T x_i$, not only can preserve the structure of $\boldsymbol{X}$ but also can reflect the discriminant information in L.

The objective function of PCA is defined as maximizing

$$J_{\text{PCA}} = \frac{1}{n}\sum_{i=1}^{n}\left(w^T x_i - w^T m\right)^2 = w^T S_T w \tag{1}$$

where $m = \frac{1}{n}\sum_{i=1}^{n} x_i$, $S_T = \frac{1}{n}\sum_{i=1}^{n}(x_i - m)(x_i - m)^T$ is the covariance matrix and also called as the normalized total scatter matrix. For the convenience of discussion, one-dimensional case is considered here but it is not difficult to extend to high-dimensions.

From Eq. 1, PCA does not use the discriminant information in labeled data set $\boldsymbol{L}$ at all. To make PCA have the discriminant power, without losing its data representation character, we propose the following objective function

$$J_{\text{DPCA}} = J_{\text{D}} + \lambda J_{\text{PCA}} \tag{2}$$

Here, J_{PCA} which is defined in Eq. 1, is the criterion of PCA, and J_{D} denotes some discriminant criterion on labeled data set $\boldsymbol{L}$. In Eq. 2, λ is a regularized coefficient balancing the contributions of two terms. In this paper, we adopt the following criterion as maximizing J_{D}

$$J_{\mathrm{D}} = w^T \left(S_B^L - \eta S_W^L \right) w \tag{3}$$

Where $\mathbf{S}_B^L$ and $\mathbf{S}_W^L$ are respectively defined in the following Eqs. 4 and 5, and η is a regularized coefficient balancing the contributions of two terms.

$$\mathbf{S}_B^L = \frac{1}{|\Omega_B|} \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \Omega_B} \left(\boldsymbol{x}_i - \boldsymbol{x}_j \right) \left(\boldsymbol{x}_i - \boldsymbol{x}_j \right)^T \tag{4}$$

$$\mathbf{S}_W^L = \frac{1}{|\Omega_W|} \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \Omega_W} \left(\boldsymbol{x}_i - \boldsymbol{x}_j \right) \left(\boldsymbol{x}_i - \boldsymbol{x}_j \right)^T \tag{5}$$

Where $|A|$ denotes the cardinality of a set A, and Ω_B and Ω_W is respectively defined by Eqs. 6 and 7 as follows

$$\Omega_B = \left\{ \left(x_i, x_j \right) \middle| \, x_i, x_j \in L \quad \text{and} \quad y_i \neq y_j \right\} \tag{6}$$

$$\Omega_W = \left\{ \left(x_i, x_j \right) \middle| \, x_i, x_j \in L \quad \text{and} \quad y_i = y_j \right\} \tag{7}$$

We call $\mathbf{S}_B^L$ and $\mathbf{S}_W^L$ as generalized between-class scatter matrix and generalized within-class scatter matrix respectively. The intuition between Eq. 3 is to let the average distance in the transformed low-dimensional space between data examples in different classes as large as possible, while distance between data examples with the same class as small as possible.

Substituting Eqs. 1 and 3 into Eq. 2, we obtain the objective function of DPCA as maximizing J_{DPCA} w.r.t.$w^T w = 1$, where

$$J'_{\mathrm{DPCA}} = w^T \left(S_B^L - \eta S_W^L + \lambda S_T \right) w \tag{8}$$

Clearly, Eq. 8 is a typical eigen-problem, which has a closed form solution by computing the eigen vectors of $S_B^L - \eta S_W^L + \lambda S_T$ corresponding to the largest eigen values. The whole procedure of the proposed DPCA algorithm is summarized in Algorithm 1 as below.

Algorithm 1: DPCA

Input: Data set $X = [x_1, x_2, ..., x_n]$, labeled data set $L = \left[x_{i_1}, x_{i_2}, ..., x_{i_l} \right] \subseteq X$ and corresponding class labels $y_{i_r} \in \{1, 2, ..., c\}$, $i_r|_{r=1}^{l} \in \{1, 2, ..., n\}$; parameters η,λ,d.

Output: Projective matrix $W = [w_1, w_2, ..., w_d]$.

Step 1: Construct the sets Ω_B and Ω_W from labeled data set $\boldsymbol{L}$ according to Eqs. 6 and 7 respectively.

Step 2: Compute $\mathbf{S}_B^L$ and $\mathbf{S}_W^L$ using Eqs. 4 and 5 respectively.

Step 3: Compute $S_T = \frac{1}{n} \sum_{i=1}^{n} (x_i - m) (x_i - m)^T$, $m = \frac{1}{n} \sum_{i=1}^{n} x_i$.

Step 4: Compute the d eigenvectors W of $\mathbf{S}_B^L - \eta \mathbf{S}_W^L + \lambda \mathbf{S}_T$ corresponding to the largest d eigenvalues.

3.2 DPCA with Pairwise Constraints

In general, domain knowledge can be expressed in diverse forms, such as class labels, pairwise constraints or other prior information [14]. Pairwise constraints arise naturally in many tasks such as image retrieval. In those applications, considering the pairwise constraints is more practical than trying to obtain class labels, because the true labels may not be known a priori, while it could be easier for a user to specify whether some pairs of instances belong

to the same class or not. Moreover, the pairwise constraints can be derived from labeled data but not vice versa. Furthermore, unlike class labels, the pairwise constraints can sometimes be automatically obtained without human intervention [6]. Fortunately, our DPCA algorithm can easily utilize both pairwise constraints and labeled data.

Given some supervision information in the form of must-link constraint set $M = \{(x_i, x_j) | x_i$ and x_j belongs to the same class$\}$ and cannot-link constraint set $C = \{(x_i, x_j) | x_i$ and x_j belongs to the different classes$\}$, we can define the new generalized between-class scatter matrix $\mathbf{S}_B^{L'}$ and generalized with-class scatter matrix $\mathbf{S}_W^{L'}$ using both pairwise constraints sets M, C and the labeled data set L as follows

$$\mathbf{S}_B^{L'} = \frac{1}{|\Omega_B \cup C|} \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \Omega_B \cup C} (\boldsymbol{x}_i - \boldsymbol{x}_j) (\boldsymbol{x}_i - \boldsymbol{x}_j)^T \tag{9}$$

$$\mathbf{S}_W^{L'} = \frac{1}{|\Omega_W \cup M|} \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \Omega_W \cup M} (\boldsymbol{x}_i - \boldsymbol{x}_j) (\boldsymbol{x}_i - \boldsymbol{x}_j)^T \tag{10}$$

Then we can obtain the new objective function of DPCA as maximizing J'_{DPCA} w.r.t.$w^T w = 1$, where

$$J'_{\mathrm{DPCA}} = w^T \left(S_B^{L'} - \eta S_W^{L'} + \lambda S_T \right) w. \tag{11}$$

4 Experiments

In this section, we evaluate the performance of our proposed DPCA algorithm on several UCI data sets [15] including *Dermatology*, *Horse*, *Iris*, *Lymph*, *Sonar*, *Soybean*, *Vowel* and *Wine*, and on one face database: YaleB [16]. Table 1 gives the statistics of the 8 UCI data sets. For each UCI data set, we choose the first half of samples from each class as the training data, and the remaining for testing data. Then we randomly select a few data samples from the training data as the labeled data. The process is repeated for 100 runs and the averaged results are recorded.

The performances of all algorithms are measured by the classification accuracy on testing data. In all experiments, the nearest neighborhood (1-NN) classifier is employed for classification, after dimensionality reduction with the above algorithms. For DPCA, we choose the values for parameters η and λ from the set $\{0.1, 1, 10\}$. More specifically, For *Horse*, *Iris*,

Table 1 Statistics of the UCI data sets

Data sets	Size	Dimension	# Of classes
Dermatology	366	33	6
Horse	368	27	2
Iris	150	4	3
Lymph	148	18	4
Sonar	208	60	2
Soybean	47	35	4
Vowel	528	10	11
Wine	178	13	3