Quadratic Nonnegative Matrix Factorization

Zhirong Yang*, Erkki Oja

Aalto University, Department of Information and Computer Science, P.O.Box 15400, FI-00076, Aalto, Finland

Abstract

In Nonnegative Matrix Factorization (NMF), a nonnegative matrix is approximated by a product of lower-rank factorizing matrices. Most NMF methods assume that each factorizing matrix appears only once in the approximation, thus the approximation is linear in the factorizing matrices. We present a new class of approximative NMF methods, called Quadratic Nonnegative Matrix Factorization (QNMF), where some factorizing matrices occur twice in the approximation. We demonstrate QNMF solutions to four potential pattern recognition problems in graph partitioning, two-way clustering, estimating hidden Markov chains, and graph matching. We derive multiplicative algorithms that monotonically decrease the approximation error under a variety of measures. We also present extensions in which one of the factorizing matrices is constrained to be orthogonal or stochastic. Empirical studies show that for certain application scenarios, QNMF is more advantageous than other existing nonnegative matrix factorization methods.

Keywords: nonnegative matrix factorization, multiplicative update,

*Corresponding author

Email addresses: zhirong.yang@aalto.fi (Zhirong Yang), erkki.oja@aalto.fi (Erkki Oja)

stochasticity, orthogonality, clustering, graph partitioning, Hidden Markov Chain Model, graph matching

1. Introduction

Extensive research on Nonnegative Matrix Factorization (NMF) has emerged in recent years (e.g. [1, 2, 3, 4, 5, 6]). NMF has found a variety of applications in machine learning, signal processing, pattern recognition, data mining, information retrieval, etc. (e.g. [7, 8, 9, 10, 11]). Given an input data matrix, NMF finds an approximation that is factorized into a product of lower-rank matrices, some of which are constrained to be nonnegative. The approximation error can be measured by a variety of divergences between the input and its approximation (e.g. [12, 13, 14, 6]), and the factorization can take a number of different forms (e.g. [15, 16, 17]).

In most existing NMF methods, each factorizing matrix appears only once in the approximation. We call them *linear* NMF because the approximation is linear with respect to each factorizing matrix. However, such linearity assumption does not hold in some important real-world problems. A typical example is graph matching, when it is presented as a matrix factorizing problem, as pointed out by Ding et al. [18]. If two graphs are represented by their adjacency matrices $\mathbf{A}$ and $\mathbf{B}$, then they are isomorphic if and only if a permutation matrix $\mathbf{P}$ can be found such that $\mathbf{A} - \mathbf{PBP}^T = 0$. Minimizing the norm or some other suitable error measure of the left-hand side with respect to $\mathbf{P}$, with suitable constraints, reduces the problem to an NMF problem. The approximation is now quadratic in $\mathbf{P}$.

Another example is clustering: if $\mathbf{X}$ is a matrix whose n columns need

to be clustered into r clusters, then the classical K-means objective function can be written as [19] $\mathcal{J}_1 = Tr(\mathbf{X}^T\mathbf{X}) - Tr(\mathbf{U}^T\mathbf{X}^T\mathbf{X}\mathbf{U})$ where $\mathbf{U}$ is the $(n \times r)$ binary cluster indicator matrix. The global minimum with respect to $\mathbf{U}$ gives the optimal clustering. It was shown in [20] that minimizing $\mathcal{J}_2 = \|\mathbf{X}^T - \mathbf{W}\mathbf{W}^T\mathbf{X}^T\|^2_{\text{Frobenius}}$ with respect to an orthogonal and nonnegative matrix $\mathbf{W}$ gives the same solution, except for the binary constraint. This is another NMF problem where the approximation is quadratic in $\mathbf{W}$.

Although the need of quadratic factorizations has been occasionally addressed in the literature (e.g. [21, 18, 17]), there has been no systematic study of higher-order NMF. A general way to obtain efficient optimization algorithms is lacking as well.

In this paper we focus on a class of NMF methods where some factorizing matrices occur twice in the approximation. We call these methods *Quadratic Nonnegative Matrix Factorization* (QNMF). Our major contributions are highlighted as follows: (1) QNMF objectives that are composed of different factorization forms and various approximation error measures are formulated. (2) Solving a QNMF problem is generally more difficult than the linear ones, because the former usually involves a higher-degree objective function with respect to the doubly-occurring factorizing matrices. In [22], we recently introduced a general approach for deriving multiplicative update rules for NMF, based on most commonly used approximation error measures. We now apply it to more general QNMF factorizations. Iterative algorithms that utilize the derived rules can all be shown to guarantee monotonic decrease, thus convergence, of the objective function. (3) We present techniques for extending the algorithms to easily accommodate the stochas-

ticity or orthogonality constraint, needed for such applications as estimating a hidden Markov chain, clustering, or graph matching. (4) In addition to the advantages in clustering, already shown in our previous publication [20], we demonstrate that the proposed QNMF algorithms outperform the existing NMF implementations in graph partitioning, two-way clustering, estimating a hidden Markov chain, and graph matching.

In the rest of the paper, we first briefly review the nonnegative matrix factorization problem and related work in Section 2. Next we formulate the Quadratic Nonnegative Matrix Factorization problem in Section 3, with its multiplicative optimization algorithms, learning with the stochasticity and orthogonality constraints given in Section 4. Section 5 presents experimental results, including comparisons between QNMF and other state-of-the-art NMF methods for four selected applications, as well as the demonstration of accelerated and online QNMF. Some conclusions and future perspectives are given in Section 6.

2. Related Work

Given an input data matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, *Nonnegative Matrix Factorization* (NMF) finds an approximation $\widehat{\mathbf{X}}$ which can be factorized into a product of matrices:

$$\mathbf{X} \approx \widehat{\mathbf{X}} = \prod_{q=1}^{Q} \mathbf{F}^{(q)} \tag{1}$$

and some of these matrices are constrained to be nonnegative. The dimensions of the factorizing matrices $\mathbf{F}^{(1)}, \ldots, \mathbf{F}^{(Q)}$ are $m \times r_1, r_1 \times r_2, \ldots, r_{Q-1} \times n$, respectively. Usually $r_1, \ldots, r_{Q-1}$ are much smaller than m or n. In this way

the factorization represents the data in a more compact way and thus favors certain applications.

The difference between the input matrix $\mathbf{X}$ and its approximation $\widehat{\mathbf{X}}$ can be measured by a variety of divergences, for which theoretically convergent multiplicative algorithms of NMF have been proposed[1]. Originally, NMF is based on the Euclidean distance or I-divergence (non-normalized Kullback-Leibler divergence) [1, 2]. These two measures were later unified by using the β-divergence [12]. Alternatively, Cichocki et al. [14] generalized NMF from I-divergence to the whole family of α-divergences. NMF has been further extended to a even broader class called Bregman divergence, for many cases of which a general convergence proof can be found [13]. In addition to separable ones, divergences that are non-separable with respect to the matrix entries such as the γ-divergence and Rényi divergence can also be employed by NMF [11]. Many other divergences for measuring the approximation error in NMF exist, although they may lack theoretically convergent algorithms.

In its original form [1, 2], the NMF approximation is factorized into two matrices ($Q = 2$). Later the factorization was generalized to three factors (e.g. [23, 15, 16, 17]). Note that the input matrix and some factorizing matrices are not necessarily nonnegative, (e.g. Semi-NMF in [17]). Also, one of the factorizing matrix can be the input matrix itself, for example, the Convex NMF [17].

Besides the nonnegativity, various constraints or regularizations can be

[1]Unless otherwise stated, the term "convergence" in this paper generally refers to the *objective function convergence* or, equivalently, the monotonic decrease of the NMF approximation error.

imposed on the factorizing matrices. Matrix norms such as L_1- or L_2-norm have been used for achieving sparseness or smoothness (e.g. [24, 3, 8]). Orthogonality combined with nonnegativity can significantly enhance the part-based representation or category indication (e.g. [16, 25, 4, 26, 20]).

3. Quadratic Nonnegative Matrix Factorization

In most existing NMF approaches, the factorizing matrices $\mathbf{F}^{(q)}$ in Eq. (1) are all different, and thus the approximation $\widehat{\mathbf{X}}$ as a function of them is *linear*. However, there are useful cases where some matrices appear more than once in the approximation. In this paper we consider the case that some of them may occur twice, or formally, $\mathbf{F}^{(s)} = \mathbf{F}^{(t)^T}$ for a number of non-overlapping pairs $\{s, t\}$ and $1 \leq s < t \leq Q$. We call such a problem and its solution *Quadratic Nonnegative Matrix Factorization* (QNMF) because $\widehat{\mathbf{X}}$ as a function is quadratic to each twice appearing factorizing matrix[2].

To avoid notational clutter, we focus on the case where the input matrix and all factorizing matrices in the approximation are nonnegative, while our discussion can easily be extended to the cases where some matrices may contain negative entries by using the decomposition technique presented in [17].

3.1. Factorization forms

To begin with, let us consider the QNMF objective with only one doubly occurring matrix $\mathbf{W}$. The general approximating factorization form is given

[2]Though equality without matrix transpose, namely $\mathbf{F}^{(s)} = \mathbf{F}^{(t)}$, is also possible, to our knowledge there are no corresponding real-world applications.

by

$$\widehat{\mathbf{X}} = \mathbf{AWBW}^T\mathbf{C}, \tag{2}$$

where we merge the products of the other, linearly appearing factorizing matrices into single symbols. It is also possible that matrix $\mathbf{A}$, $\mathbf{B}$, and/or $\mathbf{C}$ is the identity matrix and thus vanishes from the expansion. Here we focus on the optimization over $\mathbf{W}$, as learning the matrices that occur only once can be solved by using the conventional NMF methods of alternative optimization over each matrix separately [11].

The above factorization form unifies all previously suggested QNMF objectives. For example, it becomes the *Projective Nonnegative Matrix Factorization* (PNMF) when $\mathbf{A} = \mathbf{B} = \mathbf{I}$ and $\mathbf{C} = \mathbf{X}$, which was first introduced by Yuan and Oja [21] and later extended by Yang et al. [25, 27, 20]. This factorized form is also named *Clustering NMF* in [17] as a constrained case of *Convex NMF*. Even without any explicit orthogonality constraint, the matrix $\mathbf{W}$ obtained by using PNMF is highly orthogonal and can thus serve two purposes: (1) when the columns of $\mathbf{X}$ are samples, $\mathbf{W}$ can be seen as the basis for part-based representations, and (2) when the rows of $\mathbf{X}$ are samples, $\mathbf{W}$ can be used as a cluster indicator matrix.

If $\mathbf{X}$ is a square matrix and $\mathbf{A} = \mathbf{C} = \mathbf{I}$, the factorization can be used in two major scenarios if the learned $\mathbf{W}$ is highly orthogonal. (1) When $\mathbf{B}$ is much smaller than $\mathbf{X}$, the three-factor approximation corresponds to a blockwise representation of $\mathbf{X}$ [23, 28]. If $\mathbf{B}$ is diagonal, then the representation becomes diagonal blockwise, or a partition. In the extreme case $\mathbf{B} = \mathbf{I}$, the factorization reduces to the *Symmetric Nonnegative Matrix Factorization* (SNMF) $\widehat{\mathbf{X}} = \mathbf{WW}^T$ [16]. (2) When $\mathbf{X}$ and $\mathbf{B}$ are of the same size,

the learned $\mathbf{W}$ with the constraint $\mathbf{W}^T\mathbf{W} = \mathbf{I}$ approximates a permutation matrix and thus QNMF can be used for learning order of relational data, for example, graph matching [18]. Alternatively, under the constraint that $\mathbf{W}$ has column-wise unitary sums, the solution of such a QNMF problem provides parameter estimation of hidden Markov chains (See Section 5.3).

The factorization form in Eq. (2) also generalizes the concepts of *Asymmetric Quadratic Nonnegative Matrix Factorization* (AQNMF) $\widehat{\mathbf{X}} = \mathbf{W}\mathbf{W}^T\mathbf{C}$ and *Symmetric Quadratic Nonnegative Matrix Factorization* (SQNMF) $\widehat{\mathbf{X}} = \mathbf{W}\mathbf{B}\mathbf{W}^T$ in our previous work [22].

Note that the factorization form in Eq. (2) is completely general: it can be recursively applied to the cases where there are more than one factorizing matrices appearing quadratically in the approximation. For example, the case $\mathbf{A} = \mathbf{C}^T = \mathbf{U}$ yields $\widehat{\mathbf{X}} = \mathbf{U}\mathbf{W}\mathbf{B}\mathbf{W}^T\mathbf{U}^T$, and $\mathbf{A} = \mathbf{B} = \mathbf{I},\ \mathbf{C} = \mathbf{X}\mathbf{U}\mathbf{U}^T$ yields $\widehat{\mathbf{X}} = \mathbf{W}\mathbf{W}^T\mathbf{X}\mathbf{U}\mathbf{U}^T$. An application of the latter example is shown in Section 5.2, where the solution of such a QNMF problem can be used to group the rows and columns of $\mathbf{X}$ simultaneously. This is particularly useful for the biclustering or coclustering problem. These factorizing forms can be further generalized to any number of factorizing matrices. In such cases we employ alternative optimization over each doubly occurring matrix.

It is important to notice that quadratic NMF problems are not special cases of linear NMF. The optimization methods of the latter generally cannot be extended to QNMF. For linear NMF, the factorizing matrices are different variables and the approximation error can alternatively be minimized over one of them while keeping the others constant. In contrast, the optimization of QNMF is more comprehensive because matrices in two places vary simul-

taneously, which leads to higher-order objectives. For example, given the squared Frobenius norm (Euclidean distance) as approximation error measure, the objective of linear NMF $\|\mathbf{X} - \mathbf{WH}\|_F^2$ is quadratic with respect to $\mathbf{W}$ and $\mathbf{H}$, whereas the PNMF objective $\|\mathbf{X} - \mathbf{WWX}\|_F^2$ is quartic with respect to $\mathbf{W}$. Minimizing such a fourth-order objective with the nonnegativity constraint is considerably more challenging than minimizing a quadratic function.

In contrast to the rich selection of approximation error measures for linear NMF, there is little research on such measures for quadratic NMF. In this paper we show that QNMF can be built on a very wide class of dissimilarity measures, quite as rich as those for linear NMF. Furthermore, as long as the QNMF objective can be expressed in a generalized polynomial form described below, we show that there always exists a multiplicative algorithm that theoretically guarantees convergence, or the monotonic decrease of the objective function, in each iteration.

In the rest of the paper, we distinguish the variable $\widetilde{\mathbf{W}}$ from its current estimate $\mathbf{W}$. We write $\widetilde{\mathbf{X}} = \mathbf{A}\widetilde{\mathbf{W}}\mathbf{B}\widetilde{\mathbf{W}}^T\mathbf{C}$ to denote the approximation that contains the variable $\widetilde{\mathbf{W}}$, and $\widehat{\mathbf{X}} = \mathbf{AWBW}^T\mathbf{C}$ for the current estimate (constant).

3.2. Deriving multiplicative algorithms

Multiplicative updates have been commonly used for optimizing NMF objectives. In each epoch, a factorizing matrix is updated by elementwise multiplication with a nonnegative matrix which can easily be obtained from the gradient. The multiplicative algorithms have a couple of advantages over the conventional additive gradient descent approach. Firstly, they naturally

maintain the nonnegativity of the factorizing matrices without any extra projection steps. Secondly, the fixed-point algorithm that iteratively applies the update rule requires no user-specified parameters such as the learning step size, which facilitates its implementation and applications. Although some heuristic connections to conventional additive update rules exist [2, 25], they cannot theoretically justify the objective function convergence.

The rigorous convergence proof, or theoretical guarantee of monotonic decrease of the objective function, focusses on minimizing a certain auxiliary upper-bounding function which is defined as follows. $G(\mathbf{W}, \mathbf{U})$ is called an auxiliary function if it is a tight upper bound of the objective function $\mathcal{J}(\mathbf{W})$, i.e. $G(\mathbf{W}, \mathbf{U}) \geq \mathcal{J}(\mathbf{W})$, and $G(\mathbf{W}, \mathbf{W}) = \mathcal{J}(\mathbf{W})$ for any $\mathbf{W}$ and $\mathbf{U}$. Define

$$\mathbf{W}^{\text{new}} = \arg\min_{\widetilde{\mathbf{W}}} G(\widetilde{\mathbf{W}}, \mathbf{W}). \tag{3}$$

By construction, $\mathcal{J}(\mathbf{W}) = G(\mathbf{W}, \mathbf{W}) \geq G(\mathbf{W}^{\text{new}}, \mathbf{W}) \geq G(\mathbf{W}^{\text{new}}, \mathbf{W}^{\text{new}}) = \mathcal{J}(\mathbf{W}^{\text{new}})$, where the first inequality is the result of minimization and the second comes from the upper bound. Iteratively applying the update rule (3) thus results in a monotonically decreasing sequence of $\mathcal{J}$. Besides the tight upper bound, it is often desired that the minimization (3) has a closed-form solution. In particular, setting $\partial G / \partial \widetilde{\mathbf{W}} = 0$ should lead to the iterative update rule in analysis. The construction of such an auxiliary function, however, has not been a trivial task so far.

In [22], we recently derived the convergent multiplicative update rules for a wide class of divergences for the NMF problem. For conciseness, we repeat the central steps here but for details refer to [22].

Let us first generalize the definitions of monomials and polynomials: a monomial with real-valued exponent, or shortly *monomial*, of the scalar vari-

able z is of the form az^b where a and b can take any real value, without restriction to nonnegative integers. A sum of a (finite) number of monomials is called a (finite) *generalized polynomial*.

This form of expression has two nice properties: (1) individual monomials, denoted by az^b, are either convex or concave with respect to z and thus can easily be upper-bounded; (2) an exponential is multiplicatively decomposable, i.e. $(xy)^a = x^a y^a$, which is critical in deriving the multiplicative update rule. Note that we unify the logarithm function that is involved in many information-theoretic divergences to our generalized polynomial form by using the *logarithm limit*

$$\ln z = \lim_{\epsilon \to 0^+} \frac{z^\epsilon - 1}{\epsilon}. \tag{4}$$

Notice that limits in 0^+ and 0^- are the same. We use the former to remove the convexity ambiguity. In this way, the logarithm can be decomposed into two monomials where the first contains an infinitesimally small positive exponent.

Next, we show that the auxiliary upper-bounding function always exists as long as the approximation objective function can be expressed in the finite generalized polynomial form. This is formalized by the following theorem in our previous work [22].

Theorem 1. *Assume* $\Omega_d(z) = \rho_d \cdot z^{\chi_d}$, *and* ρ_d, χ_d, g_{dij} *and* ϕ_d *are constants independent of* $\widetilde{\mathbf{W}}$. *If the approximation error has the form*

$$D(\mathbf{X}||\widehat{\mathbf{X}}) = \sum_{d=1}^{p} \Omega_d \left(\sum_{i=1}^{m} \sum_{j=1}^{n} g_{dij} \widehat{X}_{ij}^{\phi_d} \right) + \textit{constant}, \tag{5}$$

then there are real numbers ψ_{max} and ψ_{min} ($\psi_{max} > \psi_{min}$) such that

$$G(\widetilde{\mathbf{W}}, \mathbf{W}) = \sum_{ik} \frac{W_{ik}}{\psi_{max}} \left(\frac{\widetilde{W}_{ik}}{W_{ik}}\right)^{\psi_{max}} \nabla^+_{ik} - \frac{W_{ik}}{\psi_{min}} \left(\frac{\widetilde{W}_{ik}}{W_{ik}}\right)^{\psi_{min}} \nabla^-_{ik} + constant \tag{6}$$

is an auxiliary upper-bounding function of $\mathcal{J}(\widetilde{\mathbf{W}}) \triangleq D\left(\mathbf{X}||\widetilde{\mathbf{X}}\right)$. There ∇^+ and ∇^- denote the sums of positive and unsigned negative terms of $\nabla = \left.\frac{\partial \mathcal{J}(\widetilde{\mathbf{W}})}{\partial \widetilde{\mathbf{W}}}\right|_{\widetilde{\mathbf{W}}=\mathbf{W}}$ (i.e. $\nabla = \nabla^+ - \nabla^-$), respectively.

The theorem proof, or the construction procedure of the auxiliary function is summarized in the following procedure: 1) Write the objective function in the form of finite generalized polynomials given by Eq. (5); Use the logarithm limit Eq. (4) when necessary; 2) If the objective is not a separable sum over i and j, i.e. there is $\chi_d \neq 1$, derive a separable upper-bound of the objective using the concavity or convexity inequality on Ω_d; 3) Upper bound individual monomials according to their concavity/convexity and leading signs; 4) If there are more than two upper-bounds with distinct exponents, combine them into two monomials according to their exponents and leading signs. The derivation details are similar to those given in our previous work [22] and thus omitted here.

Taking the derivative of the auxiliary function with respect to $\widetilde{\mathbf{W}}$ leads to

$$\frac{\partial G(\widetilde{\mathbf{W}}, \mathbf{W})}{\partial \widetilde{W}_{ik}} = \left(\frac{\widetilde{W}_{ik}}{W_{ik}}\right)^{\psi_{\max}-1} \nabla^+_{ik} - \left(\frac{\widetilde{W}_{ik}}{W_{ik}}\right)^{\psi_{\min}-1} \nabla^-_{ik}. \tag{7}$$

As stated above, setting this to zero gives the iterative update rule, which results in a monotonically decreasing, hence convergent, sequence of $\mathcal{J}(\mathbf{W})$.

In most cases, this multiplicative update rule has the form

$$W_{ik}^{\text{new}} = W_{ik} \left(\frac{\nabla_{ik}^{-}}{\nabla_{ik}^{+}} \right)^{\eta}, \tag{8}$$

where $\eta = 1/(\psi_{\max} - \psi_{\min})$. An exception, for example the dual I-divergence, happens when the logarithm limit is applied and $\lim_{\epsilon \to 0+} (\psi_{\max} - \psi_{\min}) = 0$. In this case the limit of the derivative Eq. (7) has the form $\frac{0}{0}$ and can be resolved by using the L'Hôpital's rule to obtain the limit before setting it to zero.

The finite generalized polynomial form Eq. (5) covers most commonly used dissimilarity measures. Here we present the multiplicative update rules for QNMF based on α-divergence, β-divergence, γ-divergence and Rényi divergence. These families include, for example, the squared Euclidean distance ($\beta = 1$), Hellinger distance ($\alpha = 0.5$), χ^2-divergence ($\alpha = 2$), I-divergence ($\alpha \to 1$ or $\beta \to 0$), dual I-divergence ($\alpha \to 0$), Itakura-Saito divergence ($\beta \to -1$) and Kullback-Leibler divergence ($\gamma \to 0$ or $r \to 1$). Though not presented here, more QNMF objectives, for example the additive hybrids of the above divergences, as well as many other unnamed Csiszár divergences and Bregman divergences, can be easily derived using the proposed principle.

In general, the multiplicative update rules take the following form:

$$W_{ik}^{\text{new}} = W_{ik} \left[\frac{\left(\mathbf{A}^T \mathbf{Q} \mathbf{C}^T \mathbf{W} \mathbf{B}^T + \mathbf{C} \mathbf{Q}^T \mathbf{A} \mathbf{W} \mathbf{B}\right)_{ik}}{\left(\mathbf{A}^T \mathbf{P} \mathbf{C}^T \mathbf{W} \mathbf{B}^T + \mathbf{C} \mathbf{P}^T \mathbf{A} \mathbf{W} \mathbf{B}\right)_{ik}} \cdot \theta \right]^{\eta}, \tag{9}$$

where $\mathbf{P}$, $\mathbf{Q}$, θ, and η are specified in Table 1. For example, the rule for QNMF $\mathbf{X} \approx \mathbf{W}\mathbf{B}\mathbf{W}^T$ based on the squared Euclidean distance ($\beta \to 1$) reads

$$W_{ik}^{\text{new}} = W_{ik} \left[\frac{\left(\mathbf{X} \mathbf{W} \mathbf{B}^T + \mathbf{X}^T \mathbf{W} \mathbf{B}\right)_{ik}}{\left(\mathbf{W} \mathbf{B} \mathbf{W}^T \mathbf{W} \mathbf{B}^T + \mathbf{W} \mathbf{B}^T \mathbf{W}^T \mathbf{W} \mathbf{B}\right)_{ik}} \right]^{1/4}.$$

Table 1: Notations in the multiplicative update rules of QNMF examples, where $\widehat{\mathbf{X}} = \mathbf{AWBW}^T\mathbf{C}$.

divergence	P_{ij}	Q_{ij}	θ	η
α-divergence	1	$X_{ij}^{\alpha}\widehat{X}_{ij}^{-\alpha}$	1	$1/(2\alpha)$ for $\alpha > 1$ $1/2$ for $0 < \alpha < 1$ $1/(2\alpha - 2)$ for $\alpha < 0$
β-divergence	$\widehat{X}_{ij}^{\beta}$	$X_{ij}\widehat{X}_{ij}^{\beta-1}$	1	$1/(2 + 2\beta)$ for $\beta > 0$ $1/(2 - 2\beta)$ for $\beta < 0$
γ-divergence	$\widehat{X}_{ij}^{\gamma}$	$X_{ij}\widehat{X}_{ij}^{\gamma-1}$	$\frac{\sum_{ab}\widehat{X}_{ab}^{\gamma+1}}{\sum_{ab}X_{ab}\widehat{X}_{ab}^{\gamma}}$	$1/(2 + 2\gamma)$ for $\gamma > 0$ $1/(2 - 2\gamma)$ for $\gamma < 0$
Rényi divergence	1	$X_{ij}^{r}\widehat{X}_{ij}^{-r}$	$\frac{\sum_{ab}\widehat{X}_{ab}}{\sum_{ab}X_{ab}^{r}\widehat{X}_{ab}^{1-r}}$	$1/(2r)$ for $r > 1$ $1/2$ for $0 < r < 1$

As an exception, the update rules for the dual I-divergence takes a different form

$$W_{ik}^{\text{new}} = W_{ik} \exp\left[\frac{1}{2}\frac{\left(\mathbf{A}^T\mathbf{Q}\mathbf{C}^T\mathbf{W}\mathbf{B}^T + \mathbf{C}\mathbf{Q}^T\mathbf{A}\mathbf{W}\mathbf{B}\right)_{ik}}{\left(\mathbf{A}^T\mathbf{P}\mathbf{C}^T\mathbf{W}\mathbf{B}^T + \mathbf{C}\mathbf{P}^T\mathbf{A}\mathbf{W}\mathbf{B}\right)_{ik}}\right]$$

with $P_{ij} = 1$ and $Q_{ij} = \ln\left(X_{ij}/\widehat{X}_{ij}\right)$. In practice, iteratively applying the above update rules except the dual-I case can also achieve K.K.T. optimality asymptotically [22].

Multiplicative algorithms using an identical update rule throughout the iterations are simple to implement. However, the exponent η that remains constant is often conservative (too small) in practice. This can be accelerated by using a more aggressive choice of the exponent, which adaptively changes during the iterations. A simple strategy is to increase the exponent steadily

if the new objective is smaller than the old one and otherwise shrink back to the safe choice, η. We find that such a strategy can significantly speed up the convergence while still maintaining the monotonicity of the updates [22].

For very large data matrices, there are scalable implementations of QNMF for the case when the approximation error is the squared Euclidean distance. An online learning algorithm of PNMF was presented in our previous work [29], where we do not need to operate on the whole data matrix but only on a small intermediate variable instead. The resulting algorithm not only has low memory cost but runs faster than the batch version for large datasets.

4. Constrained QNMF

NMF objectives are often accompanied with a set of constraints. The manifolds induced by these constraints intersect the nonnegative quadrant and form an extensive and non-smooth function domain, which makes the optimization difficult. Conventional optimization approaches which operate in such domains only work for a few particular cases of linear NMF but seldom succeed for quadratic NMF problems. Alternatively, we adopt a relaxation technique to solve the constrained QNMF problems: first we attach the (soft) constraints to the objective as regularization terms; next we write a multiplicative update rule for the augmented objective, where the Lagrangian multipliers are solved by using the K.K.T. conditions; finally putting back the multipliers we obtain new update rules with the (soft) constraints incorporated. A similar idea was also employed by Ding et al. [16]. We call the resulting algorithm *iterative Lagrangian solution* of the constrained QNMF problem. Such an algorithm jointly minimizes the QNMF approximation er-

ror and forces the factorizing matrices to approach the constraint manifold. In what follows, we show how to get such a solution for QNMF with the stochasticity or orthogonality constraint.

4.1. Stochastic matrices

Nonnegative matrices are often used to represent probabilities, where all or a part of the matrix elements must sum up to one. For concreteness we only focus on the case of a left stochastic matrix with column-wise unitary sum, i.e. $\sum_i W_{ik} = 1$, while the same method can easily be extended to row-wise constraints (right stochastic matrix) or matrix-wise constraints (doubly stochastic matrix). A general principle that incorporates the stochasticity constraint to an existing convergent QNMF algorithm is given by the following theorem.

Theorem 2. *Suppose a QNMF objective* $\mathcal{J}(\widetilde{\mathbf{W}})$ *to be minimized can be upper-bounded by the auxiliary function* $G(\widetilde{\mathbf{W}}, \mathbf{W})$ *in the form of Eq. (6), whose gradient with respect to* $\mathbf{W}$ *is given in Eq. (7). Introducing a set of Lagrangian multipliers* $\{\lambda_k\}$*, the augmented objective* $\mathcal{L}(\mathbf{W}, \boldsymbol{\lambda}) = \mathcal{J}(\widetilde{\mathbf{W}}) + \sum_k \lambda_k \left(1 - \sum_i W_{ik}\right)$ *is non-increasing under the update rule*

$$W_{ik}^{new} = W_{ik} \left[\frac{\nabla_{ik}^- + \sum_a \nabla_{ak}^+ W_{ak}}{\nabla_{ik}^+ + \sum_a \nabla_{ak}^- W_{ak}} \right]^{\sigma}, \tag{10}$$

where $\sigma = 1/[\max(\psi_{max}, 1) - \min(\psi_{min}, 1)]$.

The proof is given in Appendix A. Note that the above reforming principle also includes the dual-I divergence case where $\max(\psi_{\max}, 1) - \min(\psi_{\min}, 1) = 1$ or $\sigma = 1$. The inclusion also holds in the following principle for the orthogonality constraint.

4.2. Orthogonal matrices

Orthogonality is another frequently used constraint in NMF [see e.g. 16, 18, 25, 4, 20] because a nonnegative orthogonal matrix has only one non-zero entry in each row. In this way the matrix can serve as a membership indicator in learning problems such as clustering or classification. Strict orthogonality is however of little interest in NMF because the optimization problem remains discrete and often NP-hard. Relaxation is therefore needed, as long as there is only one large non-zero entry in each row of the resulting $\mathbf{W}$ while the other entries are close to zero. An approximative orthogonal matrix is then obtained by simple thresholding and rescaling.

Here we present a general principle that incorporates the orthogonality constraint to a theoretically convergent QNMF algorithm. In Appendix we have proven the following results.

Theorem 3. *Suppose a QNMF objective $\mathcal{J}(\widetilde{\mathbf{W}})$ to be minimized can be upper-bounded by the auxiliary function $G(\widetilde{\mathbf{W}}, \mathbf{W})$ in the form of Eq. (6), whose gradient with respect to $\mathbf{W}$ is given in Eq. (7). Introducing a set of Lagrangian multipliers $\{\Lambda_{kl}\}$, the augmented objective $\mathcal{L}(\mathbf{W}, \mathbf{\Lambda}) = \mathcal{J}(\widetilde{\mathbf{W}}) + Tr\left(\mathbf{\Lambda}\left(\mathbf{I} - \widetilde{\mathbf{W}}^T\widetilde{\mathbf{W}}\right)\right)$ is non-increasing under the update rule*

$$W_{ik}^{new} = W_{ik}\left[\frac{\left(\nabla^- + \mathbf{W}\mathbf{W}^T\nabla^+\right)_{ik}}{\left(\nabla^+ + \mathbf{W}\mathbf{W}^T\nabla^-\right)_{ik}}\right]^{\sigma}, \tag{11}$$

where $\sigma = 1/[\max(\psi_{max}, 2) - \min(\psi_{min}, 0)]$.

Corollary 4. *If $\mathbf{\Lambda}^+ = \frac{1}{2}\mathbf{W}^T\nabla^+$ is positive semi-definite, then $\sigma = 1/[\max(\psi_{max}, 2) - \min(\psi_{min}, 1)]$ in Eq. (11).*

Our derivation of the multiplicative rules for learning nonnegative projections is based on the Lagrangian approach. That is, one should apply $\mathbf{W}^T\mathbf{W} = \mathbf{I}$, one of the K.K.T. conditions, to simplify the resulting multiplicative update rule after transforming Eq. (8) to its orthogonal counterpart Eq. (11). Furthermore, removing duplicate terms that appear in both numerator and denominator in practice will lead to faster convergent update rule because $(a + c)/(b + c)$ is closer to one than a/b for $a, b, c > 0$.

Although with highly orthogonal results, it is important to notice that $\mathbf{W}$ never exactly reaches the Stiefel or Grassmann manifold during the optimization if it is initialized with positive entries. Therefore the reforming principle using natural gradients in such manifolds (e.g. [26, 4]) must be seen as an approximation.

5. Experiments

Quadratic Nonnegative Matrix Factorization can be applied to a variety of learning problems. Extensive empirical study of PNMF, a special case of QNMF, for feature extraction and clustering have been provided in [21, 25, 20]. In this paper we demonstrate four new example applications. The first one employs PNMF for grouping the nodes of a graph; the second one applies a two-sided QNMF to achieve two-way clustering; the third one uses QNMF with the stochasticity constraint to estimate the parameters in hidden Markov chains; and the fourth application uses QNMF with the orthogonality

constraint for finding node correspondence of two graphs.

The following publicly available datasets have been used in our experiments: Newman's collection[3], the Pajek database[4], the *3Conference* graph[5], the *webkb* text dataset[6], the top 58112 English words[7], the GenBank database[8], and the University of Florida Sparse Matrix Collection[9].

5.1. Graph partitioning

Given an undirected graph, the *graph partitioning* problem is to divide the graph vertices into several disjoint subsets, such that there are relatively few connections between the subsets. The optimization of many graph partitioning objectives, for example, minimizing the number of removed edges in partitioning, is NP-hard due to the complexity of algorithms in a discrete space. It is therefore advantageous to employ a continuous approximation of the objectives, which enables the development of efficient gradient descent - type optimization algorithms by using differential calculus.

Suppose the connections in the graph are represented by a real-valued $N \times N$ affinity matrix $\mathbf{A}$, whose element A_{ij} gives the weight of the edge connecting vertices i, j. A classical approximation approach is *spectral clustering* (e.g. [30]) which minimizes $\mathrm{Tr}\left\{\mathbf{U}^T(\mathbf{D}-\mathbf{A})\mathbf{U}\right\}$ subject to $\mathbf{U}^T\mathbf{D}\mathbf{U} = \mathbf{I}$, where $\mathbf{D}$ is a diagonal matrix with $D_{ii} = \sum_j A_{ij}$. Though the problem has a

[3] http://www-personal.umich.edu/~mejn/netdata/

[4] http://vlado.fmf.uni-lj.si/pub/networks/data/

[5] http://users.ics.tkk.fi/rozyang/3conf/

[6] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/

[7] http://www.mieliestronk.com/wordlist.html

[8] http://www.ncbi.nlm.nih.gov/genbank/

[9] http://www.cise.ufl.edu/research/sparse/matrices/index.html

closed form solution by using singular value decomposition, the resulting $\mathbf{U}$ may contain negative entries and thus cannot directly be used as an indicator for more than two clusters. An extra step that projects $\mathbf{U}$ to the positive quadrant is needed [31].

Ding et al. [18] presented the *Nonnegative Spectral Clustering* (NSC) method by introducing the nonnegativity constraint. They derived a multiplicative update rule as an iterative Lagrangian solution that jointly minimizes $\mathrm{Tr}\left\{\mathbf{U}^T(\mathbf{D}-\mathbf{A})\mathbf{U}\right\}$ and forces $\mathbf{U}$ to approach the manifold specified by $\mathbf{U}^T\mathbf{D}\mathbf{U}=\mathbf{I}$ and $\mathbf{U}\geq 0$.

We can connect the NSC problem to the PNMF approach. Rewriting $\mathbf{W}=\mathbf{D}^{1/2}\mathbf{U}$, which does not change the cluster indication because $\mathbf{D}$ is diagonal and positive, the NSC problem becomes maximization of $\mathrm{Tr}\left\{\mathbf{W}^T\left(\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}\right)\mathbf{W}\right\}$ over nonnegative $\mathbf{W}$ and subject to $\mathbf{W}^T\mathbf{W}=\mathbf{I}$. This is a non-negative PCA problem which can be solved by PNMF [25, 20]. Furthermore, we can replace $\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ with some other symmetric matrices or kernel matrices. A good kernel can enhance the clustering effect of the graph and yield better partitioning results. Here we use a non-parametric kernel called *random-walk* kernel that minimizes the combinatorial formulation of the Dirichlet integral [32, 33, 34]. In brief, the kernel is obtained by symmetrizing the solution of the following equation: $(2\mathbf{D}-\mathbf{A})\,\mathbf{B}=\mathbf{A}$. Consider the reproducing Hilbert space induced by the kernel $\mathbf{K}=\frac{\mathbf{B}+\mathbf{B}^T}{2}$ whose entries are nonnegative because $\mathbf{B}=(2\mathbf{D}-\mathbf{A})^{-1}\,\mathbf{A}=\frac{1}{2}\mathbf{D}^{-1}\sum_{i=0}^{\infty}\left(\frac{1}{2}\mathbf{A}\mathbf{D}^{-1}\right)^i\mathbf{A}$. The graph partitioning problem can thus be solved by applying kernel PNMF [20] in the implicit feature space to minimize $\|\mathbf{\Phi}(\mathbf{X})^T-\mathbf{W}\mathbf{W}^T\mathbf{\Phi}(\mathbf{X})^T\|_F^2$ subject to $\mathbf{W}\geq 0$, where $\mathbf{K}=\mathbf{\Phi}(\mathbf{X})^T\mathbf{\Phi}(\mathbf{X})$. The convergent update rule of ker-

nel PNMF derived from the proposed principle is $W_{ik}^{\text{new}} = W_{ik}\left[\frac{(2\mathbf{KW})_{ik}}{(\mathbf{WW}^T\mathbf{KW}+\mathbf{KWW}^T\mathbf{W})_{ik}}\right]^{1/3}$.

We have compared the Nonnegative Spectral Clustering and our kernel PNMF methods on four graphs: (1) *Korea*, a communication network of 39 women of two classes in Korea about family planning. (2) *WorldCities*, a dataset consists of the service values (indicating the importance of a city in the office network of a firm) of 100 global advanced producer service firms over 315 world cities. The firms are grouped into six categories: accountancy, advertising, banking/finance, law, insurance and management consultancy. (3) *PolBlogs*, a network of weblogs on US politics, with 1224 nodes. (4) *3Conference*, the coauthorship network in three conferences SIGIR, ICML, and ICCV from 2002-2007. There are 2817 authors, 14048 edges in total. The value on each edge represents the number of papers that two authors have coauthored. The first three datasets are gathered from Newman's collection[3] and the Pajek database[4]. The *3Conference* graph is available at the authors' website[5].

We have computed the purities of the resulting partitions by using the ground truth class information: given r partitions and q classes, purity $= \frac{1}{N}\sum_{k=1}^{r}\max_{1\le l\le q} N_k^l$, where N_k^l is the number of vertices in the partition k that belong to ground-truth class l. A larger purity in general corresponds to better clustering result. For each dataset, we repeated the compared methods 100 times with different random initializations and recorded the purities, whose means and standard deviations are illustrated in Figure 1. For small graphs *Korea* and *WorldCities*, kernel PNMF is at the same level or slightly better than the Nonnegative Spectral Clustering method. For larger graphs *PolBlogs* and *3Conference*, kernel PNMF is significantly superior with much

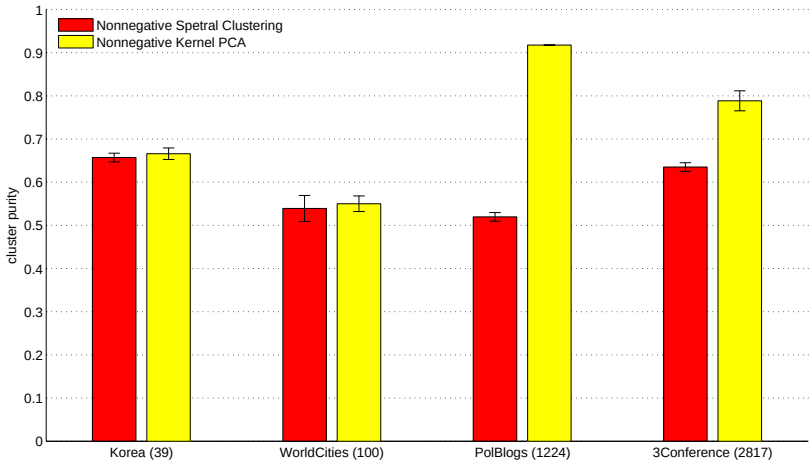

Figure 1: Cluster purities using the two compared graph partitioning methods. Numbers of graph nodes are shown in parentheses.

higher clustering purities.

The computational cost of both NSC and kernel PNMF is $O(N^2r)$ per iteration. We have performed the above experiments using Matlab R2010b software on a personal computer with Intel Core i7 CPU, 8G memory and Ubuntu 10 operating system. For small graphs such as *Korea* and *WorldCities*, the algorithms converged within a few seconds. For larger graphs such as *PolBlogs* and *3Conference*, they converged within ten to fifteen minutes.

5.2. Two-way clustering

Biclustering, coclustering, or two-way clustering is a data mining problem which requires simultaneous clustering of matrix rows and columns. Here we demonstrate an application of QNMF for finding biclusters in which the matrix entries have similar values. A good biclustering of this kind should

generate a blockwise visualization of the matrix when the rows and columns are ordered by their bicluster indices.

Two-way clustering has previously been addressed by the linear NMF methods (e.g. [8]). Given the factorization of the input matrix $\mathbf{X} \approx \mathbf{WH}$, the bicluster index for each row of $\mathbf{X}$ is determined by the index of maximal entry in each row of $\mathbf{W}$. The bicluster index for columns are likewise obtained. The biclustering problem has also been attacked by three-factor linear NMF $\mathbf{X} \approx \mathbf{LSR}^T$ with the orthogonality constraint on $\mathbf{L}$ and $\mathbf{R}$. For tri-factorizations, Ding et al. [16] gave a multiplicative algorithm called BiOR-NM3F when the approximation error is measured by the squared Euclidean distance. However, when this method is extended to other divergences such as the I-divergence, it is often stuck in trivial local minima where $\mathbf{S}$ tends to be smooth or even uniform because of the sparsity of $\mathbf{L}$ and $\mathbf{R}$ [15]. An extra constraint on $\mathbf{S}$ is therefore needed.

Here we propose to use a QNMF formulation for the biclustering problem: $\mathbf{X} \approx \mathbf{L}\mathbf{L}^T\mathbf{X}\mathbf{R}\mathbf{R}^T$. Compared with the BiOR-NM3F approximation, here we constrain the middle factorizing matrix $\mathbf{S}$ to be $\mathbf{L}^T\mathbf{X}\mathbf{R}$. The resulting two-sided QNMF objectives can be optimized by alternating the one-sided algorithms, that is, interleaving optimizations of $\mathbf{X} \approx \mathbf{L}\mathbf{L}^T\mathbf{Y}^{(R)}$ with $\mathbf{Y}^{(R)} = \mathbf{X}\mathbf{R}\mathbf{R}^T$ fixed and $\mathbf{X} \approx \mathbf{Y}^{(L)}\mathbf{R}\mathbf{R}^T$ with $\mathbf{Y}^{(L)} = \mathbf{L}\mathbf{L}^T\mathbf{X}$ fixed. The bicluster indices of rows and columns are given by taking the maximum of each row in $\mathbf{L}$ and $\mathbf{R}$. We call the new method *Biclustering QNMF* (Bi-QNMF) or *Two-way QNMF*.

We have compared Bi-QNMF with linear NMF based on the I-divergence and BiOR-NM3F for two-way clustering on both synthetic and real-world

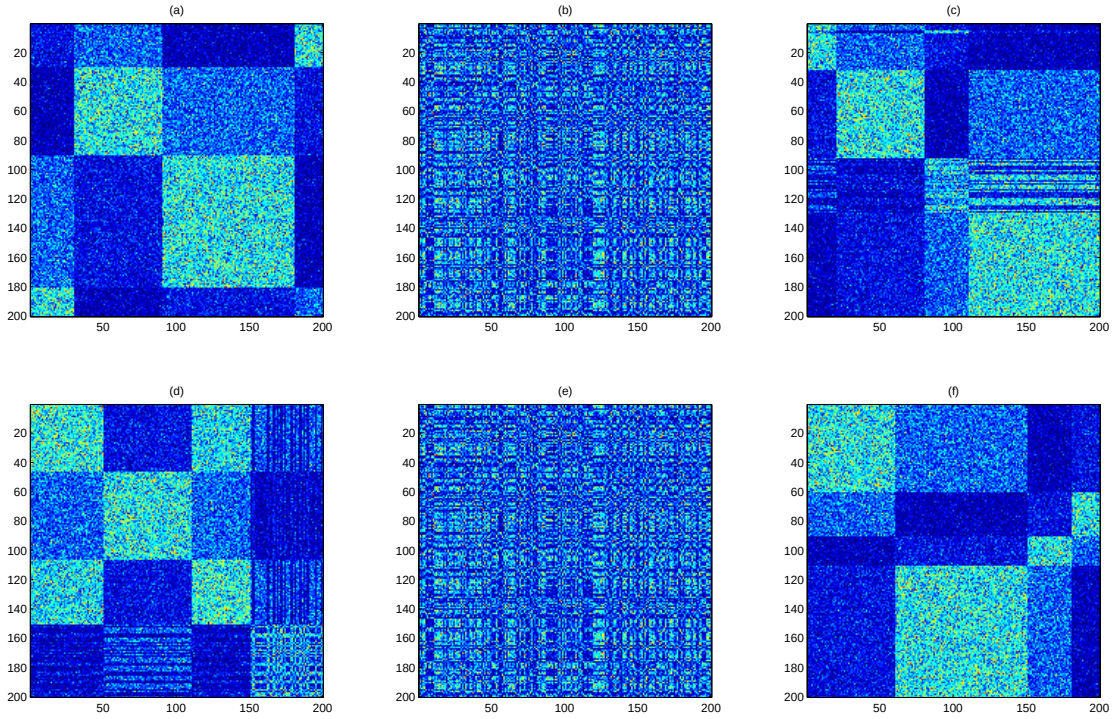

Figure 2: Biclustering using α-NMF and α-PNMF based on the I-divergence for the synthetic data: (a) the original matrix, (b) the disordered or testing matrix. The biclustering results use (c) two-factor linear NMF, (d) BiOR-NM3F, (e) BiOR-NM3F-I and (f) Bi-QNMF.

data. To our knowledge, BiOR-NM3F is implemented only for the squared Euclidean distance. For comparison, we apply the same development procedure by [16] (see also Section 4.2) to obtain the multiplicative update rules for the I-divergence, which we denote by BiOR-NM3F-I.

Firstly, a 200×200 blockwise nonnegative matrix is generated, where each block has dimensions 20, 30, 60 or 90. The matrix entries in a block are randomly drawn from the same Poisson distribution whose mean is chosen

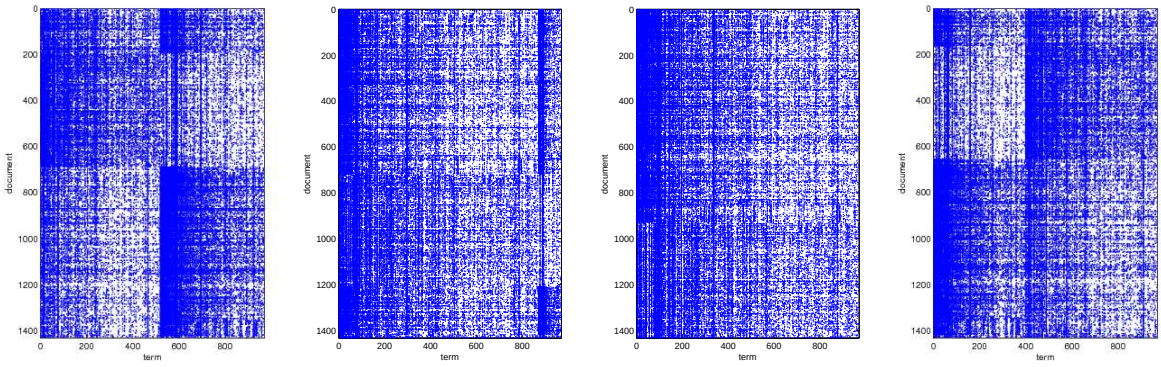

Figure 3: Biclustering using (from left to right) two-factor linear NMF, BiOR-NM3F, BiOR-NM3F-I, and Bi-QNMF for the *webkb* data.

from 1, 2, 4, or 7. The resulting generated matrix is visualized in Figure 2 (a) by using the Matlab command *imagesc*. The testing matrix is then obtained by randomly re-ordering the rows and columns of the original matrix. The biclustering task is to recover groups of rows and columns. With the learned factorizing matrices and corresponding bicluster indices, we reordered the rows and columns of the disordered matrix by ascending indices.

We ran each compared algorithm ten times and picked the result with the smallest NMF approximation error. The resulting matrices are displayed in Figure 2 (c)-(f). It can be seen that the two-factor linear NMF method finds some but not all biclusters, especially the ones with a few rows. BiOR-NM3F performs even worse than the two-factor linear NMF, with the block sizes totally mismatched and many blocks containing dissimilar values. The failure of BiOR-NM3F could be caused by the wrong divergence type. We thus also compared the result by using BiOR-NM3F-I. However, the extended BiOR-NM3F generates identical columns in $\mathbf{L}$ and $\mathbf{R}$, as well as a uniform $\mathbf{S}$, which

results in still disordered visualization. Compared to the above methods, Bi-QNMF can well reconstruct all biclusters up to a block-wise permutation.

Next, we compared the four methods on the real-world *webkb* dataset[6]. The data matrix contains a subset of the whole dataset, with two classes of 1433 documents and 933 terms. The ij-th entry of the matrix is the number of the j-th term that appears in the i-th document. Same as the tests on synthetic data, we reordered the matrix rows and columns by using the learned factorizing matrices of the compared methods. The resulting matrices are visualized in Figure 3 using the Matlab command *spy*.

The BiOR-NM3F method basically finds no biclusters. Only some small groups can barely be seen on the rightmost of the display. BiOR-NM3F-I is even worse, as there is no visually blockwise pattern in its result. The row cluster sizes found by the two-factor linear NMF and Bi-QNMF are roughly the same, about 650 rows for the first cluster and the rest for the second. However, linear NMF results in incoherent blocks, where the upper rows in the first row cluster are quite different from the others but very similar to the ones in the second row cluster. Such incoherence does not take place in the Bi-QNMF result, where the matrix is clearly divided into 2×2 biclusters.

Suppose the input matrix is of size $m \times n$. The computational cost of BiOR-NM3F, BiOR-NM3F-I and Bi-QNMF is $O(mn \cdot \max\{r_l, r_r\})$ per iteration, where r_l and r_r are number of columns of $\mathbf{L}$ and $\mathbf{R}$. For synthetic data, all NMF methods finish 20,000 iterations (converged) within two minutes. For the larger *webkb* dataset, the NMF biclustering algorithms converged within two hours.

5.3. Estimating hidden Markov chains

In a stationary *Hidden Markov Chain Model* (HMM), the observed output and the hidden state at time t are denoted by $x(t) \in \{1, \ldots, n\}$ and $y(t) \in \{1, \ldots, r\}$, respectively. The joint probabilities of a consecutive pair are then given by $X_{ij} \triangleq P(x(t) = i, x(t+1) = j)$ and $Y_{kl} \triangleq P(y(t) = k, y(t+1) = l)$ accordingly. For the noiseless model, we have $\mathbf{X} = \mathbf{WYW}^T$ with $\mathbf{W} \triangleq P(x(t) = i | y(t) = k)$. When noise is considered, this becomes an approximative QNMF problem $\mathbf{X} \approx \mathbf{WYW}^T$. Particularly, when the approximation error is measured by squared Euclidean distance, the parameter estimation problem of HMM can be formulated as minimization of $\|\mathbf{X} - \mathbf{WYW}^T\|_F^2$ over nonnegative $\mathbf{W}$ and $\mathbf{Y}$ and subject to $\sum_i W_{ik} = 1$ for all k and $\sum_{kl} Y_{kl} = 1$.

Previously, the above optimization problem has been difficult because the objective is quartic with respect to $\mathbf{W}$. An earlier algorithm, named NNMF-HMM [35], interleaves minimizations over $\mathbf{Y}$ and one of the appearances of $\mathbf{W}$, each of which is implemented by using matrix pseudo-inversion and truncation of negative entries.

Now we have a much nicer algorithm for this QNMF problem. Using the presented deriving principle, we can obtain the multiplicative update rule in the form of Eq. (10), where $\nabla^-_{\mathbf{W}} = \mathbf{XWY}^T + \mathbf{X}^T\mathbf{WY}$, $\nabla^+_{\mathbf{W}} = \mathbf{WYW}^T\mathbf{WY}^T + \mathbf{WY}^T\mathbf{W}^T\mathbf{WY}$, $\nabla^-_{\mathbf{Y}} = \mathbf{W}^T\mathbf{XW}$, and $\nabla^+_{\mathbf{Y}} = \mathbf{W}^T\mathbf{WYW}^T\mathbf{W}$.

We have compared the two methods on three datasets. The first is a synthetic sequence generated by using the procedure by Lakshminarayanan and Raich [35]. The other two are real-world datasets, one for letter sequence in the top 58112 English words[7] and the other for a genetic code sequence of

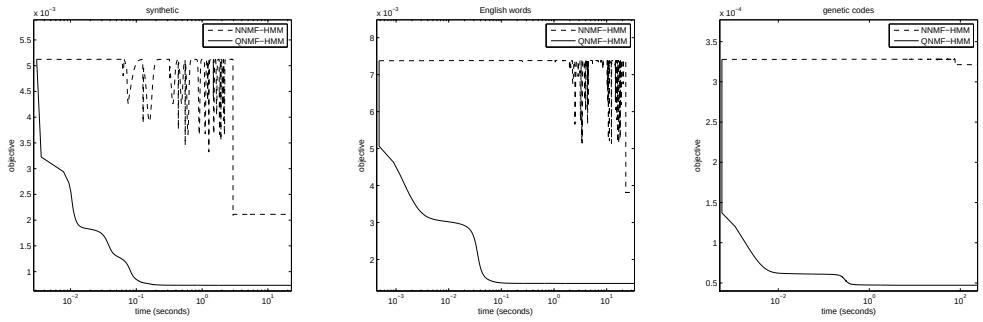

Figure 4: Evolutions of the HMM estimation objective.

homo sapiens[8]. The sizes of the observed input matrix are 22×22, 26×26, and 64×64, respectively. We empirically set the number of hidden states in all experiments.

Both compared algorithms were run at least 10000 iterations. The evolutions of HMM estimation objective are shown in Figure 4. The new method is significantly more efficient than the old algorithm for all selected datasets. For the two smaller datasets, the objectives using NNMF-HMM tend to fluctuate a lot during the iterations, which is possibly caused by the brute-force truncation of negative entries. NNMF-HMM is more problematic for the largest dataset, where the approximation error only barely drops after thousands of epochs. By contrast, the objectives using QNMF-HMM decrease much faster and more stably.

QNMF-HMM uses soft constraints during its learning. To verify that the constraint errors are trivial in the converged results, we have checked the quantities $\sum_k |1 - \sum_i W_{ik}|$ and $|1 - \sum_{kl} Y_{kl}|$ as constraint error measures for $\mathbf{W}$ and $\mathbf{Y}$, respectively. We repeated each experiment 100 times with different random initial guess of $\mathbf{W}$ and $\mathbf{Y}$. The recorded means and standard

Table 2: Constraint errors using QNMF-HMM.

	synthetic	English words	genetic codes
W	8.9e-06±3.8e-05	3.3e-05±3.5e-05	6.1e-06±6.1e-06
Y	1.7e-08±1.2e-07	2.2e-07±5.6e-07	2.7e-07±2.2e-07

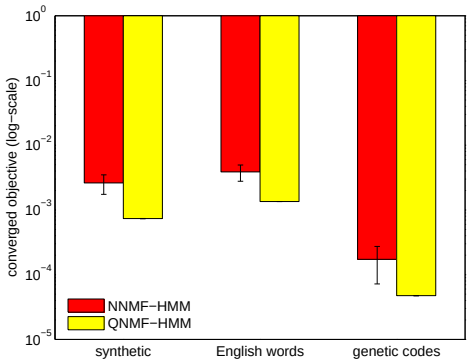

Figure 5: HMM estimation objectives for the selected datasets.

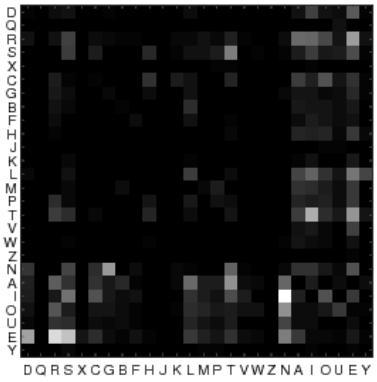

Figure 6: Reordered joint probability matrix by the letter clusters.

deviations of the above constraint errors are shown in Table 2. We can see that the constraint errors are so small that they are negligible compared to the data dimensions.

We also examined the final objectives using the compared methods. For QNMF-HMM, we normalized $\mathbf{W}$ in the end of learning to force the strict constraints. The normalization actually causes little loss because of the negligible constraint errors. The error bars of the final objectives are illustrated in Figure 5, from which we can see that the new method can achieve much smaller approximation errors than the old one. In addition, the variations of QNMF-HMM results are very small, which indicates the our method is pretty robust.

The squared Euclidean distance in the HMM estimation can be replaced with the Kullback-Leibler divergence, which is a more canonical difference measure for probabilities. With such divergence we can reveal some interesting structure in the fitted model of the *English words* data. Here we only focus on the QNMF-HMM algorithm because there is no existing implementation or straightforward extension of NNMF-HMM for Kullback-Leibler divergence.

The latent or blockwise structure of the joint probability matrix $\mathbf{X}$ is visualized in Figure 6 by reordering the rows and columns according to the letter clusters, where lighter dots correspond to larger values. Each English letter was assigned to one of six clusters according to largest entries in the learned $\mathbf{W}$ rows. In this way we can group the letters into the following six groups: (1) D, Q, R, S, X, (2) C, G, (3) B, F, H, J, K, L, M, P, T, V, W, Z, (4) N, (5) A, I, O, U, and (6) E, Y. The first four groups are consonants,

Table 3: Errors (mean±standard deviation) in directed graph matching.

Graph	size	EU-QNMF	I-QNMF
SAMPIN	18	1.20±0.38	**0.00±0.00**
WOLFK	20	**1.80±0.30**	2.10±0.07
football35	35	**7.20±2.28**	8.20±0.81
cage5	37	**9.20±1.03**	12.80±0.53
BKOFFC	40	**0.00±0.00**	0.00±0.00
curtis54	54	96.40±5.38	**56.60±2.63**
will57	57	87.60±3.24	**35.80±1.36**
PRISON	67	**4.40±0.45**	6.70±0.37
CAG_mat72	72	0.20±0.06	**0.00±0.00**
GlossGT	72	**4.60±0.67**	4.80±0.19

while the remaining two are basically vowels. A plausible interpretation of such division is that English words consist of syllables where a consonant is often followed by a vowel and vice versa. The letter N itself forms a group probably because we used a pretty large list of English words where suffixes such as *ING* and *TION* occur frequently.

5.4. Graph matching

Given a graph and its node-permuted version, the *graph matching* problem is to find the correspondence of graph nodes. If the graph and its permutation are represented by (weighted) adjacency matrices, denoted by $\mathbf{A}$ and $\mathbf{B}$, the isomorphism can be written as $\mathbf{B} = \mathbf{PAP}^T$, where $\mathbf{P}$ is a permutation matrix.

The computation time of existing combinatorial algorithms to find the exact correct permutation grows exponentially as the number of nodes increases. For large graphs, approximation is therefore needed. A classical approximation algorithm by Umeyama [36] uses eigendecomposition and simply forces nonnegativity by using absolute values of the eigenvectors. Ding et al. [18] pointed out that the results of Umeyama's algorithm are not satisfactory, and proposed a special case of QNMF formulation based on the Euclidean distances: to minimize $\|\mathbf{B}-\mathbf{WAW}^T\|_F^2$ subject to $\mathbf{W}^T\mathbf{W}=\mathbf{I}$ and $\mathbf{W}\geq 0$. After the multiplicative updates have converged, the permutation $\mathbf{P}$ is obtained by using the classical Hungarian algorithm on elementwise inverse of $\mathbf{W}^T$ [37]. This method, abbreviated by EU-QNMF, was reported to be superior to Umeyama's algorithm on *dense* graphs generated as follows: $A_{ij}=100r_{ij}$ and $B_{ij}=(\mathbf{P}_t\mathbf{AP}_t^T)_{ij}(1+\epsilon s_{ij})$, where r_{ij} and s_{ij} are uniform random numbers in [0, 1], $\mathbf{P}_t$ is a permutation and ϵ sets the noise level. However, many real-world networks are not generated like this. They should be *sparse* and thus the Gaussian assumption that underlies the Euclidean-NMF does not hold. We are thus motivated to use the I-divergence, i.e. to minimize $D_{\mathrm{I}}(\mathbf{B}||\mathbf{WAW}^T)$ over $\mathbf{W}\geq 0$ and subject to $\mathbf{W}^T\mathbf{W}=\mathbf{I}$. This corresponds to underlying Poisson distribution which is more suitable for sparse occurrences. We abbreviate the new method by I-QNMF.

The comparison results of EU-QNMF and I-QNMF for graph matching are shown in Tables 3 and 4 for directed and undirected graphs, respectively. All graphs in use are binary-valued. Most graphs were downloaded from the University of Florida Sparse Matrix Collection[9]. Some others were taken from Newman's network data collection[3] and from the Pajek datasets[4]. For fair

Table 4: Errors (mean±standard deviation) in undirected graph matching.

Graph	size	EU-QNMF	I-QNMF
RDPOS	14	4.80±0.41	**4.20±0.27**
PADGB	16	**9.20±0.98**	10.20±0.64
PADGM	16	**5.60±0.66**	6.20±0.40
WOLFN	20	3.20±0.59	**2.00±0.34**
strike	24	25.60±1.36	**12.20±1.02**
ZACHC	34	43.20±4.76	**10.40±0.08**
ZACHE	34	27.60±2.97	**5.00±0.36**
korea1	35	39.20±2.72	**12.40±0.82**
korea2	35	29.60±1.78	**12.80±0.50**
mexican_power	35	20.00±5.25	**10.80±0.62**
Sawmill	36	70.40±2.14	**16.00±0.69**
dolphins	62	137.80±8.31	**31.20±1.17**

comparison, we used a neutral quantity to measure the matching error, which is defined as $\#\left\{\mathrm{XOR}(\mathbf{PAP}^T, \mathbf{B})\right\}$, i.e. the number of different edges between the estimated permuted matrix and the true permuted matrix. A smaller error indicates better matching quality. In summary, I-QNMF performs almost at the same level as EU-QNMF for small or easy-matched graphs. However, the former is significantly better for larger and more difficult graphs in terms of both smaller mean errors and smaller deviations. This advantage is even clearer for undirected graphs.

The computational cost of both NMF graph matching algorithms is $O(N^3)$ for a graph with N nodes. For small graphs such as *SAMPIN* and *strike*,

the algorithms can finish 20,000 iterations (converged) within a few seconds. For larger graphs such as *GlossGT* and *dolphins*, the algorithms converged within about ten minutes.

6. Conclusions

We have formulated the general problem of approximative quadratic nonnegative matrix factorization and proposed a framework to develop multiplicative optimization algorithms that are guaranteed to decrease the objective function. Multiplicative algorithms for two typical QNMF factorization forms based on a great variety of approximation error measures, as well as the stochasticity and orthogonality constraints, were presented. The proposed method has been applied to four different pattern recognition problems, where QNMF is shown to be more advantageous than other state-of-the-art NMF methods.

Our work opens the door to higher-order nonnegative matrix factorizations, but this is definitely not the end of this stream of research. For optimization, recently some faster additive methods for fast optimization of NMF objectives have been developed (e.g. [38, 39, 5]). However, the speedup is achieved for only limited kinds of divergences, especially the squared Euclidean distance where the Hessian has a particularly simple form. Otherwise, these additive approaches could be sensitive to the choice of extra parameters such as learning rate or line search step. How to extend these methods to many other divergences remains an open problem. Alternatively, we could implant efficient approximative second-order optimization techniques into the multiplicative updates for faster convergence. In addition, scalable QNMF

algorithms for more divergence types other than squared Euclidean distance could be developed using advanced streaming or distributed computation techniques. Moreover, we have found that many QNMF multiplicative algorithms converge to similar local minima up to a certain component order. The theoretical analysis of the global minimum up to certain permutations needs further investigation.

Besides matrix products, the factorized approximation may involve nonlinear operators, for example, a nonlinear activation function that interleaves a factorizing matrix and its transpose. This kind of approximation could be extended to the field of nonnegative neural networks and connected to the deep learning principle when multiple such groups of elements are stacked.

Appendix A. Proof of Theorem 2

Decompose $\boldsymbol{\lambda}$ into two nonnegative parts: $\boldsymbol{\lambda} = \boldsymbol{\lambda}^+ - \boldsymbol{\lambda}^-$, where $\lambda_k^+ = (|\lambda_k| + \lambda_k)/2$ and $\lambda_k^- = (|\lambda_k| - \lambda_k)/2$. We can then write the augmented objective as $\mathcal{L}(\widetilde{\mathbf{W}}, \boldsymbol{\lambda}) = \mathcal{J}(\widetilde{\mathbf{W}}) + \sum_k \lambda_k^+ \left(1 - \sum_i \widetilde{W}_{ik}\right) - \sum_k \lambda_k^- \left(1 - \sum_i \widetilde{W}_{ik}\right)$. Next, we apply the original upper-bounding on $\mathcal{J}(\widetilde{\mathbf{W}})$: $\mathcal{L}(\widetilde{\mathbf{W}}, \boldsymbol{\lambda}) \leq G(\widetilde{\mathbf{W}}, \mathbf{W}) + \sum_k \lambda_k^+ \left(1 - \sum_i \widetilde{W}_{ik}\right) - \sum_k \lambda_k^- \left(1 - \sum_i \widetilde{W}_{ik}\right)$. Combining individual upper-bounds into two, we obtain the ultimate auxiliary function:

$$\widetilde{G}(\widetilde{\mathbf{W}}, \mathbf{W}) = \sum_{ik} \frac{W_{ik}}{u} \left(\frac{\widetilde{W}_{ik}}{W_{ik}}\right)^u \left(\nabla_{ik}^+ + \lambda_k^-\right) - \sum_{ik} \frac{W_{ik}}{v} \left(\frac{\widetilde{W}_{ik}}{W_{ik}}\right)^v \left(\nabla_{ik}^- + \lambda_k^+\right),$$

up to an additive constant, where $u = \max(\psi_{\max}, 1)$ and $v = \min(\psi_{\min}, 1)$. Setting $\partial \widetilde{G} / \partial \widetilde{W}_{ik} = 0$ leads to

$$W_{ik}^{\text{new}} = W_{ik} \left[\frac{\nabla_{ik}^- + \lambda_k^+}{\nabla_{ik}^+ + \lambda_k^-}\right]^\sigma \tag{A.1}$$

with $\sigma = 1/(u-v)$. From $\partial\mathcal{L}(\widetilde{\mathbf{W}},\boldsymbol{\lambda})/\partial\widetilde{W}_{ik} = 0$, we get $(\nabla^+ - \nabla^-)_{ik} - \lambda_k = 0$. By multiplying both sides with $\sum_i W_{ik}$ and making use of the fact $\partial\mathcal{L}(\mathbf{W},\boldsymbol{\lambda})/\partial\lambda_k = 0$, i.e. $\sum_i W_{ik} = 1$, we obtain $\lambda_k = \sum_i W_{ik}\nabla^+_{ik} - \sum_i W_{ik}\nabla^-_{ik}$. Inserting this to Eq. (A.1), the multiplicative update rule becomes Eq. (10).

Appendix B. Proof of Theorem 3 and Corollary 4

Similar to the proof of Theorem 2, we decompose $\boldsymbol{\Lambda}$ into two nonnegative parts: $\boldsymbol{\Lambda} = \boldsymbol{\Lambda}^+ - \boldsymbol{\Lambda}^-$. We can then construct the auxiliary function of the generalized objective:

$$\begin{aligned}
\mathcal{L}(\widetilde{\mathbf{W}},\boldsymbol{\Lambda}) =& \mathcal{J}(\widetilde{\mathbf{W}}) - \mathrm{Tr}\left(\boldsymbol{\Lambda}^+\mathbf{W}^T\mathbf{W}\right) + \mathrm{Tr}\left(\boldsymbol{\Lambda}^-\mathbf{W}^T\mathbf{W}\right) + \text{constant} \\
\leq& \mathcal{J}(\widetilde{\mathbf{W}}) - \sum_{ikl}\Lambda^+_{kl}W_{ik}W_{il}\left(1 + \ln\frac{\widetilde{W}_{ik}\widetilde{W}_{il}}{W_{ik}W_{il}}\right) \\
&+ \sum_{ik}\frac{\left(\mathbf{W}\boldsymbol{\Lambda}^-\right)_{ik}\widetilde{W}^2_{ik}}{W_{ik}} + \text{constant} \\
\leq& \sum_{ik}\frac{W_{ik}}{u}\left(\frac{\widetilde{W}_{ik}}{W_{ik}}\right)^u\left(\nabla^+ + 2\mathbf{W}\boldsymbol{\Lambda}^-\right)_{ik} \\
&- \sum_{ik}\frac{W_{ik}}{v}\left(\frac{\widetilde{W}_{ik}}{W_{ik}}\right)^v\left(\nabla^- + 2\mathbf{W}\boldsymbol{\Lambda}^+\right)_{ik} + \text{constant} \triangleq \widetilde{G}(\widetilde{W},W),
\end{aligned}$$

where $u = \max(\psi_{\max}, 2)$ and $v = \min(\psi_{\min}, 0)$. Setting $\partial\widetilde{G}/\partial\widetilde{W}_{ik} = 0$ gives

$$W^{\text{new}}_{ik} = W_{ik}\left[\frac{\left(\nabla^- + 2\mathbf{W}\boldsymbol{\Lambda}^+\right)_{ik}}{\left(\nabla^+ + 2\mathbf{W}\boldsymbol{\Lambda}^-\right)_{ik}}\right]^\sigma \tag{B.1}$$

with $\sigma = 1/(u-v)$. From $\partial\mathcal{L}(\mathbf{W},\boldsymbol{\Lambda})/\partial\mathbf{W} = \mathbf{0}$, we get $2\mathbf{W}\boldsymbol{\Lambda} = \nabla^+ - \nabla^-$. Using $\partial\mathcal{L}(\mathbf{W},\boldsymbol{\Lambda})/\partial\boldsymbol{\Lambda} = \mathbf{0}$, i.e. $\mathbf{W}^T\mathbf{W} = \mathbf{I}$, one obtains $\boldsymbol{\Lambda} = \frac{1}{2}\mathbf{W}^T\left(\nabla^+ - \nabla^-\right)$. Inserting this to (B.1), the multiplicative update rule becomes Eq. (11).

The proof of Corollary 4 is similar to that of the theorem, except that we replace the bound of $-\mathrm{Tr}\left(\mathbf{\Lambda}^{+}\mathbf{W}^{T}\mathbf{W}\right)$ with a hyper-plane $-2\sum_{ikl}\widetilde{W}_{il}\Lambda^{+}_{lk}+$ constant. The corollary thus results in a larger exponential or step size σ than Theorem 3.

References

[1] D. D. Lee, H. S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (1999) 788–791.

[2] D. D. Lee, H. S. Seung, Algorithms for non-negative matrix factorization, Advances in Neural Information Processing Systems 13 (2001) 556–562.

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

[4] S. Choi, Algorithms for orthogonal nonnegative matrix factorization, in: Proceedings of IEEE International Joint Conference on Neural Networks, 2008, pp. 1828–1832.

[5] D. Kim, S. Sra, I. S. Dhillon, Fast projection-based methods for the least squares nonnegative matrix approximation problem, Statistical Analysis and Data Mining 1 (1) (2008) 38–51.

[6] C. Févotte, N. Bertin, J.-L. Durrieu, Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis, Neural Computation 21 (3) (2009) 793–830.

[7] S. Behnke, Discovering hierarchical speech features using convolutional non-negative matrix factorization, in: Proceedings of IEEE International Joint Conference on Neural Networks, Vol. 4, 2003, pp. 2758–2763.

[8] H. Kim, H. Park, Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis, Bioinformatics 23 (12) (2007) 1495–1502.

[9] A. Cichocki, R. Zdunek, Multilayer nonnegative matrix factorization using projected gradient approaches, International Journal of Neural Systems 17 (6) (2007) 431–446.

[10] C. Ding, T. Li, W. Peng, On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing, Computational Statistics and Data Analysis 52 (8) (2008) 3913–3927.

[11] A. Cichocki, R. Zdunek, A.-H. Phan, S. Amari, Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis, John Wiley, 2009.

[12] R. Kompass, A generalized divergence measure for nonnegative matrix factorization, Neural Computation 19 (3) (2006) 780–791.

[13] I. S. Dhillon, S. Sra, Generalized nonnegative matrix approximations with Bregman divergences, in: Advances in Neural Information Processing Systems, Vol. 18, 2006, pp. 283–290.

[14] A. Cichocki, H. Lee, Y.-D. Kim, S. Choi, Non-negative matrix factorization with α-divergence, Pattern Recognition Letters 29 (2008) 1433–1440.

[15] A. Pascual-Montano, J. M. Carazo, K. Kochi, D. Lehmann, R. D. Pascual-Marqui, Nonsmooth nonnegative matrix factorization (nsNMF), IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (3) (2006) 403–415.

[16] C. Ding, T. Li, W. Peng, H. Park, Orthogonal nonnegative matrix t-factorizations for clustering, in: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, 2006, pp. 126–135.

[17] C. Ding, T. Li, M. I. Jordan, Convex and semi-nonnegative matrix factorizations, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (1) (2010) 45–55.

[18] C. Ding, T. Li, M. I. Jordan, Nonnegative matrix factorization for combinatorial optimization: Spectral clustering, graph matching, and clique finding, in: Proceedings of the 8th IEEE International Conference on Data Mining (ICDM), 2008, pp. 183–192.

[19] C. Ding, X. He, K-means clustering via principal component analysis, in: Proceedings of International Conference on Machine Learning (ICML), 2004, pp. 225–232.

[20] Z. Yang, E. Oja, Linear and nonlinear projective nonnegative matrix

factorization, IEEE Transaction on Neural Networks 21 (5) (2010) 734–749.

[21] Z. Yuan, E. Oja, Projective nonnegative matrix factorization for image compression and feature extraction, in: Proceedings of 14th Scandinavian Conference on Image Analysis (SCIA), Joensuu, Finland, 2005, pp. 333–342.

[22] Z. Yang, E. Oja, Unified development of multiplicative algorithms for linear and quadratic nonnegative matrix factorization, IEEE Transaction on Neural NetworksAccepted, to appear.

[23] B. Long, Z. Zhang, P. S. Yu, Coclustering by block value decomposition, in: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, 2005, pp. 635–640.

[24] W. Liu, N. Zheng, X. Lu, Non-negative matrix factorization for visual coding, in: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Vol. 3, 2003, pp. 293–296.

[25] Z. Yang, J. Laaksonen, Multiplicative updates for non-negative projections, Neurocomputing 71 (1-3) (2007) 363–373.

[26] J. Yoo, S. Choi, Orthogonal nonnegative matrix factorization: Multiplicative updates on Stiefel manifolds, in: Proceedings of the 9th International Conference on Intelligent Data Engineering and Automated Learning, 2008, pp. 140–147.

[27] Z. Yang, Z. Yuan, J. Laaksonen, Projective non-negative matrix factorization with applications to facial image processing, International

Journal on Pattern Recognition and Artificial Intelligence 21 (8) (2007) 1353–1362.

[28] E. M. Airoldi, D. M. Blei, S. E. Fienberg, E. P. Xing, Mixed membership stochastic blockmodels, Journal of Machine Learning Research 9 (2008) 1981–2014.

[29] Z. Yang, E. Oja, Online projective nonnegative matrix factorization for large datasets, in: NIPS Workshop on Low-rank Methods for Large-scale Machine Learning, 2010.

[30] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 888–905.

[31] S. X. Yu, J. Shi, Multiclass spectral clustering, in: Proceedings of the 9th IEEE International Conference on Computer Vision, Vol. 2, 2003, pp. 313–319.

[32] X. Zhu, Z. Ghahramani, J. Lafferty, Semi-supervised learning using gaussian fields and harmonic functions, in: Proceedings of the 20th International Conference on Machine Learning (ICML), 2003, pp. 912–919.

[33] L. Grady, Random walks for image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (11) (2006) 1768–1783.

[34] L. van der Maaten, G. Hinton, Visualizing data using t-SNE, Journal of Machine Learning Research 9 (2008) 2579–2605.

[35] B. Lakshminarayanan, R. Raich, Non-negative matrix factorization for parameter estimation in hidden markov models, in: Proceedings of IEEE International Workshop on Machine Learning For Signal Processing, 2010, pp. 89–94.

[36] S. Umeyama, An eigendecomposition approach to weighted graph matching problems, IEEE Transactions on Pattern Analysis and Machine Intelligence 10 (5) (1988) 695–703.

[37] H. W. Kuhn, The Hungarian method for the assignment problem, Naval Research Logistics Quarterly 2 (1955) 83–97.

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

[39] R. Zdunek, A. Cichocki, Nonnegative matrix factorization with quadratic programming, Neurocomputing 71 (10-12) (2008) 2309–2008.