Adaptive multiplicative updates for quadratic nonnegative matrix factorization

He Zhang*, Zhirong Yang, Erkki Oja

Department of Information and Computer Science, Aalto University, Espoo, Finland

Abstract

In Nonnegative Matrix Factorization (NMF), a nonnegative matrix is approximated by a product of lower-rank factorizing matrices. Quadratic Nonnegative Matrix Factorization (QNMF) is a new class of NMF methods where some factorizing matrices occur twice in the approximation. QNMF finds its applications in graph partition, bi-clustering, graph matching, etc. However, the original QNMF algorithms employ constant multiplicative update rules and thus have mediocre convergence speed. Here we propose an adaptive multiplicative algorithm for QNMF which is not only theoretically convergent but also significantly faster than the original implementation. An adaptive exponent scheme has been adopted for our method instead of the old constant ones, which enables larger learning steps for improved efficiency. The proposed method is general and thus can be applied to QNMF with a variety of factorization forms and with the most commonly used approximation error measures. We have performed extensive experiments, where the results demonstrate that the new method is effective in various QNMF applications on both synthetic and real-world datasets.

Keywords:
Adaptive, Multiplicative updates, Quadratic, Nonnegative Matrix Factorization

1. Introduction

Nonnegative Matrix Factorization (NMF) has attracted a lot of research effort in recent years (e.g. [1, 2, 3, 4, 5, 6]). NMF has a variety of applications e.g. in machine learning, signal processing, pattern recognition, data mining, and information retrieval. (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, the approximation is linear with respect to each factorizing matrix; that is, these matrices appear only once in the approximation. However, such a 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 A and B, then they are isomorphic if and only if a permutation matrix P can be found such that $A - PBP^T = 0$. Minimizing the norm or some other suitable error measure of the left-hand side with respect to P, with suitable constraints, reduces the problem to an NMF problem. Note that both adjacency matrices and permutation matrices are nonnegative, and the approximation is now quadratic in P.

Another example is clustering: if 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] $\mathcal{J}_1 = \mathrm{Tr}(X^T X) - \mathrm{Tr}(U^T X^T X U)$ where U is the $(n \times r)$ binary cluster indicator matrix. It was shown in [20] that minimizing $\mathcal{J}_2 = \left\|X^T - WW^T X^T\right\|^2_{\text{Frobenius}}$ with respect to an orthogonal and nonnegative matrix W gives the same solution, except for the binary constraint. This is another NMF problem where the approximation is quadratic in W.

Methods for attacking this kind of problems are called Quadratic Nonnegative Matrix Factorization (QNMF). A systematic study on QNMF was given by Yang and Oja [21], where they presented a unified development method for multiplicative QNMF optimization algorithms.

The original QNMF multiplicative update rules have a fixed form, where an exponent in the multiplying factor in these rules remains the same in all iterations. Despite its simplicity, the constant exponent corresponds to overly conservative learning steps and thus often leads to mediocre convergence speed.

Here we propose new multiplicative algorithms for QNMF to overcome this drawback. We drop the restriction of constant exponent in multiplicative update rules, which relaxes the updates by using variable exponents in different iterations. This turns out to be an effective strategy for accelerating the optimization while still maintaining the monotonical objective decrease. The acceleration for Projective NMF, a special case of QNMF, was presented in our preliminary work [22]. In this paper we demonstrate that the new method can bring improvement for many other QNMF optimizations. We generalize the adaptive multiplicative algorithm for a wide variety of

*Corresponding author.
Email addresses: he.zhang@aalto.fi (He Zhang), zhirong.yang@aalto.fi (Zhirong Yang), erkki.oja@aalto.fi (Erkki Oja)

QNMF problems. In addition to Projective Nonnegative Matrix Factorization, we also apply the strategy on two other special cases of QNMF, with corresponding application scenarios in biclustering and estimation of hidden Markov chains. Extensive empirical results on both synthetic and real-world data justify the efficiency advantage by using our method.

In the following, Section 2 recapitulates the essence of the QNMF objectives and their previous optimization methods. Section 3 presents the fast QNMF algorithm by using adaptive exponents. In Section 4, we provide empirical comparison between the new algorithm and the original implementation on three applications of QNMF. Section 5 concludes the paper.

2. Quadratic Nonnegative Matrix Factorization

Nonnegative Matrix Factorization (NMF) finds an approximation $\widehat{X}$ to an input data matrix $X \in \mathbb{R}^{m\times n}$:

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

and some of these matrices are constrained to be nonnegative. The dimensions of the factorizing matrices $F^{(1)}, \ldots, F^{(Q)}$ 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 smaller than m or n.

In most conventional NMF approaches, the factorizing matrices $F^{(q)}$ in Eq. (1) are all different, and thus the approximation $\widehat{X}$ 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, $F^{(s)} = F^{(t)^T}$ for a number of non-overlapping pairs $\{s, t\}$ and $1 \le s < t \le Q$. We call such a problem and its solution *Quadratic Nonnegative Matrix Factorization* (QNMF) because $\widehat{X}$ as a function is quadratic to each twice appearing factorizing matrix[1].

When there is only one doubly occurring matrix W in the QNMF objective, the general approximating factorization form is given by [21]

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

where the products of the other, linearly appearing factorizing matrices are merged into single symbols. QNMF focuses on the optimization over W, while learning the matrices that occur only once can be solved by using the conventional NMF methods of alternative optimization over each matrix separately [11].

As stated by the authors in [21], The factorization form unifies many previously suggested QNMF objectives. For example, it becomes the *Projective Nonnegative Matrix Factorization* (PNMF) when $A = B = I$ and $C = X$ [23, 24, 25, 20, 17]. If X is a square matrix and $A = C = I$, the factorization can be used in two major scenarios if the learned W is highly orthogonal: (1) When B is much smaller than X, the three-factor approximation corresponds to a blockwise representation of X [26, 27]. If B is diagonal, then the representation becomes diagonal blockwise, or a partition. In the extreme case $B = I$, the factorization reduces to the *Symmetric Nonnegative Matrix Factorization* (SNMF) $\widehat{X} = WW^T$ [16]. (2) When X and B are of the same size, the learned W with the constraint $W^T W = I$ 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 W has columnwise unitary sums, the solution of such a QNMF problem provides parameter estimation of hidden Markov chains (See Section 4.3).

The factorization form in Eq. (2) can be recursively applied to the cases where there are more than one factorizing matrices appearing quadratically in the approximation. For example, the case $A = C^T = U$ yields $\widehat{X} = UWBW^T U^T$, and $A = B = I$, $C = XUU^T$ yields $\widehat{X} = WW^T XUU^T$. An application of the latter example is shown in Section 4.2, where the solution of such a QNMF problem can be used to group the rows and columns of 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 [21]. In 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 harder because matrices in two places vary simultaneously, which leads to higher-order objectives. For example, given the squared Frobenius norm (Euclidean distance) as approximation error measure, the objective of linear NMF $\|X - WH\|_F^2$ is quadratic with respect to W and H, whereas the PNMF objective $\|X - WWX\|_F^2$ is quartic with respect to W. Minimizing such a fourth-order objective with the nonnegativity constraint is considerably more challenging than minimizing a quadratic function.

3. Adaptive QNMF

The difference between the input matrix X and its approximation $\widehat{X}$ can be measured by a variety of divergences. Yang and Oja [21] presented a general method for developing optimization algorithms with multiplicative updates for QNMF based on α-divergence, β-divergence, γ-divergence or Rényi divergence. These families include the most popular used NMF objectives, 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 $r \to 1$). Multiplicative update rules can be developed for more QNMF objectives, for example the additive hybrids of the above divergences, as well as many other unnamed Csiszár divergences and Bregman divergences.

In general, the multiplicative update rules take the following

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

Table 1: Notations in the multiplicative update rules of QNMF examples, where $\widehat{X} = AWBW^TC$. Here α, β, γ, and r stand for *alpha*-, β-, γ- and Rényi-divergences, respectively.

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

form:

$$W_{ik}^{\text{new}} = W_{ik}\left[\frac{\left(A^TQC^TWB^T + CQ^TAWB\right)_{ik}}{(A^TPC^TWB^T + CP^TAWB)_{ik}}\cdot\theta\right]^{\eta}, \tag{3}$$

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

$$W_{ik}^{\text{new}} = W_{ik}\left[\frac{\left(XWB^T + X^TWB\right)_{ik}}{(WBW^TWB^T + WB^TW^TWB)_{ik}}\right]^{1/4}. \tag{4}$$

Such multiplicative learning rules have the essential advantage that matrix W always stays nonnegative. Note especially the role of the exponent η in the algorithms. The value given in the Table guarantees that the objective function will be non-increasing in the iteration, and hence it will converge to a local minimum. Multiplicative algorithms using an update rule with a constant exponent in all iterations are simple to implement.

However, the exponent η also strongly affects the convergence speed. It may be that a constant value as given in the Table is unnecessarily small in practice. The convergence can be accelerated by using a more aggressive choice of the exponent, which adaptively changes during the iterations. A straightforward 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. The pseudo-code for such an implementation is given in Algorithm 1, where $D(X\|\widehat{X})$, A, B and η are defined according to the type of cost function (Euclidean distance or I-divergence). We have empirically used $\mu = 0.1$ in all related experiments in this work. Although more comprehensive adaptation approaches could be applied, we find that such a simple strategy can already significantly speed up the convergence while still maintaining the monotonicity of updates.

Algorithm 1 Multiplicative Updates with Adaptive Exponent for QNMF

Usage: $W \leftarrow \text{FastQNMF}(X, \eta, \mu)$.
Initialize W; $\rho \leftarrow \eta$.
repeat
$$U_{ik} \leftarrow W_{ik}\left[\frac{\left(A^TQC^TWB^T + CQ^TAWB\right)_{ik}}{(A^TPC^TWB^T + CP^TAWB)_{ik}}\cdot\theta\right]^{\rho}$$
if $D(X\|AUBU^TC) < D(X\|AWBW^TC)$ **then**
$W \leftarrow U$
$\rho \leftarrow \rho + \mu$
else
$\rho \leftarrow \eta$
end if
until convergent conditions are satisfied

4. Experiments

4.1. Projective Nonnegative Matrix Factorization

We have selected eight datasets that are commonly used in machine learning for our experiments. These datasets were obtained from the UCI repository[2], the University of Florida Sparse Matrix Collection[3], and the LSI text corpora[4], as well as other publicly available websites. The statistics of the datasets are summarized in Table 2.

Table 2: Datasets used in the PNMF experiments.

	Dimensions	#Samples
`wine`	13	178
`mfeat`	292	2000
`orl`	10304	400
`feret`	1024	2409
`swimmer`	1024	256
`cisi`	1460	5609
`cran`	1398	4612
`med`	1033	5831

Projective Nonnegative Matrix Factorization (PNMF) is a special case of QNMF. When squared Euclidean distance is used as the error measure, PNMF solves the following optimization problem:

$$\underset{W\geq 0}{\text{minimize}} \quad \|X - WW^TX\|_F^2, \tag{5}$$

where $\|A\|_F^2 = \sum_{ij}A_{ij}^2$. The original PNMF algorithm using constant exponent iteratively applies the following update rule [28, 21]

$$W_{ik} \leftarrow W_{ik}\left[\frac{2\left(XX^TW\right)_{ik}}{(WWXX^TW + XX^TWW^TW)_{ik}}\right]^{1/3} \tag{6}$$

The fast adaptive PNMF algorithm uses variable exponents (Algorithm 1) instead of the constant $1/3$.

Figure 1 shows the objective function evolution curves using the compared methods for eight selected datasets. One can see that the dashed lines are clearly below the solid ones in the plots, which indicates that the adaptive alogorithm is significantly faster than the original implementation.

[2] `http://archive.ics.uci.edu/ml/`

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

[4] `http://www.cs.utk.edu/\~lsi/corpa.html`

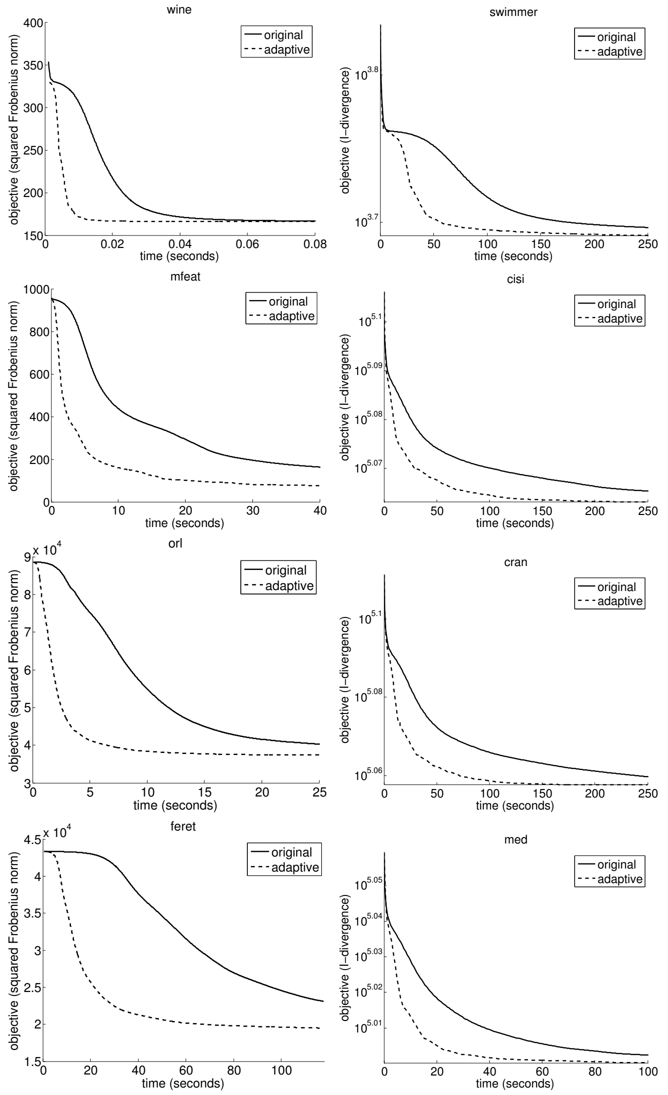

Figure 1: Evolutions of objectives using the compared methods based on (left) squared Euclidean distance and (right) I-divergence.

In addition to qualitative analysis, we have also compared the benchmark on convergence time of the three methods. Table 3 summarizes the means and standard deviations of the resulting convergence time. The convergence time is calculated at the earliest iteration where the objective D is sufficiently close to the minimum D^*, i.e. $|D - D^*|/D^* < 0.001$. Each algorithm on each dataset has been repeated 100 times with different random seeds for initialization. These quantitative results confirm that the adaptive algorithm is significantly faster: it is 3 to 5 times faster than the original method.

4.2. *Two-way clustering*

Biclustering, coclustering, or two-way clustering is a data mining problem which requires simultaneous clustering of matrix rows and columns. QNMF can be applied for finding biclusters in which the matrix entries have similar values. A good biclustering of this kind should be able to discover a blockwise structure in the input matrix when the rows and columns are ordered by their bicluster indices.

Table 3: The mean (μ) and standard deviation (σ) of the convergence time (seconds) of PNMF using the compared algorithms.

(a) PNMF based on Euclidean distance

dataset	original	adaptive
`wine`	0.22±0.11	0.06±0.03
`mfeat`	68.57±1.75	19.10±0.70
`orl`	117.26±1.74	29.89±1.48
`feret`	107.58±24.43	19.97±5.60

(b) PNMF based on I-divergence

dataset	original	adaptive
`swimmer`	613.04±20.63	193.47±5.43
`cisi`	863.89±69.23	193.23±18.70
`cran`	809.61±62.64	189.41±18.50
`med`	566.99±64.44	132.67±13.86

Two-way clustering has previously been addressed by the linear NMF methods such as three-factor NMF (e.g. [16]). However, this method is often stuck in trivial local minima where the middle factorizing matrix tends to be smooth or even uniform because of the sparsity of the left and right factorizing matrices [15]. An extra constraint on the middle is therefore needed.

The biclustering problem can be formulated by a QNMF problem: $X \approx LL^T XRR^T$ [21]. The resulting two-sided QNMF objectives can be optimized by alternating the one-sided algorithms, that is, interleaving optimizations of $X \approx LL^T Y^{(R)}$ with $Y^{(R)} = XRR^T$ fixed and $X \approx Y^{(L)} RR^T$ with $Y^{(L)} = LL^T X$ fixed. The bicluster indices of rows and columns are given by taking the maximum of each row in L and R. This method is called *Biclustering QNMF* (Bi-QNMF) or *Two-way QNMF* [21].

Bi-QNMF was implemented by interleaving multiplicative updates between L and R using constant exponents. Here we compare the previous implementation with the adaptive algorithm with variable exponents, using both synthetic and real-world data.

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 from 1, 2, 4, or 7. Next, we compared the four methods on the real-world *webkb* dataset[5]. 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.

The resulting objective evolutions over time are shown in Figure 2. We can see that the dashed curves are below the solid ones for both datasets, which indicates that the adaptive algorithm brings efficiency improvement. The advantage is further quantified in Table 4, where we ran each algorithm ten times and recorded their mean and standard deviation of the convergence times.

[5] `http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/`

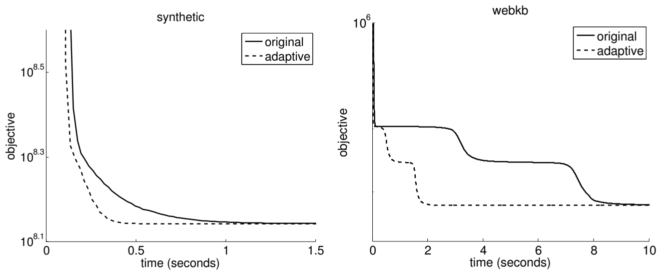

Figure 2: Evolutions of objectives of Bi-QNMF using the original implementation and adaptive multiplicative updates: (left) synthetic data and (right) `webkb` data.

Table 4: The mean (μ) and standard deviation (σ) of the converged time (seconds) of Bi-QNMF using the compared algorithms.

data	original	adaptive
`synthetic`	17.96 ± 0.26	5.63 ± 0.10
`webkb`	139.35 ± 76.81	25.93 ± 13.61

4.3. 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 $x(t) \in \{1, \dots, n\}$ and $y(t) \in \{1, \dots, r\}$, 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 $X = WYW^T$ with $W \triangleq P(x(t) = i|y(t) = k)$. When noise is considered, this becomes an approximative QNMF problem $X \approx WYW^T$. Particularly, when the approximation error is measured by squared Euclidean distance, the parameter estimation problem of HMM can be formulated as (QNMF-HMM)

$$\underset{W \geq 0, Y \geq 0}{\text{minimize}} \quad \mathcal{J}(W) = \|X - WYW^T\|_F^2 \tag{7}$$

$$\text{s.t.} \quad \sum_i W_{ik} = 1 \text{ for all } k, \text{ and } \sum_{kl} Y_{kl} = 1 \tag{8}$$

The QNMF multiplicative update rule for the above constrained problem is

$$W_{ik}^{\text{new}} = W_{ik} \left[\frac{\nabla_{ik}^- + \sum_a \nabla_{ak}^+ W_{ak}}{\nabla_{ik}^+ + \sum_a \nabla_{ak}^- W_{ak}} \right]^{1/4}, \tag{9}$$

where $\nabla_W^- = XWY^T + X^TWY$, $\nabla_W^+ = WYW^TWY^T + WY^TW^TWY$, $\nabla_Y^- = W^TXW$, and $\nabla_Y^+ = W^TWYW^TW$. The update rule guarantees that the Lagrangian $\mathcal{L}(W, \lambda) = \mathcal{J}(\widetilde{W}) + \sum_k \lambda_k (1 - \sum_i W_{ik})$ is non-increasing for $\lambda_k = \sum_i W_{ik}\nabla_{ik}^+ - \sum_i W_{ik}\nabla_{ik}^-$.

Here we apply our adaptive algorithm to relax the constant exponent in Eq. 9. We have compared the new algorithm with the original one on six datasets: the first two are synthetic sequences generated by using the procedure by Lakshminarayanan and Raich [29], with 1000 and 10000 samples respectively; the third and fourth ar letter sequences in the top 1510 and 58112

`www/data/`

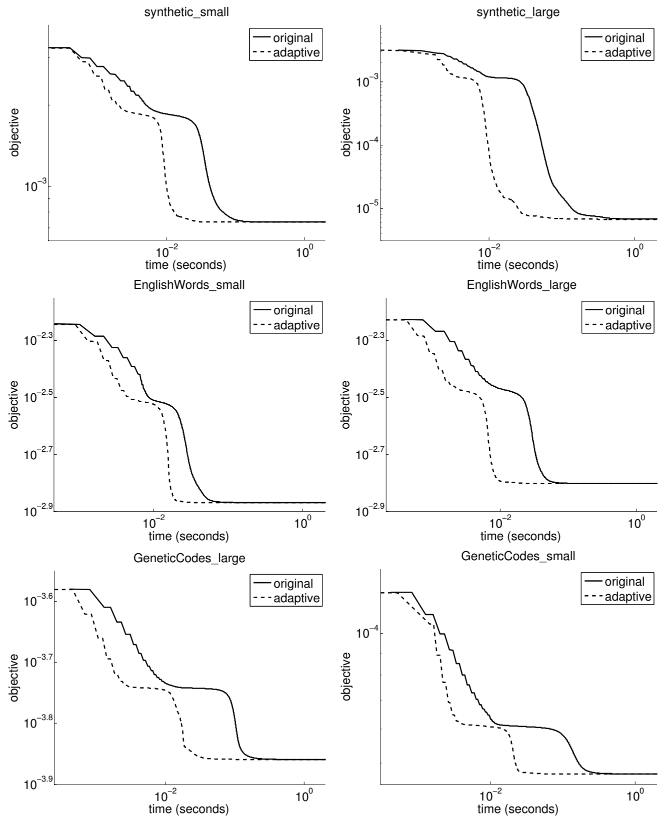

Figure 3: Evolutions of objectives of QNMF-HMM using the original implementation and adaptive multiplicative updates: (left column) datasets with less samples and (right column) with more samples; (top) synthetic data, (middle) English words, and (bottom) genetic codes.

English words[6]; the fifth and sixth are genetic code sequences of homo sapiens[7], respective with lengths 30105 and 187334.

The objective evolution curves are presented in Figure 3. We can see that, for all six datasets, the adaptive method is significantly faster than the original implementation, because the dashed lines are clearly below the solid ones. The improvement is quantified in Table 5, where we ran each method ten times. The results show that our new method can be 2 to 6 times faster in general.

5. Conclusions

We have proposed a simple but effective algorithm to accelerate optimization over nonnegative matrices in quadratic matrix factorization problems, using adaptive multiplicative update rules. The acceleration is achieved by using more aggressive multiplicative learning steps during the iterations. Whenever the monotonicity is violated, we switch back to the safe learning step. This simple strategy has demonstrated considerable advantages in three QNMF applications, for a variety of synthetic and real-world datasets.

[6] `http://www.mieliestronk.com/wordlist.html`
[7] `http://www.ncbi.nlm.nih.gov/genbank/`

Table 5: The mean (μ) and standard deviation (σ) of the converged time (seconds) of QNMF-HMM using the compared algorithms.

data	original	adaptive
`synthetic_small`	7.67 ± 4.23	1.18 ± 0.62
`synthetic_large`	55.81 ± 27.26	14.28 ± 11.24
`EnglishWords_small`	16.23 ± 6.35	8.33 ± 3.95
`EnglishWords_large`	12.07 ± 4.02	3.61 ± 0.90
`GeneticCodes_small`	86.44 ± 16.53	19.39 ± 8.07
`GeneticCodes_large`	22.54 ± 15.68	4.08 ± 3.46

The accelerated algorithms facilitate applications of the QNMF methods. More large-scale datasets will be tested in the future. Moreover, the proposed adaptive exponent technique is readily extended to other fixed-point algorithms that use multiplicative updates.

6. Acknowledgement

This work was financially supported by the Academy of Finland (Finnish Center of Excellence in Computational Inference Research COIN, grant no 251170; Zhirong Yang additionally by decision number 140398).

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. Yang, E. Oja, Quadratic nonnegative matrix factorization, Pattern Recognition 45 (4) (2012) 1500–1510.

[22] H. Zhang, Z. Yang, E. Oja, Adaptive multiplicative updates for projective nonnegative matrix factorization, in: Proceedings of the 19th International Conference Neural Information Processing (ICONIP), 2012, pp. 277–284.

[23] 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.

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

[25] 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.

[26] 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.

[27] 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.

[28] Z. Yang, E. Oja, Unified development of multiplicative algorithms for linear and quadratic nonnegative matrix factorization, IEEE Transactions on Neural Networks 22 (12) (2011) 1878–1891.

[29] 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.