NONNEGATIVE MATRIX FACTORIZATION WITH MATRIX EXPONENTIATION

Siwei Lyu

Computer Science Department
University at Albany, SUNY
Albany, NY 12222, USA

ABSTRACT

Nonnegative matrix factorization (NMF) has been successfully applied to different domains as a technique able to find part-based linear representations for nonnegative data. However, when extra constraints are incorporated into NMF, simple gradient descent optimization can be inefficient for high-dimensional problems, due to the overhead to enforce the nonnegativity constraints. We describe an alternative formulation based on matrix exponentiation, where the nonnegativity constraints are enforced implicitly, and a direct gradient descent algorithm can have better efficiency. In numerical experiments, such a reformulation leads to significant improvement in running time.

Index Terms— nonnegative matrix factorization, matrix exponentiation

1. NMF AND CONSTRAINED NMF

Nonnegative matrix factorization (NMF) [1, 2] aims to find a concise representation for nonnegative data. Given an $m \times n$ data matrix V with $V_{ij} \geq 0$ and an integer $k \leq \min(m, n)$, NMF finds two nonnegative factors $W \in \mathcal{R}^{m\times k}$ and $H \in \mathcal{R}^{k\times n}$ so that $V \approx WH$. The nonnegativity constraints make the representation purely additive that affords a part-based interpretation. NMF has been successfully applied to neural coding [3], brain imaging [4], face recognition [5], color science [6] and polyphonic music transcription [7], among others.

Mathematically, NMF is formulated as minimizing a loss function $L(W, H; V)$, subject to the nonnegativity constraints. Two most commonly used objective functions are the squared Euclidean distance

$$L_2(W, H; V) = \|V - WH\|_2^2 = \textstyle\sum_{ij} \left(V_{ij} - (WH)_{ij}\right)^2, \qquad (1)$$

and the matrix divergence

$$D(W, H; V) = \textstyle\sum_{ij} \left(V_{ij} \log \frac{V_{ij}}{(WH)_{ij}} - V_{ij} + (WH)_{ij}\right). \qquad (2)$$

Optimizing these objective function with W and H jointly is usually difficult, and basically all NMF algorithms iteratively optimize one of W and H given the other. The original NMF algorithm in [2] used multiplicative update steps, as $W \leftarrow W \otimes (VH^T) \oslash (WHH^T)$ and $H \leftarrow H \otimes (W^T V) \oslash (W^T WH)$ for the matrix Euclidean distance (1), where operators $\otimes$ and $\oslash$ denotes the point-wise multiplication and division, respectively. Similar multiplicative updates can also be obtained for for the matrix divergence (2). As long as the current estimations of W and H satisfy the nonnegativity constraints, so do the updated values. These multiplicative updates equal to gradient descent with specific step sizes that always decrease the objective function.

In many practical applications, however, nonnegativity constraints are not sufficient, as the domain knowledge may provide further relevant requirements on the representations. We loosely refer to such variations of NMF that have extra constraints on the nonnegative factors as *constrained NMF*. For instance, in problems such as face recognition, a part-based representation is also preferably a sparse representation, where a small set of factors with non-zero weights suffice to reconstruct an objective. This leads to the development of sparse NMF, which has been the focus of many recent research works [3, 8, 9]. Other requirements such as localness and smoothness have also been incorporated into the NMF objective function for different applications, leading to local NMF [10] and smooth NMF [4]. A general way to introduce these further requirements on the NMF factors is to add penalty terms in the NMF objective function[1].

Nevertheless, when these new objective functions are used, the multiplicative update steps are no longer viable, as they do not guarantee to decrease the modified objective function of the constrained NMF. Numerical algorithms for constrained NMF usually take a simpler yet inefficient method. Basically, W and H are updated with a gradient descent, yet whenever negative values appear in W or H, they are clipped to zero. The procedure guarantees to converge to a local minimum of the objective function, yet its efficiency in high-dimensional problem is greatly affected by these clipping operations. Especially, if following the gradient will break the nonnegativity constraint on W_{ij} or H_{ij}, progress in updating them can only be made with a projection of a full gradient on the constraints, which can be significantly smaller

[1] Adding new constraints to the system is another way to incorporate new requirements, yet through Lagrange multipliers, those constraints can be treated similarly as penalty terms in the objective functions.

than a full gradient update step.

2. NMF WITH MATRIX EXPONENTIATION

To find an efficient optimization scheme applicable to NMF and its constrained variants, in this paper, we take an alternative perspective of the NMF optimization. Specifically, we reformulate NMF to find factors that are the point-wise exponentiation of another matrices. The introduction of matrix exponentiation annihilates the nonnegativity constraints, and a gradient descent with matrix exponentiation always searches the optimal solution within the feasible set, thus avoids moving on the constraint boundaries. Furthermore, gradient descent with matrix exponentiation can also make greater progress in regions relatively faraway from the optimal solution. All these help to reduce the number of iterations to converge and improve the running efficiency of the gradient-based solution to NMF and its constrained variants.

To be more specific, we assume the two factors W and H as point-wise exponentiation of two other matrices $\mathcal{W}$ and $\mathcal{H}$, respectively, as $W_{ij} = e^{\mathcal{W}_{ij}}$ and $H_{ij} = e^{\mathcal{H}_{ij}}$, or more compactly, $W = e \owedge \mathcal{W}$ and $H = e \owedge \mathcal{H}$, where operator $\owedge$ denotes the point-wise exponentiation. NMF optimization with matrix exponentiation is solved with gradient descent of $\mathcal{W}$ and $\mathcal{H}$, as: $\mathcal{W}_{t+1} \leftarrow \mathcal{W}_t - \eta_{\mathcal{W}} \frac{\partial L}{\partial \mathcal{W}}\Big|_{\mathcal{W}_t}$ and $\mathcal{H}_{t+1} \leftarrow \mathcal{H}_t - \eta_{\mathcal{H}} \frac{\partial L}{\partial \mathcal{H}}\Big|_{\mathcal{H}_t}$, where $\eta_{\mathcal{W}}$ and $\eta_{\mathcal{H}}$ are suitable step sizes for $\mathcal{W}$ and $\mathcal{H}$, respectively. The gradients of the NMF objective function with regards to $\mathcal{W}$ and $\mathcal{H}$ have relatively simple form. For (1), assuming all point-wise operations have higher priority in computation, and using shorthand notation $\otimes$ and $\oslash$ for point-wise multiplication and division, the derivatives of $L_2(W, H; V)$ with regards to $\mathcal{W}$ and $\mathcal{H}$ are computed as:

$$\frac{\partial \|V - WH\|_2^2}{\partial \mathcal{W}_{mn}} = \frac{\partial}{\partial \mathcal{W}_{mn}} \sum_{ij} \left(\sum_k \exp(\mathcal{W}_{ik} + \mathcal{H}_{kj}) - V_{ij} \right)^2$$

$$= 2 \sum_j \left(\sum_k \exp(\mathcal{W}_{mk} + \mathcal{H}_{kj}) - V_{mj} \right) \frac{\partial}{\partial \mathcal{W}_{mn}} \exp(\mathcal{W}_{mn} + \mathcal{H}_{nj})$$

$$= 2 \exp(\mathcal{W}_{mn}) \sum_j \left(\sum_k \exp(\mathcal{W}_{mk} + \mathcal{H}_{kj}) - V_{mj} \right) \exp(\mathcal{H}_{nj})$$

$$= 2 \left[W \otimes \left((WH - V) H^T \right) \right]_{mn},$$

and

$$\frac{\partial \|V - WH\|_2^2}{\partial \mathcal{H}_{mn}} = \frac{\partial}{\partial \mathcal{H}_{mn}} \sum_{ij} \left(\sum_k \exp(\mathcal{H}_{ik} + \mathcal{W}_{kj}) - V_{ij} \right)^2$$

$$= 2 \sum_i \left(\sum_k \exp(\mathcal{W}_{ik} + \mathcal{H}_{kn}) - V_{in} \right) \frac{\partial}{\partial \mathcal{H}_{mn}} \exp(\mathcal{W}_{im} + \mathcal{H}_{mn})$$

$$= 2 \exp(\mathcal{H}_{mn}) \sum_i \left(\sum_k \exp(\mathcal{W}_{ik} + \mathcal{H}_{kn}) - V_{in} \right) \exp(\mathcal{W}_{im})$$

$$= 2 \left[H \otimes \left(W^T (WH - V) \right) \right]_{mn}.$$

Similarly, for (2), the derivatives of $D(W, H; V)$ with regards to $\mathcal{W}$ and $\mathcal{H}$ are computed as:

$$\frac{\partial D(V \| WH)}{\partial \mathcal{W}_{mn}} = \frac{\partial}{\partial \mathcal{W}_{mn}} \sum_{ij} \left(V_{ij} \log \frac{V_{ij}}{(WH)_{ij}} - V_{ij} + (WH)_{ij} \right)$$

$$= \sum_j \left(\exp(\mathcal{W}_{mn} + \mathcal{H}_{nj}) - \frac{V_{ij} \exp(\mathcal{W}_{mn} + \mathcal{H}_{nj})}{\sum_k \exp(\mathcal{W}_{mk} + \mathcal{H}_{kj})} \right)$$

$$= \mathcal{W}_{mn} \sum_j \frac{\mathcal{H}_{jn}((WH)_{mj} - V_{mj})}{(WH)_{mj}}$$

$$= \left[W \otimes \left(((WH - V) \oslash (WH)) H^T \right) \right]_{mn},$$

and

$$\frac{\partial D(V \| WH)}{\partial \mathcal{H}_{mn}} = \frac{\partial}{\partial \mathcal{H}_{mn}} \sum_{ij} \left(V_{ij} \log \frac{V_{ij}}{(WH)_{ij}} - V_{ij} + (WH)_{ij} \right)$$

$$= \sum_i \left(\exp(\mathcal{W}_{im} + \mathcal{H}_{mn}) - \frac{V_{in} \exp(\mathcal{W}_{im} + \mathcal{H}_{mn})}{\sum_k \exp(\mathcal{W}_{ik} + \mathcal{H}_{kn})} \right)$$

$$= \mathcal{H}_{mn} \sum_i \frac{\mathcal{W}_{im}((WH)_{in} - V_{in})}{(WH)_{in}}$$

$$= \left[H \otimes \left(W^T ((WH - V) \oslash (WH)) \right) \right]_{mn}.$$

It is interesting to note that the derivatives given in these equations are also the KKT complementarity conditions for objective function (1) and (2) [11]. The major difference is that instead of updating $\mathcal{W}$ and H directly as in most existing works of NMF, their point-wise logarithm $\mathcal{W}$ and $\mathcal{H}$ are used to remove the nonnegativity constraints. In terms of numerical optimization techniques, matrix exponentiation formulation of NMF can be viewed as an interior-point style solution to the bounded optimization problem in the NMF, with the exponentiation serves as the barrier function [12]. Note also that matrix exponentiation naturally implies a multiplicative update of $\mathcal{W}$ and $\mathcal{H}$ as in the original NMF algorithm [2], since the gradient updating step is equivalent to $W_{t+1} \leftarrow W_t e \owedge \left(-\eta_{\mathcal{W}} \frac{\partial L}{\partial \mathcal{W}}\Big|_{\mathcal{W}_t} \right)$ and $H_{t+1} \leftarrow H_t e \owedge \left(-\eta_{\mathcal{H}} \frac{\partial L}{\partial \mathcal{H}}\Big|_{\mathcal{H}_t} \right)$. However, the matrix exponentiation formulation can also be employed in other constrained NMF algorithms where the original multiplicative updates are not applicable. In practice, to avoid small updates for $\mathcal{W}_{ij}$ and $\mathcal{H}_{ij}$ with values close to zero, when they are below a pre-set threshold, we set these entries to zeros and stop optimizing them in the subsequent steps.

3. EXPERIMENT

In this section, we report the empirical performance of gradient descent with matrix exponentiation (ME) and compare with the original NMF update algorithm (NMF) and the projected gradient descent of the (constrained) NMF objective function without any transformation (PGD). We tested the algorithm on both artificial data, where V is a randomly generated 100×100 positive matrix, and we search for factors with $k = 25$, and a real data set from the a subset of the USPS handwritten digit database, as proposed in [3]. All algorithms compared started from the same randomly chosen initial values and stopped when the objective function did not decrease more than a pre-given threshold. PGD and ME used optimal step size in gradient updating found from an optimal search, which is obtained by numerically solve a $1D$ nonlinear equation at each step.

3.1. Sparse NMF

In the first set of experiments, we enforce the sparseness requirement to the NMF problem by adding penalty term in the NMF objective function that penalizes nonsparse decompositions. A particular effective measurement of sparseness of an n-dimensional vector x is given in [3], as

$$s(x) = \frac{\sqrt{n} - \|x\|_1 / \|x\|_2}{\sqrt{n} - 1} = \frac{\sqrt{n} - \sum_{i=1}^n x_i / \sqrt{\sum_{i=1}^n x_i^2}}{\sqrt{n} - 1}.$$

which is 0 for a vector with equal entries (most nonsparse) and 1 for a vector with only one nonzero entry (most sparse). Note that as the factors are constrained to be nonnegative, the l_1 norm in $s(x)$ is replaced with a summation, which is differentiable. Following the method in [3], we used an augmented objective function based on (1),

$$L_s(W, H; V) = L_2(W, H; V) + \lambda_W \sum_{i=1}^k s(W_i) + \lambda_H \sum_{i=1}^k s(H_i^T) \quad (3)$$

which penalizes nonsparse columns in W and H^T, with λ_W and λ_H constants controlling the balance between data fitting and sparsity requirements in the factors. Matrix exponentiation of this objective function is easy to set up with point-wise exponentiation. Figure 1 are the experimental results of applying NMF, PGD and ME to the sparse NMF objective function (3) with both synthesized and USPS data set. The use of multiplicative update in the original NMF on this problem is inappropriate in the sense that those steps are not designed to minimize objective function (3). Therefore, though it converges with the least number of steps and least running time, it does not always lead to a local minimum of the sparseness-enhanced objective function. This is more clearly illustrated in the right panel of Figure 1: NMF will stops at a local minimum of the original NMF objective, which is not necessarily a local minimum of the enhanced sparse NMF objective function. On the other hand, algorithm ME and PGD both lead to a local minimum of (3), but ME achieves so with less iteration steps and running time, due to its quick movement in regions faraway from the solution and the avoidance of nonnegativity constraints. Nevertheless, the optimal step search costs more time in the case of ME than PGD, as the nonlinear equation to be solved is slightly complicated, which accounted for the disproportion of improvements in running time and iteration steps.

3.2. Smooth NMF

Smooth NMF [4] is a constrained variant of NMF where one seeks nonnegative factors that are also smooth, i.e., having small difference between consecutive elements in the factors. Smooth NMF has application in analyzing fMRI data, where a desirable property of the recovered factors is their smoothness in the temporal and voxel domain. Mathematically, smooth NMF is formulated as optimizing the following objective function:

$$L_m(W, H; V) = L_2(W, H; V) + \lambda_W \sum_{i=1}^k \|\mathcal{D} W_i\|_2 + \lambda_H \sum_{i=1}^k \|\mathcal{D} H_i^T\|_2, \quad (4)$$

where $\mathcal{D}$ is a difference operator. Less smooth columns in $\mathcal{W}$ and H^T are penalized by the of l_2 norm of their differentials. Shown in Figure 2 are the performance of algorithm NMF, PGD and ME with the smooth NMF objective function on both synthesized and USPS data. As in the case of sparse NMF, NMF updates does not guarantees to decrease the objective function, and thus not able to find a locally optimal solution to smooth NMF. On the other hand, ME converges with significantly less iteration steps than PGD. However, the difference in running time is less significant, the reason is that in solving the smooth NMF optimization, the optimal 1D line search step for ME involves numerically solving a more complicated nonlinear equation, and thus slow down the running time of individual step. A practical strategy is to switch to PGD whenever the improvement in objective function is improved less than a pre-given threshold in ME. In practice, this ad hoc solution achieved better (30%) improvement in the running time of smooth NMF (not shown here).

4. CONCLUSION

In this paper, we describe a new formulation of NMF and its constrained variants with matrix exponentiation. The major advantage of this new formulation is that nonnegativity constraints are enforced implicitly and thus obviate the overhead of a gradient descent algorithm to check the nonnegativity constraints. Furthermore, the nonlinear transformation of the objective function introduced by matrix exponentiation can also accelerate convergence speed in regions that are faraway from the optimal solution, which also helps to reduce the overall running time of constrained NMF. The bottleneck of the proposed algorithm is the expensive step of searching for an optimal update step, which usually involves numerical solution of nonlinear equations. We are working on approximation schemes that can further improve the efficiency of the current algorithm.

5. REFERENCES

[1] P. Paatero and U. Tapper, "Positive matrix factorization: A nonnegative factor model with optimal utilization of error," *En-*

	run time (sec.)	iteration steps	obj fun error
NMF	42.7	140.3	0.91
PGD	1012.7	794.9	8.7×10^{-3}
ME	767.6	391.2	8.6×10^{-3}

synthesized data

	run time (sec.)	iteration steps	obj fun error
NMF	161.5	258.4	13.8
PGD	1521.1	1282.6	0.77
ME	1129.4	862.7	0.74

USPS

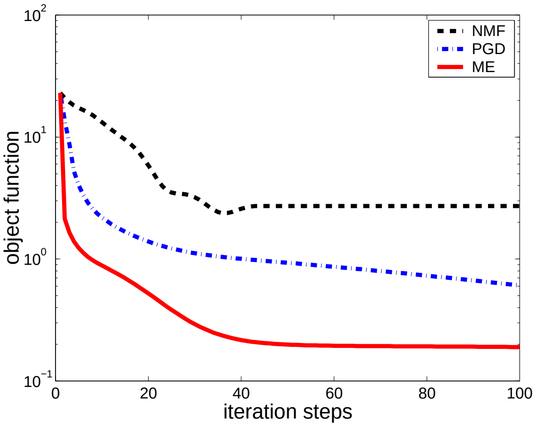

Fig. 1. Left Empirical performance of algorithm NMF, PGD and ME on two data sets for sparse NMF objective function. **Right** Plot of iteration steps vs. objective function for a typical run of the three algorithm on a synthesized datum.

	run time (sec.)	iteration steps	obj fun error
NMF	79.4	145.3	8.9
PGD	1332.2	1253.7	9.5×10^{-3}
ME	1037.2	482.3	9.5×10^{-3}

synthesized data

	run time (sec.)	iteration steps	obj fun error
NMF	197.2	232.5	21.4
PGD	1342.5	994.7	0.53
ME	1152.8	565.4	0.54

USPS

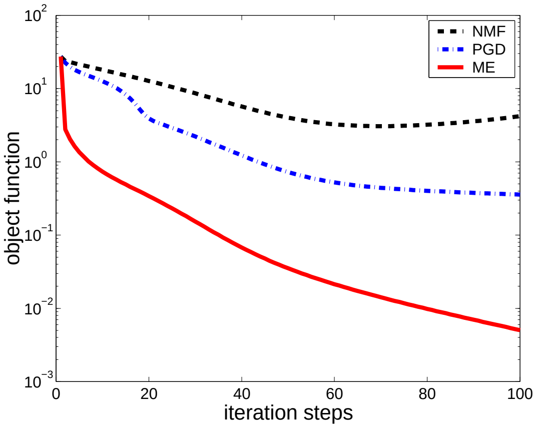

Fig. 2. Left Empirical performance of algorithm NMF, PGD and ME on two data sets for smooth NMF objective function. **Right** Plot of iteration steps vs. objective function for a typical run of the three algorithm on a synthesized datum.

vironmetrics, vol. 5, pp. 111–126, 1994.

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

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

[4] A. Pascual-Montano *et.al*, "Nonsmooth nonnegative matrix factorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 3, pp. 403–417, 2006.

[5] D. Guillamet and J. Vitria, "Nonnegative matrix factorization for face recognition," in *Proc. Conf. Topics in Artificial Intelligence*, 2002, pp. 336–344.

[6] G. Buchsbaum and O. Bloch, "Color categories revealed by nonnegative matrix factorization of Munsell color spectra," in *Vision research*, 2002, vol. 42, pp. 559–563.

[7] P. Smaragdis and J. C. Brown, "Nonnegative matrix factorization for polyphonic music transcription," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2003.

[8] P. O. Hoyer, "Nonnegative sparse coding," in *Proc. IEEE Workshop on Neural Networks for Signal Processing*, 2002.

[9] W. Liu, N. Zheng, and X. Lu, "Nonnegative matrix factorization for visual coding," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2003.

[10] T. Feng, S.Z. Li, H. Y. Shum, and H. Zhang, "Local nonnegative matrix factorization as a visual representation," in *Proc. of Second International Conference on Development and Learning*, 2002.

[11] M. Chu, F. Diele, R. Plemmons, and S. Ragni, "Optimality, computation, and interpretations of nonnegative matrix factorizations," Unpublished report, http://www.wfu.edu/lemmons/papers.htm, 2004.

[12] R. Fletcher, *Practical Methods of Optimization*, John Wiley and Sons, 2nd edition, 1987.