On the computation of a nonnegative matrix factorization and its application in telecommunications

Ana Almeida†, Luís Merca Fernandes¥ , Joaquim Júdice*, João Patrício#

† University of Coimbra and Informatics and Systems Center, Coimbra, Portugal. E-mail: amca@mat.uc.pt

¥ Polytechnic Institute of Tomar and Institute of Telecommunications, Portugal. E-mail: lmerca@co.it.pt

* University of Coimbra and Institute of Telecommunications, Coimbra, Portugal. E-mail: Joaquim.Judice@co.it.pt

Polytechnic Institute of Tomar and Institute of Telecommunications, Portugal. E-mail: Joao.Patricio@co.it.pt

Abstract — **The Nonnegative Matrix Factorization (NMF) has become an increasingly popular approach in many areas of telecommunications. A number of properties and a nonlinear programming formulation for NMF are introduced, which allow approximations to the solution of diverse image processing problems, ranging from data analysis to video summarization, pattern recognition and image reconstruction. A spectral projected-gradient algorithm is investigated for the solution of the corresponding optimization problem. Techniques for finding an initial point of the decomposition are also discussed. Some computational experience is reported to highlight the efficacy of these techniques in practice.**

I. Introduction

The Nonnegative Matrix Factorization (NMF) problem can be stated as follows: given a nonnegative matrix $V \in \mathbb{R}^{m\times n}$, find nonnegative matrices $W \in \mathbb{R}^{m\times r}$ and $H \in \mathbb{R}^{r\times n}$ such that $V = WH$. This problem has been intensively applied in recent years in various fields of science and engineering, such as biomedical applications, face and object recognition, data mining and semantic analysis. This approach seems to be particularly effective for problems where the identification of components is of utmost importance [8]. In this paper we mainly focus on video and image signal processing. For these problems, the matrices V,W and H have a particular structure that reflects the nature of the problem. For this reason, NMF is recommended for data compression prior to analysis and component identification. Furthermore, NMF has become increasingly popular for component and feature identification in image and video signals, when compared with other techniques, such as principal component analysis [4]. Other fields of application include sensor technology and computer hardware, where massive quantities of data have to be processed and classical data analysis tools easily become inadequate. The processing of these huge amounts of data created the need of new tools for data representation, disambiguation and dimensionality reduction. Furthermore, in many situations the data observed from complex phenomena represent the integrated result of several interrelated variables acting together. A reduced system model may provide the retrieval of information at a level close to the original system. In practice, quite often the data to be analyzed has to be nonnegative. Since classic tools cannot guarantee the maintenance of this property, the NMF may be a valid approach to deal with such problems.

II. Characterization of the NMF problem

We start by noting that, whenever there is one exact factorization, there are infinity possible pairs for the decomposition. In fact, if (W^*, H^*) is a solution for NMF, so are all the pairs of the form $(W^*D, D^{-1}H^*)$ for any diagonal matrix of order r with positive diagonal elements. The question to be answered is not how many decompositions there are but instead when one exists. It is a classical result [3] that the existence of an exact factorization for a given matrix V depends on the parameter r. The following definition helps to understand the NMF problem.

Definition: Given a nonnegative matrix of order $m\times n$, the minimum positive integer r such that there are matrices $W_{m\times r}$ and $H_{r\times n}$ satisfying $V = WH$ is called the positive rank of V and is denoted by $posrank(V)$.

Then the following result holds [3].

Theorem: For a given nonnegative matrix V of order $m\times n$,

$$rank(V) \le posrank(V) \le \min\{m, n\}$$

Therefore finding an exact factorization for a nonnegative matrix V directly depends on the value of r that is used. The exact factorization can only be achieved by fixing $r \ge posrank(V)$. For values of r below this limit, only an approximate decomposition for V can be computed. On the other hand the increase of the value of r implies a bigger computational work for computing the NMF of a matrix. Finally there are no efficient algorithms to compute the posrank of a nonnegative matrix. Fortunately, in some applications such as image recognition, computing an exact factorization is not an important issue [11]. Contrary to the LU decomposition of a square matrix, there is no direct algorithm for computing the nonnegative decomposition of a nonnegative matrix. The most common approach is to formulate NMF as an optimization problem and seek to obtain an approximation of a global optimum to this problem of good quality. The most important formulation for the NMF employs the Frobenius norm of a matrix and is given as below [8]:

$$\text{Minimize } \frac{1}{2}\|V - WH\|_F^2 \quad \text{subject to } W, H \geq 0 \tag{1}$$

where V is a given nonnegative matrix of order $m \times n$, W is a $m \times r$ matrix, H is a $r \times n$ matrix and $\|\bullet\|_F$ denotes the Frobenius norm.
Clearly, the product WH is an approximate factorization of the matrix V. As stated before, an appropriate decision for the choice of the value of r is critical in practice. The dimension of this formulation is $(n+m)r$, which implies that, the bigger is the order of the data matrix V and of the value of the parameter r, the larger is the dimension of the optimization problem. Other important challenges affecting the numerical solution of the formulation (1) include the existence of different local minima due to the non-convexity of the objective function. Nevertheless, (1) constitutes a global optimization problem for which the global optimal value is known to be zero provided an exact factorization for the matrix V exists for the chosen value of r.

III. Algorithms for the NMF problem

A great deal of effort has been devoted to the design of efficient algorithms. In particular, the multiplicative Lee-Seung algorithm [1, 7, 9] has been recommended by several authors. This procedure is essentially a fixed-point approach that can be expressed as follows:

Lee-Seung Algorithm:

1. Initialize *W, H;*
2. While stopping condition is not satisfied, repeat

$$w_{ij} = w_{ij}\frac{[VH^T]_{ij}}{[WHH^T]_{ij}}$$

$$h_{jk} = h_{jk}\frac{[W^TV]_{jk}}{[W^TWH]_{jk}}$$

It is possible to show that the algorithm converges under some reasonable assumptions. However, there is no guarantee that the accumulation point found by the algorithm is even a stationary point of the objective function on the constraint set defined by the nonnegative constraints mentioned above. On the positive side, the algorithm is quite simple to implement for a dense or a sparse matrix V.
Another approach for the solution of this problem is the so-called Alternate Least-Squares (ALS) Algorithm [1, 7, 11]. This method consists of alternately solving two linear least-squares problems and can be stated as follows:

Alternate Least-Squares (ALS) Algorithm:

1. Given $\overline{W} \geq 0$;
2. While stopping condition is not satisfied, repeat
   a. Solve linear least-squares problem
   $$\min_{H\geq 0}\frac{1}{2}\|V - \overline{W}H\|_F^2 \rightarrow \overline{H}$$
   b. Solve linear least-squares problem
   $$\min_{W\geq 0}\frac{1}{2}\|V - W\overline{H}\|_F^2 \rightarrow \overline{W}$$

This algorithm derives from the observation that the NMF objective function is convex on either of the two variables H and W. Therefore, given one of these two matrices, the other one can be computed by linear least-squares calculations. The nonnegative constraints on both the linear least-squares problems ensure convergence of the Alternate Least-Squares Algorithm to a stationary point of the objective function on the set defined by the nonnegative constraints. However, a great amount of work is required as two least-squares problems with nonnegative constraints have to be solved in each iteration. In practice, the optimal solutions of the unconstrained least-squares problems are found and then projected to the constraint set of the NMF optimization problem (1). This is simply done by transforming all the negative components of the unconstrained optimal solutions of the linear least-squares problems to zero. This modification reduces the amount of work of the algorithm to a great extent. On the negative side, there is no guarantee that the modified algorithm converges to a stationary point of the objective function on the constraint set defined by the nonnegative constraints.
Recognizing the drawbacks of the two approaches discussed before, we propose to use of the so-called Spectral Projected-Gradient (SPG) method to the solution of the optimization problem (1). This algorithm has been considered to be an efficient technique for the solution of large-scale optimization structured problems [2, 4, 8]. In some cases [8] the algorithm even outperforms some the most important commercial nonlinear programming codes, such as LOQO [12] and MINOS [10]. The SPG algorithm can be applied to general minimization problems of the form

$$\min \varphi(x) \quad \text{subject to } x \in K$$

where $\varphi : \mathbb{R}^p \rightarrow \mathbb{R}$, and $K \subset \mathbb{R}^p$ is a convex and closed set. If x = (W, H), the objective function is given by

$$\varphi(W,H) = \frac{1}{2}\|V - WH\|_F^2 = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}(v_{ij} - \sum_{k=1}^{r} w_{ik}h_{kj})^2$$

and the constraint set is defined by nonnegative constraints, the steps of the SPG algorithm can be sated as below.

Spectral Projected-Gradient (SPG) Algorithm:

0. Given $W_0 \geq 0$, $H_0 \geq 0$, $\beta,\tau \in]0,1]$, $\varepsilon > 0$ and $k = 0$;
1. Compute $\nabla_W \varphi(W_k, H_k)$ and $\nabla_H \varphi(W_k, H_k)$;
2. Compute the projected gradient direction:
$$(z_W, z_H) = P((W_k, H_k) - \eta_k(\nabla_W \varphi(W_k, H_k), \nabla_H \varphi(W_k, H_k)))$$
$$(d_w, d_H) = (z_w, z_H) - (W_k, H_k)$$
where $P(X,Y)$ is the projection of (X,Y) on the set defined by the nonnegative constraints.
3. If $\|(d_w, d_H)\| < \varepsilon$ then (W_k, H_k) is a stationary point;
Stop;
4. Compute $\delta_k = \beta^{m_k}$, where m_k is the first nonnegative integer m satisfying
$$\varphi(W_k + \beta^m d_W, H_k + \beta^m d_H) \leq \varphi(W_k, H_k) - \tau\beta^m (d_W, d_H)^T \theta$$
where $\theta = (\nabla_W \varphi(W_k, H_k), \nabla_H \varphi(W_k, H_k))$.
5. Update $(W_{k+1}, H_{k+1}) = (W_k, H_k) + \delta_k(d_w, d_H)$
6. Set $k = k+1$ and return to Step 1.

According to [8], the parameter η_k is computed by the following procedure:
$$k = 0 \Rightarrow \eta_k = 1$$
$$\forall k > 0 : \eta_k = \begin{cases} P_{[\eta_{\min}, \eta_{\max}]}\left(\dfrac{\xi_k}{v_k}\right) \text{if } v_k > 0 \\ \eta_{\max} \quad \text{otherwise} \end{cases}$$
where $P_{[l,u]}(\alpha)$ represents the projection of $\alpha \in \mathbb{R}$ on the interval $[l,u]$,
$$\xi_k = \left((W_k, H_k) - (W_{k-1}, H_{k-1})\right)^T \left((W_k, H_k) - (W_{k-1}, H_{k-1})\right)$$
$$v_k = \left((W_k, H_k) - (W_{k-1}, H_{k-1})\right)^T (\nabla_W \varphi(W_k, H_k), \nabla_H \varphi(W_k, H_k))$$
and $\eta_{\min}$ and $\eta_{\max}$ are small and large numbers respectively. In practice $\eta_{\min} = 10^{-2}$ and $\eta_{\max} = 10^2$ are appropriate in general.
Furthermore the projection $(Z,U) = P(X,Y)$ is computed as follows:
$$Z_{ij} = \min\{0, X_{ij}\}, \quad U_{ij} = \min\{0, Y_{ij}\}, \quad \text{for all } i,j$$
As before, the implementation of the SPG method is quite simple. Furthermore it is possible to show [2] that under reasonable hypotheses the algorithm converges to a stationary point of the objective function on the constraint set defined by the nonnegative constraints.
All the three methods described in this section are local in the sense that can guarantee at most a stationary point of the objective function on the set defined by the nonnegative constraints. However, the main objective of these algorithms is to find a feasible solution with a small objective function value. Computational experience reported elsewhere shows that the choice of initial point (W_0, H_0) is quite important to this goal. In practice the algorithms are run several times with different initial points and the NMF is chosen as the feasible solution (H,W) of (1) with a smaller objective function value. As discussed in [1] several approaches have been recommended for finding such initial points. One of the simplest techniques that has been quite employed in practice consists of randomly generate the elements of these initial matrices. Alternatively, a second simple approach assigns fixed values to the elements of the matrices. It is believed that the first approach performs well and usually better than the one that uses initial iterates with fixed elements.

IV. Computational Experience

In this section we study the numerical efficiency of the SPG algorithm and the effect of the initial approximations on the performance of the SPG algorithm. For this purpose, a random test problem was built, such that the nonnegative matrix V is constructed so that
$$V = \left[V_1 : u_1 u_2 \dots u_{n-r}\right], u_i \in \mathbb{R}^m, V_1 \in \mathbb{R}^{m \times r}$$
where V_1 is a given randomly generated matrix, u_i are column vectors of the form $u_i = [v_{1,r+i} \dots v_{n,r+i}]^T$, $i = 1, \dots, n-r$ and
$$v_{k,r+i} = \sum_{j=1}^{r} v_{kj} \alpha_{ji}, k = 1, \dots, m$$
with $\alpha_{ji} \geq 0$, for all i,j. Trivially, at least one factorization $V = W'H'$ exists and is given by
$$W' = V_1 \text{ and } H' = \begin{bmatrix} & \vdots & \alpha_{11} & \dots & \alpha_{1,n-r} \\ I_r & \vdots & \vdots & \dots & \vdots \\ & \vdots & \alpha_{r1} & \dots & \alpha_{r,n-r} \end{bmatrix}$$

Two NMF problems (PROB1 and PROB2) were generated following this procedure. Tables 1, 2 and 3 display the performances of Lee-Seung, ALS and SPG algorithms, respectively on the solution of these problems. In these tables, NVAR represents the total number of variables of the optimization problem (1), CPU is the total CPU time required for the solution of the optimization problem (1) on a Pentium 4 computer with 1Gb RAM running at 2Ghz, ITER is the total number of iterations and VAL is the value of the objective function at the solution found by the algorithms. Finally we have used the tolerance $\varepsilon = 10^{-4}$ for the SPG algorithm and the stopping criteria stated in [6] for the two remaining algorithms with the same tolerance. Four initial matrices were tested in the experience. The first technique for the construction of the initial matrix is denoted by Rand and consists of choosing all the elements as randomly positive values in the interval $[0,1]$. The remaining three initial matrices have all its elements equal to 0.25, 0.50 and 0.75 respectively. The values in the row Rand displayed in these tables correspond to the average of the computational effort required by the three algorithms for the solution of each one of the problems PROB1 and PROB2 with 5 different initial points.

The results seem to indicate that the random initial point strategy leads to good approximations for the solution of the NMF problems by the SPG algorithm, at the expense of large iteration counts and CPU times. On the other hand, fixed strategies force the algorithm to compute stationary points that are not solutions of the NMF problems in a quite small number of iterations and CPU time. These conclusions also seem to be valid for the two remaining algorithms. The experiences also show that the SPG algorithm always finds stationary points of (1) with a small objective function value when the RAND initial strategy is employed, while the two remaining methods are not so consistent in this extent.

Problems	M	N	r	NVAR	INIT	CPU	ITER	VAL
PROB1	12	24	4	144	Rand	0.08	416.2	0.07
					0.25	0.02	2	2.13
					0.50	0.02	2	2.13
					0.75	0.02	2	2.13
PROB2	24	48	4	288	Rand	0.11	431.6	0.15
					0.25	0.02	2	4.87
					0.50	0.02	2	4.87
					0.75	0.02	2	4.87

Table 1 – Performance of the Lee-Seung algorithm.

Problems	M	N	r	NVAR	INIT	CPU	ITER	VAL
PROB1	12	24	4	144	Rand	0.06	85	1.61
					0.25	0.03	3	2.12
					0.50	0.03	3	2.12
					0.75	0.03	3	2.12
PROB2	24	48	4	288	Rand	0.06	118	0.08
					0.03	0.03	3	4.86
					0.03	0.03	3	4.86
					0.03	0.02	3	4.86

Table 2 – Performance of the ALS algorithm

Problems	M	N	r	NVAR	INIT	CPU	ITER	VAL
PROB1	12	24	4	144	Rand	0.38	3505.6	0.00492
					0.25	0.04	59	4.47
					0.50	0.07	110	4.47
					0.75	0.03	43	4.47
PROB2	24	48	4	288	Rand	1.00	5194.4	0.003748
					0.25	0.05	47	4.86
					0.50	0.05	67	4.86
					0.75	0.05	39	4.86

Table 3 – Performance of the SPG algorithm.

V. Conclusions

In this paper some important issues concerning the nonnegative decomposition of a matrix are first introduced. The use of the SPG algorithm for computing an approximate NMF is investigated. Some experiences with the SPG and the traditional Lee-Seung and ALS algorithms on a set of NMF instances are also reported. The numerical results seem to indicate that the SPG algorithm is in general able to compute good quality approximate factorizations in a reasonable amount of time. Furthermore, the SPG algorithm has shown to be competitive with Lee-Seung and ALS methods in terms of computational effort and seems to be more consistent than their alternative techniques for computing good approximate decompositions. The incorporation of preconditioning techniques in the SPG algorithm may improve its efficiency and efficacy and should be investigated in future.

Keywords: matrix factorization, nonlinear programming, large-scale problems, image processing models.

REFERENCES

[1] M. W. BERRY, M. BROWNE, A. N. LANGVILLE, P. V. PAUCA, AND R. J. PLEMMONS, *Algorithms and applications for approximate nonnegative matrix factorization*, Computational Statistics & Data Analysis, 52 (2007), pp. 155–173.

[2] E. G. BIRGIN, J. M. MARTINEZ, AND M. RAYDAN, *Nonmonotone spectral projected gradient methods on convex sets*, SIAM Journal on Optimization, 10 (2000), pp. 1196–1211.

[3] J. E. COHEN, AND U. G. ROTHBLUM, *Nonnegative Ranks, Decompositions, and Factorizations of Nonnegative Matrices,* Linear Algebra and its Applications 190 (1993), pp. 149-168.

[4] E. G. BIRGIN, J. M. MARTINEZ, AND M. RAYDAN, *Algorithm 813 SPG: software for convex–constrained optimization*, ACM Transactions on Mathematical Software, 27 (2001), pp. 340–349.

[5] W.-S. CHEN, B. PAN, B. FANG, M. LI, AND J. TANG, *Incremental nonnegative matrix factorization for face recognition*, Mathematical Problems in Engineering, 2008.

[6] A. CICHOCKI AND R. ZDUNEK, *NMFLAB – MATLAB Toolbox for Non-Negative Matrix Factorization*, http://www.bsp.brain.riken.jp/ICALAB/nmflab.html

[7] P. O. HOYER, *Nonnegative matrix factorization with sparseness constraints*, Jour. Machine Learning Research, 5 (2004), pp. 1457–1469.

[8] J. JÚ DICE, M. RAYDAN, S. ROSA, AND S. SANTOS, *On the solution of the symmetric eigenvalue complementarity problem by the spectral projected gradient algorithm*, Numerical Algorithms, 45 (2008), pp. 391–407.

[9] D. D. LEE AND H. S. SEUNG, *Learning the parts of objects by non-negative matrix factorization*, Nature, 401 (1999), pp. 788–791.

[10] B. A. MURTAGH AND M. A. SAUNDERS, *MINOS 5.1 user guide*, tech. report, Department of Operations Research, Stanford University, 1987.

[11] P. PAATERO AND U. TAPPER, *Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values*, Environmetrics, 5 (1994), pp. 111–126.

[12] R. J. VANDERBEI, *LOQO: An interior point code for quadratic programming*, Optimization Methods and Software, 12 (1999), pp. 451–484.