2009 Ninth International Conference on Intelligent Systems Design and Applications

A penalty function for computing orthogonal non-negative matrix factorizations

Nicoletta Del Buono
Department of Mathematics
University of Bari
via E. Orabona 4, 70125 Bari, Italy
delbuono@dm.uniba.it

Abstract

Nonnegative matrix factorization (NMF) is a widely-used method for multivariate analysis of nonnegative data to obtain a reduced representation of data matrix only using a basis matrix and a encoding variable matrix having non-negative elements. A NMF of a data matrix can be obtained by finding a solution of a nonlinear optimization problem over a specified cost function. In this paper we investigate the formulation and then the computational techniques to obtain orthogonal NMF, when the orthogonal constraint on the columns of the basis is added. We propose a penalty objective function to be minimized on the intersection of the set of non-negative matrices and the Stiefel manifold in order to derive a projected gradient flow whose solutions preserve both the orthogonality and the non-negativity.

1. Introduction

The problem of analysing a large amount of non-negative data collected in large sparse matrices is essential for many data mining applications, including document and image analysis, recommendations systems, neural learning process, remote sensing and object characterization, molecular pattern discovery, and so on. Some common goals can be identified in mining information stored in a non-negative data matrix: to automatically cluster similar data into groups; to retrieve items most similar to a user's query, to identify interpretable critical dimensions within the data collection. Typically, the notion of low rank approximation has played a fundamental role in processing and conceptualizing large sparse matrices effectively and efficiently. Singular value decomposition (SVD), factor analysis (FA), principal component analysis (PCA) are some examples of classical methods used to accomplish the goal of reducing the number of variables and detecting structures among the variables. However, these classical tools are not able to guarantee to maintain the non-negativity of the data matrix. Moreover, the SVD (even it constitutes the basis of the well known mechanism of Latent Semantic Indexing and Analysis) fails to provide users with any interpretation of its mathematical factors and of why it works so well. The recent approach of low-rank non-negative matrix factorization (NMF) thus becomes particularly attractive to obtain a reduced representation of data by using additive components only. This constraint have been motivated in a couple of ways. First, in many applications one knows (for example by the rules of physics) that the quantities involved cannot be negative. Second, non-negativity has been argued for based on the intuition that parts are generally combined additively (and not subtracted) to form a whole; hence, these constraints might be useful for learning parts-based representations ([9]). An interesting example of the part-based representation of the original data can be found in the context of image articulation libraries. Here, NMF are able to extract realistic parts (limbs) from image depicting stick figures with four limbs with different articulations [7].

The mathematical problem can be stated as follows: given an initial dataset expressed by a $n \times m$ matrix Y, where each column is an n-dimensional non-negative vector of the original database (m vectors), find an approximate decomposition of the data matrix into a basis matrix U and a encoding variable matrix V having non-negative elements, that is $Y \approx UV$.

The dimensions of the two non-negative factors U and V are $n \times p$ and $p \times m$, respectively. Generally the rank p of the matrices U and V is much lower than the rank of Y (usually it is chosen so that $(n + m)p < nm$). Each column of the matrix U contains a basis vector, while each column of V contains the weights needed to approximate the corresponding column in Y using the bases from U.

A NMF of the data matrix Y can be obtained by finding a solution of a nonlinear optimization problem over a specified cost function. Two simple objective functions are often used to measure the error between the original data Y and its low rank approximation UV: the sum of squared errors (or Frobenius norm), which leads

978-0-7695-3872-3/09 $26.00 © 2009 IEEE
DOI 10.1109/ISDA.2009.59

to the minimization of $\|Y - UV\|^2$ subject to the non-negativity constraints over the elements U_{ij} and V_{ij}, and the generalized Kullback-Leibler divergence to the positive matrices ($Div(Y\|UV) = \sum_{ij}(Y_{ij}\log(Y_{ij}/((UV)_{ij})) - Y_{ij}(UV)_{ij})$), subject to the non-negativity of U and V. The most popular approach to solve the NMF problem is the multiplicative update algorithm proposed in [10]. Other techniques, such as alternating nonnegative least squares method or bound-constrained optimization algorithms, such as projected gradient method, have also been used [3, 4, 12].

The NMF can be adopted in place of other factorizations, such as the singular value decomposition (SVD), because due to the non-negativity constraints it produces advantages in terms of storage and interpretability of its factor. In fact, the factors U and V are generally naturally sparse, thereby saving a great deal of storage. Moreover, they produce a so called "additive parts-based" representation of the data which allows some benefits in the interpretation since the basis vectors naturally correspond to conceptual properties of the data (for instance individual components of the structure of a picture in image recognition, or singular term in a term-by-document matrix in a text retrieval context).

Of course, NMF presents also some disadvantages concerning the lack of uniqueness of its factors and the lack of a robust computation. The problem of uniqueness of the solution can be overcome considering the orthogonal non-negative factorization (ONMF), where additionally to low-rank and non-negativity, the factor U is required to possess orthogonal columns.

The main advantages of considering ONMF are the uniqueness of the solution and the capability of clustering the rows of the data matrix in an equivalent manner to k-means clustering [6]. Of course, the computation of ONMF becomes a harder problem with respect to the standard NMF. Some modifications of the multiplicative update algorithm have been proposed in different applicative contexts (thus incorporating prior information on the problem into the estimation mechanism), but this issue lacks of a general approach [1, 11].

The aims of the present paper is to investigate the formulation and then the computational technique to obtained NMF, when the orthogonal constraint on the columns of the matrix U is added. Particularly, the orthogonal NMF can be reformulated as an optimization problem, in the Frobenius norm, on the intersection of the Stiefel manifold (the set of rectangular matrices with orthogonal columns) and the cone (with many facets) of non-negative matrices. Even the underlying geometry of the problem is easy to understand, the difficulty lies in the fact that it is hard to characterize which and when a facet of the cone of the nonnegative matrices is active or not in the optimization. As concerning the computational mechanisms, projected gradient flow approaches can be adopted which take into account the special orthogonal structure of the solution and preserve both the orthogonality and the non-negativity.

The rest of this paper is organized as follows. The next section describes how the orthogonal NMF can be formulated as an optimization problem on a intersection of the Stiefel manifold and the cone of non-negative matrices. Then, making use of the underlying geometry of the problem, we propose a penalty objective function to be minimized in order to derive a projected gradient flow whose solutions preserve both the orthogonality and the non-negativity.

2. Background and gradient flow

Let us denote by $Y \in R^{n\times m}$ a given non-negative data matrix. A Non-negative Matrix Factorization (NMF) consists of finding an approximate decomposition of the data matrix into a basis matrix and a encoding variable matrix having non-negative elements, i.e.,

$$Y \approx UV, \tag{1}$$

where the basis factor $U \in R_+^{n\times k}$ and the encoding factor $V \in R_+^{k\times m}$ (where $R_+^{p\times q}$ represents the cone of all $p \times q$ matrices whose elements are non-negative). Generally, the rank of the matrices U, V is much lower than the rank of Y (i.e., $k << \min(m, n)$).

In this paper we consider constrained NMF, and particularly, we farther require that the columns of the matrix U satisfy an orthogonality constraint, that is the basis matrix $U \in \mathcal{O}(n, k)$, where

$$\mathcal{O}(n, k) = \{Q \in R^{n\times k} | Q^\top Q = I_k\}$$

denotes the set of all real $n \times k$ matrices with orthogonal columns. This set is known as the Stiefel manifold and forms a smooth manifold.

The orthogonal non-negative matrix factorization problem (ONMF) can be defined in terms of the following cost function

$$\begin{array}{l} \min \frac{1}{2}\|Y - UV\|_F^2 \\ \text{subject to } U \in \mathcal{O}(n, k) \cap R_+^{n\times k} \text{ and } V \in R_+^{k\times m}, \end{array} \tag{2}$$

where $\|\cdot\|_F$ is the Frobenius norm on matrices defined as:

$$\|M\|_F^2 = trace(MM^\top) := \langle M, M\rangle,$$

being $\langle\cdot,\cdot\rangle$ the Frobenius inner product on $R^{n\times m}$.

The cone of non-negative matrices can be parameterized as follows

$$R_+^{p\times q} = \{S \in R^{p\times q} | S = E \odot E, \quad E \in R^{p\times q}\}, \tag{3}$$

where $\odot$ denotes the Hadamard product (i.e., the element-wise matrix multiplication).

Using (3), the ONMF of the data matrix Y can be obtained by finding a solution of the following constrained nonlinear optimization problem:

$$\begin{array}{l} \min F(Q;\Sigma) = \min \frac{1}{2}\|Y - (Q \odot Q)(\Sigma \odot \Sigma)\|_F^2 \\ \text{subject to } U = (Q \odot Q) \in \mathcal{O}(n,k) \end{array} \quad (4)$$

Note that we have also assumed $V = \Sigma \odot \Sigma$, with $\Sigma \in R^{k\times m}$.

In order to solve the above constrained optimization problem we can adopt the project gradient technique. This mechanism consists in computing the gradient of the objective function in (4) and then projecting it on the tangent space of the involved constrains. This piece of information is valuable and useful to apply many iterative scheme available in the literature. On the other hand, we find it most nature and convenient to use the dynamical system. Here we briefly summarise the main steps of the gradient flow technique, addressing the reader to [2, 5] for a complete description of the overall approach.

The main steps of the projected gradient flow technique are:

1. Compute the gradient of the objective function F in the ambient space $R^{n\times k} \times R^{k\times m}$, that is $\nabla F = (\frac{\partial F}{\partial Q}, \frac{\partial F}{\partial \Sigma})^\top$;
2. Evaluate the projections $\mathcal{P}(\frac{\partial F}{\partial Q})$ onto the tangent space of $\mathcal{O}(n,k) \cap R_+^{n\times k}$ and of $\mathcal{P}(\frac{\partial F}{\partial \Sigma})$ onto the tangent space $R^{k\times m}$;
3. Solve the dynamical system:

$$\dot{Q} = -\mathcal{P}(\frac{\partial F}{\partial Q}), \quad \dot{\Sigma} = -\mathcal{P}(\frac{\partial F}{\partial \Sigma}). \quad (5)$$

At the end of the overall process, the approximate factorization is equal to the product of the limiting solutions of (5), that is $U_\infty = Q_\infty \odot Q_\infty$ and $V_\infty = \Sigma_\infty \odot \Sigma_\infty$.

From the Riesz representation theorem with respect to the Frobenius inner product, the first component of the gradient of F can be represented as

$$\frac{\partial F}{\partial Q} = -2Q \odot (\delta(Q,\Sigma)(\Sigma \odot \Sigma)^\top), \quad (6)$$

where, for convenience, we have adopted

$$\delta(Q,\Sigma) := Y - (Q \odot Q)(\Sigma \odot \Sigma). \quad (7)$$

Similarly, the second component of the gradient is given by

$$\frac{\partial F}{\partial \Sigma} = -(2\Sigma \odot (Q \odot Q))^\tau \delta(Q,\Sigma). \quad (8)$$

Formulas (6) and (8) constitute the gradient of the objective function F. To obtained the projected gradient flow of (4), we should first note that the factor $U \in \mathcal{O}(n,k)$, therefore, $\dot{U} = \dot{Q} \odot Q + Q \odot \dot{Q}$. Moreover, by taking advantage of the product topology, the tangent space $\mathcal{T}_{(U,\Sigma)}(\mathcal{O}(n,k) \times R^{k\times m})$ of the product manifold $\mathcal{O}(n,k) \times R^{k\times m}$ at $(U,\Sigma) \in \mathcal{O}(n,k) \times R^{k\times m}$ can be decomposed as the product of tangent spaces, i.e.,

$$\mathcal{T}_{(U,\Sigma)}(\mathcal{O}(n,k) \times R^{k\times m}) = \mathcal{T}_U\mathcal{O}(n,k) \times R^{k\times m}. \quad (9)$$

The projection of $\nabla F(U,\Sigma)$ onto $\mathcal{T}_{(Q,\Sigma)}(\mathcal{O}(n,m) \times R^{k\times m})$, therefore, is the product of the projection of the $\frac{\partial F}{\partial Q}$ onto $\mathcal{T}_Q\mathcal{O}(n,k)$ and the projection of $\frac{\partial F}{\partial S}$ onto $R^{k\times m}$, respectively.

Firstly note that the projection of $\frac{\partial F}{\partial \Sigma}$ onto $R^{k\times m}$ is just itself. As concerning the projection $\frac{\partial F}{\partial U}$, it should be observed that $\mathcal{O}(n,k)$ can be embedded in the Euclidean space $R^{n\times k}$ equipped with the Frobenius inner product, hence any vector H in the tangent space $\mathcal{T}_U(\mathcal{O}(n,k))$ is of the form

$$H = UK + (I_n - UU^\top)W, \quad (10)$$

where $K \in R^{k\times k}$ and $W \in R^{n\times k}$ are arbitrary, and K is skew-symmetric. Furthermore, the space $R^{n\times k}$ can be written as the direct sum of three mutually perpendicular subspaces

$$R^{n\times k} = U\mathcal{S}(k) \oplus \mathcal{N}(U^\top) \oplus U\mathcal{S}(k)^\perp, \quad (11)$$

where $\mathcal{S}(k)$ is subspace of $k\times k$ symmetric matrices, $\mathcal{S}(k)^\perp$ is the subspace of $k \times k$ skew-symmetric matrices, and $\mathcal{N}(Q^\top) := \{X \in R^{n\times k} | U^\top X = 0\}$. Any $M \in R^{n\times k}$ can be uniquely split as

$$M = U\frac{U^\top M - M^\top U}{2} + (I - UU^\top)M + U\frac{U^\top M + M^\top U}{2}. \quad (12)$$

Hence, it follows that the projection $\mathcal{P}_{\mathcal{O}(n,k)}(M)$ of any $M \in R^{n\times k}$ onto the tangent space $\mathcal{T}_U(\mathcal{O}(n,k))$ is given by

$$\mathcal{P}_{\mathcal{O}(n,k)}(M) = U\frac{U^\top M - M^\top U}{2} + (I - UU^\top)M. \quad (13)$$

The projected gradient flow (to be numerically solved) is given by:

$$\begin{array}{l} \dot{U} = -\mathcal{P}_{\mathcal{T}_U(\mathcal{O}(n,k))}(\frac{\partial F}{\partial U}), \\ \dot{\Sigma} = \Sigma \odot (Q \odot Q)^\top \delta(Q,\Sigma). \end{array} \quad (14)$$

where $\frac{\partial F}{\partial U} = \frac{\partial F}{\partial Q}\dot{U}$ (for a more detailed formulation of the projection operator we address the reader to the description reported in [5]).

3. The penalized gradient flow approach

In order to preserve $U \in \mathcal{O}(n,k) \cap R_+^{n\times k}$ a penalty term can be added so that the functional to minimize becomes:

$$F(Q,\Sigma) = \tfrac{1}{2}\|Y - (Q\odot Q)(\Sigma\odot\Sigma)\|_F^2 + \tfrac{1}{2}P(Q) \tag{15}$$

where, $U = Q\odot Q$, $V = \Sigma\odot\Sigma$ and the penalty term is given by

$$P(Q) = \|(Q\odot Q)^\top (Q\odot Q) - I\|_F^2. \tag{16}$$

From the Riesz representation theorem, with respect to the Frobenius inner product, the Fréchet derivative of P at Q is given by

$$\frac{\partial P}{\partial Q} = 2[(Q\odot Q)\Delta\odot Q + Q\odot(Q\odot Q)\Delta), \tag{17}$$

where the additional matrix function Δ is defined as

$$\Delta = [(Q\odot Q)^\top (Q\odot Q) - I].$$

The explicit formulation of the gradient flow obtained by using the penalty function (15) is given by

$$\begin{aligned} \dot{Q} &= Q\odot[\delta(Q,\Sigma)(\Sigma\odot\Sigma)^\top] - 1/2\tfrac{\partial P}{\partial Q}, \\ \dot{\Sigma} &= \Sigma\odot[(Q\odot Q)^\top \delta(Q,\Sigma)]. \end{aligned} \tag{18}$$

The flow (18), which we shall call the penalised flow for later reference, moves along the steepest descent direction to minimize the objective functional F in (15). Since (18) defines a descent flow by an analytic vector field, it is known by the Lojasiewicz theorem that the flow converges to a single point $(Q;U)$ of equilibrium at which

$$U = Q\odot Q \qquad \text{and} \qquad V = \Sigma\odot\Sigma, \tag{19}$$

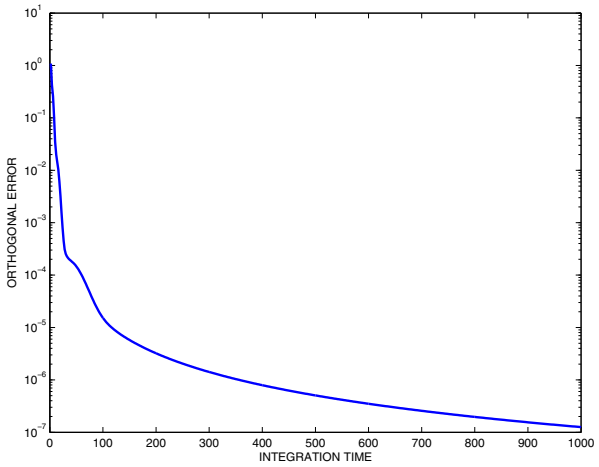
is locally the best non-negative matrix factorization of Y subject to orthogonal constraint on the columns of U.

It is important to note that the cost function $F(Q,\Sigma)$ is convex in each of the factors Q and Σ, but it is not convex in the two factors at the same time. Hence, important challenges affecting the numerical solution of the penalized dynamical system flow (18) include the existence of different local minima due to the non-convexity of the objective function. Therefore, the "locally" best non-negative matrix factorization in (19) means that the point $(Q;U)$ may be a local minima of (18).

4. Numerical Experiments

In this section we report some experimental results from using the above-mentioned dynamical system. At the moment, our primary concern is not so much on the efficiency of this mechanism. Rather, we focus on the behavior of the resulting flow from this differential system. For the purpose of demonstration, we shall employ existing routines in Matlab as the ODE integrators. It is understood that many other ODE solvers, especially the recently developed geometric integrators, can be used as well. The ODE Suite [13] in Matlab contains in particular a Klopfenstein-Shampine, quasi-constant step size, stiff system solver ode15s. Assuming the original data matrix Y is not precise in its own right in practice, high accuracy approximation of Y is not needed. We set both local tolerance $AbsTol = RelTol = 10^{-6}$ while maintaining all other parameters at the default values of the Matlab codes. The numerical tests have been conducted using non-negative matrices Y randomly generated. The initial value for the the penalised flow are a non-negative randomly generated matrix Σ_0 and a random perturbation of a permutation matrix U_0. We measure the orthogonality of bases by $\|UU^\top - I\|$ and GOF by $\|Y - UV\|_2$. Figure 1 shows the decreasing behaviour of the orthogonal error during the integration of the penalized flow for a 20×10 example approximated by rank 5 matrices in the time-interval $[0, 1000]$. Figure 2, instead, plots the behaviour of the goodness-of-fit (GOF) for the same example during the first part of the numerical integration, i.e. in the time interval $[0, 100]$ where the GOF-values have been stabilized.

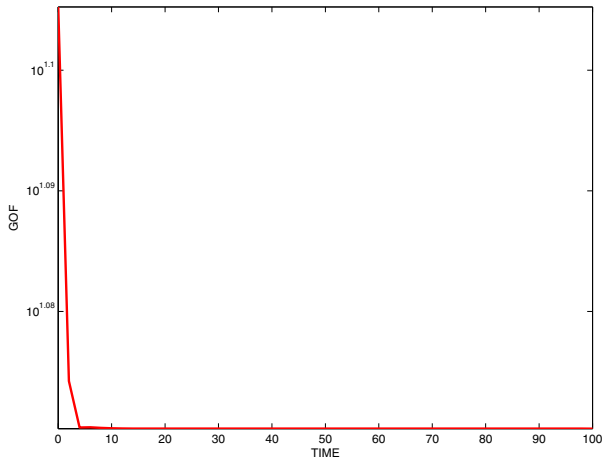
Figure 1. Orthogonal Error during the integration of penalized flow



5. Conclusions and Future Work

We have presented a penalised projected gradient flow for orthogonal NMF where orthogonality on the non-negative basis matrix vectors is imposed in learning the de-

Figure 2. GOF during the integration of penalized flow



composition. The core idea was to directly use the projected gradient in Stiefel manifold in order to obtain a dynamical system, whose solutions evolve on the intersection of the Stiefel manifold and the cone of non-negative matrices. Moreover, bearing in mind that we adopted a continuous technique, based on the solution of a dynamical system, additional feature information can be obtained for the approximation matrices: this represents a clear advantage when comparing our approach with other standard methods.

An interesting issue, strictly tied with the computation of the orthogonal NMF when the adopted cost function is the generalized KL-divergence, is the connections with some family of probabilistic latent variable models. Particularly, in [6], it has been pointed out that the objective function of a probabilistic latent semantic indexing model is the same of the objective function of NMF with an additional orthogonal constraint. It should be of interest to further explore this relationship and also the implication that the adoption of continuous computational techniques for ONMF could have in treating texts analysis task [8].

References

[1] S. Choi. Algorithms for orthogonal nonnegative matrix factorization. In *Proceedings of the IJCNN 2008*, 2008.

[2] M. Chu. Group theory, linear transformations, and flows: Dynamical systems on manifolds. Technical report, Preprint NCSU, 2004.

[3] M. Chu, F. Diele, R. Plemmons, and S. Ragni. Optimality, computation and interpretation of nonnegative matrix factorizations. Technical report, Preprint NCSU, 2005.

[4] M. Chu and M. Lin. Low dimensional polytope approximation and its applications to nonnegative matrix factorization. *SIAM J. Sci. Comput.*, (30):1131–1151, 2008.

[5] N. Del Buono and S. Fiori. Gradient flow for orthogonal non-negative matrix factorization. Technical report, Department of Mathematics, University of Bari, Italy, 2009.

[6] C. Ding, T. Li, and W. Peng. Nonnegative matrix factorizations and probabilistic semantic indexing: Equivalence, chi-square statistic, and hybrid method. In *Proceeding of National Conference on Artificial Intelligence, (AAAI-06)*, 2006.

[7] D. Donoho and V. Stodden. When does non-negative matrix factorization give correct decomposition into parts? In *Advances in Neural Information Processing Systems*, 2003.

[8] E. Gaussier and C. Goutte. Relation between plsa and nmf and implications. In *Proceedings of SIGIR05, August 15-19, 2005, Salvador Brazil*, 2005.

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

[10] D. Lee and H. Seung. Algorithms for non-negative matrix factorization. *Advances in Neural Information Processing Systems*, 13:788–791, 2001.

[11] H. Li, T. Adali, W. Wang, D. Emge, and A. Cichocki. Non-negative matrix factorization with orthogonality constraints and its application to raman spectroscopy. *J. VLSI Signal Processing*, 2007.

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

[13] L. Shampine and M. W. Reichelt. The matlab ode suite. *SIAM J. Sci. Comput.*, (18):1–22, 1997.