Coordinate Descent for mixed-norm NMF

Vamsi K. Potluru
Department of Computer Science,
University of New Mexico
ismav@cs.unm.edu

Jonathan Le Roux
Mitsubishi Electric Research Labs,
Cambridge, Massachusetts
leroux@merl.com

Barak A. Pearlmutter
Department of Computer Science,
National University of Ireland Maynooth
barak@cs.nuim.ie

John R. Hershey
Mitsubishi Electric Research Labs,
Cambridge, Massachusetts
hershey@merl.com

Matthew E. Brand
Mitsubishi Electric Research Labs,
Cambridge, Massachusetts
brand@merl.com

Abstract

Nonnegative matrix factorization (NMF) is widely used in a variety of machine learning tasks involving speech, documents and images. Being able to specify the structure of the matrix factors is crucial in incorporating prior information. The factors correspond to the feature matrix and the learnt representation. In particular, we allow an user-friendly specification of sparsity on the groups of features using the L_1/L_2 measure. Also, we propose a pairwise coordinate descent algorithm to minimize the objective. Experimental evidence of the efficacy of this approach is provided on the ORL faces dataset.

1 Introduction

Nonnegative matrix factorization is useful for extracting latent features [5], a problem which arises in a wide range of application domains, including such diverse areas as image analysis [8], document clustering, recommender systems, and many others [1, 2]. Given a nonnegative matrix $\mathbf{X}$ of size $m \times n$, we want to approximate it as the product of two nonnegative matrices $\mathbf{W}$ and $\mathbf{H}$ of sizes $m \times r$ and $r \times n$, respectively:

$$\mathbf{X} \approx \mathbf{WH}.$$

Sparsity can be specified by the L_0-norm but it is usually difficult to optimize directly. L_1 norm is used as a proxy for sparsity but it is not scale-invariant. So, the following measure of sparsity has been proposed in the literature [5, 10]:

$$\text{sp}(\boldsymbol{W}) = \frac{\sqrt{mr} - \|\boldsymbol{W}\|_1/\|\boldsymbol{W}\|_F}{\sqrt{mr} - 1} \tag{1}$$

This measure is always between zero and one, with higher values corresponding to sparser solutions, and has also been shown to satisfy many appealing properties [6], one such being invariance to scaling.

Here, we consider an explicit sparse NMF formulation which sets a hard constraint on $\text{sp}(\boldsymbol{W})$ building on the sparse model byHoyer [5]. Previous work required that the sparsity be set for each feature

individually [5, 4, 11]. However, we would like to be able to set a sparsity budget for groups of features and allow the individual features to adapt their sparsity which best suits the data. Without loss of generality, let us assume that we wish for the "feature" matrix $\boldsymbol{W}$ to be sparse—analysis for the symmetric case where we wish for $\boldsymbol{H}$ to be sparse is analogous. A formulation which restricts the norms of the features to unity can be utilized to enforce the sparsity measure (1) using the following sparse NMF objective:

$$\begin{aligned} \min_{\mathbf{W},\mathbf{H}} f(\boldsymbol{W},\boldsymbol{H}) = \frac{1}{2}\|\mathbf{X}-\mathbf{W}\mathbf{H}\|_F^2 \quad \text{s.t.} \quad & \mathbf{W}\geq\mathbf{0}, \mathbf{H}\geq\mathbf{0} \\ & \|\boldsymbol{W}_i\|_2 = 1 \qquad \forall\, i\in\{1,\ldots,r\}, \\ & \sum_{i\in\boldsymbol{I}_g}\|\boldsymbol{W}_i\|_1 = \alpha_g \qquad \forall\, g\in\{1,\ldots,G\} \end{aligned} \tag{2}$$

where g indexes the partition set of the columns of $\boldsymbol{W}$ into the corresponding G groups and $\{\boldsymbol{I}_1,\ldots,\boldsymbol{I}_G\}$ is a partition of $\{1,\ldots,r\}$. A similar formulation for sparse NMF [10] did not enforce norm constraints at the feature level. Another, closely related implicit mixed-norm formulation of sparse NMF has also been recently introduced [7].

We will follow the usual alternating updates strategy to solve for the matrices $\boldsymbol{W},\boldsymbol{H}$. That is we optimize for one of the matrices while keeping the other fixed. The updates for the matrix $\boldsymbol{H}$ are the same as in NMF and standard solvers can be applied such as those based on multiplicative updates [8] or projected gradient[9]. Therefore, we will solely focus on the optimization problem in the matrix $\boldsymbol{W}$.

Similar to the algorithm presented by Hoyer [5], we could apply standard gradient descent followed by a projection step to satisfy the norm constraints. However, projection algorithms are generally inefficient. Another approach would be to attempt a Frank-Wolfe type algorithm [3] by considering a linear approximation to the objective function. These algorithms generally have a sub-linear rate of convergence. A recent work based on block coordinate descent [11] proposed updating one feature at a time. This would not work in our case since the L_1 constraint is across the features in a group and updating features independently will freeze the L_1 constraint to the initial feasible solution. Instead, we propose to optimize over two features of a group at a time. This would allow the L_1 norm to mix across the features while maintaining the L_2 norm constraints.

2 Proposed Algorithm for Sparse NMF

We will show how the structure of the problem can be exploited to make coordinate descent a very efficient linear operator.

2.1 Pairwise Coordinate Descent

Let us denote a subset of m rows by index set $\boldsymbol{A}$ and the corresponding elements of $\boldsymbol{W}_i$ by $\boldsymbol{W}_i^A$. Choose $\boldsymbol{A},\boldsymbol{B}$ to be two non-overlapping subsets of m rows. Observe that the objective (2) reduces to the following linear function while fixing the rest of the elements of matrices $\mathbf{W},\mathbf{H}$:

$$\begin{aligned} \min_{\boldsymbol{W}_i^A\geq\mathbf{0},\boldsymbol{W}_j^B\geq\mathbf{0}} \tilde{f}(\boldsymbol{W}_i^A,\boldsymbol{W}_j^B) &= \frac{1}{2}d_i\|\boldsymbol{W}_i^A\|_2^2 + \boldsymbol{u}_i^\top\boldsymbol{W}_i^A + \frac{1}{2}d_j\|\mathbf{W}_j\|_2^2 + \boldsymbol{u}_j^\top\boldsymbol{W}_j^B \\ &= \begin{bmatrix}\boldsymbol{u}_i^\top & \boldsymbol{u}_j^\top\end{bmatrix}\begin{bmatrix}\boldsymbol{W}_i^A \\ \boldsymbol{W}_j^B\end{bmatrix} + \text{constant} \\ &\text{s.t. } \|\boldsymbol{W}_i^A\|_2=\gamma,\ \|\boldsymbol{W}_j^B\|_2=\delta,\ \text{and } \|\boldsymbol{W}_i^A\|_1+\|\boldsymbol{W}_j^B\|_1=\beta \end{aligned} \tag{3}$$

where $d_i = \boldsymbol{H}_i^\top\boldsymbol{H}_i$ and $\boldsymbol{u}_i = [-\boldsymbol{X}\boldsymbol{H}_i^\top + \sum_{l\neq i}\boldsymbol{W}_l(\boldsymbol{H}\boldsymbol{H}^\top)_{li}]^A$ (similarly for index j) and γ,δ,β are initialized from the previous run.

We can reduce it to the following projection problem:

$$\max_{\boldsymbol{y}\geq 0}\boldsymbol{b}^\top\boldsymbol{y} \qquad \text{s.t.} \quad \mathbf{1}^\top\boldsymbol{y}=k, \qquad \|\boldsymbol{y}_1\|_2=1, \qquad \|\boldsymbol{y}_2\|_2=1 \tag{4}$$

where $\boldsymbol{y}=[\boldsymbol{y}_1^\top,\boldsymbol{y}_2^\top]^\top$ and $\boldsymbol{b}=[\boldsymbol{b}_1^\top,\boldsymbol{b}_2^\top]^\top$ and $\dim(\boldsymbol{b}_1)=m_1, \dim(\boldsymbol{b}_2)=m_2$. We solve it approximately by the following projection Algorithm 1. It uses the Sparse-opt routine (Algorithm 3) [11] for efficiently projecting onto the intersection of $\mathrm{L}_1/\mathrm{L}_2$ constraints.

Algorithm 1 Sparse-biopt($\boldsymbol{b}, k, m_1$)

for i = 1 to Δ **do**
  Obj(i)= Sparse-opt($\boldsymbol{b}_1$, $\frac{k}{\Delta}i$)+ Sparse-opt($\boldsymbol{b}_2, k - \frac{k}{\Delta}i$)
end for
$i^* = \min_i \text{Obj}(i)$
$\boldsymbol{y}_1^* = $ Sparse-opt($\boldsymbol{b}_1$, $\frac{k}{\Delta}i^*$)
$\boldsymbol{y}_2^* = $ Sparse-opt($\boldsymbol{b}_1, k - \frac{k}{\Delta}i^*$)
$\boldsymbol{y}^* = [\boldsymbol{y_1^{*\top}}, \boldsymbol{y_2^{*\top}}]^\top$

2.2 Matrix Update

Without loss of generality, let us assume that there are at least two columns per group. If there is only a single column in a particular group, we can just apply Algorithm 3 to update the group. In practice, we also update each feature as if it belonged to a group of unit size which helps in learning smooth features. We optimize for the matrix $\boldsymbol{W}$ as shown in Algorithm 2.

Algorithm 2 Update($\boldsymbol{X}, \boldsymbol{W}, \boldsymbol{H}$)

$\boldsymbol{C} = -\boldsymbol{X}\boldsymbol{H}^\top + \boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^\top$
$\boldsymbol{Z} = \boldsymbol{H}\boldsymbol{H}^\top$
repeat
  for g = 1 to $\boldsymbol{G}$ (randomly) **do**
    for i,j in g **do**
      Pick two random non-overlapping sets of rows $\boldsymbol{A}, \boldsymbol{B}$.
      $\boldsymbol{U}_i^A = \boldsymbol{C}_i^A - \boldsymbol{W}_i^A \boldsymbol{Z}_{ii}$
      $\boldsymbol{U}_j^A = \boldsymbol{C}_j^B - \boldsymbol{W}_j^B \boldsymbol{Z}_{jj}$
      Obtain new values of $\boldsymbol{U}_i^A, \boldsymbol{U}_j^B$ by Sparse-biopt. Denote the output by $\boldsymbol{t}_i, \boldsymbol{t}_j$.
      $\boldsymbol{C}^A = \boldsymbol{C}^A + (\boldsymbol{t}_i - \boldsymbol{W}_i^A)\boldsymbol{Z}_i^\top$
      $\boldsymbol{C}^B = \boldsymbol{C}^B + (\boldsymbol{t}_j - \boldsymbol{W}_j^B)\boldsymbol{Z}_j^\top$
    end for
  end for
until convergence

2.3 Sparse Projection onto the L_1/L_2 Constraint

We review the sparse projection method of Potluru et al. [11], which solves a linear optimization on the intersection of a hyperplane and a hypersphere. Consider the following subproblem which arises when solving (2):

$$\max_{\boldsymbol{y} \geq 0} \boldsymbol{b}^\top \boldsymbol{y} \qquad \text{s.t.} \qquad \mathbf{1}^\top \boldsymbol{y} = k, \qquad \|\boldsymbol{y}\|_2 = 1 \tag{5}$$

where $\dim(\boldsymbol{b}) = m$.

The following two algorithms Sparse-trans and Sparse-opt together compute the solution for any given k.

For a given $k^\dagger$, we can compute the objective after running Algorithm 3. Note that the running time of Algorithm 4 is $O(n \log n)$. We could have utilized a more efficient linear time algorithm [12]. However, the benefit of the presented projection is that it computes the full solution path over the parameter k efficiently.

3 Experiments

Images from the ORL faces dataset were obtained (`http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html`). It consists of 400 images of size 112×92. We obtain a feasible solution to our problem by applying the Sparse-opt routine for each column with $\boldsymbol{b}$ initialized to a random vector and k set to α_g/dim($\boldsymbol{I}_g$) of the corresponding

Algorithm 3 Sparse-trans($\boldsymbol{b}$)

Set $\boldsymbol{a} = \text{sort}(\boldsymbol{b})$ and $p^* = m$. Obtain a mapping π such that $a_i = b_{\pi(i)}$ and $a_j \geq a_{j+1}$ for all valid i, j. Denote the cumulative sums as follows: $s(p) = \sum_{i=0}^{p} a_i$ and $t(p) = \sum_{i=0}^{p} a_i^2$.
Compute values of $\mu(p), \lambda(p)$ as follows:
for p = 1 to $m - 1$ **do**
    $\lambda(p) = \sqrt{(t(p) + (p+1)a(p)^2 - 2a(p)s(p)}$
    $\mu(p) = -a(p)$
    $obj(p) = (t(p) - a(p)s(p))/\lambda(p)$
    $k(p) = (s(p) - (p+1)a(p))/\lambda(p)$
end for

Algorithm 4 Sparse-opt($\boldsymbol{b}, k^\dagger$)

Run the Sparse-trans routine (Algorithm 3) on $\boldsymbol{b}$ if its output is not cached.
Find $p*$ such that $k^\dagger \in (k(p^*), k(p^*+1))$. the corresponding solution vector $\boldsymbol{y}^*$ can be derived as follows:
$\lambda = -\sqrt{\frac{(p^*+1)t(p) - s(p^*)^2}{(p^*+1-k^{\dagger 2})}}$
$\mu = -\frac{s(p^*)+k^\dagger \lambda}{\lambda}$
$\text{obj} = -\frac{\mu \boldsymbol{s} + \boldsymbol{t}}{\lambda}$
$x_i = -\frac{a_i + \mu}{\lambda} \qquad \forall i \in \{0, \cdots, p^*\}$ and zero otherwise.
$y^*_{\pi(i)} = x_i \qquad \forall i \in \{0, \cdots, m-1\}$

group sparsity parameter. Next, we ran our proposed algorithm on the dataset in 3 different settings. The first two settings involved setting the average sparsity value across all features to $0.4, 0.6$. In the third run, we divided the features into 3 groups of sizes $\{5, 15, 5\}$ with their group sparsity set to $\{0.2, 0.5, 0.8\}$ respectively. The resulting features are shown in Figure 1.

4 Conclusion and Future Work

We can learn structurally rich models via mixed norms by paying a small price computationally. This enables one to specify models which are closer to user requirements. In this paper, we proposed to solve the sparse NMF problem when the mean sparsity was provided on groups of features.

We used a two-column update strategy to learn the sparsity patterns. This can be extended to more columns by using dynamic programming. Also, we would like to analyze the theoretical properties of our algorithm in terms of convergence and running times and apply them to a wider variety of datasets. We considered only the intersection of L_1/L_2 balls and it would be interesting to extend it to more general norm-balls [12]. Finally, we would like to parallelize the updates in the multi-core/GPU settings.

Acknowledgement

The first author would like to thank the support from MERL.

References

[1] Sanjeev Arora, Rong Ge, Ravindran Kannan, and Ankur Moitra. Computing a nonnegative matrix factorization – provably. In *Proceedings of the 44th symposium on Theory of Computing*, STOC '12, pages 145–162, New York, NY, USA, 2012. ACM. ISBN 9781-450-3124-5-5. doi: 10.1145/2213977.2213994. URL `http://doi.acm.org/10.1145/2213977.2213994`.

[2] Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan, and Shun-ichi Amari. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. Wiley, 2009.

Figure 1: 25 features learnt from the dataset with the overall mean sparsity value set to 0.4 (left), 0.6 (center). 3 groups with $\{5, 15, 5\}$ features were learnt with sparsities of $\{0.2, 0.5, 0.8\}$ respectively. Notice that the groups help us to learn a mixture of global, mid-level and sparse features.

[3] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, 1956. ISSN 1931-9193. doi: 10.1002/nav.3800030109. URL http://dx.doi.org/10.1002/nav.3800030109.

[4] Matthias Heiler and Christoph Schnörr. Learning sparse representations by non-negative matrix factorization and sequential cone programming. *The Journal of Machine Learning Research*, 7:2006, 2006. ISSN 1532-4435.

[5] Patrik O. Hoyer. Non-negative matrix factorization with sparseness constraints. *J. Mach. Learn. Res.*, 5:1457–1469, December 2004. ISSN 1532-4435. URL http://portal.acm.org/citation.cfm?id=1005332.1044709.

[6] Niall Hurley and Scott Rickard. Comparing measures of sparsity. *IEEE Trans. Inf. Theor.*, 55: 4723–4741, October 2009. ISSN 0018-9448. doi: 10.1109/TIT.2009.2027527. URL http://portal.acm.org/citation.cfm?id=1669375.1669404.

[7] Jingu Kim, Renato Monteiro, and Haesun Park. Group sparsity in nonnegative matrix factorization. In *SDM*, pages 851–862, 2012.

[8] Daniel D. Lee and Sebastian H. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562.

[9] Chih-Jen Lin. Projected gradient methods for nonnegative matrix factorization. *Neural Comp.*, 19(10):2756–2779, October 2007. URL http://neco.mitpress.org/cgi/content/abstract/19/10/2756.

[10] Morten Mørup, Kristoffer Hougaard Madsen, and Lars Kai Hansen. Approximate L0 constrained non-negative matrix and tensor factorization. In *ISCAS*, pages 1328–1331, 2008.

[11] Vamsi K. Potluru, Sergey M. Plis, Jonathan Le Roux, Barak A. Pearlmutter, Vince D. Calhoun, and Thomas P. Hayes. Block coordinaate descent for sparse NMF. In *Proc. International Conference on Learning Representations (ICLR)*, May 2013.

[12] Adams Wei Yu, Hao Su, and Fei-Fei Li. Efficient euclidean projections onto the intersection of norm balls. In *ICML*. icml.cc / Omnipress, 2012. URL http://dblp.uni-trier.de/db/conf/icml/icml2012.html#YuSL12a.