Journal of Machine Learning Research 5 (2004) 1457–1469 Submitted 8/04; Published 11/04

Non-negative Matrix Factorization with Sparseness Constraints

Patrik O. Hoyer PATRIK.HOYER@HELSINKI.FI
HIIT Basic Research Unit
Department of Computer Science
P.O. Box 68, FIN-00014
University of Helsinki
Finland

Editor: Peter Dayan

Abstract

Non-negative matrix factorization (NMF) is a recently developed technique for finding parts-based, linear representations of non-negative data. Although it has successfully been applied in several applications, it does not always result in parts-based representations. In this paper, we show how explicitly incorporating the notion of 'sparseness' improves the found decompositions. Additionally, we provide complete MATLAB code both for standard NMF and for our extension. Our hope is that this will further the application of these methods to solving novel data-analysis problems.

Keywords: non-negative matrix factorization, sparseness, data-adaptive representations

1. Introduction

A fundamental problem in many data-analysis tasks is to find a suitable representation of the data. A useful representation typically makes latent structure in the data explicit, and often reduces the dimensionality of the data so that further computational methods can be applied.

Non-negative matrix factorization (NMF) (Paatero and Tapper, 1994; Lee and Seung, 1999) is a recent method for finding such a representation. Given a non-negative data matrix $\mathbf{V}$, NMF finds an approximate factorization $\mathbf{V} \approx \mathbf{WH}$ into non-negative factors $\mathbf{W}$ and $\mathbf{H}$. The non-negativity constraints make the representation purely additive (allowing no subtractions), in contrast to many other linear representations such as principal component analysis (PCA) and independent component analysis (ICA) (Hyvärinen et al., 2001).

One of the most useful properties of NMF is that it usually produces a *sparse* representation of the data. Such a representation encodes much of the data using few 'active' components, which makes the encoding easy to interpret. Sparse coding (Field, 1994) has also, on theoretical grounds, been shown to be a useful middle ground between completely distributed representations, on the one hand, and unary representations (grandmother cells) on the other (Földiák and Young, 1995; Thorpe, 1995). However, because the sparseness given by NMF is somewhat of a side-effect rather than a goal, one cannot in any way control the degree to which the representation is sparse. In many applications, more direct control over the properties of the representation is needed.

In this paper, we extend NMF to include the option to control sparseness explicitly. We show that this allows us to discover parts-based representations that are qualitatively better than those given

by basic NMF. We also discuss the relationship between our method and other recent extensions of NMF (Li et al., 2001; Hoyer, 2002; Liu et al., 2003).

Additionally, this contribution includes a complete MATLAB package for performing NMF and its various extensions. Although the most basic version of NMF requires only two lines of code and certainly does not warrant distributing a separate software package, its several extensions involve more complicated operations; the absense of ready-made code has probably hindered their widespread use so far. We hope that our software package will alleviate the problem.

This paper is structured as follows. In Section 2 we describe non-negative matrix factorization, and discuss its success but also its limitations. Section 3 discusses why and how to incorporate sparseness constraints into the NMF formulation. Section 4 provides experimental results that verify our approach. Finally, Sections 5 and 6 compare our approach to other recent extensions of NMF and conclude the paper.

2. Non-negative Matrix Factorization

Non-negative matrix factorization is a *linear*, *non-negative* approximate data representation. Let's assume that our data consists of T measurements of N non-negative scalar variables. Denoting the (N-dimensional) measurement vectors $\mathbf{v}^t$ ($t = 1, \ldots, T$), a linear approximation of the data is given by

$$\mathbf{v}^t \approx \sum_{i=1}^{M} \mathbf{w}_i h_i^t = \mathbf{W}\mathbf{h}^t,$$

where $\mathbf{W}$ is an $N \times M$ matrix containing the *basis vectors* $\mathbf{w}_i$ as its columns. Note that each measurement vector is written in terms of the *same* basis vectors. The M basis vectors $\mathbf{w}_i$ can be thought of as the 'building blocks' of the data, and the (M-dimensional) coefficient vector $\mathbf{h}^t$ describes how strongly each building block is present in the measurement vector $\mathbf{v}^t$.

Arranging the measurement vectors $\mathbf{v}^t$ into the columns of an $N \times T$ matrix $\mathbf{V}$, we can now write

$$\mathbf{V} \approx \mathbf{W}\mathbf{H},$$

where each column of $\mathbf{H}$ contains the coefficient vector $\mathbf{h}^t$ corresponding to the measurement vector $\mathbf{v}^t$. Written in this form, it becomes apparent that a linear data representation is simpy a factorization of the data matrix. Principal component analysis, independent component analysis, vector quantization, and non-negative matrix factorization can all be seen as matrix factorization, with different choices of objective function and/or constraints.

Whereas PCA and ICA do not in any way restrict the signs of the entries of $\mathbf{W}$ and $\mathbf{H}$, NMF requires all entries of both matrices to be non-negative. What this means is that the data is described by using additive components only. This constraint has 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. In such cases, it can be difficult to interpret the results of PCA and ICA (Paatero and Tapper, 1994; Parra et al., 2000). 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 (Lee and Seung, 1999).

Given a data matrix $\mathbf{V}$, the optimal choice of matrices $\mathbf{W}$ and $\mathbf{H}$ are defined to be those non-negative matrices that minimize the reconstruction error between $\mathbf{V}$ and $\mathbf{WH}$. Various error functions have been proposed (Paatero and Tapper, 1994; Lee and Seung, 2001), perhaps the most widely

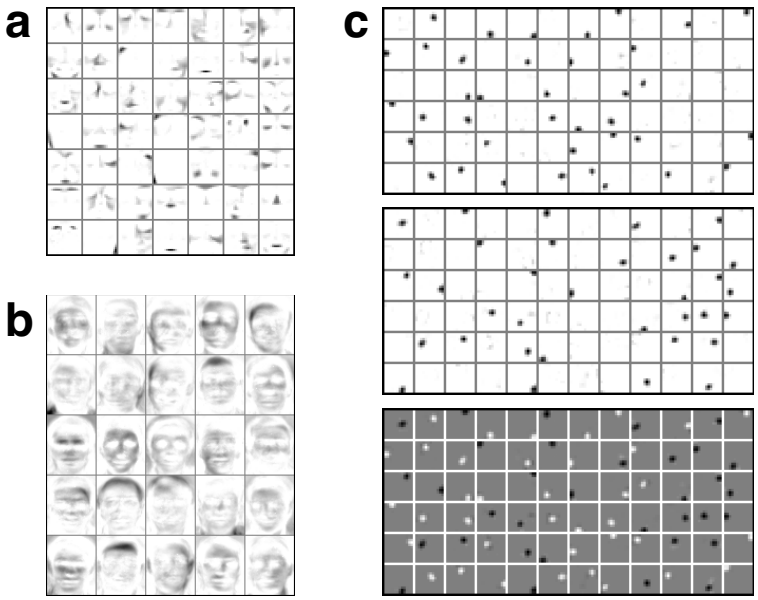

Figure 1: NMF applied to various image data sets. **(a)** Basis images given by NMF applied to face image data from the CBCL database (`http://cbcl.mit.edu/cbcl/software-datasets/FaceData2.html`), following Lee and Seung (1999). In this case NMF produces a parts-based representation of the data. **(b)** Basis images derived from the ORL face image database (`http://www.uk.research.att.com/facedatabase.html`), following Li et al. (2001). Here, the NMF representation is global rather than parts-based. **(c)** Basis vectors from NMF applied to ON/OFF-contrast filtered natural image data (Hoyer, 2003). Top: Weights for the ON-channel. Each patch represents the part of one basis vector $\mathbf{w}_i$ corresponding to the ON-channel. (White pixels denote zero weight, darker pixels are positive weights.) Middle: Corresponding weights for the OFF-channel. Bottom: Weights for ON minus weights for OFF. (Here, gray pixels denote zero.) Note that NMF represents this natural image data using circularly symmetric features.

used is the squared error (euclidean distance) function

$$E(\mathbf{W},\mathbf{H}) = \|\mathbf{V}-\mathbf{WH}\|^2 = \sum_{i,j}(V_{ij}-(\mathbf{WH})_{ij})^2.$$

Although the minimization problem is convex in $\mathbf{W}$ and $\mathbf{H}$ separately, it is not convex in both simultaneously. Paatero and Tapper (1994) gave a gradient algorithm for this optimization, whereas Lee and Seung (2001) devised a multiplicative algorithm that is somewhat simpler to implement and also showed good performance.

Although some theoretical work on the properties of the NMF representation exists (Donoho and Stodden, 2004), much of the appeal of NMF comes from its empirical success in learning meaningful features from a diverse collection of real-life data sets. Lee and Seung (1999) showed that, when the data set consisted of a collection of face images, the representation consisted of

basis vectors encoding for the mouth, nose, eyes, etc; the intuitive features of face images. In Figure 1a we have reproduced that basic result using the same data set. Additionally, they showed that meaningful topics can be learned when text documents are used as data. Subsequently, NMF has been successfully applied to a variety of data sets (Buchsbaum and Bloch, 2002; Brunet et al., 2004; Jung and Kim, 2004; Kim and Tidor, 2003).

Despite this success, there also exist data sets for which NMF does not give an intuitive decomposition into parts that would correspond to our idea of the 'building blocks' of the data. Li et al. (2001) showed that when NMF was applied to a different facial image database, the representation was global rather than local, qualitatively different from that reported by Lee and Seung (1999). Again, we have rerun that experiment and confirm those results, see Figure 1b. The difference was mainly attributed to how well the images were hand-aligned (Li et al., 2001).

Another case where the decomposition found by NMF does not match the underlying elements of the data is shown in Figure 1c. In this experiment (Hoyer, 2003), natural image patches were high-pass filtered and subsequently split into positive ('ON') and negative ('OFF') contrast channels, in a process similar to how visual information is processed by the retina. When NMF is applied to such a data set, the resulting decomposition does not consist of the oriented filters which form the cornerstone of most of modern image processing. Rather, NMF represents these images using simple, dull, circular 'blobs'.

We will show that, in both of the above cases, explicitly controlling the sparseness of the representation leads to representations that are parts-based and match the intuitive features of the data.

3. Adding Sparseness Constraints to NMF

In this section, we describe the basic idea of sparseness, and show how to incorporate it into the NMF framework.

3.1 Sparseness

The concept of 'sparse coding' refers to a representational scheme where only a few units (out of a large population) are effectively used to represent typical data vectors (Field, 1994). In effect, this implies most units taking values close to zero while only few take significantly non-zero values. Figure 2 illustrates the concept and our sparseness measure (defined below).

Numerous sparseness measures have been proposed and used in the literature to date. Such measures are mappings from $\mathbb{R}^n$ to $\mathbb{R}$ which quantify how much energy of a vector is packed into only a few components. On a normalized scale, the sparsest possible vector (only a single component is non-zero) should have a sparseness of one, whereas a vector with all elements equal should have a sparseness of zero.

In this paper, we use a sparseness measure based on the relationship between the L_1 norm and the L_2 norm:

$$\text{sparseness}(\mathbf{x}) = \frac{\sqrt{n} - (\sum |x_i|) / \sqrt{\sum x_i^2}}{\sqrt{n} - 1},$$

where n is the dimensionality of $\mathbf{x}$. This function evaluates to unity if and only if $\mathbf{x}$ contains only a single non-zero component, and takes a value of zero if and only if all components are equal (up to signs), interpolating smoothly between the two extremes.

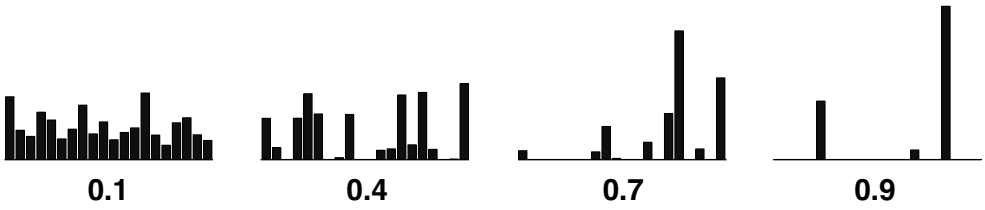

Figure 2: Illustration of various degrees of sparseness. Four vectors are shown, exhibiting sparseness levels of 0.1, 0.4, 0.7, and 0.9. Each bar denotes the value of one element of the vector. At low levels of sparseness (leftmost), all elements are roughly equally active. At high levels (rightmost), most coefficients are zero whereas only a few take significant values.

3.2 NMF with Sparseness Constraints

Our aim is to constrain NMF to find solutions with desired degrees of sparseness. The first question to answer is then: what exactly should be sparse? The basis vectors $\mathbf{W}$ or the coefficients $\mathbf{H}$? This is a question that cannot be given a general answer; it all depends on the specific application in question. Further, just transposing the data matrix switches the role of the two, so it is easy to see that the choice of which to constrain (or both, or none) must be made by the experimenter.

For example, a doctor analyzing disease patterns might assume that most diseases are rare (hence sparse) but that each disease can cause a large number of symptoms. Assuming that symptoms make up the rows of her matrix and the columns denote different individuals, in this case it is the 'coefficients' which should be sparse and the 'basis vectors' unconstrained. On the other hand, when trying to learn useful features from a database of images, it might make sense to require both $\mathbf{W}$ and $\mathbf{H}$ to be sparse, signifying that any given object is *present* in few images and *affects* only a small part of the image.

These considerations lead us to defining NMF with sparseness constraints as follows:

Definition: NMF with sparseness constraints

Given a non-negative data matrix $\mathbf{V}$ of size $N \times T$, find the non-negative matrices $\mathbf{W}$ and $\mathbf{H}$ of sizes $N \times M$ and $M \times T$ (respectively) such that

$$E(\mathbf{W},\mathbf{H}) = \|\mathbf{V} - \mathbf{W}\mathbf{H}\|^2 \tag{1}$$

is minimized, under *optional constraints*

$$\begin{aligned} \text{sparseness}(\mathbf{w}_i) &= S_w, \ \forall i \\ \text{sparseness}(\mathbf{h}_i) &= S_h, \ \forall i, \end{aligned}$$

where $\mathbf{w}_i$ is the i:th *column* of $\mathbf{W}$ and $\mathbf{h}_i$ is the i:th *row* of $\mathbf{H}$. Here, M denotes the number of components, and S_w and S_h are the desired sparsenesses of $\mathbf{W}$ and $\mathbf{H}$ (respectively). These three parameters are set by the user.

Note that we did not constrain the scales of $\mathbf{w}_i$ or $\mathbf{h}_i$ yet. However, since $\mathbf{w}_i\mathbf{h}_i = (\mathbf{w}_i\lambda)(\mathbf{h}_i/\lambda)$ we are free to arbitrarily fix any norm of either one. In our algorithm, we thus choose to fix the L_2 norm of $\mathbf{h}_i$ to unity, as a matter of convenience.

3.3 Algorithm

We have devised a projected gradient descent algorithm for NMF with sparseness constraints. This algorithm essentially takes a step in the direction of the negative gradient, and subsequently projects onto the constraint space, making sure that the taken step is small enough that the objective function (1) is reduced at every step. The main muscle of the algorithm is the projection operator which enforces the desired degree of sparseness. This operator is described in detail following this algorithm.

Algorithm: NMF with sparseness constraints

1. Initialize $\mathbf{W}$ and $\mathbf{H}$ to random positive matrices
2. If sparseness constraints on $\mathbf{W}$ apply, then project each column of $\mathbf{W}$ to be non-negative, have unchanged L_2 norm, but L_1 norm set to achieve desired sparseness
3. If sparseness constraints on $\mathbf{H}$ apply, then project each row of $\mathbf{H}$ to be non-negative, have unit L_2 norm, and L_1 norm set to achieve desired sparseness
4. Iterate
   (a) If sparseness constraints on $\mathbf{W}$ apply,
      i. Set $\mathbf{W} := \mathbf{W} - \mu_{\mathbf{W}}(\mathbf{WH} - \mathbf{V})\mathbf{H}^T$
      ii. Project each column of $\mathbf{W}$ to be non-negative, have unchanged L_2 norm, but L_1 norm set to achieve desired sparseness

   else take standard multiplicative step $\mathbf{W} := \mathbf{W} \otimes (\mathbf{VH}^T) \oslash (\mathbf{WHH}^T)$

   (b) If sparseness constraints on $\mathbf{H}$ apply,
      i. Set $\mathbf{H} := \mathbf{H} - \mu_{\mathbf{H}}\mathbf{W}^T(\mathbf{WH} - \mathbf{V})$
      ii. Project each row of $\mathbf{H}$ to be non-negative, have unit L_2 norm, and L_1 norm set to achieve desired sparseness

   else take standard multiplicative step $\mathbf{H} := \mathbf{H} \otimes (\mathbf{W}^T\mathbf{V}) \oslash (\mathbf{W}^T\mathbf{WH})$

Above, $\otimes$ and $\oslash$ denote elementwise multiplication and division, respectively. Moreover, $\mu_{\mathbf{W}}$ and $\mu_{\mathbf{H}}$ are small positive constants (stepsizes) which must be set appropriately for the algorithm to work. Fortunately, they need not be set by the user; our implementation of the algorithm automatically adapts these parameters. The multiplicative steps are directly taken from Lee and Seung (2001) and are used when constraints are not to be applied.

Many of the steps in the above algorithm require a projection operator which enforces sparseness by explicitly setting both L_1 and L_2 norms (and enforcing non-negativity). This operator is defined as follows

---

problem Given any vector $\mathbf{x}$, find the closest (in the euclidean sense) *non-negative* vector $\mathbf{s}$ with a given L_1 norm and a given L_2 norm.

algorithm The following algorithm solves the above problem. See below for comments.

1. Set $s_i := x_i + (L_1 - \sum x_i)/\dim(\mathbf{x}),\ \forall i$
2. Set $Z := \{\}$
3. Iterate
   (a) Set $m_i := \begin{cases} L_1/(\dim(\mathbf{x}) - \text{size}(Z)) & \text{if } i \notin Z \\ 0 & \text{if } i \in Z \end{cases}$
   (b) Set $\mathbf{s} := \mathbf{m} + \alpha(\mathbf{s} - \mathbf{m})$, where $\alpha \geq 0$ is selected such that the resulting $\mathbf{s}$ satisfies the L_2 norm constraint. This requires solving a quadratic equation.
   (c) If all components of $\mathbf{s}$ are non-negative, return $\mathbf{s}$, end
   (d) Set $Z := Z \cup \{i; s_i < 0\}$
   (e) Set $s_i := 0,\ \forall i \in Z$
   (f) Calculate $c := (\sum s_i - L_1)/(\dim(\mathbf{x}) - \text{size}(Z))$
   (g) Set $s_i := s_i - c,\ \forall i \notin Z$
   (h) Go to (a)

In words, the above algorithm works as follows: We start by projecting the given vector onto the hyperplane $\sum s_i = L_1$. Next, within this space, we project to the closest point on the joint constraint hypersphere (intersection of the sum and the L_2 constraints). This is done by moving radially outward from the center of the sphere (the center is given by the point where all components have equal values). If the result is completely non-negative, we have arrived at our destination. If not, those components that attained negative values must be fixed at zero, and a new point found in a similar fashion under those additional constraints.

Note that, once we have a solution to the above *non-negative* problem, it would be straightforward to extend it to a general solution without non-negativity constraints. If a given component of $\mathbf{x}$ is positive (negative), we know because of the symmetries of L_1 and L_2 norms that the optimal solution $\mathbf{s}$ will have the corresponding component positive or zero (negative or zero). Thus, we may simply record the signs of $\mathbf{x}$, take the absolute value, perform the projection in the first quadrant using the algorithm above, and re-enter the signs into the solution.

In principle, the devised projection algorithm may take as many as $\dim(\mathbf{x})$ iterations to converge to the correct solution (because at each iteration the algorithm either converges, or at least one component is added to the set of zero valued components). In practice, however, the algorithm converges much faster. In Section 4 we show that even for extremely high dimensions the algorithm typically converges in only a few iterations.

3.4 Matlab Implementation

Our software package, available at `http://www.cs.helsinki.fi/patrik.hoyer/` implements all the details of the above algorithm. In particular, we monitor the objective function E throughout the optimization, and adapt the stepsizes to ensure convergence. The software package contains, in

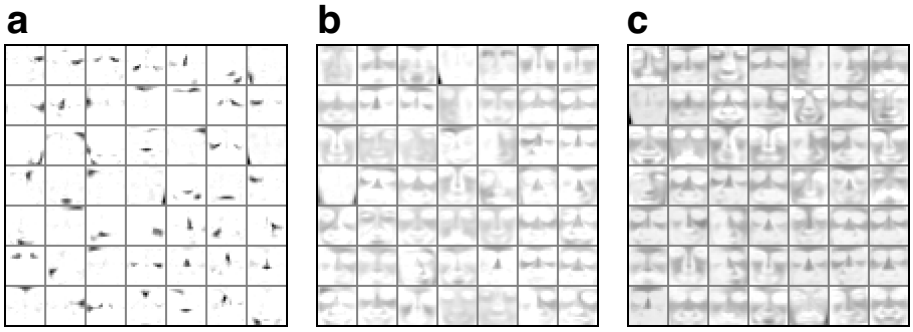

Figure 3: Features learned from the CBCL face image database using NMF with sparseness constraints. **(a)** The sparseness of the basis images were fixed to 0.8, slightly higher than the average sparseness produced by standard NMF, yielding a similar result. The sparseness of the coefficients was unconstrained. **(b)** Here, we switched the sparseness constraints such that the coefficients were constrained to 0.8 but the basis images were unconstrained. Note that this creates a global representation similar to that given by vector quantization (Lee and Seung, 1999). **(c)** Illustration of another way to obtain a global representation: setting the sparseness of the basis images to a low value (here: 0.2) also yields a non-local representation.

addition to the projection operator and NMF code, all the files needed to reproduce the results described in this paper, with the exception of data sets. For copyright reasons the face image databases are not included, but they can easily be downloaded separately from their respective www addresses.

4. Experiments with Sparseness Constraints

In this section, we show that adding sparseness constraints to NMF can make it find parts-based representations in cases where unconstrained NMF does not. In addition, we experimentally verify our claim that the projection operator described in Section 3.3 converges in only a few iterations even when the dimensionality of the vector is high.

4.1 Representations Learned from Face Image Databases

Recall from Section 2 the mixed results of applying standard NMF to face image data. Lee and Seung (1999) originally showed that NMF found a parts-based representation when trained on data from the CBCL database. However, when applied to the ORL data set, in which images are not as well aligned, a global decomposition emerges. These results were shown in Figure 1a and 1b. To compare, we applied sparseness constrained NMF to both face image data sets.

For the CBCL data, some resulting bases are shown in Figure 3. Setting a high sparseness value for the basis images results in a local representation similar to that found by standard NMF. However, we want to emphasize the fact that sparseness constrained NMF does not always lead to local solutions: Global solutions can be obtained by deliberately setting a low sparseness on the basis images, or by requiring a high sparseness on the coefficients (forcing each coefficient to try to represent more of the image).

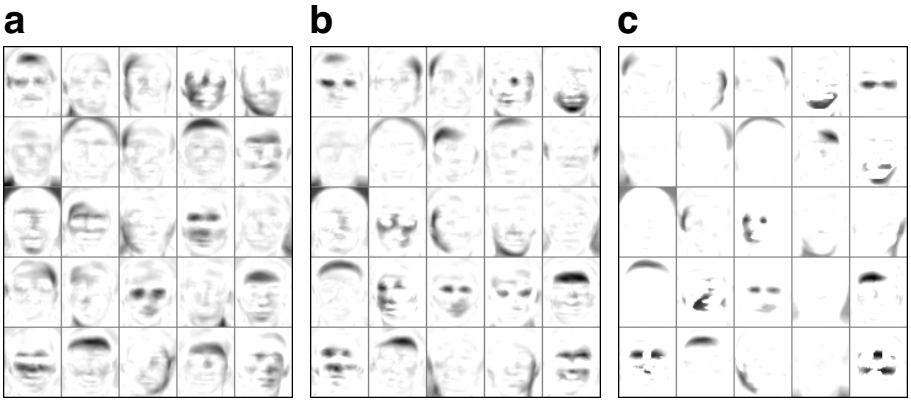

Figure 4: Features learned from the ORL face image database using NMF with sparseness constraints. When increasing the sparseness of the basis images, the representation switches from a global one (like the one given by standard NMF, cf Figure 1b) to a local one. Sparseness levels were set to **(a)** 0.5 **(b)** 0.6 **(c)** 0.75.

The ORL database provides the more interesting test of the method. In Figure 4 we show bases learned by sparseness constrained NMF, for various sparseness settings. Note that our method can learn a parts-based representation of this data set, in contrast to standard NMF. Also note that the representation is not very sensitive to the specific sparseness level chosen.

4.2 Basis Derived from Natural Image Patches

In Figure 1c we showed that standard NMF applied to natural image data produces only circular features, not oriented features like those employed by modern image processing techniques. Here, we tested the result of using additional sparseness constraints. Figure 5 shows the basis vectors obtained by putting a sparseness constraint on the coefficients ($S_h = 0.85$) but leaving the sparseness of the basis vectors unconstrained. In this case, NMF learns oriented features that represent edges and lines. Such oriented features are widely regarded as the best type of low-level features for representing natural images, and similar features are also used by the early visual system of the biological brain (Field, 1987; Simoncelli et al., 1992; Olshausen and Field, 1996; Bell and Sejnowski, 1997). This example illustrates that sparseness constrained NMF does not simply 'sparsify' the result of standard, unconstrained NMF, but rather can find qualitatively different parts-based representations that are more compatible with the sparseness assumptions.

4.3 Convergence of Algorithm Implementing the Projection Step

To verify the performance of our projection method we performed extensive tests, varying the number of dimensions, the desired degree of sparseness, and the sparseness of the original vector. The desired and the initial degrees of sparseness were set to 0.1, 0.3, 0.5, 0.7, and 0.9, and the dimensionality of the problem was set to 2, 3, 5, 10, 50, 100, 500, 1000, 3000, 5000, and 10000. All combinations of sparsenesses and dimensionalities were analyzed. Based on this analysis, the worst case (most iterations on average required) was when the desired degree of sparseness was high (0.9)

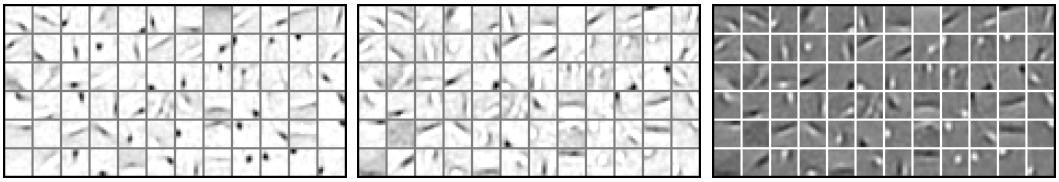

Figure 5: Basis vectors from ON/OFF-filtered natural images obtained using NMF with sparseness constraints. The sparseness of the coefficients was fixed at 0.85, and the sparseness of the basis images was unconstrained. As opposed to standard NMF (cf Figure 1c), the representation is based on oriented, Gabor-like, features.

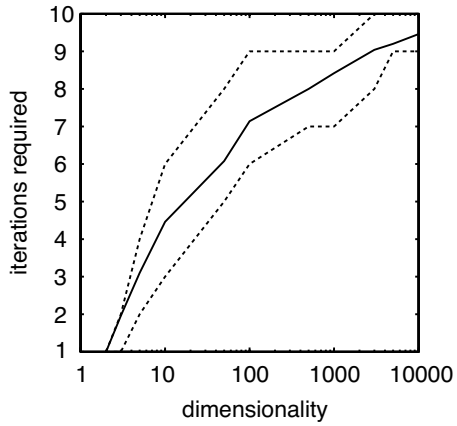

Figure 6: Number of iterations required for the projection algorithm to converge, in the worst-case scenario tested (desired sparseness 0.9, initial sparseness 0.1). The solid line shows the average number (over identical random trials) of iterations required, the dashed lines show the minimum and maximum iterations. Note that the number of iterations grows very slowly with the dimensionality of the problem.

but the initial sparseness was low (0.1). In Figure 6 we plots the number of iterations required for this worst case, as a function of dimensionality. Even in this worst-case scenario, and even for the highest tested dimensionality, the algorithm never required more than 10 iterations to converge. Thus, although we do not have analytical bounds on the performance on the algorithm, empirically the projection method performs extremely well.

5. Relation to Other Recent Work

Here, we describe how our method relates to other recently developed extensions of NMF and to non-negative independent component analysis.

5.1 Extensions of NMF

Several authors have noted the shortcomings of standard NMF, and suggested extensions and modifications of the original model. Li et al. (2001) noted that NMF found only global features from the ORL database (see Figure 1b) and suggested an extension they call *Local* Non-negative Matrix Factorization (LNMF). Their method indeed produces local features from the ORL database, similar to those given by our method (Figure 4c). However, it does not produce oriented filters from natural image data (results not shown). Further, there is no way to explicitly control the sparseness of the representation, should this be needed.

Hoyer (2002) extended the NMF framework to include an adjustable sparseness parameter. The present paper is an extension of those ideas. The main improvement is that in the present model sparseness is adjusted explicitly, rather than implicitly. This means that one does not any more need to employ trial-and-error to find the parameter setting that yields the desired level of sparseness.

Finally, Liu et al. (2003) also noted the need for incorporating the notion of sparseness, and suggested an extension termed *Sparse* Non-negative Matrix Factorization (SNMF). Their extension is similar in spirit and form to that given by Hoyer (2002) with the added benefit of yielding a more convenient, faster algorithm. Nevertheless, it also suffers from the drawback that sparseness is only controlled implicitly. Furthermore, their method does not yield oriented features from natural image data (results not shown).

In summary, the framework presented in the present paper improves on these previous extensions by allowing explicit control of the statistical properties of the representation.

In order to facilitate the use of, and comparison between, the various extensions of NMF, they are all provided as part of the Matlab code package distributed with this paper. Using this package readers can effortlessly verify our current claims by applying the algorithms to the various data sets. Moreover, the methods can be compared head-to-head on new interesting data sets.

5.2 Non-negative Independent Component Analysis

Our method has a close connection to the statistical technique called independent component analysis (ICA) (Hyvärinen et al., 2001). ICA attempts to find a matrix factorization similar to ours, but with two important differences. First, the signs of the components are in general not restricted; in fact, symmetry is often assumed, implying an approximately equal number of positive and negative elements. Second, the sources are not forced to any desired degree of sparseness (as in our method) but rather sparseness is incorporated into the objective function to be optimized. The sparseness goal can be put on either $\mathbf{W}$ or $\mathbf{H}$, or both (Stone et al., 2002).

Recently, some authors have considered estimating the ICA model in the case of one-sided, non-negative sources (Plumbley, 2003; Oja and Plumbley, 2004). In these methods, non-negativity is not specified as a constraint but rather as an objective; hence, complete non-negativity of the representation is seldom achieved for real-life data sets. Nevertheless, one can show that if the linear ICA model holds, with non-negative components, these methods can identify the model.

6. Conclusions

Non-negative matrix factorization (NMF) has proven itself a useful tool in the analysis of a diverse range of data. One of its most useful properties is that the resulting decompositions are often intuitive and easy to interpret because they are sparse. Sometimes, however, the sparseness achieved

by NMF is not enough; in such situations it might be useful to control the degree of sparseness explicitly. Our main contributions of this paper were (a) to describe a projection operator capable of simultaneously enforcing both L_1 and L_2 norms and hence any desired degree of sparseness, (b) to show its use in the NMF framework for learning representations that could not be obtained by regular NMF, and (c) to provide a software package to enable researchers and practitioners to easily perform NMF and its various extensions. We hope that all three contributions will prove useful to the field of data-analysis.

Acknowledgments

The author wishes to thank Jarmo Hurri, Aapo Hyvärinen, and Fabian Theis for useful discussions and comments on the manuscript.

References

A. J. Bell and T. J. Sejnowski. The 'independent components' of natural scenes are edge filters. *Vision Research*, 37:3327–3338, 1997.

J. P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov. Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the National Academy of Sciences*, 101:4164–4169, 2004.

G. Buchsbaum and O. Bloch. Color categories revealed by non-negative matrix factorization of munsell color spectra. *Vision Research*, 42:559–563, 2002.

D. Donoho and V. Stodden. When does non-negative matrix factorization give a correct decomposition into parts? In *Advances in Neural Information Processing 16 (Proc. NIPS*2003)*. MIT Press, 2004.

D. J. Field. Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America*, 4:2379–2394, 1987.

D. J. Field. What is the goal of sensory coding? *Neural Computation*, 6:559–601, 1994.

P. Földiák and M. P. Young. Sparse coding in the primate cortex. In Michael A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 895–898. The MIT Press, Cambridge, Massachusetts, 1995.

P. O. Hoyer. Non-negative sparse coding. In *Neural Networks for Signal Processing XII (Proc. IEEE Workshop on Neural Networks for Signal Processing)*, pages 557–565, Martigny, Switzerland, 2002.

P. O. Hoyer. Modeling receptive fields with non-negative sparse coding. In E. De Schutter, editor, *Computational Neuroscience: Trends in Research 2003*. Elsevier, Amsterdam, 2003. Also published in: Neurocomputing 52-54 (2003), pp 547-552.

A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley Interscience, 2001.

K. Jung and E. Y. Kim. Automatic text extraction for content-based image indexing. *Advances in Knowledge Discovery and Data Mining: Proceedings Lecture Notes in Artificial Intelligence*, 3056:497–507, 2004.

P. M. Kim and B. Tidor. Subsystem identification through dimensionality reduction of large-scale gene expression data. *Genome Research*, 13:1706–1718, 2003.

D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing 13 (Proc. NIPS*2000)*. MIT Press, 2001.

S. Z. Li, X. Hou, H. Zhang, and Q. Cheng. Learning spatially localized parts-based representations. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Vol. I*, pages 207–212, Hawaii, USA, 2001.

W. Liu, N. Zheng, and X. Lu. Non-negative matrix factorization for visual coding. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP'2003)*, 2003.

E. Oja and M. Plumbley. Blind separation of positive sources by globally convergent gradient search. *Neural Computation*, 16(9):1811–1825, 2004.

B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5:111–126, 1994.

L. Parra, C. Spence, P. Sajda, A. Ziehe, and K.-R. Müller. Unmixing hyperspectral data. In *Advances in Neural Information Processing 12 (Proc. NIPS*99)*, pages 942–948. MIT Press, 2000.

M. Plumbley. Algorithms for non-negative independent component analysis. *IEEE Transactions on Neural Networks*, 14(3):534–543, 2003.

E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger. Shiftable multiscale transforms. *IEEE Transactions on Information Theory*, 38:587–607, 1992.

J. V. Stone, J. Porrill, N. R. Porter, and I. D. Wilkinson. Spatiotemporal independent component analysis of event-related fMRI data using skewed probability density functions. *Neuroimage*, 15: 407–421, 2002.

S. Thorpe. Localized versus distributed representations. In Michael A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 549–552. The MIT Press, Cambridge, Massachusetts, 1995.