SIAM J. Math. Data Sci.
Vol. 4, No. 2, pp. 431–463

A Nonlinear Matrix Decomposition for Mining the Zeros of Sparse Data[*]

Lawrence K. Saul[†]

Abstract. We describe a simple iterative solution to a widely recurring problem in multivariate data analysis: given a sparse nonnegative matrix $\mathbf{X}$, how to estimate a low-rank matrix $\mathbf{\Theta}$ such that $\mathbf{X} \approx f(\mathbf{\Theta})$, where f is an elementwise nonlinearity? We develop a latent variable model for this problem and consider those sparsifying nonlinearities, popular in neural networks, that map all negative values to zero. The model seeks to explain the variability of sparse high-dimensional data in terms of a smaller number of degrees of freedom. We show that exact inference in this model is tractable and derive an expectation-maximization (EM) algorithm to estimate the low-rank matrix $\mathbf{\Theta}$. Notably, we do not parameterize $\mathbf{\Theta}$ as a product of smaller matrices to be alternately optimized; instead, we estimate $\mathbf{\Theta}$ directly via the singular value decomposition of matrices that are repeatedly inferred (at each iteration of the EM algorithm) from the model's posterior distribution. We use the model to analyze large sparse matrices that arise from data sets of binary, grayscale, and color images. In all of these cases, we find that the model discovers much lower-rank decompositions than purely linear approaches.

Key words. matrix factorization, latent variable modeling, unsupervised learning

AMS subject classifications. 15A23, 15B48, 62R07, 65F50, 65F55, 69T09

DOI. 10.1137/21M1405769

1. Introduction.

Many empirical disciplines depend increasingly on principled and transparent methods for high-dimensional data analysis [29, 123]. The simplest methods arise from basic tools of linear algebra, for example, when data is stored in a large matrix $\mathbf{X}$, we can use singular value decomposition (SVD) to approximate $\mathbf{X}$ by another matrix $\mathbf{\Theta}$ of lower rank [30]. Very often such decompositions can help to analyze high-dimensional data in terms of a much smaller number of degrees of freedom [26, 118].

Many types of data, however, have low-dimensional structure that is not revealed by such methods. This is especially true for nonnegative or binary data that is represented by a large sparse matrix (i.e., a matrix whose elements are mostly zero). Such data is better analyzed by models that introduce some additional constraint or nonlinearity to prohibit values that the data cannot realize [24, 41, 72, 103, 119]. Typically, these models parameterize their low-rank matrices as the product of smaller matrices—for instance, writing $\mathbf{\Theta} = \mathbf{WH}$, where $\mathbf{W}$ is tall and thin and $\mathbf{H}$ is short and wide—and then optimize the factors $\mathbf{W}$ and $\mathbf{H}$ in an alternating fashion.

These types of alternating approaches are common for problems in high-dimensional data analysis that do not have closed-form solutions. In such problems, it is natural to identify the largest subsets of model parameters that can be efficiently optimized while holding the

[*] Received by the editors March 18, 2021; accepted for publication (in revised form) November 3, 2021; published electronically April 7, 2022.
https://doi.org/10.1137/21M1405769

[†] Department of Computer Science and Engineering, University of California, San Diego, La Jolla, CA USA (saul@cs.ucsd.edu, http://www.cs.ucsd.edu/~saul).

others fixed. The conceptual goal here is simple—to encapsulate the model updates as well-understood optimizations, and then to rely on the most powerful black-box solvers for those optimizations that we have at our disposal. In a nutshell, that is why many generalized low-rank models perform alternating least-squares or convex optimizations over factors $\mathbf{W}$ and $\mathbf{H}$ of a low-rank matrix $\mathbf{\Theta}$.

One might ask, though, whether this parameterization is always necessary for generalized low-rank models—whether, instead, it might be possible to optimize the low-rank matrix $\mathbf{\Theta}$ directly, in a way that provides an even higher-level encapsulation of the model updates? There are few tractable optimizations over nonconvex sets of low-rank matrices, but as mentioned above, one notable exception is to compute (via SVD) an optimal low-rank approximation to some other matrix. This suggests using truncated SVDs as the basis of an iterative strategy for generalized low-rank modeling. Of course, such a strategy presumes that there exist generalized low-rank models of interest that can be estimated in this way. The motivation to find such models is compelling: if they do exist, then their fitting procedures can piggyback for the rest of time on better and faster algorithms for SVD. Note that this strategy takes a completely agnostic view of how the truncated SVD is computed. There are many choices for this purpose (e.g., bidiagonalization and column pivoting [22, 38], alternating least-squares [46], other nonconvex optimizations [19, 57], randomized methods [34, 44, 80, 116, 117]), but in this framework the choice becomes a lower-level operational decision that can be based on the specific problem instance. The goal of higher-level encapsulation is achieved no matter which routine is ultimately chosen.

In this paper we develop a nonlinear matrix decomposition (NMD) for sparse nonnegative data that achieves this goal. By this we mean that the decomposition is *not* computed by an alternating optimization over the factors $\mathbf{W}$ and $\mathbf{H}$ of a low-rank matrix $\mathbf{\Theta} = \mathbf{WH}$. Instead we describe an iterative algorithm that leverages the full power of SVD to optimize $\mathbf{\Theta}$ directly; in particular, at each iteration, the algorithm uses SVD to reestimate $\mathbf{\Theta}$ as the optimal low-rank approximation to another matrix of inferred values. In addition, as in many previous studies, the decomposition exploits a single but essential nonlinearity [1, 6, 5, 25, 36, 43, 87, 95, 97, 130, 131]: given a sparse nonnegative matrix $\mathbf{X}$, it attempts to estimate a low-rank matrix $\mathbf{\Theta}$ such that $\mathbf{X} \approx f(\mathbf{\Theta})$, where f is an elementwise nonlinearity that does not take on negative values. As we shall see, the form of this nonlinearity can be purposefully tailored to reveal low-dimensional structure in sparse data.

Another contribution of our work lies in its appeal to latent variable modeling [3, 9]. There are well-known latent variable models for many canonical problems in high-dimensional data analysis (e.g., Gaussian mixture distributions for clustering [85], factor analysis for dimensionality reduction [98]). These models have been widely applied, in large part because there are simple, provably convergent algorithms for estimating their parameters. In this paper, we derive an equally tractable latent variable model for NMD—one in which inference does not require sampling-based or variational approximations, and one in which parameter estimation does not require line searches, learning rates, or projected gradients. This model should be of broad interest because NMD can also be viewed as a framework for unsupervised learning in two-layer neural networks—specifically, those networks [74, 83] with one hidden layer of binary threshold or rectified linear units (ReLU). This is perhaps the paper's main conceptual contribution: it elevates yet another core problem of unsupervised learning into the canon that can be studied by especially tractable latent variable models.

The paper is organized as follows. In section 2, we present several motivations for this work, and in section 3, we contrast our goals to those of previous approaches. In section 4, we show how to formulate NMD as a problem in latent variable modeling; here, in particular, we show how repeated SVDs can be used to estimate a low-rank matrix $\mathbf{\Theta}$ such that $\mathbf{X} \approx f(\mathbf{\Theta})$. In section 5, we present experimental results on several illustrative data sets and examine the low-dimensional representations discovered by NMD. Finally, in section 6, we offer our main conclusions and discuss important directions for future work.

2. Motivation.

Sparse nonnegative matrices arise in many areas of application. The elements of these matrices can record, for example, the edges of objects in grayscale images [15], the gene expression levels in cells [11], the presence of links in a social network [51], the word counts in a large corpus of documents [26], and the ratings or purchases of users on the internet [66]. In addition, it is widely believed that natural images and sounds are encoded by the brain as sparse distributed patterns of neural activity [28, 33, 49, 91], so that any collection of these patterns (e.g., over time) can also be visualized as a sparse nonnegative matrix.

From many disparate applications, then, there arises the same question: how to discover low-dimensional structure in sparse high-dimensional data? The possibility of such structure is precisely what motivates the search for low-rank models [24, 41, 72, 88, 103, 107, 119]. The rank of a matrix is equal to the number of its columns (or rows) that are linearly independent. If the columns of such a matrix store the patterns of a data set, then the rank provides one way to quantify the data's underlying degrees of freedom. In practice, we can settle for a low-rank matrix that does not exactly match the data, but still provides a close approximation. Moreover, for certain types of data, we can hypothesize a nonlinear relationship between the data and its underlying degrees of freedom. The next sections motivate our low-rank model with these ideas in mind.

2.1. Strengths and limitations of SVD.

As is well known, the rank of a real-valued $d \times n$ matrix $\mathbf{X}$ can be immediately ascertained from its SVD. To do so, we write $\mathbf{X} = \mathbf{U\Sigma V}^\top$, where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices (of size $d \times d$ and $n \times n$, respectively) and $\mathbf{\Sigma}$ is a rectangular diagonal matrix that stores the singular values $\Sigma_{ii} \geq 0$. The rank of $\mathbf{X}$ is given by the number of its nonzero singular values.

The SVD of a matrix can also be used to find its optimal low-rank approximation, where the error of the approximation is measured by the Frobenius norm [30]. In particular, the SVD is used to solve the following optimization,

$$\min \|\mathbf{X} - \mathbf{\Theta}\|_F \quad \text{such that} \quad \text{rank}(\mathbf{\Theta}) = r, \tag{2.1}$$

where r is less than the rank of $\mathbf{X}$. In this case, the solution is given by the *truncated* SVD, with $\mathbf{\Theta} = \mathbf{U\Sigma}_r\mathbf{V}^\top$, where $\mathbf{\Sigma}_r$ has the same dimensions as $\mathbf{\Sigma}$ but retains only its top r singular values and replaces the others by zero.

The SVD is widely used in this way[1] for high-dimensional data analysis. Suppose, for instance, that the columns of $\mathbf{X}$ store the patterns of a data set, and also that $\mathbf{X}$ is well

[1] This usage of SVD is equivalent to principal components analysis if the data has been centered to have zero mean. For sparse data, however, it is not usual to perform this centering.

approximated by a matrix of lower rank. Then we can regard the rank r of the approximation as a measure of the data's most consequential degrees of freedom, and we can regard the error in the approximation as a measure of imprecision or noise.

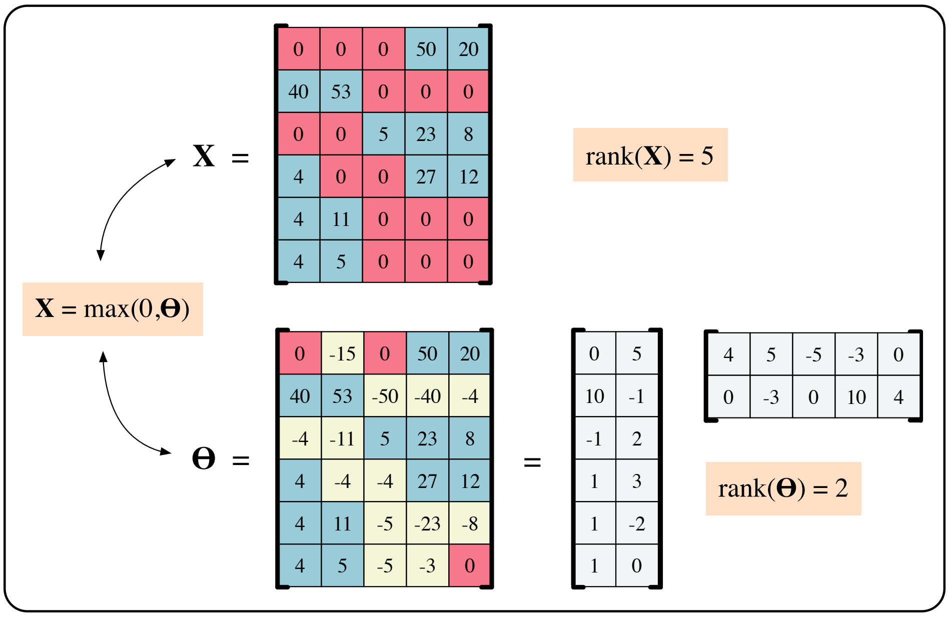
It is worth pausing to appreciate the closed-form solution to (2.1) provided by the truncated SVD. The domain of the optimization is the set of $d \times n$ matrices with rank r, where $r < \min(d, n)$. Note that these matrices do not form a convex set, and as a result, this overall optimization is not convex. Nevertheless, the truncated SVD yields the globally optimal solution. This is the strength of the SVD, and naturally, one might hope to leverage this strength in the search for low-rank models that are specialized, in some way, to sparse nonnegative matrices. We develop this idea further in subsection 2.5, and then again more fully in section 4.

2.2. Mining the zeros of sparse data. The goal of NMD is easily stated: given a sparse nonnegative matrix $\mathbf{X}$, it attempts to estimate a low-rank matrix $\mathbf{\Theta}$ such that $\mathbf{X} \approx f(\mathbf{\Theta})$, where f is an elementwise nonlinearity. In general, for nonnegative matrices, we will consider approximations of the form $\mathbf{X} \approx \max(0, \mathbf{\Theta})$; this is the same hinge nonlinearity that appears in neural networks with ReLUs. However, for binary matrices in particular—-where the elements are equal to either zero or one—we will consider approximations of the form $\mathbf{X} \approx \frac{1}{2}[1+\text{sign}(\mathbf{\Theta})]$ with a threshold nonlinearity. Thus, in both cases, the nonlinearity serves to restrict the approximated values of $\mathbf{X}$ to their underlying domain—an idea that has been widely explored for models of binary, ordinal, and mixed data [1, 43, 51, 52, 87, 97, 96, 131]. This is not, however, the only purpose of the nonlinearity in our model. For nonnegative matrices, as we shall see, it is especially important that the nonlinearity maps all negative values of $\mathbf{\Theta}$ into zeros of $\mathbf{X}$. As a result, when $\mathbf{X}$ is sparse—with mostly zero elements—NMD has much more flexibility than SVD to discover low-rank decompositions.

This idea is illustrated in Figure 2.1. Here, $\mathbf{X}$ is a sparse nonnegative matrix of full rank. However, it is possible, by replacing the zeros of $\mathbf{X}$ with strategically chosen negative values, to find a matrix $\mathbf{\Theta}$ of lower rank such that $\mathbf{X} = \max(0, \mathbf{\Theta})$. More generally, the goal is to find a lower-rank matrix $\mathbf{\Theta}$ such that the reconstruction $\mathbf{X} \approx \max(0, \mathbf{\Theta})$ is accurate to an acceptable degree of approximation.

As an aside, we note that a similar idea has been very extensively explored for the problem of matrix completion [14, 46]. In this problem, one considers a matrix (not necessarily sparse or nonnegative) whose elements are only partially specified. Then, to complete the matrix, one fills in its missing elements by assuming that the matrix has the lowest possible rank. Thus the larger the number of missing elements, the more flexibility one has to complete the matrix with a low-rank model.

In our problem, this flexibility is derived from the *zeros* of the sparse nonnegative matrix $\mathbf{X}$. In particular, as shown in Figure 2.1, the sparser the matrix $\mathbf{X}$, the more zeros we can mine to lower the rank of $\mathbf{\Theta}$. As we shall see, NMD fully exploits this flexibility to replace the zeros of $\mathbf{X}$ by strategically chosen negative values. Finally, we note that an even greater flexibility is obtained when the matrix $\mathbf{X}$ is binary. In this case, NMD searches for approximations of the form $\mathbf{X} \approx \frac{1}{2}[1 + \text{sign}(\mathbf{\Theta})]$; thus, in constructing $\mathbf{\Theta}$, not only can it replace the zeros of $\mathbf{X}$ by arbitrary negative values, but it can also replace the ones of $\mathbf{X}$ by arbitrary positive values.



$$\mathbf{X} = \begin{bmatrix} 0 & 0 & 0 & 50 & 20 \\ 40 & 53 & 0 & 0 & 0 \\ 0 & 0 & 5 & 23 & 8 \\ 4 & 0 & 0 & 27 & 12 \\ 4 & 11 & 0 & 0 & 0 \\ 4 & 5 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{\Theta} = \begin{bmatrix} 0 & -15 & 0 & 50 & 20 \\ 40 & 53 & -50 & -40 & -4 \\ -4 & -11 & 5 & 23 & 8 \\ 4 & -4 & -4 & 27 & 12 \\ 4 & 11 & -5 & -23 & -8 \\ 4 & 5 & -5 & -3 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 5 \\ 10 & -1 \\ -1 & 2 \\ 1 & 3 \\ 1 & -2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 4 & 5 & -5 & -3 & 0 \\ 0 & -3 & 0 & 10 & 4 \end{bmatrix}$$

Figure 2.1. *Top: $\mathbf{X}$ is a sparse nonnegative matrix containing mainly zeros (shown in red). Bottom: $\mathbf{\Theta}$ is a lower-rank matrix satisfying $\mathbf{X} = \max(0, \mathbf{\Theta})$; it is obtained by replacing the zeros of $\mathbf{X}$, as needed, with strategically chosen negative values (shown in yellow). In general, the sparser the matrix $\mathbf{X}$, the more zeros we can mine to lower the rank of $\mathbf{\Theta}$.*

While the example in Figure 2.1 illustrates the basic idea behind NMD, it does not particularly convey why such a decomposition might be useful. The matrix $\mathbf{X}$ in Figure 2.1 was randomly generated, and for that reason, it is not especially deserving of further study. Next we consider a more interesting example where the low-rank model provides exactly this type of insight.

2.3. Low-rank models of pattern manifolds. We adapt an example from [74] showing how an elementwise nonlinearity can model large disparities in rank between the matrices $\mathbf{\Theta}$ and $\mathbf{X}$. To this end, let $\alpha > 0$, and consider the $n \times n$ circulant matrices with elements

$$\Theta_{ij} = 1 - \alpha \left[1 - \cos \frac{2\pi}{n}(i-j) \right], \tag{2.2}$$

$$X_{ij} = \max(0, \Theta_{ij}). \tag{2.3}$$

For these matrices, it is straightforward to show that $\text{rank}(\mathbf{\Theta}) = 3$ for all $n \geq 3$, while $\mathbf{X}$ is of full rank for a range of values for α. Note that when α is very large, the matrix $\mathbf{X}$ in (2.3) reduces to the identity matrix, which is obviously of full rank. However, we will be focused on values of α for which $\mathbf{X}$ also has an interesting interpretation as a data set of sparse images; see Figure 2.2.

Let us quickly prove the above claims. First we show that $\text{rank}(\mathbf{\Theta}) = 3$. This is most easily done by defining the orthogonal column vectors $\mathbf{c}, \mathbf{s}, \mathbf{u} \in \mathbb{R}^n$ with elements

$$c_i = \cos\left(\frac{2\pi i}{n}\right), \qquad s_i = \sin\left(\frac{2\pi i}{n}\right), \qquad u_i = 1 \tag{2.4}$$

for $1 \leq i \leq n$. These vectors correspond in fact to the only eigenvectors of $\boldsymbol{\Theta}$ with nonzero eigenvalues, as can be seen by writing $\Theta = (1-\alpha)\,\mathbf{u}\mathbf{u}^\top + \alpha(\mathbf{c}\mathbf{c}^\top + \mathbf{s}\mathbf{s}^\top)$. This decomposition establishes that $\text{rank}(\boldsymbol{\Theta}) = 3$ for $n \geq 3$. Note that this result holds for all values of $\alpha \notin \{0, 1\}$.

Next we prove that $\mathbf{X}$ has full rank. The idea of the proof is straightforward. First we show that the diagonal elements of $\mathbf{X}$ are all equal to one; then we show that the remaining (off-diagonal) elements of $\mathbf{X}$ are either zero or sufficiently small that none of its eigenvalues can deviate too far from unity. To simplify the analysis, we consider the fixed value

$$\alpha = \frac{1}{2\sin\frac{\pi}{n}\sin\frac{\pi}{2n}}, \tag{2.5}$$

but it is not difficult to show that the results also hold for all larger values of α. We start by substituting (2.5) into (2.2), which after some simplification yields

$$\Theta_{ij} = 1 - \frac{\sin^2\frac{\pi}{n}(i-j)}{\sin\frac{\pi}{n}\sin\frac{2\pi}{n}}. \tag{2.6}$$

Immediately we see that $\Theta_{ii} = 1$, and hence from (2.3) we also have $X_{ii} = 1$ for all terms on the diagonal. In addition, among the off-diagonal terms, we see that $\Theta_{ij} < 0$ (and hence $X_{ij} = 0$) unless $i - j \equiv 1 \bmod n$. Next let us evaluate (2.6) for terms just above or below the diagonal. Then we see (for $n \geq 3$) that each row of $\mathbf{X}$ has exactly two positive off-diagonal terms bounded by

$$X_{i,i\pm1} = 1 - \frac{\sin\frac{\pi}{n}}{\sin\frac{2\pi}{n}} = 1 - \left(2\cos\frac{\pi}{n}\right)^{-1} < \frac{1}{2}. \tag{2.7}$$

From this last result, we have shown not only that $X_{ii} = 1$ everywhere on the diagonal, but also that $\sum_{j\neq i}|X_{ij}| < 1$. It follows from the Gershgorin circle theorem that $\mathbf{X}$ has no zero eigenvalues and hence must be of full rank. The particular choice of α in (2.5) was convenient for this proof, but note that any larger choice pushes $\mathbf{X}$ even closer to the identity matrix (and its eigenvalues closer to unity).

The example in (2.2) and (2.3) may seem contrived, but as shown in [74], it arises naturally from a data set of sparse one-dimensional images (i.e., each image consisting of a single row of n pixels) that are related by simple translations. In particular, consider a data set of n such images in which each image contains a symmetric blip—three pixels wide, and darkest in its center pixel—against a light background. Here we also allow the blip to wrap around the edge of the image if its center lies on a boundary; see Figure 2.2. If we order the images by the center of these blips, then we see that each successive image is formed by shifting the pixels of the previous one; moreover, the data set as a whole is invariant under this operation. This invariance is, in turn, reflected in the eigenvectors of $\boldsymbol{\Theta}$, which correspond to simple Fourier modes. In sum, the data set for $\mathbf{X}$ in (2.3) can be viewed as arising from a pattern manifold of one-dimensional translations, and the essential structure of this manifold is reflected in the much lower rank matrix $\boldsymbol{\Theta}$ of (2.2). This transformation is illustrated in Figure 2.2.

It is worth emphasizing two points about this transformation. First, it elicits a manifold-like structure that is not immediately apparent in the original space of images. In particular, note that most pairs of images in this data set have zero overlap (i.e., they are orthogonal)

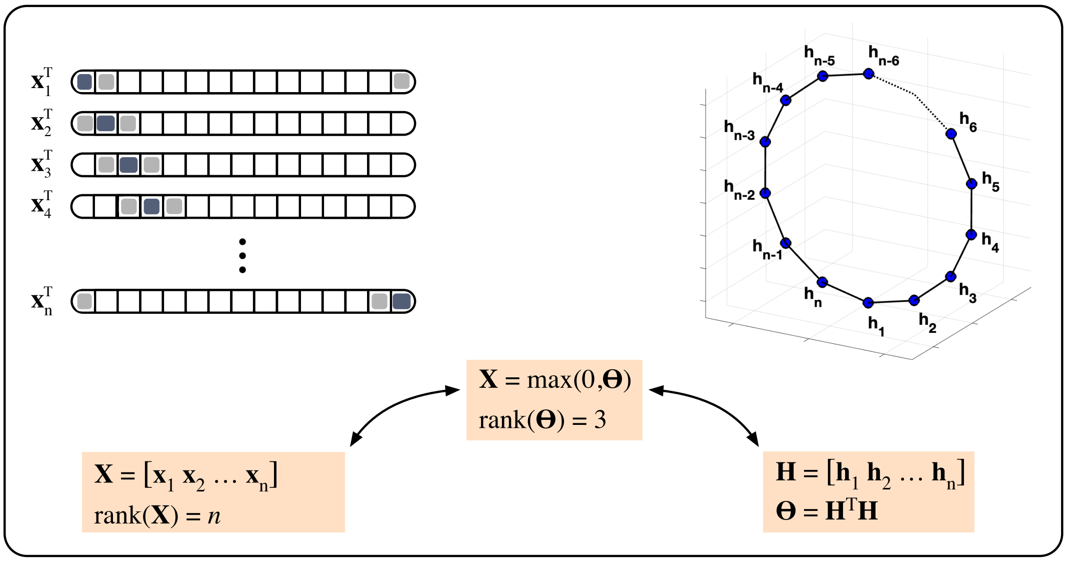

x_1^T
x_2^T
x_3^T
x_4^T
x_n^T
h_{n-6}
h_{n-5}
h_{n-4}
h_{n-3}
h_{n-2}
h_{n-1}
h_n
h_1
h_2
h_3
h_4
h_5
h_6
$\mathbf{X} = \max(0,\boldsymbol{\Theta})$
$\mathrm{rank}(\boldsymbol{\Theta}) = 3$
$\mathbf{X} = [\mathbf{x}_1\ \mathbf{x}_2 \ldots \mathbf{x}_n]$
$\mathrm{rank}(\mathbf{X}) = n$
$\mathbf{H} = [\mathbf{h}_1\ \mathbf{h}_2 \ldots \mathbf{h}_n]$
$\boldsymbol{\Theta} = \mathbf{H}^T\mathbf{H}$

Figure 2.2. *Illustration of the matrices* $\boldsymbol{\Theta}$ *and* $\mathbf{X}$ *in* (2.2) *and* (2.3)*. The matrix* $\mathbf{X}$ *stores a data set of sparse, one-dimensional images in which a darkened blip (three pixels wide) is translated across a light background. The matrix* $\boldsymbol{\Theta}$*, of much lower rank, satisfies* $\mathbf{X} = \max(0, \boldsymbol{\Theta})$*. See text for further details.*

when viewed as vectors of pixel values. Second, the transformation does not precisely reveal the number of degrees of freedom in the data, but it does provide, through the rank of $\boldsymbol{\Theta}$, an upper bound on this number. To be precise, the rank specifies a dimensionality in which the data's underlying manifold can be embedded. In Figure 2.2, for instance, the data have one essential degree of freedom, and the underlying one-dimensional manifold—the circle—is embedded in the three-dimensional space spanned by the columns of $\boldsymbol{\Theta}$.

It is widely believed that many data sets—and perhaps even the neural responses that underlie brain activity—can be understood or analyzed in terms of these pattern manifolds [102]. The degrees of freedom in these manifolds may arise from spatial transformations (e.g., translation, rotation) in the physical world; they may also reflect the continuous variabilities that are inherent to classes of diverse objects (e.g., the different shapes of faces). In section 5, we consider several sparse data sets, of increasing complexity, that are motivated by the example of this section.

2.4. Interpretation as a neural network. Low-rank models with elementwise nonlinearities have also been studied as a paradigm for unsupervised learning in neural networks [49, 50, 74, 83]. Consider the network shown in Figure 2.3, with a bottom layer of r hidden units, a top layer of d visible units, and an $r \times d$ weight matrix $\mathbf{W}$ connecting the units in these layers. We assume that $r < d$, so that the network is attempting to explain a larger pattern of activities in the visible layer by a smaller pattern of activities in the hidden one. Such a network can be trained, in an unsupervised fashion, from a data set $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ of n visible patterns, each of dimensionality d. In this case, the goal of learning is to estimate a weight matrix $\mathbf{W}$ *and* infer a corresponding set $(\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n)$ of hidden patterns, each of dimensionality r, such that

$$\mathbf{x}_i \approx f(\mathbf{W}\mathbf{h}_i), \tag{2.8}$$

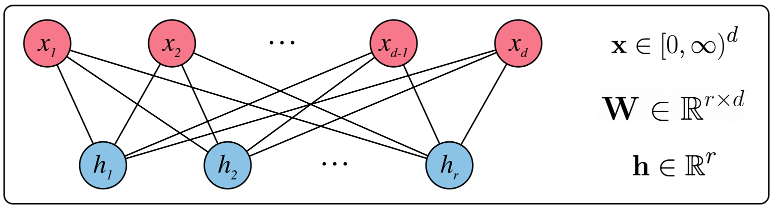

x_1
x_2
$\cdots$
x_{d-1}
x_d
h_1
h_2
$\cdots$
h_r
$\mathbf{x} \in [0,\infty)^d$
$\mathbf{W} \in \mathbb{R}^{r\times d}$
$\mathbf{h} \in \mathbb{R}^r$

Figure 2.3. *A two-layer neural network for unsupervised learning. In the top layer, visible units (shown in red) encode nonnegative patterns of activity. In the bottom layer, hidden units (shown in blue) encode a lower-dimensional representation of these patterns. The weights and hidden units are allowed to take on arbitrary real values, but the visible units are constrained to nonnegative values by an elementwise nonlinearity* $\mathbf{x} = f(\mathbf{Wh})$.

where f is an elementwise nonlinearity of the sort we have described above. Equation (2.8) makes clear the connection to the low-rank models that we are considering in this paper. In particular, suppose that we represent the data set by a $d \times n$ matrix $\mathbf{X}$ and that we represent the network's inferred patterns by an $r \times n$ matrix $\mathbf{H}$. Then (2.8) reduces to $\mathbf{X} \approx f(\mathbf{\Theta})$, where $\mathbf{\Theta} = \mathbf{WH}$ is a matrix of rank r.

Unsupervised neural networks are commonly trained by alternating procedures for inference and learning. Roughly speaking, the goal of inference is to reveal target activities for the hidden units, and the goal of learning is to stabilize these target activities by adapting the network's weights. For the network in Figure 2.3, this can be implemented as an alternating optimization over the factors $\mathbf{H}$ and $\mathbf{W}$ of the low-rank matrix $\mathbf{\Theta} = \mathbf{WH}$. As already mentioned, this paper explores a different approach, optimizing directly over the set of low-rank matrices from which $\mathbf{\Theta}$ must be chosen. This approach can be viewed as a strategy for *jointly* estimating the network's weights along with the hidden unit activities for all of the training patterns. We develop this approach further in the next section.

2.5. The idea in a nutshell.

Suppose that $\mathbf{X}$ is a sparse nonnegative matrix. It is possible to convey the main idea behind our approach without fully developing it as a Gaussian latent variable model. We do so here for the problem of estimating a low-rank matrix $\mathbf{\Theta}$ such that $\mathbf{X} \approx \max(0, \mathbf{\Theta})$. (The binary case is similar.) Consider the following optimization over the matrix $\mathbf{\Theta}$ and an additional auxiliary matrix $\mathbf{Z}$, which is not required to be of low rank:

$$\min_{\mathbf{Z},\mathbf{\Theta}} \|\mathbf{Z} - \mathbf{\Theta}\|_F^2 \quad \text{such that} \quad \begin{cases} \operatorname{rank}(\mathbf{\Theta}) = r, \\ \max(\mathbf{0}, \mathbf{Z}) = \mathbf{X}. \end{cases} \tag{2.9}$$

We make two simple observations: first, the bottom constraint in (2.9) enforces that the elements of $\mathbf{X}$ can be perfectly recovered from those of $\mathbf{Z}$, and second, the objective function in (2.9) is bounded below by zero and only obtains this minimum value when $\mathbf{X} = \max(0, \mathbf{\Theta})$.

This objective function can be minimized by a simple alternating optimization over the matrix variables $\mathbf{Z}$ and $\mathbf{\Theta}$. Suppose that $\mathbf{\Theta}$ is fixed. Then $\mathbf{Z}$ is optimized by choosing

$$Z_{ij} = \begin{cases} X_{ij} & \text{if} \quad X_{ij} > 0, \\ \min(0, \Theta_{ij}) & \text{if} \quad X_{ij} = 0. \end{cases} \tag{2.10}$$

Likewise, suppose that $\mathbf{Z}$ is fixed. Then $\mathbf{\Theta}$ is optimized by computing (via a truncated SVD) the best matrix approximation of rank r to $\mathbf{Z}$. These updates for $\mathbf{Z}$ and $\mathbf{\Theta}$ can be alternated to compute a (possibly local) minimum of the objective function. Intuitively, if this minimum value is close to zero, then it is likely that $\mathbf{X} \approx \max(0, \boldsymbol{\Theta})$.

We develop this approach more fully as a latent variable model in section 4. There is a useful analogy here to the k-means algorithm for clustering [79], which also involves an alternating optimization. The k-means algorithm may be viewed as a limiting case of the EM algorithm [27] for parameter estimation in Gaussian mixture models [85]. The optimization in (2.9) plays an analogous role for the latent variable model at the heart of this paper.

3. Relation to previous work.

There is a large literature on low-rank models for multivariate data analysis, and our work builds on many previous approaches. In this section we survey these approaches, focusing on those for nonnegative or binary data, while highlighting the essential similarities and differences with our own work. We also return to some of these comparisons in section 6.

3.1. Nonnegative matrix factorization.

Interest in low-rank models for nonnegative data [92] exploded after the work of Lee and Seung [72, 73] on nonnegative matrix factorization (NMF). Given a nonnegative matrix $\mathbf{X}$, the goal of NMF is to discover a low-rank factorization $\mathbf{X} \approx \mathbf{W}\mathbf{H}$ where the factors $\mathbf{W}$ and $\mathbf{H}$ are also constrained to be nonnegative. In NMF these factors are estimated to minimize the error of the approximation, as measured by either the Frobenius norm or generalized Kullback–Liebler divergence. The popularity of NMF is owed in large part to closed-form, multiplicative updates that can be derived for updating $\mathbf{W}$ and $\mathbf{H}$. These updates are not only simple to implement, but can also be shown to monotonically decrease the error of the approximation. In fact, the updates are widely used even though more traditional solvers (e.g., based on quasi-Newton methods [12]) are often much faster. It also remains an active area of research [2, 20, 31, 55, 63] to develop even faster approaches.

The popularity of NMF is owed in addition to the interpretability of its low-rank models. By constraining the factors $\mathbf{W}$ and $\mathbf{H}$ to be nonnegative, NMF is able to discover parts-based representations of the objects that it is used to model. For example, NMF can model images of faces as the compositions of different facial features, such as eyes, noses, and lips. These representations are notably different than those discovered by SVD or vector quantization, and they have been explored for many different applications [21, 35, 37, 94, 111]. Models based on similar representations have also been explored for factorizations of binary matrices [8] (whose elements are either zero or one) and stochastic matrices [54, 100] (whose rows or columns sum to one).

Our work differs in motivation from NMF; whereas NMF's low-rank models seek parts-based representations of the data, ours are predicated on the search for pattern manifolds. These are two different types of low-dimensional structure that can exist (or coexist) in high-dimensional data. The inventors of NMF were, in fact, quite aware that not all pattern manifolds can be described by purely linear models [74, 102]. The reason is simple: some form of nonlinearity may be required to express the data in terms of its essential degrees of freedom. Note also that due to the nonnegativity constraints on $\mathbf{W}$ and $\mathbf{H}$, NMF generally requires factorizations of *higher* rank than SVD to achieve the same degree of approximation. Thus, while NMF excels at discovering parts-based representations, it cannot reveal many other types of low-dimensional structure (e.g., the example in Figure 2.2).

3.2. Exponential-family Principal Component Analysis. A different low-rank model for nonnegative or binary data was developed by Collins, Dasgupta, and Schapire [24]. Their approach can be viewed as an extension of generalized linear modeling [84] to the problem of matrix factorization. As usual, let $\mathbf{X}$ denote the matrix of data, and let $\boldsymbol{\Theta}$ denote a low-rank matrix parameter of the same size as $\mathbf{X}$. In this model, the type of data is paired with its appropriate distribution from the exponential family. Thus, for binary matrices, the model maximizes the log-likelihood

$$\log P(\mathbf{X}|\boldsymbol{\Theta}) = -\sum_{ij}\left[X_{ij}\log\left(1+e^{-\Theta_{ij}}\right)+(1-X_{ij})\log\left(1+e^{\Theta_{ij}}\right)\right], \tag{3.1}$$

whose form is derived from the Bernoulli distribution for binary random variables, while for matrices of whole numbers, it maximizes the log-likelihood

$$\log P(\mathbf{X}|\boldsymbol{\Theta}) \;=\; \sum_{ij}\left[X_{ij}\log\Theta_{ij}-\Theta_{ij}-\log(X_{ij}!)\right] \tag{3.2}$$

whose form is derived from the Poisson distribution over nonnegative counts. (In the latter case, the matrix $\boldsymbol{\Theta}$ is also constrained to be nonnegative.) The model is known as exponential-family principal component analysis (efPCA) because it contains, as a special case, the standard method of principal component analysis (PCA): this is the case when the model employs a Gaussian distribution over real-valued matrix elements. The model for binary data in (3.1) is known as logistic PCA.

As in NMF, these models are estimated by parameterizing $\boldsymbol{\Theta} = \mathbf{WH}$ as the product of smaller matrices and alternately optimizing over the factors $\mathbf{W}$ and $\mathbf{H}$. In efPCA, each of these alternating optimizations is convex (although it is not generally possible to derive closed-form updates). For binary data, in particular, these optimizations take the form of logistic regressions. It is also possible, by introducing an auxiliary function, to optimize (3.1) by an alternating least-squares method [101].

Our model for NMD uses elementwise nonlinearities in a similar way to efPCA. But whereas efPCA is rooted in distributions from the exponential family, NMD is formulated as a Gaussian latent variable model—even for data that is binary or nonnegative. (In NMD, the values of observed data arise by quantizing or clipping the model's Gaussian latent variables, an idea that has been explored in a wide variety of contexts [1, 6, 5, 25, 36, 43, 87, 97, 131].) As we shall see, it is due to this formulation that NMD can optimize $\boldsymbol{\Theta}$ directly without resorting to an alternating minimization over its smaller factors.

Naturally, it is also possible to develop models that combine intuitions from both NMF and efPCA. This has been done mainly for binary data: such models maximize the log-likelihood in (3.1) while in addition constraining one or both of the factors $\mathbf{W}$ and $\mathbf{H}$ to be nonnegative [70, 114].

3.3. Probabilistic models. Many researchers have more fully developed probabilistic models of matrix factorization [3, 10, 17, 39, 48, 86, 88, 93, 112, 113]. These models often incorporate a prior distribution over possible factorizations, and some also seek to learn a proper distribution that describes how the data were generated. The estimation of these models can involve extra complications, such as additional iterative procedures—based on Markov chain Monte Carlo (MCMC) [89] or variational methods [61]—for approximate inference. On the other hand, these models are often better regularized and easier to interpret. In this section, we survey some especially prominent applications of these models to nonnegative and binary matrices.

One such application has been topic modeling. While NMF can be used to factor large word document matrices, it does not provide a fully probabilistic model of how words come to appear in text. Such a model was developed by Blei, Ng, and Jordan [10] to discover prevalent topics that run through a large corpus of documents. The model is known as latent Dirichlet allocation (LDA) because it places a Dirichlet prior over the distribution of topics in a document; the number of topics in LDA plays a role analogous to the rank of the factors in NMF. It is not possible to perform an exact inference over the latent variables in LDA, but it is possible to scale variational methods for approximate inference to very large corpora [53]. The model is trained to estimate a distribution over words for each topic (analogous to the nonnegative matrix $\mathbf{W}$ in NMF), and in the course of this training, it infers a posterior distribution over topics for each document (analogous to the nonnegative matrix $\mathbf{H}$). Thus LDA discovers similar parts-based representations of documents as NMF, but the topics in LDA are even easier to interpret by virtue of the model's explicitly probabilistic semantics. As mentioned earlier, NMD was not conceived to learn parts-based representations of data, nor is it specialized to count-based data. It is possible, however, that it could serve as a complement to LDA for topic modeling; we discuss this possibility further in section 6.

Probabilistic models of matrix factorization have also been widely applied to recommender systems [66]; most relevant to NMD are those that explicitly model the user-item matrix as nonnegative or binary. One such framework was introduced by Mnih and Salakhutdinov [88], who used a Gaussian model with a sigmoid nonlinearity to bound matrix elements between minimum and maximum ratings. A related approach—known as logistic matrix factorization—was taken by Johnson [60] to model the probability of a user choosing an item; it extends the binary model from efPCA [24] by adding bias terms and Gaussian priors on the factors. Large-scale models for binary data have also been investigated in a Bayesian framework using variational methods for approximate inference [48]. While some user-item matrices record ratings on a binary or Likert scale, others record the number of times that users have purchased items; this type of data is most naturally treated in models of Poisson matrix factorization [39, 40]. Like LDA, Poisson matrix factorization relies on variational methods for approximate inference of its latent variables. For very sparse matrices of implicit feedback data, it also has two key advantages: first, it can incorporate domain knowledge through informative priors, and second, its computations scale linearly in the number of *nonzero* elements of the user-item matrix.

Probabilistic models of matrix factorization have also been widely applied to linkage analysis in social and biological networks. Typically, these networks are represented by binary matrices where zeros and ones indicate the absence and presence of links. A low-rank factorization for such matrices was explored in an influential line of work by Hoff [51, 52]. Hoff's initial eigenmodels were estimated using Bayesian methods, but later approaches by Wu, Levina, and Zhu [124] and Ma, Ma, and Yuan [78] led to more efficient algorithms based on projected gradient methods. These later approaches also substituted a logistic link function for the probit link function in Hoff's model. The factorizations in these models are more general than what we explore here, as they incorporate auxiliary information about individual nodes in the network. But even ignoring these terms, it is fair to say that these models differ considerably in both their mechanics and motivation from our approach. In particular, unlike Hoff's model, we do not attempt a Bayesian treatment (thereby avoiding the expense of MCMC methods), and unlike the projected gradient methods, the optimizations for NMD do not involve the tuning of variable or adaptive step sizes. Finally, NMD was motivated more generally for sparse rectangular matrices with nonnegative elements as opposed to square (and typically, symmetric) matrices with binary elements.

In addition to the above applications, there has also been a great deal of methodological and theoretical interest in Gaussian latent variable models with elementwise nonlinearities. These models have been widely developed for matrices of binary, ordinal, and mixed data [3, 5, 6, 25, 36, 43, 87, 97], most recently through the use of copula methods [1, 130, 131]. All of these models use an elementwise nonlinearity to relate an observed (or partially observed) matrix of values to an unobserved matrix of Gaussian latent variables; this is also the starting point of our work. To the best of our knowledge, though, such models have not been empirically or systematically investigated for sparse nonnegative data using an elementwise ReLU nonlinearity. We develop these ideas further in section 4.

3.4. Theoretical guarantees. Suppose that $\mathbf{X} \approx f(\mathbf{\Theta})$, where f is an elementwise nonlinearity and $\mathbf{\Theta}$ is a low-rank matrix. Then each observed element of $\mathbf{X}$ provides some information about the corresponding element of $\mathbf{\Theta}$. It is known that under certain assumptions, a low-rank matrix $\mathbf{\Theta}$ can be recovered (or completed) given the values of only a small fraction of its elements [14, 18, 62]. It is natural to ask whether these results can be extended to the nonlinear setting where $\mathbf{\Theta}$ must be additionally inferred from observed elements of the higher-rank matrix $\mathbf{X} \approx f(\mathbf{\Theta})$.

This question has been investigated by a number of authors. Davenport et al. [25] studied the problem of 1-bit matrix completion, where $\mathbf{X}$ is a partially observed binary matrix, and presented a convex program to recover $\mathbf{\Theta}$ with high accuracy. Bhaskar [5] studied the more general problem, where the elements of $\mathbf{X}$ are quantized, but not necessarily binary, and presented a globally convergent algorithm with similar guarantees of recovery; her paper also compares a number of related approaches [13, 16, 42, 68, 67, 69, 106] for these problems. More recently, Mazumdar and Rawat [83] studied this problem where $f(z) = \max(0, z)$ is a ReLU operation and proved that a maximum likelihood estimate could recover $\mathbf{\Theta}$ with high probability and small error. (They did not, however, describe or empirically investigate an algorithm for maximum likelihood estimation.) Ganti Balzano, and Willett [36] described an even more general approach to jointly estimate both the low-rank matrix $\mathbf{\Theta}$ and the elementwise nonlinearity f; they assume only that the function f is monotonic, and within this framework, they also provide bounds on the mean squared error of the recovered matrix. Finally, more specialized guarantees are also available for the recovery of rank-one matrices [6, 7], and nonrigorous results have been obtained in the thermodynamic limit of very large matrices [75].

At the moment, we cannot provide similar theoretical guarantees for the method in this paper. Our approach differs from the above lines of work in two main respects: first, we assume that the matrix $\mathbf{X}$ of data is fully observed, and second, we are motivated by the possibility that $\mathrm{rank}(\mathbf{\Theta}) \ll \mathrm{rank}(\mathbf{X})$, thus revealing the low-dimensional structure of some underlying manifold. The latter hypothesis may be at odds with typical assumptions that the matrix $\mathbf{\Theta}$

is incoherent or not very spiky. In particular, many of the above analyses require the elements of $\mathbf{\Theta}$ to be of order unity or similarly bounded; note, however, that this intuition does not hold for the example of subsection 2.3, where from (2.6) we see that $\max_{ij}|\Theta_{ij}| \sim O(n^2)$. In addition, many of the above guarantees are obtained by minimizing a convex surrogate for the rank of a matrix (e.g., the nuclear norm). But other work in matrix completion has shown that the manifold structure of data is sometimes best recovered by pursuing a nearly opposite objective (e.g., *maximizing* the trace of a positive semidefinite matrix [110, 120, 121]).

3.5. Generalized low-rank models.

There have been many efforts to treat both probabilistic and nonprobabilistic models of matrix factorization in a unified framework [41, 81, 103, 107, 119]. These efforts have been able to identify core optimizations at the heart of many different models. As above, let us denote the factors in these models by $\mathbf{W}$ and $\mathbf{H}$. Then it has been observed, across a very wide range of low-rank models, that their optimizations are biconvex in these factors. This observation has led to an extensive study of alternating minimization algorithms for low-rank models. Such algorithms reflect a common wisdom, namely, that in any nonconvex problem, it is often expeditious to identify and solve the largest subproblems that are convex or otherwise tractable.

The latent variable model in this paper presents an intriguing special case where this wisdom can be even more powerfully applied. The model has a matrix parameter $\mathbf{\Theta}$ of fixed rank, and its overall optimization is nonconvex. Nevertheless it is not necessary to parameterize the matrix $\mathbf{\Theta}$ as the product of smaller factors; instead it is possible to update the low-rank matrix $\mathbf{\Theta}$ directly, without line searches or learning rates. In addition, these updates are guaranteed to converge monotonically in the model's likelihood. Thus our approach demonstrates the potential for larger substructures to be exploited in the optimization of some generalized low-rank models.

3.6. Exceptions to the above.

It is worth noting some exceptions to the above, where generalized low-rank models were estimated by a novel application of SVD rather than an alternating optimization over factors. There are two studies, in particular, that strongly motivated our work.

The first is the continuous latent variable model for binary data in Lee and Sompolinksy [74]. Their model was fit by a moment-matching method that takes an especially simple form for unbiased data (where each bit has equal probability to be zero or one). Our work extends their approach in two ways—first, by showing that similarly motivated models can be fit by maximum likelihood estimation, and second, by considering the case of sparse nonnegative data, to which their moment-matching method is less easily generalized. We also adapted the example in subsection 2.3 from their paper.

The second is the work of Srebro and Jaakkola [108] on weighted low-rank approximations. As discussed in subsection 2.1, the errors of low-rank approximations are conventionally measured by a Frobenius norm that treats all matrix elements with equal weight. Suppose, however, that the approximation is chosen to minimize a weighted sum of elementwise errors. For this case, Srebro and Jaakkola derived an EM algorithm where a truncated SVD is used to improve the approximation at each iteration. Our work is based on an analogous application of EM, but to low-rank models with an elementwise nonlinearity.

Finally, on a related note, we mention a recent study on nonnegative low-rank matrix approximations [104], in which a low-rank nonnegative matrix is estimated without assuming that it can be written as the product of two smaller nonnegative matrices. This is a novel generalization of NMF that can discover significantly better low-rank approximations—as measured by the Frobenius norm—to a nonnegative matrix of higher rank. We note, however, that the quality of these approximations is still (necessarily) worse than what is provided by a truncated SVD, which is not hampered by nonnegativity constraints. Our method differs from this approach in two important ways: first, it focuses exclusively on sparse matrices, and second, for these matrices, it exploits an elementwise nonlinearity to obtain decompositions of even lower rank.

3.7. Semidiscrete decompositions.

Yet another type of matrix decomposition was explored in a series of papers by O'Leary and Peleg [90] and Kolda and O'Leary [64, 65]. They studied a semidiscrete decomposition (SDD) whose goal is to express a matrix as a weighted sum of constrained outer products. Unlike the models considered earlier, however, this decomposition imposes a discrete constraint: the elements of vectors in these outer products are chosen from the set $\{-1, 0, 1\}$. This decomposition can provide comparable approximations as SVD while requiring much less storage.

Our motivation in this paper differs from that of SDD. In particular, with NMD we are not seeking a low-rank decomposition that requires the least amount of storage, but one that suggests or hints at the number of underlying degrees of freedom in the data. Generally speaking, while the quantized elements of SDD are ideally suited to compress dense matrices, they are less apt to model continuous modes of variability. On the other hand, while NMD is ideally suited to analyze sparse matrices, it is not designed to provide further levels of compression.

4. Model.

In this section we formally describe our approach. We start by formulating the latent variable model for NMD and deriving its log-likelihood. Next we discuss the twin problems of inference and parameter estimation; in NMD, as in most latent variable models, these problems are closely intertwined. Finally we describe how we initialize the model and test for convergence.

4.1. Formulation.

NMD shares many aspects of previous low-rank models, particularly in its interplay of Gaussian latent variables and elementwise nonlinearities. NMD is used to analyze a $d \times n$ matrix $\mathbf{X}$, where each d-dimensional column of $\mathbf{X}$ stores a single instance of some data set and n denotes the number of such examples. Next we use a similarly sized matrix $\mathbf{\Theta}$ of rank $r < \min(d, n)$ to parameterize a Gaussian latent variable model for the data [51, 108, 112]. In particular, given the matrix $\mathbf{\Theta}$, we generate a distribution over nonnegative or binary matrices in the following way. First, for each element Θ_{ij}, we sample a Gaussian latent variable

$$Z_{ij} \sim \mathcal{N}\left(\Theta_{ij}, \sigma^2\right), \tag{4.1}$$

where the variance σ^2 is a single additional parameter of the model that we will estimate. Then, we obtain the matrix $\mathbf{X}$ deterministically from the elementwise nonlinear mapping

$$X_{ij} = f(Z_{ij}), \tag{4.2}$$

where $f(z) = \frac{1}{2}[1 + \mathrm{sign}(z)]$ for binary data and $f(z) = \max(0, z)$ for nonnegative data. Thus the nonlinearity ensures that $X_{ij} \in \{0, 1\}$ in the former case and $X_{ij} \in [0, \infty)$ in the latter. Our main interest is in the case of sparse nonnegative data, but we present both models for completeness.

The model is estimated by maximizing the likelihood of the data $\mathbf{X}$ in terms of the matrix $\boldsymbol{\Theta}$ and variance σ^2. To obtain this likelihood, however, we must first compute the marginal distribution $P(X_{ij}|\Theta_{ij}, \sigma^2)$ for an individual observed element; this is done by integrating over those values of its corresponding latent variable Z_{ij} that are consistent with the observation $X_{ij} = f(Z_{ij})$. For zero elements of $\mathbf{X}$, we note that $f(z) = 0$ if and only if $z \leq 0$. Thus we have

$$P\left(X_{ij} = 0|\Theta_{ij}, \sigma^2\right) = \int_{-\infty}^{0} dz\, P\left(Z_{ij} = z|\Theta_{ij}, \sigma^2\right) = \Phi(-\Theta_{ij}/\sigma), \tag{4.3}$$

where $\Phi(\cdot)$ denotes the cumulative distribution function for a normal distribution with zero mean and unit variance. For nonzero elements of $\mathbf{X}$, the form of the marginal distribution depends on the type of data. For *binary* data, where all nonzero elements of $\mathbf{X}$ are equal to one, we can simply use the complement of the previous result,

$$P(X_{ij} = 1|\Theta_{ij}, \sigma^2) = 1 - P\left(X_{ij} = 0|\Theta_{ij}, \sigma^2\right). \tag{4.4}$$

On the other hand, for *nonnegative* data, we note that $X_{ij} = Z_{ij}$ if and only if $Z_{ij} > 0$. Hence for positive values $x > 0$, we have

$$P(X_{ij} = x|\Theta_{ij}, \sigma^2) = P\left(Z_{ij} = x|\Theta_{ij}, \sigma^2\right), \tag{4.5}$$

where Z_{ij} is normally distributed by (4.1). From the above results we easily obtain the overall log-likelihood $\log P(\mathbf{X}|\Theta, \sigma^2)$ of the data under this model. First we observe that in NMD, the elements of $\mathbf{X}$ are conditionally independent given those of $\boldsymbol{\Theta}$. Then it follows that

$$\log P(\mathbf{X}|\boldsymbol{\Theta}, \sigma^2) = \sum_{ij} \log P(X_{ij}|\Theta_{ij}, \sigma^2). \tag{4.6}$$

The parameters $\boldsymbol{\Theta}$ and σ^2 in our model are estimated by maximizing this sum, thus accounting for all the observed elements in $\mathbf{X}$.

Before proceeding down this path, it is instructive to compare the form of the log-likelihood in (4.6) to the log-likelihoods for efPCA in subsection 3.2. For binary data, the log-likelihoods in (3.1) and (4.6) are qualitatively similar, except that the logistic model in efPCA is replaced by a probit model in NMD (as in many Gaussian latent variable models for binary-valued matrices [3, 51]). On the other hand, for nonnegative data, the log-likelihood in (4.6) is qualitatively different than common forms of efPCA. In particular, unlike efPCA based on the Poisson distribution, NMD is not specialized to count-based data, nor does it constrain the mean and variance of the distribution $P(X_{ij}|\Theta_{ij})$ to have equal values. Also, unlike efPCA based on the exponential distribution, NMD employs a distribution $P(X_{ij}|\Theta_{ij}, \sigma^2)$ that can be peaked at or away from zero.

Finally, we observe that the model for binary data in (4.1) and (4.2) has a degeneracy in its parameterization: in particular, the model's predictions in (4.3) and (4.4) are invariant under the change of parameters $\mathbf{\Theta} \to \lambda\mathbf{\Theta}$ and $\sigma^2 \to \lambda^2\sigma^2$, where $\lambda > 0$. The model can be made identifiable in this case by fixing $\sigma^2 = 1$. We note, however, that a similar degeneracy arises in models of probit regression, and there it has been observed that the EM algorithm converges more quickly when a variance σ^2 is estimated alongside the model's other parameters [76]. We observed a similar phenomenon in NMD when the value of σ^2 was fixed to unity; we therefore retained and estimated the variance parameter σ^2 for all the experiments in this paper. More recent work also suggests that the EM algorithm may discover better solutions with overparameterized models [125].

4.2. Inference. With the above formulation, we can follow a well-traveled path for latent variable modeling of high-dimensional data. Given a matrix of observed data $\mathbf{X}$, we estimate the parameters $\mathbf{\Theta}$ and σ^2 of the model by maximizing the likelihood in (4.6). To do this, we avail ourselves of the well-known EM procedure [27] for latent variable models. This procedure alternates between two steps: the E-step computes the posterior means and variances of the model's latent variables, and the M-step uses these posterior statistics to reestimate the model's parameters. The EM algorithm can also be viewed as a way of solving difficult, nonconvex optimizations via a sequence of simpler and better understood procedures [57] (e.g., fitting a Gaussian mixture model via a sequence of least-squares problems). In this section, we focus on the problem of probabilistic inference; these calculations are necessary to perform the E-step of the model's EM algorithm.

We begin by noting that exact inference in this model is tractable. In particular, all the required integrals can be expressed in terms of simple functions that arise from the model's elementwise nonlinearity. Also, by design, the model does not attempt a Bayesian treatment; i.e., it does not incorporate a prior distribution over its parameters $\mathbf{\Theta}$ and σ^2. As a result, it is not necessary to resort to approximate procedures based on MCMC methods [89] or variational inference [9, 61]. Indeed, the integral in (4.3) is typical of the calculations required for inference.

The problem of inference in NMD is to compute the posterior statistics of the model's latent variables. First let us see why the posterior distributions in NMD take an especially simple form. Recall that for each observed matrix element X_{ij}, there is a corresponding latent variable Z_{ij}. Its posterior distribution is given by Bayes rule:

$$P\left(Z_{ij}|X_{ij},\Theta_{ij},\sigma^2\right) = \frac{P\left(X_{ij}|Z_{ij},\Theta_{ij},\sigma^2\right)\, P\left(Z_{ij}|\Theta_{ij},\sigma^2\right)}{P(X_{ij}|\Theta_{ij},\sigma^2)}. \tag{4.7}$$

Note that the first term in the numerator equals one if $X_{ij} = f(Z_{ij})$ and zero otherwise. Thus there are three possible cases for the posterior distribution in (4.7): (i) for the case $X_{ij}=0$, it reduces to a right-truncated Gaussian, with no probability mass for positive values of Z_{ij}; (ii) for binary data with $X_{ij}=1$, it reduces to a left-truncated Gaussian, with no probability mass for negative values of Z_{ij}; (iii) finally, for nonnegative data with $X_{ij}>0$, it reduces to a Dirac delta function centered at $Z_{ij}=X_{ij}$. The last of these cases is trivial, and the first two give rise to truncated Gaussian distributions, of the sort shown in Figure 4.1.

As we shall see, the EM algorithm for NMD relies on repeatedly computing the posterior means and variances of the latent variables Z_{ij}. As shorthand, we denote these statistics by

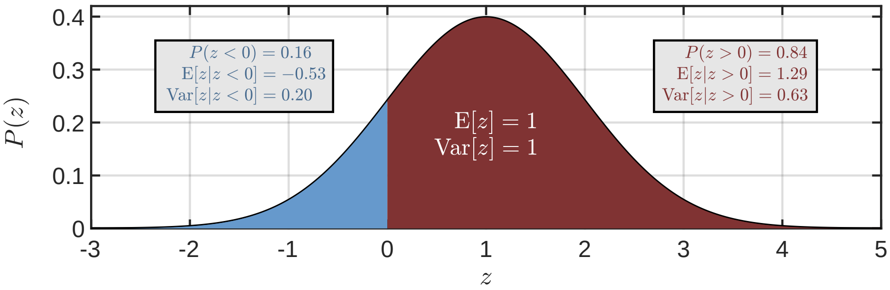


Figure 4.1. *A normal distribution can be viewed as the mixture of two truncated distributions, one for negative values (shown in blue), and one for positive values (shown in red). The means and variances of such truncated distributions, as shown in the figure, are needed to implement the EM algorithm for NMD; their forms are shown in Tables* 4.1 *and* 4.2.

Table 4.1

Inference in NMD for binary data, along with the equations in which these results are used. Results are expressed in terms of the dimensionless parameter $\gamma_{ij}=\sigma^{-1}\Theta_{ij}$ *and the functions* $\Phi(z)$, *and* $\psi(z)$ *from subsection* 4.2.

		NMD for binary data	
	likelihood	(4.6)	$P(X_{ij}=0\vert\Theta_{ij},\sigma^2) = \Phi(-\gamma_{ij})$ $P(X_{ij}=1\vert\Theta_{ij},\sigma^2) = \Phi(\gamma_{ij})$
$\overline{Z}_{ij}$	posterior mean	(4.8)	$\mathrm{E}[Z_{ij}\vert X_{ij}=0,\Theta_{ij},\sigma^2] = \Theta_{ij} - \sigma\psi(-\gamma_{ij})$ $\mathrm{E}[Z_{ij}\vert X_{ij}=1,\Theta_{ij},\sigma^2] = \Theta_{ij} + \sigma\psi(\gamma_{ij})$
$\overline{\delta Z}^2_{ij}$	posterior variance	(4.9)	$\mathrm{Var}[Z_{ij}\vert X_{ij}=0,\Theta_{ij},\sigma^2] = \sigma^2\left[1+\gamma_{ij}\psi(-\gamma_{ij})-\psi(-\gamma_{ij})^2\right]$ $\mathrm{Var}[Z_{ij}\vert X_{ij}=1,\Theta_{ij},\sigma^2] = \sigma^2\left[1-\gamma_{ij}\psi(\gamma_{ij})-\psi(\gamma_{ij})^2\right]$
$\hat{X}_{ij}$	expected observed value	(4.10)	$\mathrm{E}[X_{ij}\vert\Theta_{ij},\sigma^2] = \Phi(\gamma_{ij})$

$$\overline{Z}_{ij}=\mathrm{E}\left[Z_{ij}|X_{ij},\Theta_{ij},\sigma^2\right], \tag{4.8}$$

$$\overline{\delta Z}^2_{ij}=\mathrm{E}\left[(Z_{ij}-\overline{Z}_{ij})^2|X_{ij},\Theta_{ij},\sigma^2\right]. \tag{4.9}$$

The nontrivial statistics that we need are illustrated in Figure 4.1. For each zero element of $\mathbf{X}$, we must compute the posterior mean and variance of a distribution that has been truncated on the right (shown in blue). Likewise, for each non-zero element of $\mathbf{X}$ (when $\mathbf{X}$ is binary), we must compute the posterior mean and variance of a distribution that has been truncated on the left (shown in red).

These statistics are most easily expressed in terms of some elementary functions, which we define next. To start, let $\varphi(z)=\frac{1}{\sqrt{2\pi}}e^{-z^2/2}$ and $\Phi(z)=\frac{1}{2}\left[1+\mathrm{erf}\left(z/\sqrt{2}\right)\right]$ denote, respectively, the probability density and cumulative distribution function for a normal distribution with zero mean and unit variance. Also, as further shorthand, let $\psi(z)=\varphi(z)/\Phi(z)$ denote

Table 4.2

Inference in NMD for nonnegative data, along with the equations in which these results are used. Results are expressed in terms of the dimensionless parameter $\gamma_{ij} = \sigma^{-1}\Theta_{ij}$ and the functions $\varphi(z)$, $\Phi(z)$, and $\psi(z)$ from subsection 4.2.

NMD for nonnegative data ($x>0$)			
	likelihood	(4.6)	$P(X_{ij}=0\mid\Theta_{ij},\sigma^2) \;=\; \Phi(-\gamma_{ij})$ $P(X_{ij}=x\mid\Theta_{ij},\sigma^2) \;=\; \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{1}{2\sigma^2}(x-\Theta_{ij})^2}$
$\overline{Z}_{ij}$	posterior mean	(4.8)	$\mathrm{E}[Z_{ij}\mid X_{ij}=0,\Theta_{ij},\sigma^2] \;=\; \Theta_{ij} \;-\; \sigma\psi(-\gamma_{ij})$ $\mathrm{E}[Z_{ij}\mid X_{ij}=x,\Theta_{ij},\sigma^2] \;=\; x$
$\overline{\delta Z^2_{ij}}$	posterior variance	(4.9)	$\mathrm{Var}[Z_{ij}\mid X_{ij}=0,\Theta_{ij},\sigma^2] \;=\; \sigma^2\left[1 \;+\; \gamma_{ij}\psi(-\gamma_{ij}) \;-\; \psi(-\gamma_{ij})^2\right]$ $\mathrm{Var}[Z_{ij}\mid X_{ij}=x,\Theta_{ij},\sigma^2] \;=\; 0$
$\hat{X}_{ij}$	expected observed value	(4.10)	$\mathrm{E}[X_{ij}\mid\Theta_{ij},\sigma^2] \;=\; \Theta_{ij}\Phi(\gamma_{ij}) \;+\; \sigma\varphi(\gamma_{ij})$

the ratio of the above functions. Then straightforward calculations, of the sort in (4.3), give the posterior statistics in (4.8) and (4.9) in terms of these functions. For convenience, the results of these calculations are collected in Tables 4.1 and 4.2, respectively, for NMD with binary and nonnegative data.

The above are not the only inferences that we can make from the model. We can also compute the expected value of an observed matrix element:

$$\mathrm{E}\left[X_{ij}|\Theta_{ij},\sigma^2\right] \;=\mathrm{E}\left[f(Z_{ij})|\Theta_{ij},\sigma^2\right] \;=\int_{-\infty}^{\infty}\! dz\,\frac{1}{\sqrt{2\pi\sigma^2}}\,e^{-\frac{1}{2\sigma^2}(z-\Theta_{ij})^2}\,f(z). \tag{4.10}$$

The results of this calculation are also shown in Table 4.1 when f is a threshold nonlinearity (for binary data) and in Table 4.2 when f is a ReLU nonlinearity (for nonnegative data). As evident from (4.10), these expected values behave very simply in the limit of vanishing variance with $\mathrm{E}[X_{ij}|\Theta_{ij},\sigma^2] \to f(\Theta_{ij})$ as $\sigma^2 \to 0$. In practice, of course, the estimated value of σ^2 will never be exactly equal to zero. Instead, when σ^2 is small but finite, the expected value $\mathrm{E}[X_{ij}|\Theta_{ij},\sigma^2]$ approaches this limiting behavior as shown in Figure 4.2.

4.3. Learning. We derive parameter updates for NMD using the EM procedure for maximum likelihood estimation in latent variable models [27]. In particular, given the model's current parameter estimates $\boldsymbol{\Theta}$ and σ^2, this procedure yields updated estimates $\tilde{\boldsymbol{\Theta}}$ and $\tilde{\sigma}^2$ that increase the log-likelihood in (4.6). Since the derivation may be of less interest than the final result, we begin by presenting the EM updates as a fait accompli:

$$\tilde{\boldsymbol{\Theta}} = \mathrm{argmin}_{\boldsymbol{\Theta}} \left\|\boldsymbol{\Theta} - \overline{\mathbf{Z}}\right\|_F \quad \text{such that} \quad \mathrm{rank}(\boldsymbol{\Theta}) = r, \tag{4.11}$$

$$\tilde{\sigma}^2 = \frac{1}{dn}\sum_{ij}\left[(\overline{Z}_{ij} - \tilde{\Theta}_{ij})^2 \;+\; \overline{\delta Z^2_{ij}}\right]. \tag{4.12}$$

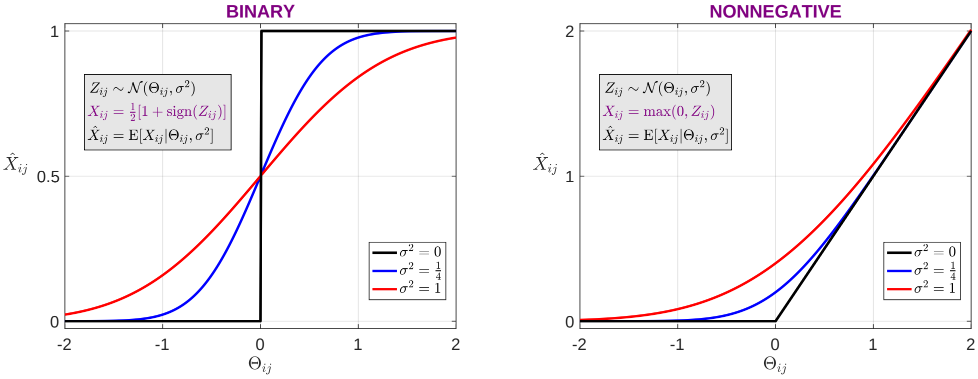


Figure 4.2. *Expected value* $\hat{X}_{ij} = \mathrm{E}[X_{ij}|\Theta_{ij}, \sigma^2]$ *from NMD as a function of* Θ_{ij} *for different values of* σ^2*. The expectation depends on whether the data are modeled as binary (left) or nonnegative (right); see* (4.10) *and Tables* 4.1 *and* 4.2*.*

Note how simply in these expressions the model parameters are reestimated from the posterior statistics in (4.8) and (4.9) of the model's latent variables. We make a number of further observations. First, the update in (4.11) is obtained by performing a truncated SVD of the matrix $\overline{\mathbf{Z}}$ of posterior means. Thus we see that the update leverages the full power of SVD to optimize (at each iteration) over the nonconvex set of low-rank matrices in its parameter space. Second, at each iteration, the updates should be performed in the order shown; this is necessary because the result for $\tilde{\sigma}^2$ in (4.12) depends on the result for $\tilde{\boldsymbol{\Theta}}$ in (4.11). Finally, we observe that these updates have the desirable guarantee of increasing the likelihood in (4.6) at each iteration (except at stationary points). In practice, we iterate these steps to a desired level of convergence. The final result is a nonlinear low-rank decomposition $\mathbf{X} \approx f(\boldsymbol{\Theta})$, where the error of the approximation is modeled by the magnitude of σ^2.

Let us now show, in more detail, how these updates are derived. The EM algorithm works, at each iteration, by calculating a surrogate for the log-likelihood that is easier to optimize [27]. For NMD, this surrogate is given by

$$\begin{aligned}
&\mathrm{E}\left[\log P\big(\mathbf{Z}|\tilde{\boldsymbol{\Theta}}, \tilde{\sigma}^2\big)\,\Big|\,\mathbf{X}, \boldsymbol{\Theta}, \sigma^2\right] \\
&\qquad = \mathrm{E}\left[-\frac{dn}{2}\log(2\pi\tilde{\sigma}^2) - \frac{1}{2\tilde{\sigma}^2}\big\|\mathbf{Z} - \tilde{\boldsymbol{\Theta}}\big\|_F^2\,\Big|\,\mathbf{X}, \boldsymbol{\Theta}, \sigma^2\right], \\
&\qquad = -\frac{dn}{2}\log(2\pi\tilde{\sigma}^2) - \frac{1}{2\tilde{\sigma}^2}\sum_{ij}\Big[(\overline{Z}_{ij} - \tilde{\Theta}_{ij})^2 + \overline{\delta Z}^2_{ij}\Big],
\end{aligned} \tag{4.13}$$

where $\overline{\mathbf{Z}}$ and $\overline{\delta\mathbf{Z}^2}$ denote the posterior means and variances of the model's latent variables, as defined in (4.8) and (4.9). The M-step of the EM algorithm reestimates the model's parameters by maximizing the expression in (4.13). This maximization is straightforward and leads immediately to the reestimation formulas in (4.11) and (4.12).

4.4. Initialization and convergence. The optimization of the log-likelihood in (4.6) is not convex and cannot be guaranteed to reach a global maximum. In practice, this means that the results from the EM algorithm can depend on how the model parameters are initialized.

We adopted the following simple heuristic for initializing these parameters. First we computed the mean and variance of all the elements in $\mathbf{X}$, given by

$$\overline{x} = \frac{1}{dn}\sum_{ij} X_{ij} \quad \text{and} \quad \overline{\delta x^2} = \frac{1}{dn}\sum_{ij}(X_{ij} - \overline{x})^2. \tag{4.14}$$

Then we used these statistics to initialize the model parameters $\boldsymbol{\Theta}$ and σ^2. Specifically, for nonnegative data, we set $\Theta_{ij} = \overline{x}$ and $\sigma^2 = \overline{\delta x^2}$, while for binary data, we set $\Theta_{ij} = \Phi^{-1}(\overline{x})$ and $\sigma^2 = 1$. Note that in both cases $\boldsymbol{\Theta}$ was initialized by a rank-one matrix in which every element is equal.

We also adopted a simple heuristic for evaluating the convergence of the EM updates. Specifically, we terminated the updates if either (a) they did not improve the log-likelihood (normalized per matrix element) by a minimal increment of 10^{-5}, or (b) they had already been applied for a maximum of 512 iterations. In general, we observed that the lower-rank models for NMD were more likely to meet the first of these criteria for early stopping.

For the binary model of NMD, we supplemented these updates with one further optimization for the variance σ^2. In preliminary experiments, we observed that the estimates for σ^2 converged rather slowly at the end of learning. (In this regime, it appears that the surrogate function in (4.13) does not closely track the effect of σ^2 on the log-likelihood.) To account for this, we performed one additional (golden-section) search of values for the variance σ^2 while holding $\boldsymbol{\Theta}$ fixed. In practice this final search led to smaller estimates for the variance σ^2 and larger values for the log-likelihood in (4.6).

5. Experiments.

We investigated the performance of NMD on five large matrices derived from data sets of binary, grayscale, and color images. We describe these matrices and data sets in subsection 5.1, and we present our empirical results on them in subsection 5.2. Finally, in subsection 5.3, we compare the results from NMD on binary data to the closely related model of logistic PCA.

5.1. Data sets.

We describe the data sets in order from least to most complex. Our first data set (DOTS) consisted of 64×64 binary images, with each image containing a single black dot of fixed radius against a white background. Some representative images from this data set are shown in the top left plot of Figure 5.1. The data set had $n = 1024$ images, each containing $d = 4096$ pixels, of which a very small fraction ($\rho = 0.011$) were shaded. This data set can be viewed as a higher-dimensional analog of the example in Figure 2.2; in this case, the images lie on a pattern manifold with two translational degrees of freedom. One might expect these data to be modeled very well by an NMD of rank $r = 5$; intuitively, NMD should require two additional eigenvectors beyond those in Figure 2.2 to account for the extra translational degree of freedom. We will confirm this hypothesis in the next section.

Our second data set (SPRITES) also consisted of 64×64 binary images, but in this case, each image contained a single sprite of varying shape, size, location, and orientation. Some representative images from each class of shapes—squares, ovals, and hearts—are shown in the top right plot of Figure 5.1. The images in our experiments were evenly subsampled from a larger publicly available data set of sprite images [82]; our subsample had $n = 15246$ images, each containing $d = 4096$ pixels, of which a small fraction ($\rho = 0.048$) on average were shaded.

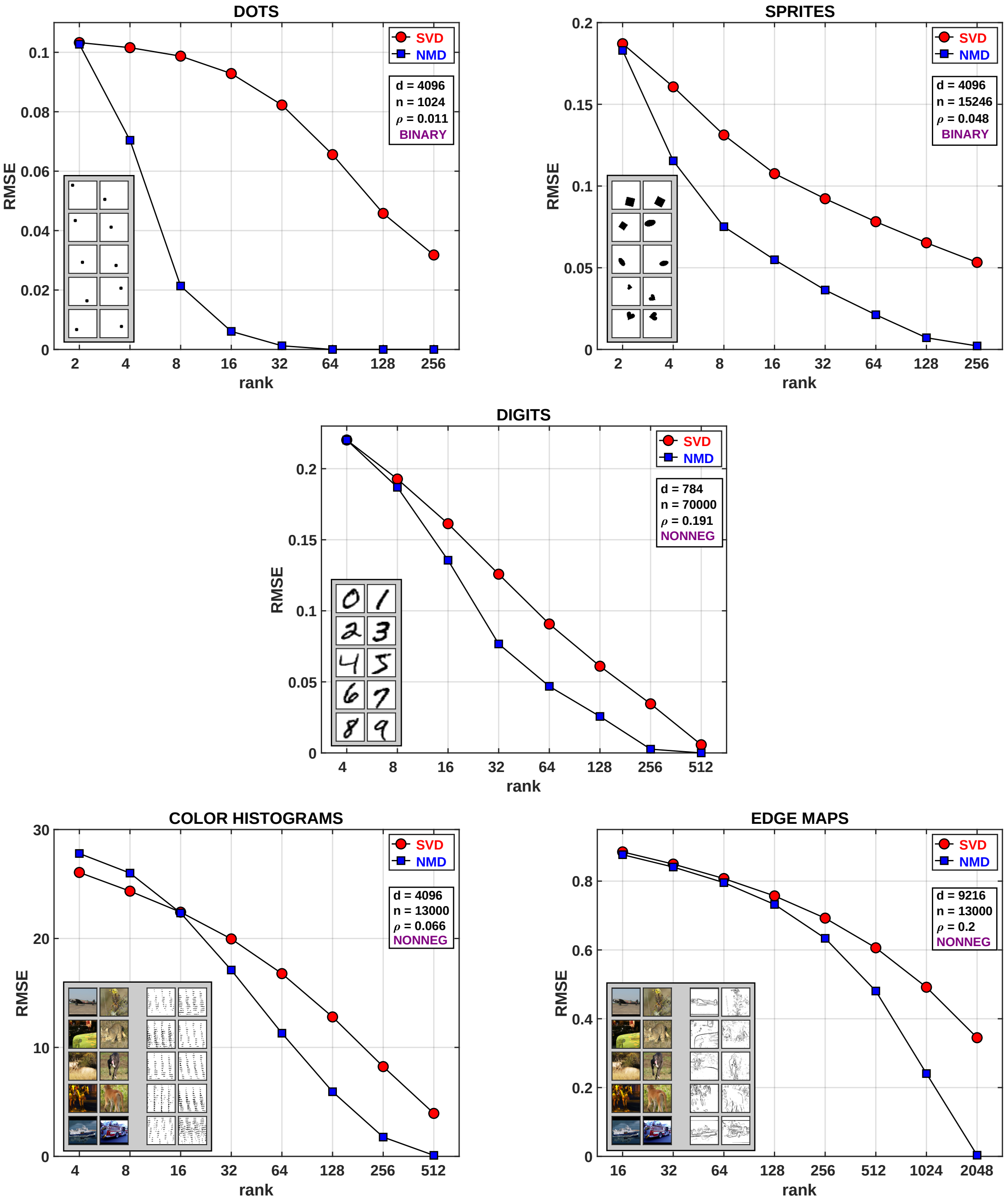


Figure 5.1. *For matrix decompositions of the same rank, NMD can produce significantly more accurate reconstructions than SVD. The panels compare the results from these methods on two data sets with binary values (top) and three data sets with nonnegative values (middle and bottom). Each data set was represented as a $d \times n$ sparse matrix that contains some fraction ρ of nonzero elements. A few representative examples from each data set are also shown.*

This data set has more variability than the previous one, but the basic degrees of freedom remain simple to characterize.

Our third data set (DIGITS) consisted of 28 × 28 grayscale images of MNIST handwritten digits [71]. A representative image from each digit class is shown in the center plot of Figure 5.1. We experimented on the $n = 70000$ combined images of the training and test set. These images are relatively sparse, with a small fraction ($\rho = 0.191$) of pixels on average having nonzero values. The variability in this data set arises not only from different classes of digits, but also from different templates within each class as well as variable aspects of penmanship (e.g., slant, thickness).

Our fourth and fifth data sets were derived from 96 × 96 color images in the STL-10 data set [23]. For these experiments, we used the subset of $n = 13000$ labeled images, each containing an object from one of ten classes. A representative image from each digit class is shown in the bottom plots of Figure 5.1. It is evident that these images are not sparse—and thus not suitable in their own right for models of NMD. However, there are many ways to derive sparse high-dimensional descriptors from dense color images, and we investigated two ways in particular, which we describe next.

For our fourth data set, we computed color histograms for the labeled subset of STL-10 images. To compute these histograms, we first requantized each color channel from 8 bits to 4 bits, so that each RGB pixel was represented by a 12-bit number. Then, for each image, we compiled a color histogram with $2^{12} = 4096$ bins. In this way we obtained a data set of $n = 13000$ sparse color histograms in which on average only a small fraction ($\rho = 0.066$) of the $d = 4096$ bins had nonzero counts. Note that each such histogram (conveniently, with $4096 = 64^2$ bins) can be also visualized as 64 × 64 grayscale image in its own right. For illustration, we have accompanied each color image in the bottom left plot of Figure 5.1 by its corresponding histogram, as visualized in this way.

For our fifth data set, we computed approximate edge maps for the labeled subset of STL-10 images. This was done by computing the image gradient magnitude (across all three color channels) at each pixel in these images and then thresholding the values of these magnitudes, zeroing out all but those in the top 20%. In this way, we obtained the grayscale edge maps shown in the bottom right plot of Figure 5.1. Thus the resulting data set consisted of $n = 13000$ edge maps, each with $d = 9216$ grayscale pixels of which a fixed fraction ($\rho = 0.2$) per image were nonzero.

5.2. Empirical results.

The main goal of our experiments was to compare the accuracy of the low-rank approximations obtained by NMD versus SVD. To this end, we estimated models of widely varying sizes on all five of the data sets described in the previous section. Specifically, for the matrix $\mathbf{X}$ of each data set, we used the updates in (4.11) and (4.12) to learn low-rank matrices $\boldsymbol{\Theta}_{\text{NMD}}$ and variances σ^2_{NMD} maximizing the log-likelihood in (4.6). In addition, as a baseline, we performed truncated SVDs to obtain low-rank matrices $\boldsymbol{\Theta}_{\text{SVD}}$ minimizing $\|\mathbf{X} - \boldsymbol{\Theta}_{\text{SVD}}\|^2_F$. We compared the quality of these approximations by attempting to reconstruct $\mathbf{X}$ from either $\boldsymbol{\Theta}_{\text{SVD}}$ or $\boldsymbol{\Theta}_{\text{NMD}}$ and then calculating the root mean squared error (RMSE) of these reconstructions. The reconstructions were computed as

$$\hat{\mathbf{X}}_{\text{SVD}} = \max(0, \min(u, \boldsymbol{\Theta}_{\text{SVD}})), \tag{5.1}$$

$$\hat{\mathbf{X}}_{\text{NMD}} = \mathrm{E}\left[\mathbf{X} | \boldsymbol{\Theta}_{\text{NMD}}, \sigma^2_{\text{NMD}}\right], \tag{5.2}$$

where the upper limit in (5.1) was set to $u = 1$ for binary data and $u = \infty$ for nonnegative data, and the RMSE was computed as

$$\text{RMSE} \;=\; \frac{1}{\sqrt{dn}}\|\mathbf{X} - \hat{\mathbf{X}}\|_F. \tag{5.3}$$

Note that in (5.1), we clipped the predicted values from SVD to lie within the bounds of the data. This always lowers the RMSE for the model obtained by SVD, thus yielding a stronger baseline. Also, in (5.2), we computed the predictions for NMD from the expected values in (4.10), thus incorporating the model's variance parameter, and not merely by setting $\hat{X}_{ij} = f(\Theta_{ij})$. In practice, the former predictions were considerably better for small model sizes (where the estimated variance is higher).

The results of these experiments are shown in Figure 5.1. Each panel compares the RMSE from SVD and NMD on a particular data set and for different values of the rank r. The results show a clear pattern. On one hand, the reconstructions from SVD and NMD behave similarly for small values of the rank (where they both capture only the coarsest properties of the data) and also for sufficiently large values (where they both reconstruct the data with high accuracy). On the other hand, between these two extremes, SVD requires factorizations of much higher rank to achieve the same overall RMSE.

Figure 5.1 provides quantitative evidence that NMD can learn more accurate low-rank models than SVD. But it is also interesting to examine the qualitative differences between these models. Some of these differences are illustrated in Figure 5.2, which shows the different reconstructions from NMD and SVD as images in their own right. The former are much sharper and contain many fewer artifacts, suggesting that NMD has better modeled the degrees of freedom at play in this data.

5.3. Comparison to logistic PCA. As mentioned in subsection 3.2, NMD decomposes binary matrices in a similar manner to logistic PCA (henceforth, σPCA). These methods can be viewed as generalizations of probit and logistic regression to the problem of matrix factorization. It is well known that probit and logistic regression produce generally comparable models for binary classification, so we might also expect NMD and σPCA to obtain comparable low-rank decompositions.

To test this hypothesis, we trained a family of comparable models for σPCA on the DOT and SPRITE data sets of binary images. For these models, we parameterized the matrix $\mathbf{\Theta}$ in (3.1) as the product of two smaller matrices $\mathbf{W}$ and $\mathbf{H}$, and we estimated these smaller matrices using an alternating least-squares method [101]. We initialized the matrices $\mathbf{W}$ and $\mathbf{H}$ with small random values and adopted the same convergence criteria as described in subsection 4.4. We also performed a final exhaustive sweep to fine-tune the overall scale of the matrix $\mathbf{\Theta}$.

The results of these experiments are shown in Figure 5.3. The figure compares NMD and σPCA over a wide range of model sizes. The main plots show that NMD and σPCA yield similar reconstruction accuracies, confirming our expectations based on the similarity of probit and logistic regression. The insets of these plots, however, compare the computation time per iteration of these algorithms, and here we see a marked difference. It is clear that the update for $\mathbf{\Theta}$ in NMD scales better with the model size than the updates for the individual factors $\mathbf{W}$ and $\mathbf{H}$ in σPCA. We explain this next.

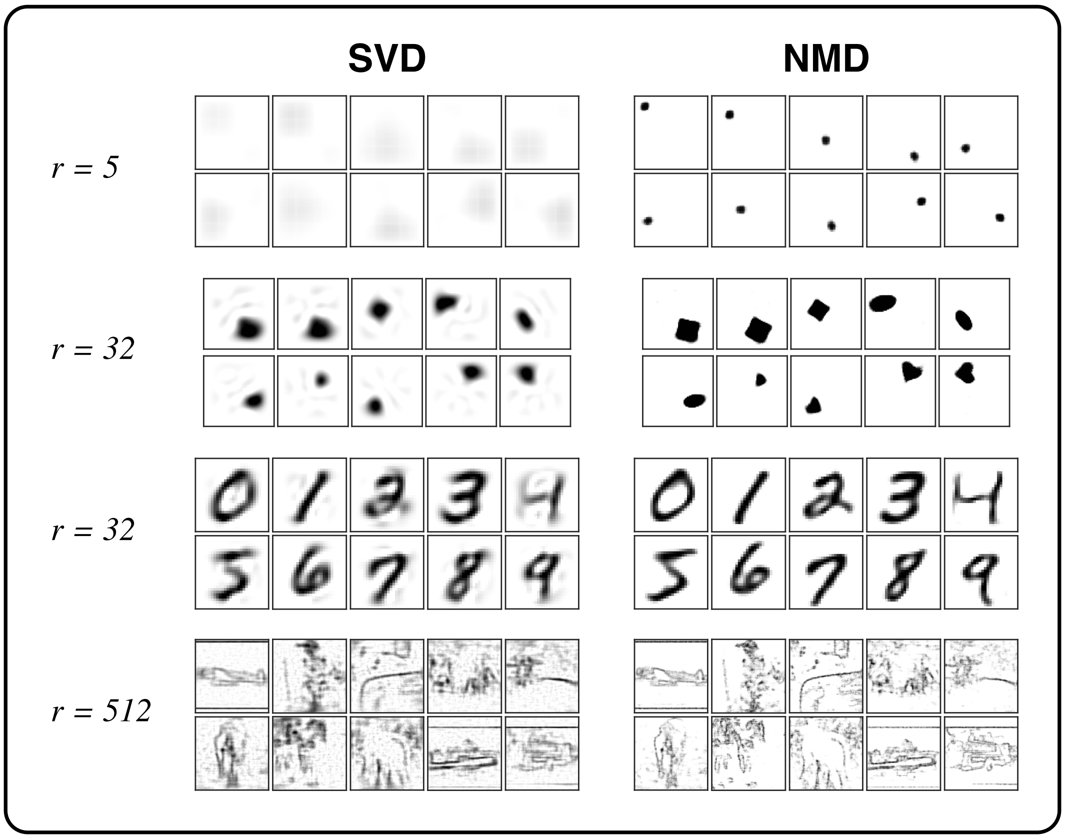


Figure 5.2. *Images of dots, sprites, handwritten digits, and edge maps are reconstructed much more accurately by low-rank models from NMD (right) than SVD (left).*

To begin, we examine the time complexity of each EM iteration for NMD. The time complexity in these experiments was dominated by the truncated SVD that maximizes (4.11). The complete SVD for a $d \times n$ matrix requires $O(\min(dn^2, d^2n))$ operations, and for matrices that are not too large, it is actually faster in MATLAB to perform a complete SVD than a truncated one. This is the case for the data set of DOT images ($d = 4096$, $n = 1024$), and so in this case the time per iteration of the EM algorithm does not depend on the desired rank r of $\mathbf{\Theta}$. This analysis is consistent with the flat curve for NMD in the left inset plot of Figure 5.3. On the other hand, for larger matrices—such as arise from the data set of SPRITE images ($d = 4096$, $n = 15426$)—it is faster to perform a truncated SVD; this overall complexity is harder to estimate for the iterative method employed by MATLAB (based on Lanczos bidiagonalization[2]), but from the right inset plot of Figure 5.3, this routine appears to scale as $O(dnr)$.

[2] We also note that for large matrices, it can be much faster in practice to perform a truncated SVD via randomized methods [44, 116, 117]. For the EM algorithm of NMD, where the matrices to be analyzed are dense, these methods should scale as $O(dnr + (d + n)r^2)$. This was not done, however, for the experiments in this paper.

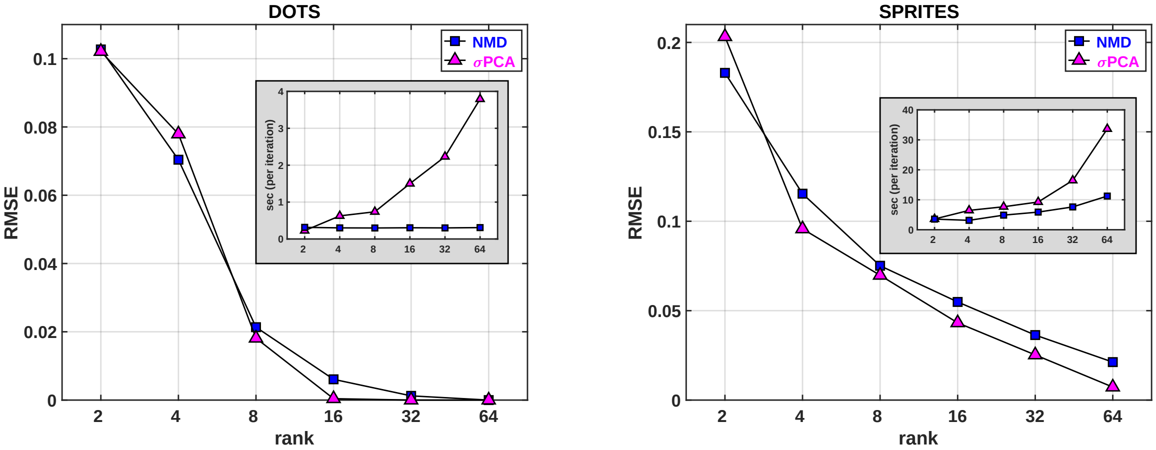


Figure 5.3. *NMD produces comparably accurate decompositions of binary matrices as σPCA via alternating least squares. As shown in the insets, however, the two approaches scale quite differently in the rank of the decomposition.*

Next we discuss the time complexity for σPCA; specifically, we consider the approach based on alternating least-squares updates [101]. In each iteration of this approach, it is necessary to construct and solve multiple systems of linear equations; in particular, a different system is needed for each row of $\mathbf{W}$ (which is $d \times r$) and each column of $\mathbf{H}$ (which is $r \times n$). In total, it requires $O(dnr^2)$ operations to construct these systems of linear equations and $O((d+n)r^2)$ operations to solve them. This analysis is consistent with the poorer scaling of σPCA in both inset plots of Figure 5.3.

In general, we observed that σPCA and NMD took comparable numbers of iterations to converge. Thus even though these models lead to similar results in reconstruction accuracy, it appears that NMD has computational advantages for all but the lowest-rank decompositions.

6. Conclusion. In this paper we have investigated a generalized low-rank model for sparse nonnegative matrices. The crux of our model is a sparsifying elementwise nonlinearity; with this nonlinearity, the model can reveal decompositions of much lower rank than linear approaches. We found this to be the case in all the data sets that we analyzed. Our results also illuminate the role of these nonlinearities in unsupervised neural networks with ReLU hidden units. Though not explored here, one hopeful idea is that NMD might serve as a primitive for layerwise learning in deep neural networks [32, 45, 50, 105, 115, 126, 127, 129]. This idea remains an important area for future work.

The starting point of our work was to introduce a low-rank matrix as the parameter of a Gaussian latent variable model. This matrix could then be reestimated, at each iteration of the model's EM algorithm, from the truncated SVD of an inferred matrix with the same number of rows and columns. It should be possible to apply NMD to much larger matrices than we considered here by computing this truncated SVD via randomized methods [44]. These methods are both simple to use and highly accurate. This is the benefit of encapsulating the model updates as truncated SVDs: the implementation of EM can seamlessly leverage other ongoing work in low-rank matrix approximation [116, 117].

There may be other ways to optimize the parameters of the latent variable model for NMD. In subsection 5.3, we observed that the EM algorithm scaled better with the rank of the approximation than an earlier implementation of σPCA. In general, however, we have not shown that the EM updates (based on truncated SVDs) converge more quickly or discover better solutions than other approaches [19] to nonconvex optimization (e.g., conjugate gradient ascent, alternating least squares). Arguably, the main virtue of the EM algorithm is conceptual: it exposes and highlights large tractable substructures of the underlying intractable optimization [57]. It seems difficult to ignore these tractable substructures once they have been revealed; that is why, perhaps, it is more usual to accelerate the fitting of latent variable models *within* the framework of EM [47, 58, 59, 76, 99, 128] than outside of it.

We noted in section 3 that NMD builds on many previous approaches for high-dimensional data analysis. Some benefits of NMD are most likely to be realized in conjunction with these other approaches. For instance, NMD seems ideally suited to further decompose the sparse parts-based representations discovered by algorithms such as NMF [72] and LDA [10]. NMF in particular is known to discover sparse basis vectors (e.g., pen strokes, facial features) that exhibit continuous modes of variability (e.g., translations, rotations) akin to those in the data sets we studied. Likewise, recall that for every document, LDA infers a high-dimensional vector of nonnegative topic proportions, and these vectors tend to be sparse for large corpora with fine-grained topics. It seems likely that NMD could detect low-rank structure in these vectors as well.

There remains a lingering debate over the merits of linear (parts-based) decompositions versus nonlinear (manifold-based) decompositions of nonnegative matrices [77]. On one hand, the former are observed to be more interpretable; on the other hand, the latter are claimed to be more compact. Suffice it to say, many data sets exhibit both parts-based and manifold-like structure. In general, it seems wiser to combine these intuitions than to insist on one at the expense of the other.

Another application of NMD may arise from problems in extreme multilabel classification [4, 56]. Specifically, consider the problem of learning a large number of binary classifiers in parallel, where all the classifiers use the same features as input, and where the labels from parallel tasks are known to be correlated. Let $\mathbf{Y}$ denote the $\ell \times n$ binary label matrix, where ℓ is the number of tasks and n is the number of training examples. For this problem, NMD could be used to learn a matrix $\mathbf{\Theta}$ of rank $r \ll \ell$ such that $\mathbf{Y} \approx \frac{1}{2}(1 + \text{sign}(\mathbf{\Theta}))$. Writing $\mathbf{\Theta} = \mathbf{WH}$, we see that each r-dimensional column of $\mathbf{H}$ provides a low-dimensional latent encoding of the ℓ observed labels for its corresponding example. Thus we can reformulate the problem of ℓ-way multilabel classification as a single nonlinear regression from the shared feature space of training examples to a shared latent space of dimensionality $r \ll \ell$.

In this paper we have not addressed the problem of missing data, namely, when some elements of the matrix $\mathbf{X}$ are not observed. In fact, the EM updates of subsection 4.3 can incorporate missing elements simply by equating the prior and posterior statistics of their corresponding latent variables, i.e., setting $\overline{Z}_{ij} = \Theta_{ij}$ and $\overline{\delta Z}^2_{ij} = \sigma^2$ whenever X_{ij} does not have an observed value. With this minor accommodation, the updates retain exactly the same form[3] as (4.11) and (4.12) for fully observed matrices. These updates may not be especially

[3] If the elementwise nonlinearity in NMD is replaced by the identity function, then this approach reduces to a special case of the model of Srebro and Jaakkola [108] for weighted low-rank approximations of real-valued matrices.

efficient, however, when the number of missing elements far exceeds the number of observed ones. A better solution may be to leverage more scalable methods for matrix completion—e.g., methods based on alternating least squares [46] or nuclear norm minimization [14]—that have been expressly developed for very large matrices with many missing entries. Likewise, in place of (4.11), we can also consider regularized decompositions that yield more interpretable solutions [122] or better performance in certain domains [66, 88, 109]. All of these methods can benefit in turn from the elementwise nonlinearity in NMD, which may yield lower-rank matrices for them to decompose. These possibilities suggest many interesting directions for future work.

Acknowledgments. This paper benefited greatly from the comments of anonymous reviewers on earlier drafts. Special thanks are also owed to L. Balzano and T. Kolda for valuable feedback.

REFERENCES

[1] C. Anderson-Bergman, T. G. Kolda, and K. Kincher-Winoto, *XPCA: Extending PCA for a Combination of Discrete and Continuous Variables*, preprint, arXiv:1808.07510, 2018.

[2] A. M. S. Ang and N. Gillis, *Accelerating nonnegative matrix factorization algorithms using extrapolation*, Neural Comput., 31 (2019), pp. 417–439.

[3] D. J. Bartholomew, M. Knott, and I. Moustaki, *Latent Variable Models and Factor Analysis: A Unified Approach*, Wiley, Chichester, UK, 2011.

[4] S. Bengio, K. Dembczynski, T. Joachims, M. Kloft, and M. Varma, *Extreme Classification*, Dagstuhl Rep., 8 (2019), pp. 62–80.

[5] S. A. Bhaskar, *Probabilistic low-rank matrix completion from quantized measurements*, J. Mach. Learn. Res. (JMLR), 17 (2016), pp. 1–34.

[6] S. A. Bhaskar and A. Javanmard, *1-bit matrix completion under exact low-rank constraint*, in Proceedings of the 49th Annual Conference on Information Sciences and Systems (CISS-15), IEEE, Piscalaway, NJ, 2015, pp. 1–6.

[7] Y. Bi and J. Lavaei, *On the absence of spurious local minima in nonlinear low-rank matrix recovery problems*, Proc. Mach. Learn. Res. (PMLR), 130 (2021), pp. 379–387.

[8] E. Bingham, A. Kaban, and M. Fortelius, *The aspect Bernoulli model: Multiple causes of presences and absences*, PAA Pattern Anal. Appl., 12 (2009), pp. 55–78.

[9] D. Blei, *Build, compute, critique, repeat: Data analysis with latent variable models*, Annu. Rev. Stat. Appl., 1 (2014), pp. 203–232.

[10] D. M. Blei, A. Y. Ng, and M. I. Jordan, *Latent Dirichlet allocation*, J. Mach. Learn. Res. (JMLR), 3 (2003), pp. 993–1022.

[11] J.-P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov, *Metagenes and molecular pattern discovery using matrix factorization*, Proc. Nat. Acad. Sci. USA, 101 (2004), pp. 4164–4169.

[12] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, *A limited memory algorithm for bound constrained optimization*, SIAM J. Sci. Comput., 16 (1995), pp. 1190–1208.

[13] T. Cai and W.-X. Zhou, *A max-norm constrained minimization approach to 1-bit matrix completion*, J. Mach. Learn. Res. (JMLR), 14 (2013), pp. 3619–3647.

[14] E. Candés and B. Recht, *Exact matrix completion via convex optimization*, Found. Comput. Math., 9 (2009), pp. 717–772.

[15] J. Canny, *A computational approach to edge detection*, IEEE Trans. Pattern Anal. Mach. Intell., 8 (1986), pp. 679–698.

[16] Y. Cao and Y. Xie, *Categorical matrix completion*, in Proceedings of the 6th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP-15), IEEE, Piscataway, NJ, 2013, pp. 369–372.

[17] A. Cemgil, *Bayesian inference for non-negative matrix factorisation models*, Comput. Intell. Neurosci. 2009 (2009), 785152.

[18] S. Chatterjee, *Matrix estimation by universal singular value thresholding*, Ann. Statist., 43 (2015), pp. 177–214.

[19] Y. Chi, Y. M. Lu, and Y. Chen, *Non-convex optimization meets low-rank matrix factorization: An overview*, IEEE Trans. Signal Process., 67 (2019), pp. 5239–5269.

[20] A. Cichocki, R. Zdunek, and S. Amari, *Hierarchical ALS algorithms for nonnegative matrix and 3d tensor factorization*, in Independent Component Analysis and Signal Separation, M. E. Davies, C. J. James, S. A. Abdallah, and M. D. Plumbley, eds., Lecture Notes in Comput. Sci. 4666, Springer, Berlin, 2007, pp. 169–176.

[21] A. Cichocki, R. Zdunek, A. H. Phan, and S. Amari, *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*, Wiley, Hoboken, NJ, 2009.

[22] A. K. Cline and I. S. Dhillon, *Computation of the singular value decomposition*, in Handbook of Linear Algebra, CRC Press, Boca Raton, FL, 2006, 45.

[23] A. Coates, A. Ng, and H. Lee, *An analysis of single-layer networks in unsupervised feature learning*, Proc. Mach. Learn. Res, (PMLR), 15 (2011), pp. 215–223.

[24] M. Collins, S. Dasgupta, and R. E. Schapire, *A generalization of principal components analysis to the exponential family*, in Advances in Neural Information Processing Systems 14, T. G. Dietterich, S. Becker, and Z. Ghahramani, eds., MIT Press, Cambridge, MA, 2002, pp. 617–624.

[25] M. A. Davenport, Y. Plan, E. van den Berg, and M. Wooters, *1-bit matrix completion*, Inf. Inference, 3 (2014), pp. 189–223.

[26] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, *Indexing by latent semantic analysis*, J. Assoc. Inform. Sci. Tech., 41 (1990), pp. 391–407.

[27] A. P. Dempster, N. M. Laird, and D. B. Rubin, *Maximum likelihood from incomplete data via the EM algorithm*, J. R. Stat. Soc. B, Stat. Methodol., 39 (1977), pp. 1–38.

[28] E. M. Dodds and M. R. M. Robert, *On the sparse structure of natural sounds and natural images: Similarities, differences, and implications for neural coding*, Front. Comput. Neurosci., 13 (2019), pp. 1–19.

[29] D. Donoho, *High-dimensional data analysis: The curses and blessings of dimensionality*, AMS Math Challenges Lecture, AMS, Providence, RI, 2000, pp. 1–32.

[30] C. Eckart and G. Young, *The approximation of one matrix by another of lower rank*, Psychometrika, 1 (1936), pp. 211–218.

[31] N. B. Erichson, A. Mendible, S. Wihlborn, and J. N. Kutz, *Randomized nonnegative matrix factorization*, Pattern Recognit. Lett., 104 (2018), pp. 1–7.

[32] J. Fan and J. Cheng, *Matrix completion by deep matrix factorization*, Neural Netw., 98 (2018), pp. 34–41.

[33] P. Foldiak and M. Young, *Sparse coding in the primate cortex*, in The Handbook of Brain Theory and Neural Networks, MIT Press, Cambridge, MA, 1995, pp. 895–898.

[34] A. Frieze, R. Kannan, and S. Vempala, *Fast Monte-Carlo algorithms for finding low-rank approximations*, J. ACM, 51 (1998), pp. 1025–1041.

[35] X. Fu, K. Huang, N. D. Sidiropoulos, and W.-K. Ma, *Nonnegative matrix factorization for signal and data analytics: Identifiability, algorithms, and applications*, IEEE Signal Process. Mag., 36 (2019), pp. 59–80.

[36] R. S. Ganti, L. Balzano, and R. Willett, *Matrix completion under monotonic single index models*, in Advances in Neural Information Processing Systems 28, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, eds., Curran Associates, Red Hook, NY, 2015, pp. 1864–1872.

[37] N. Gillis, *Nonnegative Matrix Factorization*, SIAM, Philadelphia, 2021.

[38] G. Golub and W. Kahan, *Calculating the singular values and pseudo-inverse of a matrix*, J. Ser. B, Numer. Anal., 2 (1965), pp. 205–224.

[39] P. Gopalan, L. Charlin, and D. Blei, *Content-based recommendations with Poisson factorization*, in Advances in Neural Information Processing Systems 27, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, eds., Curran Associates, Red Hook, NY, 2014, pp. 3176–3184.

[40] P. Gopalan, J. M. Hofman, and D. M. Blei, *Scalable recommendation with hierarchical poisson factorization*, in Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence (UAI-15), Curran Associates, Red Hook, NY, 2015, pp. 326–335.

[41] G. J. Gordon, *Generalized^2 linear^2 models*, in Advances in Neural Information Processing Systems 15, S. Becker, S. Thrun, and K. Obermayer, eds., MIT Press, Cambridge, MA, 2003, pp. 593–600.

[42] S. Gunasekar, P. Ravikumar, and J. Ghosh, *Exponential family matrix completion under structural constraints*, in Proceedings of the 31st International Conference on Machine Learning (ICML-14), IEEE, Piscataway, NJ, 2014, pp. 1917–1925.

[43] J. Guo, E. Levina, G. Michailidis, and J. Zhu, *Graphical models for ordinal data*, J. Comput. Graph. Statist., 24 (2015), pp. 183–204.

[44] N. Halko, P. G. Martinsson, and J. A. Tropp, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Rev., 53 (2011), pp. 217–288.

[45] P. D. Handschutter, N. Gillis, and X. Siebert, *A survey on deep matrix factorizations*, Comput. Sci. Rev., 42, (2021), 100423.

[46] T. Hastie, R. Mazumder, J. D. Lee, and R. Zadeh, *Matrix completion and low-rank SVD via fast alternating least squares*, J. Mach. Learn. Res. (AMLR), 16 (2015), pp. 3367–3402.

[47] N. C. Henderson and R. Varadhan, *Damped Anderson acceleration with restarts and monotonicity control for accelerating EM and EM-like algorithms*, J. Comput. Graph. Statist., 28 (2019), pp. 834–846.

[48] J. M. Hernandez-Lobato, N. Houlsby, and Z. Ghahramani, *Stochastic inference for scalable probabilistic modeling of binary matrices*, in Proceedings of the 31st International Conference on Machine Learning (ICML-14), IEEE, Piscataway, NJ, 2014, pp. 379–387.

[49] G. E. Hinton and Z. Ghahramani, *Generative models for discovering sparse distributed representations*, Philos. Trans. Roy. Soc. B, 352 (1997), pp. 1177–1190.

[50] G. E. Hinton and R. R. Salakhutdinov, *Reducing the dimensionality of data with neural networks*, Science, 313 (2006), pp. 504–507.

[51] P. D. Hoff, *Bilinear mixed-effects models for dyadic data*, J. Amer. Statist. Assoc., 100 (2005), pp. 286–295.

[52] P. D. Hoff, *Modeling homophily and stochastic equivalence in symmetric relational data*, in Advances in Neural Information Processing Systems 20, J. Platt, D. Koller, Y. Singer, and S. Roweis, eds., Curran Associates, Red Hook, NY, 2008, pp. 657–665.

[53] M. Hoffman, D. Blei, J. Paisley, and C. Wang, *Stochastic variational inference*, J. Mach. Learn. Res. (JMLR), 14 (2013), pp. 1303–1347.

[54] T. Hofmann, *Probabilistic latent semantic analysis*, in Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI-99), Morgan Kaufmann, San Francisco, 1999, pp. 289–296.

[55] C.-J. Hsieh and I. S. Dhillon, *Fast coordinate descent methods with variable selection for non-negative matrix factorization*, in Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACH, New York, 2011, ACM, pp. 1064–1072.

[56] D. J. Hsu, S. M. Kakade, J. Langford, and T. Zhang, *Multi-label prediction via compressed sensing*, in Advances in Neural Information Processing Systems 22, Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, eds., Curran Associates, Red Hook, NY, 2009, pp. 772–780.

[57] P. Jain and P. Kar, *Non-convex optimization for machine learning*, Found. Trends Mach. Learn., 10 (2017), pp. 142–336.

[58] M. Jamshidian and R. I. Jennrich, *Conjugate gradient acceleration of the EM algorithm*, J. Amer. Statist. Assoc., 88 (1993), pp. 221–228.

[59] M. Jamshidian and R. I. Jennrich, *Acceleration of the EM algorithm by using quasi-Newton methods*, J. R. Stat. Soc., Ser. B., 59 (1997), pp. 569–587.

[60] C. C. Johnson, *Logistic matrix factorization for implicit feedback data*, in Neural Information Processing Systems (NIPS-14) Workshop on Distributed Matrix Computations, Curran Associates, Red Hook, NY, 2014.

[61] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, *An introduction to variational methods for graphical models*, Mach. Learn., 37 (1999), pp. 183–233.

[62] R. H. Keshavan, A. Montanari, and S. Oh, *Matrix completion from a few entries*, IEEE Trans. Inform. Theory, 56 (2010), pp. 2980–2998.

[63] J. Kim, Y. He, and H. Park, *Algorithms for nonnegative matrix and tensor factorizations: A unified view based on block coordinate descent framework*, J. Global Optim., 58 (2014), pp. 285–319.

[64] T. G. Kolda and D. P. O'Leary, *A semidiscrete matrix decomposition for latent semantic indexing in information retrieval*, ACM Trans. Inform. Systems, 16 (1998), pp. 322–346.

[65] T. G. Kolda and D. P. O'Leary, *Algorithm 805: Computation and uses of the semidiscrete matrix decomposition*, ACM Trans. Math. Software, 26 (2000), pp. 415–435.

[66] Y. Koren, R. Bell, and C. Volinsky, *Matrix factorization techniques for recommender systems*, Computer, 42 (2009), pp. 30–37.

[67] J. Lafond, *Low rank matrix completion with exponential family noise*, in Proceedings of the 28th Conference on Learning Theory (COLT-15), ACH, New York, 2015, pp. 1224–1243.

[68] J. Lafond, O. Klopp, E. Moulines, and J. Salmon, *Probabilistic low-rank matrix completion on finite alphabets*, in Advances in Neural Information Processing Systems 27, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, eds., Curran Associates, Red Hook, NY, 2014, pp. 1727–1735.

[69] A. S. Lan, C. Studer, and R. Baraniuk, *Matrix recovery from quantized and corrupted measurements*, in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-14), IEEE, Piscataway, NJ, 2014, pp. 4973–4977.

[70] J. S. Larsen and L. K. H. Clemmensen, *Non-negative matrix factorization for binary data*, in Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K-15), Springer, Cham, 2015, pp. 555–563.

[71] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-based learning applied to document recognition*, Proc. IEEE, 86 (1998), pp. 2278–2324.

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

[73] D. D. Lee and H. S. Seung, *Algorithms for non-negative matrix factorization*, in Advances in Neural Information Processing Systems 13, T. Leen, T. Dietterich, and V. Tresp, eds., MIT Press, Cambridge, MA, 2001, pp. 535–541.

[74] D. D. Lee and H. Sompolinsky, *Learning a continuous hidden variable model for binary data*, in Advances in Neural Information Processing Systems 11, M. J. Kearns, S. A. Solla, and D. A. Cohn, eds., MIT Press, Cambridge, MA, 1999, pp. 515–521.

[75] T. Lesieur, F. Krzakala, and L. Zdeborov‡, *MMSE of probabilistic low-rank matrix estimation: Universality with respect to the output channel*, in Proceedings of 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton-15), IEEE, Piscataway, NJ, 2015, pp. 680–687.

[76] C. Liu, D. B. Rubin, and Y. N. Wu, *Parameter expansion to accelerate EM: The PX-EM algorithm*, Biometrika, 85 (1998), pp. 755–770.

[77] A. Lumbreras, L. Filstroff, and C. Févotte, *Bayesian mean-parameterized nonnegative binary matrix factorization*, Data Min. Knowl. Discov., 34 (2020), pp. 1898–1935.

[78] Z. Ma, Z. Ma, and H. Yuan, *Universal latent space model fitting for large networks with edge covariates*, J. Mach. Learn. Res. (JMLR), 21 (2020), pp. 1–67.

[79] J. B. MacQueen, *Some methods for classification and analysis of multivariate observations*, in Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, Berkeley, CA, 1967, pp. 281–297.

[80] M. W. Mahoney, *Randomized algorithms for matrices and data*, Found. Trends Mach. Learn., 3 (2010), pp. 123–224.

[81] I. Markovsky, *Low Rank Approximation: Algorithms, Implementation, Applications*, Comm. Control. Engrg. Ser., Springer, London, 2012.

[82] L. Matthey, I. Higgins, D. Hassabis, and A. Lerchner, *dSprites: Disentanglement Testing Sprites Dataset*, https://github.com/deepmind/dsprites-dataset/, 2017.

[83] A. Mazumdar and A. S. Rawat, *Learning and recovery in the ReLU model*, in Proceedings of the 57th Annual Allerton Conference on Communication, Control, and Computing, IEEE, Piscataway, NJ, 2019, pp. 108–115.

[84] P. McCullagh and J. A. Nelder, *Generalized Linear Models*, Chapman & Hall/CRC, Boca Raton, FL, 1989.

[85] G. J. McLachlan and K. E. Basford, *Mixture Models: Inference and Applications to Clustering*, CRC Press, Boca Raton, FL, 1987.

[86] E. Meeds, Z. Ghahramani, R. M. Neal, and S. T. Roweis, *Modeling dyadic data with binary latent factors*, in Advances in Neural Information Processing Systems 19, B. Schölkopf, J. Platt, and T. Hofmann, eds., MIT Press, Cambridge, MA, 2007, pp. 977–984.

[87] J. J. Meulman, A. J. V. der Kooij, and W. J. Heiser, *Principal components analysis with nonlinear optimal scaling transformations for ordinal and nominal data*, in The Sage Handbook of Quantitative Methodology for the Social Sciences, Sage, Thousand Oaks, CA, 2004, pp. 49–72.

[88] A. Mnih and R. R. Salakhutdinov, *Probabilistic matrix factorization*, in Advances in Neural Information Processing Systems 20, J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, eds., Curran Associates, Red Hook, NY, 2008, pp. 1257–1264.

[89] R. M. Neal, *Probabilistic inference using Markov chain Monte Carlo methods*, Technical report CRG-TR-93-1, Department of Computer Science, University of Toronto, Toronto, 1993.

[90] D. P. O'Leary and S. Peleg, *Digital image compression by outer product expansion*, IEEE Trans. Commun., 31 (1983), pp. 441–444.

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

[92] 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.

[93] J. Paisley, D. Blei, and M. Jordan, *Bayesian nonnegative matrix factorization with stochastic variational inference*, in Handbook of Mixed Membership Models and Their Applications, E. Airoldi, D. Blei, E. Erosheva, and S. Fienberg, eds., Chapman and Hall/CRC Handb. Mod. Stat. Methods, Chapman and Hall/CRC, Boca Raton, FL, 2014.

[94] B. Ren, L. Pueyo, G. Zhu, and B. Duchêne, *Non-negative matrix factorization: Robust extraction of extended structures*, Astrophys. J., 852 (2018), p. 104.

[95] L. Rencker, F. Bach, W. Wang, and M. D. Plumbley, *Sparse recovery and dictionary learning from nonlinear compressive measurements*, IEEE Trans. Signal Process., 67 (2019), pp. 5659–5670.

[96] J. D. M. Rennie and N. Srebro, *Fast maximum margin matrix factorization for collaborative prediction*, in Proceedings of the 22nd International Conference on Machine Learning, ACH, New York, 2005, pp. 713–719.

[97] J. D. M. Rennie and N. Srebro, *Loss functions for preference levels: regression with discrete ordered labels*, in Proceedings of the IJCAI Multidisciplinary Workshop on Advances in Preference Handling, Informs, Catonsville, MD, 2005, pp. 180–186.

[98] D. B. Rubin and D. T. Thayer, *EM algorithms for ML factor analysis*, Psychometrika, 47 (1982), pp. 69–76.

[99] R. R. Salakhutdinov, S. T. Roweis, and Z. Ghahramani, *Optimization with EM and expectation-conjugate-gradient*, in Proceedings of the 20th International Conference on Machine Learning (ICML-03), ACM, New York, 2003, pp. 672–679.

[100] L. Saul and F. Pereira, *Aggregate and mixed-order Markov models for statistical language processing*, in Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing (EMNLP-97), Association for Computational Linguistics, Somerset, NY, 1997, pp. 81–89.

[101] A. I. Schein, L. K. Saul, and L. H. Ungar, *A generalized linear model for principal component analysis of binary data*, Proc. Mach. Learn. Res. (PMLR), RY (2003), pp. 240–247.

[102] H. S. Seung and D. D. Lee, *The manifold ways of perception*, Science, 290 (2000), pp. 2268–2269.

[103] A. P. Singh and G. J. Gordon, *A unified view of matrix factorization models*, in Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD-08), Springer, Berlin, 2008, pp. 358–373.

[104] G.-J. Song and M. K. Ng, *Nonnegative low rank matrix approximation for nonnegative matrices*, App. Math. Lett., 105 (2020), 106300.

[105] H. A. Song and S. Lee, *Hierarchical representation using NMF*, in Proceedings of the International Conference on Neural Information Processing (ICONIP-13), Springer, Berlin, 2013, pp. 466–473.

[106] A. Soni, S. Jain, J. Haupt, and S. Gonella, *Noisy matrix completion under sparse factor models*, IEEE Trans. Inform. Theory, 62 (2016), pp. 3636–3661.

[107] N. Srebro, *Learning with Matrix Factorizations*, PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2004.

[108] N. Srebro and T. Jaakkola, *Weighted low-rank approximations*, in Proceedings of the 20th International Conference on Machine Learning (ICML-03), ACM, New York, 2003, pp. 720–727.

[109] N. Srebro, J. Rennie, and T. S. Jaakkola, *Maximum-margin matrix factorization*, in Advances in Neural Information Processing Systems 17, L. K. Saul, Y. Weiss, and L. Bottou, eds., MIT Press, Cambridge, MA, 2005, pp. 1329–1336.

[110] J. Sun, S. Boyd, L. Xiao, and P. Diaconis, *The fastest mixing Markov process on a graph, and a connection to a maximum variance unfolding problem*, SIAM Rev., 48 (2006), pp. 681–699.

[111] L. Taslaman and B. Nilsson, *A framework for regularized non-negative matrix factorization, with application to the analysis of gene expression data*, PLOS One, 7 (2012), e46331.

[112] M. E. Tipping, *Probabilistic visualisation of high-dimensional binary data*, in Advances in Neural Information Processing Systems 11, M. J. Kearns, S. A. Solla, and D. A. Cohn, eds., MIT Press, Cambridge, MA, 1989, pp. 592–598.

[113] M. E. Tipping and C. M. Bishop, *Probabilistic principal component analysis*, J. R. Stat. Soc. Ser. B Stat. Methodol., 61 (1999), pp. 611–622.

[114] A. M. Tomé, R. Schachtner, V. Vigneron, C. G. Puntonet, and E. W. Lang, *A logistic nonnegative matrix factorization approach to binary data sets*, Multidimens. Syst. Signal Process., 26 (2013), pp. 125–143.

[115] G. Trigeorgis, K. Bousmalis, S. Zafeiriou, and B. Schuller, *A deep matrix factorization method for learning attribute representations*, IEEE Trans. Pattern Anal. Mach. Intell., 39 (2016), pp. 417–429.

[116] J. A. Tropp, A. Yurtsever, M. Udell, and V. Cevher, *Practical sketching algorithms for low-rank matrix approximation*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 1454–1485.

[117] J. A. Tropp, A. Yurtsever, M. Udell, and V. Cevher, *Streaming low-rank matrix approximation with an application to scientific simulation*, SIAM J. Sci. Comput., 41 (2019), pp. A2430–A2463.

[118] M. Turk and A. Pentland, *Eigenfaces for recognition*, J. Cogn. Neurosci., 3 (1991), pp. 71–86.

[119] M. Udell, C. Horn, R. Zadeh, and S. Boyd, *Generalized low rank models*, Found. Trends Mach. Learn., 9 (2016), pp. 1–118.

[120] K. Q. Weinberger and L. K. Saul, *Unsupervised learning of image manifolds by semidefinite programming*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR-04), IEEE Computer Society, Los Alamitos, CA, 2004, pp. 998–995.

[121] K. Q. Weinberger, F. Sha, and L. K. Saul, *Learning a kernel matrix for nonlinear dimensionality reduction*, in Proceedings of the 21st International Conference on Machine Learning (ICML-04), ACM, New York, 2004, pp. 839–846.

[122] D. M. Witten, R. Tibshirani, and T. Hastie, *A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis*, Biostatistics, 10 (2009), pp. 515–534.

[123] J. Wright and Y. Ma, *High-Dimensional Data Analysis with Low-Dimensional Models: Principles, Computation, and Applications*, Cambridge University Press, Cambridge, 2021.

[124] Y.-J. Wu, E. Levina, and J. Zhu, *Generalized Linear Models with Low Rank Effects for Network Data*, 2017, arXiv preprint, arXiv:1705.06672.

[125] J. Xu, D. Hsu, and A. Maleki, *Benefits of over-parameterization with EM*, in Advances in Neural Information Processing Systems 31, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds., Curran Associates, Red Hook, NY, 2018, pp. 10685–10695.

[126] H. J. Xue, X. Dai, J. Zhang, S. Huang, and J. Chen, *Deep matrix factorization models for recommender systems*, in Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI-17), International Joint Conferences on Artificial Intelligence Organization, 2017, pp. 3203–3209.

[127] J. Yu, G. Zhou, A. Cichocki, and S. Xie, *Learning the hierarchical parts of objects by deep nonsmooth nonnegative matrix factorization*, IEEE Access, 6 (2018), pp. 58096–58105.

[128] Y. Yu, *Monotonically overrelaxed EM algorithms*, J. Comput. Graph. Statist., 21 (2012), pp. 518–537.

[129] H. Zhao, Z. Ding, and Y. Fu, *Multi-view clustering via deep matrix factorization*, in Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17), AAAI Press, Palo Alto, CA, 2017, pp. 2921–2927.

[130] Y. Zhao and M. Udell, *Matrix completion with quantified uncertainty through low rank Gaussian copula*, in Advances in Neural Information Processing Systems 33, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, eds., Curran Associates, Red Hook, NY, 2020, pp. 20977–20988.

[131] Y. Zhao and M. Udell, *Missing value imputation for mixed data via Gaussian copula*, in Proceedings of the 26th International Conference on Knowledge Discovery and Data Mining (KDD-20), ACM, New York, 2020, pp. 636–646.