Learning Non-Negative Sparse Image Codes by Convex Programming

Matthias Heiler and Christoph Schnörr

University of Mannheim
Dept. Mathematics & Computer Science
D-68131 Mannheim, Germany
{heiler, schnoerr}@uni-mannheim.de

Abstract

Example-based learning of codes that statistically encode general image classes is of vital importance for computational vision. Recently, non-negative matrix factorization (NMF) was suggested to provide image codes that are both sparse and localized, in contrast to established non-local methods like PCA. In this paper we adopt and generalize this approach to develop a novel learning framework that allows to efficiently compute sparsity-controlled invariant image codes by a well-defined sequence of convex conic programs. Applying the corresponding parameter-free algorithm to various image classes results in semantically relevant and transformation-invariant image representations that are remarkably robust against noise and quantization.

1. Introduction and Related Work

Originally proposed to model physical and chemical processes [20, 19], *non-negative matrix factorization (NMF)* has become increasingly popular in machine learning, signal processing, and computer vision applications [25, 9, 21]. One reason for this popularity is that NMF codes naturally favor sparse, parts-based representations [11, 2] which in the context of recognition can be more robust than non-sparse, global features. Especially for computer vision applications, researchers suggested various extensions of NMF in order to enforce very localized representations [12, 7, 23]. Along this line, it was recently proposed [8] to extend NMF by adding explicit sparsity constraints.

In a different direction, to compute representations that are robust against geometric transformations, it was proposed to integrate transformation-invariance into mixture models, treating transformations as hidden variables in an EM-framework [4]. We show that NMF also benefits from such treatment, leading to sparse and descriptive image bases.

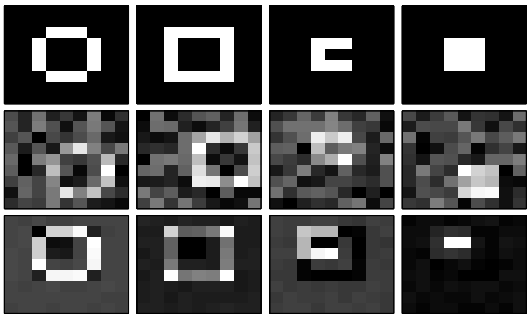


Figure 1: **Translation-invariant NMF.** Results of translation-invariant NMF on the artificial data set from [4]. The first row shows the image primitives used. Shifted and noise corrupted versions of these were used for training (second row). NMF successfully constructs a suitable image base (third row). Note that the basis functions are *not* simply centroids in image space: they represent *parts* which must be *composed* to form the images. Thus, they are potentially more general without sacrificing image quality/sharpness.

Our contributions in this paper are threefold:

- We generalize the NMF optimization problems to support supervised recognition, general constraints to control sparsity, and translation-invariant representations.
- We mathematically formulate the generalized NMF problem in a reverse-convex programming framework in terms of second-order convex cones. We then propose a novel parameter-free algorithm to efficiently compute a solution by solving a sequence of convex optimization problems.
- In the experimental section, we provide evidence that NMF coding can lead to semantically relevant local image bases that are robust against disturbances in the scene and against quantization errors in coefficient space.

In Bayesian terms the NMF problem we examine corresponds to a Gaussian data likelihood $p(V \mid W, H)$ with a noninformative prior over the non-negative orthant for the parameters W and H. Taking a statistical prior into account, this leads to a Bayesian approach to ICA and corresponding inference algorithms [14, 6].

Rather than pursuing this research direction, we focus here on the additional constraints for the NMF problem enforcing sparse and parts-based representations for computer vision, and on a corresponding novel optimization approach based on convex programming (Sec. 3). We will show, however, how available prior information, compatible with our optimization approach, can be considered (eqn. (7)).

Outline. In Section 2, we formally state the NMF optimization problems and give a brief link to signal approximation. Section 3 introduces a new optimization algorithm for sparse NMF. Experimental results are discussed in Section 4. We conclude in Section 5.

Notation. For any $m \times n$-matrix M, we denote its i-th column by M_{*i}, and its row by M_{i*}. $V \in \mathbb{R}_+^{m \times n}$ is a non-negative matrix containing n images, and $W \in \mathbb{R}_+^{m \times r}$ is a corresponding basis with r-dimensional coefficients $H \in \mathbb{R}_+^{r \times n}$. $\|x\|_p$ denotes the ℓ_p-norm for vectors x, and $\|M\|_F$ the Frobenius norm for matrices: $\|M\|_F = \sqrt{\mathrm{tr}(M^\top M)}$. Finally, $\otimes$ denotes Kronecker's matrix product and $\mathrm{vec}(M)$ the concatenation of the columns of M.

2. Variations of NMF

We consider the NMF optimization problem:

$$\begin{aligned} \min_{W,H} \quad & \|V - WH\|_F^2 \\ \text{s.t.} \quad & 0 \leq W, H. \end{aligned} \tag{1}$$

When the images in V are subject to transformations that are to be factored out, the problem reads:

$$\begin{aligned} \min_{W,H,\theta} \quad & \|T_\theta(V) - WH\|_F^2 \\ \text{s.t.} \quad & 0 \leq W, H. \end{aligned} \tag{2}$$

where T is an operator, parametrized by θ, mapping V to transformed images.

Although NMF codes often are sparse without any interventions, it has been suggested to control sparsity directly. This can lead to considerably improved basis functions (Fig. 2). Thus, Hoyer [8] recently proposed to use the following sparseness measure for vectors $x \in \mathbb{R}^n$, $x \neq 0$:

$$\mathrm{sp}(x) := \frac{1}{\sqrt{n} - 1} \left(\sqrt{n} - \frac{\|x\|_1}{\|x\|_2} \right). \tag{3}$$

Since $\frac{1}{\sqrt{n}} \|x\|_1 \leq \|x\|_2 \leq \|x\|_1$, this sparseness measure is bounded: $0 \leq \mathrm{sp}(x) \leq 1$. In particular, $\mathrm{sp}(x) = 0$ for minimally sparse vectors with equal non-zero components, and $\mathrm{sp}(x) = 1$ for maximally sparse vectors with all but one vanishing components. By a slight abuse of notation, we also write $\mathrm{sp}(M) \in \mathbb{R}^n$, meaning $\mathrm{sp}(\cdot)$ is applied to each column of matrix $M \in \mathbb{R}^{m \times n}$.

Note that $\mathrm{sp}(x)$ is invariant against permutation of the components of x and satisfies

$$(x_i - x_j) \left(\frac{\partial}{\partial x_i} \mathrm{sp}(x) - \frac{\partial}{\partial x_j} \mathrm{sp}(x) \right) \geq 0 \,, \quad x \in \mathbb{R}_+^n \,. \tag{4}$$

Thus, it is Schur-convex on $\mathbb{R}_+^n$, that is $-\mathrm{sp}(\cdot)$ is *Schur-concave* [15]. As such, it is well-suited as criterion for *best sparse basis selection* [17, 10].

Consequently, Hoyer [8] proposed to use $\mathrm{sp}(x) = const.$ to constrain the set of admissible solutions of (1). Slightly generalizing this approach to obtain a feasible set with a non-void interior, we formulate:

$$\begin{aligned} \min_{W,H} \quad & \|V - WH\|_F^2 \\ \text{s.t.} \quad & 0 \leq W, H \\ & s_w^{\min} \leq \mathrm{sp}(W) \leq s_w^{\max} \\ & s_h^{\min} \leq \mathrm{sp}(H^\top) \leq s_h^{\max} \end{aligned} \tag{5}$$

where $s_w^{\min}, s_h^{\min}, s_w^{\max}, s_h^{\max}$ are user parameters to control sparsity.

Alternatively, it can be convenient to trade sparsity for reconstruction accuracy by relaxing the hard constraints:

$$\begin{aligned} \min_{W,H} \quad & \|V - WH\|_F^2 - \lambda_h e^\top \mathrm{sp}(H^\top) - \lambda_w e^\top \mathrm{sp}(W) \\ \text{s.t.} \quad & 0 \leq W, H. \end{aligned} \tag{6}$$

Furthermore, for object recognition it is generally useful to integrate available information about object labels into the process of learning image bases [23]. We will see that with our approach it is particularly efficient to restrict, for each class i and for each of its vectors j, the coefficients H_{*j} to a cone around the class center μ_i

$$\begin{aligned} \min_{W,H} \quad & \|V - WH\|_F^2 \\ \text{s.t.} \quad & 0 \leq W, H & \text{(7a)} \\ & \|\mu_i - H_{*j}\|_2 \leq \lambda \|\mu_i\|_1 \quad \forall i, \forall j \in \mathrm{class}(i). & \text{(7b)} \end{aligned}$$

Given class label information, the μ_i are completely determined by the coefficient matrix H. Thus, they are computed implicitly by the optimization algorithm.

3. Reverse-Convex Optimization

In this Section we describe algorithms for solving the optimization problems stated above. They are meant to complement methods based on projected gradient descent that can be slow or unstable.

3.1. Second Order Cone Programming

The computational framework we will be working in is that of *second order cone programming* [13]. It is concerned with minimizing a linear objective function, subject to the constraints that several affine functions of the variables are required to lie in a *second order cone* $\mathcal{L}^{n+1} \subset \mathbb{R}^{n+1}$, i.e., in the convex set

$$\mathcal{L}^{n+1} = \left\{ \begin{pmatrix} x \\ t \end{pmatrix} = (x_1, \ldots, x_n, t)^\top \;\middle|\; \|x\|_2 \leq t \right\}. \quad (8)$$

With this notation, the general form of a *second order cone program* (SOCP) is given by

$$\begin{aligned} \inf_{x \in \mathbb{R}^n} \quad & f^\top x \\ \text{s.t.} \quad & \begin{pmatrix} A_i x + b_i \\ c_i^\top x + d_i \end{pmatrix} \in \mathcal{L}^{n+1}, \qquad i = 1, \ldots, m. \end{aligned} \quad (9)$$

SOCPs are convex programs for which efficient, large scale solvers are available.

3.2. SOCP and Sparseness

Importantly, the sparseness measure (3) and second order cones are closely related: on the non-negative orthant the vectors no sparser than s are exactly those within the second order cone

$$C(s) = \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{pmatrix} x \\ (\sqrt{n} - (\sqrt{n} - 1)s)^{-1} e^\top x \end{pmatrix} \in \mathcal{L}^{n+1} \right\}. \quad (10)$$

Thus, when only max-sparsity constraints are given, $s_h^{\min} = s_w^{\min} = 0$, program (5) can be written[1] as

$$\begin{aligned} \min_{W,H,t} \quad & t \\ \text{s.t.} \quad & \begin{pmatrix} \mathrm{vec}(V^\top) - (W \otimes I)\mathrm{vec}(H^\top) \\ t \end{pmatrix} \in \mathcal{L}^{rn+1} \\ & W \in \mathbb{R}_+^{m \times r} \cap \mathcal{C}_w(s_w^{\max}) \\ & H \in \mathbb{R}_+^{r \times n} \cap \mathcal{C}_h(s_h^{\max}). \end{aligned} \quad (11)$$

This is a *biconvex* program that can be solved by a sequence of SOCPs that alternately minimize for H and for W. Thus, NMF with max-sparsity constraints is in principle no more difficult than ordinary NMF [19].

The presence of min-sparsity constraints, however, complicates matters: they lead to *reverse-convex* constraints

$$\begin{aligned} \min_{W,H,t} \quad & t \\ \text{s.t.} \quad & \begin{pmatrix} \mathrm{vec}(V^\top) - (W \otimes I)\mathrm{vec}(H^\top) \\ t \end{pmatrix} \in \mathcal{L}^{rn+1} \\ & W \in (\mathbb{R}_+^{m \times r} \cap \mathcal{C}_w(s_w^{\max})) \setminus \mathcal{C}_w(s_w^{\min}) \\ & H \in (\mathbb{R}_+^{r \times n} \cap \mathcal{C}_h(s_h^{\max})) \setminus \mathcal{C}_h(s_h^{\min}). \end{aligned} \quad (12)$$

[1] We abbreviate $\mathcal{C}_w(s) = \{W \in \mathbb{R}^{m \times r} \mid W_{*i} \in C(s), \forall i\}$ and $\mathcal{C}_h(s) = \{H \in \mathbb{R}^{r \times n} \mid H_{i*} \in C(s), \forall i\}$.

Even when optimizing alternately for H and W the individual steps are no longer convex. In this sense, NMF with min-sparsity constraints is more difficult than ordinary NMF.

3.3. The RC-Algorithm

In order to solve (12) we start with an arbitrary non-negative initialization and alternately optimize for W and for H, while keeping the other constant. Since both optimizations are symmetric, we focus our presentation on the H step only.

Our algorithm for optimizing H is motivated by results from global optimization [22] and consists of two complementary steps: one maximizes sparsity subject to the constraint that the objective value must not increase. Dually, the other optimizes the objective function $f(H) = \|V - WH\|_F^2$ under the condition that the min-sparsity constraint may not be violated.

Let $H^0 \in \partial\mathcal{C}_h(s_h^{\min})$ be an initialization on the boundary of the min-sparsity cone. It may be computed by solving (11) for H without min-sparsity constraints (i.e., a SOCP) and projecting the solution onto $\partial\mathcal{C}_h(s_h^{\min})$. Akin to logarithmic penalty functions [24] the projection can be efficiently implemented by multiplication in the log-domain. I.e., each element x_i in x is exponentiated and replaced by $c \cdot x_i^\alpha$, with $\alpha \geq 1$ chosen appropriately. The factor $c = c(x, \alpha)$ ensures that the ℓ_2-norm of x is not affected by this transformation.

After initialization we set $k \leftarrow 0$ and, in the first step, consider the program

$$\begin{aligned} \max_{H} \quad & g(H) \equiv \min_j \{\mathrm{sp}(H_{j*})\} \\ \text{s.t.} \quad & H \in \mathbb{R}_+^{r \times n} \cap \mathcal{C}_h(s_h^{\max}) \\ & f(H) \leq f(H^k) \end{aligned} \quad (13)$$

that maximizes sparsity of the least sparse H_{j*} subject to the constraint that the solution may not measure worse than H^k in terms of the target function f. This is a *convex maximization* problem on a bounded domain. As such, it can in principle be solved to global optimality [22]. However, practical algorithms exist for small-scale problems only.

Thus we will content ourselves with a local improvement that is obtained by replacing $\mathrm{sp}(x)$ by its first order Taylor expansion at H^k, resulting in the SOCP

$$\begin{aligned} \max_{H,t} \quad & t \\ \text{s.t.} \quad & H \in \mathbb{R}_+^{r \times n} \cap \mathcal{C}_h(s_h^{\max}) \\ & \begin{pmatrix} \mathrm{vec}(V^\top) - (W \otimes I)\mathrm{vec}(H^\top) \\ f(H^k) \end{pmatrix} \in \mathcal{L}^{rn+1} \\ & t \leq \mathrm{sp}(H_{j*}^k) + \langle \nabla_{H_{j*}} \mathrm{sp}(H_{j*}^k)^\top, H_{j*} - H_{j*}^k \rangle \quad \forall j. \end{aligned} \quad (14)$$

Let H^{sp} denote the corresponding solution. Note that H^k is a feasible point of (14) and the sparsity cone is convex. Thus, optimization will in fact yield $g(H^{\text{sp}}) \geq g(H^k)$.

In the second step we solve the SOCP

$$\begin{aligned} \min_{H,t} \quad & t \\ \text{s.t.} \quad & \begin{pmatrix} \text{vec}(V^\top) - (W \otimes I)\text{vec}(H^\top) \\ t \end{pmatrix} \in \mathcal{L}^{rn+1} \quad (15) \\ & \begin{pmatrix} H_{j*} - H^{\text{sp}}_{j*} \\ \min_{q \in \mathcal{C}(s_h^{\min})} \|q - H^{\text{sp}}_{j*}\|_2 \end{pmatrix} \in \mathcal{L}^{n+1} \quad \forall j \\ & H \in \mathbb{R}_+^{r \times n} \cap \mathcal{C}_h(s_h^{\max}). \end{aligned}$$

This problem is identical to (12) restricted to H, except for the reverse-convex min-sparsity constraint that is replaced by a convex proximity constraint: each H_{j*} is restricted to lie within a "save" distance from H^{sp}_{j*}.

In summary, our algorithm for optimizing H consists of the following steps:

1.)	$H^0 \leftarrow$ solution of (11) projected on $\partial\mathcal{C}(s_h^{\min})$, $k \leftarrow 0$
2.)	repeat
3.)	$H^{\text{sp}} \leftarrow$ solution of (14)
4.)	$H^{k+1} \leftarrow$ solution of (15)
5.)	$k \leftarrow k + 1$
6.)	until $\|f(H^k) - f(H^{k-1})\| \leq \epsilon$

Akin to [22], convergence to a local optimum can be proven under mild restrictions (notably, that W is not degenerate). Proof and remarks regarding alternative optimization schemes will be presented in an extended paper.

3.4. Relaxed Formulation

The relaxed form of sparsity-controlled NMF described in (6) is optimized similarly by linearizing $\text{sp}(x)$ around H^k, yielding the SOCP

$$\begin{aligned} \min_{H,t,s} \quad & t - \lambda_h s \\ \text{s.t.} \quad & \begin{pmatrix} \text{vec}(V^\top) - (W \otimes I)\text{vec}(H^\top) \\ t \end{pmatrix} \in \mathcal{L}^{rn+1} \quad (16) \\ & s \leq \text{sp}(H^k_{j*}) + \langle \nabla_{H_{j*}} \text{sp}(H^k_{j*})^\top, H_{j*} - H^k_{j*} \rangle \quad \forall j \\ & H \in \mathbb{R}_+^{r \times n}. \end{aligned}$$

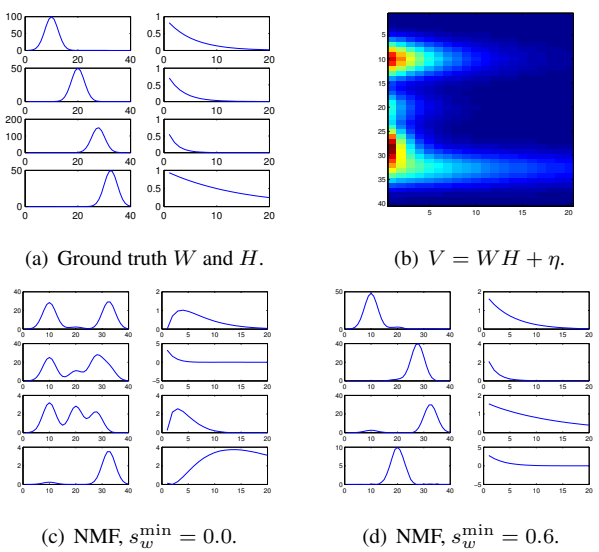
Thus, in order to solve (6) for H we iteratively solve instances of (16) until convergence.

3.5. Exploiting Information from Class Labels

The supervised variant (7) of the NMF problem is readily solved by the RC algorithm since (7b) translates, for each class i and for each coefficient vector H_{*j} belonging to class i, into a second order constraint

$$\begin{pmatrix} 1/n_i H_{(i)} e - H_{j*} \\ \lambda / n_i e^\top H_{(i)} e \end{pmatrix} \in \mathcal{L}^{n+1}, \quad \forall i, \forall j \in \text{class}(i). \quad (17)$$

Here, the $r \times n_i$-matrix of coefficients belonging to class i is abbreviated $H_{(i)}$ and we recognize $\mu_i = 1/n_i H_{(i)} e$. Adding these constraints to (14) and (15) yields an algorithm for solving supervised NMF.



(a) Ground truth W and H. (b) $V = WH + \eta$. (c) NMF, $s_w^{\min} = 0.0$. (d) NMF, $s_w^{\min} = 0.6$.

Figure 2: **Paatero experiments.** The data set is displayed in Fig. 2(a) and 2(b): Gaussian and exponential distributions are multiplied to yield matrix V. In the experiments, a small amount of Gaussian noise $\eta \sim \mathcal{N}(0, 0.1)$ is added to the product. The results for different values of the min-sparsity constraint are shown in Fig. 2(c) and 2(d): Only an active sparsity constraint makes recovery of W and H successful.

3.6. Dealing with Image Transformations

We assume a finite set of linear transformations mapping the input data $V \in \mathbb{R}^{m \times n}$ into $T_\theta(V) \in \mathbb{R}^{m \times n}$. θ_i specifies the transformations active for image $i \in \{1, \ldots, m\}$.

After each iteration we greedily replace the image data V by its most probable transformation, i.e., setting $V \leftarrow T_{\theta^*}(V)$ with

$$\theta^* = \arg\min_\theta \|T_\theta(V) - W^k H^k\|_F^2. \quad (18)$$

As long as the identity is part of the possible transformations this operation never increases the objective value to be minimized. It also does not affect convergence. However, for large images and many possible transformations it can be a very slow operation to compute. In this case, variational techniques and FFT offer greatly improved performance [3].

Table 1: **Performance comparison.** We run the reverse-convex algorithm (rca) and the projected gradient descent (pgd) to compute sparse decompositions of the digit data set. Runtime (sec.) and residual error $\|V - WH\|_F^2$ are reported. rca operates faster than pgd while achieving similar accuracy.

sparsity	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
time rca	23.97	25.10	31.80	53.76	58.14	56.27	47.07	49.07	37.55
time pgd	286.61	961.21	433.06	53.92	87.02	408.22	73.49	228.61	1223.56
quotient	11.96	38.29	13.62	1.00	1.50	7.26	1.56	4.66	32.59
error rca	0.82	0.76	0.73	0.73	0.78	0.91	0.99	1.07	1.12
error pgd	0.85	0.79	0.74	0.72	0.77	0.88	0.99	1.07	1.12
quotient	1.04	1.03	1.02	0.98	0.99	0.97	0.99	1.00	1.00

4. Experiments

In this Section we show that sparseness-controlled, transformation-invariant NMF bases are useful for computer vision and compare the RC algorithm to projected gradient descent.

4.1. Analyzing Synthetic Data

To examine the performance of the sparsity-controlled NMF algorithm we repeated an experiment suggested by Paatero [18]: He considered a synthetic dataset consisting of products of Gaussian and exponential distributions designed to resemble data from spectroscopic experiments in chemistry and physics (Fig. 2(a)–(b)). This data set is *not* easily analyzed: without prior knowledge, NMF is reported to fail to recover the original factors in the dataset (Fig. 2(c)). As a remedy, Paatero implemented a "target shape" extension to NMF. Fig. 2(d) shows that with a min-sparsity constraint on W our algorithm finds the correct factorization.

4.2. Comparison with Established Algorithms

To see how the reverse convex algorithm (rca) compares against an established method we computed sparsity-controlled decompositions into $r = 4$ basis functions for a subset of the USPS handwritten digits data set using rca and projected gradient descent (pgd) as proposed in [8]. For different choices of sparseness we report runtime and residual error in Tab. 1. Note that the stopping criterion used was different for rca and for pgd: rca stopped when after a full iteration the objective value did not improve at least by a constant, the pgd implementation used[2] stopped as soon as the norm of the gradient was smaller than some ϵ. As the error measurements shown in Tab. 1 demonstrate, both stopping criteria yield comparable results. Regarding running time we see that rca was faster in most cases and showed much less variation between individual runs.

[2]We used the pgd code kindly provided by the author of [8], and removed all logging and monitoring parts to speed up calculation. Our SOCP solver was Mosek 3.2 from MOSEK ApS, Denmark, running under Linux.

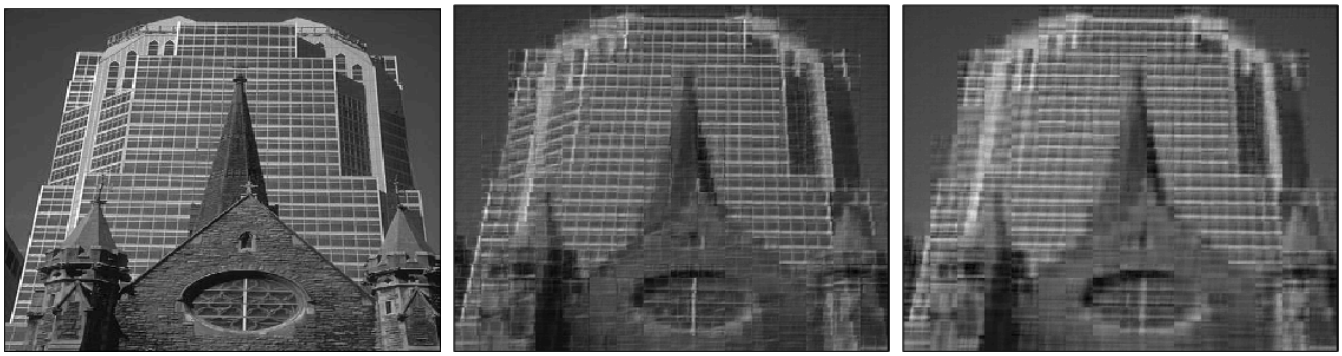
4.3. Transformation-Invariant Image Bases

In Fig. 1 we show the results of transformation-invariant NMF (TNMF) applied to the artificial data described in [4]: four image primitives are translated using circular shifts in both image dimensions and Gaussian noise is added. The resulting training data set contained 1000 randomly translated images. For these, we learned image based using a feathering mask to encourage centered bases[3]. The resulting image basis not only models the data well, it also has a nice complementary structure: even without additional sparseness constraints it tends to avoid modeling the same image location multiple times [11], leading to a true parts-based representation. A possible explanation for this behavior is that the parts-based representation offers more degrees of freedom, making a better fit to the data in the presence noise.

In a more realistic scenario we learned image bases for a skyscraper image (Fig. 3): while NMF correctly captures the dominant horizontal and vertical lines, it is forced to model the same structure multiple times to fit the data well. TNMF removes this burden, allowing for much finer detail to appear in the basis functions. In fact, looking closely one can recognize parts of the building and key architectural structures being modeled.

We also used the PCA and the TNMF image bases for reconstruction: the original image was divided into 20×20 patches and for each patch we determined the optimal translation w.r.t. the given image bases. The patches were then reconstructed and assembled to form images 3(b) and 3(c). As expected, translation-invariance ensures that the TNMF reconstruction looks notably sharper. It is also closer to the original image: the Frobenius norm of the residual image was about 7% larger for the PCA reconstruction than for the TNMF reconstruction.

[3]I.e., each basis function was weighted with a Gaussian s.t. pixels near the boundary had slightly less influence than pixels near the center.



(a) Original (b) TNMF reconstruction (c) PCA reconstruction

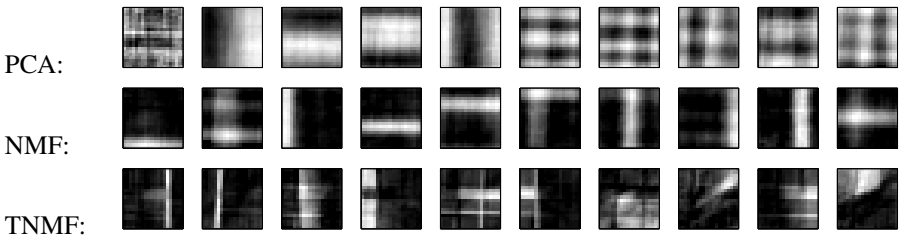


Figure 3: **Image modeling.** Different image bases learned for image 3(a) are shown. PCA learns global properties of image variation. The individual base images carry no apparent semantic meaning. NMF learns sparse, localized image features, but represents the dominant image elements (horizontal and vertical bars) multiple times. Transformation-invariant NMF (TNMF) is less redundant and captures very detailed image structure which can sometime be recognized as parts from the building. In Fig. 3(b) and 3(c) reconstructions for TNMF and PCA are displayed. The TNMF reconstruction appears sharper and is slightly more accurate (see text).

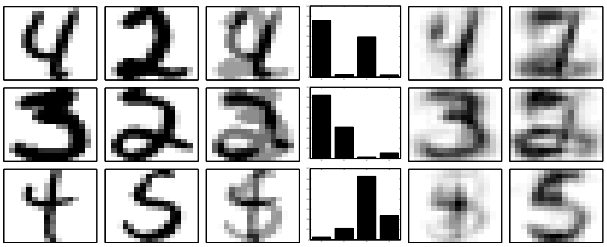


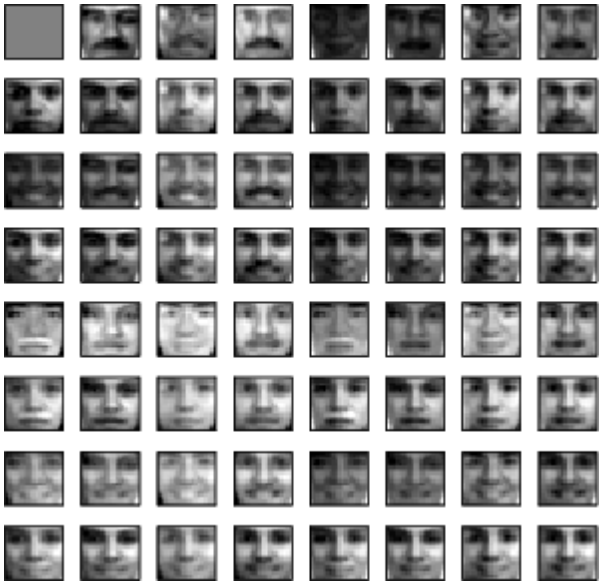
Figure 4: **Recognition and segmentation.** Based on sparse NMF, a model for the digits 2 – 5 (first columns) is built and presented with superimposed digits (third column) *after* training. The model consistently assigns the highest probabilities (fourth column) to the digits forming the image. This shows that NMF models can be relatively *stable against disturbances*. A subsequent local optimization retrieves the original digits (last columns).

4.4. Recognition with Conditional MaxEnt

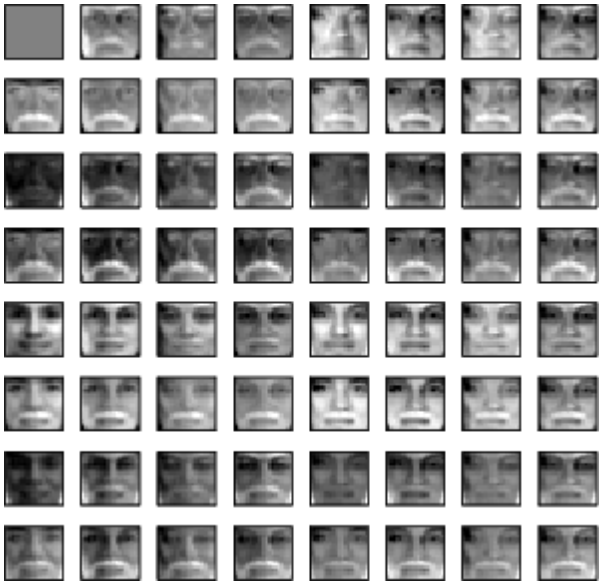
It has previously been reported that localized NMF is relatively robust against occlusions and disturbances [12]. To verify this claim for sparse NMF we repeated an image recognition experiment, previously approached with *credibility networks* [5]: a model for the individual digits 2,3,4,5 from the USPS digit database was built, then new, more complicated, images were constructed by superimposing images from two different digits (Fig. 4). Our model consisted of sparse NMF codes ($r = 20$, $s_w^{\min} = 0.6$) for the single digits and a *conditional maximum entropy* (cMaxEnt) classifier $p(y|h)$ for class labels y and given coefficients h, using mean coefficient values and distance to randomly chosen reference coefficients as features [16]. For each combined image we computed h and evaluated $p(y|h)$ for $y \in \{2, 3, 4, 5\}$. On a test data set we counted how often the two top-ranking digits, w.r.t. $p(y|h)$, were the correct digits composing the image.

Five self-selected human subjects (students) achieved classification rates between 65% and 80% (mean 75%) on our data. The NMF-cMaxEnt classifier described above scored 77% correct on 500 test samples. For the more complex credibility networks a recognition rate of 78.3% is reported [5] on a test set of 120 binarized images.

Once a decision is made for two digits y_1, y_2, we can visualize the corresponding segmentation by solving the non-



(a) NMF: $W * \delta(h > \epsilon)$



(b) PCA: $W * \delta(|h| > \epsilon)$

Figure 5: **Robustness against quantization.** The 64 faces corresponding to previously learned 6bit NMF/PCA image codes after quantization of the coefficients. The top left image corresponds to the binary coefficient vector $\#b000000$, the bottom right image to $\#b111111$. The NMF faces suffer less from quantization than their PCA counterparts.

convex program

$$\begin{aligned} \max_{h_1,h_2} \quad & p(y_1|h_1)p(h_1) \cdot p(y_2|h_2)p(h_2) \\ \text{s.t.} \quad & h = h_1 + h_2 \\ & 0 \leq h_1, h_2 \end{aligned} \tag{19}$$

i.e., we factor the reconstruction coefficient h into h_1 and h_2 such that the probability of the detected digits is maximized. Depending on the features used for training cMaxEnt and the form of the prior densities $p(h)$ this can be a very difficult problem. It turns out that for our choice of features and a Parzen estimator for $p(h)$ a conjugate gradient search already yields meaningful reconstructions (Fig. 4).

4.5. Modeling a Low-Entropy Image Class

Human faces, aligned, cropped and evenly lit, lead to highly structured images with relatively low entropy. With such images, sparse NMF appears robust against quantization: we learned a sparse image code ($r = 6, s_h^{\min} = 0.3$) for face images [1] and a PCA code for comparison. Then we enumerated possible reconstructions by setting each entry of the coefficient vector to 0 or to 1. The resulting $2^6 = 64$ images are shown in Fig. 5: while most NMF "reconstructions" look remarkably natural the corresponding PCA images mostly suffer severe degradation.

4.6. Supervised Training

To show that the supervised label constraints (7b) can be useful we trained NMF codes ($r = 4$) on only 100 samples from the handwritten digit data described above. We used different values for the parameter λ and a very simple conditional maximum entropy model $p(y|h)$ with mean coefficient values $\mathbb{E}[h_i]$ as only features for classification. The number of errors on a 300 sample test dataset is given below:

λ	1e4	1e2	1	1e-2	1e-4	1e-6
#errors	108	82	75	60	58	56

When λ is large, i.e., the supervised label constraint is inactive, the error is about 36% (108 out of 300 samples). This is slightly worse than a corresponding PCA basis (95 errors) would achieve. As the label constraint is strengthened the classification performance improves and finally is almost twice as good as in the unsupervised case.

5. Summary and Conclusions

We introduced a fast and reliable optimization scheme for sparse non-negative matrix factorization. It treats sparsity-constrained NMF as a reverse convex programming problem that is solved by a series of second order cone programs. Unlike optimization schemes relying on the gradient descent idea there are no parameters to be chosen, thus

overcoming the traditional trade-off between speed and reliability w.r.t. a learning rate.

By adding or removing individual constraints we exercise very precise control over sparseness and generalize the original formulation of sparse NNF [8] along multiple directions. For instance, we added supervised constraints keeping coefficient vectors belonging to the same object class within a cone around their mean. In an experiment this constraint already doubled the recognition rate compared to classical NMF or PCA. In a different direction, we relaxed the strict sparseness constraints by penalizing non-sparseness in the objective function similar to [17]. With NMF, this again leads to a sequence of convex programs.

Last but not least, the stable performance of the convex programs allows for intervening optimization of image transformations: by factoring out operations like translation, rotation or scaling very compact, qualitatively new image representations emerge that robustly encode image classes by local parts.

Future work concerns integration of additional a priori knowledge in form of new constraints on the feasible set. Also, the encouraging results w.r.t. quantization robustness lead us to investigate discrete probability models for well-defined image classes.

References

[1] CBCL. CBCL face database #1. MIT Center For Biological and Computational Learning, http://cbcl.mit.edu/software-datasets, 2000.

[2] D. Donoho and V. Stodden. When does non-negative matrix factorization give a correct decomposition into parts? In *Adv. in NIPS*, volume 17, 2004.

[3] B. J. Frey and N. Jojic. Fast, large-scale transformation-invariant clustering. In *Adv. in NIPS*, 2001.

[4] B. J. Frey and N. Jojic. Transformation-invariant clustering using the EM algorithm. *IEEE Trans. Patt. Anal. Mach. Intell.*, 45(1):1–17, Jan. 2003.

[5] G. E. Hinton, Z. Ghahramani, and Y. W. Teh. Learning to parse images. In *Adv. in NIPS*, pages 463–469, 2000.

[6] P. Højen-Sørensen, O. Winther, and L. Hansen. Mean field approaches to independent component analysis. *Neural Computation*, 14:889–918, 2002.

[7] P. O. Hoyer. Non-negative sparse coding. *Proc. of the IEEE Works. on Neur. Netw. for Sig. Proc.)*, pages 557–565, 2002.

[8] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *J. of Mach. Learning Res.*, 5:1457–1469, 2004.

[9] P. O. Hoyer and A. Hyvärinen. A multi-layer sparse coding network learns contour coding from natural images. *Vision Research*, 42(12):1593–1605, 2002.

[10] K. Kreutz-Delgado and B. Rao. A general approach to sparse basis selection: Majorization, concavity, and affine scaling. Technical report, ECE Dept., UCSD, 1997.

[11] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(21):788–791, Oct. 1999.

[12] S. Z. Li, X. W. Hou, H. J. Zhang, and Q. S. Cheng. Learning spatially localized, parts-based representation. In *Proc. of CVPR*, 2001.

[13] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra and its Applications*, 1998.

[14] D. J. MacKay. Maximum likelihood and covariant algorithms for independent component analysis. University of Cambridge, Cavendish Lab, 1996. http://www.inference.phy.cam.ac.uk/mackay/BayesICA.html.

[15] A. Marshall and I. Olkin. *Inequalities : Theory of Majorization and Its Applications*. Acad. Press, 1979.

[16] K. Nigam, J. Lafferty, and A. McCallum. Using maximum entropy for text classification. In *IJCAI-99 Workshop on Machine Learning for Information Filtering, pages 61–67, 1999.*, 1999.

[17] B. A. Olshausen and D. J. Field. Natural image statistics and efficient coding. *Network: Computation in Neural Systems*, 7:333–339, May 1996.

[18] P. Paatero. Least squares formulation of robust non-negative factor analysis. *Chemometrics and Intelligent Laboratory Systems*, 37, 1997.

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

[20] J. Shen and G. W. Israël. A receptor model using a specific non-negative transformation technique for ambient aerosol. *Atmospheric Environment*, 23(10):2289–2298, 1989.

[21] P. Smaragdis and J. C. Brown. Non-negative matrix factorization for polyphonic music transcription. In *IEEE Workshop on Appl. of Sign. Proc. to Audio and Acoustics*, pages 177–180, 2003.

[22] H. Tuy. Convex programs with an additional reverse convex constraint. *J. of Optim. Theory and Applic.*, 52(3):463–486, Mar. 1987.

[23] Y. Wang, Y. Jia, C. Hu, and M. Turk. Fisher non-negative matrix factorization for learning local features. In *Proc. Asian Conf. on Comp. Vision*, 2004.

[24] S. J. Wright. *Primal-dual interior-point methods*. SIAM, Society for Applied Mathematics, 3600 University City Science Center, Philadelphia, PA 19104-2688, 1997.

[25] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *SIGIR '03: Proc. of the 26th Ann. Intl. ACM SIGIR Conf. on Res. and Developm. in Info. Retrieval*, pages 267–273. ACM Press, 2003.