Honda Research Institute EU
**Honda Research Institute Europe GmbH**
https://www.honda-ri.de/

# Ternary Sparse Coding

## Georgios Exarchakis, Marc Henniges, Julian Eggert, Jörg Lücke

### 2012

**Preprint:**

This is an accepted article published in LVA/ICA 2012 Proceedings. The final authenticated version is available online at: https://doi.org/[DOI not available]

# Ternary Sparse Coding

Georgios Exarchakis[1], Marc Henniges[1], Julian Eggert[2], and Jörg Lücke[1]

[1] FIAS, Goethe-Universität Frankfurt am Main, Germany
[2] Honda Research Institute Europe, Offenbach am Main, Germany

**Abstract.** We study a novel sparse coding model with discrete and symmetric prior distribution. Instead of using continuous latent variables distributed according to heavy tail distributions, the latent variables of our approach are discrete. In contrast to approaches using binary latents, we use latents with three states (-1, 0, and 1) following a symmetric and zero-mean distribution. While using discrete latents, the model thus maintains important properties of standard sparse coding models and of its recent variants. To efficiently train the parameters of our probabilistic generative model, we apply a truncated variational EM approach (Expectation Truncation). The resulting learning algorithm infers all model parameters including the variance of data noise and data sparsity. In numerical experiments on artificial data, we show that the algorithm efficiently recovers the generating parameters, and we find that the applied variational approach helps in avoiding local optima. Using experiments on natural image patches, we demonstrate large-scale applicability of the approach and study the obtained Gabor-like basis functions.

## 1 Introduction

Since it has first been introduced, Sparse Coding (SC) [1] has become a standard model to explain the behavior of simple cells in the primary visual cortex. However, the derivation of sparse coding algorithms typically involves analytical intractabilities, e.g., because involved posterior distributions and their expectation values have no closed-form solutions. The standard approach to overcome this difficulty is the use of MAP estimates for inference and learning (e.g.,[1,2] but also compare [3] or reviews on variational methods). Using binary hidden variables, this analytical intractability can be avoided altogether because learning rules using Expectation Maximization (EM) are closed-form. However, in contrast, e.g., to a Laplace prior, the Bernoulli distribution used for binary variables is not symmetric and not zero-mean (see Fig. 1). These differences to conventional priors can have implications for the used preprocessing and the inferred basis functions, which may add to effects caused by the introduction of discrete hidden variables. To study the implications of discrete hidden variables independently of differences in prior symmetries, we investigate in this work a generative model with a symmetric and discrete prior distribution. Such a prior can be well suited for different types of data, and it directly relates to recent Sparse Coding versions with hard-sparseness constraints (compare, e.g., [4,5]).

F. Theis et al. (Eds.): LVA/ICA 2012, LNCS 7191, pp. 204–212, 2012.
© Springer-Verlag Berlin Heidelberg 2012

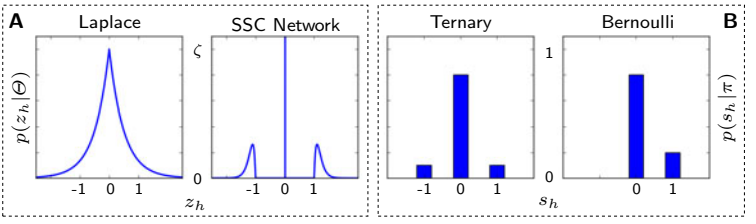


**Fig. 1.** Comparison of different prior distributions. **A** shows distributions for a continuous variable $z_h$ (Laplace as used in [1] and the SSC network prior [4]) with model parameters $\Theta$. In **B** the symmetrical Ternary Sparse Coding prior of this paper and the unsymmetrical Bernoulli prior (e.g., [13,6]) are displayed for the discrete variable $s_h$ with model parameter $\pi$.

## 2 Sparse Coding with Ternary Hidden Variables

Let $\{\boldsymbol{y}^{(n)}\}_{n=1,\ldots,N}$ be a set of $N$ independent data points, $\boldsymbol{y}^{(n)} \in \mathbb{R}^D$, i.e., the number of observed variables is $D$. We want to find the parameters $\Theta = (W, \sigma, \pi)$ that maximize the data likelihood $\mathcal{L} = \prod_{n=1}^{N} p(\boldsymbol{y}^{(n)} \,|\, \Theta)$, under the generative model:

$$p(\boldsymbol{s}|\pi) \;=\; \prod_{h=1}^{H} \left(\tfrac{\pi}{2}\right)^{|s_h|} \left(1-\pi\right)^{1-|s_h|}, \qquad p(\boldsymbol{y} \,|\, \boldsymbol{s}\,,\, W, \sigma) \;=\; \mathcal{N}(\boldsymbol{y};\, W\boldsymbol{s}\,, \sigma^2 \mathbb{1}), \quad (1)$$

where $W \in \mathbb{R}^{D\times H}$ denotes the $H$ basis vectors for the data points, and $s_h \in \{-1, 0, 1\}$, with $h = 1, \ldots, H$, is the latent variable that specifies how $W_h$ contributes to a data point. That is, we can assume that each observed variable can take the form $y_d = \sum_{h=1}^{H} W_{dh} s_h$ plus a Gaussian noise with variance $\sigma^2$. The $\pi$ parameter in the prior is the probability of a cause being active. Since our prior is symmetrical, the probability of a cause being added to generate a data point is equal to the probability of a cause being subtracted: $p(s_h = -1) = p(s_h = 1) = \frac{\pi}{2}$. The difference between previous SC generative models and this model is the choice of a symmetrical and discrete prior for latents taking values $-1$, 0, and 1. As a consequence, Ternary Sparse Coding (TSC) can explain data generated by Binary Sparse Coding (BSC,[13,6]) while the opposite does not apply (see Fig. 2). To optimize the parameters $\Theta$, we apply with Expectation Truncation (ET) a truncated variational EM approach [7]. Instead of optimizing the log-likelihood directly, variational EM optimizes the free-energy, a lower-bound which depends on the parameters and an approximate posterior distribution $q$:

$$\mathcal{F}(q, \Theta) = \sum_{n=1}^{N} \left[ \sum_{\boldsymbol{s}} q^{(n)}(\boldsymbol{s}\,; \Theta^{\text{old}}) \, \left[ \log\left(p(\boldsymbol{y}^{(n)} \,|\, \boldsymbol{s}\,, W, \sigma)\right) + \log\left(p(\boldsymbol{s} \,|\, \pi)\right) \right] \right] + H(q)\,. \quad (2)$$

Instead of conventional variational EM, ET does not assume a factored form of $q^{(n)}$ but uses truncated sums to set $q^{(n)}$ proportional to the exact posterior on

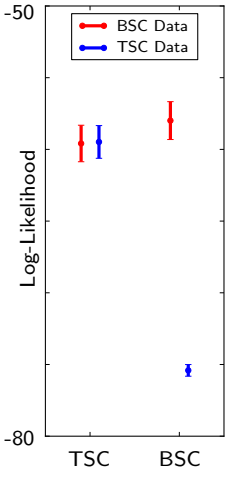


**Fig. 2.** Comparison of model performance on different data types. We created 100 sets of $N = 1000$ data points with the BSC model and the TSC model, then ran both algorithms on all data sets and compared the likelihood values. The figure shows the mean Log-likelihood values with standard deviation for the TSC model and the BSC model both on TSC data (blue, right) and BSC data (red, left). Evidently, TSC explains both data types equally well, whereas BSC describes data points created with TSC worse than its own data points. While the latter is an expected result for a model mismatch, the equal performance of TSC on both types of data points out the more general character of TSC. More particularly, BSC can be understood as a special case of TSC.

subsets $\mathcal{K}_n$ (see [7] for details). Consequently, the expectation values w.r.t. $q^{(n)}$ amount to:

$$\langle g(\boldsymbol{s}) \rangle_{q^{(n)}} = \frac{\sum_{\boldsymbol{s}} p(\boldsymbol{s}, \boldsymbol{y}^{(n)} \,|\, \Theta^{\text{old}})\; g(\boldsymbol{s})}{\sum_{\tilde{\boldsymbol{s}}} p(\tilde{\boldsymbol{s}}, \boldsymbol{y}^{(n)} \,|\, \Theta^{\text{old}})} \approx \frac{\sum_{\boldsymbol{s} \in \mathcal{K}_n} p(\boldsymbol{s}, \boldsymbol{y}^{(n)} \,|\, \Theta^{\text{old}})\; g(\boldsymbol{s})}{\sum_{\tilde{\boldsymbol{s}} \in \mathcal{K}_n} p(\tilde{\boldsymbol{s}}, \boldsymbol{y}^{(n)} \,|\, \Theta^{\text{old}})}, \quad (3)$$

where $g(\boldsymbol{s})$ is a function of $\boldsymbol{s}$ (and potentially the parameters). Eqn. 3 represents a good approximation if the set $\mathcal{K}_n$ contains most of the posterior probability mass w.r.t. a given data point $\boldsymbol{y}^{(n)}$. For our model, $\mathcal{K}_n$ in (3) is chosen to contain hidden states $\boldsymbol{s}$ with at most $\gamma$ active causes, i.e., $\|\boldsymbol{s}\|_1 \leq \gamma$. Furthermore, we only consider the combinatorics of $H' \geq \gamma$ hidden variables that are likely to have contributed to generating a given data point $\boldsymbol{y}^{(n)}$. More formally we define:

$$\mathcal{K}_n = \{\boldsymbol{s} \mid \|\boldsymbol{s}\|_1 \leq \gamma \text{ and } \forall i \notin I : s_i = 0\}, \quad (4)$$

where the index set $I$ contains those latent indices $h$ with the $H'$ largest values of a *selection function* $\mathcal{S}_h(\boldsymbol{y}^{(n)})$. In our case this function is given by:

$$\mathcal{S}_h(\boldsymbol{y}) = \max\left\{p(s_h = -1, \boldsymbol{y} \,|\, \Theta^{\text{old}}), p(s_h = 1, \boldsymbol{y} \,|\, \Theta^{\text{old}})\right\} \;\; \forall \boldsymbol{s} \; : \|\boldsymbol{s}\|_1 = 1. \quad (5)$$

A large value of $\mathcal{S}_h(\boldsymbol{y})$ signals a high likelihood that $\boldsymbol{y}$ contains the basis function $\boldsymbol{W}_h$ as a component. In numerical experiments on ground-truth data we can verify that for most data points the approach (3) with (4) and (5) indeed approximates the true expectation values with high accuracy. By applying the ET approximation, exact EM (which scales exponentially with H) is altered to an algorithm which scales polynomially with $H'$ (approximately $\mathcal{O}\left(H'^{\gamma}\right)$) and linearly with $H$. Note, however, that in general larger values of $H$ also require larger amounts of data points.

With the tractable approximations for the expectation values $\langle g(\boldsymbol{s})\rangle_{q^{(n)}}$ computed with (3) to (5) the update equations for $W$ and $\sigma$ are given by:

$$W^{\text{new}} = \left(\sum_{n\in\mathcal{M}} \boldsymbol{y}^{(n)} \langle \boldsymbol{s} \rangle_{q_n}^T\right) \left(\sum_{n'\in\mathcal{M}} \langle \boldsymbol{s}\ \boldsymbol{s}^T\rangle_{q_{n'}}\right)^{-1} \tag{6}$$

$$\sigma^{\text{new}} = \sqrt{\frac{1}{|\mathcal{M}|\,D} \sum_{n\in\mathcal{M}} \left\langle \left|\left|\boldsymbol{y}^{(n)} - W\,\boldsymbol{s}\right|\right|^2 \right\rangle_{q_n}} \tag{7}$$

Note that we do not sum over all data points $\boldsymbol{y}^{(n)}$ but only over those in a subset $\mathcal{M}$ (note that $|\mathcal{M}|$ is the number of elements in $\mathcal{M}$). The subset contains those data points for which (3) finally represents a good approximation. It is defined to contain the $N^{\text{cut}}$ data points with the highest values for $\sum_{\widetilde{\boldsymbol{s}}\in\mathcal{K}_n} p(\,\widetilde{\boldsymbol{s}}\,,\boldsymbol{y}^{(n)}\,|\,\Theta^{\text{old}})$, i.e., with the highest values for the denominator in (3). $N^{\text{cut}}$ is hereby the expected number of data points that have been generated by states with less or equal $\gamma$ non-zero entries: $N^{\text{cut}} = N \sum_{\boldsymbol{s}\,,\,\|\boldsymbol{s}\|_1\leq\gamma} p(\boldsymbol{s}\,|\,\pi) = N \sum_{\gamma_1=0}^{\gamma}\sum_{\gamma_2=0}^{\gamma-\gamma_1} \binom{H}{\gamma_1\gamma_2(H-\gamma_1-\gamma_2)} \left(\frac{\pi}{2}\right)^{\gamma_1+\gamma_2} (1-\pi)^{H-\gamma_1-\gamma_2}$.

Update equations (6) and (7) were obtained by setting the derivatives of Eqn. 2 (w.r.t. $W$ and $\sigma$) to zero. Similarly, we can derive the update equation for $\pi$. However, as the approximation only considers states $\boldsymbol{s}$ with a maximum of $\gamma$ non-zero entries, the update has to correct for an underestimation of $\pi$. If such a correction is taken into account (compare [6,14]), we obtain the update rule:

$$\pi^{\text{new}} = \frac{A(\pi)\,\pi}{B(\pi)}\,\frac{1}{|\mathcal{M}|}\sum_{n\in\mathcal{M}} \langle\|\boldsymbol{s}\|_1\rangle_{q_n} \quad \text{with} \tag{8}$$

$$A(\pi) = \sum_{\gamma_1=0}^{\gamma}\sum_{\gamma_2=0}^{\gamma-\gamma_1} \binom{H}{\gamma_1\gamma_2\,(H-\gamma_1-\gamma_2)} \left(\frac{\pi}{2}\right)^{\gamma_1+\gamma_2} (1-\pi)^{H-\gamma_1-\gamma_2} \quad \text{and}$$

$$B(\pi) = \sum_{\gamma_1=0}^{\gamma}\sum_{\gamma_2=0}^{\gamma-\gamma_1} (\gamma_1+\gamma_2) \binom{H}{\gamma_1\gamma_2\,(H-\gamma_1-\gamma_2)} \left(\frac{\pi}{2}\right)^{\gamma_1+\gamma_2} (1-\pi)^{H-\gamma_1-\gamma_2}$$

Note that if we allow all possible states (i.e., $\gamma = H$), the correction factor $\frac{A(\pi)\,\pi}{B(\pi)}$ in (8) is equal to $\frac{1}{H}$ and the set $\mathcal{M}$ becomes equal to the set of all data points (because $N^{\text{cut}} = N$). Eqn. 8 then falls back to the exact EM update rule for $\pi$. The same applies for Eqns. 6 and 7 for $\gamma = H$. By choosing a value for $\gamma$ between one and $H$ we can thus choose the accuracy of the used approximation. The higher the value of $\gamma$ the more accurate is the approximation but the larger are also the computational costs. For intermediate values of $\gamma$ we can obtain very good approximations with small computational costs.

## 3 Numerical Experiments

**Linear Bars Test.** In order to evaluate its performance, we applied the algorithm to artificial data points as shown in Fig. 3A. These data were generated

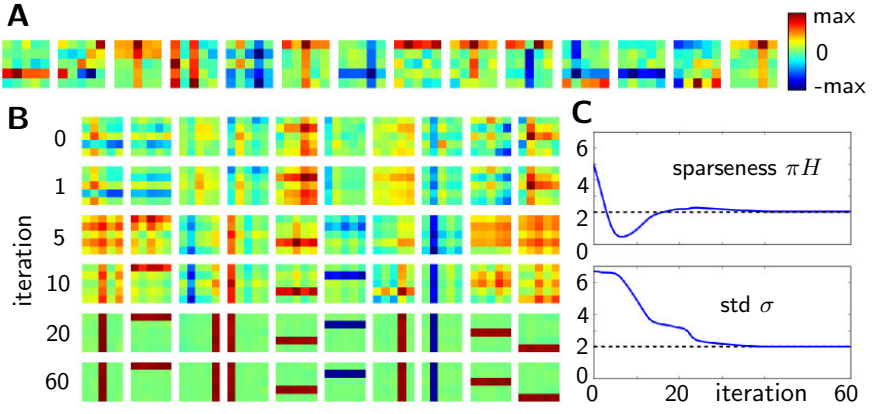


**Fig. 3.** Linear bars test with $H = 10$, $D = 5 \times 5$, and $N = 1000$. **A** 14 example data points. **B** Basis functions for iterations given on the left. **C** Sparseness and standard deviation plotted over the iterations. Ground-truth indicated by dashed horizontal line.

by $H = 10$ basis functions $\boldsymbol{W}_h$ as described by the generative model in (1). The prior parameter was set to $\pi = 0.2$ which results in $\pi H = 2$ basis functions contributing to one data point on average. Each basis function represents one $D = 25$ pixel image forming a vertical or a horizontal bar on a $5 \times 5$ grid. In particular, $W_h^d \in \{0, 10\}$ where the value 10 denotes a bar pixel and 0 represents a background pixel. To these data we added iid Gaussian noise (mean = 0, std = 2). ET approximation parameters described in (4) were set to $H' = 5$ and $\gamma = 3$. An annealing temperature (see [8,6]) was kept constant at $T = 2$ for the first 10 iterations, and then decreased linearly to $T = 1$ at iteration 40, where it was kept until termination of the algorithm at iteration 60. For the first 20 iterations we set the amount of used data points to the number of all data points, i.e. $|\mathcal{M}| = N$, then linearly decreased it to $N^{\text{cut}}$ until iteration 40, where we kept it until the end (compare [7]). We set the initial values for each basis function to be the average over the data points plus a Gaussian white noise with standard deviation 0.05. Sparseness was initialized at $\pi H = 5$, thus assuming that five of the causes contributed to an image on average. The standard deviation was initialized as the average variance of all pixels which led to a value of $\sigma \approx 6$. After each iteration we added Gaussian noise to the learned basis functions (mean = 0, std = 0.05), which we decreased linearly to zero from iteration 20 to 40. We ran the algorithm with the above parameters 1000 times, each time using a newly generated set of N = 1000 data points. In 913 of these trials we recovered all bars ($\approx 91.3\%$ reliability) and obtained a mean value of $\pi H = 2$ (0.05 std) for the sparseness and $\sigma = 2$ (0.1 std) for the data noise (i.e., the algorithm successfully recovered the generating parameters). Figs. 3B and 3C show the typical development of the model parameters over the 60 iterations.

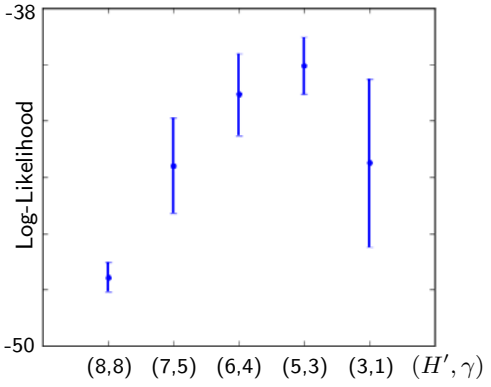


**Fig. 4.** Final log-likelihood values with standard deviation for TSC runs with different sets of ET parameters. The algorithms were applied to linear bars tests with $H = 8$, $\pi = \frac{2}{H}$, $D = 4 \times 4$. For each approximation parameter pair $(H', \gamma)$, we ran the algorithm 100 times on $N = 1000$ newly generated data point for each run. Model parameters were initialized as described in Sec. 2.

**Effect of ET on Local Optima.** To examine the effect of different approximation parameters on the convergence of the algorithm, we applied the TSC algorithms with different sets of values for $H'$ and $\gamma$ to a linear bars test (Fig. 4). As can be observed in Fig. 4, the highest likelihood values were neither obtained for very high values of $H'$ and $\gamma$ nor for very low ones. For low values the approximation gets too coarse and the number of considered data points, $N^{\mathrm{cut}}$, gets too small. For high values the approximation accuracy is high but the algorithm has a strong tendency to converge to local optima. The local optima usually correspond to solutions with, on average, more basis functions per data point than in the generation process. The highest likelihood values are obtained for intermediate values of $H'$ and $\gamma$. For such values the approximation quality is high and, at the same time, local optima are avoided much more frequently. The latter effect can be explained by ET keeping a subset of all possible fixed-points of learning stable. In our case, fixed-points of too dense solutions are avoided.

**Natural Image Patches.** As an example for large-scale applicability, we applied the TSC algorithm to $N = 200\,000$ preprocessed[1] patches of natural images, of size $D = 26 \times 26$, taken from the van Hateren image database [9]. The algorithm assumed $H = 700$ basis functions and an initial value for sparseness $\pi H = 1$. The parameters $W$ and $\sigma$ were initialized as in the linear bars test. The approximation parameters were set to $\gamma = 8$ and $H' = 10$. For the first 20 iterations, the algorithm used all data points for learning, $|\mathcal{M}| = N$, then linearly decreased the number of data points to $|\mathcal{M}| = N^{\mathrm{cut}}$ for the next 20 iterations, and kept it at this value until termination of the algorithm at iteration 200. After each iteration, the basis functions were slightly perturbed using additive Gaussian noise which was linearly decreased from iteration 30 to 48.

Fig. 5A shows a random selection of 200 of the 700 obtained basis functions. In Fig. 5B the most globular of the 700 basis functions are displayed. The monitored time-course of the data sparseness ($\pi H$) and the time-course of the data noise

[1] We used a Difference of Gaussians filter where filter parameters were chosen as in [10]; before the brightest 2% of the pixels were clamped to the maximal value of the remaining 98% (influence of light-reflections were reduced in this way).

($\sigma$) are displayed in Fig. 5C. As can be observed, we obtain Gabor-like basis functions with different orientations, spatial frequencies, and phase as well as globular basis functions with no or very little orientation preferences (compare [11]). Along with the basis functions we obtain an estimate for the noise ($\sigma = 1.8$) and, more importantly, for the data sparseness of $\pi\, H = 6.9$ active causes per $26 \times 26$ patch.

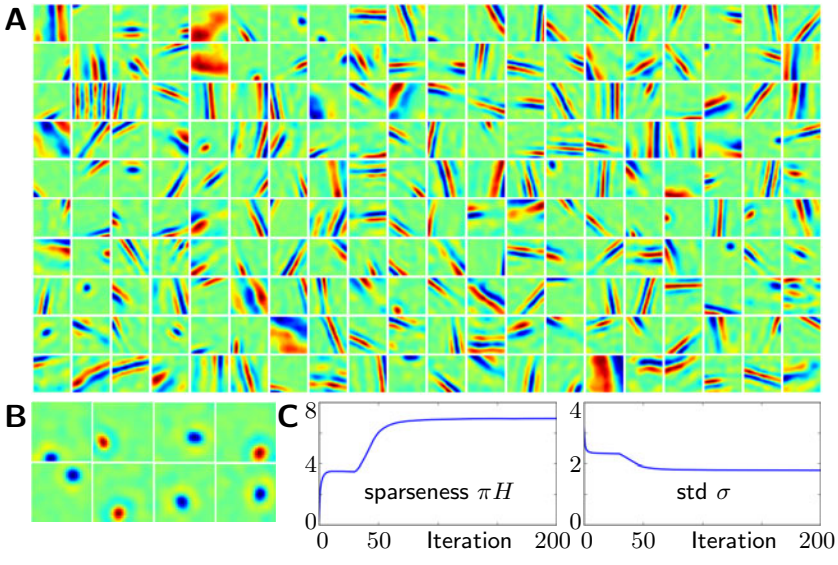


**Fig. 5.** Numerical experiment on image patches. **A** 200 basis functions randomly selected out of the total $H = 700$. **B** Example of eight of the most globular fields. **C** Time-courses of sparseness ($\pi H$) and data noise (standard deviation) $\sigma$.

## 4 Discussion

We have studied a novel sparse coding algorithm using discrete hidden variables. While in most of the former studies on sparse coding the prior distributions are continuous, we, in this work, introduced a symmetric prior defined on the set $\{-1, 0, 1\}$. Such a prior maintains important properties of continuous sparse priors which are, for instance, not preserved for binary hidden variables. As binary variables are a feature of another large class of probabilistic approaches such as deep-belief-networks [12], this work can be regarded as connecting two very successful and active lines of research.

Efficient parameter optimization for our model is made possible by applying Expectation Truncation (ET; [7]), a recent variational EM approach that allowed us to infer all model parameters including data sparsity. In numerical experiments on artificial data, we verified that the resulting algorithm accurately and efficiently recovers the parameters of the generating distribution. Additionally, we found that ET helps in avoiding local optima during learning. While

high approximation parameters $(H', \gamma)$ lead to a very high accuracy of the E-step approximation, learning often converged to local optima corresponding to a solution of relatively low sparsity. Such local solutions were efficiently avoided, however, if the used approximation parameters were lower but still large enough to result in a high approximation quality (Fig. 4). Large-scale applicability of our algorithm was demonstrated using numerical experiments on pre-processed image patches. The obtained basis functions in these experiments are instructive from a neuroscientific perspective. As in previous studies using standard sparse coding ([1] and many others) as well as in studies using binary latents [6,13], we obtained localized Gabor-like basis functions with different orientations and spatial frequencies. Additionally, we obtained globular basis functions as in binary [6] and semi-discrete models [4,5]. Such globular fields are interesting as they have been recorded in neurophysiological experiments but were not obtained with standard SC or ICA before [11]. That globular basis functions also emerge in our study is further evidence for the discreteness of hidden variables facilitating the emergence of globular fields (but also see [14]). As the use of ternary latents is very close to the priors used in [4,5], our approach could be regarded as capturing the essential functional features of these earlier works, and as taking their capabilities a step further by inferring data noise and sparsity.

**Acknowledgment.** The project was funded by the German Research Foundation (DFG), grant LU 1196/4-1 (GE, JL) and by the Honda Research Institute Europe (MH, JE). Furthermore, we gratefully acknowledge support by the Frankfurt Center for Scientific Computing.

## References

1. Olshausen, B.A., Field, D.J.: Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature 381, 607–609 (1996)
2. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online learning for matrix factorization and sparse coding. JMLR 11, 19–60 (2010)
3. Aharon, M., Elad, M., Bruckstein, A.: K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Transactions on Signal Processing 54(11), 4311–4322 (2006)
4. Rehn, M., Sommer, F.T.: A network that uses few active neurones to code visual input predicts the diverse shapes of cortical receptive fields. J. of Comp. Neurosci. 22(2), 135–146 (2007)
5. Rozell, C.J., Johnson, D.H., Baraniuk, R.G., Olshausen, B.A.: Sparse coding via thresholding and local competition in neural circuits. Neural Computation 20(10), 2526–2563 (2008)
6. Henniges, M., Puertas, G., Bornschein, J., Eggert, J., Lücke, J.: Binary Sparse Coding. In: Vigneron, V., Zarzoso, V., Moreau, E., Gribonval, R., Vincent, E. (eds.) LVA/ICA 2010. LNCS, vol. 6365, pp. 450–457. Springer, Heidelberg (2010)
7. Lücke, J., Eggert, J.: Expectation truncation and the benefits of preselection in training generative models. JMLR 11, 2855–2900 (2010)
8. Ueda, N., Nakano, R.: Deterministic annealing EM algorithm. Neural Networks 11(2), 271–282 (1998)

9. Hateren, J.H., Schaaf, A.: Independent component filters of natural images compared with simple cells in primary visual cortex. Proceedings of the Royal Society of London B 265, 359–366 (1998)
10. Lücke, J.: Receptive field self-organization in a model of the fine-structure in V1 cortical columns. Neural Computation 21(10), 2805–2845 (2009)
11. Ringach, D.L.: Spatial structure and symmetry of simple-cell receptive fields in macaque primary visual cortex. Journal of Neurophysiology 88, 455–463 (2002)
12. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Computation 18, 1527–1554 (2006)
13. Haft, M., Hofman, R., Tresp, V.: Generative binary codes. Pattern Anal. Appl. 6, 269–284 (2004)
14. Puertas, J.G., Bornschein, J., Lücke, J.: The maximal causes of natural scenes are edge filters. NIPS 23, 1939–1947 (2010)