KI - Künstliche Intelligenz manuscript No.
(will be inserted by the editor)

Sparse Coding and Selected Applications

Jens Hocke, Kai Labusch, Erhardt Barth, Thomas Martinetz

Received: date / Accepted: date

Abstract Sparse coding has become a widely used framework in signal processing and pattern recognition. After a motivation of the principle of sparse coding we show the relation to Vector Quantization and Neural Gas and describe how this relation can be used to generalize Neural Gas to successfully learn sparse coding dictionaries. We explore applications of sparse coding to image-feature extraction, image reconstruction and deconvolution, and blind source separation.

Keywords Sparse Coding · Neural Gas · K-SVD · image deconvolution · image reconstruction · digit recognition · blind source separation

1 Introduction

1.1 Sparse coding as efficient coding

Early work on sparse coding was based on the efficient-coding hypothesis, which assumes that the goal of visual coding is to faithfully represent the visual input with minimal neural activity. The idea goes back to Barlow [2]. It is based on earlier work of Ernst Mach and Donald MacKay, and has been later extended in several ways [14, 35, 29]. In a statistical information-theoretic framework of efficient coding, one assumes that the efficient code is obtained by reducing the redundancies in the original signal.

Natural images occupy only a small fraction of the entire signal space, i.e. they lie on an extremely compact submanifold within the high-dimensional signal space. Knowledge about this submanifold may be helpful in many ways as it can be used, for example, to find optimal features for classification, and to compress and reconstruct images. Image reconstruction can be considered as a projection of the observed distorted image onto the submanifold of natural images (see Figure 1).

With a sparse representation of the original signal, some of its properties are preserved while others may be lost. Typical criteria for good representations are coding-efficiency, robustness, invariance, but also more goal-oriented criteria like the resulting classification performance. In [35] and related work, a complementary, geometric view on efficient coding has been put forward. These sparsity properties cannot only be observed in images, but also in other natural signals, e.g., acoustic signals [26].

1.2 Learning a sparse code

An important further development was that sparse representations of natural images can be learned and that the resulting representations resemble receptive fields of neurons in the primary visual cortex [29]. Assuming that a signal $\mathbf{x} \in \mathbb{R}^N$ can be represented as $\mathbf{x} = W\mathbf{a} + \epsilon$ in a basis $W \in \mathbb{R}^{N \times M}$ (also called dictionary) with coefficients $\mathbf{a} \in \mathbb{R}^M$ and additive Gaussian white noise $\epsilon \in \mathbb{R}^M$ [29], the signal is sparsely encoded within the dictionary W if most of the elements of $\mathbf{a}$ are zero (or at least small). The sparse coding coefficients can be

Supported by the DFG, grant number MA 2401/2-1.

Jens Hocke, Erhardt Barth, Thomas Martinetz
Insitut für Neuro- und Bioinformatik, Universität Lübeck
Ratzeburger Allee 160, D-23562 Lübeck, Germany
Tel.: +49 451-500-5501, Fax: +49 451-500-5502
E-mail: {hocke,barth,martinetz}@inb.uni-luebeck.de
Kai Labusch
Acrolinx
Rosenstraße 2, D-10178 Berlin, Germany
Tel.: +49 30-288-8483-556, Fax: +49 30-288-8483-39
E-mail: {labusch}@gmail.com

found by solving

$$\mathbf{a} = \arg\min_{\hat{\mathbf{a}}} \left(||W\hat{\mathbf{a}} - \mathbf{x}||^2 + S(\hat{\mathbf{a}}) \right) , \quad (1)$$

where the term $S(\hat{\mathbf{a}})$ enforces sparseness, i.e., coefficient vectors with many small coefficients. For example $S(\hat{\mathbf{a}}) = ||\hat{\mathbf{a}}||_1/\sigma$ or $S(\hat{\mathbf{a}}) = \sum_i \log(1+\hat{a}_i^2)/\sigma$, with $\hat{a}_i$ being the elements of $\hat{\mathbf{a}}$, can be used. The sparse coding dictionary can then be found by solving

$$W = \arg\min_{\hat{W}} \left(\sum_{\mathbf{x}} \min_{\mathbf{a}} \left(||\hat{W}\mathbf{a} - \mathbf{x}||^2 + S(\mathbf{a}) \right) \right) . \quad (2)$$

Another way of enforcing sparseness on the coefficient vector $\mathbf{a}$ is to explicitly limit the permitted number of non-zero coefficients. This will be considered in the remainder of the paper.

Because a maximization of the sparseness is similar to the maximization of the kurtosis (peaky distributions in both cases), sparse coding and Independent Component Analysis (ICA) [7] are related [3] and sometimes deliver similar results that are beyond the capabilities of Principle Component Analysis (PCA).

Another way to find a representation adapted to the data is non-Negative Matrix Factorization (NMF) [30], which decomposes a matrix of signals into a dictionary and a coefficient matrix, both containing only non-negative entries. This approach has been extended with a sparseness constraint and is therefore a special case of sparse coding [17].

In the natural language processing and information retrieval communities latent semantic analysis (LSA) [9], also called latent semantic indexing (LSI), is a well established method. Like PCA it extracts decorrelated, orthogonal components. A probabilistic formulation of LSI termed probabilistic latent semantic indexing (PLSI) has been introduced in [16]. To this PLSI approach, which is equivalent to NMF [10], a sparse prior has been added in [33].

1.3 Sparse coding as vector quantization

Vector quantization methods can be employed in order to learn a compact representation of the submanifold of given sample data in terms of a set of reference vectors $\mathbf{w}_1, \ldots, \mathbf{w}_N$, $\mathbf{w}_i \in \mathbb{R}^N$. Each point on the submanifold is represented by its closest reference vector $\mathbf{w}_i \in \mathbb{R}^N$ (see Figure 1). In this case, a reconstruction of a distorted image is obtained by selecting the reference vector $\mathbf{w}_i$ that is closest to the distorted image.

Sparse coding represents a submanifold by a set of linear subspaces. Each linear subspace of dimension K is defined by a basis $W_i \in \mathbb{R}^{N \times K}$. If we use such a representation of the submanifold, image reconstruction can

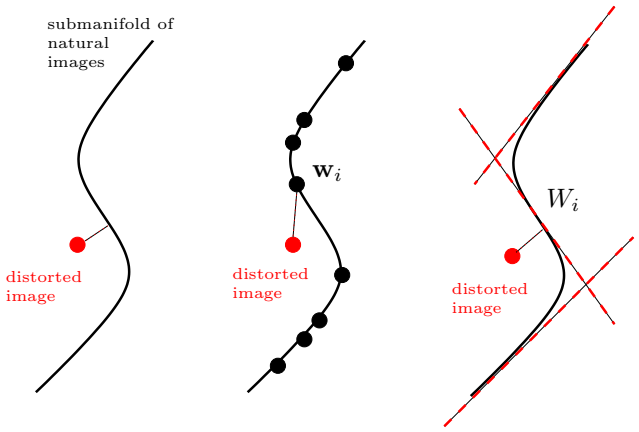


Fig. 1 **Left:** Schematical view of the submanifold of natural images. Reconstruction of an observed distorted image by projection onto this submanifold. **Center:** Representation of the submanifold in terms of a set of codebook vectors as in vector quantization. **Right:** Representation of the submanifold in terms of a set of subspaces as in sparse coding.

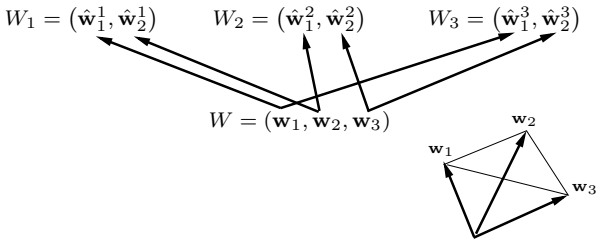


Fig. 2 Compact representation of subspaces by use of a M choose K structure with $K = 2$ and $M = 3$.

be performed by projecting the distorted image onto the closest point on the closest subspace (see Figure 1). Within this framework, vector quantization can be understood as representation in terms of linear subspaces of dimensionality zero.

If we described L linear subspaces of dimension K with individual bases W_i, we would need $L \times N \times K$ parameters. This might be highly redundant and many parameters would have to be adapted. Sparse coding realizes a much more compact approach by employing a M choose K structure. It always selects those K elements from its dictionary of size M, which form the most appropriate linear subspace for a given data point. This enables the representation of $\binom{M}{K}$ linear subspaces that have dimensionality K by means of $M \times N$ dictionary parameters (see Figure 2 for an example).

2 Sparse coding as a two-fold optimization problem

Let us consider given sample data $\mathbf{x}_i \in \mathbb{R}^N$ that stem from an unknown sub-manifold within $\mathbb{R}^N$. The task of

learning a M choose K description of a set of subspaces that covers that unknown submanifold with minimum quadratic error can be formalized by finding the $W \in \mathbb{R}^{N \times M}$ which minimizes

$$\sum_i \left(\min_{\mathbf{a}} \|\mathbf{x}_i - W\mathbf{a}\|_2^2 \text{ subject to } \|\mathbf{a}\|_0 \leq K \right). \quad (3)$$

W is the so-called dictionary that contains the basis vectors of the linear subspaces which have a dimensionality of at most K. The hidden variables $\mathbf{a}$ are the dictionary coefficients ($\|\mathbf{a}\|_0$ is the number of non-zero entries of $\mathbf{a}$). An approximative solution for this difficult optimization task can be found with a nested optimization approach which is described in the following.

Initially, the dictionary is selected randomly from the given samples of the submanifold. Then the adaptation of the dictionary W is obtained by repetitively solving an inner and an outer optimization problem. The inner optimization requires to determine the closest subspace for each data point, i.e., to determine the dictionary coefficients $\mathbf{a}$ by solving

$$\min_{\mathbf{a}} \|\mathbf{x} - W\mathbf{a}\|_2^2 \quad \text{subject to} \quad \|\mathbf{a}\|_0 \leq K. \quad (4)$$

Since this is a NP-hard combinatorial optimization problem [8], one has to use approximation methods such as Orthogonal Matching Pursuit (OMP) [31], Optimized Orthogonal Matching Pursuit (OOMP) [32], or Basis Pursuit (BP) [6], which provide a close-to-optimal solution. If the optimal solution is sparse enough and the dictionary W fulfills some conditions, these algorithms are able to provide the exact optimum of (4) [4].

After having determined the coefficients, these are considered fixed and an optimization step with respect to the dictionary is performed. A number of different approaches for the update of the dictionary have been proposed, e.g., online gradient descent in the Sparsenet algorithm [29], batch gradient descent as used in the Method of Optimal Directions (MOD) [13], its column normalized variant [25], or the K-SVD method [1] which is based on a singular value decomposition of the representation error matrix that is obtained with the current configuration of the coefficients.

A major limitation of all these approaches is that they consider only a single fixed configuration of the coefficients within the update of the dictionary. In many cases not a single configuration of coefficients but a set of close to optimal solutions exists. Hence, it is not clear which configuration should be used in the dictionary optimization. Obviously, it should be advantageous to use all the close to optimal solutions in order to optimize the dictionary. Furthermore, it has been shown recently that by using a plurality of sparse solutions one can obtain better results in some approximation tasks [12].

2.1 Soft-competitive sparse coding

Since sparse coding can be seen as a generalization of vector quantization, the Neural Gas algorithm [28,27] as a very efficient and robust method for finding quantization codebooks should also be suitable for finding dictionaries. The Neural Gas uses not only the optimal but also close to optimal codebooks for learning, which should be advantageous also for learning sparse coding dictionaries. We proposed Sparse Coding Neural Gas (SCNG) [21] and Neural Gas for Dictionary Learning (NGDL) [22] as two possibilities of extending Neural Gas to dictionary learning. Both perform a dictionary update that is based on a plurality of sparse solutions. While the SCNG algorithm is closely connected to the OOMP method and does not allow for selecting an arbitrary method for the determination of the dictionary coefficients, NGDL does not have this limitation. In [34] it is shown how the SCNG algorithm can be extended by a Sobolev-metric for handling also functional data.

In order to perform an NGDL update, one has to determine a set of close to optimal configurations $\mathbf{a}_{j_p}$ of the coefficients. These can be obtained, for instance, using the Bag of Pursuits (BOP) algorithm [22]. It has also been proposed to simply consider a set of sparse configurations of the coefficients that have been obtained from a randomized OMP [12]. The set of configurations is sorted according to the respective representation error

$$\|\mathbf{x} - W\mathbf{a}_{j_0}\| \leq \cdots \leq \|\mathbf{x} - W\mathbf{a}_{j_p}\| \leq \cdots \leq \|\mathbf{x} - W\mathbf{a}_{j_L}\| \quad (5)$$

and a soft-competitive update is applied to the dictionary that is a weighted sum of the update obtained from each single solution:

$$\Delta W = \alpha_t \sum_{p=0}^{L} e^{-\frac{p}{\lambda_t}} (\mathbf{x} - W\mathbf{a}_{j_p}) \mathbf{a}_{j_p}^T. \quad (6)$$

The learning rate α_t and the neighbourhood-size λ_t are decreasing exponentially over time.

We have shown that for $t \to \infty$ this update rule corresponds to a stochastic gradient descent on the target function (3) and that it leads to superior dictionaries compared to a number of methods for dictionary learning that consider only a single configuration of the coefficients in the update of the dictionary [23,22]. Using synthetic data where the actual underlying dictionary is known, we could show that the gain is most significant in the difficult but relevant case of a submanifold that consists of highly overlapping linear subspaces. Furthermore, we could show that the soft-competitive stochastic gradient method enables us to learn the underlying

ground truth even if there are only few samples from the sub-manifold available for learning, in contrast to other methods that do not converge towards the underlying ground truth [22].

3 Applications

3.1 Digit recognition

A crucial step in object recognition is the selection of proper features. In an early approach for feature learning based on sparse coding [18], developed prior to the invention of the soft-competitive methods for dictionary learning, the task-specific dictionary is learned by means of the Sparsenet algorithm. However, the approach does not depend on a particular method for dictionary learning and could also be realized with SCNG or NGDL.

First, patches $P(x, y)$ of size $N \times N$ are extracted from random positions (x, y) of many different images of the training data, in this case the MNIST data set [24]. The patches are then used to learn a sparse-coding dictionary. Figure 3 shows typical samples from the MNIST data set and the corresponding dictionary. Given a new digit that has to be classified, the dictionary is used in order to extract features by sparsely encoding its patches at each position (x, y), providing coefficient vectors $\mathbf{a}(x, y)$. This is done by solving (1) with gradient descent using an appropriate regularization term $S(\mathbf{a})$.

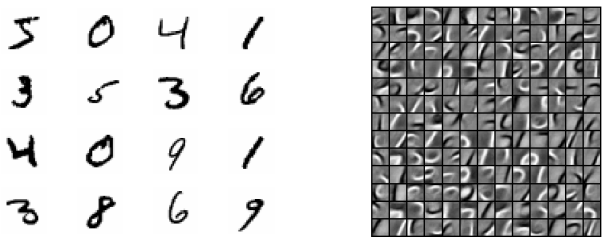
The dictionary elements $\mathbf{w}_j$ can be interpreted as features whereas the coefficients $a_j(x, y)$ indicate how well these features match at location (x, y). In order to reduce the number of features and to make them shift invariant, all the coefficients $a_j(x, y)$ are encoded in a large matrix A_j, which is subdivided into a set of regular, non-overlapping regions $R_i, i = 1, ..., M^2$. As local features, the maximum and minimum of each region with respect to each coefficient matrix are selected:

$$a_j^{max}(R_i) = \max_{x,y \in R_i} a_j(x, y) \ , \tag{7}$$

$$a_j^{min}(R_i) = \min_{x,y \in R_i} a_j(x, y) \ . \tag{8}$$

The final feature vector f of each input image consists of the maximum and minimum values of all regions with respect to all coefficient matrices:

$$\begin{aligned} f = (&a_1^{max}(R_1), \ldots, a_1^{max}(R_{M^2}), \ldots, \\ &a_K^{max}(R_1), \ldots, a_K^{max}(R_{M^2}), \\ &a_1^{min}(R_1), \ldots, a_1^{min}(R_{M^2}), \ldots, \\ &a_K^{min}(R_1), \ldots, a_K^{min}(R_{M^2})) \ . \end{aligned} \tag{9}$$



(a) (b)

Fig. 3 In (a) example digits from the MNIST set are depicted. In (b) the learned features are shown. It can be seen that the features capture significant properties of the digits.

This vector is given as input to a classifier, in this case a set of two-class support vector machines [18]. The algorithm, although being simple, yielded state-of-the-art results.

3.2 Image Reconstruction and Image Deconvolution

Here we consider a digital image $\mathbf{x}_{degr}$ that suffers from degradations, such as missing pixels or blurring due to wrong focus. The degradation can be approximately described as

$$\mathbf{x}_{degr} = A\mathbf{x} + \epsilon, \ A \in \mathbb{R}^{m \times n} \ . \tag{10}$$

A is a transform matrix that either removes pixels or blurres the image. $\mathbf{x}$ is the original image. Filling in the missing pixels or reducing the blurring by deconvolution corresponds to the inversion of the transformation A. Solving for $\mathbf{x}$ is an underdetermined problem, since $m < n$. A common hypothesis is that the image can be sparsely represented in a dictionary, i.e, $\mathbf{x} = W\mathbf{a}$, and the most plausible inversion of (10) is $\mathbf{x} \approx W\hat{\mathbf{a}}$ where

$$\hat{\mathbf{a}} = \arg\min_{\mathbf{a}} \|\mathbf{x}_{degr} - AW\mathbf{a}\|_2 \text{ subject to } \|\mathbf{a}\|_0 \leq k. \tag{11}$$

In a similar way, compressed sensing [5,11] is using sparseness to find a solution to underdetermined systems of equations.

We performed image reconstruction and deconvolution experiments using dictionaries that were learned with NGDL [23,22]. Some results are depicted in Figure 4. We have shown that soft-competitive dictionary learning yields superior dictionaries for image reconstruction tasks. Furthermore, we have shown that the resulting dictionaries adapt to specific classes of images, e.g. images of buildings and flowers.

3.3 Blind source separation

The cocktail party problem is a classical example of blind source separation and many methods have been

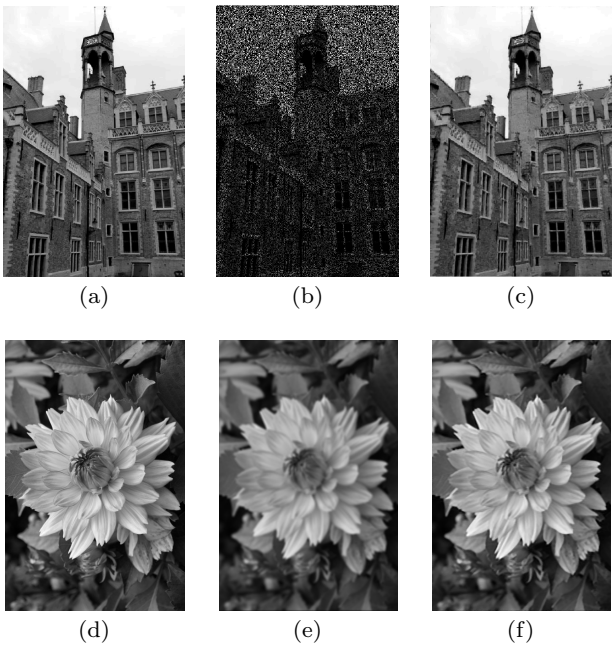


Fig. 4 By using a sparseness prior images are recovered from their degraded versions. In the left column the original images are shown ((a) and (d)). The top middle image (b) was degraded by randomly removing 70 % of the pixels, and the bottom middle image (e) was degraded by blur. The recovered images (c) and (f) are shown on the right.

proposed in order to solve it (see [15] for a review). Labusch et al. [19,20] use SCNG in order to tackle an overcomplete time-invariant and time-dependent variant of the cocktail party problem.

Let us denote the observed audio streams by $\mathbf{x}(t) \in \mathbb{R}^m$, where m is the number of observations, e.g., two for a human listener. The hypothesis is that these streams stem from a mixture $W \in \mathbb{R}^{m \times n}$ of the source audio streams $\mathbf{a}(t) \in \mathbb{R}^n$, where n is the number of sources. In the presence of noise $\epsilon(t)$ we have

$$\mathbf{x}(t) = W\mathbf{a}(t) + \epsilon(t), \tag{12}$$

which is the signal model that has been introduced in Section 1.2. The dictionary in (12) corresponds to the unknown mixing matrix, and the unknown sources correspond to the dictionary coefficients. In order to separate the streams, W and $\mathbf{a}(t)$ have to be found.

If we assume that at every time step t there are only a few sources active, we again have a sparsity prior. Treating the recordings from every time step as separate samples, we can use the SCNG algorithm to learn W and $\mathbf{a}(t)$. To model moving audio sources, the mixing matrix W needs to be time dependent. If $W(t)$ changes slowly, one can assume that it is almost constant over some short interval $[t-T, t]$. Now one can start to learn $W(t)$ at some interval and track the changes by shifting the interval. Because $W(t)$ changes only slowly, it does not need to be relearned completely.

4 Conclusion

In this paper we have provided a brief overview on sparse coding with a focus on soft-competitive learning and a few selected applications.

We started by providing a biologically motivated introduction to the principle of sparse coding and then showed that sparse coding can be seen as a generalization of vector quantization. Based on this perspective we showed that a soft-competitive approach like the well-known Neural Gas algorithm can successfully be generalized to solve the sparse coding learning problem. The results are competitive and, in many cases, even superior to computationally more intensive state-of-the-art methods such as MOD or K-SVD, both in terms of how well the representation error is minimized and how well the dictionary is reconstructed [22].

We described how sparse coding can extract features for digit recognition. A sparse basis is learned with the Sparsenet algorithm and then used to extract local coefficients. In a second step, a local maximum operation is applied to obtain shift invariance. Both operations, the sparse-coding strategy and the local maximum operation, are inspired by vision research. Finally, a Support-Vector-Machine is trained on the resulting feature vectors which yields state-of-the-art classification performance on the MNIST benchmark [18].

Furthermore, we gave examples of how sparse coding can be applied to image restoration (completion and deblurring). An important result was that when the basis functions are learned for a particular set of images, e.g. buildings or flowers, the restoration results are better than with a standard wavelet basis and also better than with a dictionary learned from unspecific natural images [23,22].

Finally, we showed that the Sparse Coding Neural Gas algorithm can be applied to a more realistic model of the cocktail-party problem. The model allows for more sources than observations, additive noise, and a time-dependent mixing matrix, which corresponds to a person that changes its position during the conversation. The proposed algorithm works online and the estimation of the underlying sources is provided immediately. The method requires that the sources are sparse and that the mixing matrix does not change too quickly [19,20].

We conclude that sparse coding is a useful generic framework anchored in the traditional disciplines of signal processing, machine learning, and neuroinformatics, allowing for significant synergies between the three.

References

1. 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)
2. Barlow, H.B.: Possible principles underlying the transformation of sensory messages. Sensory Communication pp. 217–234 (1961)
3. Bell, A., Sejnowski, T.: The "independent components" of natural scenes are edge filters. Vision Research **37**(23), 3327–3338 (1997)
4. Bruckstein, A.M., Donoho, D.L., Elad, M.: From sparse solutions of systems of equations to sparse modeling of signals and images. SIAM Review **51**(1), 34–81 (2009)
5. Candès, E.J., Tao, T.: Decoding by linear programming. IEEE Transactions on Information Theory **51**(12), 4203–4215 (2005)
6. Chen, S.S., Donoho, D.L., Saunders, M.A.: Atomic decomposition by basis pursuit. SIAM Journal on Scientific Computing **20**(1), 33–61 (1998)
7. Comon, P.: Independent component analysis, a new concept? Signal Processing **36**(3), 287–314 (1994)
8. Davis, G., Mallat, S., Avellaneda, M.: Greedy adaptive approximation. Constructive Approximation **13**, 57–89 (1997)
9. Deerwester, S., Dumais, S., Furnas, G., Landauer, T., Harshman, R.: Indexing by latent semantic analysis. Journal of the American Society for Information Science **41**(6), 391–407 (1990)
10. Ding, C., Li, T., Peng, W.: Nonnegative matrix factorization and probabilistic latent semantic indexing: Equivalence chi-square statistic, and a hybrid method. In: Proceedings of the National Conference on Artificial Intelligence, vol. 21, p. 342 (2006)
11. Donoho, D.L.: Compressed sensing. IEEE Transactions on Information Theory **52**(4), 1289–1306 (2006)
12. Elad, M., Yavneh, I.: A plurality of sparse representations is better than the sparsest one alone. IEEE Transactions on Information Theory **55**, 4701–4714 (2009)
13. Engan, K., Aase, S.O., Hakon Husoy, J.: Method of optimal directions for frame design. In: ICASSP '99: Proceedings of the Acoustics, Speech, and Signal Processing, 1999., pp. 2443–2446 (1999)
14. Field, D.J.: What is the goal of sensory coding? Neural Computation **6**(4), 559–601 (1994)
15. Haykin, S., Chen, Z.: The cocktail party problem. Neural Computation **17**(9), 1875–1902 (2005)
16. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 50–57. ACM (1999)
17. Hoyer, P.: Non-negative sparse coding. In: Proceedings of the IEEE Workshop on Neural Networks for Signal Processing XII, pp. 557–565 (2002)
18. Labusch, K., Barth, E., Martinetz, T.: Simple method for high-performance digit recognition based on sparse coding. IEEE Transactions on Neural Networks **19**(11), 1985–1989 (2008)
19. Labusch, K., Barth, E., Martinetz, T.: Approaching the time dependent cocktail party problem with online sparse coding neural gas. In: Advances in Self-Organizing Maps - WSOM 2009, vol. 5629, pp. 145–153 (2009)
20. Labusch, K., Barth, E., Martinetz, T.: Demixing jazz-music: Sparse coding neural gas for the separation of noisy overcomplete sources. Neural Network World **19**(5), 561–579 (2009)
21. Labusch, K., Barth, E., Martinetz, T.: Sparse coding neural gas: Learning of overcomplete data representations. Neurocomputing **72**(7-9), 1547–1555 (2009)
22. Labusch, K., Barth, E., Martinetz, T.: Robust and fast learning of sparse codes with stochastic gradient descent. IEEE Transactions on Selected Topics in Signal Processing **5**(5), 1048 – 1060 (2011)
23. Labusch, K., Barth, E., Martinetz, T.: Soft-competitive learning of sparse codes and its application to image reconstruction. Neurocomputing **74**(9), 1418–1428 (2011)
24. LeCun, Y.: MNIST handwritten digit database, NEC research institute. http://yann.lecun.com/exdb/mnist/
25. Lee, H., Battle, A., Raina, R., Ng, A.Y.: Efficient sparse coding algorithms. In: Advances in Neural Information Processing Systems 19, pp. 801–808 (2007)
26. Lewicki, M., et al.: Efficient coding of natural sounds. Nature Neuroscience **5**(4), 356–363 (2002)
27. Martinetz, T., Berkovich, S., Schulten, K.: "Neural-gas" network for vector quantization and its application to time-series prediction. IEEE-Transactions on Neural Networks **4**(4), 558–569 (1993)
28. Martinetz, T., Schulten, K.: A "neural-gas network" learns topologies. Artificial Neural Networks **I**, 397–402 (1991)
29. 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)
30. Paatero, P., Tapper, U.: Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. Environmetrics **5**(2), 111–126 (1994)
31. Pati, Y., Rezaiifar, R., Krishnaprasad, P.: Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. Proceedings of the 27th Annual Asilomar Conference on Signals, Systems, and Computers (1993)
32. Rebollo-Neira, L., Lowe, D.: Optimized orthogonal matching pursuit approach. IEEE Signal Processing Letters **9**(4), 137–140 (2002)
33. Shashanka, M., Raj, B., Smaragdis, P.: Sparse overcomplete decomposition for single channel speaker separation. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 2, pp. 641–644 (2007)
34. Villmann, T., Hammer, B.: Functional principal component learning using oja's method and sobolev norms. In: Advances in Self-Organizing Maps - WSOM 2009, pp. 325–333 (2009)
35. Zetzsche, C., Barth, E., Wegmann, B.: The importance of intrinsically two-dimensional image features in biological vision and picture coding. In: Digital Images and Human Vision, pp. 109–38 (1993)

Jens Hocke studied computer science at the University of Lübeck. He graduated in 2011 and works now as research assistant at the Institute for Neuro- and Bioinformatics of the University of Lübeck, where he pursues a Ph.D. degree.

Kai Labusch studied computer science at the University of Lübeck, where he graduated in 2004. He received his Ph.D. degree in computer science from the University of Lübeck. Now he is at Acrolinx.

Erhardt Barth received the Ph.D. degree in electrical and communications engineering from the Technical University of Munich, Munich, Germany. He is a Professor at the Institute for Neuro- and Bioinformatics, University of Lübeck, Lübeck, Germany, where he leads the research on human and machine vision. He has conducted research at the Universities of Melbourne and Munich, the Institute for Advanced Study in Berlin, and the NASA Vision Science and Technology Group in California. Dr. Barth is an Associate Editor of the IEEE Transactions on Image Processing.

Thomas Martinetz is full professor of computer science and director of the Institute for Neuro- and Bioinformatics. He studied Physics at the TU München and obtained his doctoral degree in Biophysics at the Beckman Institute for Advanced Science and Technology of the University of Illinois at Urbana-Champaign. From 1991 to 1996 he led the project Neural Networks for automation control at the Corporate Research Laboratories of the Siemens AG in Munich. From 1996 to 1999 he was Professor for Neural Computation at the Ruhr-University of Bochum and head of the Center for Neuroinformatics.