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

Combining Reconstruction and Discrimination with Class-Specific Sparse Coding

Stephan Hasler, Heiko Wersing, Edgar Körner

2007

Preprint:

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

LETTER Communicated by Odelia Schwartz

Combining Reconstruction and Discrimination with Class-Specific Sparse Coding

Stephan Hasler
Stephan.hasler@honda-ri.de
Heiko Wersing
Heiko.wersing@honda-ri.de
Edgar Körner
Edgar.koerner@honda-ri.de
Honda Research Institute Europe GmbH, 63073 Offenbach/Main, Germany

Sparse coding is an important approach for the unsupervised learning of sensory features. In this contribution, we present two new methods that extend the traditional sparse coding approach with supervised components. Our goal is to increase the suitability of the learned features for classification tasks while keeping most of their general representation capability. We analyze the effect of the new methods using visualization on artificial data and discuss the results on two object test sets with regard to the properties of the found feature representation.

1 Introduction

Most approaches to object recognition employ two kinds of methods: methods that learn features and methods that learn object representations. While the second group of methods can be used directly for classification by comparing a test image with the learned representation, the first group has a supporting function in finding subspaces in the data in which more robust object representations can be obtained.

For the object representation learning methods, there is a further distinction between probabilistic generative and discriminative approaches, depending on whether they model the distribution of samples in the data space (Ulusoy & Bishop, 2005). In recent years a stronger interest arose in combining the advantages of both approaches (Raina, Shen, Ng, & McCallum, 2003; Ng & Jordan, 2002). Following Ulusoy and Bishop (2005), discriminative approaches are faster and more reliable in predicting class labels, since they are trained to do so rather than to model the joint distribution of input vectors and classes. Because of this specialization in a certain classification task, these approaches suffer the drawback that they have to be retrained whenever the scenario is changed, for example, by adding a new class.

Neural Computation **19, 1897–1918** (2007) © 2007 Massachusetts Institute of Technology

Probabilistic generative methods, such as gaussian mixture models (GMMs; Mc-Lachlan & Peel, 2000), learn independent models for each class. Therefore, a new class simply adds a new model but does not influence the existing ones. Also, they are able to deal with missing information and unlabeled data. The disadvantage of generative methods is that they model details of a data distribution that may be irrelevant or even disturbing in classification tasks.

Also for the feature learning methods, a distinction exists between generative and discriminative approaches, depending on whether the learned feature subspace supports reconstruction of the data or classification. Usually both approaches train one global feature basis for the whole data distribution. The discriminative approaches are trained in a supervised and the generative approaches normally in an unsupervised manner. Although the term *generative* is often used in this context, it is misleading because the feature learning methods do not specify how new data could be generated from a learned basis by means of learning priors on how to combine the features. The probabilistic generative models do so explicitly.

There is a group of generative feature learning methods called linear generative methods. These methods search for subspaces that allow for a good reconstruction of the data vectors in terms of linear combinations of the basis functions. This means each data vector is associated with a set of coefficients that determines how the features (basis functions, weights) have to be used to yield the best reconstruction. The linear generative models differ in the constraints on how to reconstruct the data. Principal component analysis (PCA; Duda, Hart, & Stork, 2000) finds dimensions of highest variance in the data, which allows for a reconstruction with minimal information loss when using fewer features than dimensions in the data. Nonnegative matrix factorization (NMF; Lee & Seung, 1999) employs purely positive weights and coefficients and was shown to learn localized patterns that often have a direct interpretation as object parts. Sparse coding (Olshausen & Field, 1996) puts constraints on the coefficients, enforcing an efficient use of the basis functions. The principle of efficient coding resembles receptive field properties in primary visual cortex when applied to small patches of natural scenes.

As mentioned in the beginning, linear generative methods are often used to facilitate classification. So, for example, PCA was successfully applied to face recognition (Turk & Pentland, 1991), and sparse coding features were used as intermediate layer in a feature hierarchy related to the ventral visual pathway (Wersing & Körner, 2003), that yields robust classification performance for different recognition problems. However, linear generative methods for feature learning suffer the same drawback as the probabilistic generative methods: they spend resources for modeling certain dimensions in the data that might be irrelevant or disturbing for classification tasks. On the other hand, the discriminative feature learning methods concentrate on dimensions in the data that are relevant for classification but do not offer the possibility of learning features that have an interpretation as object parts. Instead, they generate very holistic, noisy-looking features. An example for those methods is the Fisher linear discriminant (Duda et al., 2000), which finds subspaces in the data where the classes are separated best in terms of Euclidean distance. This subspace is very specific for the trained scenario, which may decrease the ability to generalize to new scenarios.

The mixture of advantages and disadvantages suggests a combination of discriminative and generative properties as an attractive approach for feature learning. We decided to use the nonnegative sparse coding approach (Hoyer, 2002) as the basis for our investigations. The nonnegative sparse coding is a linear generative method that adopts the positivity constraints from the NMF. As outlined above, this property facilitates the learning of features that have an interpretation as object parts. But because the linear generative methods are mainly based on the principle of reconstruction of the data, the obtained features might not be useful for building a classifier. So the nonnegative sparse coding will focus its resources on reconstructing common parts of the classes in the first and does not concentrate on discriminative ones. By adding class-specific, supervised components to the cost function, we hope to prevent this behavior and to learn qualitatively different features that are more discriminative while keeping their interpretation as object parts.

After reviewing related work in section 2, we introduce the new methods in section 3 and analyze them using a visualization of their representation properties dependent on the cost function parameters. In section 4, we analyze the obtained feature representations for two object test sets and give our conclusions in section 5.

2 Related Work

The standard approach to sparse coding (Olshausen & Field, 1996) is formulated as a linear code representing the data. Its target is to combine efficient reconstruction with a sparse usage of the representing basis, resulting in the following cost function,

$$E_{\mathrm{S}} = \frac{1}{2}\sum_i \left\| \mathbf{x}_i - \sum_p c_{ip}\mathbf{w}_p \right\|_2^2 + \gamma \sum_i \sum_p \Phi\left(c_{ip}\right), \tag{2.1}$$

where the samples $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iK})^T$, $i = 1, \ldots, I$, and the weights $\mathbf{w}_p = \left(w_{p1}, w_{p2}, \ldots, w_{pK}\right)^T$, $p = 1, \ldots, P$, have the same dimension K. In the left reconstruction term, each $\mathbf{x}_i$ is approximated by a linear combination $\mathbf{r}_i = \sum_p c_{ip}\mathbf{w}_p$, where $\mathbf{r}_i$ is referred to as the reconstruction of the corresponding $\mathbf{x}_i$. The coefficients c_{ip} specify how much the pth weight is involved in the reconstruction of the ith data vector. The squared Euclidean norm $\|\cdot\|_2^2$ of the difference vector between an $\mathbf{x}_i$ and its reconstruction $\mathbf{r}_i$ contributes to the cost. The right sparsity term sums up the c_{ip}. The nonlinear function Φ (e.g., $\Phi(\cdot) = |\cdot|$) increases the cost the more the activation is spread over different c_{ip}, and so many of them become zero while few are highly activated. The influence of the sparsity term is scaled with the positive constant γ. The sparsity forces the weights to align more directly to the data and to reconstruct an $\mathbf{x}_i$ most sparsely if multiple possibilities exist. This enables the sparse coding to handle an overcomplete representation.

PCA (Duda et al., 2000) and NMF (Lee & Seung 1999) are based on the same measure of reconstruction as the sparse coding model described by equation 2.1, but do not put sparsity constraints on the coefficients. Using fewer weights than dimensions in the data ($P < K$), PCA forces the weights to align to the directions with the biggest variance in data space. Therefore, PCA is often used to reduce the dimensionality of data, with a minimal loss of information.

NMF differs from PCA in that it puts positivity constraints on both the weights and the coefficients. Therefore, the contribution of each weight to a certain reconstruction is purely positive and cannot be canceled out by the contribution of another weight. This limitation makes it economical to reconstruct an image with nonoverlapping weights, where each single weight is already a good reconstruction of an image part. This is often referred to as a parts-based representation. Later Hoyer (2004) added to the NMF an option to directly control the sparseness of the weights and the coefficients. He discovered that for achieving a parts-based representation with standard NMF, the same parts have to occur at the same position in the training samples. By adding sparseness constraints on the weights, a parts-based representation can be more reliably produced. In other cases, the NMF produces extremely parts-based weights. This means they contain only single pixels or small blobs and do not reveal any meaningful statistical background of the data. Adding sparseness constraints onto the coefficients forces the weights to reveal more holistic dependencies.

Nonnegative sparse coding (Hoyer, 2002) adopts the idea of a parts-based representation for sparse coding by also putting positivity constraints on the weights and the coefficients. It differs from NMF in the fact that the sparsity of the coefficients is enforced explicitly, and so nonnegative sparse coding is similar to NMF with sparseness constraints. The remaining difference between both approaches is that sparse coding methods often use simple gradient descent for the optimization of the cost function, whereas NMF methods apply multiplicative update rules that do not require the definition of a learning rate.

In the algorithms described above, each weight contributes only once to a reconstruction of a certain image, and the position of activation in the weight directly determines the position of activation in the reconstruction. Therefore, a single weight cannot represent or learn a part that occurs in different images at different locations (or in other transformations). This leads to redundancies in the coding scheme by representing transformations of one part with different weights. To overcome this limitation, Grimes and Rao (2005) proposed an extension of the sparse coding that factors an image into object features and transformations using a bilinear function. In this way the weights can contribute several times to the same reconstruction, each time undergoing another transformation beforehand (e.g., shifts to different locations). Currently the approach handles only translation, but in general it is able to deal with arbitrary transformations, such as rotation, scaling and view changes. The concept of bilinearity imposes that for a certain image, all features use the same transformations. This contradicts the notion of features as independent parts. Therefore, further extensions in Grimes and Rao (2005) go in the direction of allowing independent transformations of features per image. This is a similar method to the translation-invariant adaptation of the nonnegative sparse coding proposed in Wersing and Körner (2003) and the translation-invariant adaptation of the NMF introduced in Eggert, Wersing and Körner (2004).

The unsupervised methods mentioned above produce features with reconstructive qualities. The features are not specialized in solving a certain task and therefore could be transferred to other scenarios from those used for training. The drawback is that the extraction of statistically significant parts in high-dimensional data with unsupervised methods requires large training sets. Also, there is no guarantee that the obtained parts are useful in object recognition tasks.

Another group of methods concentrates on only discriminative properties of the features. One example of such an approach is the Fisher linear discriminant (Duda et al., 2000). It searches for a low-dimensional representation of the data that, unlike PCA, does not favor the directions of biggest variance, but the directions allowing the best separation of the classes in the data. This is done by generating a transformation matrix that minimizes the ratio of within-class scatter to between-class scatter. Thus, in some sense, this projection maximizes the signal-to-noise ratio. When Q is the number of classes in the data, the feature space has dimension $Q - 1$. This allows a linear separation of the classes only if each has a very peaked, unimodal gaussian distribution in feature space.

Discriminative features are very efficient in solving the task they are trained for, but normally lack the property of being reusable in adapted scenarios. Methods combining the advantages of unsupervised and supervised methods are rare. One is the maximum representation and discrimination feature (MRDF) approach (Talukder & Casasent 1998). It combines PCA and an adaptation of the Fisher linear discriminant, which could also handle multimodal distributions, and introduces a parameter that determines to which degree reconstruction or discrimination are desired. Since the method has no positivity constraints, the generated features are holistic and do not have a direct interpretation as object parts.

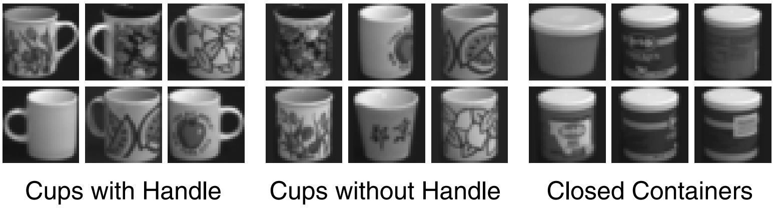

Figure 1: Example views of the three-class problem.

We propose two new methods to combine unsupervised and supervised feature learning on the basis of a nonnegative, parts-based representation.

3 Class-Specific Sparse Coding

Class specificity should denote the property of a feature to give a strong clue on the class membership of an image the feature is detected in. Following this definition, in the three-class problem shown in Figure 1, the handles show a high specificity for the cups with handle class, because whenever you recognize a handle, you can be sure that you see a view of this class. In the same way, the white caps are specific for the closed container class.

The standard sparse coding model in equation 2.1 does not care about the existence of different classes and produces features that are useful for general image reconstruction but lack the property of being class specific.

Our two new approaches extend nonnegative sparse coding with supervised components. Suppose that the data samples are split into Q subsets (classes) $\mathcal{X}_q$, with $q = 1, \ldots, Q$. Each subset has n_q elements labeled as class q. In the first approach, this class information has a direct effect on the coefficients c_{ip}, and it will therefore be referred to as coefficient coding:

$$E_C = \frac{1}{2}\sum_i \left\| \mathbf{x}_i - \sum_p c_{ip}\mathbf{w}_p \right\|_2^2 + \gamma \sum_{i,p} c_{ip} + \frac{1}{2}\,\alpha \sum_p \sum_{\substack{i,\tilde{\imath} \\ q(i)\neq q(\tilde{\imath})}} \frac{c_{ip}c_{\tilde{\imath}p}}{n_{q(i)}n_{q(\tilde{\imath})}} . \qquad (3.1)$$

We assume c_{ip} and $w_{pk} \geq 0$. For the sparsity term, we used the function $\Phi(c_{ip}) = c_{ip}$, which corresponds in the nonnegative case to the absolute value. The right coefficient term causes cost if coefficients belonging to the same weight $\mathbf{w}_p$ are active for differently labeled samples $\mathbf{x}_i$ and $\mathbf{x}_{\tilde{\imath}}$, where $q(i)$ is the label of $\mathbf{x}_i$ and $n_{q(i)}$ is the number of samples in the class of $\mathbf{x}_i$. $n_{q(i)}$ is used to normalize the effect of classes with different cardinality. The influence of the coefficient term is scaled with the positive constant α.

In the second approach, the class information has a more direct effect on the weights, and it will therefore be referred to as weight coding:

$$E_{\mathrm{W}} = \frac{1}{2}\sum_{i}\left\|\mathbf{x}_i - \sum_{p} c_{ip}\mathbf{w}_p\right\|_2^2 + \gamma\sum_{i,p} c_{ip} + \frac{1}{2}\,\beta\sum_{p}\sum_{\substack{i,\tilde{\imath}\\ q(i)\neq q(\tilde{\imath})}}\frac{\mathbf{w}_p^T\mathbf{x}_i}{n_{q(i)}}\,\frac{\mathbf{w}_p^T\mathbf{x}_{\tilde{\imath}}}{n_{q(\tilde{\imath})}}\,. \tag{3.2}$$

The right weight term causes cost if a $\mathbf{w}_p$ has a large inner product with differently labeled samples $\mathbf{x}_i$ and $\mathbf{x}_{\tilde{\imath}}$. Again, $q(i)$ denotes the label of $\mathbf{x}_i$, and $n_{q(i)}$ is the number of samples in the class of $\mathbf{x}_i$. The influence of the weight term is scaled with the positive constant β.

The minimization of the cost functions of coefficient and weight coding is done by alternately applying coefficient and weight steps as described in Wersing and Körner (2003). In the coefficient step, the cost function is minimized with respect to the c_{ip} using an asynchronous fixed-point search, while keeping the $\mathbf{w}_p$ constant. The weight step is a single gradient step with a fixed step size in the $\mathbf{w}_p$, keeping the c_{ip} constant. A more detailed description of the optimization procedure is given in the appendixes.

To give an instructive visualization of sparse coding and show the qualitatively different behavior of the new approaches, we performed optimizations for an artificial two-dimensional setting. The conclusions we draw from this toy setting hold in principle also for more complex and higher-dimensional problems. The artificial setting contains 10 samples that were distributed on the positive part of the unit circle and then assigned to two classes (see Figure 2a). These samples are reconstructed using two normalized weights. The actual visualization shows the resulting weights $\mathbf{w}_p$ and reconstructions $\mathbf{r}_i$ for different values of a control parameter of the cost function, for example, the influence of the sparsity term γ (see Figure 2b). The relative position of $\mathbf{r}_i$ and $\mathbf{w}_p$ allows conclusions on the sparsity of the reconstructions, whereas the course of the $\mathbf{w}_p$ is a direct indicator for their discriminative properties. The optimally discriminating weights are those that maximize the gradient of their dot product with samples near the border between the classes. This corresponds to the property of a linear separator. In Figure 2a, the best discriminating features would point in the direction of the coordinate axes. Hence, in the visualization of the angles (see Figure 2b), these weights lie at 0 and 90 degrees.

In Figure 3 the visualization is used to compare nonnegative sparse coding, coefficient coding, and weight coding. Figure 3a shows the typical behavior of the nonnegative sparse coding for an increasing influence factor of the sparsity term γ. Each reconstruction lies between the weights or on one of them due to the nonnegativity constraints. For $\gamma \to 0$, the reconstruction is perfect (when using at least as many weights as dimensions in the data), and the weights are aligned with the outermost $\mathbf{x}_i$. If an $\mathbf{r}_i$ lies on top of a weight symbol, this reconstruction is very sparse because it does

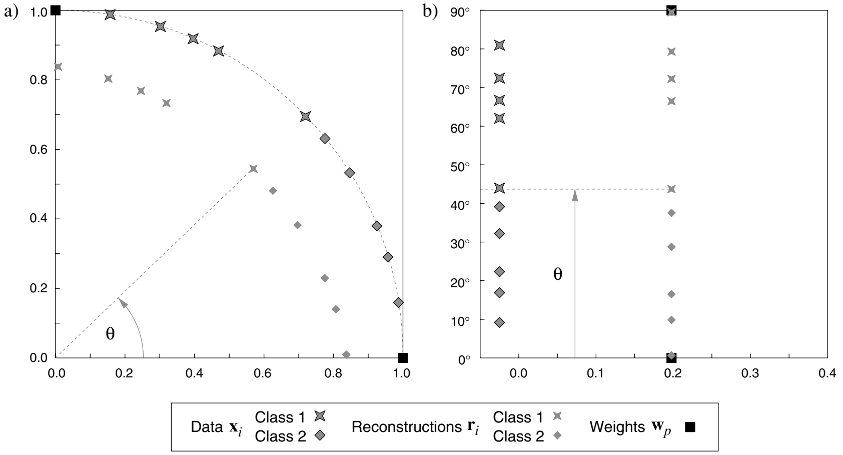

Figure 2: (a) Artificial setting, used in our visualization. The artificial setting is two-dimensional and contains 10 data samples $\mathbf{x}_i$. The $\mathbf{x}_i$ are randomly distributed on the unit circle and assigned to two classes (symbolized with a star and a diamond). The optimization employs two normalized two-dimensional weights $\mathbf{w}_p$. For a certain parameter setting, the optimization results in the shown position of weights and reconstructions $\mathbf{r}_i$. (b) Schematic description of visualization. The actual visualization shows for each symbol in *a* the angle between a ray from the origin to that symbol and the x-axis (see how θ in *a* is represented in *b*). The angles of weights and reconstructions are shown at the x-coordinate of 0.2 because the result in *a* was produced with a cost function parameter of 0.2. Visualizing $\mathbf{r}_i$ and $\mathbf{w}_p$ for different values of the same parameter will reveal its qualitative effect on the cost function.

not use the other weight at all. With increasing γ, each $\mathbf{r}_i$ gives up the use of the less suitable weight, and therefore the $\mathbf{r}_i$ unite to two main paths. At the same time, each $\mathbf{w}_p$ aligns to the center of the $\mathbf{r}_i$ assigned to it. For high values of γ, the result is therefore comparable to that of a cluster approach.

The coefficient term restricts the use of features by different classes. When its influence factor is increased for the coefficient coding (see Figure 3b), the reconstructions of each class are forced to use the same distinct weight basis (here only a single weight). So the reconstruction of the lowermost sample of the star class aligns with increasing α to the upper weight due to its class membership, while for the nonnegative sparse coding (see Figure 3a), the same $\mathbf{r}_i$ aligns to the lower weight with increasing γ due to a better sparseness. Note that the outermost two reconstructions at both sides are equal from the beginning. For high values of α, each feature is dedicated to a single class and changes its direction independent of other classes. Therefore, an increase in discriminative quality is impossible, because this would require a strong influence of different classes onto the same weight.

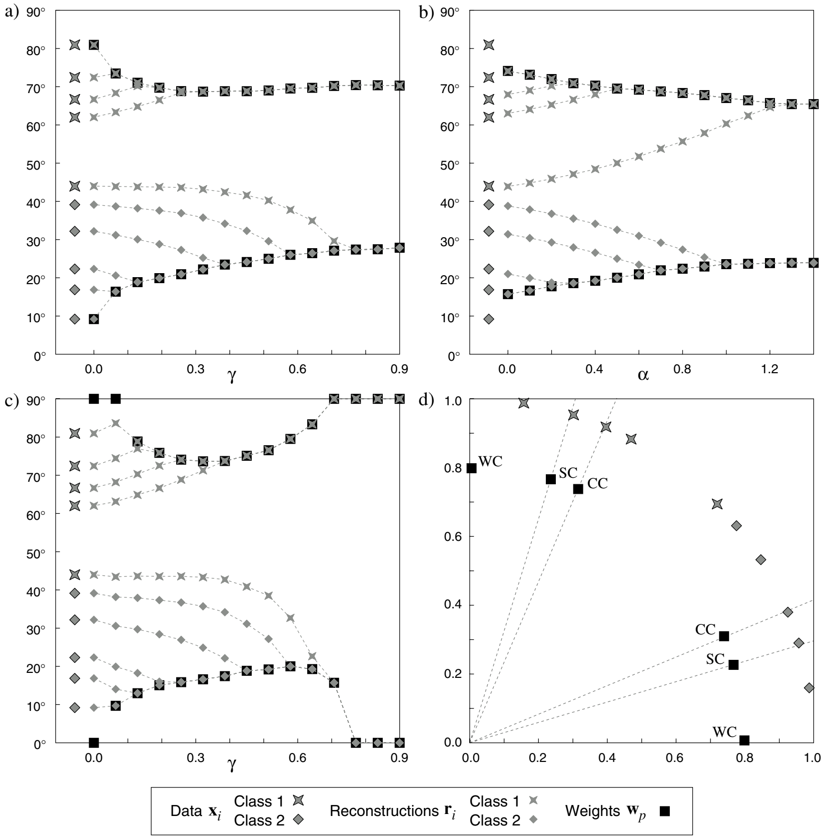

Figure 3: Influence of certain parameters on different cost functions. (a) Nonnegative sparse coding (SC). The results of optimization are plotted for 15 different influence factors of the sparsity term γ. (b) Coefficient coding (CC). The influence factor of the coefficient term α is varied, while γ is set to 0.05. (c) Weight coding (WC). The influence factor of the sparsity term γ is varied, while the influence factor of the weight term β is set to 0.3. (d) Accumulated results. The angular positions of the weights $\mathbf{w}_p$ are visualized for typical parameter settings of the different approaches. A detailed description is given in the text.

For the weight coding (see Figure 3c), there is a complex interplay between sparsity term and weight term. When the weight term dominates, as for very small values of γ, it removes activation from the lower weight that it shares with members of the upper class and vice versa. So one weight moves to the top and the other to the bottom. This means each $\mathbf{w}_p$ aligns to the direction that is most specific for the class it is representing. In the nonnegative case, this can be referred to as a gain in discriminative power. The weight term also dominates for very high values of γ. In this case, the reconstruction cost is near its maximum, and the algorithm tries to minimize the weight cost at least. Only for a certain range of parameters is there a meaningful combination of discriminative and reconstructive properties.

Figure 3d shows some typical weights obtained with the different approaches. The features of nonnegative sparse coding lie relatively close to the borders of the data distribution, offering a good compromise between reconstruction and sparseness. For the coefficient coding, it is expensive to reconstruct the samples near the class border using both weights strongly. Therefore, the features move closer to the class centers. Only the weight coding finds out that activation in y is specific or diagnostic for the star class and activation in x for the diamond class.

The results on the toy setting showed that nonnegative sparse coding limits the use of features globally, so each data sample is reconstructed using a small subset of features. To reduce the reconstruction cost, the features model parts that occur most frequently among the samples. But those parts are usually not specific for certain classes. In the extreme case, sparse coding works like a cluster approach, using a single weight per sample. The coefficient coding penalizes the use of a feature for different classes, so each class tends to use a distinct feature subset to reconstruct its samples. In this way, coefficient coding cannot avoid that parts that occur frequently in different classes attract weights. These weights are therefore not discriminative or class specific. More than this, the approach may waste resources by modeling those parts for each class independently. The weight coding directly penalizes if a feature reflects a part that occurs in different classes. As a result, the presence of a certain feature in an image gives a good indication for one or only a few classes.

By enforcing the use of a distinct set of features for reconstructing each class, the coefficient coding mimics the behavior of probabilistic generative methods like a GMM. A GMM models a data distribution with the help of a finite number of gaussians. When the GMM framework is used to build a classifier, it also trains a separate GMM for each class. Therefore, GMM and coefficient coding represent or reconstruct details of the data distribution that may be irrelevant for determining the class label (Ulusoy & Bishop, 2005).

The weight coding instead punishes directly if a weight contains activation that is shared among different classes and therefore irrelevant for determining the class label. The cost function of the weight coding can be rewritten as

$$E_{\mathrm{W}} = E_{\mathrm{S}} + \frac{1}{2}\,\beta \sum_{p} \sum_{\substack{q,\tilde{q}\\ q\neq\tilde{q}}} \left(\mathbf{w}_p^T \overline{\mathbf{x}}_q\right)\left(\mathbf{w}_p^T \overline{\mathbf{x}}_{\tilde{q}}\right) \quad \text{with:} \quad \overline{\mathbf{x}}_q = \frac{1}{n_q} \sum_{\mathbf{x}\in\mathcal{X}_q} \mathbf{x}\,, \tag{3.2}$$

where $\overline{\mathbf{x}}_q$ is the K-dimensional mean of the samples with label q. In this form, the weight coding shows some relation to the Fisher linear discriminant, when neglecting that the weight coding is nonnegative in all elements while the Fisher linear discriminant is not. The Fisher linear discriminant minimizes in the two-class case the following cost function with respect to $\mathbf{w}$:

$$E_{\mathrm{F}} = \frac{\sum_{\mathbf{x}\in\mathcal{X}_1}\left(\mathbf{w}^T\mathbf{x} - \mathbf{w}^T\overline{\mathbf{x}}_1\right)^2 + \sum_{\mathbf{x}\in\mathcal{X}_2}\left(\mathbf{w}^T\mathbf{x} - \mathbf{w}^T\overline{\mathbf{x}}_2\right)^2}{\left(\mathbf{w}^T\left(\overline{\mathbf{x}}_1 - \overline{\mathbf{x}}_2\right)\right)^2}\,. \tag{3.3}$$

The numerator prefers directions where the variance within each class is minimal and the denominator tries to separate the means of the classes as far as possible from each other. By multiplying the denominator out, you get:

$$\left(\mathbf{w}^T\left(\overline{\mathbf{x}}_1 - \overline{\mathbf{x}}_2\right)\right)^2 = \left(\mathbf{w}^T\overline{\mathbf{x}}_1\right)^2 - 2\left(\mathbf{w}^T\overline{\mathbf{x}}_1\right)\left(\mathbf{w}^T\overline{\mathbf{x}}_2\right) + \left(\mathbf{w}^T\overline{\mathbf{x}}_2\right)^2\,. \tag{3.4}$$

The second term is equivalent to the weight term. The first and the third terms force the weight to align to the input pattern. In the weight coding, this is done by the interplay of reconstruction term and sparsity term. Assuming a unimodal, peaked distribution for each class in data space, the numerator of the Fisher linear discriminant plays no significant role, and hence both approaches put comparable forces on the features.

The weight coding is also similar to the MRDF approach (Talukder & Casasent, 1998), which combines supervised and unsupervised feature learning by combining an adaptation of the Fisher linear discriminant with PCA. The advantage of the weight coding is that it can produce a parts-based, overcomplete representation, while the number of features in the Fisher linear discriminant is limited by the number of classes and in the MRDF by the number of dimensions in the data. The discriminative component of the MRDF approach tries to increase the distance of the individual members of different classes in the feature space. On the contrary, the weight term handles only the means of the class members. This limits its suitability on classes with unimodal data distributions. Another disadvantage of the weight coding is that the two parameters have to be chosen carefully. When the influence of the sparsity term is too weak, the weight term can force the features to point to meaningless dimensions—those where no class has activation.

4 Results on Two Scenarios

To further analyze the qualitative and quantitative differences between coefficient coding and weight coding, both approaches have been applied to two scenarios. The first scenario is the three-class problem shown in

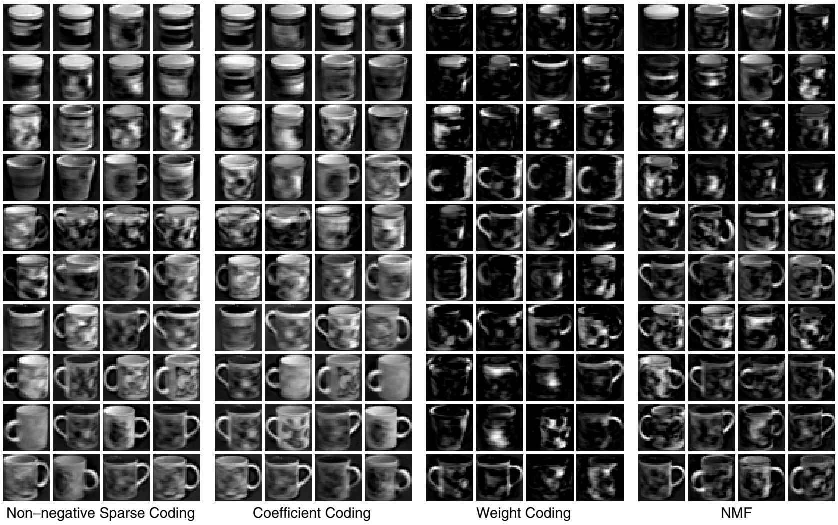

Figure 4: Features trained on first scenario with different approaches. The features for each approach are arranged from top left to bottom right by decreasing mutual information. The features of the nonnegative sparse coding and the coefficient coding are very similar to each other. The features of the weight coding are less view specific but much more parts based and class specific. In some of the features, there is a focus on the handles of the cups. In these cases also, the part of the cup opposite the handle is pronounced, since the presence of the handle at one side shifts the cup to the other side of the image frame, where otherwise no activation is present. In the cup-related features, the opening is not highlighted, since activation in this part of the image is more typical for the containers with the white caps. Therefore, there are features containing caps but no cup-related features like a handle. The NMF features are also parts based but not class specific and so lie visually somewhere in between.

Figure 1. Cups with visible handle, cups with no or occluded handle, and some round containers with caps from the COIL-100 database (Nayar, Nene, & Murase, 1996) were combined to three classes, each containing 140 views. The gray-scale images were resized to a resolution of 32×32 pixels in advance. Forty features were trained using the same influence $\gamma = 0.1$ of the sparsity term and relatively high values for the coefficient term $\alpha = 4.0$ and for the weight term $\beta = 0.1$.

Figure 4 shows the resulting features sorted by their individual mutual information, the calculation of which is described later. The features of nonnegative sparse coding and coefficient coding are very holistic and view specific. The features of weight coding and NMF are both sparser and more parts based, but the weight coding clearly emphasizes class-specific

Table 1: Values of the Terms of the Cost Functions for the First Scenario.

	Reconstruction	Sparsity	Coefficient	Weight
SC	$7.718 \cdot 10^2$	$6.039 \cdot 10^3$	(19.98)	$(5.357 \cdot 10^4)$
CC	$8.146 \cdot 10^2$	$5.909 \cdot 10^3$	9.505	$(5.607 \cdot 10^4)$
WC	$8.138 \cdot 10^2$	$8.952 \cdot 10^3$	(65.95)	$2.215 \cdot 10^4$
NMF	$6.866 \cdot 10^2$	$(7.489 \cdot 10^3)$	(46.76)	$(3.464 \cdot 10^4)$

Notes: The terms do not include their influences (γ, α, and β) on the total cost. Values in parentheses were not used for optimization, but are shown to highlight qualitative differences between features.

parts. So there is a group of features highlighting handles, while in all NMF features that contain handles, the whole cup is recognizable. The first NMF feature represents the white cap of a container, while the following features show the opening and the rim of a cup. Both feature types have a strong overlap at the top of the image frame, and therefore the cap features will also respond to cups and the rim-opening features to the containers. The weight coding does not tolerate this and makes the features sparser to better work out the differences of cups and containers.

Table 1 lists the values of the terms of the cost functions after optimization. These values are useful to interpret the effect of our two new approaches compared to the nonnegative sparse coding: The coefficient term puts a penalty on the use of features across different classes, which leads to a reduced feature basis for reconstructing each class. As a result, there is an increase in the reconstruction cost and a decrease in the sparsity cost. The demand for sparsity of the coefficients in the nonnegative sparse coding has an opposite effect on the weights, forcing them to become very view specific and leading to a higher reconstruction cost. In the weight coding, the weight term removes activation from the features. They become less view specific, which causes an increase of the sparsity cost.

To evaluate the discriminative power of the trained features, we chose to calculate the mutual information between the features and the classes. The mutual information is a measure for the dependency between two or more random variables. In our case, these variables are the detection of the different features and the class label, both varying over the set of samples. The mutual information tells how much the detection of certain features in a sample restricts the possibility of different class hypotheses. Therefore, a high mutual information is a measure of discriminative power and a desired feature property. Unfortunately, the direct optimization of mutual information conveyed by a set of features about a class demands the continuous PDF of the data. For low-dimensional data, the PDF can be estimated from the training samples using the Parzen window technique (Kwak & Choi, 2002). However, in high-dimensional data, the approach is computationally too expensive and requires a huge set of samples.

Table 2: Mutual Information for the First Scenario.

	SC	CC	WC	NMF
Mutual information	1.7831	1.7314	3.4068	2.5708
Mean error rate	0.2839	0.2950	0.2045	0.2695
Standard deviation	0.0809	0.0814	0.0840	0.0820

Notes: The table lists the mutual information conveyed by the pool of features about the three classes (see: Ullman & Bart, 2004). Also some error rates are given performing 100 nearest-neighbor classifications per approach.

For the results in Table 2, we used a calculation similar to the method applied in Ullman and Bart (2004) to select informative image fragments. First, for each feature, an optimal threshold is determined. For this purpose the dot product with each sample is calculated. By applying a threshold to the results of this calculation, a binary detection variable over the set of samples is generated. The class label is also a discrete class variable over the set of samples. The optimal threshold is the one that maximizes the mutual information between the class variable and the detection variable. There are different ways to calculate the information between the classes and the set of features. Simply taking the sum of the individual mutual information the features convey would totally neglect their dependencies. Another way is to join the single values of the detection variables to a binary feature vector per sample and then calculate the mutual information between this vector and the classes. This approach is the mathematically correct one, but a perfect result tells only that no binary feature vector is used in different classes and not how well the information is distributed over the set of features. Therefore we adopted the iterative process proposed by Ullman and Bart (2004) to calculate the values in Table 2. First, the feature conveying the most mutual information is chosen, and later the features with the most additional mutual information. Because the calculation of the additional mutual information given a set of already selected features is impractical, we also adopted the heuristic of Ullman & Bart (2004). In this heuristic, the additional mutual information of one candidate feature is calculated with respect to each single selected feature. The minimum of these values is assigned to the candidate feature. The candidate feature with the highest assigned value is selected. This heuristic guarantees that the selected feature is informative and differs from the features already selected. The sum of the mutual information of the first feature and additional information of the other features is the given value. Because of the heuristic approach, this value can be higher than the entropy of the class distribution. The weight coding has the highest value and the sparse coding and the coefficient coding, the lowest ones. The NMF has an intermediate value. Furthermore, in the sparse coding and the coefficient coding, the twenty-sixth selected feature of the 40 existing ones already has an additional mutual information of less than 0.0005. In the NMF it is the thirty-fourth and in the weight coding the thirty-ninth. So in some sense, the information is best distributed in the weight coding approach.

Table 2 also gives some error rates performing nearest-neighbor classifications on the three-class problem. In 100 runs per approach, three representatives per class were chosen randomly out of the 140 views. These representatives were transformed into feature space by calculating the dot product with the trained features. Each of the remaining views was then assigned to the closest representative in feature space. The error rates were calculated on the basis of wrong class assignments. The results strongly depend on the chosen representatives, causing a high standard deviation. Nevertheless, we confirmed with a t-test that the error rate of the weight coding, is significantly (with $p = 0.001$) lower than that of the other approaches. This supports the claim for an increased discriminative component of the weight coding features. The coefficient coding shows in the mean the worst performance because forcing each class to use a distinct set of features prevents the development of discriminative properties. Note that the projection of the image views on a feature space, which is simply a complete rotation of the original orthogonal basis system would not influence the result of a nearest-neighbor classifier. The reason for the differences shown is that the used algorithms produce a nonorthogonal basis with a reduced dimension.

To show that the better performance of the weight coding is not simply caused by the higher degree of sparseness of its features, an additional test was performed. The features of the NMF were trained again, putting additional sparsity constraints on them following the method proposed in Hoyer (2004). So each NMF feature was ensured to have the same L2 norm (1.0) as each weight coding feature and the average L1 norm of the weight coding features. In this way the error rate of the NMF decreased from 0.2695 to 0.2457 but is still 4% higher than that of the weight coding. The mutual information increased from 2.5708 to 2.9853 compared to 3.4068 of the weight coding. Although these results show that sparsity of the weights has indeed some influence on the performance, the main difference is caused by the supervised component of the weight coding.

For a second scenario, we acquired the HRI-10 database that consists of 10 classes. A single class contains 9 similar objects, each made up of 100 views taken during a rotation around the vertical axis (see Figure 5). Five objects are used for the training of the features and the remaining four objects for testing.

We trained 80 features per approach on the gray-scale images, which were scaled to a resolution of 40×40 pixels. When training on the full rotation, we observed that NMF and weight coding produced very sparse, blob-like features that showed identical classification performance, while the weight coding features had slightly higher mutual information. Only

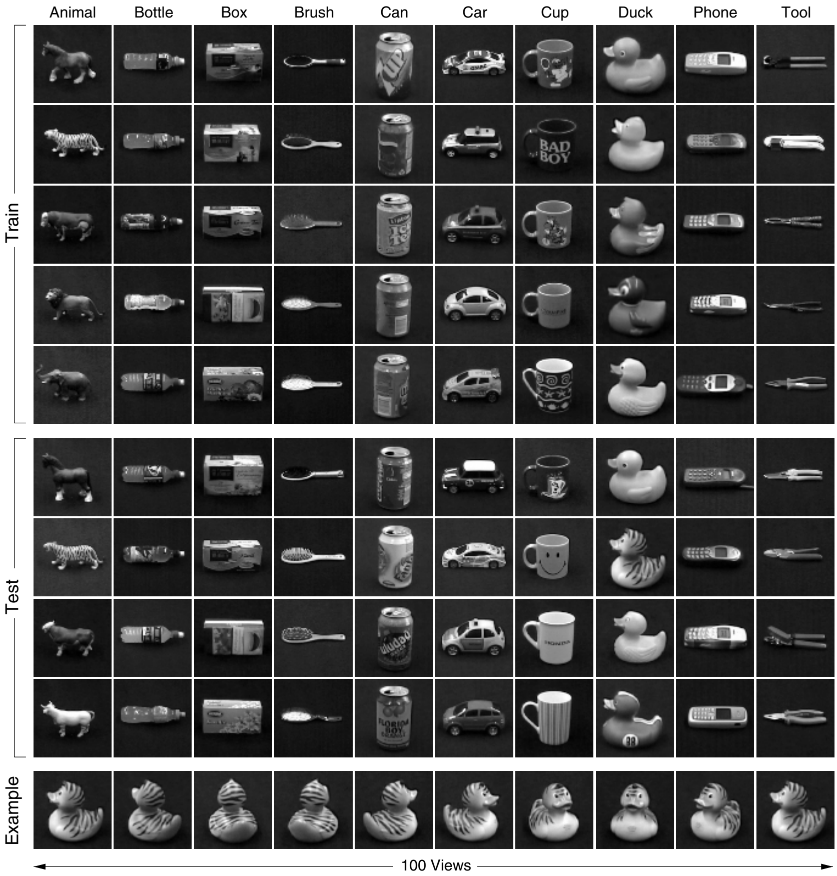

Figure 5: HRI-10 database. The database consists of 10 classes, each containing 9 similar objects. The first 5 objects per class are used for training and the remaining 4 objects for testing. Each object is represented by 100 views covering a full rotation around the vertical axis.

with such sparse features were the approaches able to reconstruct the wide variety of images. Because of this problem, we reduced the complexity of the problem by using only views taken from −35 to +35 degrees from the first side view. Alternatively we could increase the number of features, but this would heavily increase the computation time and allow the standard NMF to produce even sparser features, while the sparsity constraint on the coefficients could prevent this development for the weight coding.

The features trained on the simplified database are shown in Figure 6. The influence of the sparsity term was set to $\gamma = 0.05$, the influence of the coefficient term was $\alpha = 5$, and the influence of the weight term was

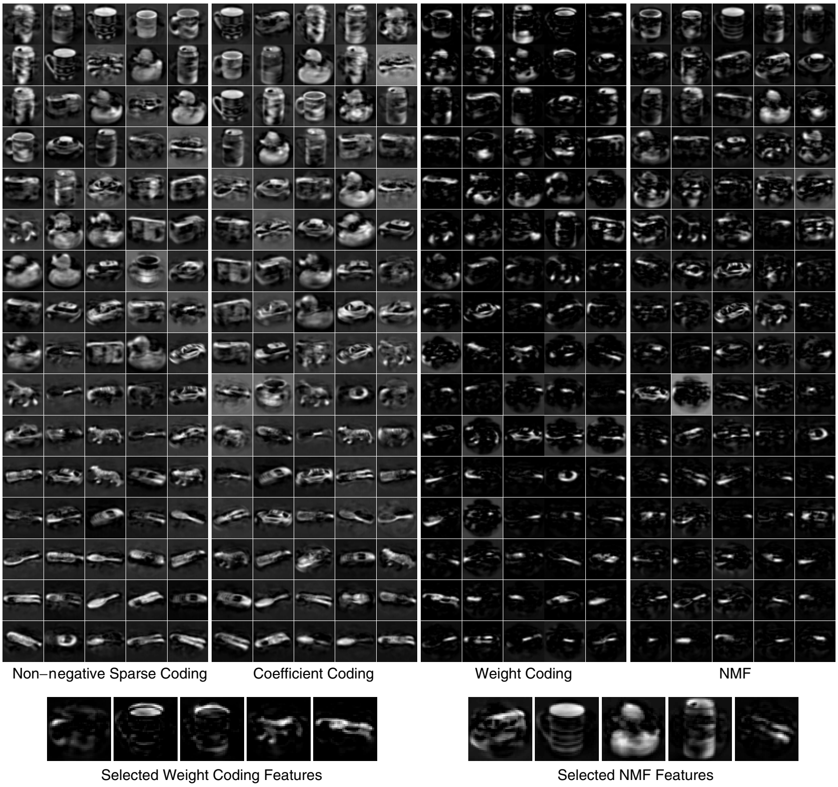

Figure 6: Features trained on the second scenario with different approaches. The features for each approach are arranged from top left to bottom right by decreasing mutual information. At the bottom, some selected features of weight coding and NMF are shown to highlight qualitative differences between both approaches. As for the first scenario, the features of nonnegative sparse coding and coefficient coding are very holistic and view specific, and little difference between them can be revealed. Weight coding and NMF again produce parts-based features, but this time the difference between both approaches is not as obvious as for the first scenario. But a carefull look shows some qualitative differences, which are discussed in the text.

$\beta = 0.4$. Again the features of the nonnegative sparse coding and the coefficient coding are holistic and similar to each other. Both NMF and weight coding produce sparser features. Although the difference between NMF and weight coding is not that obvious this time, some qualitative differences can be found. So the first selected weight coding feature again shows a separated handle that cannot be found among the NMF features. The first selected NMF feature is a car feature that looks very boxlike. The weight

Table 3: Mutual Information for the Second Scenario.

	NNSC	CC	WC	NMF
Mutual information	6.5459	6.2302	10.7695	8.8732
Mean error rate	0.2465	0.2537	0.1896	0.2074
Standard deviation	0.0330	0.0338	0.0325	0.0316

Notes: The table lists the mutual information conveyed by the pool of features about the 10 classes (see Ullman & Bart, 2004). Also some error rates are given performing 100 nearest-neighbor classifications per approach.

coding does not tolerate this and therefore makes a strong distinction between car and box features. That the weight coding features are not always sparser than the NMF features is shown on the example of the can opener (fifth selected feature). Also interesting is the selected NMF cup feature that seems to be split into two features by the weight coding. A reason may be that the upper part of the rim is a very specific pattern for all cups, while the bright opening occurs in only a certain cup and is represented because this cup would otherwise cause a very high reconstruction cost.

The evaluation of mutual information and classification rate was performed on the four test objects per class, keeping the procedure of the first scenario. The results are shown in Table 3. This time, five representatives per class were chosen randomly out of 240 views that covered the limited rotation of the four objects as described above. The weight coding has a higher mutual information than the NMF, and both approaches are superior to nonnegative sparse coding and coefficient coding. The error rate in the nearest-neighbor experiment is for the weight coding 2% lower than that of NMF and 6% lower than that of coefficient coding and nonnegative sparse coding. The high standard deviation is again caused by the random selection of representatives. But the improved mean error rate of weight coding features is significant applying a t-test with $p = 0.001$.

When adjusting the sparsity of each NMF feature to the average sparsity of the weight coding features, the error rate decreased from 0.2074 to 0.2032. This is still significantly higher than the 0.1896 of the weight coding. The mutual information increased from 8.8732 to 9.6584 compared to the 10.7695 of the weight coding. So again the performance is improved to some degree by the sparsity but does not reach the weight coding results. Also, the arrangement of the features in Figure 6 indicates that very sparse features normally provide lower information gain. This was also observed by Ullman and Bart (2004), who discovered the superiority of features with intermediate complexity.

Despite the simplicity of using segmented views of objects, the two classification problems are suitable to show that the features of the weight coding are more class specific and diagnostic than the object templates produced by sparse coding. More complex scenarios would have increased the computational cost drastically (e.g., by requiring the approaches to be invariant to position and size of the objects), while we would expect the same qualitative differences.

5 Conclusion

In this letter, two new class-specific extensions of the nonnegative sparse coding were introduced. Normally, unsupervised generative feature learning methods spend resources to model details of the data that are irrelevant for classification tasks. The goal of extending the cost function of the nonnegative sparse coding with discriminative components was to shift the focus of some resources from frequently occurring parts to diagnostic ones, in this way increasing the suitability of the trained features for classification tasks.

It was shown that the coefficient coding does not increase the discriminative quality of the features because it prevents multiple classes from influencing the same weight. This is due to the fact that the coefficient coding restricts the use of features by different classes, whereas the weight coding directly penalizes the suitability of features for different classes and so successfully combines reconstruction and discrimination. The weight coding is related to the Fisher linear discriminant and the MRDF, but does not reduce the intraclass variance. Its advantage is that it learns localized, parts-based features.

We used an artificial two-dimensional setting to introduce sparse coding to somebody new in the field and visualize the different behavior of the new approaches. Furthermore, we showed for two object scenarios that the weight coding results in qualitative other features from that produced by NMF or nonnegative sparse coding. This goes with higher mutual information of the features and increased classification performance.

To test the difficulty of the scenarios, we used our features with a nearest-neighbor classifier (NNC) and compared the performance to that of a single-layer perceptron (SLP). In this way, we always get a lower classification rate for our approaches. When using some categories as clutter for testing and evaluating the false-positive rate by means of a receiver operating characteristic, weight coding performs slightly better. But those results are more a reflection of the different nature of SLP and NNC and not of the quality of the weight coding features.

Mutual information is an unbiased measure for the discriminative quality of a feature learning method, whereas the classification rate is influenced by the auxiliary method used to calculate it, for example, NNC. A comparison of different features is only fair, when sticking to the same classifier, because otherwise it is not clear which component caused the difference. For the same reason, the NNC results are not comparable with those of superior, standard classification schemes such as an SLP or a GMM.

However, the question may arise as to how in principle a discriminative method like an SLP could benefit from the combination with a reconstructive component. Purely discriminative approaches suffer from the drawback that they may overspecialize on the training scenario; they perfectly learn in which way a class differs from negative examples present during training. If these examples do not cover well the expected variations during testing, the overspecialization impairs the classifier in rejecting unseen clutter images because they may not differ in the learned features from the class. Keeping reconstructive information means keeping information on what the class is, regardless of which other classes existed during training. This gives the classifier the chance to reject a test image based on the inability to reconstruct it.

Recent studies suggest that generative methods perform better when training data are limited (Raina et al., 2003), because they converge much faster. As more training data are available, discriminative models take the lead by reaching a lower asymptotic error (Ng & Jordan, 2002). There is also biological evidence for this process as outlined in Logothetis and Sheinberg (1996). When a new object is being learned, holistic snapshots are stored, keeping as much information as possible. With increasing familiarity of the stimulus, prototypes are generated keeping only meaningful, discriminative parts, enabling them to generalize over nonmeaningful parts. In relation to this, weight coding features are useful for building a representation of objects that are somewhere between novel and familiar by moving away from a full reconstruction of the stimulus to a prototypical representation focusing more on the diagnostic object parts. Weight coding can provide a basis for building an efficient object representation, a prerequisite for robust and fast object recognition.

Appendix A: Nonnegative Sparse Coding

The minimization of the cost function in equation 2.1 is done by alternately applying coefficient and weight steps as described in Wersing and Körner (2003). In the coefficient step, the cost function is minimized with respect to the c_{ip} using an asynchronous fixed-point search, keeping the $\mathbf{w}_p$ constant. To do that, the derivation of E_S with respect to a certain c_{ip} is set to zero, leading to the update rule,

$$c_{ip} := \sigma \left(\mathbf{w}_p^T \mathbf{x}_i - \sum_{\substack{\tilde{p} \\ \tilde{p} \neq p}} c_{i\tilde{p}} \mathbf{w}_{\tilde{p}}^T \mathbf{w}_p - \gamma \right) \left(\mathbf{w}_p^T \mathbf{w}_p \right)^{-1} , \tag{A.1}$$

where $\sigma(\cdot) = \max(0, \cdot)$ ensures the positivity of the coefficients. This update rule is applied to randomly chosen c_{ip} until convergence. The weight step is a single gradient step with a fixed step size η in the $\mathbf{w}_p$, keeping the c_{ip} constant:

$$\mathbf{w}_p := \sigma\left(\mathbf{w}_p - \eta\left[\sum_i \sum_{\tilde{p}} c_{i\tilde{p}}\mathbf{w}_{\tilde{p}}c_{ip} - \sum_i \mathbf{x}_i c_{ip}\right]\right). \tag{A.2}$$

The weight step is executed for each $\mathbf{w}_p$ at the same time, and $\sigma(\cdot)$ is applied component-wise. Before the next coefficient step, the weights are normalized using

$$\mathbf{w}_p := \frac{\mathbf{w}_p}{\|\mathbf{w}_p\|_2}. \tag{A.3}$$

Appendix B: Coefficient Coding

The optimization of the cost function, equation 3.1, is nearly the same as for the nonnegative sparse coding. Only the update rule for the coefficients changes into

$$c_{ip} := \sigma\left(\mathbf{w}_p^T\mathbf{x}_i - \sum_{\substack{\tilde{p}\\ \tilde{p}\neq p}} c_{i\tilde{p}}\mathbf{w}_{\tilde{p}}^T\mathbf{w}_p - \gamma - \alpha\sum_{\substack{\tilde{\imath}\\ q(\tilde{\imath})\neq q(i)}} \frac{c_{\tilde{\imath}p}}{n_{q(\tilde{\imath})}n_{q(i)}}\right)\left(\mathbf{w}_p^T\mathbf{w}_p\right)^{-1}, \tag{B.1}$$

while the update of the weights follows exactly equations A.2 and A.3.

Appendix C: Weight Coding

The weight term of the cost function, equation 3.2, has effect only on the weight step, and so the update rule for the coefficients remains equation A.1, while the gradient step in the weights becomes

$$\mathbf{w}_p := \sigma\left(\mathbf{w}_p - \eta\left[\sum_i \sum_{\tilde{p}} c_{i\tilde{p}}\mathbf{w}_{\tilde{p}}c_{ip} - \sum_i \mathbf{x}_i c_{ip} + \beta\sum_{\substack{i,\tilde{\imath}\\ q(i)\neq q(\tilde{\imath})}} \frac{\mathbf{x}_i\left(\mathbf{w}_p^T\mathbf{x}_{\tilde{\imath}}\right)}{n_{q(i)}n_{q(\tilde{\imath})}}\right]\right), \tag{C.1}$$

followed by the normalization of the weights, equation A.3.

References

Duda, R. O., Hart, P. E., & Stork, D. G. (2000). *Pattern classification* (2nd ed.). New York: Wiley.

Eggert, J., Wersing, H., & Körner, E. (2004). Transformation-invariant representation and NMF. In *Proc. IEEE International Joint Conference on Neural Networks* (pp. 2535–2539). Piscataway, NJ: IEEE.

Grimes, D. B., & Rao, R. P. N. (2005). Bilinear sparse coding for invariant vision. *Neural Computation 17*(1), 47–73.

Hoyer, P. (2002). Non-negative sparse coding. In *Proc. IEEE Workshop on Neural Networks for Signal Processing* (pp. 557–565). Piscataway, NJ: IEEE.

Hoyer, P. (2004). Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research, 5*, 1457–1469.

Kwak, N., & Choi, C.-H. (2002). Input feature selection by mutual information based on parzen window. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*(12), 1667–1671.

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

Logothetis, N. K., & Sheinberg, D. L. (1996). Visual object recognition. *Annual Review of Neuroscience, 19*, 577–621.

McLachlan, G., & Peel, D. (2000). *Finite mixture models*. New York: Wiley.

Nayar, S. K., Nene, S. A., & Murase, H. (1996). Real-time 100 object recognition system. In *Proc. IEEE Conference on Robotics and Automation*, (Vol. 3 (pp. 2321–2325). Piscataway, NJ: IEEE.

Ng, A. Y., & Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In T. G. Dietterich, S. Becker, & Z. Ghahramani (Eds.), *Advances in neural information processing systems, 14*. Cambridge, MA: MIT Press.

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

Raina, R., Shen, Y., Ng, A. Y., & McCallum, A. (2003). Classification with hybrid generative/discriminative models. In S. Becker, S. Thrun, & K. Obermayer (Eds.), *Advances in neural information processing systems, 15*. Cambridge, MA: MIT Press.

Talukder, A. & Casasent, D. (1998). Classification and pose estimation of objects using nonlinear features. In *Proc. SPIE Applications and Science of Computational Intelligence* (Vol. 3390, pp. 12–23). Bellingham, WA: SPIE.

Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience, 3*(1), 71–86.

Ullman, S., & Bart, E. (2004). Recognition invariance obtained by extended and invariant features. *Neural Networks, 17*(1), 833–848.

Ulusoy, I., & Bishop, C. M. (2005). Generative versus discriminative methods for object recognition. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition* (pp. 258–265). Piscataway, NJ: IEEE.

Wersing, H., & Körner, E. (2003). Learning optimized features for hierarchical models of invariant object recognition. *Neural Computation, 15*(7), 1559–1588.

Received October 14, 2005; accepted October 7, 2006.