Nonlinear Supervised Locality Preserving Projections for Visual Pattern Discrimination

Erik M. Rehn
Bernstein Center for Computational Neuroscience
Berlin, Germany
erik.m.rehn@gmail.com

Henning Sprekeler
Institute for Theoretical Biology
Humboldt-Universität zu Berlin, Germany
and the Department of Engineering
University of Cambridge, United Kingdom
h.sprekeler@hu-berlin.de

Abstract—Learning representations that disentangle hidden explanatory factors in data has proven beneficial for effective pattern classification. Slow feature analysis (SFA) is a nonlinear dimensionality reduction technique that provides a useful representation for classification if the training data is sequential and transitions between classes are rare. The pattern discrimination ability of SFA has been attributed to the equivalence of linear SFA and linear discriminant analysis (LDA) under certain conditions. LDA, however, is often outperformed by locality preserving projections (LPP) when the data lies on or near a low-dimensional manifold. Here, we take a unified manifold learning perspective on LPP, LDA and SFA. We suggest that the discrimination ability of SFA is better explained by its relation to LPP than to LDA, and give an example of a situation where linear SFA outperforms LDA. We then propose a novel supervised manifold learning architecture that combines hierarchical nonlinear expansions, as commonly used for SFA, with supervised LPP. It learns a nonlinear parametric data representation that explicitly takes both the class labels and the manifold structure of the data into account. As an experimental validation, we show that this approach outperforms previously proposed models on the NORB object recognition dataset.

Keywords—manifold learning; slow feature analysis; locality preserving projections, object recognition, classification

I. INTRODUCTION

Manifold learning has gained a lot of interest in recent years, and numerous methods have been proposed that aim at “unfolding” high-dimensional data manifolds into low-dimensional representations. However, most either suffer from being linear, e.g., principal component analysis and locality preserving projections (LPP) [1], or from lacking a globally defined input-output mapping, e.g., IsoMap [2], local linear embedding [3] and Laplacian eigenmaps (LEM) [4]. The former are of limited use for strongly curved manifolds, while the latter only provide a representation of the training data, and require interpolation techniques for previously unseen samples.

At least two approaches to overcome these caveats have been proposed [5, 6]. These techniques assume that the neighborhood relations between training data points are generated by a computable kernel function in the input space (e.g., a Gaussian heat-kernel). However, when explicit knowledge about the training data manifold is present, the best strategy might not be to compute the neighborhood relations from the data itself, as we demonstrate in the following.

Here we present a novel nonlinear parametric manifold learning architecture that does not make any assumptions on how neighborhood relations are computed, is conceptually simple, easy to implement, and as the experimental evaluation shows, well suited for high-dimensional supervised visual pattern classification.

II. A UNIFIED VIEW OF LEM, LPP, LDA AND SFA

Our approach combines SFA [7] and supervised LPP (SLPP) [8]. We will first provide a brief overview of the relation of these two methods and other similar techniques (Figure 1).

SFA is a (generally nonlinear) technique that extracts slowly varying sources from time series by minimizing the mean squared difference between the representations of subsequent data points. Algorithmically, SFA applies a specified set of nonlinear functions to expand the input and then solves a linear problem in the expanded space. SFA is closely related to manifold learning [9] and it has recently been shown that the objectives of SFA and LEM become identical if the elements of the adjacency matrix of the LEM problem correspond to the transition probabilities that generate the temporal dynamics of the SFA training data [10, 11]. By generalizing SFA from temporal to arbitrary neighborhoods one obtains a nonlinear parametric approximation of LEM, which approaches LEM as the richness of the selected nonlinear expansion grows.

Fig. 1. Relation between LEM, LPP, LDA and SFA and variations thereof. NFDA refers to nonlinear Fisher discriminant analysis as proposed in [12], and is equivalent to LDA with a preceding nonlinear expansion.
Similarly, LPP is a linear approximation of LEM, which learns a linear transformation that optimally preserves neighborhood relations in the training data, characterized by an adjacency matrix for the training data [1]. LPP is equivalent to linear SFA with generalized adjacency [13], and yields a nonlinear approximation of LEM when applied after a nonlinear expansion as is common for SFA. This algorithm, which we here will refer to as nonlinear LPP (NLPP), can thus be derived either from SFA by generalizing from temporal to arbitrary neighborhoods [10], or from LPP by means of a preceding nonlinear expansion.

SFA and LPP are both unsupervised, but can be altered to incorporate class labels. To our knowledge, Berkes was the first to acknowledge a connection between linear discriminant analysis (LDA) and SFA, although he never published this insight. Instead he proposed polynomially expanded LDA as a method for pattern recognition [12]. Later, Klampfl and Maass [14] described the relation between SFA and LDA. They demonstrated that when the temporal adjacency is close to class adjacency, i.e., in the limit of vanishing interclass transition probability and homogeneous intra-class transition probability, linear SFA becomes equivalent to LDA. They argued that this similarity between SFA and LDA is the reason for the discriminatory capabilities of SFA.

For LPP, class labels can be incorporated by allowing non-vanishing adjacency only among patterns of the same class, an approach termed supervised LPP (SLPP) [8]. As the number of neighbors for SLPP approaches the number of patterns of each class, SLPP becomes equivalent to LDA.

III. LINEAR SFA CAN OUTPERFORM LDA

Interestingly, LPP with a local adjacency matrix (rather than the full class adjacency matrix of LDA) can outperform LDA on, e.g., face recognition tasks where the data has a clear manifold structure [1]. Because SFA was originally designed for temporal sequences that contain slowly varying features, the associated adjacency matrix (i.e., the matrix of transition probabilities) is more likely to be local rather than uniform within classes. Linear SFA is therefore more closely related to LPP than to LDA, suggesting that SFA could in fact outperform LDA on classification tasks, rather than being a mere approximation thereof.

To investigate this hypothesis we compared supervised LPP, LDA, and linear SFA on a synthetically generated digit recognition dataset where the patterns have a sequential ordering. In short, the generation of a training and test set sequence is made through translation, scaling, and rotation of a prototypical class pattern (16×16 pixels). The transformations are applied sequentially such that one sequence per class is generated. There are no transitions between classes, except at the end and beginning of each class sequence. The training set thus consists of ten concatenated sequences, one for each of the digits 0-9. These sequences can be seen as trajectories on four-dimensional manifolds embedded in the pixel space, where the four dimensions correspond to the translation (x and y), rotation, and scaling of the class prototypes. A short example of such a pattern sequence can be seen in Figure 2.

Figure 3 shows the performance of SLPP with an adjacency matrix computed for different numbers of k-nearest class neighbors (according to the L2 norm), as well as the performance of linear SFA and LDA (i.e., SLPP with $k = 180$), for both the training and test dataset. Note that the neighborhood is symmetrized, i.e., $x_i$ and $x_j$ is among the $k$-nearest patterns of $x_i$, or if $x_i$ are neighbors if $x_i$ nearest patterns of $x_i$. All edges also have the same weight. Since the maximum number of linearly independent components that can be produced by LDA is equal to the number of classes minus one, only nine output dimensions were kept for classification. Following Berkes [12], a Gaussian classifier was used to compute the accuracy.

The performance on the training set increases with increasing neighborhood size, while the test performance increases initially but then starts to decline once the neighborhood size surpasses an optimal (around 40). This behavior bears a striking resemblance to an overfitting situation. Although it is not exactly clear how the number of neighbors can be related to model complexity, this analogy is consistent with the view that the local neighborhood of LPP works as a regularization of LDA, i.e., it makes LDA less sensitive to outliers [21].

Interestingly, linear SFA clearly outperforms LDA on the test set. This suggests that the previous assertion that SFA gains its pattern discriminative properties from its relationship to LDA [19] is potentially understating the discrimination powers of SFA. The (local) temporal adjacency of SFA is more akin to LPP, and can lead to better generalization than the full class adjacency of LDA, when the data has a clear manifold structure.

Fig. 2. A piece of one of the ten training sequences used in the experiment of Figure 3. The sequence starts with a prototypical “7” in the top-left corner and then moves from left to right. Translation, scaling and rotation are applied simultaneously. The test set looks similar but without any sequential ordering.

Fig. 3. SLPP accuracy for different number of neighbors on a synthetically generated digit dataset. The curves have the classical overfitting shape, with increasing training set accuracy and decreasing test set accuracy as the neighborhood size, $k$, grows. The two horizontal lines mark the SFA accuracy on the same data. The SFA accuracy on the test set is clearly much higher than for LDA ($k = 180$).


IV. HIERARCHICAL NONLINEAR SUPERVISED LOCALITY PRESERVING PROJECTIONS

For datasets with a curved manifold structure, nonlinear dimensionality reduction clearly holds more potential than linear techniques. Although nonlinear expansions have previously been used for LPP-type approaches to pattern recognition [15], this form of nonlinear manifold learning does not seem to have been exploited to its full extent. The reason is presumably that nonlinear expansions often become intractable for high-dimensional data. For SFA, this curse of dimensionality has been addressed by the use of hierarchical networks that split the input space into subspaces, which are then processed in a converging feed-forward hierarchy [7, 9]. Here, we demonstrate that the same hierarchical processing strategy can be used for NLPP to make it a tractable approach to nonlinear manifold learning of high-dimensional visual data.

A. Architecture

The architecture proposed here consists of two layers with feed-forward connections, each composed of a set of nodes with a retinotopic mapping of the input (Figure 4). In the first layer each node receives input from a small patch of the whole input image with 50% overlap in both directions, i.e., for an input patch size of N×N pixels the stride is N/2 pixels. The nodes extract a set of nonlinear features by alternation of nonlinear expansions and LPP dimensionality reduction. The expansion and reduction sequence can be repeated multiple times, a technique often used for SFA to provide more nonlinearity without the need of more high-dimensional expansions. In the first layer, we use weight sharing, i.e., all nodes learn the same features.

The second layer consists of four nodes, each receiving input from one quadrant of the image. Each node is trained individually with the input from its respective child nodes, allowing the model to adapt to differences in input statistics between receptive field quadrants. The nodes process their input in four stages. First, an LPP step reduces the dimensionality of the input from the first layer. This is followed by a quadratic expansion (see [7]) and a second LPP step. Finally, the output is normalized (see below) and fed into a classifier.

B. Nonlinear Expansion

In addition to the standard polynomial expansion used for SFA [7], we also tried a feature expansion based on k-means for image data, as suggested by Coates et al. [16]. To this end, all image patches received by the lowest layer nodes are collected and clustered by k-means clustering. Each cluster prototype is then used to generate an expansion feature:

\[ z_k = \max(0, |u_k| - u_k) \]

where \( u_k \) is the Euclidian distance from an input patch to feature \( k \), and \( |u| \) the average distance to all features.

For polynomial expansions, the expansion itself is preceded by a PCA filtering step that removes dimensions with a variance lower than 10^{-12} times the variance of the first principal component, to avoid singular matrices. For the k-means expansion, this problem of singular dimensions is much less likely to arise and therefore no filtering is applied.

C. Output normalization

Between the classifier and the last LPP step of the second layer, each output vector \( y_i \) is normalized. First, to avoid the large outliers for test data that often arise from polynomial expansions, the output is subjected to a cut-off at two standard deviations:

\[ y_{i,\text{cut}}^k = \begin{cases} \min(y_i^k, 2\sigma(y_i)) & \text{if } y_i^k \geq 0 \\ \max(y_i^k, -2\sigma(y_i)) & \text{if } y_i^k < 0 \end{cases} \]

After this cut-off, each output vector is normalized to make the smallest or largest component equal to -1 or 1, respectively:

\[ y_{i,\text{norm}}^k = y_{i,\text{cut}}^k / \max(|y_{i,\text{cut}}^k|) \]

D. Transformation adjacency

How to construct the best adjacency matrix for supervised NLPP depends on the problem at hand, and upon which information is available about the manifold structure of the training set. A basic approach for classification is to use symmetric k-nearest class neighbor adjacency, according to Euclidian distances in the input space.

However, for some datasets more information than just class labels is available. For example, we might expect two patterns to be similar because we know they are images of the same object, only from slightly different viewpoints. In such cases this information can be used to choose the adjacency matrix. Following Franzius et al. [9], we refer to the variables that describe how a pattern has been generated as the configuration of a pattern. Configurations could be, e.g., the position of an object in an image, lighting and viewing angles, etc. Let \( h_i \) denote the configuration vector of pattern \( x_i \). We can then define a transformation adjacency between patterns of the same class based on the similarity of their configuration:

\[ w_{ij} = w(h_i, h_j) \]

The key difference to the k-nearest neighbor class neighborhood (and most other types of graphs used for manifold learning) is that the adjacency is not based on some distance metric in the input space. Rather, it is computed using...
explicit knowledge of the manifold structure of the data. As we show in the following section, exploiting such knowledge can significantly improve classification performance. Note also that this approach naturally falls back to pure class adjacency (i.e., to nonlinear LDA) when no configuration is available.

V. EXPERIMENTS ON NORB

The NORB dataset is an object recognition benchmark designed to test a system’s ability to learn and recognize generic object categories merely using shape information [17]. The dataset consists of grayscale photos depicting 50 different toys divided into 5 categories (animals, humans, planes, cars and trucks). We evaluate our system on the “small” version of the dataset in which the background is uniform and all objects are centered. Each object is photographed with a stereo camera from 18 azimuth angles, 9 elevations and under 6 different lighting conditions adding up to a total of 1944 images per object. The images have a size of 96×96 pixels and the whole dataset is split up into a training and a test set with 5 category instances in each. In total the training and test set each consists of 48,600 labeled images (24,300 stereo pairs).

Along with each image, \( x_i \), a configuration vector, \( h_i \), is available, which encodes the object instance, viewpoint and lighting of the image. In addition to these four variables, we also add a flag indicating if this is the left or right stereo version of the image, resulting in a configuration vector with the following structure:

\[
h_i = (\text{lighting}, \text{stereo}, \text{elevation}, \text{azimuth}, \text{instance})
\]

A. Model specifics

In the bottom layer we use a receptive field size of 6×6 pixels, yielding in total 961 nodes (3 pixel stride). The first LPP step reduces the dimensionality to 18 dimensions, while the first LPP step in the second layer reduces the dimensionality to 42 dimensions. This is followed by a quadratic expansion, resulting in 945 dimensions, and a second LPP step which outputs 48 dimensions.

The training of the model is performed on all 48,600 images. The model is then duplicated, producing a new model for the stereo images with a receptive field of 2×96×96 pixels. For classifier training and testing this model is then applied to stereo images, i.e., the left and right version of each image are concatenated. The total output dimensionality of the duplicated model is 48×4×2 = 384 dimensions. The reported results are from a 1-nearest neighbor classifier (NNC), a support vector machine with radial basis function kernel provided no or marginal performance increase.

Following the suggestions of Coates et al. [16], all 6×6 pixel input patches are normalized to zero mean and unit variance. This corresponds to local brightness and contrast normalization, respectively, and dramatically improves the performance on NORB because it removes variability induced by the different lighting conditions. Coates et al. also suggested training a whitening filter on all the normalized patches, but this resulted in a performance reduction of our system. We suspect that the benefit of whitening reported by Coates et al. might be related to the sum-pooling technique they applied rather than the actual feature extraction.

As described above, five configuration variables are available per pattern. The performance of the model under different transformation adjacency functions based on these variables is presented in Table I. The transformation adjacency function, \( w(h_i, h_j) \), is defined for NORB as:

\[
w_{ij} = \begin{cases} 1 & \text{if } b_{ij} = 1 \\ 0 & \text{else} \end{cases}.
\]

Note that \( b_{ij} \), the distance in the transformation space, is always at least 1 since two patterns of the same class never have identical configuration. The columns of Table I, corresponding to the elements of \( h_i \) (lighting, stereo, etc.), refer to the configuration variable differences considered when computing the adjacency graph. If for example \( b_{ij} = 1 \) and this distance is due to a difference in elevation (i.e., all other variables are the same) only the rows with a “1” in the elevation column will have an adjacency matrix with \( w_{ij} = 1 \). We refer to this as the existence of an elevation neighborhood. By controlling what transformations give rise to a neighborhood it is possible to get insights into what kind of invariances are important for classification of the dataset.

The lighting and instance neighborhoods are handled in an all-to-all fashion as they do not form natural sequences. The configuration distance with respect to these two variables are therefore binary in contrast to the elevation and azimuth where the distance corresponds to the angular differences in camera viewpoint.

The results of Table I show that lighting adjacency significantly degrades the performance of the model. This is somewhat puzzling given that the presence of all other neighborhoods results in a performance increase. A possible explanation is that the lighting does not form natural sequences, since the direction and intensity of the light changes abruptly between the six conditions. However, the same is true for the instances, and the instance neighborhood is important for good performance. The problem of NORB is mainly to generalize to new object shapes, which share features, but still are quite different from the training objects. Hence, the instance neighborhood is important since it produces inner-class shape invariance. On the other hand, the lighting conditions of the test and training data are identical.

<table>
<thead>
<tr>
<th>Expansion</th>
<th>Li (^b)</th>
<th>St (^a)</th>
<th>Ef</th>
<th>Az (^c)</th>
<th>In (^d)</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sparse (k)-means (1600 features)</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>97.0</td>
<td></td>
</tr>
<tr>
<td>Quadratic</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>98.5</td>
<td></td>
</tr>
<tr>
<td>Quadratic×2</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>98.4</td>
<td></td>
</tr>
<tr>
<td>Quadratic×4</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>98.1</td>
<td></td>
</tr>
<tr>
<td>Quadratic×8</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>95.7</td>
<td></td>
</tr>
</tbody>
</table>

\(^a\) Lighting, \(^b\) Stereo, \(^c\) Elevation, \(^d\) Azimuth, \(^e\) Instance
The performance drop due to the lighting neighborhood might thus better be explained by the increased density of the adjacency matrix. As we have seen in the comparison between SLPP and LDA, a sparse local neighborhood is beneficial to avoid overfitting. When edges are added between patterns that only differ in lighting these edges might thus lead to overfitting without contributing to any invariance that helps classification.

B. Performance of k-nearest neighborhood adjacency

To allow a fair comparison with other models that do not exploit the pattern configurations, we also trained a model using k-nearest neighbor adjacency. To find the optimal neighborhood size, \( k \), we performed a nested 5-fold cross-validation on the NORB training set. For each fold one of the five instance categories of each class was selected as test set, while the model was trained on the remaining instance categories. The average accuracy of all folds was then computed for different values of \( k \). During cross-validation, a single quadratic expansion was used in the bottom level, due to computational constraints. For the final evaluation, we used the \( k \)-means expansion instead. The classification was done using a nearest neighbor classifier both during the cross-validation and for the final evaluation.

The highest cross-validation accuracy was observed for \( k = 40 \). This parameter value was then used to train a model with \( k \)-nearest neighbors adjacency on the whole NORB training set, yielding a performance on the test set of 97.4%.

We suggest that the lower performance compared to transformation adjacency arises because the neighborhood computed from Euclidian distances fails to capture the true manifold structure of the data. Euclidian distance is arguably a questionable metric for image similarity. Transformation adjacency, in contrast, creates a semantically sound distance measure and explicitly captures image transformations with respect to which an invariance is essential for the problem at hand.

C. Comparison to previously published results

Table II shows the performance on NORB for the best hierarchical models using transformation and \( k \)-nearest neighbors adjacency, together with the best results found in the literature. Our approach (nonlinear SLPP) clearly outperforms previously suggested models, both for \( k \)-nearest class neighbors adjacency and transformation adjacency.

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deep Belief Network [18]</td>
<td>95.0</td>
</tr>
<tr>
<td>Convolutional Neural Network [19]</td>
<td>95.43</td>
</tr>
<tr>
<td>Sparse ( k )-means, sum-pooling (1600 features) [16]</td>
<td>97.0</td>
</tr>
<tr>
<td>Sparse ( k )-means, sum-pooling (4000 features) [16]</td>
<td>97.27</td>
</tr>
<tr>
<td>Nonlinear SLPP (1600 features), 40-NN class adj.</td>
<td>97.4</td>
</tr>
<tr>
<td>Nonlinear SLPP (1600 features), trans. adj.</td>
<td>98.5</td>
</tr>
</tbody>
</table>

VI. RELATED WORK

The architecture presented here is strongly inspired by the architectures suggested by Coates et al. [16] and Franzius et al. [9]. Coates et al. demonstrate that if the input data is whitened and a sparse nonlinearity is used to calculate the feature activation, a simple and shallow unsupervised feature extraction strategy based on k-means is highly effective on NORB (normalized-uniform) and other datasets. We build on these results and show that the standard polynomial expansion, normally used with hierarchical SFA, can indeed be favorably replaced by k-means feature extraction in the lowest layer of the model. However, repeated quadratic expansions have lower computational complexity while giving similar performance, and might therefore be preferable in situations where the computational cost is important.

Coates et al. show that low stride is important for the performance of their model. Our model extracts features much sparser (stride 3 vs. 1 pixel on NORB), which results in considerably lower dimensionality of the first layer output. Their model uses a very simple sum-pooling strategy for dimensionality reduction while we employ supervised NLPP. Hence, we trade reduced complexity in the first layer for a more advanced pooling strategy in the second layer. However, when the dimensionality of the input grows beyond the \( 96\times96\times2 \) pixels of the NORB dataset, hierarchical NLPP might be a more viable strategy than the model of Coates et al. because the dimensionality can be kept under control.

Franzius et al. [9] use a hierarchical SFA architecture to learn invariant representations of 3D-objects. Their model is trained in an unsupervised fashion on sequences of images depicting objects as they are smoothly but randomly translated, rotated, and scaled. Both object identity and pose/viewpoint are learned. Unfortunately, the evaluation is performed on a custom dataset, which makes the performance of their architecture hard to compare to other methods. The top layers of our model are similar to that of Franzius et al., but the standard unsupervised SFA is replaced by NLPP using an adjacency matrix computed either from the data and the class labels (\( k \)-NN adjacency), or constructed using explicit knowledge of the manifold structure of the training data (transformation adjacency). SFA is commonly seen as an unsupervised learning algorithm. However, in light of its connection to SLPP and LDA it is from a pattern discrimination perspective more appropriate to view SFA as a supervised method where the temporal structure of the input acts as a supervisor. The temporal neighborhood, implicit in the standard formulation of SFA by the use of the derivative, is just one of many ways to construct a graph which encodes a possible low-dimensional manifold structure in the training data. The hierarchical architecture, with only two layers, was intentionally kept identical to the model of Coates et al. as it allows for easier comparison of the two approaches. Franzius et al. use a deeper hierarchy, with more layers, and it might be that such architecture is preferable over the relative shallowness of our model. Future work will reveal whether this is the case.

Recently, Escalante & Wiskott [11] suggested an SFA variant with generalized neighborhood (equivalent to NLPP) and showed that it yields good results for a traffic sign classification benchmark. They used full class neighborhood and did not discuss potential advantages of local neighborhoods for classification, maybe because they are not
as beneficial for the traffic sign dataset, which has a less prominent manifold structure than NORB.

VII. DISCUSSION & CONCLUSION

The experimental results presented here speak in favor of nonlinear functional manifold learning implemented through nonlinearly expanded LPP being a viable strategy for visual object recognition. By exploiting knowledge of how the training patterns are related under a set of transformations, we show increased performance on NORB compared to previously proposed models. On NORB, unsupervised $k$-nearest neighbors adjacency also yields higher performance than any previously published model. These results support the claim that within-class locality preservation is a better objective for visual pattern discrimination than class separability, as previously found for face recognition with LPP [1]. In fact, the overfitting behavior of LDA seems to be even more severe when applied with nonlinear expansions in a hierarchy. Our hierarchical NLPP model with full class adjacency shows poor performance on NORB (< 80%). Local neighborhoods are clearly preferable, and if explicit knowledge of the manifold structure is available, it can be exploited to further increase the performance. This is an important finding for real-world applications since it implies that more information than class labels can be exploited during training to improve the generalization performance of a pattern recognizer. For instance, if one wants to teach a system to classify different types of vehicles, it might be more efficient to train it on image sequences of cars rather than just a random selection of car images, as normally done when solving object recognition problems. This is in line with previous findings, that training on videos can improve the features learned for object recognition [21].

In the ideal case, the hyperparameters of the model, the output dimensionality of every NLPP step, should be found using cross-validation to avoid overfitting. Due to computational constraints these values have instead been selected through educated guesses. However, the model performance is not strongly dependent on the exact values suggested here. The accuracy is good as long as the output dimensionality of each layer is sufficiently high.

A major motivation behind this study is that the best previous models have not exploited the information that in the NORB dataset, the same objects are photographed from different angles and under different lighting conditions. They treat each image individually instead of as part of the manifold on which the images can naturally be placed. We here show that it is possible to use information of how the images relate to each other to create local neighborhood graphs, which improve classification performance and reduce model complexity. To ensure that this result generalizes to other datasets, we also trained and tested the same architecture (except for lower-dimensionality) on the digit classification problem of Figure 2, and compared it to several other techniques, including [12] and [16]. So far the suggested NLPP architecture performs better then any other method we tried, and consistently gives higher performance with transformation adjacency than $k$-NN adjacency. Furthermore, although Franzius et al. have previously shown that hierarchical SFA works for high-dimensional visual object recognition, the efficiency of their model is hard to assess due to the use of a custom dataset. The NORB dataset is a well-studied problem, and therefore permits a comparison of SFA/NLPP with other approaches to object recognition.

REFERENCES


