

Local Subspace Classifier and Local Subspace SOM

Jorma Laaksonen

Helsinki University of Technology
 Laboratory of Computer and Information Science
 P.O. Box 2200, FIN-02015 HUT, Finland

Abstract

This paper presents a new classification technique. The proposed method is named the *Local Subspace Classifier* (LSC) which indicates the kinship of the algorithm to the subspace classification methods. On the other hand, the Local Subspace Classifier is an heir of pure prototype classification methods like the k -NN rule. Therefore, it is argued that, in a way, LSC fills the gap between the subspace and prototype methods of classification. From the domain of the prototype-based classifiers, LSC brings the benefits related to the local nature of the classification, while simultaneously utilizing the capability of the subspace classifiers to produce generalizations from the training sample.

A further enhancement of the LSC principle named the *Convex Local Subspace Classifier* (LSC+) and the application of the LSC principle to the SOM are also presented. Experiments with handwritten digit data demonstrate the good classification accuracy obtainable with the LSC and LSC+ classifiers.

1 Introduction

A new statistical classification technique, named the Local Subspace Classifier (LSC) and its two further enhancements, the Convex Local Subspace Classifier (LSC+) and the Local Subspace SOM (LSSOM) are introduced in this paper. The performance of the two Local Subspace Classifier techniques is demonstrated with experiments using handwritten digit data.

When the LSC processes an input vector, it first searches for the nearest prototypes to that vector in all the classes. A local subspace – or more precisely, a *linear manifold* – is then spanned by these prototype vectors in each class. The classification is then based on the minimum distance from the input vector to these subspaces. The process is illustrated in a two-class case in Figure 1.

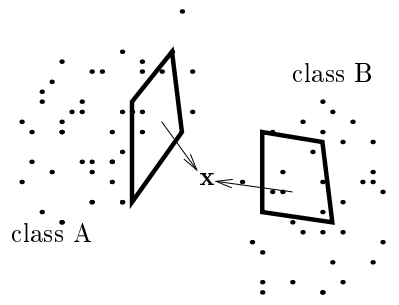


Figure 1: The Local Subspace Classifier in a two-class case. The input vector “ $\mathbf{x}$ ” is classified based on the distances to the local subspaces spanned by a few prototypes in the two classes.

2 Basic Local Subspace Classifier

A D -dimensional linear manifold $\mathcal{L}$ of the d -dimensional space is defined by a matrix $\mathbf{U} \in \mathbb{R}^{d \times D}$ having rank D , and an offset vector $\boldsymbol{\mu} \in \mathbb{R}^d$, provided that $D \leq d$,

$$\mathcal{L}_{\mathbf{U}, \boldsymbol{\mu}} = \{\mathbf{x} \mid \mathbf{x} = \mathbf{U}\mathbf{z} + \boldsymbol{\mu} ; \mathbf{z} \in \mathbb{R}^D\} . \quad (1)$$

The same manifold can alternatively be defined by $D+1$ prototypes provided that the set of prototypes is not degenerate. Figure 2 illustrates a two-dimensional linear manifold defined by the three sample handwritten '4's in the corners of the triangle. Each of the three corners represents a pure instant of a normalized 32×32 -sized human printed glyph whereas the images in between are linear combinations thereof. The interpolation produces intermediate images containing holes. Despite this shortcoming, it may be assumed that such interpolated images are as good representatives of the class as the original prototypes. Therefore, if the input vector resembles any of such artificial images, it can be considered to originate from that class.

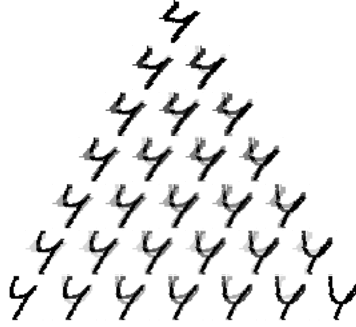


Figure 2: A piece of a two-dimensional linear manifold spanned by three examples of handwritten digits '4' in the corners. The gray areas in the intermediate virtual digit images reflect the gradual pixel-wise change from one real image to another.

The prototypes forming the classifier are denoted by $\mathbf{m}_{ij}$ where $j = 1, \dots, c$ is the index of the class and $i = 1, \dots, N_j$ indexes the prototypes in that class. The manifold dimension D may be different for each class. Therefore, the class-dependent dimensions are denoted D_j . When classifying a vector $\mathbf{x}$, the following is done for each class $j = 1, \dots, c$:

1. Find the $D_j + 1$ nearest prototype vectors to $\mathbf{x}$ and denote them $\mathbf{m}_{0j}, \dots, \mathbf{m}_{D_j j}$.
2. Form a $d \times D_j$ -dimensional basis of the vectors $\{\mathbf{m}_{1j} - \mathbf{m}_{0j}, \dots, \mathbf{m}_{D_j j} - \mathbf{m}_{0j}\}$.
3. Orthonormalize the basis using Gram-Schmidt process [2] and denote the resulting matrix $\mathbf{U}_j = (\mathbf{u}_{1j} \dots \mathbf{u}_{D_j j})$.
4. Find the projection of $\mathbf{x} - \mathbf{m}_{0j}$ on the manifold $\mathcal{L}_{\mathbf{U}_j, \mathbf{m}_{0j}}$ determined by $\mathbf{U}_j$ and $\mathbf{m}_{0j}$:

$$\hat{\mathbf{x}}'_j = \mathbf{U}_j \mathbf{U}_j^T (\mathbf{x} - \mathbf{m}_{0j}) . \quad (2)$$

5. Calculate the residual of $\mathbf{x}$ relative to the manifold $\mathcal{L}_{\mathbf{U}_j, \mathbf{m}_{0j}}$:

$$\tilde{\mathbf{x}}_j = \mathbf{x} - \hat{\mathbf{x}}_j = \mathbf{x} - (\mathbf{m}_{0j} + \hat{\mathbf{x}}'_j) = (\mathbf{I} - \mathbf{U}_j \mathbf{U}_j^T)(\mathbf{x} - \mathbf{m}_{0j}) . \quad (3)$$

The vector $\mathbf{x}$ is then classified according to minimal $\|\tilde{\mathbf{x}}_j\|$ to the class j , i.e.,

$$g_{\text{LSC}}(\mathbf{x}) = \underset{j=1, \dots, c}{\operatorname{argmin}} \|\tilde{\mathbf{x}}_j\| . \quad (4)$$

In any case, the residual length from the input vector $\mathbf{x}$ to the linear manifold is equal to or smaller than to the nearest prototype, i.e., $\|\tilde{\mathbf{x}}_j\| \leq \|\mathbf{x} - \mathbf{m}_{0j}\|$. These entities are sketched in Figure 3.

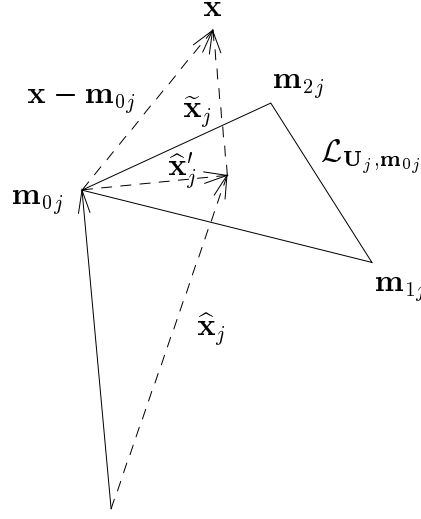


Figure 3: The d -dimensional entities involved in the LSC classification when $D_j = 2$.

By introducing the multipliers $\{c_{0j}, \dots, c_{D_jj}\}$ forming the coefficient vector $\mathbf{c}_j = (c_{0j} \dots c_{D_jj})^T$, and using the matrix $\mathbf{M}_j = (\mathbf{m}_{0j} \dots \mathbf{m}_{D_jj})$, the projection vector $\hat{\mathbf{x}}_j$ can be expressed

$$\hat{\mathbf{x}}_j = \mathbf{m}_{0j} + \hat{\mathbf{x}}'_j = \sum_{i=0}^{D_j} c_{ij} \mathbf{m}_{ij} = \mathbf{M}_j \mathbf{c}_j, \quad \sum_{i=0}^{D_j} c_{ij} = 1. \quad (5)$$

If the explicit values for the coefficients c_{ij} are needed, they can be solved with matrix pseudo-inversion

$$\mathbf{c}_j = \mathbf{M}_j^\dagger \hat{\mathbf{x}}_j = (\mathbf{M}_j^T \mathbf{M}_j)^{-1} \mathbf{M}_j^T \hat{\mathbf{x}}_j. \quad (6)$$

It can be seen that the LSC method degenerates to the 1-NN rule for $D = 0$. The two leftmost images in Figure 4 illustrate the difference of the 1-NN classifier and the LSC method when $D = 1$. It can be seen, that the input vector displayed as “ $\mathbf{x}$ ” is classified differently in the two cases even though the prototype vectors in the classifier remain the same.

3 LSC+ Classifier

The center image in Figure 4 shows that the input vector $\mathbf{x}$ may occasionally be projected outside the convex region determined by the prototypes $\mathbf{m}_{0j}, \dots, \mathbf{m}_{D_jj}$. Whether this is beneficial or disadvantageous depends on the particular data sets involved. If no such extrapolation from the original prototypes is permitted, the projection may be enforced to lie within the area delimited by the prototype vectors. As a result, the orthogonality of the projection is lost. The modified method is here called the *Convex Local Subspace Classifier* (LSC+). The appended plus sign arises from the observation that the coefficients $\{c_{0j}, \dots, c_{D_jj}\}$ in (5) are constrained to be positive in the case the projection vector $\hat{\mathbf{x}}$ is not allowed to exceed the convex region determined by the prototype vectors.

In LSC+, the projection vector $\hat{\mathbf{x}}$ is thus not the one producing the shortest residual $\tilde{\mathbf{x}}$ but primarily lies within the convex region bounded by the nearest prototypes and only secondarily minimizes the residual length. Obviously, if the orthogonal and convex projections differ, the convex projection lies on the boundary of the convex area and thus, one or more of the coefficients are equal to zero. The residual length $\|\tilde{\mathbf{x}}\|$ can in the convex case never be shorter than in the

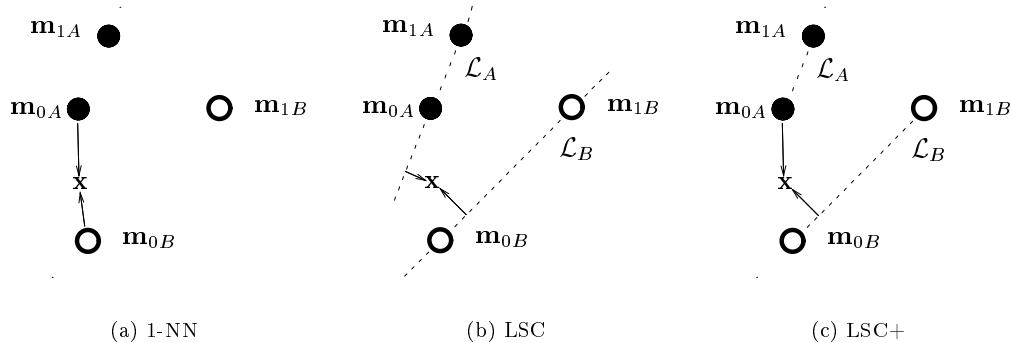


Figure 4: Comparison of the 1-NN rule and the basic and convex versions of the Local Subspace Classifier. Using the 1-NN and LSC+ classifiers, x is classified as “B”, but as “A” when using the LSC method.

orthogonal case. The two rightmost images in Figure 4 show how a vector x is classified differently using LSC and LSC+.

The convexity condition can be achieved iteratively by removing from the basis M_j the vectors which have negative coefficients. This is effectively the same as forcing the corresponding coefficients to be equal to zero. Then, the coefficients for the remaining vectors are solved again and checked for negativeness. The dimension D of the linear manifold is decreased by one for each vector removed. In general, all the degraded vectors can be removed at once and the matrix pseudo-inversion (6) needs only to be performed at most twice.

4 Local Subspace SOM

The LSC technique may also be gracefully combined with other types of prototype classifiers such as the Learning k -Nearest Neighbor classifier (L- k -NN) [6], the Learning Vector Quantization (LVQ) [4], and especially with the topology preserving vector quantization produced by the *Self-Organizing Map* (SOM) [4]. When the delta adaptation rule is applied to the Local Subspace Classifier, it is natural not to modify only one prototype vector but to extended the modification to all the D_j prototype vectors involved in the classification. Due to the proportional nature of the delta rule, the hyperplanes determined by the prototypes are moved in a fashion that preserves the directions of the hyperplanes. Therefore, the prototype hyperplane after the delta movements is parallel to the original hyperplane. Figure 5 illustrates the movements of the prototype vectors in a simple two-dimensional $D = 1$ case. In the center image, both the original and modified vector locations are shown. Due to the similar triangles, the lines joining the prototypes are parallel and the directions of the residual vectors $\tilde{x}_A$ and $\tilde{x}_B$ is not altered during the application of the delta rule. As the result of the prototype movements in Figure 5, the classification of the input vector x is changed. This is indicated by the shift of the dotted classification border line over the x vector.

When a Self-Organizing Map is trained, a set of vectors located in the neighborhood of the best matching unit are moved towards the training vector. This movement is compatible with the LSC principle, provided that all the vectors are moved with an equal $\alpha(t)$ coefficient in the delta adaptation rule. The locating of the best matching unit, on the contrary, cannot as such be combined with the LSC. Therefore, a modification of the SOM incorporating a selection rule for the best matching unit compatible with the LSC is here introduced and given the name *Local Subspace SOM* (LSSOM). In the basic SOM, each map unit is individually matched against the input vector, and the neighborhood is formed around the winning neuron thereafter. In the LSSOM, linear manifolds are formed using the vectors in the neighborhoods of *all* map units. The best matching

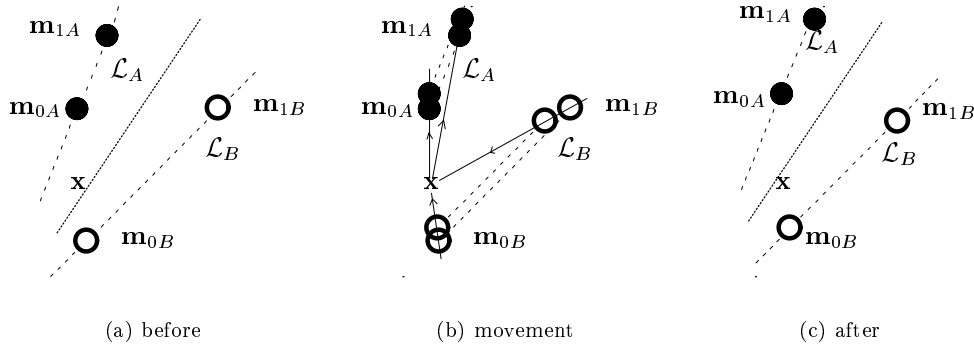


Figure 5: Delta rule in the LSC training. The locations of the prototype vectors are shown before, during and after the application of the delta rule. The movement of the dotted line indicates that the classification of $\mathbf{x}$ is changed from 'A' to 'B' in the process.

manifold is sought for as the one with the shortest distance to the input vector similarly to the classification rule (4). Each unit of the LSSOM therefore participates in many local subspaces.

As in SOM, the prototype vectors $\mathbf{m}_{ij}$ of a LSSOM are in the training moved iteratively towards the input vectors. The radius of the neighborhood in LSSOM is decreased during the training just like in the normal SOM. The size D of the neighborhood is naturally smaller at the borders and corners of the map than in the inner parts. The principle of the LSC+ may also be optionally applied with the LSSOM principle. As the result of the Local Subspace SOM, an elastic net is formed from the nodes of the SOM. The difference to the basic SOM is that the space between the nodes is in the LSSOM included in the locally linear manifolds. Therefore, there are no “holes” or “jumps” in the mapping between the input space and the map’s grid space.

5 Experiments

The experiment data was the same as that used in [3]. 17880 binary handwritten digits were normalized to the size of 32×32 pixels. One half of the images were used in training set cross-validation and the other half in testing. Karhunen-Loève transformation was calculated from the training set and applied to the 1024-dimensional images. As the result, feature vectors with the maximum dimensionality of 64 were obtained. 10-fold training set cross-validation was used to solve the optimal values for the feature vector dimensionality, d , and the dimensionality of the linear manifolds, D , common to all the classes. The two classification results attained with the testing set are shown in Table 1 together with three results from [3]. It can be seen that the LSC+ classifier is clearly superior in classification accuracy to all the other techniques.

classifier	errors %	parameters
LSC	2.5	$d = 64, D = 12$
LSC+	2.1	$d = 64, D = 23$
3-NN	3.8	$d = 38$
ALSM	3.1	$d = 64, D = 29$
LLR	2.8	$d = 36$

Table 1: Test set classification error percentages of the LSC and LSC+ classifiers together with the Nearest Neighbor rule (3-NN), Averaged Learning Subspace Method (ALSM), and the Local Linear Regression (LLR) classifier.

6 Discussion

In this paper, a new family of classification techniques has been introduced. It has been argued, that the Local Subspace Classifier (LSC) and the Convex Local Subspace Classifier (LSC+) techniques combine the benefits of the subspace and prototype based classification methods. An experimental evaluation of the two proposed classification algorithms with handwritten digit data has supported this argument.

As can be seen in Table 1, the optimal value of the manifold dimension parameter D is clearly larger in the LSC+ than in the LSC which does not demand the convexity of the subspace projection. At the same time, the classification accuracy is significantly better in the former. These two facts can be interpreted together as follows: The increased value of D provides the classifier with more prototypes for each single classification task. Due to the convexity requirement, only the most informative subset of these prototypes are actually used in the difficult regions of classification where the class distributions are sparse and overlapping.

Compared to prototype based classifiers, such as 3-NN and LLR in Table 1, both the LSC and LSC+ techniques seem to benefit from large feature vector dimensionality d . The LSC principle is thus less sensitive to the effects of the *curse of dimensionality* [1]. In this sense, the LSC techniques thus resemble the subspace methods of classification, which, in general, are able to take advantage of large vector dimensionalities.

The explicit use of the Gram-Schmidt orthonormalization process and the matrix pseudo-inversion (6) can be avoided by using an iterative algorithm for the solution of $\hat{\mathbf{x}}_j$. In addition, the LSC technique can be given a neural interpretation. These two topics are covered in [5].

The Local Subspace SOM (LSSOM) has been presented as a modification of the Self-Organizing Map algorithm. This modification is compatible with the LSC principle in the way the best matching unit is located on the map. It is possible to utilize the LSSOM technique in any application where the traditional SOM has been used. The computational complexity of the LSSOM is somewhat larger than that of the SOM. Due to this increase in computation, the LSSOM is able to better interpolate the input vectors with the same number units in the map.

References

- [1] R. Bellman. *Adaptive control processes: a guided tour*. Princeton University Press, 1961.
- [2] G. H. Golub and C. F. van Loan. *Matrix computations, 2nd ed.* Johns Hopkins University Press, 1989.
- [3] L. Holmström, P. Koistinen, J. Laaksonen, and E. Oja. Neural and statistical classifiers – taxonomy and two case studies. *IEEE Transactions on Neural Networks*, 8(1):5–17, Jan. 1997.
- [4] T. Kohonen. *Self-Organizing Maps*. Springer Series in Information Sciences 30. Springer-Verlag, 1995.
- [5] J. Laaksonen. *Subspace Methods in Recognition of Handwritten Digits*. PhD thesis, Helsinki University of Technology, 1997. To appear.
- [6] J. Laaksonen and E. Oja. Classification with learning k-nearest neighbors. In *Proceedings of the International Conference on Neural Networks*, volume 3, pages 1480–1483, Washington D.C., June 1996.