Regularizing the covariance matrix using spatial information

David M.J. Tax [a]

[a]Information and Communication Theory Group
Electrical Engineering, Mathematics and Computer Science,
Delft University of Technology
Mekelweg 4, 2628 CD Delft, The Netherlands
e-mail: D.M.J.Tax@ewi.tudelft.nl

Abstract

Learning algorithms can only perform well when the model is trained using sufficient number of training examples with respect to the complexity of the model. To obtain good generalization performance with a limited training data set, it is essential that prior knowledge of the problem is included in the representation of the objects or in the model of the data. Here we will consider image data and we propose to explicitly include the spatial connectivity of pixels in image data into the (estimated) covariance matrix of the data. This spatial regularization biases the model to solutions where remote pixels are uncorrelated. This adjusted covariance matrix can then be used in a supervised classification setting, or in unsupervised clustering or PCA. Examples for classification and feature extraction on image data are given.

1 Introduction

In pattern recognition and machine learning, one tries to fit a model to a limited data set. But in order to avoid overfitting on a limited training sample, the data model cannot be too complex [5]. Unfortunately, if the data distribution is complex, a simple model will not suffice. When just a few data examples are available, the structure of the problem cannot be reliably extracted and poor generalization will be obtained. In these cases prior knowledge on the problem should be included, either in the representation or in the model (or both). Obvious approaches are to choose a small set of informative features [10], constructing classifiers which exploit an assumed structure in the data or define suitable similarities or distances between objects [9, 8, 14]. Another approach is to simplify a complex classifier by applying regularization. An example is to regularize the (estimated) covariance matrix for the Normal-based linear discriminant [1].

Because many pattern recognition problems deal with image data, many procedures for image classification have been proposed. Images form an interesting challenge, because images consist of a large number of pixels, but their values cannot be expected to vary independently: pixels are heavily correlated. When we

consider images from a class of objects and we represent them as feature vectors of fixed size (say 256 features for 16×16 images), these images will be distributed in a (generally non-linear) subspace in the pixel space. The different directions in the subspace are due to scaling, rotation and translation of the object through the image (see for instance [16, 6, 15]).

This non-linear subspace can be estimated when sufficient examples are available. When this is not the case, artificial examples can be generated by applying the a priori assumed invariance transformations. This requires that we are able to define and apply all these possible transformations (see also the discussion in [14]. In particular when non-rigid object transformations are considered (i.e. other than translations, rotations and scalings), the number of possibilities is enormous.

To avoid this problem, we propose to define a more general spatial regularization for image data. We assume that neighboring pixels (in the spatial domain) are (positively) correlated, but that remote pixels will be largely uncorrelated. It means that close pixels will have their correlation increased, while for separate pixels the correlations will vanish. This correlation structure is used as a regularizer for the (estimated) covariance matrix. Note that only when images are carefully scaled and aligned, correlations over longer distances can be expected (in for instance face images [12]). Here we assume that it is generally not the case, and we will therefore regularize this when it is encountered in the covariance matrix estimated on a small sample.

Note that there is a fundamental difference between this approach and the modeling of the scale and rotation invariance of objects in images. When a covariance matrix is constructed to incorporate the invariances of a specific class, in principle, this matrix cannot be used for another class, because for different classes different pixels will be used in the rotation or scaling of the objects. Clearly, when these invariances per class can be estimated well, the model will have superior performance on these specific data. But when just a few objects are available, these class-specific invariances cannot be estimated reliably, and the performance will suffer. In that case a more general regularizer might be preferred. In the rest of the paper the use of this regularization for image data in dimensionality reduction and classification.

2 Defining the spatial regularization

Assume that we have a set of objects, represented by feature vectors $\mathcal{X}^{tr} = \{\mathbf{x}_i, i = 1, .., N\}$, $\mathbf{x} \in \mathbb{R}^p$. To characterize the correlation structure in the data, the covariance matrix can be estimated:

$$\hat{\Sigma} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}_i - \hat{\boldsymbol{\mu}})^T (\mathbf{x}_i - \hat{\boldsymbol{\mu}}), \tag{1}$$

where $\hat{\boldsymbol{\mu}} = \sum_i \mathbf{x}_i / N$ is the mean of data $\mathcal{X}^{tr}$. Entry $\hat{\Sigma}_{ij}$ of this matrix indicates how feature i is correlated with feature j, given these data $\mathcal{X}^{tr}$.[1]

[1] Note that the entry Σ_{ij} is not directly the correlation coefficient ρ_{ij}, but is actually scaled by the standard deviations σ_i and σ_j of the individual features: $\Sigma_{ij} = \sigma_i \sigma_j \rho_{ij}$. When the data

As argued before, when just a small sample is available, the entries of $\hat{\Sigma}$ cannot be estimated reliably. It is very likely that nonzero entries appear on places where the true correlations should vanish. The standard way of regularizing a covariance matrix, is by adding a constant to the diagonal:

$$\hat{\Sigma}_\lambda = \hat{\Sigma} + \lambda \mathcal{I}, \tag{2}$$

where $\mathcal{I}$ is the $p \times p$ identity matrix. The regularization parameter λ has to be optimized by the user. Although this regularization can always be applied, it does not take the structure of the data into account.

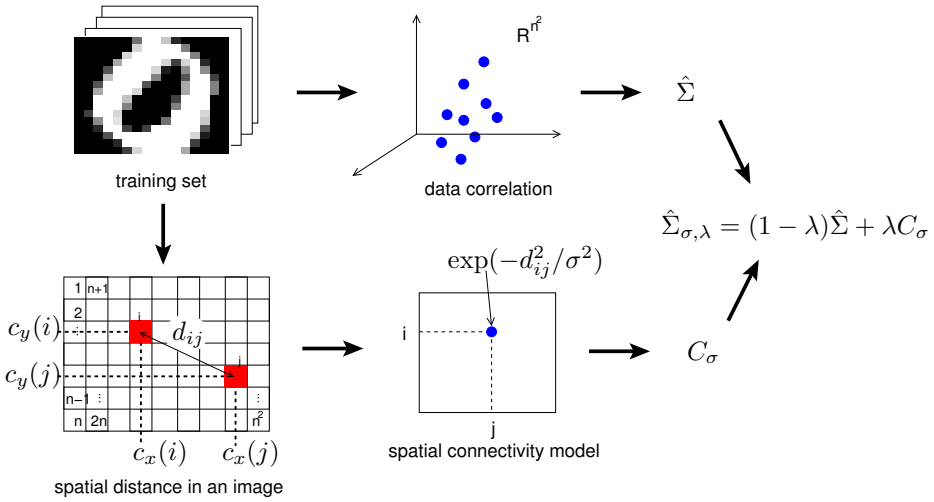

Figure 1: Graphical illustration of the two tracks of obtaining the data covariance matrix and the spatial regularization.

Now we consider images of a fixed size (for instance of size 16×16), and represent them as feature vectors (in this case 256-dimensional). This is depicted as the first step in the upper half of Figure 1. On these data we can estimate the data covariance matrix and we obtain $\hat{\Sigma}$. We now define the spatial regularization matrix C_σ, which includes the prior knowledge what pixels which are close in the spatial domain are expected to be highly correlated. We compute the *spatial distance* between each pair of pixels. We index each pixel in the image by a number between 1 and p (where in Figure 1 we considered square images with $p = n^2$). The Euclidean distance between pixels i and j is computed by considering their x and y coordinates in the image. This is shown in the lower half of Figure 1. We define the functions $c_x(i)$ and $c_y(i)$ which give the x and y coordinate of pixel number i. The distance d_{ij} is now computed by: $d_{ij} = |c_x(i) - c_x(j)|^2 + |c_y(i) - c_y(j)|^2$. The

are rescaled to have unit variance in all feature directions, $\sigma_i = \sigma_j = 1$, this distinction vanishes.

spatial regularization matrix is now defined as:

$$C_\sigma(i,j) = \exp\left(-\frac{d_{ij}}{\sigma^2}\right). \tag{3}$$

A hyper-parameter σ is introduced to give the scale at which the spatial connectivity is expected. $C_\sigma(i,j)$ will be high for pixels which are neighbors in the image, but will be low for pixels which are far apart. How fast $C_\sigma(i,j)$ will decrease with increasing distance d_{ij} is determined by σ.

The first approach to regularize the data covariance matrix (1) is by $\tilde{\Sigma} = C_\sigma \odot \hat{\Sigma}$, where $A \odot B$ indicates the element wise product (or Hadamard product) of the matrices A and B [13]. This procedure only suppresses spurious high covariances in $\hat{\Sigma}$. When the training set shows small (or zero) correlation between neighboring pixels, it will not increase this correlation. Therfore, an alternative approach is taken. Here the estimated covariance matrix is regularized by averaging with the spatial regularization matrix:

$$\hat{\Sigma}_{\sigma,\lambda} = (1-\lambda)\hat{\Sigma} + \lambda C_\sigma, \tag{4}$$

where λ is the tradeoff parameter between the training data covariances and the spatial connectivity model. This introduces an extra regularization parameter λ, but the advantage is that it can overcome cases where training data fail to show correlation between pixels, while there may be some. This regularization actively forces the covariances to become positive for neighboring pixels, and zero for remote pixels. In order to have λ in a reasonable scale between 0 and 1, the data is rescaled to have unit variance for all features. In that case the entries in $\hat{\Sigma}$ and C_σ have comparable values. We will therefore rescale all the features in the coming experiments to have unit variance.

Note that matrix C_σ in Equation (3) may not be positive definite. Fortunately, it appears that when the negative eigenvalues appeared, they stayed small, and because we are mainly interested in the largest eigenvectors, we will ignore this point further in this paper.

3 Experiments

In the coming sections we will show how the spatial regularizer C_σ improves the performance for dimensionality reduction and classification. This is compared to models without regularization, or with a standard regularization ($\mathcal{I}$ as given in Equation (2)). In all cases, the models can be trained using very small sample size, while retaining good generalization performance on new images.

Throughout the experiments we will use the NIST handwritten digits. The NIST dataset contains 16×16 images, with 200 digits per class. The dataset used in the experiments was taken from the Special Database 3 distributed on CD-ROM by the U.S. NIST, the National Institute for Standards and Technology. Currently, this database is discontinued; it is now distributed together with Database 7 as Database 19 (see `http://www.nist.gov/srd/spec19.htm`). The pre-processing used is described in [2].

3.1 Principal Component Analysis

Principal Component Analysis is one of the most well known and applied methods for feature reduction. It reduces the number of features by projecting the data onto these directions in which the variance of the original data is the highest, the principal components of the data [11]. The principal components can be derived by computing the eigenvectors of the (estimated) data covariance matrix.

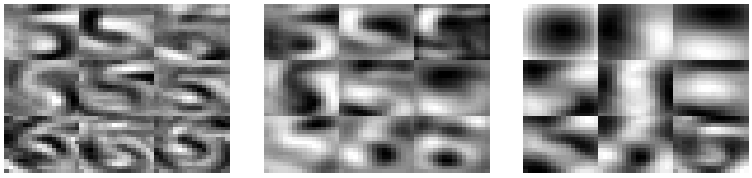

Figure 2: The first nine principal components of PCA applied on class '5' of the NIST handwritten digits (ordered according the eigenvalue). In the left picture the estimated covariance matrix $\hat{\Sigma}$ is used, in the middle the covariance matrix $\hat{\Sigma}_{\sigma,\lambda}$, with $\lambda = 0.75$ and $\sigma = 1.4$ and in the right $\hat{\Sigma}_{\sigma,\lambda}$, with $\lambda = 0.75$ and $\sigma = 4$.

If a principal component is derived from image data, this component can be interpreted as an image (see for an illustrative example [17]). In Figure 2 the principal components are visualized for the class of digits '5'. In total 200 images of 16×16 are used, where each of the features (pixels) were rescaled to unit variance. The left picture shows the first nine (standard) PCA components, using the estimated covariance matrix. Some class invariances are captured here, like the second component which encodes for moving the middle horizontal line up and downwards. The higher components show noisy behavior. Due to the relative small sample size, spurious correlations are found, which is reflected in the noisy principal component image[2]. The middle picture shows the first nine PCA components using the $\hat{\Sigma}_{\sigma,\lambda}, \lambda = 0.75, \sigma = 1.4$. By the relatively small σ only 'local' features are described. The first two components describe the translation of the left and right extremes of the fives. Larger structures are described when the σ is increased, as it is shown in the right picture. Here, $\sigma = 4$ is applied, and parts of the original digits can be clearly distinguished, similar to the results obtained by using $\hat{\Sigma}$. Although the components may look like independent components from an ICA [7], it is actually almost impossible to get these independent components using this small sample size. As in the PCA, the images have to be significantly smoothed, or many artificial examples have to be added to get a similar result.

3.2 Classification problems

To show the performance on a classification task, we consider pairs of classes from the NIST digit dataset. We model each of the classes by a single Gaussian, in

[2]Note that this effect can be suppressed when the original images are smoothed. But when a limited sample is used, it will never completely vanish.

the hope that it captures the correlation structure of each of the classes, including slight rotations, scalings and rotations. For increasing number of training objects per class, we trained a standard linear and quadratic Normal-density based classifier[4] (called LD and QD respectively). When the covariance matrices are not regularized, the classifiers cannot be computed. In order to get good performance, the following regularization scheme is used:

$$\hat{\Sigma}_{\lambda,\gamma} = (1-\lambda-\gamma)\hat{\Sigma} + \lambda \mathrm{diag}(\hat{\Sigma}_{ii}) + \gamma \mathrm{diag}(\hat{\Sigma})^T \mathbf{1}/n, \tag{5}$$

where $\mathrm{diag}(\hat{\Sigma}_{ii})$ is a diagonal matrix containing the diagonal elements of $\hat{\Sigma}$, and $\mathrm{diag}(\hat{\Sigma})^T \mathbf{1}/n$ is the average of the diagonal elements. Both regularization parameters λ and γ have to be optimized.

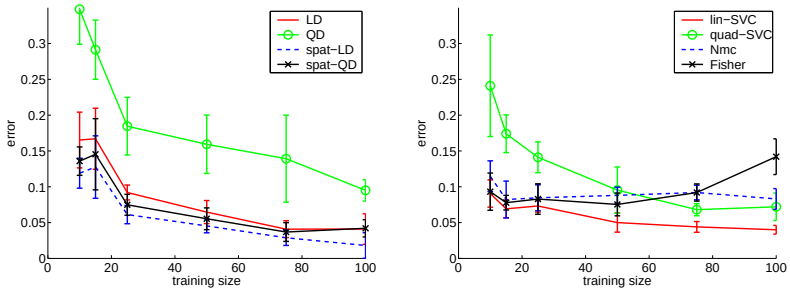

Figure 3: Learning curves for classifiers trained on classes 3 and 8. (left) LD, QD, spatial-regularized LD, spatial-regularized QD. (right) Learning curves for the linear and quadratic support vector classifier, Nearest mean classifier and the Fisher classifier.

For a fair comparison, the regularization parameters are optimized using a cross-validation procedure on the training set (three times 10-fold). The parameters are often around the values $\lambda = 0.3$ and $\gamma = 1e^{-9}$. These classifiers are compared with the spatially regularized classifiers, using $\sigma = 1$. With the same approach (three times 10-fold crossvalidation) the λ parameter in (4) is optimized. The errors are estimated in another crossvalidation loop (again three times 10-fold). The learning curves show typical behavior for all the classes (except for the classes which are very well separable and which are not shown here). In Figure 3 the results are shown for classes '3' and '8' (which show some overlap in the feature space). The left subplot shows the performance of the LD, QD and their regularized versions. The results on some other two-class problems (between digits '1' and '4' and between '7' and '9') are shown in Figure 4. We see that the quadratic discriminant QD never performs very well; even with optimized regularization it is very hard to find something. The LD performs reasonable, but it requires very careful optimization of the two parameters. The regularized QD and LD work well, even when a fixed $\sigma = 1$ is used. Performance can be improved slightly when σ is optimized further. In almost all cases the spatially regularized linear classifier outperforms the quadratic one, except for the distinction between '3' and '8'. Here the spatial regularization, combined with the relative high variance of the orientation of the digits, 'closes' the left sides of the '3' digits, such they appear more

like an '8' digit. The normal LD performs better or comparable to the spatially regularized classifiers.

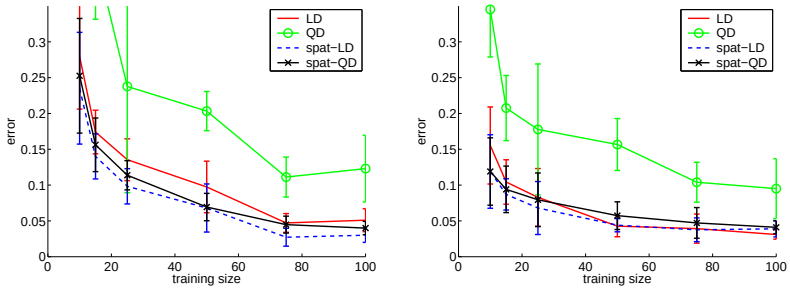

Figure 4: Learning curves for four classifiers on separating classes 1 and 4 (left) and separating classes 7 and 9 (right).

The performance of some other classifiers, the Support Vector classifier [18] with a polynomial kernel of degree 1 and 2 (called lin-SVC and quad-SVC, respectively), a nearest mean classifier and the Fisher classifier, are shown in the right subplot of Figure 3. The results of the spat-LD and spat-QD are competitive with the performance of the linear support vector classifiers, for which it is known that it performs very well on small sample size problems.

4 Conclusions

In this paper we introduced a regularization for the covariance matrix of a class of objects representing images. It forces pixels that are far apart in the image to have vanishing correlations. Neighboring pixels, on the other hand, will be highly (positively) correlated. This regularized covariance matrix can be applied in any method where image data is characterized by models which utilize estimated covariance matrices in the pixel feature space. This can be, for instance, in the density estimation of a class, a cluster analysis of a dataset, in feature reduction using PCA or in classifiers assuming Gaussian distributions (both in multiclass classifiers as in outlier detection). This paper showed that all these methods gain some generalization performance by the introduced bias, in particular when just a few training objects are available. For larger sample sizes, this correlation structure might be derived from the training data itself, and the performance increase might be lower. This regularization can not only be useful for image data, but can also be applied to other data where some correlation structure can be assumed. This can be for instance time series data, where neighboring time points are correlated.

Acknowledgments This work was partly supported by the Dutch Organization for Scientific Research (NWO).

References

[1] C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Walton Street, Oxford OX2 6DP, 1995.

[2] D. de Ridder. Shared weights neural networks in image analysis. Master's thesis, Technische Universiteit Delft, 1996.

[3] N. M. Dempster, A.P. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Statist. Soc. B*, 39:185–197, 1977.

[4] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley & Sons, second edition, 2001.

[5] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 1992.

[6] G.E. Hinton, P. Dayan, and M. Revow. Modeling the manifolds of images of handwritten digits. *IEEE Transactions on Neural Networks*, 8:65–74, 1997.

[7] A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13:411–430, 2000.

[8] D.W. Jacobs, D. Weinshall, and Y. Gdalyahu. Classification with non-metric distances: Image retrieval and class representation. *IEEE Trans. on PAMI*, 22(6):583–600, 2000.

[9] A.K. Jain and D. Zongker. Representation and recognition of handwritten digits using deformable templates. *IEEE Trans. on PAMI*, 19(12):1386–1391, 1997.

[10] A.K. Jain and D.E. Zongker. Feature selection: evaluation, application, and small sample performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):153–158, February 1997.

[11] I.T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 1986.

[12] M. Kirby and L. Sirovich. Application of the Karhunen-Loève procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):103–108, Jan 1990.

[13] H. Lütkepohl. *Handbook of matrices*. John Wiley & Sons, 1996.

[14] B. Schölkopf. *Support Vector Learning*. PhD thesis, Technischen Universität Berlin, 1997.

[15] B. Schölkopf, P.Y. Simard, A.J. Smola, and V.N. Vapnik. Prior knowledge in support vector kernels. In M.I. Jordan, M.J. Kearns, and S.A. Solla, editors, *Advances in Neural Information Processings Systems*, volume 10, pages 640–646, Cambridge, MA, 1998. MIT Press.

[16] P.Y. Simard, Y.A. LeCun, and J.S. Denker. Efficient pattern recognition using a new transformation distance. In S.J. Hanson, J.D. Cowan, and C.L. Giles, editors, *Advances in Neural Information Processing Systems*, volume 5, pages 50–58, San Mateo, CA, 1993. Morgan Kaufmann.

[17] M.A. Turk and A.P. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–96, 1991.

[18] V.N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.