Missing Data in Kernel PCA

Guido Sanguinetti and Neil D. Lawrence

Department of Computer Science, University of Sheffield
211 Portobello Street, Sheffield S1 4DP, U.K.
{guido, neil}@dcs.shef.ac.uk

Abstract. Kernel Principal Component Analysis (KPCA) is a widely used technique for visualisation and feature extraction. Despite its success and flexibility, the lack of a probabilistic interpretation means that some problems, such as handling missing or corrupted data, are very hard to deal with. In this paper we exploit the probabilistic interpretation of linear PCA together with recent results on latent variable models in Gaussian Processes in order to introduce an objective function for KPCA. This in turn allows a principled approach to the missing data problem. Furthermore, this new approach can be extended to reconstruct corrupted test data using fixed kernel feature extractors. The experimental results show strong improvements over widely used heuristics.

1 Introduction

Kernel PCA is a non-linear feature selection technique which extends the linear statistical method of Principal Component Analysis (PCA) by elegantly using the so called *kernel trick* [1]. However, while the flexibility of Kernel PCA has led to very good performance on a number of problems, the lack of a probabilistic interpretation for it means that it can be very difficult to adapt it in the presence of missing or corrupted data.

In this paper we suggest a simple way of estimating missing data in Kernel PCA. We start by reformulating Kernel PCA along the lines suggested in [2][3], we then show how the derived objective function can be used in the face of missing data. We demonstrate the resulting approach on two widely used data sets: the *Tobamovirus* data set used in [4] and [5] (where a missing data comparision was also made) and the oil flow data set used in [6]. We compare our results with other possible approaches: the crude but widely used heuristic of replacing a missing value with the mean of the corresponding component across the data set, a nearest neighbour approach and a reconstruction using linear probabilistic PCA. Both the reconstruction error and the visualisation improve dramatically through our approach.

We also consider the related problem of reconstructing missing test data: assuming we have trained a Kernel PCA feature extractor, what is the best guess for a data point with partially missing data? Our approach turns out to produce a very reasonable solution to this problem, providing again dramatic improvements in visualisation and reconstruction error.

J. Fürnkranz, T. Scheffer, and M. Spiliopoulou (Eds.): ECML 2006, LNAI 4212, pp. 751–758, 2006.

The remainder of the paper is organised as follows: we start by briefly reviewing the probabilistic interpretation of PCA (PPCA, [5]) and its dual formulation. We then show how a kernel version of dual PPCA leads naturally to an objective function for KPCA and discuss how to use this information to deal with missing data. In the third section, we present our experimental results. In the fourth section we turn to the somewhat complementary problem of estimating missing data in test data. We finally conclude by discussing the merits and limits of our approach.

2 Cross Entropy and Reconstructing Missing Data

The key idea in PCA is to identify the directions of maximal variance in a data set. This can be shown to be equivalent to an eigenvalue problem for the empirical covariance matrix constructed from the data. Probabilistic PCA [5] assumes a linear relationship between the observed variables $\mathbf{y}_i$ and a latent variable $\mathbf{x}_i$,

$$\mathbf{y}_i = \mathbf{W}\mathbf{x}_i + \boldsymbol{\epsilon}, \tag{1}$$

where $\mathbf{W}$ is a $d \times q$ matrix (d being the dimension of the observed variable and q that of the latent variables) and $\boldsymbol{\epsilon}$ is an error term assumed to be Gaussian distributed with spherical covariance, $\boldsymbol{\epsilon} \sim N\left(\mathbf{0}, \sigma^2\mathbf{I}\right)$. For dimensonal reduction we have $d > q$. Equation 1 then implies a Gaussian likelihood for the observed variable,

$$\mathbf{y}_i \sim N\left(\mathbf{W}\mathbf{x}_i, \sigma^2\mathbf{I}\right). \tag{2}$$

Placing a Gaussian prior on the latent variables $\mathbf{x}$ leads to the marginal likelihood

$$\mathbf{y}^{(j)} \sim N\left(\mathbf{0}, \mathbf{W}\mathbf{W}^{\mathrm{T}} + \sigma^2\mathbf{I}\right). \tag{3}$$

It can be proved that the maximum of the marginal likelihood is achieved when the columns of $\mathbf{W}$ span the directions of maximal variance in the data.

This picture can be reversed leading to the dual approach to probabilistic PCA taken in [2][3]. We place a prior distribution on $\mathbf{W}$ in which each element of $\mathbf{W}$ is Gaussian distributed, $w_{ij} \sim N(0, 1)$, the likelihood of equation 2 can be marginalised with respect to $\mathbf{W}$ to yield a marginal likelihood for the data set of the form

$$\mathbf{y}^{(j)} \sim N\left(\mathbf{0}, \mathbf{X}\mathbf{X}^{\mathrm{T}} + \sigma^2\mathbf{I}\right), \tag{4}$$

where $\mathbf{y}^{(j)}$ is the jth column of $\mathbf{Y}$ and each column is independent. Maximum likelihood estimation with respect to the embeddings, $\mathbf{X}$, leads to an eigenvalue problem for the inner product matrix $\mathbf{K} = \frac{1}{d}\mathbf{Y}\mathbf{Y}^{\mathrm{T}}$, which is well known to be mathematically equivalent to the eigenvalue problem for the empirical covariance matrix.

The likelihood for both PPCA and dual PPCA can be given an interesting interpretation as the *cross entropy* between two Gaussian distributions, one

Algorithm 1. The Missing Data reconstruction algorithm

Initialise the missing data;
Select the dimension of the latent space q;
repeat
  Compute the kernel matrix $\mathbf{K}$;
  Compute the approximating matrix $\mathbf{C} = \mathbf{X}\mathbf{X}^{\mathrm{T}} + \sigma^2 I$ by computing the principal components of $\mathbf{K}$;
  Minimise the cross entropy between $\mathbf{K}$ and $\mathbf{C}$ with respect to the missing data;
until convergence

specified by the empirical covariance $\mathbf{S}$ and the other by the approximating covariance $\Sigma = \mathbf{W}\mathbf{W}^T + \sigma^2 I$ in the case of PPCA and $\mathbf{C} = \mathbf{X}\mathbf{X}^T + \sigma^2 I$ in the case of dual PPCA. This is given, up to an additive constant, by the formula

$$\mathcal{L}\left(N\left(\mathbf{0}, \mathbf{C}\right) || N\left(\mathbf{0}, \mathbf{K}\right)\right) = -\frac{1}{2}\left(\log|\mathbf{C}| + \operatorname{trace}\left(\mathbf{K}\mathbf{C}^{-1}\right)\right). \tag{5}$$

We note in passing that, when $N > q$, $\mathbf{K}$ will not be positive definite, however this situation can be rectified without significant effect on the algorithm by adding a spherical term to $\mathbf{K}$ (see [7]).

Kernel PCA can be viewed as dual PCA on the images of the data set in a (possibly infinite dimensional) feature space. As the inner product matrix in (4) scales with the number of data points and not with their dimensionality, the computational burden will remain unchanged by pre-applying a feature map. Using the kernel trick, we have that the inner product matrix of the images of the data via the feature map is given by the kernel matrix $K\left(\mathbf{x}_i, \mathbf{x}_j\right)$, whose spectral decomposition provides the nonlinear feature extractors.

Therefore, it is natural to consider the cross entropy of equation (5) as an objective function for Kernel PCA. The implicit idea behind this is that nonlinear data in the observed space can be mapped, through the feature map, to a high dimensional space where the implied generative structure becomes approximately Gaussian[1]. While we are not aware of a general proof of this fact, there has been experimental evidence supporting it (see e.g. [8]).

Having obtained an objective function for Kernel PCA, we are in a position to give principled answers to a number of problems. In particular, this suggests a method for dealing with missing or corrupted data: the objective function can be optimised with respect to both the images and the values of the missing points (which are particular elements of $\mathbf{Y}$).

We chose to take an iterative approach to the optimisation, using spectral decomposition to compute principal components and a scaled conjugate gradient algorithm to optimise with respect to the missing points. This is summed up schematically in Algorithm 1.

[1] More precisely, the generative structure becomes approximately Gaussian after projection onto a suitable finite dimensional space.

3 Experimental Results

To test our approach we tried our algorithm on two well known Tobamovirus data set. This was used in [4] to demonstrate PCA and further used in [5] to demonstrate PPCA in the presence of missing data. It consists of 38 data points, each of them 18 dimensional. In our experiment we removed at random 130 values by sampling from a uniform distribution. To capture 95% of the initial variability we selected a latent dimension, q, of 8. We used an MLP kernel with weight variance and bias both equal to 10 [9]. Further experimental results are reported in [10].

Figure 1 (a-c) compares the reconstruction obtained with our method (b) with the underlying truth (KPCA on the full data set,(c)) and with the widely used heuristic of replacing missing components with the mean across the data set (a). The improvement in visualisation is dramatic.

To quantify the effectiveness of our algorithm, we repeated the experiment with ten different probabilities (from 0.05 to 0.5) and for ten different random seeds. To measure the quality of the reconstruction, we estimated the squared reconstruction error (given that we know the true positions of the points). We compared our results with three different methods: the widely used heuristic of the mean as above, a 1 nearest neighbour (1NN)[2] method which replaces the missing values with the values of the point with the nearest values in the known features, and missing point estimation for linear probabilistic PCA (initialised with the mean). The results for the Tobamovirus data set are summarised in Figure 1 (d), plotting the deletion probabilities on the x-axis versus the reconstruction error. The solid line is the mean initialisation, the dotted line is the reconstruction using our method, the dashed line shows the reconstruction errors using PPCA and the dotted and dashed line shows the reconstruction using 1NN (notice that 1NN is viable only up to deletion probabilities of 0.15).

4 Reconstructing Corrupted Test Data

Having introduced an objective function for Kernel PCA, the next question is the following: suppose we have trained a KPCA feature extractor on some training data set. If we are given a test point, we can use our feature extractors on it. Suppose though the test data has some missing components, can we use the knowledge of the feature extractors to deduce something about the missing data? We are assuming that the test point comes from the same (unknown) generative distribution as the training set; also, we do not want to recompute the feature extractors anew (which would reduce us to the previous problem).

We can again draw inspiration by the linear picture; a trained PPCA feature extractor gives us a generative distribution for the data

$$\mathbf{y}|W, \sigma \sim N\left(\boldsymbol{\mu}, WW^T + \sigma^2 I\right). \tag{6}$$

[2] It could be argued that a more sensible choice would be to use k-nearest neighbours. However, when the deletion probability is high, it is impossible to find sufficient uncorrupted data points to make k-NN viable.

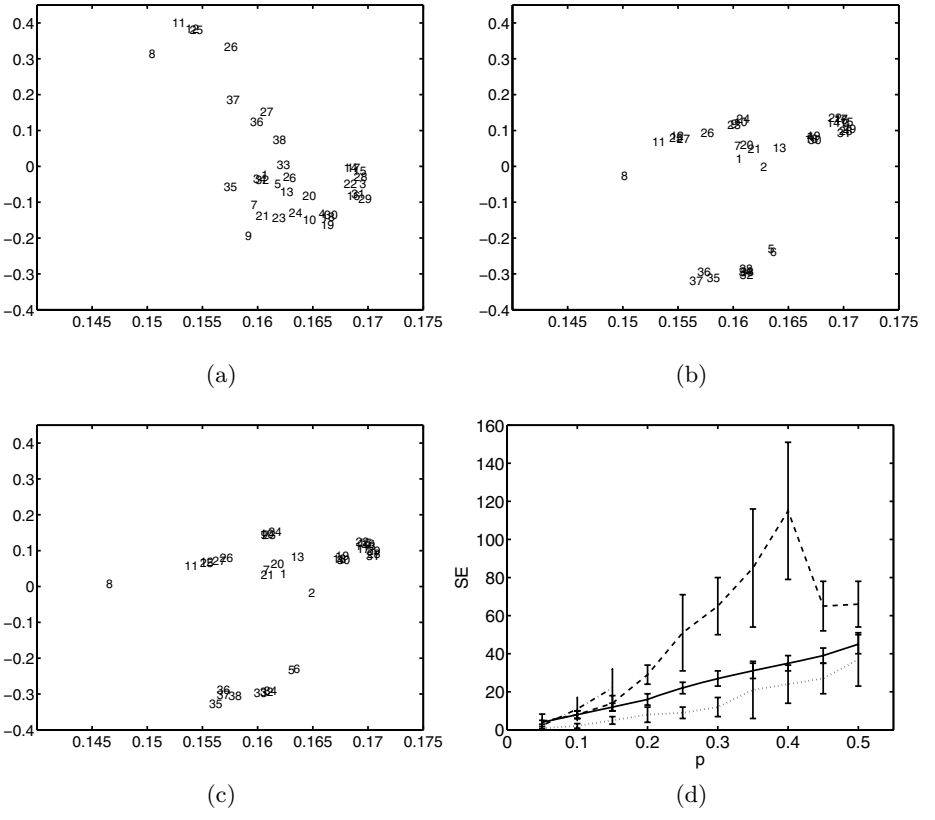

Fig. 1. KPCA with missing data. (a) shows the projection on the first two principal components of the initialisation with 20% of the values removed and initialised to the mean for the Tobamovirus data set. (b) shows the projection on the first two principal components of the optimal reconstruction of the missing data for the Tobamovirus data set. (c) shows KPCA on the Tobamovirus data set. (d) shows a comparison of the reconstruction squared errors using different methods for different deletion probabilities: the mean substitution (solid line), PPCA (dashed), 1 nearest neighbour (dotted and dashed) and our approach (dotted).

If we are given some of the entries in the test point $\mathbf{y}_t$, call them $\mathbf{y}_{t\text{Known}}$, the obvious best guess for the unknown entries would be given by the maximum of the conditional probability $p\left(\mathbf{y}_{t\text{notKnown}}|\mathbf{y}_{t\text{Known}}\right)$ (notice that this will also provide an estimate of the uncertainty on the guess).

Although it is in general impossible to estimate the conditional distribution (6) for Kernel PCA, we can still obtain a kernel version of the optimisation problem by looking back at PPCA from a geometric perspective. The maximum of the conditional probability is given by the minimum of the Mahalanobis distance of $\mathbf{y}_t$ from the mean $\boldsymbol{\mu}$, the Mahalanobis distance being measured with the inverse covariance matrix

$$C^{-1} = \left(WW^T + \sigma^2 I\right)^{-1}.$$

Therefore we can recover the maximum by optimising the quantity

$$\mathbf{y}_t^T C^{-1} \mathbf{y}_t = \sum_{i=1}^{q} \left(\lambda_i^{-1} - \sigma^{-2}\right) \left(\mathbf{y}_t \cdot \mathbf{u}_i\right)^2 + \sigma^{-2} \|\mathbf{y}_t\|^2 \tag{7}$$

where q is the number of principal components included in the model, $\mathbf{u}_i$ are the principal eigenvectors and λ_i are the corresponding eigenvalues.

As equation (7) makes clear, this distance can be expressed uniquely in terms of dot products of the test point with the principal components (and with itself), hence it is readily transferred to the kernel situation. In the RBF case, there is the further advantage that $k(\mathbf{y}, \mathbf{y}) = 1 \ \forall \mathbf{y}$ so that the second term in (7) needs not be included.

Recalling that the KPCA feature extractors in feature space are given by $\mathbf{u}_i = \sum_{j=1}^{N\text{train}} \alpha_j^i \boldsymbol{\Phi}(\mathbf{y}_j)$ where $\boldsymbol{\alpha}^i$ is the i-th eigenvector of the Gram matrix $k(\mathbf{y}_i, \mathbf{y}_j)$ (normalised so that $\lambda_i \left(\boldsymbol{\alpha}^i \cdot \boldsymbol{\alpha}^i\right) = 1$), we obtain the following objective function for a missing test point

$$\mathcal{L} = \sum_{i=1}^{q} \left(\lambda_i^{-1} - \sigma^{-2}\right) \left(\sum_{j=1}^{N\text{train}} \alpha_j^i k(\mathbf{y}_j, \mathbf{y}_t) \right)^2 . \tag{8}$$

Notice that we need both the KPCA feature extractors and the off subspace variance σ^2 to formulate our optimisation problem, which can be obtained using our approach to Kernel PCA but not using the standard non-probabilistic formulation.

To test this approach we used the oil flow data set of [6]. This consists of 1000 12 dimensional synthetically generated data points modelling the flow of a mixture of oil, water and gas in a pipeline. The points are labelled in three different classes, according to the flow being laminar, annular or homogeneous. In this case we used an RBF kernel with inverse width 0.075. The results are shown in Figure 2. We selected the points corresponding to a laminar flow in the oil flow data set. We removed a point at random and performed KPCA on the remaining data set, retaining two principal components. We then treated the point we removed as a test point and artificially corrupted its first five coordinates by multiplying them by a constant factor. The point recovered through optimising the objective function (8) is very close indeed.

To quantify the efficacy of our method, we repeated the example of Figure 2 removing a different point at random fifty times and replacing its first five coordinates with random numbers. We also increased the number of features extracted from two to ten. The results are summarised in Table 1, where a comparison with the mean substitution and 1 nearest neighbour is made. Notice that the reconstruction error tends to decrease as the number of extracted features is increased, as well as the reconstruction becoming more consistent (smaller fluctuations in the mean error).

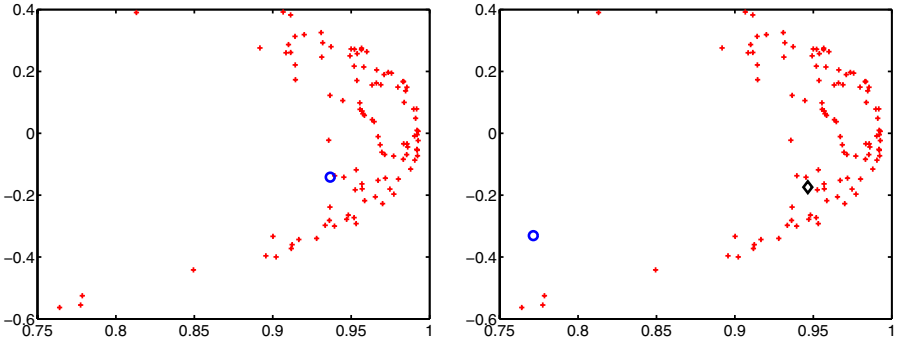

Fig. 2. Reconstructing test points with Kernel PCA:(a) training points (crosses) and original position of the test point (circle); (b) corrupted position of the test point (circle) and reconstructed position of the test point (diamond)

Table 1. Reconstructing corrupted test points using KPCA feature extractors. The first column shows the number of principal components retained, the second to fourth columns show the mean reconstruction error across 50 runs using our method, mean substitution and 1 nearest neighbour respectively. Notice that the reconstruction error using our method decreases as the number of principal components is increased; with more than three retained components our method gives the best performance.

Features extracted	KPCA	Mean	1NN
2	0.55±0.28	0.76±0.33	**0.29±0.20**
3	0.38±0.22	0.76±0.33	**0.29±0.20**
4	**0.28±0.17**	0.76±0.33	0.29±0.20
5	**0.24±0.16**	0.76±0.33	0.29±0.20

5 Discussion

In this paper we introduced an objective function for Kernel PCA, building on previous work on probabilistic PCA [5] and latent variable models in Gaussian Processes [2] [3]. This in turns allows to extend important inference techniques, such as the estimation of missing data, to the case where the features are nonlinear.

Experimental results on two benchmark data sets show that this approach yields far better results than the often recommended heuristic of replacing a missing value with the mean (which we used as our initialisation), and consistently outperforms other methods such as 1 NN and probabilistic PCA. Furthermore, the same ideas lead to a very natural solution of the related problem of estimated missing or corrupted components in test data.

Despite these positive results, our approach still falls short of providing a full probabilistic interpretation for Kernel PCA. The Gordian knot of the feature map has been severed by integrating out the nonlinear mapping. This comes

at the cost of no longer being able to predict the positions of new observed points from the latent ones. The link between the primal and the dual PCA problems in the kernelised case requires the explicit knowledge of the feature map. Similarly, the elegant interpretation in terms of probability distributions is harder to recover.

Acknowledgements. G.S. gratefully acknowledges support from a BBSRC award "Improved processing of microarray data using probabilistic models".

References

1. Schölkopf, B., Smola, A.J., Müller, K.R.: Kernel principal component analysis. In: Proceedings 1997 International Conference on Artificial Neural Networks, ICANN'97, Lausanne, Switzerland (1997) 583
2. Lawrence, N.D.: Gaussian process models for visualisation of high dimensional data. In Thrun, S., Saul, L., Schölkopf, B., eds.: Advances in Neural Information Processing Systems. Volume 16., Cambridge, MA, MIT Press (2004) 329–336
3. Lawrence, N.D.: Probabilistic non-linear principal component analysis with Gaussian process latent variable models. Journal of Machine Learning Research **6** (2005) 1783–1816
4. Ripley, B.D.: Pattern Recognition and Neural Networks. Cambridge University Press, Cambridge, U.K. (1996)
5. Tipping, M.E., Bishop, C.M.: Probabilistic principal component analysis. Journal of the Royal Statistical Society, B **6** (1999) 611–622
6. Bishop, C.M., Svensén, M., Williams, C.K.I.: GTM: the Generative Topographic Mapping. Neural Computation **10** (1998) 215–234
7. Lawrence, N.D., Sanguinetti, G.: Matching kernels through Kullback-Leibler divergence minimisation. Technical Report CS-04-12, The University of Sheffield, Department of Computer Science (2004)
8. Schölkopf, B., Smola, A.J.: Learning with Kernels. MIT Press (2001)
9. Williams, C.K.I.: Computing with infinite networks. In Mozer, M.C., Jordan, M.I., Petsche, T., eds.: Advances in Neural Information Processing Systems. Volume 9., Cambridge, MA, MIT Press (1997)
10. Sanguinetti, G., Lawrence, N.D.: Missing data in kernel PCA. Technical Report CS-06-08, The University of Sheffield, Department of Computer Science (2006)