Missing Data in Kernel PCA

Guido Sanguinetti, Neil D. Lawrence
Department of Computer Science, University of Sheffield
211 Portobello Street, Sheffield S1 4DP, U.K.

19th June 2006

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

Dimensional reduction is often the best starting point when handling high dimensional data. It is used for a variety of reasons, from visualisation to denoising, and in a variety of different applications, from signal processing to bioinformatics. One of the most widely used dimensional reduction tools is Principal Component Analysis (PCA). PCA is a well known statistical technique dating back to the first years of the last century (see *e.g.* [1]). PCA implicitly assumes that the data set under consideration is normally distributed, and selects the subspace which maximises the projected variance. The equivalent mathematical formulation is the following: we consider a centred data set, $\mathbf{Y} = [\mathbf{y}_1 \dots \mathbf{y}_N]^{\mathrm{T}} \in \Re^{N\times d}$, and construct the *sample covariance matrix*

$$\mathbf{S} = \frac{1}{N}\mathbf{Y}^{\mathrm{T}}\mathbf{Y} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{y}_i\mathbf{y}_i^{\mathrm{T}}.$$

Then q-dimensional PCA is equivalent to projecting onto the q-dimensional subspace spanned by the q eigenvectors of $\mathbf{S}$ with largest eigenvalues.

A traditional weakness of PCA was the lack of a probabilistic interpretation; however Tipping and Bishop found a probabilistic interpretation for PCA

(PPCA, [2]) by recasting the problem in terms of a latent variable model. The data is assumed to be generated from a lower dimensional embedded space with a spherical noise term

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

where the latent positions are given by $\mathbf{X} = [\mathbf{x}_1 \dots \mathbf{x}_N]^{\mathrm{T}}$. By placing a spherical Gaussian prior on $\mathbf{X}$ and marginalising we can obtain a likelihood for which an optimum value of $\mathbf{W}$ is given by the principal subspace of the data; the resulting model is therefore a probabilistic interpretation of PCA. This interpretation presents several advantages: it allows to deal effectively with missing data, it allows to use mixtures of PCA and allows us to develop Bayesian interpretations of the model [3, 4].

Both PCA and PPCA are linear models, and their usefulness is very limited in cases when the data set is inherently nonlinear. Kernel PCA presents an elegant solution to this problem [5]. It is well known that when a data analysis problem depends on the data only through scalar products it can be nonlinearised through the so called 'kernel trick'. This consists in mapping the data to a higher (possibly infinite) dimensional feature space via a nonlinear map, and then computing the dot products in the feature space. It can be shown that under certain conditions these dot products can be expressed via a nonlinear function $K(\cdot, \cdot)$ of the data, so that the dot product of the image in feature space of point $\mathbf{y}_i$ with the image of point $\mathbf{y}_j$ is given by $K(\mathbf{y}_i, \mathbf{y}_j)$ (for more details and further applications of the kernel trick see [6]). In this paper we develop an alternative interpretation for Kernel PCA which builds on work in [7]. This interpretation leads to an objective function based on the cross entropy between two probability distributions. We then show how the missing data problem can be resolved in the context of this objective function.

In [7] it was shown that an alternative probabilistic interpretation of PCA may be derived through placing the Gaussian prior over $\mathbf{W}$ instead of $\mathbf{X}$. Marginalising $\mathbf{W}$ and optimising with respect to $\mathbf{X}$ leads to an eigenvalue problem on the inner product matrix, $\mathbf{K} = \frac{1}{d}\mathbf{Y}\mathbf{Y}^{\mathrm{T}}$. The equivalence of this eigenvalue problem and that of the matrix S is well known (see *e.g.* [8]) and is the foundation for Kernel PCA. In Kernel PCA we simply replace the inner product matrix with an appropriate kernel, $K_{ij} = K(\mathbf{y}_i, \mathbf{y}_j)$. This proves to be very effective in practical applications, leading to excellent results for visualisation and as a pre-processing step for classification. However, the probabilistic interpretation is lost, which means that a lot of the useful characteristics of PPCA are very hard or impossible to obtain in Kernel PCA.

In this paper we reformulate Kernel PCA along the lines suggested in [7], 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 [9] and [2] (where a missing data comparision was also made) and the oil flow data set used in [10]. 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.

The remainder of the paper is organised as follows: we start by briefly reviewing PPCA and the dual PPCA formulation of [7]. We then introduce an objective function for KPCA and discuss how to use this information to deal with missing data. In the fourth section, we present our experimental results. In the fifth 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 Dual PCA, Kernel PCA and Cross Entropy

In probabilistic PCA [2] we assume a linear relationship between the observed variables $\mathbf{y}_i$ and a latent variable $\mathbf{x}_i$,

$$\mathbf{y}_i = \mathbf{W}\mathbf{x}_i + \boldsymbol{\epsilon}, \tag{2}$$

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 2 then implies a Gaussian likelihood for the observed variable,

$$\mathbf{y}_i \sim N\left(\mathbf{W}\mathbf{x}_i, \sigma^2\mathbf{I}\right) \tag{3}$$

The standard machinery would now entail placing a Gaussian prior over the latent variables, marginalising them and optimising the resulting marginal likelihood. Instead, we will prefer here the dual approach to probabilistic PCA taken in [7]. We place a prior distribution on $\mathbf{W}$ in which each element of $\mathbf{W}$ is Gaussian distributed, $w_{ij} \sim N\left(0, 1\right)$, the likelihood of equation 3 can be marginalised with respect to $\mathbf{W}$ to yield a marginal likelihood for the data set of the form

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

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

$$\mathbf{X} = \mathbf{U}\Lambda\mathbf{R},$$

where the columns of $\mathbf{U}$ are the principal eigenvectors of the inner product matrix $\mathbf{K} = \frac{1}{d}\mathbf{Y}\mathbf{Y}^{\mathrm{T}}$; the diagonal entries in Λ are

$$\Lambda = \left(\mathbf{V} - \sigma^2\mathbf{I}\right)^{\frac{1}{2}}$$

where $\mathbf{V}$ is a diagonal matrix of the eigenvalues of $\mathbf{K}$ and $\mathbf{R}$ is an arbitrary rotation in the latent space. The principal components for the covariance matrix $\mathbf{S} = \frac{1}{N}\mathbf{Y}^{\mathrm{T}}\mathbf{Y}$ can then be recovered by the formula

$$\mathbf{U}' = \mathbf{Y}^{\mathrm{T}}\mathbf{U}\mathbf{V}, \tag{4}$$

where $\mathbf{U}'$ are the eigenvectors of $\mathbf{S}$ and $\mathbf{U}$ the eigenvectors of $\mathbf{K}$.

The likelihood for PPCA can be given an interesting interpretation as the *cross entropy* between two Gaussian distributions, one specified by the empirical covariance $\mathbf{S}$ and the other by the approximating covariance $\Sigma = \mathbf{W}\mathbf{W}^T + \sigma^2 I$. This is given, up to an additive constant, by the formula

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

A similar interpretation can be given of the objective function of dual probabilistic PCA, where the covariance matrices are now replaced by inner product matrices

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

where $\mathbf{C} = \mathbf{X}\mathbf{X}^T + \sigma^2 I$. 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 [11]).

2.1 Cross Entropy in Kernel PCA

Dual probabilistic PCA can be turned into a non-linear algorithm in two different ways. One approach is to introduce a Gaussian Process prior on the map between the latent and observed spaces. This leads to the Gaussian Process latent variable model (GPLVM, [7]). The model retains its probabilistic nature, which makes it easy to deal with missing data. However, computation of the images of the latent variables is expensive.

The alternative approach, which leads to KPCA, is to place the nonlinearity directly on the observed points by mapping them to a high dimensional feature space using a non-linear feature map. We have seen in the introduction that, under standard conditions, 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 between the kernel matrix and an approximating Gaussian as an objective function for Kernel PCA

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

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. While we are

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

not aware of a general proof of this fact, there has been experimental evidence supporting it (see e.g. [6]). Notice that, if this holds, reduction to a finite number of features is natural: an implied Gaussian structure in an infinite dimensional space requires the covariance operator to be of trace class, which in turns implies that it can be approximated to arbitrary precision by a finite rank operator.

3 Reconstructing Missing Data

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.

4 Experimental results

4.1 Kernels used

The choice of kernel is a delicate question, and it is often dictated by the particular application of interest. In our experiments we considered two widely used kernels, the RBF and MLP kernels. The RBF kernel is given by the formula

$$k\left(\mathbf{x}_i, \mathbf{x}_j\right) = \exp\left[-\frac{\gamma}{2}\left(\|\mathbf{x}_i - \mathbf{x}_j\|\right)^2\right],$$

where γ is known as the inverse width.

The MLP kernel [12], which can be viewed as an infinite number of sigmoids,

is given by the formula

$$k(x_i, x_j) = \arcsin\left(\frac{w\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j + b}{\sqrt{\left(w\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_i + b + 1\right)\left(w\mathbf{x}_j^{\mathrm{T}}\mathbf{x}_j + b + 1\right)}}\right)$$

where w and b are parameters known as the weight variance and the bias. Experimentally, we found that the MLP kernel was more robust to changes in kernel parameters than the RBF kernel.

4.2 Data sets

To test our approach we tried our algorithm on two well known data sets. In both cases the missing values were initialised to the mean of the corresponding component across the data set[1]. In order to have a reliable reconstruction, one must capture a large part of the data variance in the first place. Therefore, we selected q in order to capture at least 95% of the variance (in terms of eigenvalues of the kernel matrix) of the initialised missing data problem.

The first example is the Tobamovirus data set. This was used in [9] to demonstrate PCA and further used in [2] 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 $w = 10$ and $b = 10^2$.

The results are shown in the left column of Figure 1, where the reconstruction obtained with our method is compared with the underlying truth (KPCA on the full data set) and with the widely used heuristic of replacing missing components with the mean across the data set. Besides providing a visualisation which is much closer to the one obtained with the full data, our method also scores very well in terms of the reconstruction errors: the mean squared error of the reconstructed data set is 6.67, compared to the mean squared error of the initialisation which is 16.63, while the mean squared error in the feature space is 0.065, compared to a mean squared error of 0.20 in the initialised feature space (feature extractors are normalised, hence the small value of the error).

The second data set we considered is the oil flow data set of [10]. 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 for a subsample of 100 data points are shown in Figure 1. We removed at random data points with deletion probability 0.5. Despite the large amount of missing data, the improvement in the visualisation compared with the mean substitution is dramatic.

[1] This is not necessarily the best initialisation, but it is widely used and does not assume any structure in the data.

[2] We noticed however that changes in w and b did not significantly affect the results.

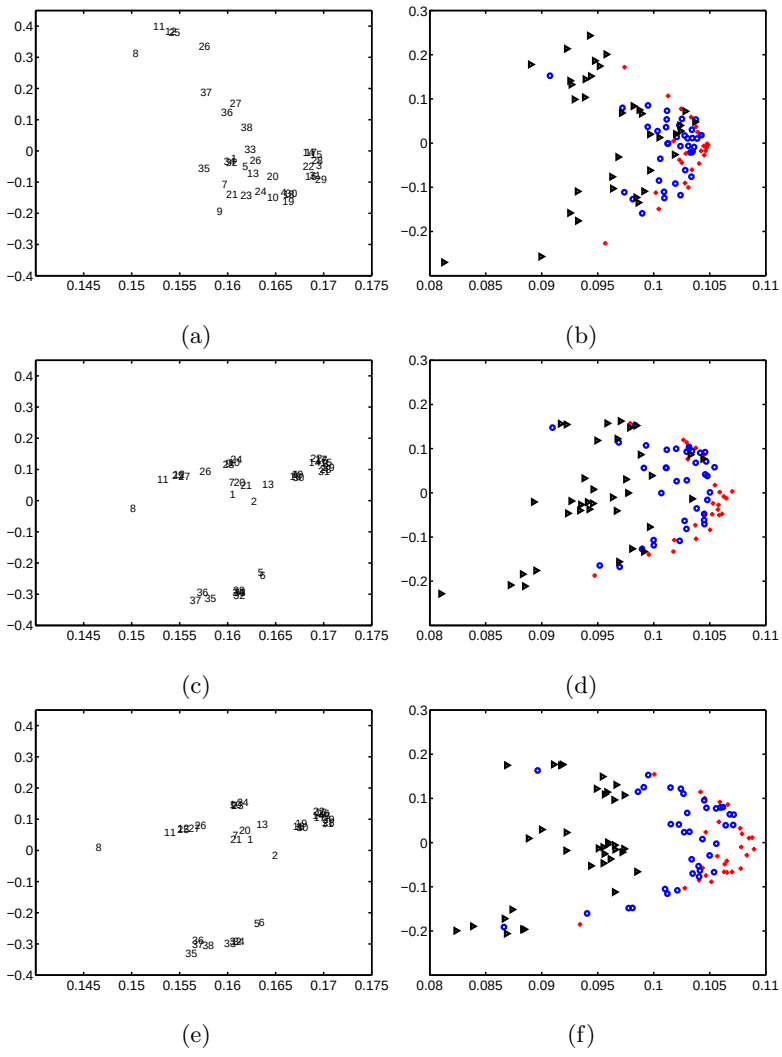


Figure 1: KPCA with missing data. (a) shows the projection on the first two principal components of the initialisation with 20% of the values removed and initialised to the mean for the Tobamovirus data set. (b) shows the projection on the first two principal components of the initialisation with 50% of the values removed and initialised to the mean fr the oil data set. Crosses, triangles and circles represent the three different phases of the oil flow, laminar annular and homogeneous. (c) shows the projection on the first two principal components of the optimal reconstruction of the missing data for the Tobamovirus data set. (d) shows the projection on the first two principal components of the optimal reconstruction of the missing data for the oil data set.(e) shows KPCA on the Tobamovirus data set. (f) shows KPCA on the oil flow data set.

Table 1: Comparison of reconstruction errors for four different methods and ten different probabilities of missing data in the Oil Flow dataset. The first column contains the probability with which data have been removed, the following columns are the mean reconstruction error for each method, averaged over ten different runs and with standard deviation.

p(del)	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
mean	13±4	28±4	43±7	53±8	70±9	81±9	97±9	109±8	124±7	139±7
1 NN	5±3	14±5	30±10	60±20	90±20	NA	NA	NA	NA	NA
PPCA	3.7±0.6	9±2	17±5	25±9	50±10	90±30	110±30	110±20	120±30	140±30
KPCA	5±1	12±3	19±5	24±6	32±6	40±7	45±4	61±8	70±10	100±20

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) 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 Oil Flow data set are summarised in Table 1. The probabilistic methods outperform the two non probabilistic methods for low to medium deletion probabilities: curiously, the mean substitution seems to be more robust, for high percentage of missing data, than probabilistic PCA. The 1NN approach is reasonable for small percentages of missing data, but badly breaks down when the number of missing data increases; ultimately, when no data point is exempt from missing data, it becomes inapplicable.

Probabilistic PCA and our algorithm perform in a very similar way for small percentages of missing data (perhaps with a slight advantage of probabilistic PCA, although the difference is not statistically significant); however, when the percentage of missing data increases, our approach yields consistently better results than probabilistic PCA.

A similar pattern is shown for the Tobamovirus data set in Table 2. Interestingly, probabilistic PCA performs extremely badly even for relatively small deletion probabilities. This is probably due to the data set having more accentuated nonlinearities than the oil flow. Our algorithm again outperforms (sometimes dramatically) the others.

5 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

Table 2: Comparison of reconstruction errors for four different methods and ten different probabilities of missing data in the Tobamovirus dataset.The first column contains the probability with which data have been removed, the following columns are the mean reconstruction error for each method, averaged over ten different runs and with standard deviation.

p(del)	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
mean	4±1	8±2	12±2	16±3	22±3	27±4	31±4	35±4	39±4	45±5
1 NN	2.3±1.9	11±6	22±10	NA	NA	NA	NA	NA	NA	NA
PPCA	4.5±3.8	8±2	14±4	29±5	51±20	65±15	85±31	115±36	65±13	66±12
KPCA	0.9±0.9	2.1±1.2	5±2	8±4	9±3	12±5	21±15	24±10	27±8	37±14

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

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

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).

Applying this to Kernel PCA meets two difficulties: the generative distribution given by the principal components lives in the feature space, which can be infinite dimensional and intractable; secondly, even if we found a point in feature space which maximises the conditional distribution, obtaining a pre-image for points in feature space is generally impossible. Notice incidentaly that, in linear PPCA, the maximum of the conditional probability does not lie on the principal components, unless the subspace obtained by varying the missing components is orthogonal to the principal components. This means that a priori we cannot assume that the point we need is a linear combination of the images of the training points.

We can though circumvent both these problems 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

$$\mathbf{y}_t^T C^{-1} \mathbf{y}_t = \sum_{i=1}^{q} \left(\lambda_i^{-1} - \sigma^{-2}\right) \left(\mathbf{y}_t \cdot \mathbf{u}_i\right)^2 + \sigma^{-2} \|\mathbf{y}_t\|^2 \tag{6}$$

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 (6) 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 (6) needs not be included.

Recalling that the KPCA feature extractors in feature space are given by

$$\mathbf{u}_i = \sum_{j=1}^{N\text{train}} \alpha_j^i \mathbf{\Phi}(\mathbf{y}_j)$$

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

$$\mathcal{L} = \sum_{i=1}^{q} \left(\lambda_i^{-1} - \sigma^{-2}\right) \left(\sum_{j=1}^{N\text{train}} \alpha_j^i k(\mathbf{y}_j, \mathbf{y}_t) \right)^2 . \tag{7}$$

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.

An example of this approach is 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 we artificially corrupted its first five coordinates by multiplying them by a constant factor. The point recovered through optimising the objective function (7) is very close indeed.

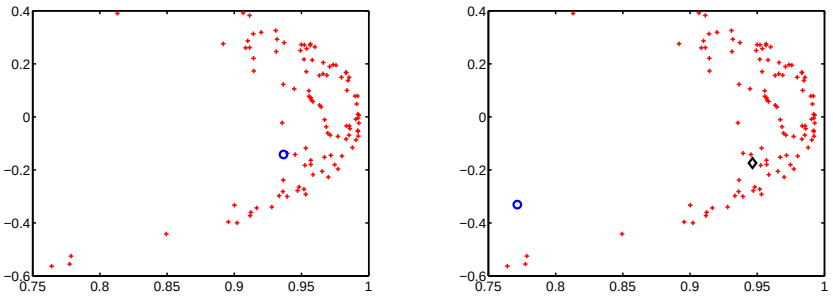
To quantify the efficacy of our method, we repeated the example of Figure 2 removing a different point at random fifty times and replacing its first five coordinates with random numbers. We also increased the number of features extracted from two to ten. The results are summarised in Table 3, where a 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).

6 Discussion

In this paper we introduced an objective function for Kernel PCA, building on previous work on probabilistic PCA [2] and latent variable models in Gaussian Processes [7]. This in turns allows to extend important inference techniques,

Table 3: 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



(a) (b)

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

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 probabilistic 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.

There can be several variations to our approach that lead to similar results. The objective function we chose, the cross entropy, is not the only possible objective function for Kernel PCA. In particular, the Kullback-Leibler (KL) divergence is often chosen as the natural objective function in a number of probabilistic settings. The two are clearly closely related, and can be both used as an objective function for Kernel PCA; however, from our experiments we found that the KL divergence is less suitable when handling missing data problems. This appears to be because the $\log |\mathbf{K}|$ term that appears in the KL divergence but not in the cross entropy favours point configurations that lead to a nearly rank-deficient matrix $\mathbf{K}$, which means the missing values tend to drift farther afield.

A somewhat dual approach to the one taken in this paper is adopted in [7]. There, a nonlinear kernel is placed on the latent points, while a Gaussian covariance governs the distribution of the observed data. As a result, the reconstruction of missing data is computationally easier, while the computation of the images of the latent points is expensive. However, the two problems are inherently distinct and result in different visualisations.

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, which in the linear case is given by (4), 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.

References

[1] Jolliffe, I.T.: Principal Component Analysis. Springer-Verlag, New York (1986)

[2] Tipping, M.E., Bishop, C.M.: Probabilistic principal component analysis. Journal of the Royal Statistical Society, B **6**(3) (1999) 611–622

[3] Tipping, M.E., Bishop, C.M.: Mixtures of principal component analysers. In: Proceedings IEE Fifth International Conference on Artificial Neural Networks, Cambridge, U.K., July. (1997) 13–18

[4] Bishop, C.M.: Bayesian PCA. In Kearns, M.J., Solla, S.A., Cohn, D.A., eds.: Advances in Neural Information Processing Systems. Volume 11., Cambridge, MA, MIT Press (1999) 482–388

[5] 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

[6] Schölkopf, B., Smola, A.J.: Learning with Kernels. MIT Press (2001)

[7] 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

[8] Tipping, M.E.: Sparse kernel principal component analysis. In Leen, T.K., Dietterich, T.G., Tresp, V., eds.: Advances in Neural Information Processing Systems. Volume 13., Cambridge, MA, MIT Press (2001) 633–639

[9] Ripley, B.D.: Pattern Recognition and Neural Networks. Cambridge University Press, Cambridge, U.K. (1996)

[10] Bishop, C.M., Svensén, M., Williams, C.K.I.: GTM: the Generative Topographic Mapping. Neural Computation **10**(1) (1998) 215–234

[11] 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)

[12] 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)