Forecasting using Twinned Principal Curves

Ying Han and Colin Fyfe,

Applied Computational Intelligence Research Unit, The University of Paisley, Scotland.

No Institute Given

Abstract. Principal curves are extensions of Principal Component Analysis and are smooth curves which pass through the middle of a data set. We extend the method so that, on pairs of data sets which have underlying nonlinear correlations, we have pairs of curves which go through the 'centre' of data sets in such a way that the nonlinear correlations between the data sets are captured. We relate this method to Canonical Correlation Analysis and give an illustrative example of this method on artificial data before applying it as a forecasting method on a real data set.

1 Introduction

Canonical Correlation Analysis (CCA) is a statistical technique for estimating the linear combination of a data set which gives the greatest correlation with a linear combination of a second data set. Let $\mathbf{x_1}$ be a vector drawn from the first data set and let $\mathbf{x_2}$ be the corresponding vector drawn from the second data set. Then CCA attempts to estimate $\mathbf{w_1}$ and $\mathbf{w_2}$ such that $y_1 = \mathbf{w_1}^T \mathbf{x_1}$ and $y_2 = \mathbf{w_2}^T \mathbf{x_2}$ have the greatest correlation over the whole set of samples $\mathbf{x_1}$ and $\mathbf{x_2}$. We have previously developed neural algorithms [2, 6, 8] for performing CCA; the neural algorithms have certain advantages over standard statistical techniques including the ability to find nonlinear projections of a data set which maximise correlations. We have also used the neural algorithms for forecasting [7]: one data set is the previous samples of a time series, the other is the sample(s) which one wishes to predict.

In this paper, we present an extension of Principal Curves which performs a type of non-parametric CCA and illustrate its use on artificial data. We then use the method to forecast on a financial data set which we have previously [3] used to test other forecasting methods.

2 Twinned Principal Curves

Principal Component Analysis (PCA) is a standard statistical technique for finding a lower dimensional linear projection of high dimensional data which

gives minimum mean square error over all projections of this dimensionality. Principal Curves [4, 5, 1] is an extension of this method in which a nonlinear manifold can be used instead of the linear subspace determined by PCA. However there is clearly a difficulty with this in that it is always possible to fit a finite training set with no error. There are several definitions of Principal Curves which constrain the curves in one way or another to overcome the problem of overfitting. In [4], every point, P, on the curve is the mean of the points that project onto P. This is known as self-consistency. The unit-speed curve (one whose derivative has norm 1) which satisfies this is the Principal Curve. In [5], the Principal Curve is defined as the curve of a specific length which minimises the mean squared distance from the data.

In this paper, we extend the Principal Curve method so that we now find a nonlinear manifold in each of two data sets. We use a non-parametric method to determine the two manifolds. Since we are drawing data iid from two data sets simultaneously, our method creates manifolds which exhibit a correlation between corresponding points on the manifolds which we can then use to subsequently forecast a sample from one data set given a sample from the other. The algorithm in outline is

- 1. With the current projections d_1^i and d_2^i , $\forall i$,
- 2. Select \mathbf{x}_1^i from the first data set and the corresponding point, \mathbf{x}_2^i from the second data set.
- 3. Find all neighbours of the point which have
 - projections close to the projections of the chosen point.
 - projections of their corresponding points in the other data set satisfying the same constraint with respect to the second data set. Note that these projections will be to different curves.
- 4. Thus, if d_1^i is the projection of \mathbf{x}_1^i and d_2^i is the projection of \mathbf{x}_2^i , then $S_i = \{k : |d_1^k d_1^i| < \epsilon_1 \text{ and } |d_2^k d_2^i| < \epsilon_2\}$ 5. Find the local average of points projecting close to \mathbf{x}_1^i and \mathbf{x}_2^i , i.e. $d_1^i(\text{new}) = \sum_{i=1}^{k} d_i^i \cdot \mathbf{x}_i^i \cdot \mathbf{$
- Mean of $d_1^j, j \in S_i$ and $d_2^i(\text{new}) = \text{Mean of } d_2^j, j \in S_i$
- 6. Return to Step 1.
- 7. $d_1^i = d_1^i \text{ (new) and } d_2^i = d_2^i \text{ (new)}, \forall i.$

The algorithm iterates till a stopping criteria is met: either the algorithm repeats for a set number of rounds or till the number of nodes to which the data is projected reaches a certain number (see below) or till the mean square error reaches a particular value.

Finally the algorithm is initialised with the projections being to the first Principal Component of each data set. Clearly there are extensions which can be made to this algorithm. For example it is possible to change the value of the width parameters ϵ_1 and ϵ_2 during the course of the iterations, though this is not implemented in the simulations discussed in this paper for reasons which will become clear in the next section. Also, the use of a weighted average rather than a simple average may improve the accuracy of the new projections. Finally, the algorithm tends to draw data from the extremes of the principal curve and so some additional local averaging may be useful in this case. Again the last two points are not implemented in the results discussed in this paper.

3 Properties of Twinned Principal Curves

This is a somewhat different algorithm from that suggested by [4] or [5] in that it iteratively uses a kernel smoother rather than attempting to approximate a principal curve by a mixture of straight lines. However it has a rather nice property of sparsification of the projections: the local averaging provides a smoothing of the data set and since we keep the values of ϵ_1 and ϵ_2 constant during the course of the simulation this smoothing progressively works out from each data point resulting in fewer and fewer projections onto the principal curve (compare the central two rows in Figure 1). We may use this property to allow the number of distinct nodes we seek to determine the value of ϵ_1 and ϵ_2 (or vice-versa).

It is worth noting also that this algorithm is able to deal with data sets which standard Principal Curve algorithms find difficult: the very fact of having two data sets with which to work simultaneously alleviates several problems. For example, since we initialise with a PCA and one of our data sets is circular, any diameter of the circle may be a Principal Component direction. This unfortunately means that points on opposite sides of the circle project onto the same part of the eigenvector and so we often have an initial twisting of the Principal Curve as it moves from the centre of mass on one side of the circle to the centre of mass on the other side, these centres of mass being caused by the finite numbers of samples. However, we only consider points to be local to the current point if they are local in both projections. This makes it much less likely that false neighbours will be chosen.

Finally, CCA maximises the correlation between two data sets under the constraint that the variance of $y_1 = \mathbf{w}_1^T \mathbf{x}_1$ and $y_2 = \mathbf{w}_2^T \mathbf{x}_2$ are both 1. Twinned Principal Curves can still meet this criterion; having found our sum of linear approximators, we may project new samples onto these Twinned Principal Curves and calculate the variance of the resultant projections. In calculating new correlations, we may simply then divide each of y_1 and y_2 by their corresponding standard deviations.

4 Experiments

4.1 Artificial data

We first create 2 sets of two dimensional artificial data which are known to have a correlation from $x_1(t) = \sin(t) + \mu_1, y_1(t) = \cos(t) + \mu_2, x_2(t) = t + \mu_3, y_2(t) = \frac{t}{3} + \sin(t) + \mu_4$ where t is drawn from a uniform distribution in $[0, 2\pi]$ and $\mu_i = N(0, 0.2)$ is Gaussian noise. Examples of this data are shown in the top row of Figure 1.

Figure 1 also shows the thinning which takes place in data set 2 after 1, 2 and 10 iterations and in data set 1 after 10 iterations. The sparsification discussed above is clearly evident.

Now we may use these projections to predict the position of a point, \mathbf{x}_2 , in data set 2 given its corresponding point \mathbf{x}_1 in data set 1. Typically we will approximate the principal curves with the sum of linear projections given by

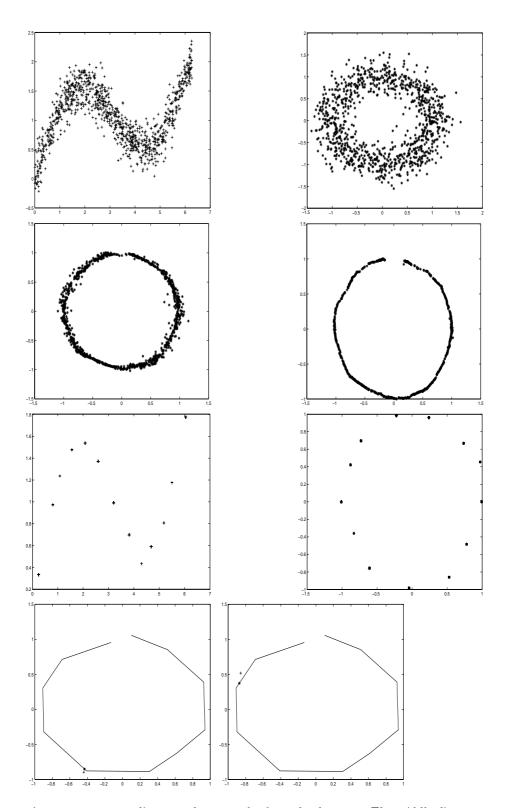


Fig. 1. The top two diagrams show samples from the data sets. The middle diagrams show the first and second projections of the second data set. The third row shows the projections of both data sets after 10 iterations. The last row shows the results of forecasting the positions of points in data set 2 given only the position of the corresponding point in data set 1.

joining the sparse points as shown in the last row of Figure 1. To forecast, we project \mathbf{x}_1 onto the current principal curve of the first data set and use the corresponding point on the current principal curve of the second data as the predictor of \mathbf{x}_2 . Typical results are shown in the last row of Figure 1, the "*" on the curve being the predictor while the "+" shows the point's actual position.

4.2 Forecasting

The problem we have modelled is a forecasting one: given the last few days' exchange rates (U.S. dollar against Japanese yen), is it possible to forecast the next day's exchange rate with some degree of accuracy? We have previously [3] used a variety of methods to find the underlying factors in this data set and then used a standard multilayered perceptron using backpropagation to predict each factor separately. To test our multilayered perceptron, we have split the data set into two sets: 1706 samples were used as the training data and 1706 for the test data. Each training input comprised a particular day's exchange rate plus the previous n days' exchange rates where values of n ranged from 5 to 25. With the current algorithm, we can simultaneously forecast as many days in advance as we wish, since our second principal curve can be as high dimensional as we wish. Typical results in terms of Mean Absolute Percentage Error on the test set are given in Table 1.

Knot points	Day 1	Day 2	Day 3	Day 4	Day5
57	1.0006	1.1086	1.2103	1.3035	1.4022
408	0.7413	0.9158	1.0685	1.1863	1.4022 1.2887 1.0880
607	0.6711	0.7939	0.9018	1.0197	1.0880

Table 1. The first column gives the number of knot points and the others give the mean absolute percentage error on a test data set predicting 1 to 5 days ahead.

5 Conclusion

We have shown that the Principal Curve method can be extended to work on two data sets simultaneously and that using two data sets is, in fact, advantageous in that there is less chance of two projections simultaneously misleading than there is of a single projection being misleading. Also when we use this algorithm to forecast, we have the advantage that it is very simple to forecast a number of days ahead simultaneously: this simply increases the dimensionality of the space through which the second principal curve moves. The results from the foreasting were comparable to that from our previous methods [3] and were considerably easier to achieve: we performed no optimisation to get the reported results and found comparable results over a wide range of parameter values.

References

- 1. P. Delicado. Another look at principal curves and surfaces. *Journal of Multivariate Analysis*, 77:84–116, 2001.
- 2. Z. Gou and C. Fyfe. A family of networks which perform canonical correlation analysis. *International Journal of Knowledge-based Intelligent Engineering Systems*, April 2001.
- 3. Y. Han and C. Fyfe. Finding underlying factors in time series. *Cybernetics and Systems: An International Journal*, March 2002.
- 4. T. Hastie and W. Stuetzle. Principal curves. Journal of the American Statistical Association, 84(406), June 1989.
- 5. B. Kegl, A. Krzyzak, T. Linder, and K. Zeger. Learning and design of principal curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):281–297, 2000.
- 6. P. L. Lai and C. Fyfe. A neural network implementation of canonical correlation analysis. *Neural Networks*, 12(10):1391–1397, Dec. 1999.
- 7. P.L. Lai, S.J. Chuang, and C. Fyfe. Power load forecasting using neural canonical correlates. In *International Conference on Pattern Recognition*, *ICPR'2000*. IEEE Cs Press, 2000.
- 8. P.L. Lai and C. Fyfe. Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems*, 10(5):365–377, 2001.