Gaussian Processes for Principal Component Analysis

Colin Fyfe,
Applied Computational Intelligence Research Unit,
The University of Paisley, Scotland.
email:colin.fyfe@paisley.ac.uk

Abstract

We show how the supervised method of Gaussian Processes may be used for Principal Component Analysis using two intuitions about the nature of the first principal component filters.

1 Introduction

A stochastic process $Y(\mathbf{x})$ is a collection of random variables indexed by $\mathbf{x} \in X$ such that values at any finite subset of X form a consistent distribution. A Gaussian Process (GP) therefore is a stochastic process on a function space which is totally specified by its mean and covariance function [8, 6, 5, 7].

Gaussian processes use supervised learning algorithms: we require a training data set on which we already know the targets for regression or classification. We have recently investigated GPs for unsupervised learning, particularly for Canonical Correlation Analysis [1, 4, 2, 3]. In this paper, we investigate the use of Gaussian processes to perform Principal Component Analysis (PCA).

2 GP for Principal Component Analysis

For Principal Component Analysis, we require to define a target value for each $\mathbf{x}$. Let us consider only the first principal component: we will create a target using intuitions about the nature of this principal component in two separate ways. The first method is an entropy based method: for each data point, our intuition is that we can use nearby points to try to predict the position of the data point but that we will be least successful in the principal component projection simply because it has most entropy. The other method is based on a geometric criterion: if we can identify points far away (in the projection manifold) from the current point, the line joining these points to the current point is liable to have a large component in the principal component direction (Figure 1). In

both cases, we parameterise the non-zero mean function by $\theta(\mathbf{x}) = \mathbf{b}^T\mathbf{x} + c$, which will identify the first principal component filter.

2.1 An Entropy-based Criterion

The first principal component contains most variance in the data set: our intuition for this model comes from the fact that, for Gaussian distributions, this is the greatest entropy projection. Knowing the positions of the neighbours of a data point on this projection, leaves you least sure about the position of the data point. Therefore, we use the closest M projections of the data set onto the current estimate of the first principal component and adjust the model parameters with respect to the average position of the data points on which these projections were based. We initialise the process with the targets as the average of the M closest data points, not including the data point whose projection we are trying to estimate and subsequently *move away* from the average position of these M closest data points. In terms of the GP model, let S_j be the set of data points whose current estimated projection is closest to that of $\mathbf{x}_j$. Then t_j, the target for the projection of $\mathbf{x}_j$ is given by $t_j = \frac{1}{|S_j - 1|}\sum_{k \in S_j, k \neq j} \mathbf{b}^T\mathbf{x}_k$ and

$$\Delta\mathbf{b}|_{\mathbf{x}=\mathbf{x}_j} \propto -\frac{\partial L}{\partial \mathbf{b}}|_{\mathbf{x}=\mathbf{x}_j} = (t(\mathbf{x}_j) - \mathbf{b}^T\mathbf{x}_j)\Sigma^{-1}\mathbf{x}; \tag{1}$$

where L is the log likelihood of the data, Σ and $\mathbf{b}^T\mathbf{x}$ are the covariance and mean of the GP respectively.

We will see that there are stability problems with this model which do not arise with the second model.

2.2 A Geometric Model

Our second intuition comes from the fact that if we have points far away from the current data point, the line joining these points is liable to contain a greater component in the direction of the first principal component than in any other direction. Figure 1 illustrates this with an admittedly extreme data set.

Therefore an alternative algorithm to the above is created by finding those data points whose projections are most different from the current data point and using as a target, the average of those data points' projections onto the current data point. Let R_j be the set of data points whose current estimated projection is furthest from that of $\mathbf{x}_j$. Then t_j, the target which we wish the projection of $\mathbf{x}_j$ to use as target, is given by $t_j = \frac{1}{|R_j|}\sum_{k \in R_j} \frac{\mathbf{x}_k^T\mathbf{x}_j}{\|\mathbf{x}_k\|}$. In practice, the normalisation has not been found to be necessary and we use $t_j = \frac{1}{|R_j|}\sum_{k \in R_j} \mathbf{x}_k^T\mathbf{x}_j$; in fact, we have empirical results which suggest that having the target with larger amplitude than the $\mathbf{b}$ vector gives faster convergence. We have consistently, stably and accurately found the first principal component by this method.

In Figure 2, we show the results of four simulations on the well known iris data set (150 samples of four dimensional data): the top line shows two simulations with the geometric method, the second of which uses $t_j = \frac{5}{|R_j|}\sum_{k \in R_j} \mathbf{x}_k^T\mathbf{x}_j$

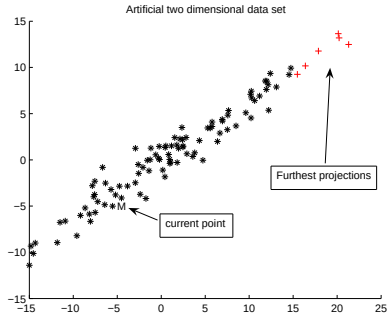

Figure 1: The line joining the labelled point M to the average of the furthest principal component projections is almost in the direction of this principal component.

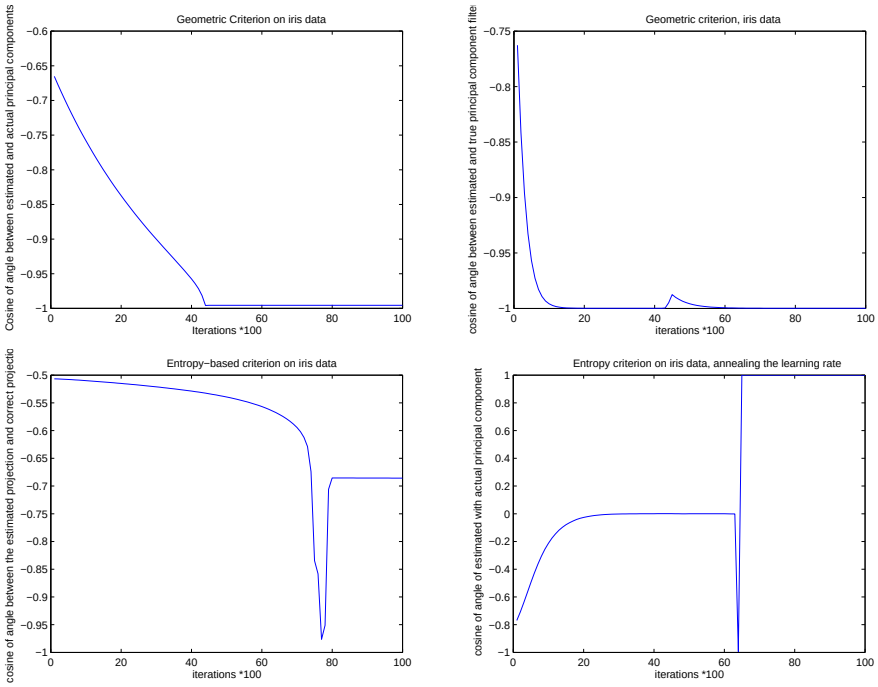

Figure 2: . Simulations on iris data. Top left: the geometric method. Top right: the geometric method when the targets are $t_j = \frac{5}{|R_j|} \sum_{k \in R_j} \mathbf{x}_k^T \mathbf{x}_j$. Bottom: two simulations with the entropy method. The former method is clearly more stable.

which seems to speed convergence. The bottom line shows two simulations with the entropy-based method. We see that the geometric method is stable while the entropy method is less predictable.

References

[1] C. Fyfe and G. Leen. Stochastic processes for canonical correlation analysis. In *14th European Symposium on Artificial Neural Networks*, 2006.

[2] P. L. Lai and C. Fyfe. A latent variable implementation of canonical correlation analysis for data visualisation. In *International Joint Conference on Neural Networks*, 2006.

[3] P. L. Lai and C. Fyfe. The sphere-concatenate method for gaussian process canonical correlation analysis. In *16th International Conference on Artificial Neural Networks, ICANN2006*, 2006.

[4] P. L. Lai, G. Leen, and C. Fyfe. A comparison of stochastic processes and artificial neural networks for canonical correlation analysis. In *International Joint Conference on Neural Networks*, 2006.

[5] D. J. C. MacKay. Introduction to gaussian processes. Technical report, University of Cambridge, http://www.inference.phy.cam.uk/mackay/gpB.pdf, 1997.

[6] C. E. Rasmussen. *Advanced Lectures on Machine Learning*, chapter Gaussian Processes in Machine Learning, pages 63–71. 2003.

[7] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[8] C. K. I. Williams. Prediction with gaussian processes: from linear regression to linear prediction and beyond. Technical report, Aston University, 1997.