ESANN'2007 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2007, d-side publi., ISBN 2-930307-07-2.

Immediate Reward Reinforcement Learning for Projective Kernel Methods

Colin Fyfe[1] and Pei Ling Lai[2],
1. Applied Computational Intelligence Research Unit,
The University of Paisley, Scotland.
email:colin.fyfe@paisley.ac.uk
2. Southern Taiwan University of Technology, Taiwan
email: pei_ling_lai@hotmail.com

Abstract. We extend a reinforcement learning algorithm which has previously been shown to cluster data. We have previously applied the method to unsupervised projection methods, principal component analysis, exploratory projection pursuit and canonical correlation analysis. We now show how the same methods can be used in feature spaces to perform kernel principal component analysis and kernel canonical correlation analysis.

1 Introduction

Unsupervised and reinforcement learning research have tended, in the main, to be two entirely separate streams of adaptive methods. This is somewhat surprising in that the former is modelled on biological processes believed to be utilised in animal and human brains while the latter is attempting to create learning processes which an entity would use to investigate its environment in order to maximise its long term utility.

A notable exception to the above dearth of interaction is [4] which uses a reinforcement learning method (albeit a non-standard one) in order to cluster data sets in an unsupervised manner. We have recently [1] investigated this reinforcement learning algorithm in order to perform a topology preserving clustering of the data and in order to perform linear projections. In this paper, we extend the method to kernel spaces and show that we can use it to perform kernel principal component analysis and kernel canonical correlation analysis.

2 Immediate Reward Reinforcement Learning

[7] investigated a particular form of reinforcement learning in which reward for an action is immediate which is somewhat different from mainstream reinforcement learning [2]. Williams [7] considered a stochastic learning unit in which the probability of any specific output was a parameterised function of its input, $\mathbf{x}$. For the i^{th} unit, this gives

$$P(y_i = \zeta | \mathbf{w}_i, \mathbf{x}) = f(\mathbf{w}_i, \mathbf{x}) \tag{1}$$

where, for example,

$$f(\mathbf{w}_i, \mathbf{x}) = \frac{1}{1 + \exp(- \parallel \mathbf{w}_i - \mathbf{x} \parallel^2)} \tag{2}$$

Williams [7] considers the learning rule

$$\Delta w_{ij} = \alpha_{ij}(r_{i,\zeta} - b_{ij})\frac{\partial \ln P(y_i = \zeta|\mathbf{w}_i, \mathbf{x})}{\partial w_{ij}} \tag{3}$$

where α_{ij} is the learning rate, $r_{i,\zeta}$ is the reward for the unit outputting ζ and b_{ij} is a reinforcement baseline which in the following we will take as the reinforcement comparison, $b_{ij} = \overline{r} = \frac{1}{K}\sum r_{i,\zeta}$ where K is the number of times this unit has output ζ. ([7], Theorem 1) shows that the above learning rule causes weight changes which maximises the expected reward.

[7] gave the example of a Bernoulli unit in which $P(y_i = 1) = p_i$ and so $P(y_i = 0) = 1 - p_i$. [4] applies the Bernoulli model to (unsupervised) clustering based on which we have recently created a topology preserving algorithm. We are more interested in the Gaussian learner from which we draw a sample $y \sim N(\mathbf{m}_i, \beta_i^2)$, the Gaussian distribution with mean $\mathbf{m}_i$ and variance β_i^2. Each learner has two parameters to adapt, its mean and variance. The learning rules can be derived [7] as

$$\Delta\mathbf{m} = \alpha_m(r - \overline{r})\frac{\| \mathbf{y} - \mathbf{m} \|}{\beta^2} \tag{4}$$

$$\Delta\beta = \alpha_\beta(r - \overline{r})\frac{\| \mathbf{y} - \mathbf{m} \|^2 - \beta^2}{\beta^3} \tag{5}$$

We have investigated such algorithms in the context of principal component analysis, exploratory projection pursuit and canonical correlation analysis [1]. In this paper, we apply the technique in kernel space and show that we may perform kernel principal component analysis [5] and kernel canonical correlation analysis [3] with immediate reward reinforcement learning.

3 Unsupervised Kernel Methods

There has been a great deal of recent interest in using kernel methods both in the supervised learning paradigm [6] and in the unsupervised paradigm [5, 3]. Kernel methods map the data first into a feature space in which a linear operation (such as principal component analysis or canonical correlation analysis) is performed. If the mapping into the feature space is a non-linear one, any linear operation in the feature space corresponds to a nonlinear operation in the original data space. Since we are particularly interested in applying the reinforcement learning paradigm to unsupervised learning, we will illustrate its use in kernel space for kernel principal component analysis and kernel canonical correlation analysis.

3.1 Kernel Principal Component Analysis (KPCA)

Let the nonlinear mapping be $\phi : \Xi \rightarrow F$ where Ξ is the original data space and F is the feature space to which the elements of Ξ are mapped. Then Kernel Principal Component Analysissearches for the filter $\mathbf{w} = \arg\max_{\mathbf{w}'} E_\Xi(\mathbf{w}'^T\phi(\mathbf{x}))$ i.e. the weight vector in feature space which maximises the variance of the (centered)

projections (under the constraint of orthonormality of the weight vectors. The crucial insight is that $\mathbf{w}$ may be defined in terms of the points $\phi(\mathbf{x})$ which span the subspace in which $\mathbf{w}$ lies. Thus $\mathbf{w} = \sum_{i=1}^{N} \alpha_i \phi(\mathbf{x}_i)$. Therefore, the problem reduces to finding the weight vector α in feature space. It is readily shown [5] that α is the eigenvector of the scalar product matrix, $K : K_{ij} = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$. Thus finding the first principal component filter $\mathbf{w}$ in the feature space can be readily done by finding $\alpha = \arg\max_\beta \beta^T K \beta$. Actually, in order to normalise $\mathbf{w}$ properly, we require to normalise α appropriately, however, in the following we shall ignore this issue since we are only interested in identifying structure in the data sets.

Thus, with the methodology defined above, we draw α from the Gaussian distribution $N(\mathbf{m}, \beta I)$ where now $\mathbf{m}$ is $N \times 1$, and I is similarly the $N \times N$ identity matrix. At each iteration, we select a specific input, e.g. $\mathbf{x}_t$ and calculate the reward as $r = \alpha^T K_t \alpha_t$ where K_t is the t^{th} column of K i.e. that corresponding to $\mathbf{x}_t$ and α_t is the element of α corresponding to $\mathbf{x}_t$. This constitutes the reward, r.

To illustrate this we create a two dimensional data set of 100 samples, the first $\frac{1}{4}$ are centered at (2,2), the next $\frac{1}{4}$ at (2,-2), and so on as shown in Figure 1. We use a squared exponential kernel so that $K_{ij} = \exp(-\gamma(\mathbf{x}_i - \mathbf{x}_j)^2)$. We then center this representation [5] using $\tilde{K} = K - \frac{1}{N}\mathbf{1}K - \frac{1}{N}K\mathbf{1} + \frac{1}{N^2}\mathbf{1}K\mathbf{1}$ where $\mathbf{1}$ is the $N \times N$ matrix of 1s. Results from simulations in which $\gamma = 0.1$ and $\gamma = 1$ are shown in Figure 1.

Since kernel methods, in general, require the construction of the scalar product (Gram) matrix, K, they are often used in batch mode rather than in online mode as above. Then the reward function becomes $r = \alpha^T K \alpha$ i.e. we are calculating the reward over the whole data set at each iteration. With all other parameters held constant and $\gamma = 0.1$, we show the results of one simulation on the same data as before in the last diagram of Figure 1 though the simulation had to be run for a slightly longer time.

3.2 Kernel Canonical Correlation Analysis

Canonical correlation analysis is a standard statistical technique for investigating two data sets which we believe have some underlying (linear) relationship. This has also been extended to utilise kernel techniques [3]. Let the covariance matrices in Feature space be defined by

$$\begin{aligned}
\Sigma_{11} &= E\{(\Phi(\mathbf{x}_1) - \mu_1)(\Phi(\mathbf{x}_1) - \mu_1)^T\} \\
\Sigma_{22} &= E\{(\Phi(\mathbf{x}_2) - \mu_2)(\Phi(\mathbf{x}_2) - \mu_2)^T\} \\
\Sigma_{12} &= E\{(\Phi(\mathbf{x}_1) - \mu_1)(\Phi(\mathbf{x}_2) - \mu_2)^T\}
\end{aligned}$$

where now $\mu_i = E(\Phi(\mathbf{x}_i))$ for $i = 1, 2$. Let us assume for the moment that the data has been centred in feature space (we actually use the same trick as [5] again to centre the data). Then we wish to find those values $\mathbf{w}_1$ and $\mathbf{w}_2$ which will maximise $\mathbf{w}_1^T \Sigma_{12} \mathbf{w}_2$ subject to the constraints $\mathbf{w}_1^T \Sigma_{11} \mathbf{w}_1 = 1$ and $\mathbf{w}_2^T \Sigma_{22} \mathbf{w}_2 = 1$.

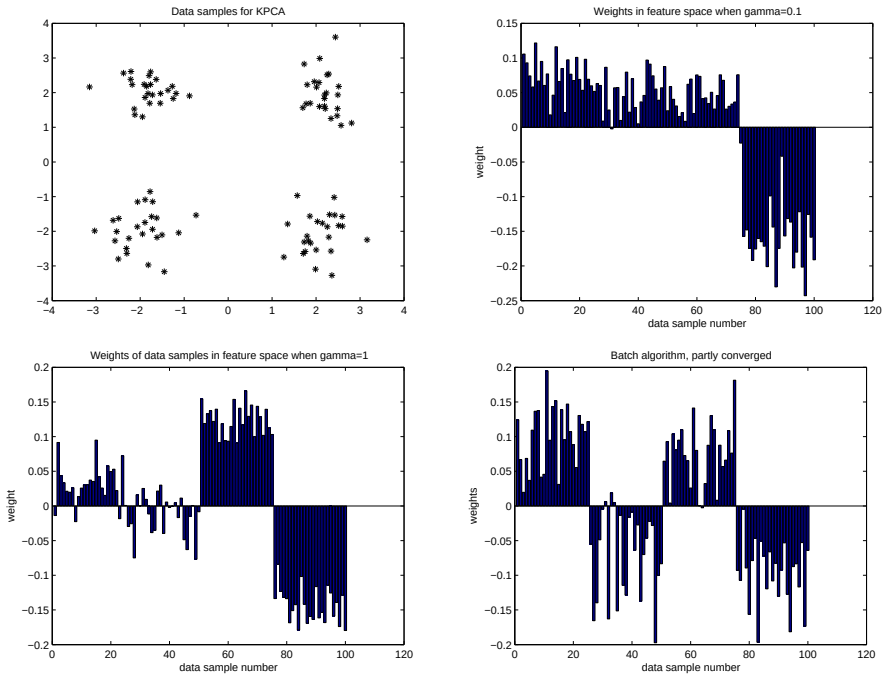

Fig. 1: Top left: the two dimensional data set. Top right: the α weights in a simulation in which $\gamma = 0.1$. Bottom left: as above but with $\gamma = 1$. Bottom right: with reward $\mathbf{w}^t\mathbf{x}$.

In [3], we define $(K_1)_{ij} = \Phi^T(\mathbf{x}_i)\Phi(\mathbf{x}_{1j})$ and $(K_2)_{ij} = \Phi^T(\mathbf{x}_i)\Phi(\mathbf{x}_{2j})$ and so maximise $\alpha^T K_1 K_2^T \beta$ subject to the constraints $\alpha^T K_1 K_1^T \alpha = 1$ and $\beta^T K_2 K_2^T \beta = 1$. Therefore if we define $\Gamma_{11} = K_1 K_1^T$, $\Gamma_{22} = K_2 K_2^T$ and $\Gamma_{12} = K_1 K_2^T$ we solved the problem in the usual way: by forming matrix $K = \Gamma_{11}^{-\frac{1}{2}} \Gamma_{12} \Gamma_{22}^{-\frac{1}{2}}$ and performing a singular value decomposition on it as before to get

$$K = (\gamma_1, \gamma_2, ..., \gamma_k) D(\theta_1, \theta_2, ..., \theta_k)^T \tag{6}$$

where γ_i and θ_i are again the standardised eigenvectors of KK^T and $K^T K$ respectively and D is the diagonal matrix of eigenvalues.

Then the first canonical correlation vectors in feature space are given by

$$\alpha_1 = \Gamma_{11}^{-\frac{1}{2}} \gamma_1 \tag{7}$$

$$\beta_1 = \Gamma_{22}^{-\frac{1}{2}} \theta_1 \tag{8}$$

with subsequent canonical correlation vectors defined in terms of the subsequent eigenvectors, γ_i and θ_i.

Again, we are only interested in proving the reinforcement learning method in this paper and so we create a 100 sample, artificial data set in which the first 80 samples in each stream are Gaussian clusters of standard deviation 0.3 round centres in each space and the last 20 samples are also Gaussian clusters of standard deviation 1 round different centres in each space. We then randomly initialise γ and θ and generate a reward, $r = \gamma K \theta$ and using this to update γ and θ. Results are shown in Figure 2. We see that the two distinct regimes incorporating the different relationships are clearly visible. Again, we note that we normalise the lengths of the weight vectors γ and θ to have length 1 when actually we should be normalising $\mathbf{w}_1$ and $\mathbf{w}_2$; however this only affects the magnitude of the results not the identification of different regimes in the data.

4 Conclusion

We have taken an existing method for immediate reinforcement learning and applied the algorithm to two linear methods in feature space and shown that the resulting method is clearly able to identify nonlinear structure in data sets. In this paper, we have merely introduced the method. Further research is needed to compare these methods with existing (both neural and standard statistical) methods. Further, we have discussed only the first principal (resp. canonical correlation) analysis projection; clearly, a Gram-Schmidt reduction is possible but it is an open question as to whether the reinforcement learning paradigm might facilitate a different approach to subsequent projections.

Given the early success of the method, we are encouraged to investigate more mappings with the method. We will investigate both linear manifold methods such as Independent Component Analysis [3] as well as nonlinear manifolds.

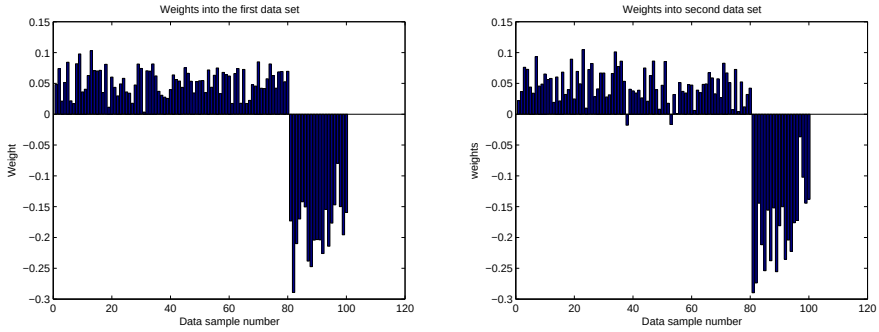

Fig. 2: The weights into the two data streams clearly identify the two clusters in the data set each pair of which has an underlying close but nonlinear relationship. The first 80 samples from both data streams are of one type while the last 20 are of the other type.

References

[1] C. Fyfe and P. L. Lai. Reinforcement learning reward functions for unsupervised learning. In *4th International Symposium on Neural Networks, ISNN 2007*, 2007.

[2] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

[3] P. L. Lai and C. Fyfe. Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems*, 10(5):365–377, 2001.

[4] A. Likas. A reinforcement learning approach to on-line clustering. *Neural Computation*, 1999.

[5] B. Scholkopf, A. Smola, and K.-R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[6] V. Vapnik. *The nature of statistical learning theory*. Springer Verlag, New York, 1995.

[7] R. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.