ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 1-6

Efficient Formation of a Basis in a Kernel Induced Feature Space

Gavin C. Cawley and Nicola L. C. Talbot

School of Information Systems
University of East Anglia
Norwich, U.K. NR4 7TJ
gcc@sys.uea.ac.uk

Abstract.
Baudat and Anouar [1] propose a simple greedy algorithm for estimation of an approximate basis of the subspace spanned by a set of fixed vectors embedded in a kernel induced feature space. The resulting set of basis vectors can then be used to construct sparse kernel expansions for classification and regression tasks. In this paper we describe five algorithmic improvements to the method of Baudat and Anouar, allowing the construction of an approximate basis with a computational complexity that is independent of the number of training patterns, depending only on the number of basis vectors extracted.

Non-linear variants of many well known linear statistical methods have been proposed, based on the so called "kernel trick" (for an overview, see e.g. Cristianini and Shawe-Taylor [3]). The data, $\{\boldsymbol{x}_i \in \mathcal{X}\}_{i=1}^{\ell}$, are projected into a fixed feature space $\mathcal{F}(\boldsymbol{\phi} : \mathcal{X} \rightarrow \mathcal{F})$, induced by a positive definite "Mercer" kernel defining the inner product $\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}(\boldsymbol{x}) \cdot \boldsymbol{\phi}(\boldsymbol{x}')$. A linear method can then be implemented in $\mathcal{F}$, provided that it can be formulated such that the data only appear in the form of inner products, giving rise to a non-linear method in $\mathcal{X}$, the space spanned by the data. The subspace of $\mathcal{F}$ spanned by the data is fully specified by the kernel matrix $\boldsymbol{K} = [k_{ij} = \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{i,j=1}^{\ell}$. As a result, kernel methods can become computationally demanding as the size of the data set grows large. Baudat and Anouar [1] suggest building a kernel model using a small sub-set of the data forming an approximate basis for the subspace of $\mathcal{F}$ populated by the data and provide a greedy algorithm for identifying a near optimal set of basis vectors. In this paper we describe five algorithmic improvements to the the existing method, resulting in a significant reduction in computational complexity.

The normalised Euclidean distance between the image of a datum in feature space, $\boldsymbol{\phi}(\boldsymbol{x}_i)$, and $\hat{\boldsymbol{\phi}}_S(\boldsymbol{x}_i)$, its optimal reconstruction using the set basis vectors

$\{\boldsymbol{\phi}(\boldsymbol{x}_i)\}_{i\in\mathcal{S}}$, is given by

$$\delta_i(\mathcal{S}) = \frac{\|\boldsymbol{\phi}(\boldsymbol{x}_i) - \hat{\boldsymbol{\phi}}_{\mathcal{S}}(\boldsymbol{x}_i)\|^2}{\|\boldsymbol{\phi}(\boldsymbol{x}_i)\|^2}.$$

This distance can be expressed in terms of inner products, and so via the "kernel trick",

$$\delta_i(\mathcal{S}) = 1 - \frac{\boldsymbol{K}_{\mathcal{S}i}^T \boldsymbol{K}_{\mathcal{S}\mathcal{S}}^{-1} \boldsymbol{K}_{\mathcal{S}i}}{k_{ii}}.$$

where $\boldsymbol{K}_{\mathcal{S}\mathcal{S}}$ is a square sub-matrix of $\boldsymbol{K}$, such that $\boldsymbol{K}_{\mathcal{S}\mathcal{S}} = \{k_{mn}\}_{m,n\in\mathcal{S}}$ and $\boldsymbol{K}_{\mathcal{S}i} = (k_{mi})_{m\in\mathcal{S}}^T$ is a column vector of inner products. To form a basis, it is sufficient to minimise the mean reconstruction error δ_i over all patterns in the training set, i.e. maximise

$$J(\mathcal{S}) = \frac{1}{\ell}\sum_{i=1}^{\ell} \frac{\boldsymbol{K}_{\mathcal{S}i}^T \boldsymbol{K}_{\mathcal{S}\mathcal{S}}^{-1} \boldsymbol{K}_{\mathcal{S}i}}{k_{ii}}. \tag{1}$$

Starting with $\mathcal{S} = \emptyset$, a basis is constructed in a greedy manner, adding the training vector maximising $J(\mathcal{S})$ at each iteration. The algorithm terminates when $\boldsymbol{K}_{\mathcal{S}\mathcal{S}}$ is no longer invertible, indicating that a basis has been identified.

1 An Efficient Algorithm

This section describes five improvements to the algorithm proposed by Baudat and Anouar [1], providing a significant reduction in computational complexity.

1.1 Efficient Matrix Inversion

A significant proportion of the computational effort expended in determining a basis is taken up by the repeated inversion of a sub-matrix of $\boldsymbol{K}$ defined by the set of basis vectors, $\boldsymbol{K}_{\mathcal{S}\mathcal{S}}$. Fortunately at each stage an extra row and column is added to a symmetric matrix for which the inverse is already known; this means that the following block matrix inversion lemma can be applied,

Lemma 1 (Block Matrix Inversion) *Given an invertible matrix $\boldsymbol{A}$, a column vector $\boldsymbol{b}$ and scalar c, the block matrix identity*

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{b} \\ \boldsymbol{b}^T & c \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{A}^{-1} + \frac{1}{k}\boldsymbol{A}^{-1}\boldsymbol{b}\boldsymbol{b}^T\boldsymbol{A}^{-1} & -\frac{1}{k}\boldsymbol{A}^{-1}\boldsymbol{b} \\ -\frac{1}{k}\boldsymbol{b}^T\boldsymbol{A}^{-1} & \frac{1}{k} \end{bmatrix},$$

holds, where $k = c - \boldsymbol{b}^T\boldsymbol{A}^{-1}\boldsymbol{b}$ is the Schur complement of $\boldsymbol{A}$.

Repeated $\mathcal{O}(n^3)$ matrix inversion operations are then replaced by an efficient $\mathcal{O}(n^2)$ alternative . To prevent numerical errors accumulating, $\boldsymbol{K}_{\mathcal{S}\mathcal{S}}^{-1}$ can optionally be evaluated directly at the end of each iteration. Note that k can be computed efficiently from the vector reconstruction errors cached during the previous iteration.

1.2 Eliminating Candidate Basis Vectors

If the addition of a candidate vector, $\boldsymbol{\phi}(\boldsymbol{x}_i)$, to the existing set of basis vectors, $\{\boldsymbol{\phi}(\boldsymbol{x}_i)\}_{i\in\mathcal{S}}$, causes $\boldsymbol{K}_{\mathcal{SS}}$ to become singular, it can be safely eliminated from the set of candidate basis vectors for subsequent iterations. In this case, $\boldsymbol{\phi}(\boldsymbol{x}_i)$ is linearly dependent on the set of established basis vectors and so does not reduce the reconstruction error for any pattern, regardless of any further vectors added to $\mathcal{S}$. The Schur complement (see Lemma 1) provides an efficient indication that a candidate vector is linearly dependent on the existing basis vectors. The second term, $\boldsymbol{K}_{\mathcal{S}i}^T\boldsymbol{K}_{\mathcal{SS}}\boldsymbol{K}_{\mathcal{S}i}$, gives the squared norm of the projection of $\boldsymbol{\phi}(\boldsymbol{x}_i)$ in the space spanned by the existing basis vectors $\{\boldsymbol{\phi}(\boldsymbol{x}_j)\}_{j\in\mathcal{S}}$ and the first term, k_{ii}, is the squared norm of $\boldsymbol{\phi}(\boldsymbol{x}_i)$; thus the Schur complement is zero if $\boldsymbol{\phi}(\boldsymbol{x}_i)$ is linearly dependent.

1.3 Efficient Computation of Mean Reconstruction Error

Any vector lying within the subspace defined by the current basis vectors (i.e. the current basis vectors, and any candidate vector for which the Schur complement (see Lemma 1) is sufficiently small), need not be included in the sum defining $J(\mathcal{S})$. This is because the reconstruction error for such patterns is essentially zero, regardless of the addition of further basis vectors, and so the corresponding terms of the summation need not be evaluated.

1.4 Stochastic Sampling of Candidate Basis Vectors

Rather than considering the entire set of candidate basis vector at each step, we consider only a random subset of a fixed size.

Lemma 2 (Maximum of Random Variables) *Denote by $\xi_1, \ldots, \xi_m$ identically distributed independent random variables with the common cumulative distribution function $F(\xi_i)$. Then the cumulative distribution function of $\xi := \max_{i\in[m]} \xi_i$ is $(F(\xi))^m$* (Smola and Schölkopf [4]).

Thus a random selection of 59 candidates, with a probability of 0.95, contains a vector with a mean reconstruction error within the lowest 5% of the entire pool of candidate basis vectors (assuming that the mean relative reconstruction errors are uniformly distributed over the range [0, 1]) [4].

1.5 Stochastic Approximation of $J(\mathcal{S})$

Following the modification described section 1.3, evaluation of the objective function (1) has a computational complexity of $\mathcal{O}(Nn^2)$, where N is the number of candidate basis vectors and n is the number of existing basis vectors. However, if ℓ is large, the following $\mathcal{O}(n^2)$ approximation can be employed:

$$J(\mathcal{S}) \approx \frac{1}{|\mathcal{T}|}\sum_{i\in\mathcal{T}} \frac{\boldsymbol{K}_{\mathcal{S}i}^T\boldsymbol{K}_{\mathcal{SS}}^{-1}\boldsymbol{K}_{\mathcal{S}i}}{k_{ii}}.$$

where $\{\boldsymbol{\phi}(\boldsymbol{x}_i)\}_{i \in \mathcal{T}}$ is a random subset of candidate basis vectors of a sufficiently large (but fixed) size.

1.6 The Algorithm

The efficient feature vector selection algorithm can be stated formally as follows:

Algorithm 1 Efficient Feature Vector Selection

```
Inputs:  $\boldsymbol{K}$ – the Gram matrix, $\boldsymbol{K} = [k_{ij} = \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j]_{i,j=1}^{\ell}$
         $\epsilon$ – threshold reconstruction error for candidate vectors
         $\rho$ – number of candidate vectors considered in each iteration
         $\upsilon$ – number of candidate vectors used to approximate $J(\mathcal{S})$
         $\tau$ – stopping criterion based on mean relative reconstruction error
         $N$ – stopping criterion based on maximum number of basis vectors
Method:
    $\mathcal{S} := \emptyset$.
    $\mathcal{C} := \{1, \ldots, \ell\}$.
    do
        Let $\mathcal{R} \underset{\text{random}}{\subset} \mathcal{C}$, such that $|\mathcal{R}| = \min(\rho,\ |\mathcal{C}|)$.
        Let $\mathcal{T} \underset{\text{random}}{\subset} \mathcal{C}$, such that $|\mathcal{T}| = \min(\upsilon,\ |\mathcal{C}|)$.
        foreach $r \in \mathcal{R}$
            $\mathcal{S}' := \mathcal{S} \cup \{r\}$.
            Invert $\boldsymbol{K}_{\mathcal{S}'\mathcal{S}'}$ via block matrix inversion lemma, (Lemma 1).
            Evaluate and cache $\mathcal{J}(\mathcal{S}') = \frac{1}{|\mathcal{T}|} \sum_{i \in \mathcal{T}} \frac{\boldsymbol{K}_{\mathcal{S}'i}^T \boldsymbol{K}_{\mathcal{S}'\mathcal{S}'}^{-1} \boldsymbol{K}_{\mathcal{S}'i}}{k_{ii}}$.
        end
        Find $r \in \mathcal{R}$ such that $\mathcal{J}(\mathcal{S} \cup \{r\}) = \max_{r' \in \mathcal{R}} \mathcal{J}(\mathcal{S} \cup \{r'\})$.
        $\mathcal{S} := \mathcal{S} \cup \{r\}$.
        $\mathcal{C} := \mathcal{C} \ \setminus \ \{c \ : \ c \in \mathcal{C},\ \delta_c(\mathcal{S}) < \epsilon\}$
    while $|\mathcal{S}| < N$ and $(1 - \mathcal{J}(\mathcal{S})) > \tau$ and $\mathcal{C} \neq \emptyset$.
Output: $\mathcal{S}$.
```

2 Results

The standard and efficient feature vector selection algorithms (FVS and EFVS respectively) were implemented in MATLAB, with vectorisation and MEX files used where appropriate, making extensive use of profiling information. The results of a comparison of the FVS and EFVS algorithms over the Boston Housing data set [2] are displayed in figure 1. As shown in figure 1 (a), reducing the number of candidate basis vectors considered during each iteration greatly reduces the run time of the efficient feature vector selection algorithm. Figure 1 (b) indicates that this is achieved without significantly reducing the quality of the resulting set of basis vectors as measured by the mean relative reconstruction error. Note even if all candidates are examined ($\rho = \infty$), the efficient FVS algorithm is still significantly faster than the conventional algorithm. Figure 1 (c) and (d) indicates that a similar reduction in run-time can also be achieved by reducing the number of candidate basis vectors used to estimate the objective function, again without a significant sacrifice in mean relative reconstruction error.

3 Summary

This paper describes an efficient algorithm for selecting a subset of the data set forming a basis for the entire data set in a kernel induced feature space. A sparse kernel machine can then be constructed using the set of basis vectors. The proposed modifications reduce the computational complexity of the existing algorithm from $\mathcal{O}(\ell^2 n^2)$ to only $\mathcal{O}(n^3)$, where ℓ is the number of training patterns and n is the number of basis vectors extracted. Note that the run-time of the algorithm depends only indirectly on the number of training patterns, and so efficient feature vector selection enables non-sparse algorithms, such as the Least-Squares Support Vector Machine [5], to be employed in large scale applications.

4 Acknowledgements

The authors thank Rob Foxall, Richard Harvey and the anonymous reviewers for helpful comments on previous drafts of this manuscript. This work was supported by Royal Society research grant RSRG-22270.

References

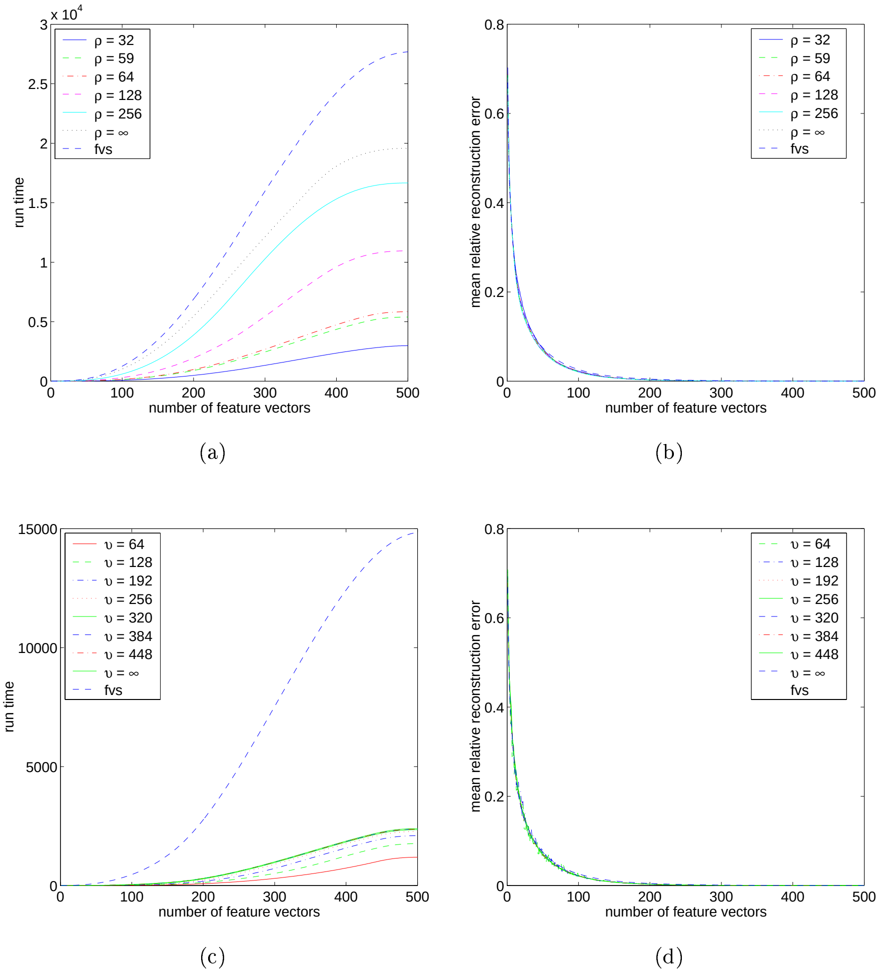
[1] G. Baudat and F. Anouar. Kernel-based methods and function approximation. In *Proc. IJCNN*, pages 1244–1249, Washington, DC, July 2001.

[2] S. D. Bay. The UCI KDD archive [`http://kdd.ics.uci.edu/`]. University of California, Department of Information and Computer Science, Irvine, CA, 1999.

[3] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines (and other kernel-based learning methods)*. Cambridge University Press, Cambridge, U.K., 2000.

[4] A. J. Smola and B. Schölkopf. Sparse greedy matrix approximation for machine learning. In *Proc. ICML*, 1999.

[5] J. A. K. Suykens, J. De Brabanter, L. Lukas, and J. Vandewalle. Weighted least squares support vector machines : robustness and sparse approximation. *Neurocomputing*, 2001.



(a)

(b)

(c)

(d)

Figure 1: Comparison of standard and efficient feature vector selection algorithms over the Boston Housing data set : (a) and (b) run time and mean relative reconstruction error as a function of the number of basis vectors as ρ, the number of candidates examined in each iteration, is varied; (c) and (d) run time and mean relative reconstruction error as a function of the number of basis vectors as υ, the number of vectors used to estimate the objective function, is varied ($\rho = 59$).