Compressed Conditional Mean Embeddings for Model-Based Reinforcement Learning - Supplementary Material

Guy Lever
University College London
London, UK
g.lever@cs.ucl.ac.uk

John Shawe-Taylor
University College London
London, UK
j.shawe-taylor@cs.ucl.ac.uk

Ronnie Stafford
University College London
London, UK
r.stafford.12@ucl.ac.uk

Csaba Szepesvári
University of Alberta
Edmonton, Canada
szepesva@cs.ualberta.ca

Supplementary Material

The following sections are supplementary to the published AAAI version of this paper.

1 Additional Details of the Feature Selection Process

1.1 Sparse-Greedy Regression with Vector-Valued Matching Pursuit

Our algorithm uses, as a subcomponent, a vector-valued version of the matching pursuit algorithm (Mallat and Zhang 1993). The use of matching pursuit for vector-valued regression is not new (Lever and Stafford 2015) but we provide details for completeness. This adaptation can handle targets in general vector spaces $\mathcal{V}$. Since the algorithm only computes inner products between vectors in the target space $\mathcal{V}$ it can be kernelized, i.e. we can learn an RKHS-valued function. This is a straightforward extension of the scalar case, we derive the method here for clarity.

Suppose we wish to regress a vector-valued function

$$f^* : \mathcal{X} \to \mathcal{V},$$

given a data sample $\mathcal{D} = \{x_i, v_i\}_{i=1}^m$ where $v_i = f^*(x_i) + \epsilon$ where ϵ is zero-mean noise, $f^*(x_i) = \mathbb{E}[V_i|x_i]$. Suppose we are given a *dictionary* $\mathcal{G} = \{g_1, ..., g_n\}$, where $g_i : \mathcal{X} \to \mathbb{R}$, of candidate real-valued functions, and we aim to find an estimate $\hat{f}$ for f^* of the form,

$$\hat{f} = \sum_{i=1}^{D} w^i \hat{g}_i$$

where $\mathcal{B}_D = \{\hat{g}_i\}_{i=1}^D \subseteq \mathcal{G}$ is called the basis and $w^i \in \mathcal{V}$. When $\mathcal{V} = \mathbb{R}$, matching pursuit (Mallat and Zhang 1993) can be used to incrementally build the basis, and we now detail the extension to the vector-valued output case. We build the basis incrementally and for each basis $\mathcal{B}_j$ we form an estimate $\hat{f}^j = \sum_{i=1}^{j} w^i \hat{g}_i$. We begin with the empty basis $\mathcal{B}_0$ and add new basis elements $\hat{g}_{j+1}$ to greedily optimize the objective. For each estimate we define the residue $\boldsymbol{r}^j$,

$$r_i^j = v_i - \hat{f}_j(x_i) \in \mathcal{V},$$

and pick the $g \in \mathcal{D}$ which minimizes the next residue when added to the current estimate,

$$g_{j+1} = \operatorname*{argmin}_{g \in \mathcal{D}} \min_{w \in \mathcal{V}} \sum_{i=1}^{m} ||v_i - ((\hat{f}_j + wg)(x_i))||_{\mathcal{V}}^2$$
$$= \operatorname*{argmin}_{g \in \mathcal{D}} \min_{w \in \mathcal{V}} \sum_{i=1}^{m} ||r_i^j - wg(x_i)||_{\mathcal{V}}^2.$$

Since $\nabla_w \sum_{i=1}^m ||r_i^j - wg(x_i)||_{\mathcal{V}}^2 = 0$ at the minimum we have,

$$0 = \sum_{i=1}^{m} \nabla_w \left(\langle g(x_i)w, g(x_i)w \rangle_{\mathcal{V}} - 2\langle g(x_i)w, r_i^j \rangle_{\mathcal{V}} \right)$$
$$= \sum_{i=1}^{m} 2wg(x_i)^2 - 2g(x_i)r_i^j$$
$$w^{j+1} = \left(\sum_{i=1}^{m} g(x_i) r_i^j \right) / \left(\sum_{i=1}^{m} g(x_i)^2 \right) \in \mathcal{V}$$

Then,

$$\sum_{i=1}^{m} ||r_i^j - w^{j+1} g(x_i)||_{\mathcal{V}}^2$$
$$= \sum_{i=1}^{m} ||r_i^j||_{\mathcal{V}}^2 - 2\sum_{i=1}^{m} g(x_i) \langle r_i^j, w^{\min} \rangle_{\mathcal{V}}$$
$$+ ||w^{\min}||_{\mathcal{V}}^2 \sum_{i=1}^{m} g(x_i)^2$$
$$= \sum_{i=1}^{m} ||r_i^j||_{\mathcal{V}}^2 - \frac{2\sum_{i=1}^m g(x_i) \langle r_i^j, \sum_{k=1}^m g(x_k) r_k^j \rangle_{\mathcal{V}}}{\sum_{k=1}^m g(x_k)^2} \quad (1)$$
$$+ \frac{||\sum_{k=1}^m g(x_k) r_k^j||_{\mathcal{V}}^2}{\sum_{i=1}^m g(x_i)^2}$$
$$= \sum_{i=1}^{m} ||r_i^j||_{\mathcal{V}}^2 - \frac{||\sum_{i=1}^m g(x_i) r_i^j||_{\mathcal{V}}^2}{\sum_{i=1}^m g(x_i)^2}$$

Thus $\hat{g}_{j+1} = \operatorname{argmax}_{g \in \mathcal{G}} \frac{||\sum_{i=1}^m g(x_i) r_i^j||_{\mathcal{V}}^2}{\sum_{i=1}^m g(x_i)^2}$. Thus at each iteration of matching pursuit we must evaluate $\frac{||\sum_{i=1}^m g(x_i) r_i^j||_{\mathcal{V}}^2}{\sum_{i=1}^m g(x_i)^2}$ for a selection of k dictionary elements (not necessarily all). We have,

$$||\sum_{i=1}^{m} g(x_i) r_i^j||_{\mathcal{V}}^2 = ||\sum_{i=1}^{m} g(x_i)(\hat{f}^j(x_i) - v_i)||_{\mathcal{V}}^2$$

For each dictionary element g this can be computed in $O(mj + md + jd)$ where $d = \dim(\mathcal{V})$, and so $O(k(mj + md + jd))$ over k dictionary elements.

It is sometimes useful, at iteration j to "backfit" all the weights $\{w^i\}_{i=1}^j$ by replacing them with the least squares solution: i.e. matching pursuit is used to find the basis but the weights are finally optimized using least squares. Alternatively this can be performed end of the process or several times throughout.

In order to find a compact representation we can also use matching pursuit adaptively by setting a tolerance δ such that the algorithm terminates when it fails to reduce the residue by more than δ. Thus the method will only add features if they significantly reduces the objective.

The output of vector valued matching pursuit is a collection of weights $\{w^i\}_{i=1}^j$ and features $\{\hat{g}_i(\cdot)\}_{i=1}^j$ such that $f^* \approx \sum_{i=1}^j w^i \hat{g}_i$.

Fast Feature Selection For RKHS-Valued Matching Pursuit Performing matching pursuit to regress a function $f : \mathcal{S} \times \mathcal{A} \to \mathcal{F}_L$ where $\mathcal{F}_L$ is an RKHS whose feature map $\phi(s) = L(s, \cdot)$ is high- or even infinite-dimensional can become expensive when the number of data points becomes large. This is because the expansion of each target residue for matching pursuit will have an expansion in the RKHS $\mathcal{F}_L$ in the number of data points, and so computing a single inner product scales with the size of the data, and feature selection by this method would be the bottleneck in our Algorithm. To solve this problem we map the target data for matching pursuit, i.e. the residues $\mathcal{R}(s_i, a_i, s_i') \in \mathcal{F}_L$ for $(s_i, a_i, s_i') \in \mathcal{D}$, into a lower dimensional subspace of $\mathcal{F}_L$ using an incomplete Cholesky decomposition of the kernel matrix $\boldsymbol{L}$ (Shawe-Taylor and Cristianini 2004). This provides an approximation $\boldsymbol{L} \approx \boldsymbol{R}^\top \boldsymbol{R}$ of the $n \times n$ matrix $\boldsymbol{L}$ where $\boldsymbol{R}$ is a $p \times n$ matrix $\boldsymbol{R} = (\boldsymbol{r}_1, ..., \boldsymbol{r}_n)^\top$, where $p \ll n$ can be chosen or chosen adaptively using a tolerance parameter. This approximation is often excellent (Bach and Jordan 2005) and captures the inner products in $\mathcal{F}_L$ since $\boldsymbol{r}_i^\top \boldsymbol{r}_j \approx L_{ij} = \langle \phi(s_i'), \phi(s_j') \rangle_L$, thus we can "project" the high-dimensional feature vectors $\phi(s_i')$ to $\mathbb{R}^p$ via $\phi(s_i') \mapsto \boldsymbol{r}_i$, approximately preserving inner products. We then have the following approximation for the loss function,

$$\begin{aligned}\hat{\text{loss}}_\lambda(\boldsymbol{W}) &= \frac{1}{n}\sum_{i=1}^{n} ||\sum_{j=1}^{n} \psi(s_i, a_i)^\top \boldsymbol{w}_j \phi(s_j) - \phi(s_i)||_{\mathcal{F}}^2 \\ &\approx \frac{1}{n}\sum_{i=1}^{n} ||\sum_{j=1}^{n} \psi(s_i, a_i)^\top \boldsymbol{w}_j \boldsymbol{r}_j^\top - \boldsymbol{r}_i^\top||^2 \qquad (2)\end{aligned}$$

Thus if we define the following "projection" of the model residues

$$\boldsymbol{q}_i := \boldsymbol{r}_i^\top - \sum_{j=1}^{n_k} \psi^k(s_i, a_i)^\top \boldsymbol{w}_k^j \boldsymbol{r}_j^\top \in \mathbb{R}^p, \qquad (3)$$

and we perform vector-valued matching pursuit using the approximate, low dimensional residue data $\{(s_i, a_i), \boldsymbol{q}_i\}_{i=1}^n$ in p-dimensional Euclidean space we will find find a feature $\psi^{\text{new}}(\cdot)$ and a weight $\boldsymbol{b}^{\text{new}} = \sum_{j=1}^n w_j^{\text{new}} \boldsymbol{r}_j^\top$ such that

$$\sum_{i=1}^{n} ||\psi^{\text{new}}(s_i, a_i)^\top \boldsymbol{b}^{\text{new}} + \sum_{j=1}^{n} \psi(s_i, a_i)^\top \boldsymbol{w}_j \boldsymbol{r}_j^\top - \boldsymbol{r}_i^\top||^2$$

is minimized w.r.t $\psi^{\text{new}}(\cdot) \in \mathcal{G}$ and $\boldsymbol{b}^{\text{new}} \in \mathbb{R}^p$. But then $\psi^{\text{new}}(\cdot)$ and a $b^{\text{new}} := \sum_{j=1}^n w_j^{\text{new}} \phi(s_j')$ are (approximately) optimally greedy for $\hat{\text{loss}}_\lambda(\boldsymbol{W})$ in the sense that

$$\sum_{i=1}^{n} ||\psi^{\text{new}}(s_i, a_i)^\top b^{\text{new}} + \sum_{j=1}^{n} \psi(s_i, a_i)^\top \boldsymbol{w}_j \phi(s_j) - \phi(s_i)||_{\mathcal{F}}^2$$

is minimized w.r.t. $\psi^{\text{new}}(\cdot) \in \mathcal{G}$ and $b^{\text{new}} \in \mathcal{F}$. i.e. performing matching pursuit on the low dimensional residues is an equivalent problem up to approximation error of the Cholesky decomposition and the same features will be returned if this decomposition is accurate. In this way d_{new} new features can be added in time $O(d_{\text{new}} p |\mathcal{G}||\mathcal{D}|)$, where p is the rank of the incomplete Cholesky decomposition. In our experiments p was set to $p = 200$ by computational budget and the kernel approximation was found to be almost perfect.

References

Bach, F. R., and Jordan, M. I. 2005. Predictive low-rank decomposition for kernel methods. In *ICML 2005*, 33–40.

Lever, G., and Stafford, R. 2015. Modelling policies in mdps in reproducing kernel hilbert space. In *AISTATS 2015*.

Mallat, S., and Zhang, Z. 1993. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing* 41(12):3397–3415.

Shawe-Taylor, J., and Cristianini, N. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press.