A Framework for Reduce Set Methods using Random Features

Chirag Gupta, Krikamol Muandet, Ilya Tolstikhin, Bernhard Schölkopf

Abstract

We re-formulate the Reduced Set problem [2] as one that seeks to approximate well to a single vector in low dimension instead of a high-dimensional RKHS expansion. The low dimensional representation is based on random features [5]. This opens up new possibilities for faster methods to solve an optimization problem that is non-convex and hitherto did not yield straightforward visualizations. In doing so, we also make way for better understanding of the Mercer kernel map, in spirit with [6].

1 Introduction

Constructing a Reduced Set for a linear combination of RKHS images of points in input space (referred to henceforth as an RKHS expansion) involves optimization of a typically non-convex objective in the RKHS. Of particular interest is the pre-image problem which asks for a single point in the input space whose RKHS image best represents the RKHS expansion. A classical approach uses the pre-image solution as a black box to solve the general Reduced Set problem [6].
We propose a novel formulation that proxies the optimization in the RKHS to an optimization in linear space, based on the use of random feature. As proposed in [5] we can use Bochner's theorem to inflict a (statistical) transformation from the RKHS into linear space while preserving approximate inner products and distances. This map is continuously consistent in the feature space, preserving relations not just between points in feature space whose projections have been evaluated, but for points in a continuous ball in the feature space. Consequently, 'hallucinated' projections in the projection space correspond to geometrically consistent pre-images in feature space. The optimization can now be performed in this projection space, within the subspace of the projection space that contains a valid pre-image in the RKHS.
The organization of this report is as follows. Section 2 briefly overviews the current state of the art in Reduced Set methods. Section 3 talks about random feature. Section 4 contains a description of our novel formulation.Section 5 presents results on a synthetic data-set. Finally, section 6 concludes with a philosophical discussion on the use of our technique, describing where it can be applied and where it cannot.

2 Reduced Set Methods

A typical shortcoming encountered when dealing with kernel methods is potentially large classification times due to the fact that RKHS inner products for the test point need to be evaluated with each of the input points in the set of support vectors. The use of a smaller expansion that well approximates the larger one was proposed in [2], trading classification accuracy for time.
Given a kernel $k(x, \cdot)$ and N points in input space (say $\mathbb{R}^d$), $x_{i=1}^N$, along with their expansion coefficients, $\alpha_{i=1}^N \in \mathbb{R}$, and $n \ll N$, the problem is stated as identifying $\hat{x}_{j=1}^n \in \mathbb{R}^d$, and expansion coefficients $\beta_{j=1}^n \in \mathbb{R}$ that minimize

$$\| \sum_{i=1}^{N} \alpha_i \phi(x_i) - \sum_{j=1}^{n} \beta_j \phi(\hat{x}_j) \|_{RKHS} \tag{1}$$

Here ϕ represents the (implicit) RKHS mapping, so that $\phi(x) = k(x, \cdot)$. Also, henceforth, we use the notation $K(\alpha, x)$ to mean $\sum_{i=1}^{N} \alpha_i \phi(x_i)$.

The classical approach to solve reduced set problems involves a black-box solution for the pre-image problem, which looks for the single point in input space x, whose image in the RKHS (possible multiplied with a constant factor) $\beta\phi(x)$ well approximates $k(\alpha, x)$. The solution is used iteratively n times, each time the *residual* element of the expansion being approximated. A fixed-point iteration method is available to solve the pre-image problem for RBF kernels [6]. The optimization problem however is non-convex, and problems of local minima have been reported by [3, 1]. Alternate methods suggested by these perform well for the pre-image problem, but not for the reduced problem, as reported in [9]. We look to re-formulate the entire problem and state it with a simpler objective, opening up possibilities for better solutions and understanding of the problem.

3 Random features

For a shift-invariant kernel $k(x, y) = k(x - y)$, it can be shown [5] that for any $x, y \in \mathbb{R}^d$,

$$k(x, y) = 2 \mathop{\mathbb{E}}_{w,b} [cos(w \cdot x + b) cos(w \cdot y + b)], \tag{2}$$

where $w \sim D_k$ and $b \sim U[0, 2\pi]$ with D_k being the Fourier transform of k. (which for the case of the gaussian kernel $k(x, y) = exp(-\gamma \|x - y\|^2)$ is $N(0, 2\gamma I_d)$).

We now sample over the distributions to obtain m samples (w_i, b_i). The RKHS expansion can now be projected to $\mathbb{R}^m$ -

$$\boldsymbol{P}^k_{\mathbf{w},\mathbf{b}} : (\mathbb{R}^N, \mathbb{R}^{N \times d}) \to \mathbb{R}^m, \tag{3}$$

given by,

$$\boldsymbol{P}(\alpha \in \mathbb{R}^N, x \in \mathbb{R}^{N \times d}) = \sqrt{2} \left[\sum_{i=1}^{N} \alpha_i cos(w_k \cdot x_i + b_k) \right]_{k=1}^{m} \in \mathbb{R}^m \tag{4}$$

Inner product evaluations are now much easier,

$$\sum_{i=1}^{N} \sum_{j=1}^{n} \alpha_i \beta_j k(x_i, y_j) \approx \boldsymbol{P}(\alpha, x) \cdot \boldsymbol{P}(\beta, y) \tag{5}$$

Clearly, the computation time goes down from $O(Nnd)$ to $O(m)$.

4 Reduced Set through Random Features

The projection $\boldsymbol{P}^k_{\mathbf{w},\mathbf{b}}$ effectively evaluates a low dimensional linear embedding of the high-dimensional RKHS object. We propose to re-formulate the reduced set problem in this space, hence allowing us to deal with the optimization problem in a familiar setting, utilizing standard results and techniques from Optimization Theory.

We have from the discussion in previous sections, $K(\alpha, x) \approx \boldsymbol{P}(\alpha, x)$. We now look to find $\hat{x}^n_{j=1} \in \mathbb{R}^d$, and $\beta^n_{j=1} \in \mathbb{R}$, such that,

$$\boldsymbol{P}(\alpha, x) \approx \boldsymbol{P}(\beta, \hat{x}) \tag{6}$$

The formulation in the space $\mathbb{R}^m$ looks like so.

$$\beta^*, \hat{x}^* = \mathop{argmin}_{\beta, \hat{x}} \left(2 \cdot \left\| \left[\sum_{i=1}^{N} \alpha_i cos(w_k \cdot x_i + b_k) \right]_{k=1}^{m} - \left[\sum_{j=1}^{n} \beta_j cos(w_k \cdot \hat{x}_j + b_k) \right]_{k=1}^{m} \right\|_2^2 \right) \tag{7}$$

We define $\hat{\mu}$ as follows -

$$\hat{\mu} = \left[\sum_{i=1}^{N} \alpha_i cos(w_k \cdot x_i + b_k) \right]_{k=1}^{m} \tag{8}$$

Now for the BFGS routine, we have functions that return the objective f, and a gradient vector with respect to each of the $x_i's$.

$$f = \left\| \hat{\mu} - \left[\sum_{j=1}^{n} \beta_j cos(w_k \cdot \hat{x}_j + b_k) \right]_{k=1}^{m} \right\|_2^2 \tag{9}$$

$$\nabla_{\hat{x}_j} = \sum_{k=1}^{m} \left(2\beta_j sin(w_k \cdot \hat{x}_j + b_k) \left[\hat{\mu}_k - \sum_{j=1}^{n} \beta_j cos(w_k \cdot \hat{x}_j + b_k) \right] \right) \cdot w_k \tag{10}$$

The optimization problem is now over $\hat{x}$ as well as β. The optimization in terms of β turns out to be simply a least squares solution, given that the $\hat{x}$ is fixed. For this, define Φ, a matrix of dimension $m \times n$, where

$$\Phi_{k,j} = cos(w_k \cdot \hat{x}_j + b_k), \tag{11}$$

so that

$$\Big(\Phi \cdot \beta \Big)_k = \sum_{j=1}^{n} \beta_j cos(w_k \cdot \hat{x}_j + b_k) \tag{12}$$

The objective can now be formulated as

$$f = \|\hat{\mu} - \Phi \cdot \beta\|_2^2 \tag{13}$$

Clearly, the solution for β is given by,

$$\beta_{opt}\hat{x} = (\Phi\Phi^T + \lambda\mathbb{I})^{-1}\Phi\hat{\mu} \tag{14}$$

regularized by the parameter λ to avoid numerical instabilities.
The optimization proceeds in alternating steps of optimizing over $\hat{x}$ and β. For optimizing over $\hat{x}$, we use BFGS feeding in the objective and the gradients explicitly. The optimization proceeds in alternating steps of optimizing over β and x since different solution are given for each. Also, to avoid local minima, multiple random initializations are used and the one that gives the best objective is used.

Randomly initialize $\hat{x}$, β ;

while *Change in objective >threshold* **do**

$\hat{x} = BFGS(\alpha_{1\ldots N}, x_{1\ldots N}, \beta_{1\ldots n})$;

$\beta = \beta_{opt}(\alpha_{1\ldots N}, x_{1\ldots N}, \hat{x}_{1\ldots n})$;

end

Algorithm 1: Reduced Set through Random Features (RSRF)

As of now, our current approach does not utilize any objective specific heuristics that could potentially be used in this framework.

5 Empirical simulations

5.1 Kernel Mean Embedding

The kernel mean embedding draws a mapping from distribution to a single RKHS point via a sample drawn from the distribution $x_i{}_{i=1}^{N}$, and a specified kernel k. [8], given by

$$\mu[X] = \frac{1}{m} \sum_{i=1}^{N} \phi(x_i) \tag{15}$$

Inspired from [4], we look to approximate this kernel mean embedding in a reduced set expansion.
Synthetic Data Set. 2-dimensional data is generated from a mixture of 3 Gaussian distributions.

For each of the Gaussians, the standard deviations (diagonal elements of the covariance matrix) and means are randomly selected from other fixed Gaussian distributions. The non-diagonal covariates are set to zero. Three random weights are selected for the Gaussians, and then their sum is normalized to 1. See Figure 5.1 for the data points drawn and contours for the kernel mean embedding function.
For our experiments, $N = 1000$. We seek to approximate the kernel mean embedding of this distribution by $n = 1 \dots 20$ points.

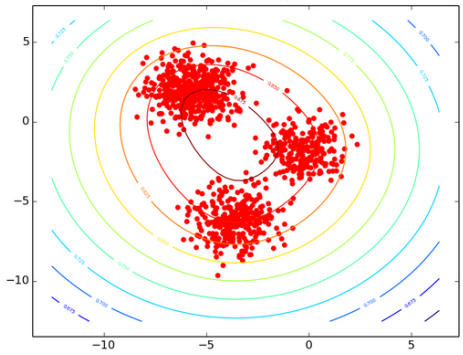

Figure 1: Contours for $\mu = \frac{1}{m} \sum_{i=1}^{N} \phi(x_i)$.

5.2 Results

We used $m = 100$ random features, which empirically gave good performance. However, it was seen that the performance was only comparable to the objective obtained by the classical algorithm [6] (see Figure 4), whereas in terms of time, our algorithm was much slower (see Figure 5). The contours were also not that good (compare 2 against 3), in particular one can see no clear resemblance to the original contours (Figure 5.1). New techniques and ideas need to be explored to make this approach more useful and stable.

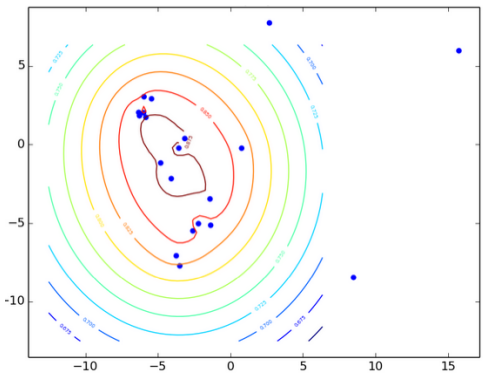

Figure 2: Contours for reduced set using proposed approach

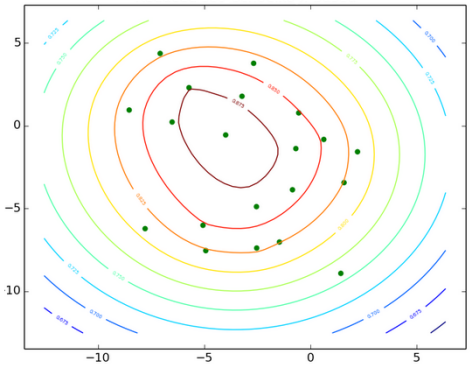

Figure 3: Contours for reduced set using classical approach

6 Discussion

The original motivation for the reduced set problem was to reduce test time. Using random features, we can project into a linear space where test times already go down from $O(Nd)$ to $O(m)$. What then would be the reason to even have a reduced set expansion anymore?

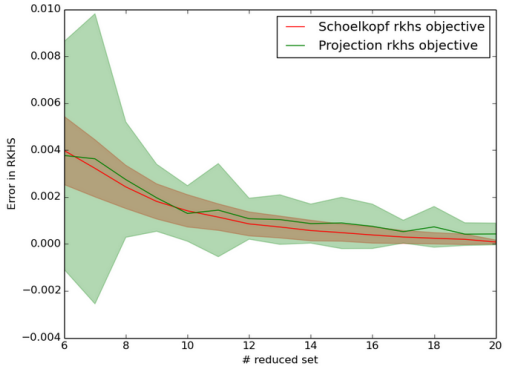

Figure 4: Objective against n

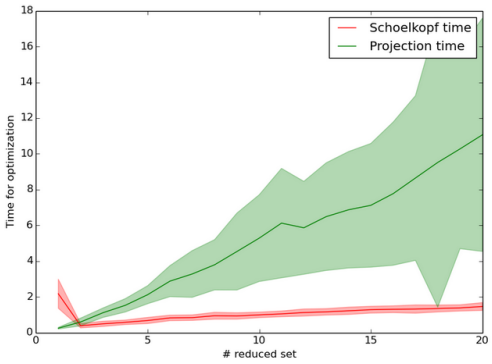

Figure 5: Optimization time against n

The Reduced Set problem is still applicable in areas that don't rely only on inner products and distances of points, but actually need the points in input space explicitly (so that the low-dimensional projections of their images in the RKHS won't do). As an example, the Kernel Probabilistic Programming Framework developed recently in [7] depends on evaluating functions of random variables and cannot be done in projection space.

References

[1] Gokhan H Bakir, Jason Weston, and Bernhard Schölkopf. Learning to find pre-images. *Advances in neural information processing systems*, 16(7):449–456, 2004.

[2] Christopher JC Burges et al. Simplified support vector decision rules. In *ICML*, volume 96, pages 71–77. Citeseer, 1996.

[3] James Tin-Yau Kwok and Ivor Wai-Hung Tsang. The pre-image problem in kernel methods. *Neural Networks, IEEE Transactions on*, 15(6):1517–1525, 2004.

[4] David Lopez-Paz, Krikamol Muandet, Bernhard Schölkopf, and Ilya Tolstikhin. Towards a learning theory of causation. *arXiv preprint arXiv:1502.02398*, 2015.

[5] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2007.

[6] Bernhard Schölkopf, Sebastian Mika, Chris JC Burges, Philipp Knirsch, Klaus-Robert Müller, Gunnar Rätsch, and Alexander J Smola. Input space versus feature space in kernel-based methods. *Neural Networks, IEEE Transactions on*, 10(5):1000–1017, 1999.

[7] Bernhard Schölkopf, Krikamol Muandet, Kenji Fukumizu, and Jonas Peters. Computing functions of random variables via reproducing kernel hilbert space representations. *arXiv preprint arXiv:1501.06794*, 2015.

[8] Alex Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. A hilbert space embedding for distributions. In *Algorithmic Learning Theory*, pages 13–31. Springer, 2007.

[9] Benyang Tang and Dominic Mazzoni. Multiclass reduced-set support vector machines. In *Proceedings of the 23rd international conference on Machine learning*, pages 921–928. ACM, 2006.