Compact Random Feature Maps

Raffay Hamid RAFFAY@CC.GATECH.EDU
eBay Research Laboratory

Ying Xiao YING.XIAO@GATECH.EDU
Georgia Institute of Technology

Alex Gittens AGITTENS@EBAY.COM
eBay Research Laboratory

Dennis DeCoste DDECOSTE@EBAY.COM
eBay Research Laboratory

Abstract

Kernel approximation using random feature maps has recently gained a lot of interest. This is mainly due to their applications in reducing training and testing times of kernel based learning algorithms. In this work, we identify that previous approaches for polynomial kernel approximation create maps that can be rank deficient, and therefore may not utilize the capacity of the projected feature space effectively. To address this challenge, we propose compact random feature maps (CRAFTMaps) to approximate polynomial kernels more concisely and accurately. We prove the error bounds of CRAFTMaps demonstrating their superior kernel reconstruction performance compared to the previous approximation schemes. We show how structured random matrices can be used to efficiently generate CRAFTMaps, and present a single-pass algorithm using CRAFTMaps to learn non-linear multi-class classifiers. We present experiments on multiple standard data-sets with performance competitive with state-of-the-art results.

1. Introduction

Kernel methods allow implicitly learning non-linear functions using explicit linear feature spaces (Schlkopf et al., 1999). These spaces are typically high dimensional and often pose what is called the *curse of dimensionality*. A solution to this problem is the well-known *kernel trick* (Aizerman et al., 1964), where instead of directly learning a classifier in $\mathbb{R}^d$, a non-linear mapping $\Phi : \mathbb{R}^d \to \mathcal{H}$ is considered, where $\forall\, \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, $\langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle_{\mathcal{H}} = \mathrm{K}(\mathbf{x}, \mathbf{y})$ for a kernel $\mathrm{K}(\mathbf{x}, \mathbf{y})$. A classifier $\mathbf{H} : \mathbf{x} \mapsto \mathbf{w}^{\mathrm{T}}\Phi(\mathbf{x})$ is then learned for a $\mathbf{w} \in \mathcal{H}$.

It has been observed however that with increase in training data size, the support of the vector $\mathbf{w}$ can undergo unbounded growth, which can result in increased training as well as testing time (Steinwart, 2003) (Bengio et al., 2006). Previous approaches to address this *curse of support* have mostly focused on embedding the non-linear feature space $\mathcal{H}$ into a low dimensional Euclidean space while incurring an arbitrarily small distortion in the inner product values (Rahimi & Recht, 2007) (Kar & Karnick, 2012) (Le et al., 2013) (Pham & Pagh, 2013). One way to do this is to construct a randomized feature map $\mathbf{Z} : \mathbb{R}^d \to \mathbb{R}^{\mathrm{D}}$ such that for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, $\langle \mathbf{Z}(\mathbf{x}), \mathbf{Z}(\mathbf{y}) \rangle \approx \mathrm{K}(\mathbf{x}, \mathbf{y})$. Each component of $\mathbf{Z}(\mathbf{x})$ can be computed by first projecting $\mathbf{x}$ onto a set of randomly generated d dimensional vectors sampled from a zero-mean distribution, followed by computing the dot-products of the projections. While randomized feature maps can approximate the more general class of dot-product kernels, in this work we focus on polynomial kernels, where $\mathrm{K}(\mathbf{x}, \mathbf{y})$ is of the form $(\langle \mathbf{x}, \mathbf{y} \rangle + q)^r$, with $q \in \mathbb{R}^+$ and $r \in \mathbb{N}_0$.

In previous works, it has been shown that $|\langle \mathbf{Z}(\mathbf{x}), \mathbf{Z}(\mathbf{y}) \rangle - \mathrm{K}(\mathbf{x}, \mathbf{y})|$ reduces exponentially as a function of D (Kar & Karnick, 2012) (Pham & Pagh, 2013). However in practice, to approximate $\mathrm{K}(\mathbf{x}, \mathbf{y})$ with sufficient accuracy, D can still need to be increased to values that might not be amenable to efficiently learn classifiers in $\mathbb{R}^{\mathrm{D}}$. This is especially true for higher values of r.

We also show that spaces constructed by random feature maps can be rank deficient. This rank deficiency can result in the under-utilization of the projected feature space, where the model parameters learned in $\mathbb{R}^{\mathrm{D}}$ can have signif-

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32. Copyright 2014 by the author(s).

icant number of components close to zero.

This presents us with the dilemma between better approximation of exact kernel values and efficient classifier learning. To resolve this dilemma, we propose compact random feature maps (CRAFTMaps) as a more concise representation of random feature maps that can approximate polynomial kernels more accurately. We show that the information content of $\mathbf{Z} : \mathbb{R}^d \rightarrow \mathbb{R}^{\mathrm{D}}$ can be captured more compactly by generating an alternate random feature map $\mathbf{G} : \mathbb{R}^{\mathrm{D}} \rightarrow \mathbb{R}^{\mathrm{E}}$, such that $\mathrm{E} < \mathrm{D}$, and $\langle \mathbf{G}(\mathbf{Z}(\mathbf{x})), \mathbf{G}(\mathbf{Z}(\mathbf{y})) \rangle$ approximates $\langle \mathbf{Z}(\mathbf{x}), \mathbf{Z}(\mathbf{y}) \rangle$. CRAFTMaps are therefore constructed by first up projecting the original data non-linearly to $\mathbb{R}^{\mathrm{D}}$ in order to minimize $|\langle \mathbf{Z}(\mathbf{x}), \mathbf{Z}(\mathbf{y}) \rangle - \mathrm{K}(\mathbf{x}, \mathbf{y})|$. This is followed by linearly down projecting the up-projected vectors to $\mathbb{R}^{\mathrm{E}}$ with $\mathrm{E} < \mathrm{D}$ in order to capture the underlying structure in $\mathbb{R}^{\mathrm{D}}$ more compactly. We present both analytical as well as empirical evidence of the fact that the "up/down" projections employed by CRAFTMaps approximate $\mathrm{K}(\mathbf{x}, \mathbf{y})$ better than a direct random polynomial feature map $\mathbf{Z} : \mathbb{R}^d \rightarrow \mathbb{R}^{\mathrm{E}}$.

The additional cost of down projecting from $\mathbb{R}^{\mathrm{D}}$ to $\mathbb{R}^{\mathrm{E}}$ incurred by CRAFTMaps is well-justified by the efficiency gains they offer in terms of training in $\mathbb{R}^{\mathrm{E}}$. To further improve the efficiency of CRAFTMaps, we show how they can be generated using structured random matrices, in particular Hadamard transform, that reduces the cost of multiplying two $n \times n$ matrices from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2 \log(n))$. This gain is exploited for both up as well as down projection steps of CRAFTMaps.

The compactness of CRAFTMaps makes them particularly suitable for using Hessian based methods to learn classifiers in a single pass over the data. Moreover, we show how CRAFTMaps can be used to learn multi-class classifiers in a streaming manner, using the previously proposed framework of error correcting output codes (ECOCs) (Dietterich & Bakiri, 1994), to minimize the least square error between the predicted and the true class labels. This combination of CRAFTMaps and ECOCs is particularly powerful as it can be formalized as a matrix-matrix multiplication, and can therefore maximally exploit the multi-core processing power of modern hardware using BLAS3 (Golub & Van Loan, 2012). Finally, by requiring minimal communication among *mappers*, this framework is well-suited for *map-reduce* based settings.

2. Related Work

Extending the kernel machines framework to large scale learning has been explored in a variety of ways (Bottou et al., 2007) (Sonnenburg et al., 2006) (Joachims, 1999). The most popular of these approaches are decomposition methods for solving Support Vector Machines (Platt, 1999) (Chang & Lin, 2011). While in general extremely useful, these methods do not always scale well to problems with more than a few hundreds of thousand data-points.

To solve this challenge, several schemes have been proposed to explicitly approximate the kernel matrix, including low-rank approximations (Blum, 2006) (Bach & Jordan, 2005), sampling individual entries (Achlioptas et al., 2002), or discarding entire rows (Drineas & Mahoney, 2005). Similarly, fast nearest neighbor look-up methods have been used to approximate multiplication operations with the kernel matrix (Shen et al., 2005). Moreover, concepts from computational geometry have also been explored to obtain efficient approximate solutions for SVM learning (Tsang et al., 2006).

An altogether different approximation approach that has recently gained much interest is to approximate the kernel function directly as opposed to explicitly operating on the kernel matrix. This can be done by embedding the non-linear kernel space into a low dimensional Euclidean space while incurring an arbitrarily small additive distortion in the inner product values (Rahimi & Recht, 2007). By relying only on the embedded space dimensionality, this approach presents a potential solution to the aforementioned *curse of support*, and is similar in spirit to previous efforts to avoid the *curse of dimensionality* in nearest neighbor problems (Indyk & Motwani, 1998).

While the work done in (Rahimi & Recht, 2007) focuses on translation invariant kernels, there have been several subsequent approaches proposed to approximate other kernels as well, some of which include group invariant (Li et al., 2010), intersection (Maji & Berg, 2009), and RBF kernels (Le et al., 2013). There has also been an interest in approximating polynomial kernels using random feature maps (Kar & Karnick, 2012) and random tensor products (Pham & Pagh, 2013). Our work builds on these approaches and provides a more compact representation of accurately approximating polynomial kernels.

3. Compact Random Feature Maps

We begin by demonstrating the rank deficiency of the previous polynomial kernel approximations (Kar & Karnick, 2012) (Pham & Pagh, 2013), followed by a detailed presentation of the CRAFTMaps framework.

3.1. Preliminaries

Following (Kar & Karnick, 2012), consider a positive definite kernel $\mathrm{K} : (\mathbf{x}, \mathbf{y}) \mapsto f(\langle \mathbf{x}, \mathbf{y} \rangle)$, where f admits a Maclaurin expansion, *i.e.*, $f(x) = \sum_{n=0}^{\infty} a_n x^n$, where $a_n = f^{(n)}(0)/n!$. An example of such a kernel is the polynomial kernel $\mathrm{K}(\mathbf{x}, \mathbf{y}) = (\langle \mathbf{x}, \mathbf{y} \rangle + q)^r$, with $q \in \mathbb{R}^+$ and $r \in \mathbb{N}_0$. By defining estimators for each individual term of the kernel expansion, one can approximate the exact kernel dot-products. To this end, let $\mathbf{w} \in \{-1, 1\}^d$ be a Rademacher vector, *i.e.*, each of its components are

Algorithm 1 – RANDOM FEATURE MAPS (RFM)

```
Input: Kernel parameters $q$ and $r$, output dimensionality D, sampling parameter $p > 1$
Output: Random feature map $\mathbf{Z} : \mathbb{R}^d \rightarrow \mathbb{R}^{\mathrm{D}}$ such that $\langle \mathbf{Z}(\mathbf{x}), \mathbf{Z}(\mathbf{y}) \rangle \approx \mathrm{K}(\mathbf{x}, \mathbf{y})$
1: Find $f(x) = \sum_{n=0}^{\infty} a_n x^n$, where $a_n = \frac{f^{(n)}(0)}{n!}$
2: for each $i$ = 1 to D do
3:     Choose variable N using $\Pr[\mathrm{N} = n] = \frac{1}{p^{n+1}}$
4:     Choose $\mathbf{w}_j \in \{-1, 1\}^d$ using N fair coin tosses
5:     Define $Z_i : \mathbf{x} \mapsto \sqrt{a_{\mathrm{N}} p^{\mathrm{N}+1}} \prod_{j=1}^{\mathrm{N}} \mathbf{w}_j^{\mathrm{T}} \mathbf{x}$
6: Construct $\mathbf{Z} : \mathbf{x} \mapsto \frac{1}{\sqrt{\mathrm{D}}}(Z_1, \cdots, Z_{\mathrm{D}})$
```

Algorithm 2 – CRAFTMAPS USING RFM

```
Input: Kernel parameters $q$ and $r$, up and down projection dimensionalities D and E such that E < D, sampling parameter $p > 1$
Output: CRAFTMap $\mathbf{G} : \mathbb{R}^d \rightarrow \mathbb{R}^{\mathrm{E}}$, such that $\langle \mathbf{G}(\mathbf{x}), \mathbf{G}(\mathbf{y}) \rangle \approx \mathrm{K}(\mathbf{x}, \mathbf{y})$
1: Up Project: Using Algorithm 1, construct random feature map $\mathbf{Z} : \mathbb{R}^d \rightarrow \mathbb{R}^{\mathrm{D}}$, such that $\langle \mathbf{Z}(\mathbf{x}), \mathbf{Z}(\mathbf{y}) \rangle \approx \mathrm{K}(\mathbf{x}, \mathbf{y})$
2: Down Project: Using Johnson-Lindenstrauss random projection, linearly down-project $\mathbf{Z}$ to construct $\mathbf{G} : \mathbb{R}^{\mathrm{D}} \rightarrow \mathbb{R}^{\mathrm{E}}$ such that $\langle \mathbf{G}(\mathbf{Z}(\mathbf{x})), \mathbf{G}(\mathbf{Z}(\mathbf{y})) \rangle \approx \langle \mathbf{Z}(\mathbf{x}), \mathbf{Z}(\mathbf{y}) \rangle$.
```

chosen independently using a fair coin toss from the set $\{-1, 1\}$. It can be shown that for $\Pr[\mathrm{N} = n] = 1/(p^{n+1})$ for some constant $p > 1$, and $\mathbf{w}_1, \cdots, \mathbf{w}_{\mathrm{N}}$ as N independent Rademacher vectors, the feature map $Z_i : \mathbb{R}^d \rightarrow \mathbb{R}$, $Z_i : \mathbf{x} \mapsto \sqrt{a_{\mathrm{N}} p^{\mathrm{N}+1}} \prod_{j=1}^{\mathrm{N}} \mathbf{w}_j^{\mathrm{T}} \mathbf{x}$ gives an unbiased estimate of the polynomial kernel. Generating D such feature maps independently and concatenating them together constructs a multi-dimensional feature map $\mathbf{Z} : \mathbb{R}^d \rightarrow \mathbb{R}^{\mathrm{D}}, \mathbf{Z} : \mathbf{x} \mapsto 1/\sqrt{\mathrm{D}}(Z_1(\mathbf{x}), \cdots, Z_{\mathrm{D}}(\mathbf{x}))$, such that $\mathbb{E}\left(\langle \mathbf{Z}(\mathbf{x}), \mathbf{Z}(\mathbf{y}) \rangle\right) = \mathrm{K}(\mathbf{x}, \mathbf{y})$. The procedure for generating random feature maps for polynomial kernels is listed in Algorithm 1 and illustrated in Figure 1.

3.2. Limitations of Random Feature Maps

The benefit of random feature maps to approximate the underlying eigen structure of the exact kernel space can come at the cost of their rank deficiency. Consider *e.g.* Figure 2(a) where the black graph shows the log-scree plot of the exact 7th order polynomial kernel ($q = 1$) obtained using 1000 randomly selected set of points from MNIST data. The red graph shows the log-scree plot for the random feature map (Kar & Karnick, 2012) in a 2^{12} dimensional space. Note that the red plot is substantially lower than the black one for majority of the spectrum range. This rank deficiency is also true for the space generated by random tensor products (Pham & Pagh, 2013) whose log-scree plot is shown in green in Figure 2(a).

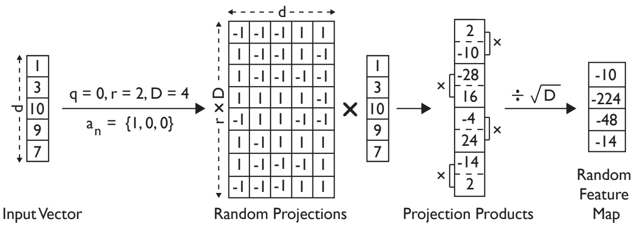


Figure 1: Projection of a 5 dimensional vector to a random feature map for a 2nd order homogenous polynomial kernel in 4 dimensions. As $a_n = \{1, 0, 0\}$ here, we need $r \times \mathrm{D} = 2 \times 4 = 8$ Rademacher vectors such that we could multiply each $r = 2$ projections to construct $\mathbf{Z}$ in $\mathrm{D} = 4$ dimensions.

This rank deficiency can result in the under-utilization of the projected feature space. Figure 2(b) shows the histogram of the linear weight vector learned in a 2^{12} dimensional random feature map (Kar & Karnick, 2012) for a 7th order polynomial kernel ($q = 1$). The plot was obtained for 1000 randomly selected points from MNIST data for two class-sets. The spike at zero shows that a majority of the learned weight components do not play any role in classification.

3.3. CRAFTMaps using Up/Down Projections

To address the limitations of random feature maps, we propose CRAFTMaps as a more accurate approximation of polynomial kernels. The intuition behind CRAFTMaps is to first capture the eigen structure of the exact kernel space comprehensively, followed by representing it in a more concise form. CRAFTMaps are therefore generated in the following two steps:

Up Projection: Since the difference between $\langle \mathbf{Z}(\mathbf{x}), \mathbf{Z}(\mathbf{y}) \rangle$ and $\mathrm{K}(\mathbf{x}, \mathbf{y})$ reduces exponentially as a function of the dimensionality of $\mathbf{Z}$ (Kar & Karnick, 2012) (Pham & Pagh, 2013), we first up project the original data non-linearly from $\mathbb{R}^d$ to a substantially higher dimensional space $\mathbb{R}^{\mathrm{D}}$ to maximally capture the underlying eigen structure of the exact kernel space.

Down Projection: Since the randomized feature map $\mathbf{Z} : \mathbb{R}^d \rightarrow \mathbb{R}^{\mathrm{D}}$ generated as a result of the up-projection step is fundamentally rank-deficient (as shown previously in § 3.2), we linearly down project $\mathbf{Z}$ to a lower-dimensional map $\mathbf{G} : \mathbb{R}^{\mathrm{D}} \rightarrow \mathbb{R}^{\mathrm{E}}$, such that E < D, and $\langle \mathbf{G}(\mathbf{Z}(\mathbf{x})), \mathbf{G}(\mathbf{Z}(\mathbf{y})) \rangle \approx \langle \mathbf{Z}(\mathbf{x}), \mathbf{Z}(\mathbf{y}) \rangle$. The procedure to generate CRAFTMaps is listed in Algorithm 2. Note that while Algorithm 2 uses random feature maps (Kar & Karnick, 2012) for up-projection, one could also use other feature maps

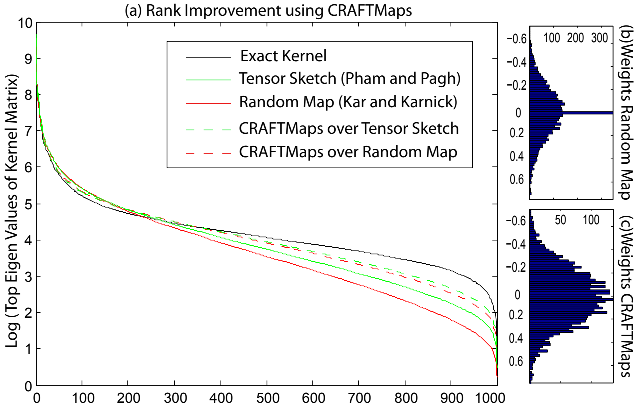


Figure 2: **(a)** Rank deficiency of tensor sketch (Pham & Pagh, 2013) and random feature maps (Kar & Karnick, 2012), along with rank improvements due to CRAFTMaps. **(b-c)** Histograms of weight vectors learned in a 2^{12} dimensional random feature map (Kar & Karnick, 2012) and CRAFTMaps (here D was set equal to 2^{14}).

e.g. tensor products (Pham & Pagh, 2013) to generate **Z**.

The rank improvement brought about by using CRAFTMaps for random feature maps and tensor sketch is shown in Figure 2-a by the dotted red and green plots respectively. The improved utilization of the projected space of random feature maps due to CRAFTMaps is demonstrated in Figure 2(c).

3.4. Error Bounds for CRAFTMaps

Recall that the following result obtained using an application of the Hoeffding inequality (Hoeffding, 1963) is central to the analysis of (Kar & Karnick, 2012):

$$\Pr\left(|\langle \mathbf{Z}(\mathbf{x}), \mathbf{Z}(\mathbf{y})\rangle - \mathrm{K}(\mathbf{x},\mathbf{y})| > \varepsilon\right) \leq 2\exp\left(-\frac{\mathrm{D}\varepsilon^2}{8\mathrm{C}_\Omega^2}\right) \quad (1)$$

where D is the dimensionality of **Z**, and C_Ω^2 is a constant (defined below). We first examine this inequality more closely for homogenous polynomial kernels $\mathrm{K}(\mathbf{x},\mathbf{y}) = \langle \mathbf{x}, \mathbf{y}\rangle^r$ for all points on the unit sphere. In that case we have,

$$\mathrm{C}_\Omega^2 = (pf(p\mathrm{R}^2))^2 = \left(\frac{1}{2^{r+1}}\right)^2 d^{2r} \quad (2)$$

where $\mathrm{R} = \max \|\mathbf{x}\|_{\ell_1} = \sqrt{d}$ and a suitable choice for p is $1/2$. We only get a non-trivial bound when $\mathrm{D} \gtrsim \varepsilon^{-2} d^{2r}$. Note however that if we used explicit kernel expansion, we would need substantially fewer features (at most $\binom{d+r-1}{r}$). The same holds for (Pham & Pagh, 2013) since they apply the same Hoeffding inequality, and the analysis produces the same asymptotics.

We therefore first present an improved error analysis of (Kar & Karnick, 2012), focusing on homogeneous polynomial kernels. We then use this analysis to prove error bounds of CRAFTMaps. Note that these bounds are independent of the dimensionality of the input space, which is a significant improvement over both (Kar & Karnick, 2012) and (Pham & Pagh, 2013).

Lemma 3.1. Fix an integer $r \geq 2$, and define $\mathrm{S_D}$ as:

$$\mathrm{S_D} = \sum_{i=1}^{\mathrm{D}} \prod_{j=1}^{r} \langle \mathbf{x}, \omega_{i,j}\rangle \langle \mathbf{x}', \omega_{i,j}\rangle$$

where $\mathbf{x}, \mathbf{x}'$ are vectors of unit Euclidean length, and $\omega_{i,j} \sim \mathcal{N}(0, I_d)$ are independent Gaussian vectors. Then whenever $\mathrm{D} \geq 3 \cdot 4^{r+2}\varepsilon^{-2}$,

$$\Pr\left(\left|\frac{1}{\mathrm{D}}\mathrm{S_D} - \langle \mathbf{x}, \mathbf{x}'\rangle^r\right| \geq \varepsilon\right) \leq c^r \exp\left(-\frac{1}{2}\left(\frac{\mathrm{D}\varepsilon^2}{11}\right)^{\frac{1}{2r+2}}\right)$$

where $0 < c < 0.766$ is a universal constant.

Proof: Let $\mathrm{Y}_i = \prod_{j=1}^{r} \langle \mathbf{x}, \omega_{i,j}\rangle \langle \mathbf{x}', \omega_{i,j}\rangle$, then the deviation of $\mathrm{S_D}$ from its mean is estimated by the rate at which the tails of Y_i decay, which is in turn determined by the rates at which the moments of Y_i grow. We first verify that the expectation of the summands indeed equals $\langle \mathbf{x}, \mathbf{x}'\rangle^r$:

$$\mathbb{E}\left(\mathrm{Y}_i\right) = \prod_{j=1}^{r} \mathbb{E}\left(\mathbf{x}^{\mathrm{T}}\omega_{i,j}\omega_{i,j}^{\mathrm{T}}\mathbf{x}'\right) = \langle \mathbf{x}, \mathbf{x}'\rangle^r$$

Similarly, the k^{th} moment of Y_i can be determined as:

$$\begin{aligned}
\mathbb{E}\left(|\mathrm{Y}_i|^k\right) &= \prod_{j=1}^{r} \mathbb{E}\left(\left|\mathrm{tr}\left(\mathbf{x}^{\mathrm{T}}\omega_{i,j}\omega_{i,j}^{\mathrm{T}}\mathbf{x}'\right)\right|^k\right) \\
&\leq \prod_{j=1}^{r} \left[\left\|\mathbf{x}'\mathbf{x}^{\mathrm{T}}\right\|_2^k \mathbb{E}\left(\mathrm{tr}\left(\mathbf{x}^{\mathrm{T}}\omega_{i,j}\omega_{i,j}^{\mathrm{T}}\mathbf{x}\right)^k\right)\right] \\
&= \prod_{j=1}^{r} \mathbb{E}\left(|\omega_{i,j}^{\mathrm{T}}\mathbf{x}|^{2k}\right) = \prod_{j=1}^{r} \mathbb{E}\left(|\gamma_j|^{2k}\right) \\
&= \left[\left(\frac{1}{2}\right)^k \frac{(2k)!}{k!}\right]^r \leq (\sqrt{2})^r \left(\frac{2k}{e}\right)^{rk} = c^r k^{rk}
\end{aligned}$$

Here $\gamma_j \sim \mathcal{N}(0,1)$, $c = \sqrt{2}(2/e)^k$, and the last three expressions above follow from the formula for the moments of a standard Gaussian random variables (Patel & Read, 1996). We now estimate moments of feature map approximation error.

$$\mathrm{Q} = \frac{1}{\mathrm{D}^k}\mathbb{E}\left(\left|\sum_{i=1}^{\mathrm{D}}(\mathrm{Y}_i - \mathbb{E}(\mathrm{Y}_i))\right|^k\right)$$

Assuming $k \geq 2$, and using Marcinkiewicz–Zygmund inequality (Burkholder, 1988) we have:

$$\mathrm{Q} \leq \left(\frac{k}{\sqrt{\mathrm{D}}}\right)^k \mathbb{E}\left(|\mathrm{Y}_i - \mathbb{E}(\mathrm{Y}_i)|^k\right)$$

A standard estimate of the right-hand quantity using Jenson's inequality allows us to conclude that

$$\mathsf{Q} \le \left(\frac{2k}{\sqrt{\mathrm{D}}}\right)^{k} \mathbb{E}\left(|Y_i|^{k}\right) \le c^{r} \left(\frac{2k}{\sqrt{\mathrm{D}}}\right)^{k} k^{rk}$$

Finally, we apply Markov's inequality to bound the tails of the approximation error:

$$\mathsf{Pr}\left(\left|\frac{1}{\mathrm{D}}\sum_{i=1}^{\mathrm{D}} \mathrm{Y}_i - \langle \mathbf{x},\mathbf{x}'\rangle^{r}\right| \ge \varepsilon\right) \le \frac{\mathrm{Q}}{\varepsilon^{k}} \le c^{r}\left(\frac{2k}{\varepsilon\sqrt{\mathrm{D}}}\right)^{k} k^{rk}$$
$$= c^{r} \exp\left(k\left[\log(2k^{r+1}) - \log(\varepsilon\sqrt{\mathrm{D}})\right]\right)$$

Fixing $\alpha > 0$ and assuming that $\mathrm{D} > e^{2\alpha}4^{r+2}\varepsilon^{-2}$ and $k = \lfloor(\varepsilon^{2}\mathrm{D}e^{-2\alpha}/4)^{1/(2r+2)}\rfloor$ ensures that

$$\log(2k^{r+1}) - \log(\varepsilon\sqrt{\mathrm{D}}) \le -\alpha$$

and $k \ge 2$, so our earlier assumption when applying Marcinkiewicz–Zygmund inequality is valid. Thus

$$\mathsf{Pr}\left(\left|\frac{1}{\mathrm{D}}\sum_{i=1}^{\mathrm{D}} \mathrm{Y}_i - \langle \mathbf{x},\mathbf{x}'\rangle^{r}\right| \ge \varepsilon\right) \le c^{r} \exp\left(-\alpha\left(\frac{\mathrm{D}\varepsilon^{2}}{4e^{2\alpha}}\right)^{\rho}\right)$$

where $\rho = 1/(2r+2)$ and $c \le \sqrt{2}(2/e)^{2} < 0.766$. Take $\alpha = 1/2$ to reach the bound in the theorem. □

Applying Lemma 3.1, the following corollary follows:

Corollary 3.2. Let $\mathrm{X} \subset \mathbb{R}^{d}$ be a set of n unit vectors. Let $\omega_{i,j} \sim N(0, I_d)$ be a set of $r \cdot \mathrm{D}$ independent Gaussian random vectors. If $\mathrm{D} \gtrsim 4^{r+1}\log(n)^{2r+2}\varepsilon^{-2}$ then we have with high probability:

$$\left|\frac{1}{\mathrm{D}}\sum_{i=1}^{\mathrm{D}}\prod_{j=1}^{r}\langle\mathbf{x},\omega_{i,j}\rangle\langle\mathbf{x}',\omega_{i,j}\rangle - \langle\mathbf{x},\mathbf{x}'\rangle^{r}\right| \le \varepsilon$$

which holds simultaneously $\forall\, \mathbf{x},\mathbf{x}' \in \mathrm{X}$.

Proof: We apply the Lemma 3.1 along with the trivial union bound over $\mathcal{O}(n^{2})$ points. Thus, we require $\exp(\log(n^{2}) - (\mathrm{D}\varepsilon^{2})^{1/(2r+2)})$ to be small. In this case, picking $\mathrm{D} \ge \log(n^{2})^{(2r+2)}\varepsilon^{-2}$ suffices. □

An alternate way to view this is to fix D, in which case the final approximation error will be bounded by:

$$\varepsilon \lesssim \log(n^{2})^{r+1}/\sqrt{\mathrm{D}} \qquad (3)$$

We can combine this with a usual Johnson-Lindenstrauss (Johnson & Lindenstrauss, 1984) random projection as follows:

Theorem 3.3. Let $\mathrm{X} \subset \mathbb{R}^{d}$ be a set of n unit vectors. Suppose we map these vectors using a random feature map $\mathbf{Z} : \mathbb{R}^{d} \to \mathbb{R}^{\mathrm{D}}$ composed with a Johnson-Lindenstrauss map $\mathrm{Q} : \mathbb{R}^{\mathrm{D}} \to \mathbb{R}^{\mathrm{E}}$, where $\mathrm{D} \ge \mathrm{E}$, to obtain $\mathbf{Z}'$, then the following holds:

$$\left|\langle\mathbf{x},\mathbf{x}'\rangle^{r} - \langle\mathbf{Z}'(\mathbf{x}),\mathbf{Z}'(\mathbf{y})\rangle\right| \lesssim \frac{2^{r+1}\log(n)^{r+1}}{\mathrm{D}^{1/2}} + \frac{\log(n)^{1/2}}{\mathrm{E}^{1/2}}$$

with high probability $\forall\, \mathbf{x},\mathbf{x}' \in \mathrm{X}$ simultaneously.

Proof: A Johnson-Lindenstrauss projection from $\mathbb{R}^{\mathrm{D}}$ to $\mathbb{R}^{\mathrm{E}}$ preserves with high probability all pairwise inner products of the n points $\{\mathbf{Z}(\mathbf{x}) : \mathbf{x} \in \mathbf{X}\}$ in $\mathbb{R}^{\mathrm{D}}$ to within an additive factor of $\varepsilon' \lesssim \log(n)^{1/2}/\mathrm{E}^{1/2}$. Applying the triangle inequality:

$$|\langle\mathbf{x},\mathbf{y}\rangle^{r} - \langle\mathbf{Z}'(\mathbf{x}),\mathbf{Z}'(\mathbf{y})\rangle| \le |\langle\mathbf{x},\mathbf{y}\rangle^{r} - \langle\mathbf{Z}(\mathbf{x}),\mathbf{Z}(\mathbf{y})\rangle| + |\langle\mathbf{Z}(\mathbf{x}),\mathbf{Z}(\mathbf{y})\rangle - \langle\mathbf{Z}'(\mathbf{x}),\mathbf{Z}'(\mathbf{y})\rangle| := \varepsilon + \varepsilon'$$

Referring to Equation 3 to bound ε, we obtain the final error bound:

$$\varepsilon + \varepsilon' \lesssim \frac{2^{r+1}\log(n)^{r+1}}{\mathrm{D}^{1/2}} + \frac{\log(n)^{1/2}}{\mathrm{E}^{1/2}}$$

□

In particular, the error is lower than random feature maps (Kar & Karnick, 2012) whenever:

$$\frac{2^{r+1}\log(n)^{r+1}}{\mathrm{D}^{1/2}} + \frac{\log(n)^{1/2}}{\mathrm{E}^{1/2}} \lesssim \frac{2^{r+1}\log(n)^{r+1}}{\mathrm{E}^{1/2}}$$

Fixing $\mathrm{D} = g(r)\mathrm{E}$ for some constant $g(r) \ge 1$, CRAFTMaps provide a better error bound when:

$$g(r) \gtrsim \left(\frac{\log(n)^{r+1/2}}{\log(n)^{(r+1/2)} - 2^{-(r+1)}}\right)^{2} \approx 1$$

3.5. Efficient CRAFTMaps Generation

Recall that for Hessian based optimization of linear regression problems, the dominant cost of $\mathcal{O}(n\mathrm{D}^{2})$ is spent calculating the Hessian. By compactly representing random feature maps in $\mathbb{R}^{\mathrm{E}}$ as opposed to $\mathbb{R}^{\mathrm{D}}$ for $\mathrm{E} < \mathrm{D}$, CRAFTMaps provide a factor of $\mathrm{D}^{2}/\mathrm{E}^{2}$ gain in the complexity of Hessian computation. A straightforward version of CRAFTMaps would incur an additional cost of $\mathcal{O}(n\mathrm{DE})$ for the down-projection step. However, since for problems at scale $n >> \mathrm{D}$, the gains CRAFTMaps provide for classifier learning over random feature maps is well worth the relatively small additional cost they incur.

These gains can be further improved by using structured random matrices for the up/down projections of CRAFTMaps. One way to do this is to use the Hadamard matrix as a set of orthonormal bases, as opposed to using a random bases-set sampled from a zero mean distribution. The structured nature of Hadamard matrices enables efficient recursive matrix-matrix multiplication that

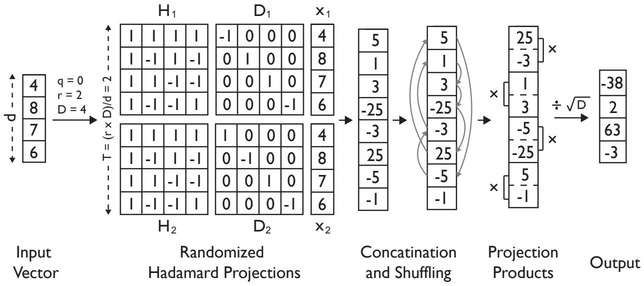


Figure 3: Randomized Hadamard basis to up-project an input vector in 4 dimensional space to a random map for a 2nd order homogenous kernel in a 4 dimensional space.

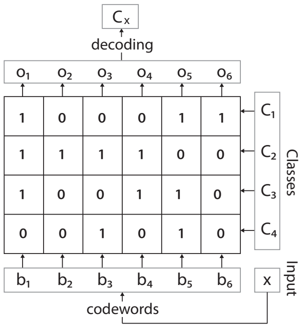
only requires $\mathcal{O}(n^2\log(n))$ operations compared to the $\mathcal{O}(n^3)$ operations needed for the product of two $n \times n$ non-structured matrices. Constructing CRAFTMaps using Hadamard transform can therefore reduce the complexity of up projection from $\mathcal{O}(n\mathrm{D}d)$ to $\mathcal{O}(n\mathrm{D}\log(d))$, and that of down projection from $\mathcal{O}(n\mathrm{D}^2)$ to $\mathcal{O}(n\mathrm{D}\log(\mathrm{D}))$ respectively. To employ Hadmard matrices for efficient CRAFTMaps generation, we use the sub-sampled randomized Hadamard transform (SRHT) (Tropp, 2011).

While SRHT can be used directly for the down-projection step, we need to incorporate a few novel modifications to it before it can be used for up-projection. In particular, given a kernel function $\mathrm{K} : (\mathbf{x}, \mathbf{y}) \mapsto f(\langle \mathbf{x}, \mathbf{y} \rangle)$ and a d dimensional[1] vector $\mathbf{x}$, we first construct $\mathrm{T} = \lceil (\sum_{i=1}^{D} \mathrm{N}_i)/d \rceil$ copies of $\mathbf{x}$, where N is defined in Algorithm 1. Each copy $\mathbf{x}_t$ is multiplied by a diagonal matrix $\mathbf{M}_t$ whose entries are set to $+1$ or -1 with equal probability. Each matrix $\mathbf{M}_t\mathbf{x}_t$ is implicitly multiplied by the $d \times d$ Hadamard matrix $\mathbf{H}$. All rows of $\mathbf{H}\mathbf{M}_t\mathbf{x}_t$ for all $t = \{1, \cdots, \mathrm{T}\}$ are first concatenated, and then randomly permuted, to be finally used according to Algorithm 1 to non-linearly up-project $\mathbf{x}$ from $\mathbb{R}^d$ to $\mathbb{R}^{\mathrm{D}}$ (see Figure 3).

4. Classification Using ECOCs

To solve multi-class classification problems, we use error correcting output codes (ECOCs) (Dietterich & Bakiri, 1994) which employ a unique binary "codeword" of length c for each of the k classes, and learn c binary functions, one for each bit position in the codewords. For training, using an example from class i, the required outputs of the c binary functions are specified by the codeword for class i. Given a test instance $\mathbf{x}$, each of the c binary functions are evaluated to compute a c-bit string s. This string is compared to the k codewords, assigning $\mathbf{x}$ to the class whose codeword is closest to s according to some distance (see Figure 4).

Overall, given d dimensional data from k classes, we use up/down projections to construct its CRAFTMap representation in $\mathbb{R}^{\mathrm{E}}$. We then use ECOCs to learn c binary linear regressors in $\mathbb{R}^{\mathrm{E}}$. To test a point, we up/down project it to $\mathbb{R}^{\mathrm{E}}$ and then pass it through the ECOCs to be classified to one of the k classes.



1	0	0	0	1	1	C₁
1	1	1	1	0	0	C₂
1	0	0	1	1	0	C₃
0	0	1	0	1	0	C₄

Figure 4: To learn bit 3 classifier, all points from C_2 & C_4 are considered positive, and those from C_1 & C_3 negative. If *e.g.* the detected labels for a test instance are 110110, it is assigned to C_3 as it is at minimum distance from C_3 codeword.

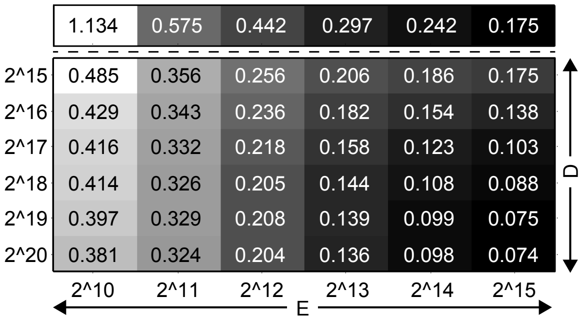
5. Experiments and Results

We now present CRAFTMaps results on multiple data-sets. Unless otherwise mentioned, we use the H-0/1 heuristic of (Kar & Karnick, 2012) for random feature maps (RFM) and CRAFTMaps on RFM.

5.1. Reconstruction Error

Figure 5 shows the normalized root mean square errors (NRMSE) for MNIST data obtained for an $r = 7$ and $q = 1$ kernel using random feature maps (RFM – top row) (Kar & Karnick, 2012) versus CRAFTMaps (bottom table) over RFM. These results were obtained using the same set of 1000 randomly selected data points. As shown, CRAFTMaps provide a significant improvement consistent over a range of D and E. Similar trends can be found in Figure 6 where NRMSE for 6 different data-sets over a range of E are shown.

Figure 7 (a) shows reconstruction results as a function of polynomial degree obtained using 10 sets of 1000 randomly picked points from MNIST data. As shown, CRAFTMaps consistently improve the reconstruction error over a wide range of polynomial degrees.



	2^10	2^11	2^12	2^13	2^14	2^15
	1.134	0.575	0.442	0.297	0.242	0.175
2^15	0.485	0.356	0.256	0.206	0.186	0.175
2^16	0.429	0.343	0.236	0.182	0.154	0.138
2^17	0.416	0.332	0.218	0.158	0.123	0.103
2^18	0.414	0.326	0.205	0.144	0.108	0.088
2^19	0.397	0.329	0.208	0.139	0.099	0.075
2^20	0.381	0.324	0.204	0.136	0.098	0.074

Figure 5: NRMSE for polynomial kernel reconstruction using r=7 and q=1 on MNIST. Top row is for RFM and bottom table for CRAFTMaps on RFM.

[1] As Hadamards exist in powers of 2, usually $\mathbf{x}$ needs to be zero-padded to the closest higher power of 2.

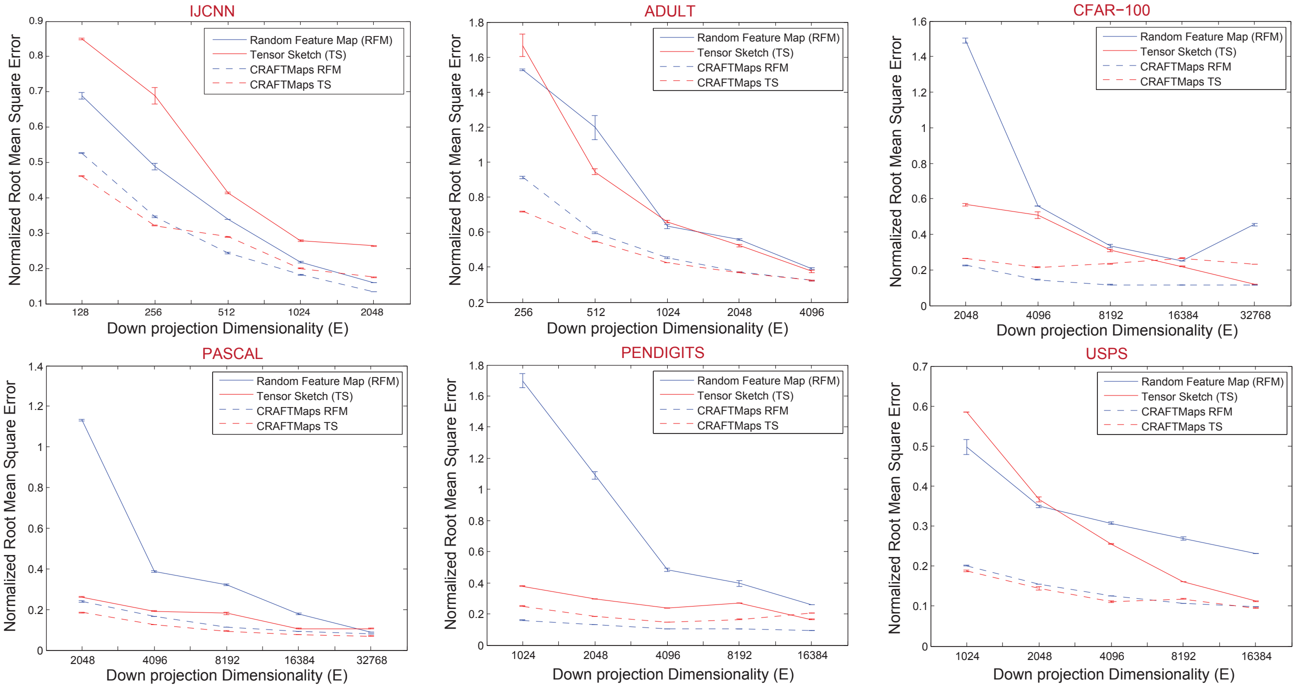


Figure 6: NRMSE obtained while reconstructing the polynomial kernel with $r = 7$ and $q = 1$. Here D = 2 × max(E).

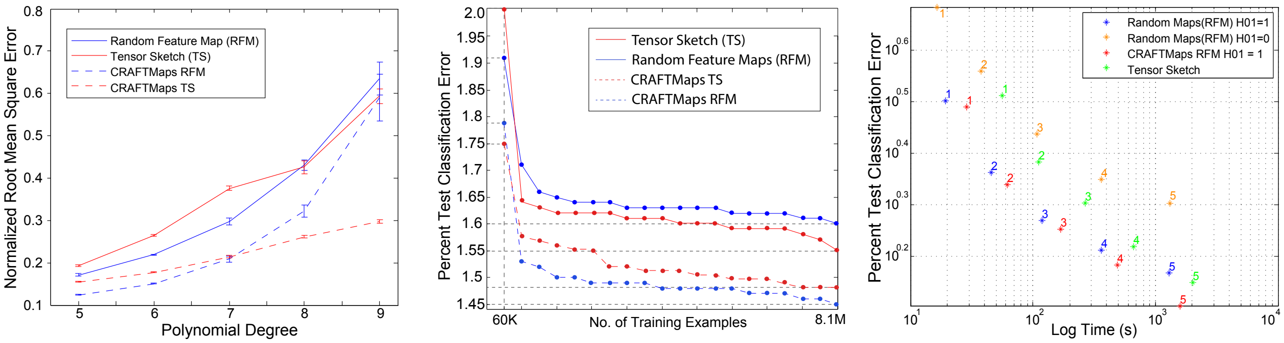


Figure 7: (a) Reconstruction error as a function of polynomial degree, averaged over 10 randomly sampled 1000 points of MNIST data. Here $\mathbb{R}^{\mathrm{D}} = 2^{15}$ while $\mathbb{R}^{\mathrm{E}} = 2^{13}$. (b) Test classification for MNIST8M. Here $\mathrm{D} = 2^{17}$, $\mathrm{E} = 2^{14}$, $q = 1$, $r = 7$ and ECOCs = 200. (c) Log-log scatter plot of compute times (projection+Hessian) for MNIST data. For CRAFTMaps, projection took 18.1s, 36.3s, 75.9s, 186.3s, and 419.2s, while finding Hessian took 9.5s, 26.2s, 81.1s, 334.8s, and 1100.3s respectively. Note that CRAFTMaps show significant per unit-time classification improvements for larger feature spaces.

5.2. Classification Error

Table 1 shows the test classification error using CRAFTMaps on random feature maps (Kar & Karnick, 2012) and tensor sketch (Pham & Pagh, 2013) compared to 5 alternate approaches over 4 different data-sets. As can be observed, CRAFTMaps consistently delivered improved classification performance, mostly outperforming the considered alternatives.

We now explain results for CRAFTMaps on MNIST data for small and substantially large projected feature spaces. We also explain CRAFTMaps results on very large amounts of training data using MNIST8M.

Small Feature Spaces: Table 1-**a** shows MNIST results on feature space sizes 300 to 700 dimensions. Note that for E $< d$ (which for MNIST is 784), the random feature maps cannot use the **H**-$0/1$ heuristic of (Kar & Karnick, 2012). CRAFTMaps however do not have this limitation as even for E $< d$, D can still be $>> d$. This allows CRAFTMaps to use the **H**-$0/1$ heuristic in $\mathbb{R}^{\mathrm{D}}$, which in turn reflects in $\mathbb{R}^{\mathrm{E}}$. This results in substantial classification gains achieved by CRAFTMaps for small-sized feature spaces, and highlights their usefulness in applications with low memory footprint such as mobile phone apps.

Large Feature Spaces: Table 1-**b** shows the MNIST results on feature sizes 2^{12} to 2^{16}, where CRAFTMaps managed to achieve **1.12%** test error using the original 60K training data (unit-length normalized, non-jittered and non-deskewed). While for small-sized feature spaces the exact kernel on MNIST data perform quite well, CRAFTMaps outperform all alternatives as the size of the feature space grows to larger values.

Results on MNIST8M Data Figure 7 (b) shows the comparative performance of CRAFTMaps for a given sized E (2^{14}) as training size varies from 60K to 8.1M. This experiment uses the same set of 10 thousand test points as used for the experiments with MNIST data. It can be seen that CRAFTMaps on random feature maps converge the fastest, and consistently gives better classification performance compared to the other representations. These results were obtained using a polynomial kernel with $r = 7$, $q = 1$, D $= 2^{17}$, E $= 2^{14}$, and ECOCs equal to 200. As we increase E to 2^{16} and D to 2^{19} using CRAFTMaps on RFM for 7th order polynomial kernel ($q = 1$), we achieved test classification error of **0.91%** on MNIST8M data-set.

5.3. Run-Time Analysis

Figure 7(c) shows the log-log scatter plot of the compute times (projection + Hessian) for random feature maps (Kar & Karnick, 2012), tensor sketching (Pham & Pagh, 2013), and CRAFTMaps using random feature maps (with H-01 heuristic). These times were recorded for MNIST data using a 40-core machine. Notice that CRAFTMaps show significant per unit-time classification improvements towards the right end of the x-axis. This is because as the size of the projected space increases, the Hessian computation cost becomes dominant. This naturally gives CRAFTMaps an edge given their ability to encode information more compactly. The gains of CRAFTMaps are expected to grow even more as the training size further increases.

6. Conclusions and Future Work

In this work, we proposed CRAFTMaps to approximate polynomial kernels more concisely and accurately compared to previous approaches. An important context where CRAFTMaps are particularly useful is the map-reduce setting. By computing a single Hessian matrix (with different gradients for each ECOC) in a concise feature space, CRAFTMaps provide an effective way to learn multi-class classifiers in a single-pass over large amounts of data. Moreover, their ability to compactly capture the eigen structure of the kernel space makes CRAFTMaps suitable for smaller scale applications such as mobile phone apps.

Going forward, we would like to further explore how to better allocate the set of random bases available to us to approximate the different Maclaurin coefficients of a kernel more accurately. Furthermore, we currently commit to a particular kernel function at the start of the training process. However that kernel may not be optimal for the specific learning problem at hand. We are interested in exploring if one can simultaneously solve what is know as the "kernel alignment" problem (Cristianini et al., 2001), as well as learning a low-dimensional kernel embedding using random projections.

a-MNIST 1	300	400	500	600	700
FastFood	32.8	26.6	15.3	11.5	8.4
RKS	8.9	6.7	5.9	5.3	5.0
RFM	14.0	12.3	11.4	10.3	9.5
TS	13.1	11.2	10.0	8.6	8.0
CM RFM	9.5	7.7	7.2	6.6	5.9
CM TS	12.6	10.8	8.9	7.9	7.3
Exact	**6.0**	**5.4**	**5.0**	**4.5**	**4.1**

b-MNIST 2	2^{12}	2^{13}	2^{14}	2^{15}	2^{16}
FastFood	2.78	2.20	2.02	1.87	1.50
RKS	2.94	2.51	2.13	1.91	1.52
RFM	3.17	2.30	1.91	1.62	1.49
TS	3.25	2.41	2.01	1.65	1.41
CM RFM	3.09	**2.18**	1.79	1.52	1.27
CM TS	2.90	2.20	**1.75**	**1.44**	**1.12**
Exact	**2.49**	2.21	1.80	1.49	1.20

c-USPS	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}
FastFood	5.87	5.18	4.83	4.78	4.65
RKS	5.89	5.78	5.53	4.98	4.78
RFM	5.97	5.33	4.68	4.48	4.13
TS	5.92	**5.03**	4.63	4.48	4.33
CM RFM	**5.68**	**5.03**	4.48	4.28	4.03
CM TS	5.77	**5.03**	**4.28**	**4.23**	**3.93**
Exact	5.73	5.08	4.83	−−	−−

d-COIL100	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}
FastFood	8.25	7.83	6.80	6.32	5.21
RKS	8.14	7.36	6.50	5.97	4.81
RFM	11.11	7.55	6.33	5.05	4.83
TS	10.08	7.19	5.69	4.75	4.27
CM RFM	8.94	6.86	5.47	4.52	4.08
CM TS	**8.16**	**5.97**	**4.75**	**4.02**	**3.96**
Exact	9.55	−−	−−	−−	−−

e-PENDIGITS	2^{6}	2^{7}	2^{8}	2^{9}	2^{10}
FastFood	8.74	5.03	3.08	2.85	2.71
RKS	5.83	4.25	2.63	2.17	2.08
RFM	7.94	3.94	2.85	2.28	1.91
TS	11.20	4.57	2.37	1.80	1.77
CM RFM	**7.43**	**3.57**	**2.28**	**1.97**	**1.57**
CM TS	8.03	3.80	2.37	2.05	1.74
Exact	9.29	3.74	2.74	2.31	2.87

Table 1: Test classification errors for multiple data-sets is shown. Here r = 7, 7, 5, 5 and 9 respectively while q = 1. RFM, TS, RKS, Fastfood, and CM stand for (Kar & Karnick, 2012), (Pham & Pagh, 2013), (Rahimi & Recht, 2007), (Le et al., 2013), and CraftMaps respectively. First row of each table shows E, while D $= 8\times$E. Some entries for exact kernel are left empty as training examples for these cases were less than projection dimensionality.

References

Achlioptas, D., McSherry, F., and Schlkopf, B. Sampling Techniques for Kernel Methods. In *Advances in Neural Information Processing Systems 14*, 2002.

Aizerman, A., Braverman, E. M., and Rozoner, L. I. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837, 1964.

Bach, F. R. and Jordan, M. I. Predictive low-rank decomposition for kernel methods. In *Proceedings of the 22nd International Conference on Machine Learning*, 2005.

Bengio, Y., Delalleau, O., and Le Roux, N. The Curse of Highly Variable Functions for Local Kernel Machines. In *Advances in Neural Information Processing Systems 18*, 2006.

Blum, A. Random Projection, Margins, Kernels, and Feature-Selection. In *Subspace, Latent Structure and Feature Selection*. Springer, 2006.

Bottou, L., Chapelle, O., DeCoste, D., and Weston, J. (eds.). *Large-Scale Kernel Machines*. MIT Press, 2007.

Burkholder, D. L. Sharp inequalities for martingales and stochastic integrals. *Asterisque*, 157-158:75–94, 1988.

Chang, C. and Lin, C. LIBSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27, 2011.

Cristianini, Nello, Shawe-Taylor, John, Elisseeff, Andre, and Kandola, Jaz. On kernel target alignment. In *NIPS*, volume 2, pp. 4, 2001.

Dietterich, T. G. and Bakiri, G. Solving Multiclass Learning Problems via Error-Correcting Output Codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1994.

Drineas, P. and Mahoney, M. W. On the Nyström method for approximating a Gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6:2153–2175, 2005.

Golub, G. H. and Van Loan, C. F. *Matrix Computations*. Johns Hopkins University Press, 2012.

Hoeffding, W. Probability Inequalities for Sums of Bounded Random Variables. *Journal of the American Satistical Association*, 58:13–30, 1963.

Indyk, P. and Motwani, R. Approximate Nearest Neighbor: Towards Removing the Curse of Dimensionality. In *Proceedings of the 30th annual ACM Symposium on the Theory of Computing*, 1998.

Joachims, Thorsten. Making large scale svm learning practical. 1999.

Johnson, W. B. and Lindenstrauss, J. Extensions of Lipschitz mappings into a Hilbert space. In *Conference on Modern Analysis and Probability*, 1984.

Kar, P. and Karnick, H. Random Feature Maps for Dot Product Kernels. *Journal of Machine Learning Research*, 22:583–591, 2012.

Le, Quoc, Sarlos, Tamas, and Smola, Alexander. Fastfood-computing hilbert space expansions in loglinear time. In *ICML*, pp. 244–252, 2013.

Li, F., Ionescu, C., and Sminchisescu, C. Random Fourier Approximations for Skewed Multiplicative Histogram Kernels. In *Pattern Recognition*. Springer, 2010.

Maji, S. and Berg, A. C. Max-Margin Additive Classiers for Detection. In *IEEE 12th International Conference on Computer Vision and Pattern Recognition*, 2009.

Patel, J. K. and Read, C. B. *Handbook of the Normal Distribution*. CRC Press, 1996.

Pham, N. and Pagh, R. Fast and Scalable Polynomial Kernels via Explicit Feature Maps. In *Proceedings of the 19th International Conference on Knowledge Discovery and Data Mining*, 2013.

Platt, J. C. Using Analytic QP and Sparseness to Speed Training of Support Vector Machines. In *Advances in Neural Information Processing Systems 11*, 1999.

Rahimi, A. and Recht, B. Random Features for Large-Scale Kernel Machines. In *Advances in Neural Information Processing Systems 20*, 2007.

Schlkopf, B., Burges, C. J. C., and Smola, A. J. (eds.). *Advances in Kernel Methods: Support Vector Learning*. MIT Press, 1999.

Shen, Y., Ng, A., and Seeger, M. Fast Gaussian Process Regression using KD-Trees. In *Advances in Neural Information Processing Systems 18*, pp. 1225–1232. Citeseer, 2005.

Sonnenburg, Sören, Rätsch, Gunnar, Schäfer, Christin, and Schölkopf, Bernhard. Large scale multiple kernel learning. *The Journal of Machine Learning Research*, 7: 1531–1565, 2006.

Steinwart, I. Sparseness of Support Vector Machines. *Journal of Machine Learning Research*, 4:1071–1105, 2003.

Tropp, J. A. Improved analysis of the subsampled randomized Hadamard transform. *Advances in Adaptive Data Analysis*, 3:115–126, 2011.

Tsang, I. W., Kwok, J. T., and Cheung, P. Core vector machines: Fast SVM training on very large data sets. *Journal of Machine Learning Research*, 6(1):363, 2006.