Finding Small Sets of Random Fourier Features for Shift-Invariant Kernel Approximation

Frank-M. Schleif[1,2,3(✉)], Ata Kaban[3], and Peter Tino[3]

[1] School of Computer Science, University of Applied Sciences Würzburg-Schweinfurt, 97074 Würzburg, Germany
schleify@cs.bham.ac.uk

[2] Computational Intelligence Group, University of Applied Sciences Mittweida, Technikumplatz 17, 09648 Mittweida, Germany

[3] School of Computer Science, University of Birmingham, Edgbaston B15 2TT, UK

Abstract. Kernel based learning is very popular in machine learning, but many classical methods have at least quadratic runtime complexity. Random fourier features are very effective to approximate *shift-invariant* kernels by an explicit kernel expansion. This permits to use efficient linear models with much lower runtime complexity. As one key approach to kernelize algorithms with linear models they are successfully used in different methods. However, the number of features needed to approximate the kernel is in general still quite large with substantial memory and runtime costs. Here, we propose a simple test to identify a small set of random fourier features with linear costs, substantially reducing the number of generated features for low rank kernel matrices, while widely keeping the same representation accuracy. We also provide generalization bounds for the proposed approach.

1 Introduction

Kernel based learning methods are very popular in various machine learning tasks like regression, classification or clustering [1–5]. The operations, used to calculate the respective models, typically evaluate the full kernel matrix, leading to quadratic or even cubic complexity. As a consequence, the approximation of positive semi-definite (psd) kernels has raised wide interest [6,7]. Most approaches focus on approximating the kernel by the (clustered) Nyström approximation or specific variations of the singular value decompositions [8,9]. A recent approach effectively combining multiple strategies was presented in [6]. Random fourier features (RFF) have been introduced in [10] to the field of kernel based learning. The aim is to approximate shift invariant kernels by mapping the input data into a randomized feature space and then apply existing fast *linear* methods [11–13]. This is of special interest if the number of samples N is very large and the obtained kernel matrix $K \in \mathbb{R}^{N \times N}$ leads to high storage and calculation costs. The features are constructed so that the inner products of the transformed data are approximately equal to those in the feature space of a user specified shift-invariant kernel:

© Springer International Publishing AG 2016
F. Schwenker et al. (Eds.): ANNPR 2016, LNAI 9896, pp. 42–54, 2016.
DOI: 10.1007/978-3-319-46182-3_4

$$k(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle \approx z(\mathbf{x})' z(\mathbf{y}) \tag{1}$$

With $\phi : \mathcal{X} \mapsto \mathcal{H}$ being a non-linear mapping of patterns from the original input space $\mathcal{X}$ to a high-dimensional space $\mathcal{H}$. The mapping function is in general not given in an explicit form. Unlike the kernel lifting by using $\phi(\cdot)$, z is a (comparable) low dimensional feature vector. In [14] it was empirically shown that for data with a large eigenvalue gap random fourier features are less efficient than a standard Nyström approximation. However, the authors used only a rather small data independent set of fourier features. Here, we propose a selection strategy which not only reduces the number of necessary random fourier features but also helps to select a reasonable set of features, which provides a good approximation of the original kernel function. In this line our focus is less on best possible approximation accuracy but rather on saving memory and obtaining compact representations to address the usage of random fourier features in low resource environments. In [10] it is shown how random fourier feature vectors z can be constructed for various shift invariant kernels $k(\mathbf{x}-\mathbf{y})$, e.g. the RBF kernel upto an error using only $D = \mathcal{O}(d\epsilon^{-2} \log \frac{1}{\epsilon^2})$ dimensions, where d is the input dimension of the original input data. Assuming that $d = 10$ and $\epsilon = 0.01$ one gets $D \approx 100.000$.

However, in [10] it is empirically shown that the approximation is already reasonable enough for smaller $D \approx 500$–5000. While very efficient in general there is not yet a reasonable strategy how to choose an appropriate number D nor which features have to be generated in a more systematic way. If we assume that the images of training inputs in the feature space given (implicitly) by the kernel lie in an intrinsically low dimensional space one can expect that a much smaller number of features should be sufficient to describe the data. A reasonable strategy to test for a reliable number D of random fourier features is to compare the approximated kernel using Eq. (1) with the true kernel K based on the original data using some appropriate measure. This however is in general not possible or very costly for larger N because one would need to generate two $N \times N$ kernel matrices. We suggest to use a constructive approach, generating as many features as necessary to obtain a low reconstruction error between the two kernel matrices. Our approach is very generic as we do not focus on dedicated cost functions used in (semi-)supervised classification nor clustering or embedding measures, such that the constructed feature set provides a reasonable approximation of the shift-invariant kernel matrix. Standard feature reduction techniques for high dimensional data sets like random projection [15,16], unsupervised feature selection techniques based on statistical measures [17] or supervised approaches [18,19] are not suitable because they start from a high-dimensional feature space or are specific to the underlying cost-function. In [1] random fourier features were used in combination with a singular value decomposition to reduce the number of features after generating the random fourier features, again with in general rather large initial D. To avoid high costs in the construction procedure we employ the Nyström approximation at different points to evaluate the accuracy of the constructed random fourier feature set using the Frobenius norm. We assume that the considered kernel is in fact

intrinsically low dimensional. The paper is organized as follows, first we review the main theory for random Fourier features, subsequently we detail our approach of finding small sets of random fourier features, still sufficiently accurate to approximate the kernel matrix using the Frobenius norm. In a subsection we review the Nyström approximation and derive a linear time calculation of the Frobenius norm for the difference of two Nyström approximated matrices. Later we derive error bounds for the presented approach and show the efficiency of our method on various standard datasets employing two state of the art linear time classifiers for vectorial input data.

2 Random Fourier Features

Random fourier features as introduced in [10], project the vectorial data points onto a randomly chosen line, and then pass the resulting scalar through a sinusoid. The random lines are drawn from a distribution so as to guarantee that the inner product of two transformed points approximates the desired shift-invariant kernel. The motivation for this approach is given by Bochners theorem:

Definition 1. *A continuous kernel* $k(\mathbf{x}, \mathbf{y}) = k(\mathbf{x} - \mathbf{y})$ *on* $\mathbb{R}^d$ *is positive definite if and only if* $k(\mathbf{x} - \mathbf{y})$ *is the Fourier transform of a non-negative measure.*

Algorithm 1. RFF selection by optimizing an approximated Frobenius norm

Input: $X, NyK, n, D_{max}, iter, \epsilon$
Output: H_{final}
function RFF SELECTION($(X, NyK, n, D_{max}, iter, \epsilon)$)
    try=0,#F = 0, E_{old} = realmax
    while $\#F < D_{max}$ **do**
        get n random fourier features H
        construct $N\hat{y}K$ based on H
        calculate $E_{\text{new}} = \|N\hat{y}K - NyK\|_F$ Eq. (7)
        if $E_{\text{old}} > E_{\text{new}}$ & $E_{\text{old}} - E_{\text{new}} \geq \epsilon$ **then**
            add H to H_{final}
            #F := #F + n
            update E_{old}
            try := 0
        else
            if try > iterMax **then**
                break
            else
                try := try +1
            end if
        end if
    end while
return H_{final}
end function

If the kernel $k(\mathbf{x} - \mathbf{y})$ is properly scaled, Bochners theorem guarantees that its Fourier transform $p(\omega)$ is a proper probability distribution. The idea in [10] is to approximate the kernel as

$$k(\mathbf{x} - \mathbf{y}) = \int_{\mathbb{R}^d} p(\omega) e^{j\omega(\mathbf{x}-\mathbf{y})} d\omega$$

with some extra normalizations and simplifications one can sample the features for k using the mapping $z_\omega(\mathbf{x}) = [\cos(\mathbf{x})\sin(\mathbf{x})]$. In [10] the authors also give a proof for the uniform convergence of Fourier features to the kernel $k(\mathbf{x}-\mathbf{y})$. A detailed derivation can be found in [10].

To generate the random fourier features one eventually needs a psd kernel matrix $k(\mathbf{x},\mathbf{y}) = k(\mathbf{x}-\mathbf{y})$ and a random feature map $z(\mathbf{x}) : \mathbb{R}^d \rightarrow \mathbb{R}^{2D}$ s.t. $z(\mathbf{x})^\top z(\mathbf{y}) \approx k(\mathbf{x}-\mathbf{y})$. One draws D i.i.d. samples $\{\omega_1, \ldots, \omega_D\} \in \mathbb{R}^d$ from $p(\omega)$ and generates $z(\mathbf{x}) = \sqrt{1/D}[\cos(\omega_1^\top \mathbf{x}) \ldots \cos(\omega_D^\top \mathbf{x})\ \sin(\omega_1^\top \mathbf{x}) \ldots \sin(\omega_D^\top \mathbf{x})]^\top$.

3 Finding Small Sets of Random Fourier Features

To incrementally add fourier features to the approximation of kernel k we use the Frobenius norm to calculate the difference between the two kernels. For real valued data the Frobenius norm of two squared matrices is simply the sum of the squared difference between the individual kernel entries:

$$\left\|\hat{k} - k\right\|_F = \sqrt{\sum_i^N \sum_j^N (\hat{k}(\mathbf{x}_i - \mathbf{y}_j) - k(\mathbf{x}_i - \mathbf{y}_j))^2} \tag{2}$$

This has $\mathcal{O}(N^2)$ costs in memory and runtime and we would need to generate the full kernel $\hat{k}$ and k. To avoid these costs we use the Nyström approximation for kernel matrices [20] to approximate both kernels by using only $\mathcal{O}(N)$ coefficients and provide a formulation for calculating the Frobenius norm of the difference of two Nyström approximated matrices. The Nyström approximation of the original kernel matrix NyK (detailed in the next sub-section) can be done once prior to calculations of the random fourier features. Subsequently the approximated kernel is constructed by iteratively adding those n random fourier features which significantly improve the Frobenius error in Eq. (2) with $\epsilon = 1e^{-3}$. This iterative procedure is continued until either no further significant improvement was found for a number $iterMax = 5$ of random selections or an upper limit of features D_{max} is obtained. The detailed procedure is given in Algorithm 1.

4 Nyström Approximated Matrix Processing

The Nyström approximation technique has been proposed in the context of kernel methods in [20]. One well known way to approximate a $N \times N$ Gram matrix, is to use a low-rank approximation. This can be done by computing the eigendecomposition of the kernel matrix $K = U\Lambda U^T$, where U is a matrix, whose columns are orthonormal eigenvectors, and Λ is a diagonal matrix consisting of eigenvalues $\Lambda_{11} \geq \Lambda_{22} \geq ... \geq 0$, and keeping only the m eigenspaces which correspond to the m largest eigenvalues of the matrix. The approximation is $\tilde{K} \approx U_{(N,m)}\Lambda_{(m,m)}U_{(m,N)}$, where the indices refer to the size of the corresponding submatrix restricted to the larges m eigenvalues. The Nyström

method approximates a kernel in a similar way, without computing the eigendecomposition of the whole matrix, which is an $O(N^3)$ operation.

By the Mercer theorem, kernels $k(\mathbf{x}, \mathbf{x}')$ can be expanded by orthonormal eigenfunctions φ_i and non negative eigenvalues λ_i in the form

$$k(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{\infty} \lambda_i \varphi_i(\mathbf{x}) \varphi_i(\mathbf{x}').$$

The eigenfunctions and eigenvalues of a kernel are defined as solutions of the integral equation

$$\int k(\mathbf{x}', \mathbf{x}) \varphi_i(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} = \lambda_i \varphi_i(\mathbf{x}'),$$

where $p(\mathbf{x})$ is a probability density over the input space. This integral can be approximated based on the Nyström technique by an i.i.d. sample $\{\mathbf{x}_k\}_{k=1}^m$ from $p(\mathbf{x})$:

$$\frac{1}{m} \sum_{k=1}^{m} k(\mathbf{x}', \mathbf{x}_k) \varphi_i(\mathbf{x}_k) \approx \lambda_i \varphi_i(\mathbf{x}'). \tag{3}$$

Using this approximation we denote with $K^{(m)}$ the corresponding $m \times m$ Gram sub-matrix and get the corresponding matrix eigenproblem equation as:

$$\frac{1}{m} K^{(m)} U^{(m)} = U^{(m)} \Lambda^{(m)}$$

with $U^{(m)} \in \mathbb{R}^{m \times m}$ is column orthonormal and $\Lambda^{(m)}$ is a diagonal matrix.

Now we can derive the approximations for the eigenfunctions and eigenvalues of the kernel k

$$\lambda_i \approx \frac{\lambda_i^{(m)} \cdot N}{m}, \quad \varphi_i(\mathbf{x}') \approx \frac{\sqrt{m/N}}{\lambda_i^{(m)}} \mathbf{k}_x'^{,\top} \mathbf{u}_i^{(m)}, \tag{4}$$

where $\mathbf{u}_i^{(m)}$ is the ith column of $U^{(m)}$. Thus, we can approximate φ_i at an arbitrary point $\mathbf{x}'$ as long as we know the vector $\mathbf{k}_x' = (k(\mathbf{x}_1, \mathbf{x}'), ..., k(\mathbf{x}_m, \mathbf{x}'))$. For a given $N \times N$ Gram matrix K one may randomly choose m rows and respective columns. The corresponding indices are called landmarks, and should be chosen such that the data distribution is sufficiently covered. Strategies how to chose the landmarks have recently been addressed in [8,21] and [22,23]. We denote these rows by $K_{(m,N)}$. Using the formulas Eq. (4) we can reconstruct the original kernal matrix,

$$\tilde{K} = \sum_{i=1}^{m} 1/\lambda_i^{(m)} \cdot K_{(m,N)}^T (\mathbf{u}_i^{(m)})^T (\mathbf{u}_i^{(m)}) K_{(m,N)},$$

where $\lambda_i^{(m)}$ and $\mathbf{u}_i^{(m)}$ correspond to the $m \times m$ eigenproblem (3). Thus we get the approximation,

$$\tilde{K} = K_{(N,m)} K_{(m,m)}^{-} K_{(m,N)}. \tag{5}$$

This approximation is exact, if $K_{(m,m)}$ has the same rank as K.

Nyström Approximation Based Frobenius Norm. Instead of the Frobenius norm definition given in Eq. (2) we will use an equivalent formulation based on the trace of the matrix:

$$\left\|\hat{k} - k\right\|_F = \sqrt{\sum_{i=1}^{N} \ddot{k}(\mathbf{x}_i, \mathbf{y}_i)} \tag{6}$$

$\ddot{k}(\mathbf{x}_i, \mathbf{y}_i)$ is given by the (i,j)'th entry of the matrix $\ddot{K}$ defined as $\ddot{K} = (\hat{K} - K) \cdot (\hat{K} - K)^\top$ (in matrix notation). This formulation is useful because we can obtain the diagonal elements of a Nyström approximated matrix very easy.

We approximate $\hat{k}(\mathbf{x}_i, \mathbf{y}_j)$ and $k(\mathbf{x}_i, \mathbf{y}_j)$ using the Nyström approximation and obtain matrices $\hat{K}_{(nm)}, \hat{K}^{-1}_{(nm)}$ and $K_{(nm)}, K^{-1}_{(nm)}$ as defined before. With some basic algebraic operations one gets the following equation for the Frobenius norm of the difference of two Nyström approximated matrices (in matrix notation). Let $C = \hat{K}_{(nm)} \otimes K_{(nm)}$ and $W = C^{-1}_{(m,m)}$. Further we introduce matrices $\hat{C}$ with entries $\hat{C}_{[i,j]} = C^2_{[i,j]}, \hat{W} = \hat{C}^{-1}_{(m,m)}$ and C' with entries $C'_{[i,j]} = K^2_{[i,j]}, W' = K^{-1}_{(m,m)}$. Then the approximated Frobenius norm can be derived as:

$$\left\|\hat{k} - k\right\|_F = \sqrt{\sum(\sum(\hat{C} \cdot \hat{W})) \cdot \hat{C}^\top + (\sum(C' \cdot W')) \cdot C'^{,\top} - 2 \cdot (\sum(C \cdot W)) \cdot C^\top} \tag{7}$$

This operation can be done with linear costs.

Complexity and Error Analysis. The Nyström approximation of k can be calculated once prior to random fourier feature selection, with costs of $\mathcal{O}(N \times m + m^3)$, to obtain the two submatrices of the Nyström approximation. If we assume that $m \ll N$ this is summarized by $\mathcal{O}(N \times m)$. The Nyström approximation of $\hat{k}$ needs to be calculated in each enhancing step of the feature construction. If we add one feature per iteration the costs of m^3 can be avoided by use of the matrix inversion lemma. If we assume that in each step 1 feature is added. If we restrict the number of added features by D_{max} extra costs of $\mathcal{O}(D_{max} \times N \times m)$ are present to calculate the Nyström approximation of k. If we assume that $D_{max} \ll N$ this is again reduced to $\mathcal{O}(N \times m)$.

The calculation of the Frobenius norm can be done in linear time using Eq. (7). Hence, we finally have costs of $\mathcal{O}(N \times m)$ for generating the random fourier features. The number of effectively chosen random fourier features is in general much smaller then D_{max}. In the following we use $m = 50$ and $D_{max} = 5000$ and report crossvalidation accuracies using the approximated $\hat{k}$ in comparison to k for different classification tasks.

As mentioned before and shown from the runtime analysis the approach is reasonable only if the number of landmarks m is low with respect to N, or the intrinsic dimensionality of the datasets is low, respectively. Taking e.g. an RBF kernel, the σ parameter controls the width of the Gaussian. If the RBF kernel is employed in a kernel classifier we observe that for very small σ

a Nearest Neighbor approach is approximated and the intrinsic dimensionality of the data or number of non-vanishing eigenvalues gets large. In these cases the RBF representation can not be approximated without a corresponding high loss in the prediction accuracy of the model and our approach can not be used. In the proposed procedure we observe two approximation errors, namely the error introduced by the random fourier feature approximation and the error introduced by the Nyström approximation. We have:

$$\left\|\hat{K} - K\right\|_F = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N} |\hat{K}(\mathbf{x}_i, \mathbf{x}_j) - K(\mathbf{x}_i, \mathbf{x}_j)|^2} \tag{8}$$

where $\hat{K}$ is the Nyström approximated kernel matrix of the kernel matrix obtained from the random fourier features of the training data. By the triangle inequality we get

$$\left\|\hat{K} - K\right\|_F \leq \left\|\varphi^\top \varphi - K\right\|_F + \left\|\varphi^\top \varphi - \tilde{K}\right\|_F \tag{9}$$

where $\tilde{K}$ is the Nyström approximated kernel matrix of the linear kernel matrix on the random fourier features and $\varphi^\top \varphi$ is given as

$$\varphi(\mathbf{x}_i)^\top \varphi(\mathbf{x}_j) = \frac{\alpha}{D} \sum_{l=1}^{D} \cos(\omega_l^\top (\mathbf{x}_i - \mathbf{x}_j)) \tag{10}$$

where D is the number of random fourier features $\omega_l \sim N(0, I_d), \alpha > 0$ and $\mathbf{x}_i, \mathbf{x}_j$ are training points with RFF feature values stored in φ. In the following we derive and combine the bounds for both approximation schemes. The Frobenius error of the approximated kernel using the random fourier features is given as

$$\left\|\varphi^\top \varphi - K\right\|_F = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N} |\varphi(\mathbf{x}_i)^\top \varphi(\mathbf{x}_j) - K(\mathbf{x}_i, \mathbf{x}_j)|^2} \tag{11}$$

with K as the kernel matrix of the training points $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$. For a fixed pair of points $(\mathbf{x}_i, \mathbf{x}_j)$ we have:

$$Pr\{|\varphi(\mathbf{x}_i)^\top \varphi(\mathbf{x}_j) - \underbrace{E[\varphi(\mathbf{x}_i)^\top \varphi(\mathbf{x}_j)]}_{K(\mathbf{x}_i, \mathbf{x}_j)}| > t\} =$$

$$= Pr\{|\frac{\alpha}{D} \sum_{l=1}^{D} \cos(\omega_l^\top (\mathbf{x}_i - \mathbf{x}_j)) - K(\mathbf{x}_i, \mathbf{x}_j)| > t\} \tag{12}$$

$$\leq 2 \exp\left\{\frac{-t^2 D}{2\alpha^2}\right\} \tag{13}$$

the last inequation follows by the Höffding inequality because the terms $\cos(\omega_l^\top (\mathbf{x}_i, \mathbf{x}_j))$ are independent w.r.t. $\{\omega_1, \ldots, \omega_D\}$ and are bounded $\in [-1, 1]$.

The above condition can be generalized asymptotically to all pairs from $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$. Hence the following holds simultaneously for all pairs:

$$Pr\{\exists(i,j) : |\varphi(\mathbf{x}_i)^\top \varphi(\mathbf{x}_j) - K(\mathbf{x}_i, \mathbf{x}_j)| > t\} \leq 2N^2 \exp\left\{\frac{-t^2 D}{2\alpha^2}\right\} \tag{14}$$

by union bound. Hence we obtain

$$\left\|\varphi^\top \varphi - K\right\|_F \leq \sqrt{N^2 t^2} = N \cdot t \tag{15}$$

with probability of at least $1 - 2N^2 \exp\left\{\frac{-t^2 D}{2\alpha^2}\right\}$. To get the failure probability to an arbitrary small δ:

$$\begin{aligned} 2N^2 \exp\left\{\frac{-t^2 D}{2\alpha^2}\right\} &= \delta \\ \frac{t^2 D}{2\alpha^2} &= \log\frac{2N^2}{\delta} = \log\frac{2}{\delta} + 2\log N \\ t &= \alpha\sqrt{\frac{2}{D}\left(\log\frac{2}{\delta} + 2\log N\right)} \end{aligned}$$

We get

$$\left\|\varphi^\top \varphi - K\right\|_F \leq N \cdot \alpha\sqrt{\frac{2}{D}\left(\log\frac{2}{\delta} + 2\log N\right)} = \tilde{\mathcal{O}}\left(\frac{N}{\sqrt{D}}\right) \tag{16}$$

We can ensure that $\left\|\varphi^\top \varphi - K\right\|_F \leq \epsilon$ by choosing D large enough:

$$\begin{aligned} N \cdot \alpha\sqrt{\frac{2}{D}\left(\log\frac{2}{\delta} + 2\log N\right)} &\leq \epsilon \\ \frac{2}{D}\left(\log\frac{2}{\delta} + 2\log N\right) \leq \frac{\epsilon^2}{N^2\alpha^2} \rightarrow \frac{2}{D} &\leq \frac{\epsilon^2}{\left(\log\frac{2}{\delta} + 2\log N\right) N^2\alpha^2} \\ D &\geq \frac{2N^2\alpha^2\left(\log\frac{2}{\delta} + 2\log N\right)}{\epsilon^2} = \tilde{\mathcal{O}}\left(\frac{N}{\epsilon}\right) \end{aligned}$$

For the second approximation error we bound the error of inner product of the random fourier feature vectors obtained from the training data with respect to a Nyström approximation of the kernel based on the random fourier features. This is just a classical Nyström approximation of a kernel matrix. Hence we can use bounds already provided in [24]. According to Theorem 2 given in [24] the following inequality holds with probability of at least $1 - \delta$:

$$\left\|\varphi^\top \varphi - \hat{K}\right\|_F \leq \left\|\varphi^\top \varphi - K_k\right\|_F + \left[\frac{64D}{m}\right]^{\frac{1}{4}} NK_{max}\left[1 + \sqrt{\frac{N-m}{n-1/2}\frac{1}{\beta(m,N)}\log\frac{1}{\delta}d^K_{max}/K^{\frac{1}{2}}_{max}}\right]^{\frac{1}{2}} \tag{17}$$

with $\beta(m,n) = 1 - \frac{1}{2\max m,N-m}$ and K_k the best k approximation of K and K_{max} the maximal diagonal entry of K and d^K_{max} the maximum Euclidean distance defined over K. Which maybe summarized in accordance to [22] as $\left\|\varphi^\top\varphi - K_k\right\|_F + [\frac{D}{m}]^{\frac{1}{4}} N \left\|K\right\|_2$. Combining both bounds we get

$$\begin{aligned}\left\|\hat{K} - K\right\|_F &\leq \left\|\varphi^\top\varphi - K\right\|_F + \left\|\varphi^\top\varphi - \tilde{K}\right\|_F \\ &\leq \left\|\varphi^\top\varphi - K_k\right\|_F + \\ &\quad N \cdot \left(\alpha\sqrt{\frac{2}{D}\left(\log\frac{2}{\delta} + 2\log N\right)} + [\frac{D}{m}]^{\frac{1}{4}} \left\|K\right\|_2\right) \end{aligned} \tag{18}$$

We see that both approximation terms increase as $\tilde{\mathcal{O}}(N)$ - that is, up to log factors the kernel approximation error increases linearly with the number of training points N. This was expected since the gram matrix K has size $N \times N$. We may also notice a tradeoff for the value of D: The random Fourier feature approximation bound tightens as k increases whereas the Nyström approximation loosens. (One could possibly use the value of D that minimizes the approximation error bound, although we have not tried this in the experiments.) The approximation error bound presented here is uniform only over the training points - which was much simpler to achieve than a bound that holds uniformly over the whole input domain - as in the original paper [10]. Nevertheless we can still expect it to be informative since for kernel based learning there exist generalization bounds whose complexity term only depends on the gram matrix constructed from the training set (e.g. the Rademacher complexity for kernel based linear classification works out as the trace of the gram matrix).

5 Experiments

We evaluate the approach on multiple public datasets most of them already used in the paper [10][1]. For the Nyström approximation step we use 50 landmarks. The checkerboard data (checker) is a two class problem consisting of 9000 2d samples organized like a checkerboard on a 3×3 grid. The data are separable with low error using an rbf kernel. The coil-20 dataset (coil) consists of 1440 image files in 16384 dimensions from the coil database categorized in 20 classes. The spam database consists of 4601 samples in 57 dimensions in two classes. The adult dataset consists of 30162 samples in 44 dimensions given in 2 classes[2]. The coderna dataset with 59535 samples and 8 dimensions from http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/. Two further used datasets are the famous USPS data with 11000 samples in 256 dimensions and the MNIST data with 60000 samples and 256 dimensions. Both datasets are organized in 10 classes

[1] We skip the KDDCUP data which is very simple as already reported in [10]. Further for some of the datasets the original configuration was not exactly reconstructable e.g. Adult data such that we could not directly copy results from.

[2] Preprocessed as reported in http://ssdi.di.fct.unl.pt/nmm/scripts/mdatasets/.

Table 1. Test set accuracy (% ± std) of the various benchmark datasets for constructing small sets of random fourier features. The second row of each dataset contains the mean cputime of a single cycle in the crossvalidation.

	D	D^*	SVM+Full-RFF	SVM+Small-RFF	LS+Full-RFF	LS+Small-RFF
coil	5000	175	98.75 ± 1.46	97.22 ± 2.15	99.38 ± 0.51	96.67 ± 1.49
			5 s	0.1 s	7 s	1 s
spam	5000	185	92.98 ± 0.76	89.74 ± 1.18	92.87 ± 1.09	89.13 ± 1.71
			1 s	0.1 s	1 s	0.1 s
checker	5000	20	100.00 ± 0	99.96 ± 0.06	100.00 ± 0.00	99.09 ± 0.29
			1.5 s	0.02 s	2 s	0.02 s
usps	5000	820	97.96 ± 0.32	96.85 ± 0.49	98.11 ± 0.42	96.25 ± 0.76
			19 s	3 s	28.15 s	4.33 s
adult	500	105	82.44 ± 0.50	82.20 ± 0.58	82.48 ± 0.44	82.08 ± 0.76
			1 s	0.35 s	0.65 s	0.14 s
code-rna	500	20	95.31 ± 0.24	91.40 ± 0.50	94.65 ± 0.37	91.27 ± 0.39
			1.4 s	0.21 s	1.4 s	0.1 s
mnist	5000	235	96.30 ± 0.30	89.89 ± 0.41	95.12 ± 0.20	87.12 ± 0.43
			3.4 min	14 s	1.5 min	6 s
cover	1000	90	83.44 ± 0.20	75.85 ± 0.18	80.56 ± 0.19	76.18 ± 0.18
			19 min	48 s	3 min	17 s
forest	1000	150	83.15 ± 0.08	76.18 ± 0.18	79.94 ± 0.11	74.31 ± 0.21
			2.8 min	2 s	1.9 min	6 s

originating from a character recognition problem. Finally the covertype data with 495141 entries (classes 1 and 2) and 54 dimensions and the Forest data with 522.000 samples and 54 dimensions both taken from the UCI database were analyzed. All datasets have been z-transformed. For the σ parameter of the rbf kernel we use values reported before elsewhere. To evaluate the classification performance we follow [10] and use a least squares regression (LS) model as well as the liblinear, which is a high performance linear Support Vector Machine[3]. The parameter C of the Liblinear-SVM was fixed to 1 as suggested by the liblinear authors. Multiclass problems have been approached in LS using a one vs rest scheme. In Table 1 we report 10-fold crossvalidation results and the minimal number of features D* as obtained by the proposed strategy. The maximal number of random fourier features D per dataset is in general 5000 as suggested in [10] with exceptions for the larger datasets to keep memory consumption tractable only 500–1000 features where chosen.

For the coil data we see that the identified small RFF-model contains 29−times less features than the full model while loosing ≈2 % discrimination accuracy on the test set. For the spam database we observe a similar result with

[3] http://www.csie.ntu.edu.tw/~cjlin/liblinear/.

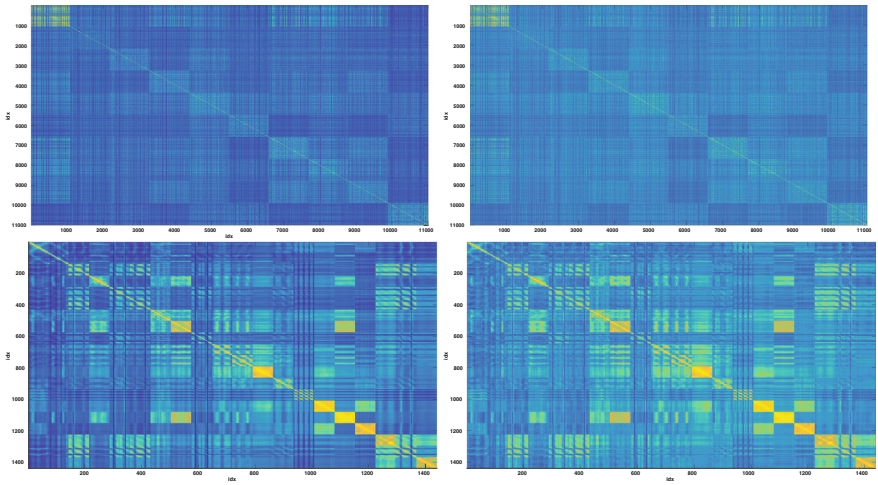

Fig. 1. Top left: reconstruction of the USPS radial basis function kernel with 5000 random fourier features, right: reconstruction of the USPS radial basis function kernel with the identified random fourier features. Bottom-left: reconstruction of the coil radial basis function kernel with 5000 random fourier features and right with the random fourier features as obtained by the proposed approach.

27−times less features and a small decay in the accuracy of 3 % for SVM. At the simulated checkerboard data we have almost the same accuracy in the reduced set while the number of features is reduced by a factor of 250. For the USPS data we have ≈6 times less features with almost the same prediction accuracy, slightly reduced by 1–2 %. The Adult dataset keeps almost the same accuracy while having 5-times less features similar observations can be made for the code-dna data. For MNIST the accuracy drops by 7–8 % with 21 times less features. Finally the cover data are represented by 28 times less features with a similar good accuracy like the full model and the forest data could be represented with 22 times less features with a slight decay on 7 % in the accuracy. For USPS and MNIST we found that the number of remaining features is still a bit high which can be potentially attributed to a more complex eigenvalue structure of these datasets such that the proposed test was less efficient. The other datasets have basically almost the same accuracy on a drastically reduced feature set. For the coil and the usps data the kernel reconstructions are exemplarily shown in Fig. 1.

6 Conclusions

In this paper we proposed a test for selecting a small set of random fourier features such that the approximated shift invariant kernel is close to the original one with respect to the Frobenius norm. In general we found that the proposed approach is efficient to reduce the number of features, already during the construction, by in general a magnitude or more with low costs with respect to N.

The approach is especially applicable if the approximated kernel is of low rank and N is large. Thereby the proposed selection procedure is efficient to obtain small random fourier features sets with high representation accuracy. The effect of sometimes reduced accuracy for random fourier features as observed in [14] could not be confirmed as long as the RFF set is either large enough or appropriately chosen by the proposed method. The proposed approach saves runtime and memory costs during training but is also very valuable if memory is constrained under test conditions e.g. within an embedded system environment. The obtained transformation matrix P has $d \times D$ coefficients which is most often small enough to be of use also under system conditions with limited resources. The original data needs to be transformed into the random fourier feature space using P by a simple matrix multiplication and can subsequently be fed into a linear classifier. The obtained models are in general very efficient as seen above. The small $D*$ also avoids the need to sparsify the linear models by using ridge regression (instead of simple LS) or sparse linear SVM models like the support feature machine [18], such that efficient high performance implementations of linear classifiers can be directly used. In future work we will analyze the effect of our approach on tensor sketching [25] which was used to approximate polynomial kernels.

Acknowledgment. Marie Curie Intra-European Fellowship (IEF): FP7-PEOPLE-2012-IEF (FP7-327791-ProMoS) is greatly acknowledged.

References

1. Chitta, R., Jin, R., Jain, A.K.: Efficient kernel clustering using random Fourier features. In: 12th IEEE International Conference on Data Mining, ICDM, pp. 161–170. IEEE (2012)
2. Villmann, T., Haase, S., Kaden, M.: Kernelized vector quantization in gradient-descent learning. Neurocomputing **147**, 83–95 (2015)
3. Schleif, F.-M., Villmann, T., Hammer, B., Schneider, P.: Efficient kernelized prototype-based classification. J. Neural Syst. **21**(6), 443–457 (2011)
4. Hofmann, D., Schleif, F.-M., Hammer, B.: Learning interpretable kernelized prototype-based models. Neurocomputing **131**, 43–51 (2014)
5. Schleif, F.-M., Zhu, X., Gisbrecht, A., Hammer, B.: Fast approximated relational and kernel clustering. In: Proceedings of ICPR 2012, pp. 1229–1232. IEEE (2012)
6. Si, S., Hsieh, C.-J., Dhillon, I.S.: Memory efficient kernel approximation. In: Proceedings of the 31th International Conference on Machine Learning, ICML, volume 32 of JMLR Proceedings, pp. 701–709. JMLR.org (2014)
7. Cortes, C., Mohri, M., Talwalkar, A.: On the impact of kernel approximation on learning accuracy. In: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, AISTATS, volume 9 of JMLR Proceedings, pp. 113–120. JMLR.org (2010)
8. Zhang, K., Kwok, J.T.: Clustered Nyström method for large scale manifold learning and dimension reduction. IEEE Trans. Neural Netw. **21**(10), 1576–1587 (2010)
9. Gisbrecht, A., Schleif, F.-M.: Metric and non-metric proximity transformations at linear costs. Neurocomputing **167**, 643–657 (2015)

10. Rahimi, A., Recht, B.: Random features for large-scale kernel machines. In: Proceedings of the 21st Annual Conference on Neural Information Processing Systems, NIPS 2007. Curran Associates, Inc. (2007)
11. Agarwal, A., Kakade, S.M., Karampatziakis, N., Song, L., Valiant, G.: Least squares revisited: scalable approaches for multi-class prediction. In: Proceedings of the 31th International Conference on Machine Learning, ICML, volume 32 of JMLR Proceedings, pp. 541–549. JMLR.org (2014)
12. Bunte, K., Kaden, M., Schleif, F.-M.: Low-rank kernel space representations in prototype learning. WSOM 2016. AISC, vol. 428, pp. 341–353. Springer, Switzerland (2016)
13. Schleif, F.-M., Hammer, B., Villmann, T.: Margin based active learning for LVQ networks. Neurocomputing **70**(7–9), 1215–1224 (2007)
14. Yang, T., Li, Y.-F., Mahdavi, M., Jin, R., Zhou, Z.-H., Nystroem method vs random Fourier features: a theoretical and empirical comparison. In: Proceedings of the 26st Annual Conference on Neural Information Processing Systems, NIPS 2012, pp. 485–493 (2012)
15. Durrant, R.J., Kabán, A.: Random projections as regularizers: learning a linear discriminant from fewer observations than dimensions. Mach. Learn. **99**(2), 257–286 (2015). doi:10.1007/s10994-014-5466-8
16. Freund, Y., Dasgupta, S., Kabra, M., Verma, N.: Learning the structure of manifolds using random projections. In: Proceedings of the 21st Annual Conference on Neural Information Processing Systems, NIPS 2007. Curran Associates, Inc. (2007)
17. Vergara, J.R., Estévez, P.A.: A review of feature selection methods based on mutual information. Neural Comput. Appl. **24**(1), 175–186 (2014)
18. Klement, S., Anders, S., Martinetz, T.: The support feature machine: classification with the least number of features and application to neuroimaging data. Neural Comput. **25**(6), 1548–1584 (2013)
19. Schleif, F.-M., Villmann, T., Zhu, X.: High dimensional matrix relevance learning. In: Proceedings of IEEE Internation Conference on Data Mining Workshop (ICDMW), pp. 661–667 (2014)
20. Williams, C.K.I., Seeger, M.: Using the Nyström method to speed up kernel machines. In: Proceedings of the 13th Annual Conference on Neural Information Processing Systems, NIPS 2000, pp. 682–688 (2000)
21. Zhang, K., Tsang, I.W., Kwok, J.T.: Improved Nystrom low-rank approximation and error analysis. In: Proceedings of the 25th International Conference on Machine Learning, ICML 2008, pp. 1232–1239. ACM, New York (2008)
22. Gittens, A., Mahoney, M.W.: Revisiting the Nystrom method for improved large-scale machine learning. In: Proceedings of the 30th International Conference on Machine Learning, ICML 2013, volume 28 of JMLR Proceedings, pp. 567–575. JMLR.org (2013)
23. De Brabanter, K., De Brabanter, J., Suykens, J.A.K., De Moor, B.: Optimized fixed-size kernel models for large data sets. Comput. Stat. Data Anal. **54**(6), 1484–1504 (2010)
24. Kumar, S., Mohri, M., Talwalkar, A.: Sampling methods for the Nyström method. J. Mach. Learn. Res. **13**, 981–1006 (2012)
25. Pham, N., Pagh, R.: Fast and scalable polynomial kernels via explicit feature maps. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, pp. 239–247. ACM (2013)