Random Feature Maps for the Itemset Kernel

Kyohei Atarashi
Hokkaido University
atarashi_k@complex.ist.hokudai.ac.jp

Subhransu Maji
University of Massachusetts, Amherst
smaji@umass.cs.edu

Satoshi Oyama
Hokkaido University/RIKEN AIP
oyama@ist.hokudai.ac.jp

Abstract

Although kernel methods efficiently use feature combinations without computing them directly, they do not scale well with the size of the training dataset. Factorization machines (FMs) and related models, on the other hand, enable feature combinations efficiently, but their optimization generally requires solving a non-convex problem. We present random feature maps for the itemset kernel, which uses feature combinations, and includes the ANOVA kernel, the all-subsets kernel, and the standard dot product. Linear models using one of our proposed maps can be used as an alternative to kernel methods and FMs, resulting in better scalability during both training and evaluation. We also present theoretical results for a proposed map, discuss the relationship between factorization machines and linear models using a proposed map for the ANOVA kernel, and relate the proposed feature maps to prior work. Furthermore, we show that the maps can be calculated more efficiently by using a signed circulant matrix projection technique. Finally, we demonstrate the effectiveness of using the proposed maps for real-world datasets.

1 Introduction

Kernel methods enable learning in high, possibly infinite dimensional feature spaces without explicitly expressing them. In particular, kernels that model feature combinations such as polynomial kernels, the ANOVA kernel, and the all-subsets kernel (Blondel et al. 2016a; Shawe-Taylor and Cristianini 2004) have been shown to be effective for a number of tasks in computer vision and natural language understanding (Lin, RoyChowdhury, and Maji 2015; Fukui et al. 2016). However their scalability remains a challenge; support vector machines (SVMs) with non-linear kernels require $O(n^2)$ time and $O(n^2)$ memory for training and $O(n)$ time and memory for evaluation, where n is the number of training instances (Chang and Lin 2011).

To address this issue several researchers have proposed randomized feature maps $Z(\cdot) : \mathbb{R}^d \mapsto \mathbb{R}^D$ for kernels $K(\cdot,\cdot) : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$ that satisfy

$$\mathbb{E}[\langle Z(\boldsymbol{x}), Z(\boldsymbol{y})\rangle] = K(\boldsymbol{x}, \boldsymbol{y}). \tag{1}$$

The idea is to perform classification, regression, or clustering on a corresponding high-dimensional feature space approximately but efficiently using linear models in a low dimensional space by mapping the data points using $Z(\cdot)$. Examples include *random Fourier feature maps* that approximate shift-invariant kernels: $K(\boldsymbol{x}, \boldsymbol{y}) = k(|\boldsymbol{x} - \boldsymbol{y}|)$ (Rahimi and Recht 2008); *random Maclaurin feature maps* that approximate dot product kernels (Kar and Karnick 2012): $K(\boldsymbol{x}, \boldsymbol{y}) = k(\langle \boldsymbol{x}, \boldsymbol{y}\rangle)$; and *tensor sketching* for polynomial kernels: $K^m_{\mathrm{P}}(\boldsymbol{x}, \boldsymbol{y}; c) := (c + \langle \boldsymbol{x}, \boldsymbol{y}\rangle)^m$ (Pham and Pagh 2013). Although polynomial kernels are dot product kernels and can be approximated by random Maclaurin feature maps, tensor sketching can be more efficient.

Factorization machines (FMs) (Rendle 2010; 2012) and variants (Blondel et al. 2016a; 2016b; Novikov, Trofimov, and Oseledets 2016) also model feature combinations without explicitly computing them, similar to kernel methods, but have better scalability during evaluation. These methods can be thought of as a two-layer neural network with polynomial activations with a fixed number of learnable parameters (See Equation (5)). However, unlike kernel methods, their optimization problem is generally non-convex and difficult to solve. But due to their efficiency during evaluation FMs are attractive for large-scale problems and have been successfully applied to applications such as link prediction and recommender systems. This work analyzes the relationship between polynomial kernel models and factorization machines in more detail.

Our contributions. We present a random feature map for the *itemset kernel* that takes into account all feature combinations within a family of itemsets $\mathcal{S} \subseteq 2^{[d]}$. To the best of our knowledge, the random feature map for the itemset kernel is novel. The itemset kernel includes the ANOVA kernel, all-subsets kernel, and standard dot product, so linear models using this map are an alternative to the ANOVA or all-subsets kernel SVMs, FMs, and all-subsets model. They scale well with the size of the training dataset, unlike kernel methods, and their optimization problem is convex and easy to solve, unlike that of FMs. We also present theoretical analyses of the proposed random feature map and discuss the relationship between linear models trained on these features and factorization machines. Furthermore, we present a faster and more memory-efficient random feature map for the ANOVA kernel based on the signed circulant matrix technique (Feng, Hu, and Liao 2015). Finally, we evaluate the effectiveness of the feature maps on several datasets.

2 Background and Related Work

2.1 Kernels Using Feature Combinations

First, we present the ANOVA kernel and all-subsets kernel, which use feature combinations (Blondel et al. 2016a; Shawe-Taylor and Cristianini 2004), and define the itemset kernel. We also describe several models that use feature combinations.

The ANOVA kernel is similar to the polynomial kernel. The definition of an m-order ANOVA kernel between $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$ is

$$K_{\mathrm{A}}^m(\boldsymbol{x}, \boldsymbol{y}) := \sum_{j_1 < \cdots < j_m}^{d} x_{j_1} \cdots x_{j_m} y_{j_1} \cdots y_{j_m}, \quad (2)$$

where $2 \leq m \leq d \in \mathbb{N}$ is the order of the ANOVA kernel. For convenience, $0/1$-order ANOVA kernels are often defined as $K_{\mathrm{A}}^0(\boldsymbol{x}, \boldsymbol{y}) = 1$ and $K_{\mathrm{A}}^1(\boldsymbol{x}, \boldsymbol{y}) = \langle \boldsymbol{x}, \boldsymbol{y} \rangle$. The difference between the ANOVA kernel and the polynomial kernel is that the ANOVA kernel does not use feature combinations that include the same feature (e.g., $x_1 x_1$, $x_2^2 x_3$) while the polynomial kernel does. Although the evaluation of the m-order ANOVA kernel involves $O(d^m)$ terms, it can be computed in $O(dm)$ time using dynamic programming (Blondel et al. 2016a; Shawe-Taylor and Cristianini 2004). In some applications, ANOVA-kernel-based models have achieved better performance than polynomial-kernel-based models (Blondel et al. 2016a; 2016b). We discuss these models later in this section.

While the ANOVA kernel uses only m-order different feature combinations, the all-subsets kernel K_{all} uses all different feature combinations and is defined as

$$K_{\mathrm{all}}(\boldsymbol{x}, \boldsymbol{y}) := \prod_{j=1}^{d} (1 + x_j y_j). \quad (3)$$

Clearly, evaluation of the all-subsets kernel takes only $O(d)$ time.

Here, we define the *itemset kernel*. For a given family of itemsets $\mathcal{S} \subseteq 2^{[d]}$, where $[d] = \{1, \ldots, d\}$ and $2^{[d]} = \{\emptyset, \{1\}, \{2\}, \ldots, \{d\}, \{1, 2\}, \ldots, [d]\}$, we define the itemset kernel as

$$K_{\mathcal{S}}(\boldsymbol{x}, \boldsymbol{y}) := \sum_{V \in \mathcal{S}} \prod_{j \in V} x_j y_j = \langle \phi_{\mathcal{S}}(\boldsymbol{x}), \phi_{\mathcal{S}}(\boldsymbol{y}) \rangle. \quad (4)$$

The itemset kernel clearly uses feature combinations in the family of itemsets $\mathcal{S}$. The itemset kernel can be regarded as an extension of the ANOVA kernel, all-subsets kernel, and standard dot product. For example, when $\mathcal{S} = 2^{[d]}$, $K_{2^{[d]}}$ clearly uses all feature combinations and hence is equivalent to the all-subsets kernel K_{all} in Equation (3). When $\mathcal{S} = \binom{[d]}{m} := \{V \subseteq [d] \mid |V| = m\}$, the itemset kernel $K_{\mathcal{S}}$ is equivalent to m-order ANOVA kernel K_{A}^m. Furthermore, when $\mathcal{S} = \{\{1\}, \ldots, \{d\}\}$, the itemset kernel $K_{\mathcal{S}}$ clearly represents the standard dot product.

2.2 Factorization Machines

Rendle proposed using a *factorization machine (FM)* as the ANOVA-kernel-based model (Rendle 2010; 2012). The FM model equation is

$$f_{\mathrm{FM}}(\boldsymbol{x}; \boldsymbol{w}, \boldsymbol{P}, \boldsymbol{\lambda}) := \langle \boldsymbol{w}, \boldsymbol{x} \rangle + \sum_{s=1}^{k} \lambda_s K_{\mathrm{A}}^2(\boldsymbol{p}_s, \boldsymbol{x}), \quad (5)$$

where $\boldsymbol{w} \in \mathbb{R}^d$, $\boldsymbol{P} \in \mathbb{P}^{d \times k}$, and $\boldsymbol{\lambda} \in \mathbb{R}^k$ are learnable parameters, and $k \in \mathbb{N}$ is a rank-hyper parameter. The computational cost for evaluating FMs is $O(dkm)$ and does not depend on the amount of training data. However, the FM optimization problem is non-convex and hence challenging. Fortunately, it can be solved relatively efficiently using a coordinate descent method because it is multi-convex w.r.t $w_1, \ldots, w_d, p_{1,1}, \ldots, p_{d,k}$. Although parameter $\boldsymbol{\lambda}$ was not introduced in the original FMs (Rendle 2010; 2012), Blondel et al. showed that introducing $\boldsymbol{\lambda}$ increases the capacity of FMs (Blondel et al. 2016b).

Polynomial networks (PNs) (Livni, Shalev-Shwartz, and Shamir 2014) are models based on polynomial kernels. They are depth two neural networks with a polynomial activation function. Although both PNs and FMs use feature combinations, there is a key difference: PNs can be represented by the polynomial kernel while FMs can be represented by the ANOVA kernel. This difference means PNs use feature combinations among the same features while FMs do not. Experiments have shown that FMs achieve better performance than PNs (Blondel et al. 2016a).

Blondel et al. proposed the *all-subsets model* (Blondel et al. 2016a), which uses all feature combinations:

$$f_{\mathrm{all}}(\boldsymbol{x}; \boldsymbol{P}, \boldsymbol{\lambda}) := \sum_{s=1}^{k} \lambda_s K_{\mathrm{all}}(\boldsymbol{p}_s, \boldsymbol{x}). \quad (6)$$

Although the all-subsets model sometimes performed better than FMs and PNs on link prediction tasks, it tended to have lower performance (Blondel et al. 2016a).

2.3 Random Feature Maps for Polynomial Kernels

The random Maclaurin (RM) feature map (Kar and Karnick 2012) is for dot product kernels: $K(\boldsymbol{x}, \boldsymbol{y}) = k(\langle \boldsymbol{x}, \boldsymbol{y} \rangle)$. It uses the Maclaurin expansion of $k(\cdot)$: $k(x) = \sum_{n=0}^{\infty} a_n x^n$, where $a_n = k^{(n)}(0)/n!$ is the n-th coefficient of the Maclaurin series. It uses two distributions: $p_{\mathrm{order}}(N = n) = 1/p^{n+1}$, where $p > 1$, and the Rademacher distribution (a fair coin distribution). Its computational cost is $O\left(\sum_{s=1}^{D} N_s d\right)$ time and memory, where N_s ($s \in [D]$) is the order of the s-th randomized feature, especially $O(Ddm)$ time and memory when the objective kernel is the homogeneous polynomial kernel: $K_{\mathrm{HP}}^m(\boldsymbol{x}, \boldsymbol{y}) = \langle \boldsymbol{x}, \boldsymbol{y} \rangle^m$.[1]

The tensor sketching (TS) (Pham and Pagh 2013) is a random feature map for homogeneous polynomial kernels : $K_{\mathrm{HP}}^m(\boldsymbol{x}, \boldsymbol{y}) = \langle \boldsymbol{x}, \boldsymbol{y} \rangle^m$. Because polynomial kernels $K_{\mathrm{P}}^m(\boldsymbol{x}, \boldsymbol{y}; c) = (c + \langle \boldsymbol{x}, \boldsymbol{y} \rangle)^m$ can be written as K_{HP}^m by concatenating $\sqrt{c}$ to each vector, a TS can approximate K_{P}^m.

[1] When the objective kernel is a homogeneous polynomial kernel, one can fix $n = m$ and $p_{\mathrm{order}}(N = m) = 1$ otherwise 0; that is, do not sample n.

Algorithm 1 Random Kernel Feature Map

Input: $\boldsymbol{x} \in \mathbb{R}^d, \mathcal{S} \subseteq 2^{[d]}$

1: Generate D Rademacher vectors $\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_D \in \{-1, +1\}^d$

2: Compute D itemset kernels $K_{\mathcal{S}}(\boldsymbol{x}, \boldsymbol{\omega}_s)$ for all $s \in [D]$

Output: $Z(\boldsymbol{x}) = \frac{1}{\sqrt{D}}\big(K_{\mathcal{S}}(\boldsymbol{x}, \boldsymbol{\omega}_1), \ldots, K_{\mathcal{S}}(\boldsymbol{x}, \boldsymbol{\omega}_D)\big)^\top$

Although an RM feature map can also approximate polynomial kernels, a TS can approximate them more efficiently. It uses the count sketch method, which is a method for estimating the frequency of all items in a stream, as a specific random projection that approximates the dot product in the original feature space. Although the standard dot product in the original feature space can be approximated by using only count sketch, Pham and Pagh proposed combining count sketch with a fast algorithm for computing the count sketch of the outer product without computing the outer product directly, which was proposed by Pagh (Pagh 2013). Their tensor sketch algorithm takes $O(m(d + D \log D))$ time and $O(md \log D)$ memory and is thus more efficient than the random Maclaurin algorithm.

Linear models using the TS or RM feature map are a good alternative to polynomial kernel SVMs and PNs (Kar and Karnick 2012; Pham and Pagh 2013). Similarity, although linear models using a random feature map that approximates the itemset kernel would be a good alternative for the ANOVA or all-subsets kernel SVMs, FMs, and all-subsets models, such a map has not yet been reported.

3 Random Feature Map for the Itemset Kernel

We propose a random feature map for the itemset kernel. As shown in Algorithm 1, the proposed *random kernel (RK) map* is simple: (1) generate D Rademacher vectors from a Rademacher distribution and (2) compute D itemset kernels between the original feature vector and each Rademacher vector. We next present some theoretical results for the RK feature map.

Proposition 1. *Let $Z_{\mathrm{RK}} : \mathbb{R}^d \mapsto \mathbb{R}^D$ be the random kernel (RK) feature map in Algorithm 1. Then, for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$ and $\mathcal{S} \subseteq 2^{[d]}$,*

$$\mathbb{E}_{\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_D}[\langle Z_{\mathrm{RK}}(\boldsymbol{x}), Z_{\mathrm{RK}}(\boldsymbol{y})\rangle] = K_{\mathcal{S}}(\boldsymbol{x}, \boldsymbol{y}). \tag{7}$$

Proposition 1 says that the proposed RK feature map approximates the itemset kernel. Hence, linear models using the proposed RK feature map can use feature combinations efficiently and are a good alternative to FMs and all-subsets models.

We next analyze the precision of the RK feature map. Let $\mathcal{E}(\boldsymbol{x}, \boldsymbol{y})$ be the approximation error: $\mathcal{E}(\boldsymbol{x}, \boldsymbol{y}) := \langle Z_{\mathrm{RK}}(\boldsymbol{x}), Z_{\mathrm{RK}}(\boldsymbol{y})\rangle - K_{\mathcal{S}}(\boldsymbol{x}, \boldsymbol{y})$. We assume that the L^1 norm of the feature vector is bounded: $\|\boldsymbol{x}\|_1 \leq R$, where $R \in \mathbb{R}_{++}$. This assumption is the same as the one used in previous research (Kar and Karnick 2012; Pham and Pagh 2013; Rahimi and Recht 2008). For convenience, we use the same notation as Kar and Karnick (Kar and Karnick 2012): $\mathcal{B}_p(\boldsymbol{0}, R) = \{\boldsymbol{x} \mid \|\boldsymbol{x}\|_p \leq R\}$. With this notation, the assumption above is written as $\boldsymbol{x} \in \mathcal{B}_1(\boldsymbol{0}, R)$. Then, we have the following useful absolute error bound.

Lemma 1. *For all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{B}_1(\boldsymbol{0}, R) \subset \mathbb{R}^d$, and $\mathcal{S} \subseteq 2^{[d]}$,*

$$p(|\mathcal{E}(\boldsymbol{x}, \boldsymbol{y})| \geq \varepsilon) \leq 2\exp\left(\frac{-D\varepsilon^2}{2e^{4R}}\right). \tag{8}$$

This upper bound does not depend on the family of itemsets $\mathcal{S}$ or on the dimension of the original feature vectors d. This result comes from the assumption that data points are restricted in $\mathcal{B}_1(\boldsymbol{0}, R)$.

Next, we consider the uniform bound on the absolute error of the RK feature map. Kar and Karnick (Kar and Karnick 2012) derived the uniform bound on the absolute error of the RM feature map and we follow their approach. Let the domain of feature vectors $\mathcal{B} \subset \mathcal{B}_1(\boldsymbol{0}, R)$ be the compact subset of $\mathbb{R}^d$. Then, $\mathcal{B}$ can be covered by a finite number of balls (Cucker and Smale 2002), and one can obtain the following uniform bound.

Lemma 2. *Let $\mathcal{B} \subset \mathcal{B}_1(\boldsymbol{0}, R)$ be a compact subset of $\mathbb{R}^d$. Then, for all $\mathcal{S} \subseteq 2^{[d]}$,*

$$p\left(\sup_{\boldsymbol{x}, \boldsymbol{y} \in \mathcal{B}} |\mathcal{E}(\boldsymbol{x}, \boldsymbol{y})| \geq \varepsilon\right) \leq 2\left(\frac{32R\sqrt{d}e^{2R}}{\varepsilon}\right)^{2d} \exp\left(-\frac{D\varepsilon^2}{8e^{4R}}\right). \tag{9}$$

This uniform bound says that, by taking $D = \Omega\Big(\frac{de^{4R}}{\varepsilon^2} \log\Big(\frac{R\sqrt{d}e^{2R}}{\varepsilon\delta}\Big)\Big)$, the absolute error is uniformly lower than a ε with a probability of at least $1 - \delta$. This uniform bound also does not depend on the family of itemsets $\mathcal{S}$; it depends only on ε, the dimension of random feature map D, the dimension of the original feature vectors d, and the upper bound on the L^1 norm of the original feature vectors R. The behavior of this uniform bound w.r.t d, ε, and δ is expressed in the form of $D = \Omega\Big(\frac{d}{\varepsilon^2} \log\Big(\frac{\sqrt{d}}{\varepsilon\delta}\Big)\Big)$. This is the same as for the RM feature map (Kar and Karnick 2012).

We have discussed the upper bounds of the RK feature map for the itemset kernel. Next, we consider the absolute error bound for $K_{\mathcal{S}} = K_{\mathrm{A}}^m$ (that is, $\mathcal{S} = \binom{[d]}{m}$). Here, we also assume that $\boldsymbol{x} \in \mathcal{B}_1(\boldsymbol{0}, R)$.

Lemma 3. *Let $\mathcal{S} = \binom{[d]}{m}$. Then, for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{B}_1(\boldsymbol{0}, R) \subset \mathbb{R}^d$,*

$$p(|\mathcal{E}(\boldsymbol{x}, \boldsymbol{y})| \geq \varepsilon) \leq 2\exp\left(-\frac{D\varepsilon^2}{2R^{4m}}\right). \tag{10}$$

The absolute error bound of Lemma 3 is the same as the absolute error bound of the Tensor Sketching (Pham and Pagh 2013).

As described above, the algorithm of the proposed RK feature map uses the Rademacher distribution for random vectors. Here, we discuss the generalized RK feature map, which allows the use of other distributions.

Proposition 2. *If the distribution of $\boldsymbol{\omega}_s$ for all $s \in [D]$ in Algorithm 1 has (i) a mean of 0 and (ii) a variance of 1, the RK feature map approximates the itemset kernel.*

There are many distributions with a mean of 0 and a variance of 1: the standard Gaussian distribution $\mathcal{N}(0,1)$, the uniform distribution $\mathcal{U}(-\sqrt{3},\sqrt{3})$, the Laplace distribution $\mathrm{Laplace}\left(0,1/\sqrt{2}\right) = \frac{1}{\sqrt{2}}\exp\left(-\sqrt{2}|\omega|\right)$, and so on. Which distribution should be used? The next lemma says that the Rademacher distribution should be used.

Lemma 4. *Let $\mathfrak{P}_{0,1}$ be the set of all distributions with a mean of 0 and a variance of 1, and let $p^* \in \mathfrak{P}_{0,1}$ be the Rademacher distribution. Then, for all $p \in \mathfrak{P}_{0,1}$ and $\mathcal{S} \subset 2^{[d]}$,*

$$\sup_{\boldsymbol{x},\boldsymbol{y}\in\mathcal{B}_\infty(0,R)} \mathbb{V}_{\boldsymbol{\omega}_1,\ldots,\boldsymbol{\omega}_D\sim p^*}[\langle Z_{\mathrm{RK}}(\boldsymbol{x}), Z_{\mathrm{RK}}(\boldsymbol{y})\rangle] \leq \sup_{\boldsymbol{x},\boldsymbol{y}\in\mathcal{B}_\infty(0,R)} \mathbb{V}_{\boldsymbol{\omega}_1,\ldots,\boldsymbol{\omega}_D\sim p}[\langle Z_{\mathrm{RK}}(\boldsymbol{x}), Z_{\mathrm{RK}}(\boldsymbol{y})\rangle]. \quad (11)$$

That is, a Rademacher distribution achieves the minimax optimal variance for the RK feature map among the valid distributions.

Finally, we discuss the computational complexity of the RK feature map in two special cases. When $K_{\mathcal{S}}(\cdot,\cdot) = K_{\mathrm{A}}^m(\cdot,\cdot)$, a D-dimensional RK feature map takes $O(Ddm)$ time and $O(Dd)$ memory because an m-order ANOVA kernel can be computed in $O(dm)$ time and $O(m)$ memory by using dynamic programming (Blondel et al. 2016a; Shawe-Taylor and Cristianini 2004). This is the same as the computational cost for an RM feature map for an m-order polynomial kernel. For $K_{\mathcal{S}}(\cdot,\cdot) = K_{\mathrm{all}}(\cdot,\cdot)$, a D-dimensional RK feature map can be computed in $O(Dd)$ time and $O(Dd)$ memory.

4 Loglinear Time RK Feature Map for the ANOVA Kernel

As described above, the computational cost of the proposed RK feature map in Algorithm 1 clearly depends on the computational cost of the itemset kernel $K_{\mathcal{S}}$. This is a drawback of the RK feature map. The computational cost of the RK feature map for an m-order ANOVA kernel is $O(Ddm)$ time. This cost is the same as that of the RM feature map for an m-order polynomial kernel and larger than that for the TS ($O(m(d+D\log D))$). The number of parameters for the proposed method for an m-order ANOVA kernel is $O(Dd)$, which is also larger than that of the TS ($O(md\log D)$) because $m << d < D$ in most cases.

While the random Fourier (RF) feature map, which does not have the order parameter m ($Z_{\mathrm{RF}}(\boldsymbol{x}) = \sqrt{2/D}\cos(\boldsymbol{\Pi}\boldsymbol{x}+\boldsymbol{b})$, where $\boldsymbol{\Pi}\in\mathbb{R}^{D\times d}, \boldsymbol{b}\in\mathbb{R}^d$), also takes $O(Dd)$ time and $O(Dd)$ memory, methods have recently been proposed that take $O(D\log d)$ time and $O(D)$ memory (Feng, Hu, and Liao 2015; Le, Sarlós, and Smola 2013). In this section, we present a faster and more memory efficient RK feature map for the ANOVA kernel based on these recently proposed methods, especially that of Feng, Hu, and Liao, which takes $O(mD\log d)$ time and $O(D)$ memory.

First we explain *signed circulant random feature* (SCRF) (Feng, Hu, and Liao 2015). The $O(Dd)$ time complexity of the RF feature map is caused by the computation of $\boldsymbol{\Pi}\boldsymbol{x}$. The SCRF reduced it to $O(D\log d)$ time without loss of the key property of the RF feature map; approximating the shift-invariant kernel. In the SCRF, without loss of generality, it is assumed that D is divisible by d ($D/d := T$) and that $\boldsymbol{\Pi}$ is replaced by the concatenation of T projection matrices: $\tilde{\boldsymbol{\Pi}} = (\boldsymbol{P}^{(1)};\boldsymbol{P}^{(2)};\cdots;\boldsymbol{P}^{(T)})$. $\boldsymbol{P}^{(t)}\in\mathbb{R}^{d\times d}$, $t\in[T]$, is called a *signed circulant random matrix*, which is a variant of the circulant matrix: $\boldsymbol{P}^{(t)} = \mathrm{diag}(\boldsymbol{\sigma}_t)\mathrm{circ}(\boldsymbol{\omega}_t)$, where $\boldsymbol{\sigma}_t\in\{-1,+1\}^d$ is a Rademacher vector, $\boldsymbol{\omega}_t\in\mathbb{R}^d$ is a random vector generated from an appropriate distribution (e.g., the Gaussian distribution for the radial basis function kernel), and $\mathrm{circ}(\boldsymbol{\omega}_t)\in\mathbb{R}^{d\times d}$ is a circulant matrix in which the first column is $\boldsymbol{\omega}_t$. This formulation clearly reduces the memory required for the RF feature map from $O(Dd)$ to $O(2Td) = O(2D)$. Moreover, the product of $\tilde{\boldsymbol{\Pi}}$ and $\boldsymbol{x}$ surprisingly can be converted into fast Fourier transform, inverse fast Fourier transform, and the element-wise product of vectors which means that time complexity can be reduced from $O(Dd)$ to $O(D\log d)$.

Unfortunately, it is difficult to apply the SCRF technique to the RK feature map because the computation of the itemset kernel does not require the product of a random projection matrix and a feature vector in general. Fortunately, the ANOVA kernel, which is a special case of the itemset kernel, can be computed efficiently (Blondel et al. 2016b) by using recursion:

$$K_{\mathrm{A}}^m(\boldsymbol{\omega},\boldsymbol{x}) = \frac{1}{m}\sum_{t=1}^{m}(-1)^{t+1}K_{\mathrm{A}}^{m-t}(\boldsymbol{\omega},\boldsymbol{x})\langle\boldsymbol{\omega}^{\circ t},\boldsymbol{x}^{\circ t}\rangle, \quad (12)$$

where $\boldsymbol{x}^{\circ p}$ represents the p-times element-wise product of $\boldsymbol{x}$. Hence, the RK feature map for the ANOVA kernel can be written in matrix form:

$$Z_{\mathrm{RK}}(\boldsymbol{x}) = \frac{1}{m\sqrt{D}}\sum_{t=1}^{m}\boldsymbol{a}^{m-t}\circ(\boldsymbol{\Omega}^{\circ t}\boldsymbol{x}^{\circ t}), \quad (13)$$

where $\boldsymbol{\Omega} := (\boldsymbol{\omega}_1^\top;\cdots;\boldsymbol{\omega}_D^\top)\in\mathbb{R}^{D\times d}$ is a matrix in which each row is the random vector of the RK map, and $\boldsymbol{a}^t := (K_{\mathrm{A}}^t(\boldsymbol{\omega}_1,\boldsymbol{x}),\ldots,K_{\mathrm{A}}^t(\boldsymbol{\omega}_D,\boldsymbol{x}))^\top\in\mathbb{R}^D$ is the vector of the t-order ANOVA kernels (clearly, $\boldsymbol{a}^t$ can be regarded as an RK feature of the t-order ANOVA kernel). Although computing $\boldsymbol{\Omega}^{\circ t}$ in Equation (13) seems costly, it is actually trivial when each random vector $\boldsymbol{\omega}_s$ for all $s\in[D]$ is generated from a Rademacher distribution. In this case, $\boldsymbol{\Omega}^{\circ t} = \boldsymbol{\Omega}$ if t is odd; otherwise, it is an all-ones matrix. Therefore, the SCRF technique can be applied to the RK feature map for the ANOVA kernel. Doing this reduces the computational cost of $\boldsymbol{\Omega}^{\circ t}\boldsymbol{x}^{\circ t}$ from $O(Dd)$ to $O(D\log d)$ and thus that of the RK feature map for an m-order ANOVA kernel from $O(mDd)$ time and $O(Dd)$ memory to $O(mD\log d)$ time and $O(D)$ memory. We call a random kernel feature map with the signed circulant random feature a *signed circulant random kernel (SCRK) feature map*.

Although the original SCRF for the RF feature map introduces $\boldsymbol{\sigma}$, resulting in a low variance estimator for the shift-

Method	$D = 2d$	$D = 4d$	$D = 8d$	$D = 16d$
RK (Rademacher)	**6.53e-4 ± 3.86e-5**	**4.62e-4 ± 2.19e-5**	**3.29e-4 ± 1.26e-5**	**2.33e-4 ± 1.02e-5**
RK (Gaussian)	7.31e-4 ± 6.82e-5	5.22e-4 ± 3.71e-5	3.73e-4 ± 1.83e-5	2.62e-4 ± 1.06e-5
RK (Uniform)	6.85e-4 ± 4.96e-5	4.92e-4 ± 2.90e-5	3.50e-4 ± 1.68e-5	2.47e-4 ± 1.05e-5
RK (Laplace)	8.29e-4 ± 1.36e-4	6.16e-4 ± 8.30e-5	4.39e-4 ± 4.03e-5	3.11e-4 ± 2.00e-5
SCRK	7.22e-4 ± 2.13e-4	5.01e-4 ± 9.74e-5	3.60e-4 ± 8.46e-5	2.54e-4 ± 4.34e-5

(a) Second-order ANOVA kernel

Method	$D = 2d$	$D = 4d$	$D = 8d$	$D = 16d$
RK (Rademacher)	**2.26e-5 ± 1.74e-6**	**1.64e-5 ± 8.69e-7**	**1.17e-5 ± 4.70e-7**	**8.35e-6 ± 2.29e-7**
RK (Gaussian)	2.67e-5 ± 3.89e-6	1.97e-5 ± 2.35e-6	1.45e-5 ± 1.17e-6	1.05e-5 ± 6.06e-7
RK (Uniform)	2.40e-5 ± 2.58e-6	1.77e-5 ± 1.46e-6	1.30e-5 ± 8.25e-7	9.27e-6 ± 3.93e-7
RK (Laplace)	3.09e-5 ± 8.56e-6	2.44e-5 ± 5.08e-6	1.80e-5 ± 3.01e-6	1.31e-5 ± 1.54e-6
SCRK	2.29e-5 ± 4.93e-6	1.65e-5 ± 2.28e-6	1.19e-5 ± 1.50e-6	8.40e-6 ± 6.73e-7

(b) Third-order ANOVA kernel

Method	$D = 2d$	$D = 4d$	$D = 8d$	$D = 16d$
RK (Rademacher)	4.24e-2 ± 1.14e-2	2.94e-2 ± **7.07e-3**	2.01e-2 ± **4.79e-3**	1.49e-2 ± 4.94e-3
RK (Gaussian)	4.25e-2 ± 1.23e-2	3.07e-2 ± 8.23e-3	2.12e-2 ± 5.29e-3	1.54e-2 ± 4.79e-3
RK (Uniform)	4.32e-2 ± 1.11e-2	2.96e-2 ± 7.61e-3	**1.99e-2** ± 5.05e-3	**1.45e-2 ± 3.93e-3**
RK (Laplace)	**4.15e-2 ± 1.04e-2**	**2.89e-2** ± 7.34e-3	2.00e-2 ± 5.12e-3	1.49e-2 ± 4.17e-3

(c) All-subsets kernel

Table 1: Absolute errors of RK feature maps for second-order ANOVA kernel, third-order ANOVA kernel, and all-subsets kernel using different distributions for Movielens 100K dataset.

invariant kernel, when order m is even, $\boldsymbol{\sigma}$ is unfortunately meaningless in the proposed RK feature map for the m-order ANOVA kernel case because $K^m_{\mathrm{A}}(-\boldsymbol{\omega}, \boldsymbol{x}) = K^m_{\mathrm{A}}(\boldsymbol{\omega}, \boldsymbol{x})$. Therefore, the SCRK feature map for an even-order ANOVA kernel may not be effective.

5 Relationship between FMs and RK Feature Map for the ANOVA Kernel

The equation for linear models using the RK feature map for the second-order ANOVA kernel $Z_{\mathrm{RK}}(\boldsymbol{x})$ is:

$$f_{\mathrm{LM}}(Z_{\mathrm{RK}}(\boldsymbol{x}); \tilde{\boldsymbol{w}}) = \frac{1}{\sqrt{D}} \sum_{s=1}^{D} \tilde{w}_s K^2_{\mathrm{A}}(\boldsymbol{\omega}_s, \boldsymbol{x}), \qquad (14)$$

where $\tilde{\boldsymbol{w}} \in \mathbb{R}^D$ is the weight vector for the RK feature map $Z_{\mathrm{RK}}(\boldsymbol{x})$. Hence, linear models using the RK feature map can be regarded as FMs with $\boldsymbol{\lambda} = \tilde{\boldsymbol{w}}/\sqrt{D}$ and only one learnable parameter $\boldsymbol{\lambda}$ and without the linear term. Therefore, theoretical results that guarantee the generalization error of linear models using the RK map can be applied to the theoretical analysis of that of FMs. We leave this to future work. The same relationship holds between linear models using the RK feature map for the all-subsets kernel and the all-subsets model. Interestingly, it also holds between linear models using the RM feature map for the polynomial kernel and multi-convex PNs, which are multi-convex formulation models of PNs (Blondel et al. 2016b).

6 Evaluation

We first evaluated the accuracy of our proposed RK feature map on the Movielens 100K dataset (Harper and Konstan 2016), which is a dataset for recommender systems. The age, living area, gender, and occupation of users and the genre and release year of items were used as features in the same way as Blondel et al. (Blondel et al. 2016a). The dimension of the feature vectors was 78.

We calculated the absolute error of the approximation of ANOVA kernels ($m = 2$ or 3) and all-subsets kernel on the training datasets. Each feature vector was normalized by its L^1 norm. Only $10,000$ instances were used. We calculated the mean absolute errors for these instances for 100 trials using Rademacher, Gaussian, Uniform, and Laplace distributions in the RK feature maps and compared the results. For the ANOVA kernels, we also compared them with the SCRK feature map. We varied the dimension of the random features: $2, 4, 8$ and 16 times that of the original feature vectors. We used Scipy (Jones, Oliphant, and Peterson 2001) implementations of FFT and IFFT (scipy.fftpack) in the SCRK and TS feature maps.

As shown in Table 1, the RK feature map with the Rademacher distribution had the lowest absolute error and variance for the second- and third-order ANOVA kernels. In contrast, the differences in the absolute errors between the distributions were small for the all-subsets kernel. The variances were large even for $D = 16d$, so the RK feature map for the all-subsets kernel requires a larger D. For the third-order ANOVA kernel, the performance of the SCRK feature map was as good as that of the RK feature map with the Rademacher distribution. However, for the second-order ANOVA kernel, that of the SCRK feature map was not good. As described above, the SCRK feature map is not efficient when order m is even because $\boldsymbol{\sigma}$ is meaningless.

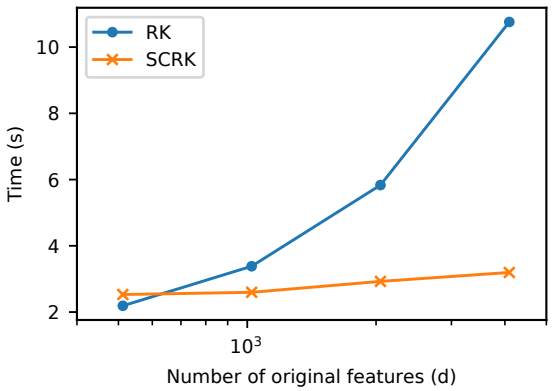


Figure 1: Comparison of mapping times of RK and SCRK feature maps for second-order ANOVA kernel with different dimensions of original feature vector for synthesis dataset (d is shown in log scale).

We next evaluated the effectiveness of the SCRK feature map, which is more time and memory efficient than the RK one w.r.t the dimension of the original feature vector. We created synthesis data with various dimensions of the original features and compared the mapping times of the SCRK and RK feature maps for the second-order ANOVA kernel. We used $\mathcal{N}(0, 1)$ as the distribution of original features and changed the dimension of the original features: $d = 512,\ 1024,\ 2048$ and 4096. We set $D = 8092$ for all d.

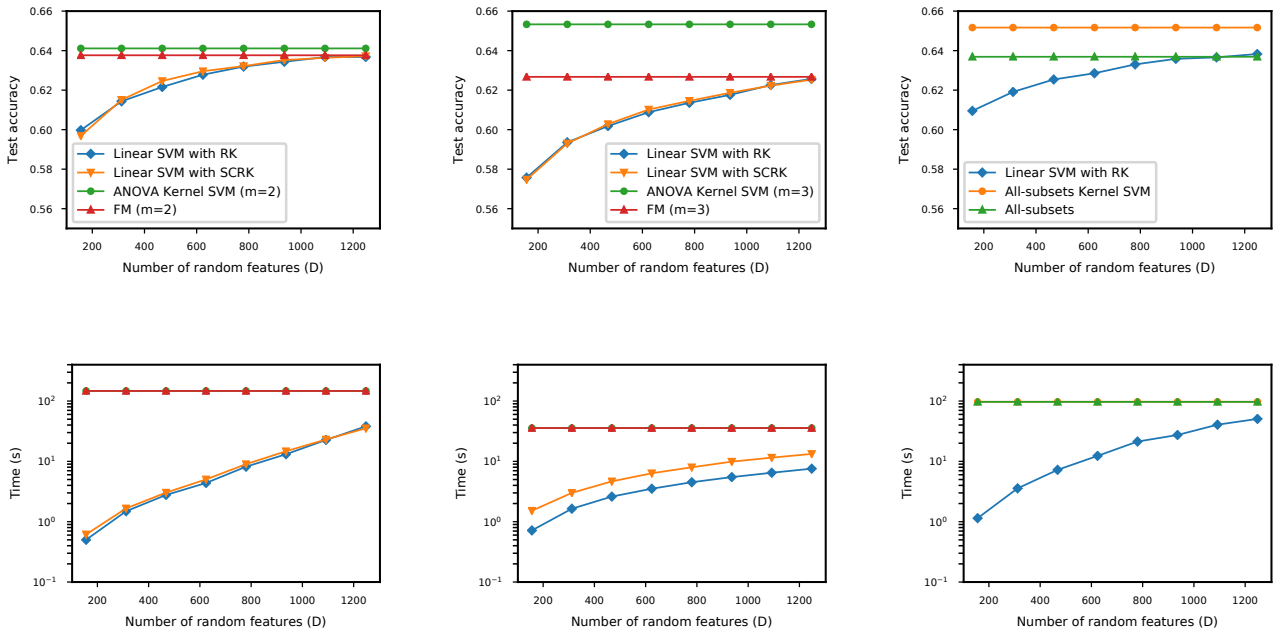
As shown in Figure 1, when the dimension of the original feature vector d was large, the SCRK feature map was more efficient. Although the running time of the RK feature map increased linearly w.r.t d, that of the SCRK feature map increased logarithmically. However, when $d = 512$, the RK feature map was faster than the SCRK feature map. This may be because of the following reasons. First, the difference between d and $\log d$ is small, if d is small. Furthermore, the SCRK feature map requires FFT and IFFT, and hence its dropped constants in Big-O notation are larger than that of the RK feature map.

We next evaluated the performance of linear models using our proposed RK/SCRK feature maps for the Movielens 100K dataset. We converted the recommender system problem to a binary classification problem. We binarized the original ratings (from 1 to 5) by using 5 as a threshold. There were $21,200$, $1,000$, and $20,202$ training, validation, and testing examples. We normalized each feature vector and varied the random features dimension in a manner similar to that used in the first evaluation. We compared the accuracies and learning and testing times for linear SVMs using the proposed RK feature map for the ANOVA/all-subsets kernel, for linear SVMs using the SCRK feature map for the ANOVA kernel, for non-linear SVMs with the ANOVA/all-subsets kernel, and for m-order FMs, and for the all-subsets model. Although there was a linear term in the original FMs, we ignored it because using it or not had little effect on accuracy. All the methods have a regularization hyperparameter, which we set on the basis of the validation accuracy of the non-linear SVMs. For the linear SVMs using random feature maps, we ran ten trials with a different random seed for each trial and calculated the mean of the values. We used a Rademacher distribution for the random vectors. For the FMs and all-subsets model, we also ran ten trials and calculated the mean of the values. We used coordinate descent (Blondel et al. 2016a) as the optimization method. Because this optimization requires many iterations and much time, we ran the optimization process for the same length of time used for the non-linear SVMs. For the rank hyperparameter, we followed Blondel et al. (Blondel et al. 2016a) and set it to 30. We used LinearSVC and SVC in scikit-learn (Pedregosa et al. 2011) as implementations of linear SVMs and non-linear SVMs. LinearSVC used liblinear (Fan et al. 2008) and SVC used libsvm (Chang and Lin 2011). For the implementation of FMs, we used FactorizationMachineClassifier in polylearn (Niculae 2016).

As shown in the Figure 2, when the number of random features $D = 1,248 = 16d$, the accuracies of the linear SVMs using the proposed RK feature map were as good as those of the non-linear SVMs, FMs, and all-subsets model. Furthermore, even though $D = 1,248$, their training and testing times were 2–5 times less than those of the non-linear SVMs, FMs, and all-subsets model. Because the dimension of the original feature vector was small, the running times of the linear SVMs using the SCRK feature map were longer than those of the linear SVMs using the RK feature map when $m = 3$. The accuracies of the linear SVMs using the SCRK feature map were as good as those of the linear SVMs using the RK feature map, and the SCRK feature map required only $O(D \log d)$ time.

We also compared the accuracies and learning and testing times among random-feature-based methods for the polynomial-like kernel: linear SVMs using the proposed RK/SCRK feature map for the ANOVA kernel, TS feature map, and the RM feature map for the polynomial kernel. Similar to the evaluation above, we set the regularization parameter on the basis of the validation accuracy of the non-linear SVMs (we also ran the polynomial kernel SVMs). We again ran ten trials with a different random seed for each trial and calculated the mean of the values.

As shown in Figure 3, when the number of random features D is small, the accuracies of linear SVMs using the TS/RM feature map were better than those of linear SVMs using the RK feature map. However, when the numbers were larger, the accuracies of linear SVMs using the RK feature map were as good as those of linear SVMs using the TS feature map. The linear SVMs using the RM feature map achieved the best performance. However, their running times were clearly longer compared to those of the other methods. Moreover, the RM feature map is not memory efficient: it requires $O(Ddm)$ memory for the m-order polynomial kernel while the proposed RK/SCRK feature map for an m-order ANOVA kernel requires only $O(Dd)$/$O(D)$ memory. The training and testing times of linear SVMs using the RK feature map were the lowest among all methods.



(a) Second-order ANOVA kernel (b) Third-order ANOVA kernel (c) All-subsets kernel

Figure 2: Test accuracies and times for linear SVM using RK feature map approximating (a) second-order ANOVA kernel, (b) third-order ANOVA kernel, and (c) all-subsets kernel and for two existing methods for Movielens 100K dataset. Upper graphs show test accuracies; lower ones show training and test times (time is shown in log scale).

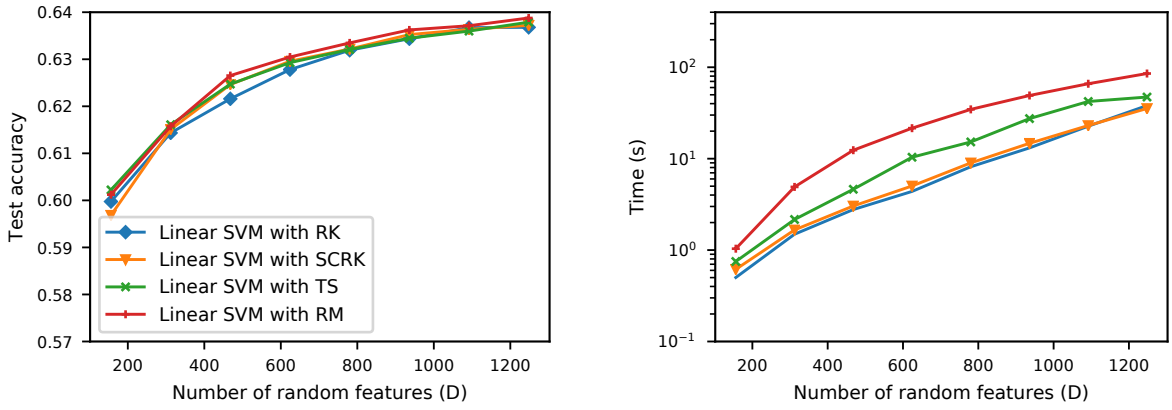


Figure 3: Test accuracies and times for linear SVM using RK/SCRK feature map approximating second-order ANOVA kernel and linear SVM using RM/TS feature map approximating second-order polynomial kernel for Movielens 100K dataset. Left graph shows test accuracies; right one shows training and test times (time is shown in log scale).

We also evaluated the performance of the linear models using the RK/SCRK feature maps and the existing models for the phishing and IJCNN datasets (Mohammad, Thabtah, and McCluskey 2012; Prokhorov 2001). The experimental results were similar to those for the Movielens 100K dataset.

7 Conclusion

We presented a random feature map that approximates the itemset kernel. Although the itemset kernel depends on $\mathcal{S}$, the error bound we presented does not depend on it or the original dimension d. Moreover, we showed that the proposed random kernel feature can be used not only with the

Rademacher distribution but also with other distributions with zero mean and unit variance. Furthermore, we showed that the Rademacher distribution achieves the min-max optimal variance both theoretically and empirically. We also showed how to efficiently compute the random kernel feature map for the ANOVA kernel by using a signed circulant matrix projection technique. Our evaluation showed that linear models using the proposed random kernel feature map are good alternatives to factorization machines and kernel methods for several classification tasks.

Acknowledgement

This work was supported by Global Station for Big Data and Cybersecurity, a project of Global Institution for Collaborative Research and Education at Hokkaido University. Kyohei Atarashi acknowledges support from JSPS KAKENHI Grant Number JP15H05711. Subhransu Maji acknowledges support from NSF via the CAREER grant (IIS 1749833).

References

Blondel, M.; Fujino, A.; Ueda, N.; and Ishihata, M. 2016a. Higher-order factorization machines. In *Advances in Neural Information Processing Systems (NIPS)*, 3351–3359.

Blondel, M.; Ishihata, M.; Fujino, A.; and Ueda, N. 2016b. Polynomial networks and factorization machines: New insights and efficient training algorithms. In *International Conference on Machine Learning (ICML)*.

Chang, C.-C., and Lin, C.-J. 2011. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2(3):27.

Cucker, F., and Smale, S. 2002. On the mathematical foundations of learning. *Bulletin of the American mathematical society* 39(1):1–49.

Fan, R.-E.; Chang, K.-W.; Hsieh, C.-J.; Wang, X.-R.; and Lin, C.-J. 2008. Liblinear: A library for large linear classification. *Journal of Machine Learning Research* 9:1871–1874.

Feng, C.; Hu, Q.; and Liao, S. 2015. Random feature mapping with signed circulant matrix projection. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 3490–3496.

Fukui, A.; Park, D. H.; Yang, D.; Rohrbach, A.; Darrell, T.; and Rohrbach, M. 2016. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Harper, F. M., and Konstan, J. A. 2016. The movielens datasets: history and context. *ACM Transactions on Interactive Intelligent Systems* 5(4):19.

Jones, E.; Oliphant, T.; and Peterson, P. 2001. Scipy: Open source scientific tools for python.

Kar, P., and Karnick, H. 2012. Random feature maps for dot product kernels. In *Artificial Intelligence and Statistics (AISTAS)*, 583–591.

Le, Q.; Sarlós, T.; and Smola, A. 2013. Fastfood-approximating kernel expansions in loglinear time. In *International Conference on Machine Learning (ICML)*.

Lin, T.-Y.; RoyChowdhury, A.; and Maji, S. 2015. Bilinear cnn models for fine-grained visual recognition. In *International Conference on Computer Vision (ICCV)*.

Livni, R.; Shalev-Shwartz, S.; and Shamir, O. 2014. On the computational efficiency of training neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 855–863.

Mohammad, R. M.; Thabtah, F.; and McCluskey, L. 2012. An assessment of features related to phishing websites using an automated technique. In *International Conference for Internet Technology And Secured Transactions (ICITST)*, 492–497.

Niculae, V. 2016. A library for factorization machines and polynomial networks for classification and regression in python. https://github.com/scikit-learn-contrib/polylearn/.

Novikov, A.; Trofimov, M.; and Oseledets, I. 2016. Exponential machines. *International Conference on Learning Representations Workshop*.

Pagh, R. 2013. Compressed matrix multiplication. *ACM Transactions on Computation Theory* 5(3):9.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research* 12:2825–2830.

Pham, N., and Pagh, R. 2013. Fast and scalable polynomial kernels via explicit feature maps. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, 239–247.

Prokhorov, D. 2001. IJCNN 2001 neural network competition. Slide Presentation in IJCNN.

Rahimi, A., and Recht, B. 2008. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems (NIPS)*, 1177–1184.

Rendle, S. 2010. Factorization machines. In *International Conference on Data Mining (ICDM)*, 995–1000.

Rendle, S. 2012. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology* 3(3):57.

Shawe-Taylor, J., and Cristianini, N. 2004. *Kernel methods for pattern analysis*. Cambridge, UK: Cambridge University Press.