Big Data Analytics

RESEARCH Open Access

# SDRNF: generating scalable and discriminative random nonlinear features from data

Haoda Chu[1], Kaizhu Huang[1*], Rui Zhang[1] and Amir Hussian[2]

*Correspondence:
Kaizhu.Huang@xjtlu.edu.cn
[1]Xi'an Jiaotong-Liverpool University, Ren'ai Road, 215123 Suzhou, China
Full list of author information is available at the end of the article

## Abstract

**Background:** Real world data analysis problems often require nonlinear methods to get successful prediction. Kernel methods, e.g. Kernelized Principal Component Analysis, are a common way to get nonlinear properties based on linear representations in a high-dimensional feature space. Unfortunately, traditional kernel methods are unscalable for large-size or even medium-size data. On the other hand, randomized algorithms have been recently proposed to extract nonlinear features in kernel methods. Compared with exact kernel methods, this family of approaches is capable of speeding up the training process dramatically, while maintaining acceptable the classification accuracy. However, these methods fail to engage discriminative features. This significantly limits their classification accuracy.

**Results:** In this paper, we propose a scalable and approximate technique called SDRNF for introducing both nonlinear and discriminative features based on randomized methods. By combining randomized kernel approximation with a couple of generalized eigenvector problems, the proposed approach proves both scalable and accurate for large-scale data.

**Conclusion:** A series of experiments on two benchmark data sets MNIST and CIFAR-10 reveal that our method is fast and scalable, and also generates better classification accuracy over other competitive kernel approximation methods.

**Keywords:** Scalable, Random features, Nonlinear, Discriminative

## Background

Working in linear spaces of function has the benefit of facilitating the construction and analysis of learning algorithms while at the same time allowing large classes of functions [1]. Particularly, in feature selection or dimensionality transformation, there are many famous linear models, for instance, Principal Component Analysis (PCA) [2] and Linear Discriminant Analysis (LDA) [3, 4].

Kernel Principal Component Analysis (KPCA) [5] and Kernel Discriminant Analysis (KDA) [6] are two common methods to enhance the compressed representation of the data. More specifically, they both utilize the kernel trick to map data into a high-dimensional Reproducing Kernel Hilbert Space, where a regular linear PCA and LDA is then performed. However, these two methods are both inefficient and are hard to use in real applications, especially when the data scale is large. Typically, the computational complexity of both KPCA and KDA is of order $O(n^3)$, which is obviously not scalable, when sample number $n$ becomes too large.

To speed up the process of kernel methods, one recent active research focused on using randomized tricks to build scalable kernel approximation [7–10]. In the context of classification, these methods first generate nonlinear feature maps fast and then a linear classifier like Support Vector Machine (SVM) or any large margin linear classifier [11] is used to predict the result. One major shortcoming of this line of methods is that they focus merely on generating nonlinear representation fast and scalably while paying less attention on selecting discriminative features. However, as shown in many research proposals, discriminative features prove highly critical for learning an accurate classifier [12, 13]. Lack in discriminativeness hence limits the system accuracy greatly.

To tackle this problem, we propose in this paper both a scalable and discriminative solution for kernel feature selection methods. More specifically, we first generate multiple random projections based on a sampling probability function, which is dependent on the given kernel matrix. Nonlinear features are derived based on these random projections. A sequence of generalized eigen-problems are then formed to increase the feature separation ability for each pair of different classes. Since our approach can generate Scalable and Discriminative Randomized Nonlinear Features, we name it as SDRNF in short. The proposed SDRNF approach is appealing in many aspects. (1) Its time complexity is $O(m^2n)$. Here $m$ is a very small number, which is usually far less than $n$, the number of data samples. This time complexity is comparable with linear PCA which holds the complexity of $O(d^2n)$ ($d$ is the feature dimensionality). (2) A theoretical bound can be derived to guarantee the excellent approximation between random nonlinear features and the ones implicitly implied by the kernel matrix. (3) A set of discriminative features could be generated for each pair of classes, which will significantly benefit the overall accuracy if used in classification. (4) The proposed framework is simple yet effective, making it very easy to be used in many applications extensively.

The rest of this paper is organized as follows. In the next section, we introduce the random projection method to approximate a kernel matrix. Following that, we describe our model for generating scalable and discriminative nonlinear features. We then show our results on two benchmark large-scale datasets MNIST and CIFAR-10. We discuss some important issues after that. Finally, we set out the concluding remarks.

## Method

### Randomized nonlinear features from kernel matrix

The motivation of randomized methods for kernel-based classification is to map the input data embedded in the kernel matrix to a nonlinear randomized low-dimensional feature space. Then any off-the-shelf fast linear methods can be plugged so that a nonlinear classifier w.r.t. the original data features can be derived [7]. These features should be appropriately designed to guarantee the inner products of the transformed data are approximately equal to those in the feature space of a specified shift-invariant kernel. In this paper, we mainly focus on engaging random Fourier features to approximate a kernel, in particular, the RBF kernel. Some other random features could be also explored [14].

Considering the map $z : R^d \to R^m$, we describe the kernel approximation as follows:

$$k(x, y) = < \phi(x), \phi(y) > \approx z(x)'z(y) \tag{1}$$

Different from the traditional kernel methods, where $\phi$ is usually high-dimensional or even infinite-dimensional (e.g. in in RBF kernel), the mapping given by $z$ is low-dimensional. Thus we can simply regard the data implicitly or explicitly embedded in a nonlinear kernel matrix are nonlinearly transformed to $z$. Since the feature set $z$ is already nonlinear, any fast linear classifier can be applied so as to generate an overall non-linear classifier. Obviously, the nonlinear classifier is given by non-linear features+linear classifier. This is different from original linear features+nonlinear classifiers, but they two could be considered equivalent in the overall viewpoint.

In the context of classification, we are given a training data set containing $n$ samples $x_1, \dots, x_n$ and their corresponding class labels $y_1, \dots, y_n$. Informally, the basic task here is to construct a function $f(x)$ that can best predict its actual label $y$ even when $x$ is a future data point. Many learning algorithms assume the classifier function as a weighted sum of certain simpler functions:

$$f(x) = \sum_{i=1}^{\infty} \alpha_i \phi(x; \theta_i) \tag{2}$$

The parameters of this model are the weights $\alpha$ and the function parameters $\theta$.

The idea behind fandom features is to pick $\theta_i$ randomly in a batch style, and then to solve $\alpha$ exactly via a simple batch convex optimization. Specially, for RBF kernel we randomly sample the parameter $w_i \in R^d$ from a data-independent distribution $p(w)$ and construct an $m$-dimensional randomized feature map $z(X)$ for the input data $X \in R^{n\times m}$ that obeys the following structure:

$$\begin{aligned} & w_1, \dots, w_m \sim p(w) \\ & z_i = \left[cos\left(w_i^T x_1 + b_i\right), \dots ,cos\left(w_i^T x_n + b_i\right)\right] \in R^n \\ & z(X) = [\, z_1 \dots z_m] \in R^{n\times m} \end{aligned} \tag{3}$$

The random Fourier features are constructed by first generating $m$ projections $w_1,\dots,w_m$ from the sampling distribution $p(w)$ that is dependent on the kernel function. Some examples of popular shift-invariant kernels and the corresponding sampling distributions can be seen in Table 1. The process is then projecting each example $x$ to $w_1,\dots,w_m$ separately, and then passing them through cosine functions. The mapping $z_i(x) = cos\left(w_i' x + b_i\right)$ additionally rotates this circle by a random amount $b$ and projects the points onto the interval $[\,0, 1]$. Here $b$ is drawn uniformly from $[\,0, 2\pi]$.

Given these randomized fourier features, we could then learn a linear machine $f(x) = a^T z(x)$, e.g., by solving the following optimization problem involved in the linear SVM:

$$\min_{a\in R^m} \frac{\lambda}{2} ||a||_2^2 + \frac{1}{n} \sum_{i=1}^{n} \ell\left(a^T z(x_i), y_i\right) \ , \tag{4}$$

where $l$ is the loss function. Then we can use this linear machine to approximate the kernel machine.

Although the above randomized process is simple, it is theoretically appealing in that it could guarantee a close approximation for a given kernel matrix.

Intuitively, since both the probability distribution $p(w)$ and the kernel $k(\triangle)$ are real, the integral converges when the complex exponentials are replaced with cosines. We could see this from the following:

$$\begin{aligned} k(x,y) &= \int_{\mathrm{R^d}} \mathrm{p(w)e}^{-jw^T(x-y)}\,\mathrm{d}w \\ &\approx \sum_{i=1}^{m} \frac{1}{m} \mathrm{e}^{-jw_i^T x} \mathrm{e}^{-jw_i^T y} \mathrm{d}w \\ &= \sum_{i=1}^{m} \frac{1}{m} \cos\left(w_i^T x + b_i\right) \cos\left(w_i^T y + b_i\right) \\ &=< \frac{1}{\sqrt{m}} z(x), \frac{1}{\sqrt{m}} z(y) > \end{aligned} \tag{5}$$

Consider:

$$\tilde{K} = \frac{1}{m} z(X) z(X)^T = \frac{1}{m} \sum_{i=1}^{m} z_i z_i^T \tag{6}$$

As consequence of this theorem , the approximate kernel matrix will eventually approximate the true one as the number of random features $m$ tends to infinity. In fact, a formal theoretical bound can be given as follows

$$E||\tilde{K} - K|| \leq \sqrt{\frac{3n^2 \log n}{m} + \frac{2n \log n}{m}}. \tag{7}$$

Detailed proof of the above error bound can be seen in [15].

**Remarks** *Note that the above error bound is very tight. Since the kernel matrix is of size $n \times n$, the average bounded error will be $\sqrt{\frac{3 \log n}{n^2 m} + \frac{2 \log n}{mn}}$. when $n \gg m$, the average value will be be close to zero.*

### Generating discriminative features

In this section, we introduce how to extend the scalable nonlinear features from the previous section to its discriminative version. We mainly engage the famous Niko$'s$ model [16] and manage to find the discriminative features by solving a sequence of generalized eigenproblems. Inspired by the random tricks used in kernel methods, we would find the transformed discriminative features by maximizing the following quotient:

$$\max_{v} R_{ij}(v) = \frac{v^T K_i v}{v^T K_j v} \tag{8}$$

where the $K_i = \frac{1}{m} z(X) z(X)^T$ is the kernel matrix of the *ith* class by random methods. The above objective is trying to maximize the second order information in one class while minimize another class. This problem is actually a generalized eigenproblem and the vector $v$ can be easily obtained. In this paper, we would enumerate all the pairs of classes to form the above quotient. Note that, when the class number becomes large, enumeration of all the possible class pairs may lead to huge computational load. However, this problem may be alleviated by choosing only some pairs based on certain criteria. We will discuss this point later in the next section.

However, the dimension of the kernel matrix is the number of samples in the *ith* class. This leads to different size for each kernel matrix due to the different number of samples in each class. Hence, generalized eigen-problem solutions cannot be applied here. On the other hand, the time complexity is still $O(n^3)$, this would be computationally infeasible.

To solve this problem, we instead use the covariance matrix $C_i = \frac{1}{n} z(X)^T z(X)$ in feature space $R^m$. Although it seems to give away the kernel trick, this is reasonable. Note that the approximate nonlinear mapping function can be easily obtained by random methods in our model. Hence it makes it possible to compute the covariance matrix directly. Then we focus on this quotient:

$$R_{ij}(\nu) = \frac{\nu^T C_i \nu}{\nu^T C_j \nu} \tag{9}$$

Then in this feature space, the complexity of solving eigen-problems becomes $O\left(\alpha m^3\right)$, where $\alpha$ is the number of pairwise classes numbers. In addition, the complexity to calculate the covariance matrix is $O\left(m^2 n\right)$. The overall complexity of our algorithm becomes $O\left(\alpha m^3 + m^2 n\right)$. Since $\alpha$ and $m$ are very small compared with $n$, the complexity of our algorithm is $O\left(m^2 n\right)$.

On the other hand, the computational complexity is $O\left(n^3\right)$ for KPCA, and $O\left(d^2 n\right)$ for PCA. In practice, our method is faster than KPCA when $n$ becomes large. Moreover, PCA and our method are both linear in the sample size $n$, while our method is both nonlinear and discriminative. This presents a great advantage of our method over PCA Our method can also generate much more number of features than traditional KPCA and PCA because we solve many generalized eigen-problems between different classes and each of this problem can give us discriminative features.

## Results and discussion

In this section, we evaluate the proposed method of Scalable and Discriminative Randomized Nonlinear Features (SDRNF) in comparison with other competitive methods, e.g., the famous approach Random Kitchen Sinks (RKS) and Nikos Generalized Eigenvectors for Multi-class (GEM). The two benchmark large-scale data sets used are MNIST and CIFRA-10, which are widely used in the community. Note that, bosth MNIST AND CIFAR data contain a separate training and test data set. Hence no cross-validation is needed to report the average result. We will mainly generate our SDRNF features from RBF kernel functions. However, it should be noted that it is easy and straightforward to generate similar features from different kernels by choosing different sampling distributions.

In our experiments, we will first use different methods to generate features. A linear classifier will then be trained based on these features. We first investigate the classification performance when the linear SVM is exploited as the classifier. We then report the accuracy when a recent popular linear model called CLS [17] to further validate the effectiveness of the proposed approach. All the parameters involved in the experiments were tuned via cross validation. These parameters include the trade-off constant used in the linear SVM.

### Results

We first report the experimental results on MNIST data in Table 2 when the linear SVM is used as the final classifier. We intentionally selected different number of approximated

**Table 1** Examples of popular shift-invariant kernels and the corresponding sampling distributions

| Kernel name | $k(\triangle)$ | $p(w)$ |
|---|---|---|
| Gaussian | $e^{\left(-\frac{\|\triangle\|_2^2}{2}\right)}$ | $(2\pi)^{-\frac{D}{2}} e^{\left(-\frac{\|w\|_2^2}{2}\right)}$ |
| Laplacian | $e^{(-\|\triangle\|_1)}$ | $\prod_d \frac{1}{\pi(1+w_d^2)}$ |

**Table 2** Error rate (%) given by linear SVM on MNIST data

| Methods | 100 | 200 | 300 | 400 | 500 | 1000 | 1500 | 2000 | 2500 | 3000 |
|---|---|---|---|---|---|---|---|---|---|---|
| RKS | 13.64 | 8.55 | 6.58 | 5.96 | 4.79 | 3.6 | 3.48 | 3.10 | 2.82 | 2.27 |
| SDRNF | 5.35 | 3.13 | 2.23 | 2.22 | 2.08 | 1.62 | 1.61 | 1.63 | 1.59 | 1.55 |

The first row indicates different number of approximate features from the kernel matrix. The proposed SDRNF clearly outperforms the other randomized method RKS

features so as to perform a comprehensive investigation between different algorithms. In this experiment, we mainly compared our proposed SDRNF with another popular randomized method called RKS. As seen in Table 2, our model consistently outperformed RKS in all the cases of different features. The highest accuracy of our model in MNIST is 98.45 %, it beats the RKS method which holds 97.73 % accuracy. This is reasonable since RKS method fails to extract discriminative information while our method can appropriately select discriminative features.

The situation is also similar on CIFAR data, which is shown in Table 3. In this table, we also report the error rates when different number of features were generated by SDRNF and RKS. Also the classifier adopted is the linear SVM. The results again demonstrated the effectiveness by engaging discriminative features. To clearly visualize the difference between the proposed SDRNF method and RKS, we plot the error rate curves in Fig. 1. It can be even clearly seen that SDRNF demonstrated a clear distinction over RKS.

After carefully examining the performance of SDRNF and RKS by using linea SVM in terms of different feature numbers, we also report in Table 4 the performance of more competitive models when a recent promising linear model called CLS is used as the classification model. In particular, we compared our proposed approach with the famous GEM, RKS, and PCA. For simplicity, we only report the best results achieved by different algorithms. Some interesting points are highlighted in the following. First, we note that SDRNF, GEM, and RKS significantly outperformed PCA, since all these three algorithms could generate non-linear features, while PCA is merely a linear feature selection method. Secondly, SDRNF ranks the second on MNIST (just slightly lower than GEM) while it is significantly better than the remaining methods on CIFAR. This shows that incorporation of discriminative learning into ramdomized non-linear feature selection is indeed useful. Finally, we should note that the time complexity of SDRNF, GEM, and RKS are basically in the same order. Hence we do not report the computational time in the experiments.

## Discussion

We discuss some important issues in this section. First, as mentioned in the above, our model can generate much more features than KPCA and KDA. The experimental results on the two data sets indicates the effectiveness of our feature. However, when features

**Table 3** Error rate (%) given by linear SVM on CIFAR data

| Methods | 100 | 200 | 300 | 400 | 500 | 1000 | 1500 | 2000 | 2500 | 3000 |
|---|---|---|---|---|---|---|---|---|---|---|
| RKS | 64.15 | 60.45 | 58.15 | 57.31 | 56.31 | 53.79 | 52.27 | 52.21 | 52.61 | 51.75 |
| SDRNF | 58.94 | 52.44 | 51.15 | 49.29 | 49.11 | 45.94 | 44.02 | 45.28 | 45.53 | 47.19 |

The first row indicates different number of approximate features from the kernel matrix. The proposed SDRNF clearly outperforms the other randomized method RKS

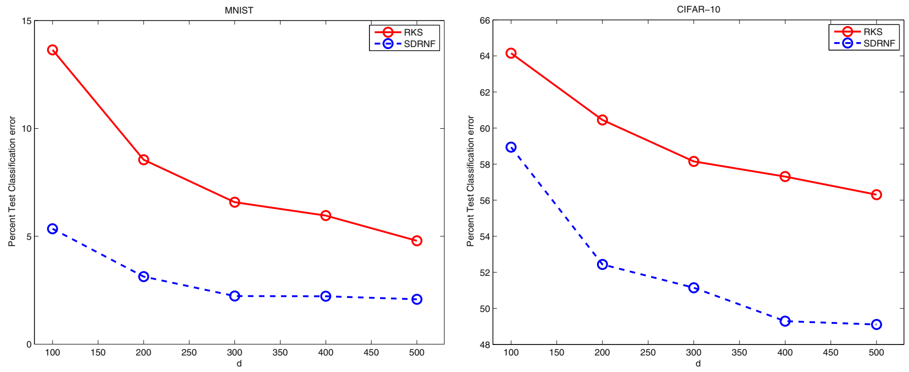

**Fig. 1** Test classification error rate comparison on MNIST and CIFAR. Linear SVM was adopted as the classifier

becomes too large, even linear machine will be very slow. The situation is more serious, especially when the number of classes becomes very large, for instance when $k$ is more than 100. Hence, we should try to remove redundant features while keep the most discriminative ones. The speedup might be speed up by choosing only a subset of class pairs. However, this process might not be easy because it is hard to distinguish whether some class pairs should be removed, even though some heuristics may be available for doing so. One possible solution is to use parallel methods, since discriminative features can be generated independently. We will explore this in the future.

Second, although random methods provide us a fast way to generate nonlinear feature, it often leads to dense feature representation. Even though the original feature is sparse, random method will still make the nonlinear feature dense. In this case, it will then incur unnecessary computational cost. Note that, when the feature is denser, even a linear classifier takes more time for classifying patterns. Hence, a sparse random method may be helpful, which will further speed up the system speed. Recently, some work has already been done in this direction [18]. We will explore this property in order to make our model more powerful.

## Conclusion

The main objective in this paper is to investigate scalable methods to extract discriminative and nonlinear features. To this end, we have proposed a scalable and approximate technique called SDRNF for introducing both nonlinear and discriminative features based on randomized methods. By combining randomized kernel approximation with a couple of generalized eigenvector problems, the proposed approach proves both scalable and accurate for large-scale data. We have done a series of experiments on the benchmark datasets MNIST and CIFAR-10. Experimental results showed that our method is fast and scalable, and works remarkably better than other competitive methods. Due to its scalable and discriminative properties, we believe our model can be used in a variety areas in machine learning.

**Table 4** The lowest classification error rates achieved by different algorithms on the two data sets. All the methods used the promising CLS [17] linear classifier

| | SDRNF | GEM | RKS | PCA |
|---|---|---|---|---|
| MNIST | 1.55 | 1.12 | 2.27 | 9.34 |
| CIFAR-10 | 45.02 | 59.29 | 51.75 | 60.30 |

**Acknowledgement**
The research was partly supported by the National Basic Research Program of China (2012CB316301), National Science Foundation of China (61473236), and Jiangsu University Natural Science Research Programme (14KJB520037).

**Authors' contributions**
HC and KH conceived the project. HC and KH proposed the method and drafted the manuscript. HC implemented the algorithm. RZ and AH joined the project and participated in the design of the study. All authors read, improved, and approved the manuscript.

**Competing interests**
The authors declare that they have no competing interests.

**Author details**
[1]Xi'an Jiaotong-Liverpool University, Ren'ai Road, 215123 Suzhou, China. [2]Division of Computing Science & Maths, University of Stirling, Stirling FK9 4LA, UK, Stirling, UK.

Received: 12 November 2015 Accepted: 21 September 2016
Published online: 28 September 2016

**References**

1. Hofmann T, Schölkopf B, Smola AJ. Kernel methods in machine learning. Ann Stat. 2008;36(3):1171–1220.
2. Pearson K. Liii. on lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science. 1901;2(11):559–72.
3. Fukunaga K. Introduction to Statistical Pattern Recognition, 2nd. San Diego: Academic Press; 1990.
4. Xu B, Huang K, Liu CL. Maxi-min discriminant analysis via online learning. Neural Netw. 2012;34:56–64.
5. Schölkopf B, Smola A, Müller KR. Kernel principal component analysis. In: Proceedings of 7th International Conference on Artificial Neural Networks. Springer; 1997. p. 583–8. October 8–10, ISBN 3-540-40408-2.
6. Scholkopft B, Mullert KR. Fisher discriminant analysis with kernels In: Hu YH, Larsen J, Wilson E, Douglas S, editors. Neural networks for signal processing IX. 1st edition. IEEE; 1999. ISBN-10: 078035673X.
7. Rahimi A, Recht B. Random features for large-scale kernel machines. In: Advances in Neural Information Processing Systems. Cambridge: The MIT Press; 2007. p. 1177–1184.
8. Rahimi A, Recht B. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In: Advances in Neural Information Processing Systems. Cambridge: The MIT Press; 2009. p. 1313–1320.
9. Hamid R, Xiao Y, Gittens A, DeCoste D. Compact random feature maps. In: Proceedings of the 31th International Conference on Machine Learning. Cambridge: The MIT Press; 2014.
10. Le Q, Sarlós, Tamás. Fastfood–approximating kernel expansions in loglinear time. In: Proceedings of the 30th International Conference on Machine Learning. Cambridge: The MIT Press; 2013.
11. Huang K, Yang H, King I, Lyu MR. Machine Learning: Modeling Data Locally and Gloablly. Berlin: Springer; 2008.
12. Jebara T. Machine Learning: Discriminative and Generative: Springer US; 2003. ISBN 1-4020-7647-9.
13. Huang K, King I, Lyu MR. Discriminative training of bayesian chow-liu tree multinet classifiers. In: Proceedings of International Joint Conference on Neural Network (IJCNN-2003), Oregon, Portland, U.S.A.. The IEEE Press; 2003. p. 484–8.
14. Yang T, Li YF, Mahdavi M, Jin R, Zhou ZH. Nyström method vs random fourier features: A theoretical and empirical comparison. In: Advances in Neural Information Processing Systems. Cambridge: The MIT Press; 2012. p. 476–84.
15. Lopez-Paz D, Sra S, Smola A, Ghahramani Z, Schölkopf B. Randomized nonlinear component analysis. In: Proceedings of the 31th International Conference on Machine Learning. Cambridge: The MIT Press; 2014.
16. Karampatziakis N, Mineiro P. Discriminative features via generalized eigenvectors. In: Proceedings of the 31th International Conference on Machine Learning. Cambridge: The MIT Press; 2014.
17. Agarwal A, Kakade SM, 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. Cambridge: The MIT Press; 2014.
18. Huang K, Zheng D, Sun J, Hotta Y, Fujimoto K, Naoi S. Sparse learning for support vector classification. Pattern Recogn Lett. 2010;31(13):1944–51.