An Ensemble of Kernel Ridge Regression for Multi-class Classification

Suganthan, Ponnuthurai Nagaratnam; Rakesh, Katuwal

2017

Rakesh, K., & Suganthan, P. N. (2017). An Ensemble of Kernel Ridge Regression for Multi-class Classification. Procedia Computer Science, 108, 375-383.

https://hdl.handle.net/10356/88058

https://doi.org/10.1016/j.procs.2017.05.109

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 108C (2017) 375–383

Procedia Computer Science

International Conference on Computational Science, ICCS 2017, 12-14 June 2017, Zurich, Switzerland

An Ensemble of Kernel Ridge Regression for Multi-class Classification

Katuwal Rakesh and P.N. Suganthan*

Nanyang Technological University, Singapore
rakeshku001@e.ntu.edu.sg
*epnsugan@ntu.edu.sg

Abstract
We propose an ensemble of kernel ridge regression based classifiers in this paper. Kernel ridge regression admits a closed form solution making it faster to compute and also making it suitable to use for ensemble methods for small and medium sized data sets. Our method uses random vector functional link network to generate training samples for kernel ridge regression classifiers. Several kernel ridge regression classifiers are constructed from different training subsets in each base classifier. The partitioning of the training samples into different subsets leads to a reduction in computational complexity when calculating matrix inverse compared with the standard approach of using all N samples for kernel matrix inversion. The proposed method is evaluated using well known multi-class UCI data sets. Experimental results show the proposed ensemble method outperforms the single kernel ridge regression classifier and its bagging version.

Keywords: kernel ridge regression, multi-class classification, random vector functional link network

1 Introduction

"Many heads are better than one." "Many are smarter than the Few." An avalanche of such dicta, accentuating the power and wisdom of crowds or experts, are very popular in our society. Several studies based on sociological, psychological and other aspects of human life concord these statements [19, 16]. Inspired by the power of crowds, such a committee have been used not only on the decision making process of humans but also in machine related tasks. Machine Learning is such a field where such committees or ensembles are widely popular and consequential.

In machine learning, ensemble methods employ multiple models to achieve improved performance compared to a single model. Extensive researches, both theoretical and empirical, advocate the advantages of ensemble methods. Bias-variance decomposition [10], margin-theory [18] and strength-correlation [3], all explain the theory behind the superior performance of ensemble methods.

Bagging [2], Boosting [8], and Random Forest [3] are the most popular ensemble techniques [15]. These models mainly use decision trees as base classifiers. There is a myriad of research on ensemble methods with decision trees in the literature [24, 22, 21]. However, an ensemble can be created with any base classifier provided its performance is slightly better than random guessing [20, 15].

Kernel Ridge Regression (KRR) and Support Vector Machine (SVM) are the best known members using kernel methods. When there is non-linear structure in the data, kernel based methods are quite useful [6]. KRR is simpler and faster to train with its closed form solution, and can achieve performance comparable to sophisticated methods such as SVM. In the literature, however, there is limited work on KRR ensemble. Thus, in this paper, we propose an ensemble of classifiers based on kernel ridge regression and random vector functional link (RVFL) network.

One difficulty in applying KRR technique is the inversion of kernel matrix, which requires $\mathcal{O}(N^3)$ time and $\mathcal{O}(N^2)$ memory. Such scaling is prohibitive when N, the number of data samples, is large. Although time may not be a constraint except in case of large data sets, the application of KRR can be halted by out-of-memory failures. Thus, to reduce the computational complexity of KRR, we use RVFL to partition the data samples into several subsets. This helps to avoid the out-of-memory failures to an extent. This is because, the matrix inverse is computed for $n(< N)$ training samples where, n is the number of samples in each subset. Our proposed method thus, befits naturally to parallel and distributed computation.

KRR can be generally regarded as a strong classifier. However, we create an unstable KRR classifier, appropriate for ensemble methods, by training it with only a subset of the whole training data set generated using RVFL. For the purpose of brevity, we name our method as ERK: an ensemble of RVFL with KRR.

In ERK, multiple KRR classifiers are constructed in each base classifier (a base classifier comprises of RVFL and m KRR classifiers where, m is the number of classes) with training samples from each subset. Each subset is associated with a class and consists of training samples with all the samples from its true class and only a fraction of samples from other classes (different subsets will have different fractions of samples from other classes). Thus, the training sets for each m KRR classifiers are different. Our approach fits perfectly to multi-class classification however, it is unsuitable for data sets with binary classes. This is because training samples for two KRR classifiers will be the same resulting in two identical KRR classifiers in a base classifier.

The remainder of this paper is organized as follows. In Sections 2 and 3, we provide a background on kernel ridge regression and random vector functional link network. Our proposed ensemble method, ERK, is elucidated in Section 4. We present our experimental setup along with the performance of ERK compared with other classifiers in Section 5. Lastly, in Section 6, we give our conclusion and discuss future works.

2 Kernel Ridge Regression

Suppose we have a training set $((x_1, y_1), \ldots, (x_N, y_N))$, where N is the total number of training samples. X is a features matrix, $[x_1, x_2, \ldots, x_N]$, of size $N \times d$ and $Y = [1, 2, \ldots, m]$ is a $N \times 1$ vector of class labels.

Kernel ridge regression is based on Ordinary Least Squares (OLS) and ridge regression [17]. The OLS minimizes the squared loss:

$$\min_{\beta} \|Y - X\beta\|^2 \tag{1}$$

where $\|.\|$ denotes the L_2 norm. A shrinkage or ridge parameter λ is added to the above expression to control the trade-off between the bias and variance of the estimate that leads to the following problem:

$$\min_{\beta} \|Y - X\beta\|^2 + \lambda\|\beta\|^2 \tag{2}$$

The closed form solution for the above problem is $\beta = (X^T X + \lambda I)^{-1} X^T Y$ where, I is an identity matrix. The predicted label of a new unlabeled example x is given by $\beta^T x$.

Kernel ridge regression extends linear regression into non-linear and high-dimensional (or even infinite dimensional) space using the "kernel trick". The data x_i in X is replaced with the feature vectors: $x_i \to \Phi = \Phi(x_i)$ induced by the kernel where $K_{ij} = k(x_i, x_j) = \Phi(x_i)\Phi(x_j)$. Thus, the predicted class label of a new example x is now given by:

$$Y^T (K + \lambda I)^{-1} k \tag{3}$$

where $k = (k_1, \ldots, k_N)^T$, $k_n = x_n \cdot x$ and $n = 1, \ldots, N$.

In Kernel ridge regression, one can simply employ the commonly used kernel functions such as Gaussian or linear or polynomial without accessing the feature vectors $\Phi(x)$.

Since KRR is naturally poised for handling regression problems, we extend its use for classification by transforming classification problems as regression problems with class labels. To achieve this, we define the output target Y with 0-1 coding [1]. More specifically, if there are m classes with N samples, the output Y is a $N * m$ matrix which can be generated by the following equation:

$$Y_{ij} = \begin{cases} 1 & \text{if } i^{th} \text{ sample belongs to } j^{th}\text{class} \\ 0 & \text{otherwise} \end{cases}$$

3 Random Vector Functional Link Network

Random Vector Functional Link Network is a randomized variant of Functional Link Neural Network (FLNN) [13, 4]. It is mainly characterized by the absence of backpropagation (BP) and the presence of direct links between the input and output nodes. The weights between the input and hidden neurons in RVFL are randomly generated from a suitable range. The direct links in RVFL regularize the network from the effects of randomization leading to a simpler model with small number of hidden neurons while increasing the prediction accuracy [23]. Using signal processing dialects, the direct links can be interpreted as the delay lines of finite impulse response (FIR) filter [14]. The output layer of RVFL consists of nodes corresponding to the number of classes, with each node assigning a score for each class. The predicted class for a sample x is the class represented by a node with $\arg\max(s_i(x))$, $i \in \{1, \ldots, m\}$ where, s is the score given by each output node i. Since only output weights need to be computed with a closed form solution available, RVFL is faster to train and test [14, 23].

4 Construction of ERK

Bagging is a popular method of ensemble generation. In Bagging, multiple versions (bags) $(D_1, D_2, \ldots, D_L)$ of original data set D is constructed by sampling with replacement, and a classifier is constructed for each D_j. The same bagging principle is employed here. However, in ERK, each bag (D_j) is divided into m subsets, equivalent to the number of classes in each data set, and an independent kernel ridge regression classifier is computed for each subset. In other words, instead of learning a single KRR classifier on all the training samples of a bag, we divide the training samples of the bag into m subsets and compute a local KRR classifier for each subset. A visual representation of this idea is shown in Fig. 1 for $m = 4$.

In the training phase, each sample x_i is passed to RVFL. As mentioned above, the output nodes of the RVFL bear a score for each class. In general classification by RVFL, the predicted class is the class represented by the node with the maximum score. But, instead of classification by RVFL, it is employed in our method to partition the training samples into m subsets. The sample x_i is used as the training sample for the classes with the highest and the second-highest scores. Here, by classes we imply the subset associated with each class. In this way, a training set is created for each subset and a KRR classifier is created. The final model is an ensemble of such base classifiers. In cases, where the true class of x_i is neither the highest nor the second-highest class, the training sample is still placed in its true class subset. The algorithm for the construction of ERK is presented in Table 1.

Table 1: ERK Construction

```
Algorithm:
Training:
For $j = 1, \dots, L$
    1. For $i = 1, \dots, N$
        (a) Pass the training data $x_i$ to the RVFL network.
        (b) Select the classes with the highest and the second highest scores and store $x_i$ as
            the training data for the two subsets associated with those two classes.
    2. Construct KRR classifier from each subset.
Testing:
The test data is first passed to RVFL. Similar to the training phase, two classes with the
highest and the second highest scores are selected based on the output of RVFL. The test data
is pushed down to the KRR classifiers belonging to those two classes. The final prediction
is based on the majority votes of the KRR classifiers.
```

5 Performance Evaluation

5.1 Experimental Setup and Data set specification

To verify the effectiveness of ERK, 10 multi-class UCI data sets [11] are used in the experiment. Table 2 gives an overview of the data sets used in this paper. The experimental setup and data partitions for training, parameter tuning and testing is based on [7]. Randomized stratified sampling is employed to generate a training and a test set (each with 50% of the available patterns). These sets are used for parameter tuning. Then, using the tuned parameters a 4-fold cross validation is developed using the whole data. The test result is the average over the 4 test sets. For those data sets (optical, st-landstat), with already available training and testing data, the classifiers (with the tuned parameters) are trained and tested on the respective sets. The feature vectors of the data sets are normalized by removing the mean and dividing by its variance. The ensemble size, L, for the experiment is fixed at 500.

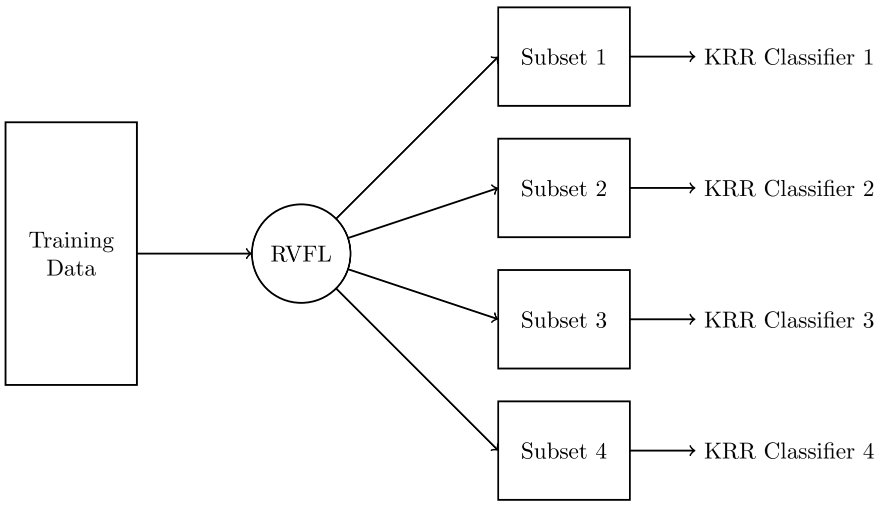

Figure 1: A base classifier of ERK

5.2 Parameter Settings

To foster diversity amongst the classifiers of the ensemble, RVFL is randomized in each base classifier. The parameter settings for RVFL is based on [23, 7] where each RVFL randomly picks the activation function and network parameters from the parameter settings listed below:

1. Number of hidden neurons, N = 3:203
2. λ (=$(1/2)^C$) in ridge regression, C = -5:14
3. Activation Functions: radbas, sine and tribas
4. Range of the randomization for weights [-S,+S] and bias [0,S], where $S = 2^t$ with t = -1.5:0.5:1.5

The RVFL has direct link from input layer to output layer with bias term in the output neuron. All the data sets in our experiment use the Gaussian radial basis kernel

$$K(x_1, x_2) = \exp(-\gamma \|x_1 - x_2\|_2^2)$$

The kernel parameter γ and ridge parameter λ for KRR are tuned for each data set. γ is set to 2^m where m = -15:3 and λ is set to 2^n where n = -5:14.

5.3 Experimental Results

In this section, the performance of ERK is compared with three other classifiers. The two classifiers are single KRR, denoted by KRR(S), and its bagging version, denoted by KRR(B). These two classifiers are tuned with the same set of parameters as used in ERK. The Bagged KRR, KRR(B), is trained with the same 500 bags and same Gaussian kernel in each base classifier as done in ERK to avoid any bias. The other classifier is RVFL. The performance of RVFL on these data sets are extracted from [23] as it uses the same data partitions as ours. In the first phase, we compare the performance of ERK with KRR(S) and RVFL. In the second phase, we discuss the performance of two ensemble methods, ERK and KRR(B).

Table 2: Overview of the Data sets

Datasets	Patterns	Features	Classes
Molec-biol-splice	3190	60	3
Optical	3823	62	10
St-image	2310	18	7
St-landstat	4435	36	6
St-vehicle	846	18	4
Steel plates	1941	27	7
Thyroid	3772	21	3
Wall-following	5456	24	4
W-qua-white	4898	11	7
Yeast	1484	8	10

Table 3: Classification accuracies of different methods

Datasets	KRR(S)	KRR(B)	ERK	RVFL
Molec-biol-splice	79.49	78.95	**86.42**	83.97
Optical	96.93	96.99	**98.07**	97.22
St-image	95.53	95.62	**97.05**	94.84
St-landstat	90.15	90.25	**91.34**	86.40
St-vehicle	71.56	71.45	**83.18**	82.58
Steel plates	**77.01**	76.60	**77.01**	76.60
Thyroid	94.46	94.25	**96.41**	93.96
Wall-following	87.51	87.68	**92.14**	86.62
W-qua-white	65.62	65.80	**66.16**	56.17
Yeast	59.56	59.50	60.44	**61.12**

Bold values represent the highest accuracy.

Friedman test is used, as suggested in [5], to perform statistical comparison of the classifiers. It uses the rank of each classifier in each data set to reflect the overall performance of the classifier. In this approach, the classifiers are ranked for each data set separately, with the best performing algorithm getting the rank of 1, the second best rank 2..., average ranks are assigned in case of ties. The overall average rank of each classifier based on the 10 ranks in each data set is summarized in Table 4.

Let N and k denote the number of data sets and classifiers respectively. The Friedman test

Table 4: Average rank values based on classification accuracies of each method. Lower rank reflects better performance.

	KRR(S)	KRR(B)	ERK	RVFL
Rank	2.85	2.95	1.15	3.05

compares the average ranks of the classifiers, $R_j = \sum_i r_i^j$ where, r_i^j is the rank of the j^{th} of the k classifier on the i^{th} of N data set. The null hypothesis states that the performance of all the classifiers are similar and so their ranks R_j should be equal. When N and k are large enough, the Friedman statistic

$$\chi_F^2 = \frac{12N}{k(k+1)} \left[\sum_j R_j^2 - \frac{k(k+1)^2}{4} \right], \tag{4}$$

is distributed according to χ_F^2 with k-1 degrees of freedom under the null hypothesis. However, in this case, χ_F^2 is undesirably conservative. A better statistics is given by

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1) - \chi_F^2}, \tag{5}$$

which is distributed according to F-distribution with k-1 and (k-1)(N-1) degrees of freedom. If the null-hypothesis is rejected, the Nemenyi post-hoc test [12] can be used to check whether the performance of two among k classifiers is significantly different. The performance of two classifiers is significantly different if the corresponding average ranks of the classifiers differ by at least the critical difference (CD)

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}}, \tag{6}$$

where critical values q_α are based on the Studentized range statistic divided by $\sqrt{2}$. α is the significance level and is equal to 0.05 in this paper.

By simple calculations we obtain, $\chi_F^2 = 14.7$ and $F_F = 8.65$. With 4 classifiers and 10 data sets, F_F is distributed according to the F-distribution with $4 - 1 = 3$ and $(4-1)(10-1) = 27$ degrees of freedom. The critical value for $F_{(3,27)}$ for $\alpha = 0.05$ is 2.96, so we reject the null-hypothesis. Based on the Nemenyi test, the critical difference is CD $= q_\alpha \sqrt{(k(k+1))/(6N)} = 2.569 * \sqrt{4 * 5/(6 * 10)} \simeq 1.48$.

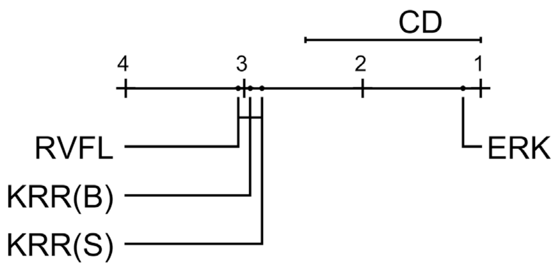

Figure 2: Comparison of classifiers against each other with the Nemenyi test. Groups of classifiers that are not significantly different (at $\alpha = 0.05$) are connected.

From Figure 2, we can see that our proposed method ERK is significantly better than KRR(S) and RVFL. ERK either performs equally or outperforms the other two classifiers in every data set except one, the Yeast data set. Although, KRR(S) is slighly better than RVFL, there is no significant difference between the two classifiers. The mean accuracies of KRR(S), KRR(B), ERK and RVFL in these data sets are 81.78, 81.7, 84.82 and 81.94 respectively. From Table 3, it is worth noticing that ERK consistently performs better than KRR(S) and RVFL even if one among KRR or RVFL is performing worse. This is in congruent with the generally accepted notion that ensemble methods perform better than a single classifier.

5.3.1 Comparison with Bagged KRR

In all the data sets above, ERK has outperformed KRR(B). We also notice that KRR(B) is either only slightly better or worse in most data sets when compared with KRR(S). The bad performance of bagged KRR can be explained by the bias-variance decomposition of error. Generally, bagging improves the performance of unstable methods, such as decision trees and neural network, by lowering the variance with slight increase in bias. However, for stable methods with low bias and low variance, such as KRR, bagging can degrade the performance as stated by Breiman in [2]. However, in ERK, each KRR is constructed with only a subset of the training data which makes it unstable and hence, performance is improved by using an ensemble of several unstable KRR classifiers.

6 Conclusion and Future Work

In this paper, we proposed an ensemble of kernel ridge regression classifiers, ERK, for multi-class classification. Kernel ridge regression has an elegant closed form solution, and can easily be extended to perform classification. Several KRR classifiers were constructed in each base classifier of the ensemble from the subset of samples partitioned using random vector functional link network. In our method, matrix inverse is computed on reduced number of samples than original sample size, thus out-of-memory issues during matrix inversion can be circumvented to an extent. Experimental results show that our method consistently performs better than single and bagged KRR and RVFL.

In ERK, we used a single Gaussian kernel in each base classifier. But, as suggested in [9], it can be useful to use multiple kernels instead of a single kernel. One possible future direction can be to extend our method using multiple kernels and introduce further diversity amongst the base classifiers. We also plan to use our method in all multi-class UCI data sets and compare its performance with other ensemble methods.

References

[1] Senjian An, Wanquan Liu, and Svetha Venkatesh. Face recognition using kernel ridge regression. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–7. IEEE, 2007.

[2] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[3] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[4] Satchidananda Dehuri and Sung-Bae Cho. A comprehensive survey on functional link neural networks and an adaptive pso–bp learning for cflnn. *Neural Computing and Applications*, 19(2):187–205, 2010.

[5] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30, 2006.

[6] Petros Drineas and Michael W Mahoney. On the nyström method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6(Dec):2153–2175, 2005.

[7] Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. Do we need hundreds of classifiers to solve real world classification problems. *J. Mach. Learn. Res*, 15(1):3133–3181, 2014.

[8] Yoav Freund. Boosting a weak learning algorithm by majority. In *COLT*, volume 90, pages 202–216, 1990.

[9] Mehmet Gönen and Ethem Alpaydın. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12(Jul):2211–2268, 2011.

[10] Ron Kohavi, David H Wolpert, et al. Bias plus variance decomposition for zero-one loss functions. In *ICML*, volume 96, pages 275–83, 1996.

[11] M. Lichman. UCI machine learning repository, 2013.

[12] Peter Nemenyi. Distribution-free multiple comparisons. In *Biometrics*, volume 18, page 263. International Biometric SOC 1441 I ST, NW, SUITE 700 Washington, DC, 20005-2210, 1962.

[13] Yoh-Han Pao, Stephen M Phillips, and Dejan J Sobajic. Neural-net computing and the intelligent control of systems. *International Journal of Control*, 56(2):263–289, 1992.

[14] Ye Ren, PN Suganthan, N Srikanth, and Gehan Amaratunga. Random vector functional link network for short-term electricity load demand forecasting. *Information Sciences*, 2016.

[15] Lior Rokach. Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1-2):1–39, 2010.

[16] Louis B Rosenberg. Human swarms, a real-time paradigm for collective intelligence. *Collective Intelligence*, 2015.

[17] Craig Saunders, Alexander Gammerman, and Volodya Vovk. Ridge regression learning algorithm in dual variables. In *(ICML-1998) Proceedings of the 15th International Conference on Machine Learning*, pages 515–521. Morgan Kaufmann, 1998.

[18] Robert E Schapire, Yoav Freund, Peter Bartlett, Wee Sun Lee, et al. Boosting the margin: A new explanation for the effectiveness of voting methods. *The annals of statistics*, 26(5):1651–1686, 1998.

[19] James Surowiecki. *The wisdom of crowds.* Anchor, 2005.

[20] Giorgio Valentini and Francesco Masulli. Ensembles of learning machines. In *Italian Workshop on Neural Nets*, pages 3–20. Springer, 2002.

[21] Le Zhang, Ye Ren, and Ponnuthurai N Suganthan. Instance based random forest with rotated feature space. In *Computational Intelligence and Ensemble Learning (CIEL), 2013 IEEE Symposium on*, pages 31–35. IEEE, 2013.

[22] Le Zhang, Ye Ren, and Ponnuthurai N Suganthan. Towards generating random forests via extremely randomized trees. In *Neural Networks (IJCNN), 2014 International Joint Conference on*, pages 2645–2652. IEEE, 2014.

[23] Le Zhang and PN Suganthan. A comprehensive evaluation of random vector functional link networks. *Information Sciences*, 367:1094–1105, 2016.

[24] Le Zhang and Ponnuthurai N Suganthan. Oblique decision tree ensemble via multisurface proximal support vector machine. *IEEE transactions on cybernetics*, 45(10):2165–2176, 2015.