Random Projection Ensemble Classifiers

Alon Schclar and Lior Rokach

Department of Information System Engineering,
and
Deutsche Telekom Research Laboratories,
Ben-Gurion University, Beer-Sheva, Israel

Summary. We introduce a novel ensemble model based on random projections. The contribution of using random projections is two-fold. First, the randomness provides the diversity which is required for the construction of an ensemble model. Second, random projections embed the original set into a space of lower dimension while preserving the dataset's geometrical structure to a given distortion. This reduces the computational complexity of the model construction as well as the complexity of the classification. Furthermore, dimensionality reduction removes noisy features from the data and also represents the information which is inherent in the raw data by using a small number of features. The noise removal increases the accuracy of the classifier.
The proposed scheme was tested using WEKA based procedures that were applied to 16 benchmark dataset from the UCI repository.

Key words: Ensemble methods, Random projections, Classification, Pattern recognition

1 Introduction

Ensemble methods are very popular tools in pattern recognition due to their robustness and higher accuracy relatively to non-ensemble methods [18]. Rather than relying on a single classifier, they incorporate several classifiers where, ideally, the combination of classifiers outperforms each of the individual classifiers. In fact, ensemble methodology imitates our second nature to seek several opinions before making any crucial decision. We weigh the individual opinions, and combine them before reaching a final decision [24].

Successful applications of the ensemble methodology can be found in many fields: finance [20], manufacturing [26] and medicine [22], to name a few.

One of the most common approaches for creating an ensemble classifier constructs multiple classifiers based upon a single given inducer e.g the nearest neighbor inducer and C4.5 [25]. The classifiers are constructed via a training step. Each classifier is trained on a different training set, all of which are derived from the original training set. The classification result of the ensemble algorithm combines the results of the different classifiers (e.g. by a voting scheme). Ensemble methods can also be applied to regressors in which case a multivariate function is used to combine the individual regression results of the classifiers [27]. In this paper, we focus on classifiers rather than on regressors.

Two crucial components in an effective ensemble method are *accuracy* and *diversity*. Accuracy requires that each individual classifier will generalize as much as possible to new test instances i.e. *individually* minimize the generalization error. Diversity [19] requires that the individual generalization errors will be uncorrelated as much as possible. These components are contradictory in nature. On one hand, if every individual classifier in completely accurate, then the ensemble is not diverse and is not required at all. On the other hand, if the ensemble is completely diverse the ensemble classification is equivalent to random classification. In [23], "kappa-error" diagrams are introduced in order to show the effect of diversity at the expense of reduced individual accuracy.

When using classifiers that are derived from a single inducer, the diversity is achieved by construction of different training sets. One of the most common ensemble methods of this type is the Bagging algorithm [5] which obtains the diversity by creating the various training sets via bootstrap sampling (allowing repetitions) of the original dataset. This method is simple yet effective. Bagging was successfully applied to a variety of problems e.g. spam detection [30] and analysis of gene expressions [28].

In this paper, we utilize random projections to construct a novel ensemble algorithm. Specifically, a set of random matrices is generated. The training sets of the different classifiers are constructed by projecting the original training set onto the random matrices. This approach is different from the random subspaces [16] method that is used in [27]. In random subspace, each training set is composed of a random subset of *features*. However, in random projection, every derived feature is a random linear combination of the original features. In this sense, random subspaces are equivalent to random feature *selection* while random projections are equivalent to random feature *extraction*.

When designing the proposed algorithm, we aimed to construct an algorithm which will require limited computational resources i.e. the algorithm was designed so its complexity would be as low as possible. Accordingly, it stands to reason to compare the proposed algorithm only with algorithms of the same complexity category. The most prominent algorithm in this complexity category is the Bagging algorithm, whose complexity is only slightly lower than the complexity of the proposed algorithm. No comparison is made with more complex ensemble algorithms such as AdaBoost [12] and Rotation Forests [1].

1.1 Random projections

The utilization of random projections as a tool for dimensionality reduction stems from the pioneering work of Johnson and Lindenstrauss [17] who laid the theoretical foundations of dimensionality reduction by proving its feasibility. Specifically, they showed that N points in N dimensional space can almost always be projected onto a space of dimension $C\log N$ with control on the ratio of distances and the error (distortion). Bourgain [4] showed that any metric space with N points can be embedded by a bi-Lipschitz map into an Euclidean space of $\log N$ dimension with a bi-Lipschitz constant of $\log N$. Thus, random projections reduce the dimensionality of a dataset while preserving its geometrical structure.

Applications of the above theorems, which use random projections for dimensionality reduction, were successfully used for protein mapping [21], reconstruction of

frequency-sparse signals [7, 9], face recognition [13] and textual and visual information retrieval [3].

Random projections were also utilized as part of an ensemble algorithm for clustering in [10] and for gene expression data analysis in [11]. Essentially, the random projection algorithm is used to reduce the dimensionality of the dataset. Then an EM (of Gaussian mixtures) clustering algorithm is applied to the dimension-reduced data. However, it is reported in [10] that using a single run of the random projection algorithm produces poor and unstable results. This is due the unstable nature of random projections. Accordingly, an ensemble algorithm is proposed. Each iteration in the algorithm is composed of two steps: (a) dimensionality reduction via random projection and (b) application of the EM clustering algorithm. The ensemble algorithm achieves results that are better and more robust than those obtained by single runs of random projection/clustering and are also superior to a similar scheme which uses PCA to reduce the dimensionality of the data.

In the following, we formally describe the random projection algorithm for dimensionality reduction. Let

$$\Gamma = \{x_i\}_{i=1}^{N} \tag{1}$$

be the original high-dimensional dataset given as a set of column vectors where $x_i \in \mathbb{R}^n$, n is the (high) dimension and N is the size of the dataset. All dimensionality reduction methods embed the vectors into a lower dimensional space $\mathbb{R}^q$ where $q \ll n$. Their output is a set of column vectors in the lower dimensional space

$$\widetilde{\Gamma} = \{\widetilde{x_i}\}_{i=1}^{N}, \widetilde{x_i} \in \mathbb{R}^q \tag{2}$$

where q approximates the intrinsic dimensionality of Γ [15, 14]. We refer to the vectors in the set $\widetilde{\Gamma}$ as the *embedding vectors*.

In order to reduce the dimensionality of Γ using random projections, a random vector set $\Upsilon = \{r_i\}_{i=1}^{n}$ is first generated where $r_i \in \mathbb{R}^q$. Two common choices for generating the random basis are:

1. The vectors $\{r_i\}_{i=1}^{n}$ are uniformly (or normally) distributed over the q dimensional unit sphere.
2. The elements of the vectors $\{r_i\}_{i=1}^{n}$ are chosen from a Bernoulli +1/-1 distribution and the vectors are normalized so that $\|r_i\|_{l_2} = 1$ for $i = 1, \ldots, n$.

Next, a $q \times n$ matrix R whose columns are composed of the vectors in Υ, is constructed. The embedding $\widetilde{x_i}$ of x_i is obtained by

$$\widetilde{x_i} = R \cdot x_i$$

2 The proposed algorithm

In the proposed algorithm, random projections are used in order to create the training sets on which the classifiers will be trained. Using random projections provides the required diversity component of the ensemble method. Although the complexity of using

random projections is slightly higher than that of the Bagging algorithm, random projections possess useful properties that can help obtain better classification results than those achieved by the Bagging algorithm. In particular, random projections reduce the dimensionality of the dataset while maintaining its geometrical structure within a certain distortion rate [17, 4]. This reduces the complexity of the classifier construction as well as the complexity of the classification of new members while producing classifications that are close to or better than those of the original dataset. Furthermore, dimensionality reduction removes noisy features and thus can improve the generalization error.

One of the crucial parameters to any dimensionality reduction algorithm is the dimension of the target space. In the proposed algorithm, we set the dimension of the target space to a portion of the dimension of the ambient space where the training members reside.

Algorithm description Given a training set Γ as described in Eq. 1, we construct a matrix G of size $n \times N$ whose columns are composed of the column vectors in Γ

$$G = (x_1 | x_2 | \ldots | x_N)$$

Next, we generate k random matrices $\{R_i\}_{i=1}^{k}$ of size $q \times n$ where q and n are described in the previous section and k is the number of classifiers in the ensemble. The columns are normalized so that their l_2 norm will be 1.

The training sets $\{T_i\}_{i=1}^{k}$ for the ensemble classifiers are constructed by projecting G onto the random matrices $\{R_i\}_{i=1}^{k}$, i.e. $T_i = R_i \cdot G$ where $i = 1, \ldots, k$. These training sets are input to an inducer $\mathfrak{I}$ and the outcome in a set of classifiers $\{\mathfrak{C}_i\}_{i=1}^{k}$.

In order to classify a new member u by a classifier $\mathfrak{C}_i$, u must first be embedded into the dimension-reduced space $\mathbb{R}^q$. This is achieved by projecting u onto the random matrix R_i

$$\tilde{u} = R_i \cdot u.$$

where $\tilde{u}$ is the embedding of u. The classification of u is set to the classification of $\tilde{u}$ by $\mathfrak{C}_i$. The final classification of $\tilde{u}$ by the proposed ensemble algorithm is produced via a voting scheme that is applied to the classification outcomes of all the classifiers $\{\mathfrak{C}_i\}_{i=1}^{k}$ for the $\tilde{u}$.

3 Experimental results

We tested our approach on 16 datasets from the UCI repository [2] which contains commonly used benchmark datasets that are used to test machine learning algorithms e.g. classifiers. We used the nearest-neighbor inducer (WEKA's B1 lazy classifier) to construct 10 classifiers in each ensemble where the results are the average of 10 ensembles. The size of the dimension-reduced space was set to half of the dimension of the training set. The random matrices were generated from a Gaussian distribution.

Table 1 describes the results of the experiments comparing the performance of the proposed algorithm with the performance of the Bagging algorithm. We also include the

results of the simple non-ensemble nearest-neighbor (NENN) classifier and an ensemble algorithm which is based on the Random subspaces (RS) approach (each training set contains 50 percent randomly chosen features). For each dataset, we specify the number of instances, the number of features (original dimensionality) and the *Generalized Accuracy* of the two algorithms. The generalized accuracy represents the mean probability that an instance was classified correctly and it was calculated via a 10-fold cross-validation procedure which was repeated ten times. Since the average accuracy is a random variable, the confidence interval was estimated by using the normal approximation of the binomial distribution. The one-tailed paired t-test [8] with a confidence level of 95% verified whether the differences in accuracy between the proposed algorithm and the Bagging algorithm were statistically significant. It can be seen in Table 1 that the proposed algorithm significantly outperforms the Bagging algorithm in six (Hill Valley, Isolet, Madelon, Sat, Waveform with noise and Waveform without noise) datasets out of the 16 benchmark datasets. Furthermore, the proposed algorithm outperforms without statistical significance the Bagging algorithm in five (Musk1, Musk2, Ecoli, Glass, Ionosphere) out of the 16 benchmark datasets. On the other hand, the Bagging algorithm significantly outperforms the proposed algorithm in five datasets (Multiple features, Segment, Shuttle, Spambase and Wine). However, the dimensionality in these cases is less than 101 and the proposed algorithm dominates the datasets whose dimension is higher than 100. The proposed algorithm outperforms each of the NENN and the RS algorithms in 11 of the test datasets. The NENN algorithm and the RS algorithm outperform the proposed algorithm in five datasets. In three of which (Segment, Multiple features and Wine), both them outperform it.

In order to conclude which algorithm performs best over all the benchmark datasets, we use the Wilcoxon test [8] whose definition follows: let δ_i be the difference between the performance scores of the two classifiers on the j-th out of the $\Omega = 16$ datasets. We rank the differences according to their absolute values. In case of ties, average ranks are assigned. Let ρ^+ be the sum of ranks for the data sets on which the proposed algorithm outperformed the Bagging algorithm, and ρ^- be the sum of ranks for the opposite. Cases for which $\delta_i = 0$ are split evenly between the sums. Formally, $\rho^+$and ρ^- are defined as follows:

$$\rho^+ = \sum_{\delta_i>0} rank(\delta_i) + \frac{1}{2}\sum_{\delta_i=0} rank(\delta_i);$$

$$\rho^- = \sum_{\delta_i<0} rank(\delta_i) + \frac{1}{2}\sum_{\delta_i=0} rank(\delta_i)$$

Let $\tau = \min(\rho^+, \rho^-)$ be the smaller of the sums. Define the statistic

$$z = \frac{\tau - \frac{1}{4}\Omega(\Omega+1)}{\sqrt{\frac{1}{24}\Omega(\Omega+1)(2\Omega+1)}}$$

which for a larger number of data sets is distributed approximately normally. For the datasets that we used, we got $z = -1.29$ to $\alpha = 0.1$. Thus, the proposed algorithm significantly outperforms the Bagging algorithm with $z = -1.29, p < 0.1$.

Table 1: Properties of the benchmark datasets along with a comparison between the performance of the proposed algorithm and the Bagging algorithm. The '++' postfix means that the proposed algorithm is significantly more accurate than the Bagging algorithm. The converse is marked by a '–' postfix. The '+' postfix indicates that the proposed algorithm is more accurate than the Bagging algorithm without statistical significance. The two right columns contain the results of a Random-subspace based ensemble algorithm and a non-ensemble nearest neighbor classification algorithm.

Dataset Name	Instance#	Feature#	Proposed algorithm	Bagging	Random subspaces	Non-ensemble NN
Hill Valley ++	2424	100	73.15±7.41	61.38±5.09	61.89±4.11	61.38±4.3
Isolet ++	7797	617	90.56±1.02	89.77±1.02	90.01±1.03	87.09±1.19
Madelon ++	2000	500	67.72±3.36	55.63±3.29	55.1±3.47	53.25±3.13
Multiple features –	2000	649	96.11±1.3	97.9±0.9	97.9±0.92	97.65±1.01
Sat ++	6435	36	91.06±1.06	90.41±0.92	90.41±0.97	88.97±1.12
Segment –	2310	19	96.27±1.21	97.03±1.17	97.15±1.11	96.76±1.1
Shuttle –	58000	9	99.79±0.06	99.93±0.03	99.93±0.03	99.75±0.06
Spambase –	4601	57	85.56±1.44	91±1.35	90.78±1.36	86.56±1.71
Waveform w noise ++	5000	40	79.63±1.88	73.8±1.7	73.41±1.82	73.22±2.13
Waveform w/o noise ++	5000	21	80.93±1.81	77.4±1.67	77.17±1.63	71.91±1.88
Wine –	178	13	76.96±8.87	95.07±4.31	95.12±4.34	91.07±6.12
Musk1 +	476	166	86.98±4.73	85.65±4.91	85.55±4.79	83.51±5.27
Musk2 +	6598	166	95.89±0.67	95.79±0.7	95.7±0.72	95.43±0.72
Ecoli +	336	7	83.42±5.38	80.98±6.1	80.66±6.16	73.96±6.15
Glass +	214	9	71.44±9.18	69.98±9.25	70.3±8.96	73.03±9.95
Ionosphere +	351	34	90.4±4.55	87.36±5.06	87.1±5.12	85.62±5.21

4 Conclusion and future work

In the this paper, we introduced an alternative ensemble method to the Bagging algorithm. The method uses random projections instead of the bootstrap sampling that is used by the Bagging algorithm. The proposed method proves to be superior to the bagging algorithm in several datasets while producing competitive results for the other datasets.

The results in this paper are promising. However, a question that needs further investigation is when does the proposed method outperform the Bagging algorithm. Ideally, rigorous criteria should be formulated.

Furthermore, the proposed method should also be tested using other inducers e.g. classification and regression trees [6], SVM [29], etc.

References

1. C. J. Alonso. Rotation forest: A new classifier ensemble method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1619–1630, 2006.
2. A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.
3. E. Bingham and H. Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the 7th ACM SIGKDD International Conference on*

Knowledge Discovery and Data Mining (KDD-2001), pages 245–250, San Francisco, CA, USA, August 26-29 2001.
4. J. Bourgain. On lipschitz embedding of finite metric spaces in Hilbert space. *Israel Journal of Mathematics*, 52:46–52, 1985.
5. L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
6. L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Chapman & Hall, Inc., New York, 1993.
7. E. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, February 2006.
8. J. Demsar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.
9. D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, April 2006.
10. X. Zhang Fern and C. E. Brodley. Random projection for high dimensional data clustering: A cluster ensemble approach. pages 186–193, 2003.
11. R. Folgieri. *Ensembles based on Random Projection for gene expression data analysis*. PhD thesis, University of Milano, 2007.
12. Y. Freund and R. Schapire. Experiments with a new boosting algorithm. machine learning. In *Proceedings for the Thirteenth International Conference*, pages 148–156, San Francisco, 1996. Morgan Kaufmann.
13. N. Goel, G. Bebis, and A. Nefian. Face recognition experiments with random projection. In *Proceedings of SPIE*, volume 5779, 426, 2005.
14. C. Hegde, M. Wakin, and R. G. Baraniuk. Random projections for manifold learning. In *Neural Information Processing Systems (NIPS)*, December 2007.
15. M. Hein and Y. Audibert. Intrinsic dimensionality estimation of submanifolds in Euclidean space. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 289–296, 2005.
16. T. Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:832844, 1998.
17. W. B. Johnson and J. Lindenstrauss. Extensions of Lipshitz mapping into Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.
18. L. I. Kuncheva. *Combining Pattern Classifiers. Methods and Algorithms*. John Wiley and Sons, 2004.
19. L. I. Kuncheva. Diversity in multiple classifier systems (editorial). *Information Fusion*, 6(1):3–4, 2004.
20. W. Leigh, R. Purvis, and J. M. Ragusa. Forecasting the nyse composite index with technical analysis, pattern recognizer, neural networks, and genetic algorithm: a case study in romantic decision support. *Decision Support Systems*, 32(4):361–377, 2002.
21. M. Linial, N. Linial, N. Tishby, and G. Yona. Global self-organization of all known protein sequences reveals inherent biological signatures. *Journal of Molecular Biology*, 268(2):539–556, May 1997.
22. P. Mangiameli, D. West D, and R. Rampal. Model selection for medical diagnosis decision support systems. *Decision Support Systems*, 36(3):247–259, 2004.
23. D. D. Margineantu and T. G. Dietterich. Pruning adaptive boosting. In *Proceedings of the 14th International Conference on Machine Learning*, pages 211–218, 1997.
24. R. Polikar. Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, 6(3):21–45, 2006.
25. R. R. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., 1993.

26. L. Rokach. Mining manufacturing data using genetic algorithm-based feature set decomposition. *International Journal of Intelligent Systems Technologies and Applications*, 4(1/2):57–78, 2008.
27. N. Rooney, D. Patterson, A. Tsymbal, and 10 February 2004 S. Anand. Random subspacing for regression ensembles. Technical report, Department of Computer Science, Trinity College Dublin, Ireland, 2004.
28. G. Valentini, M. Muselli, and F. Ruffino. Bagged ensembles of svms for gene expression data analysis. In *Proceeding of the International Joint Conference on Neural Networks - IJCNN*, pages 1844–1849, Portland, OR, USA, July 2003. Los Alamitos, CA: IEEE Computer Society.
29. V. N. Vapnik. *The Nature of Statistical Learning Theory (Information Science and Statistics)*. Springer, November 1999.
30. Z. Yang, X. Nie, W. Xu, and J. Guo. An approach to spam detection by naive bayes ensemble based on decision induction. In *Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications (ISDA'06)*, 2006.