NON-UNIFORM FEATURE SAMPLING FOR DECISION TREE ENSEMBLES

Anastasios Kyrillidis and Anastasios Zouzias

IBM Research Lab, Zurich
{nas,azo}@zurich.ibm.com

ABSTRACT

We study the effectiveness of non-uniform randomized feature selection in decision tree classification. We experimentally evaluate two feature selection methodologies, based on information extracted from the provided dataset: (i) *leverage scores-based* and (ii) *norm-based* feature selection. Experimental evaluation of the proposed feature selection techniques indicate that such approaches might be more effective compared to naive uniform feature selection and moreover having comparable performance to the random forest algorithm [3].

1. INTRODUCTION

Living in the era of Big Data, massive amount of information is now publicly available, aggrandizing our expectations for new developments, both in well-established and contemporary scientific tasks. However, this ever increasing data often contradicts with the principle of *parsimony*: in a high-dimensional feature space the proper selection of features, that results in succinct descriptions of the problem, cannot be easily derived. This fact jeopardizing the interpretability of the solution. This curse of dimensionality can also pose difficulties with respect to the qualitative performance of methods, imperilling their accuracy as well as their robustness in the case of noise and outlier presence.

An important application that suffers from this difficulty is classification. An abstract description for the case of binary classification is given below:

BINARY CLASSIFICATION PROBLEM: *Assume*

$$\mathcal{D}_{train} = \{(\mathsf{X}_1, y_1), \ldots (\mathsf{X}_n, y_n)\}$$

be a collection of n supervised train feature vectors $\mathsf{X}_i \in \mathbb{R}^d$ with corresponding labels $y_i \in \{\pm 1\}$. Given $\mathcal{D}_{train}$, we want to learn a classifier $\mathcal{C} : \mathbb{R}^d \to \{\pm 1\}$ such that, for an unsupervised *input $\mathcal{D}_{test} = \{\mathsf{X}_j : \mathsf{X}_j \notin \mathcal{D}_{train}\}$, $\mathcal{C}$ computes labels on the elements of $\mathcal{D}_{test}$ with the lowest possible classification error.*

Several cases have been reported in the literature where classification using the over-complete set of features (i.e., without proper selection or pre-processing) can be as poor as random guessing, due to noise accumulation in the high-dimensional feature space [10].[1] Wherefore, irrelevant or redundant information "interfere" with useful one and its removal could gain in classification. Fortunately, common wisdom indicates that, in practice only a few features are important for classification and thus such removal is applicable [1]; e.g., in DNA data [32, 11], only a few genes are influential in a gene sequence expression. Moreover, an excessive number of attributes usually results in prohibitive running times and storage requirements during training for real-time applications; in memory-limited cases, further post-processing is required [21].

In stark contrast, training a well-behaved *individual* classifier with a *predetermined and fixed subset* of features over a restricted train dataset is a difficult task; it often creates overfitting issues, where the loss of generalization is observed on incoming new data. To overcome this difficulty, recent developments [18, 17, 15], based on [20], have proposed the systematic construction of classifiers: randomly and independently selected subsets of features are used per learner and the final decision is taken as a *majority (averaging) rule* over the collection of learners for the given data; such structures are generally known as classifier ensembles [7].

However, a *naive selection* of features might still doubt the practicality of classifier ensembles in such settings: the random selection might lead to extremely "weak" learners, increasing drastically the number of ensemble components required for a desired classification error. Moreover, the authors of [19] highlight the exponentially increasing space-complexity of tree-based ensembles to achieve a given accuracy; thus, more sophisticated selection procedures might lead to less expensive constructions.

To this end, a compromise between these two extremes is imperative in practice: by judiciously selecting a subset of *significant* attributes, allowing randomness during the selection, one can achieve acceptable classification accuracy and desired generalization attributes with low space complexity.

Our contributions: In this context, sophisticated dimensionality reduction techniques might play a crucial role. Instead of selecting features uniformly at random, we utilize linear algebraic techniques to "bias" the selection procedure. Based on the work [9, 23, 30], we use matrix-based information scores to define a non-uniform probability distribution that favors more dominant features.

Empirical results show an overall improved classification capability using our approach, as compared to classic state-of-the-art schemes for a given training time period.

2. RELATED WORK

As already mentioned above, a classical technique in classification focuses on the idea of *ensemble classifiers*: by combining a set of "weak" learners that approximate the training data,

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement n^o 259569.

[1]The authors in [10] demonstrate further that almost all linear discriminants can perform as poorly as the random guessing.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

one could obtain a "strong" classifier, i.e., a classifier with better generalization performance (see e.g., [13]). Based on this approach, one can generate several "weak" learners by applying one or combination of the following designs:

(i) *Feature selection*: each classifier is trained over a selected subset of features—e.g., for an excellent introduction, see the subspace method proposed in [18, 17] and the Randomized C4.5 algorithm in [8], following the work [1].

(ii) *Training data subsampling and reweighting:* each classifier is trained over a subset of the training samples; then, the sample selection scheme is re-weighted, based on the classification error in the previous iteration—e.g., see the celebrated Boosting technique [31] and references therein.[2]

(iii) *Linear combination of features under random low-dimensional embeddings*, where each learner uses the whole spectrum of features, trasformed by random linear mappings.

There are several works in the literature where combinations of the above designs are used in practice—e.g., Random Forests with feature selection and data subsampling [3].

In [1, 17, 8, 12], the authors consider case (i) where the *random* model is proposed: to train a "weak" learner, each feature is selected uniformly at random, ignoring any prior information. While such strategy is maximally "unbiased" and easy to implement, it might lead to lower classification accuracy when a small number of features is selected each time or to higher complexity, due to the larger amount of classifiers required for a given accuracy level. In [18, 12], the authors further extend this strategy to *node optimization* for tree-based ensembles: at every level of each decision tree in the ensemble, the splitting decision rule over each node is derived using a random subset of features; in this work, our proposal does not consider this case and we leave it as a future research direction.

In this context, [3, 12] show that randomization yields strict improvements over simple deterministic selection heuristics.[3] While simplicity helps, randomization is particularly useful in "adversarial" settings where "bad" features are present. [3] states the accuracy of RFs is insensitive to randomness in practice, see also [12]. However, subsequent developments on tree-based classifiers do not espouse this virtue. The author in [28] proposes the `ReliefF` feature evaluation metric: the significance of each attribute is inversely proportional to what extent its values separate similar observations into different classes; details of this algorithm are provided in [29]. [4] proposes the *Probabilistic RFs* where feature selection procedure is further linearly transformed with linear kernels. Such works amplify the suspicion that one can achieve more by applying sophisticated selection strategies rather than the blind randomization model.

In this paper, we attempt to raise this suspicion even more: we study several matrix-based sampling measures in order to signify important features in the selection procedure for higher classification accuracy.

3. PROBLEM SETTING

Throughout this paper, we use *decision trees* as "weak" learners; an illustrative example for a binary decision tree is given in Figure 1. These structures have inherent interpretation capabilities due to the explicit decision rules on the splitting nodes, are non-parametric, easy to implement and extremely fast to train, as compared to other classifiers. A non-exhaustive list of alternatives include linear classifiers [33, 22], Support Vector machines [6, 14], Neural Networks [34], etc; we conjecture that our proposed feature selection scheme can be easily applied to these cases and we leave this research direction for future work.

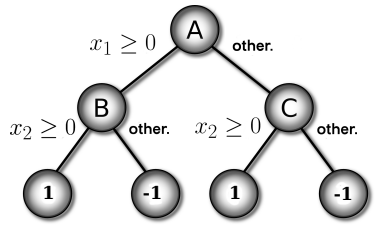


Fig. 1: Toy-example decision tree with two features x_1, x_2. Here, A denotes the full sample set, $B, C \subseteq A$ are subsets satisfying the decision rule on node A. Leaf nodes contain only samples from class 1 or -1.

In the realm of random decision tree ensembles, we generate a set of decision trees, built on a subset of the initial feature set. We construct the ensembles as follows: for each tree, we sample non-uniformly and independently at random a set of k features. Then, we train a decision tree classifier in its entirety (with a deterministic splitting criterion), restricted on these k features.

It is important to notice that the above randomized procedure is quite different than Breiman's random forest algorithm (RFs) [3, 18]. In a parameter free implementation of RFs without bagging, each tree of the forest utilizes randomness in the splitting process of its construction. Conventional wisdom indicates that, at each node during the tree growing process, an uniformly random (and possibly different) sample of $\sqrt{d}$ features is utilized.

4. OUR APPROACH IN A NUTSHELL

To describe the main ideas of our approach, assume that we represent the training dataset of n objects with d features as an $n \times d$ real[4] matrix A. We propose the `LEverage ScoreS (LESS)` tree ensemble algorithm, a two-phase classifier construction, as reported in Algorithm 1. Let $\Pi := \{\pi_1, \pi_2, \ldots, \pi_d\}$ denote a probability distribution over the set of features that signifies the importance of features over $\mathcal{D}_{\texttt{train}}$. In the first phase, we compute Π, based on ideas described in Section 5. Next, we "feed" Π into the second phase of our approach where: (i) we select k features according to Π and, (ii) based on these features, we generate t decision trees. Finally, the collection of these decision trees is gathered and a standard majority voting scheme is applied to derive the predicted labelling of the model. Namely, given a set of decision trees and an unlabelled example, the algorithm returns as its predicted label the most frequent label over all the decision trees.

[2] To use such approach, one assumes many passes over the data.

[3] A non-exhaustive list of such rules in the case of tree-based classifiers includes Gain ratio and Gini index, designed to result into simple models for classification.

[4] We implicitly assume that features are real-valued.

Algorithm 1 `LEverage ScoreS (LESS) Trees`

procedure `LESS`($\mathsf{A}, \mathbf{y}, t, k$) ▷ $\mathsf{A} \in \mathbb{R}^{n \times d}, \mathbf{y} \in \{\pm 1\}^n$
▷ $t, k \in \mathbb{N}$: # of trees and features
  Compute Π, according to Eqn. (1).
  for $k = 0, 1, 2, \ldots, t-1$ **do**
    Sample k features of A using Π.
    Construct $\mathsf{A}^{(k)} \in \mathbb{R}^{n \times k}$ restricted to the k features.
    Train decision tree using $(\mathsf{A}^{(k)}, \mathbf{y})$
  end for
  Output: collection of t trees.
end procedure

Several remarks can be highlighted about the above algorithm. First, each decision tree is constructed on only a subset of features of size k (usually k is between 10 and 50); hence, as we show in Section 6, Algorithm 1 has computational advantages over models that compute the best split over the whole feature set. Along the same lines, the resulting collection of trees are more interpretable than RFs since each tree depends only on a small set of features. Last, the random process described in Algorithm 1 is simpler than the random forest algorithm [3] and hence might be amenable to theoretical justification in the future.

5. FEATURE SELECTION SCHEMES

Exploiting the full spectrum of features creates a tradeoff between interpretability and predictive accuracy. Thus, an important step to process such large-scale data is to construct an "importance score" for each column of A to denote the influence of the corresponding feature. Given such measure, we can then sample a predefined number of features k for each decision tree, based on these scores.

In this section, we describe three subsampling techniques for feature selection: (i) uniform sampling, (ii) column squared-norm based sampling and, (iii) leverage scores-based sampling [9] (to be defined shortly).

Uniform sampling: each feature is selected with equal probability. Both strategies, where features are selected with or without replacement, have been tested; cf., [17, 18]. We use this policy as the baseline performance in our experiments.

Norm-based sampling: Recent developments on geometric functional analysis have dictated that squared norm subsampling can approximate well large datasets incurring small spectral norm [30]. Namely, in our setting sampling the i-th feature with probability proportional to $\|\mathsf{A}_i\|_2^2$ where A_i is the i-th column of A and $\|\cdot\|_2$ denote the Euclidean norm.

Statistical leverage scores sampling: The goal of statistical leverage scores is to construct a judiciously-chosen nonuniform importance sampling distribution over the set of columns, based on factorizations of A, according to the following definition:

Definition 1 (Statistical leverage scores [5, 9]) *Let $\mathsf{A} \in \mathbb{R}^{n \times d}$ be a data matrix with n objects and d features with $r := \mathit{rank}(\mathsf{A}) \leq n$ for $n \ll d$. Moreover, let $\mathsf{A} = \mathsf{U}\Sigma\mathsf{V}^\top$ be its* Singular Value Decomposition *(SVD) where $\mathsf{V} \in \mathbb{R}^{n \times r}$ contains the set of right singular column vectors. The normalized statistical leverage scores over the set of columns of A are defined as:*

$$\pi_j = \frac{1}{r} \sum_{i=1}^{r} (v_i(j))^2, \ \textit{for } j = 1, \ldots, d, \tag{1}$$

where $v_i(j)$ denotes the j-th entry of the i-th right singular vector.

We highlight that, while sampling schemes (ii) and (iii) are slower than the simple uniform sampling, both result into generally higher classification accuracy for given sampling volume as we show in Section 6. Moreover, strategy (iii) can be well-approximated using fast randomized algorithms [23]. We note that leverage scores converge to uniform sampling when the coherence of the data matrix is small and scheme (i) turns to be the optimal; thus, scheme (iii) can be considered as a more generic selection strategy that includes (i) as a special case.

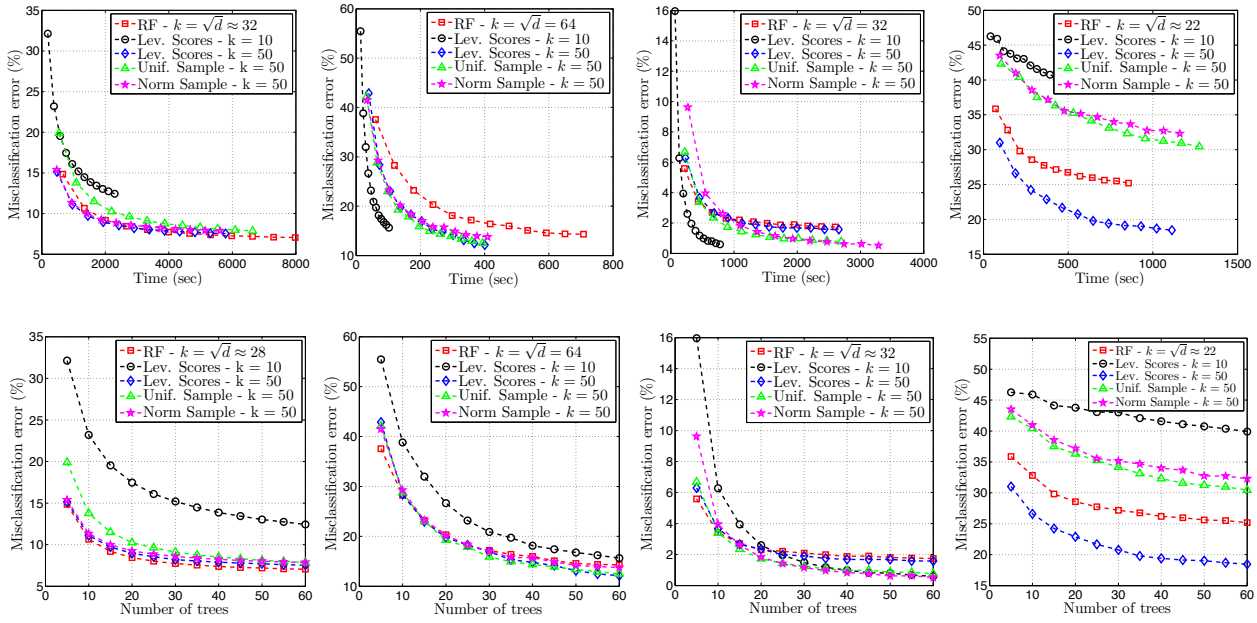
6. EXPERIMENTS

In this section, we experimentally compare Algorithm 1 with two variants of Algorithm 1: (i) the case where Step 4 is replaced with uniform/norm-based feature sampling and (ii) the classical Breiman's RFs algorithm [3]. Here, we highlight a subtle distinction between the RFs algorithm and all other algorithms under comparison. During the tree construction, RFs utilizes randomness for deciding the next split. Namely, at each node (assuming an additional split has been decided to be made), RFs selects $\lceil \sqrt{d} \rceil$ features uniformly at random on which the best split is selected. Therefore, one expects each resulting tree to possibly depend on all features as opposed to Algorithm 1 that depends on only k features.

Datasets: For the real-world datasets we used four publicly available[5] datasets that we denote by `MNIST`, `ORL`, `PIE` and `MADELON`. The `MNIST` dataset of handwritten digits has a training set of 60,000 examples, and a test set of 10,000 examples [2] (a sample of $10,000$ training examples and $5,000$ test examples is used here). `ORL` contains ten different images each of 40 distinct subjects [26]. There are 400 different objects in total, each having 4096 dimensions. `PIE` is a database of 41,368 images of 68 people [27]. Namely, there are in total 2856 data points with 1024 dimensions. The `MADELON` dataset is an artificial test case, multivariate and highly non-linear, part of the NIPS 2003 feature selection challenge. The data points of `MADELON` is 2000 containing 500 features. These datasets have been selected due to their high-dimensional feature space and/or their heavily usage as benchmarks for classification.

Experimental setup: To measure the impact of leverage scores on the classification performance on random decision trees, we perform a series of diverse experiments on the above datasets. In all reported results, we use the average values of 30 independent executions of its corresponding algorithm. For the `LESS` algorithm, we truncate the computation of SVD to rank $r = 50$ for acceleration. We measured the performance of Algorithm 1 for various settings of k and t, i.e., the number of features to be sampled and the number of trees, respectively. No bagging is performed on the RFs algorithm and the

[5]Most of the datasets used here are available under UCI's machine learning repository [2].



(a) MNIST (b) ORL (c) PIE (d) MADELON

Fig. 2: Each column corresponds to a dataset. The first row depicts the classification error versus the elapsed training time for increasing number of trees. The second row depicts the classification accuracy versus the number of trees.

default number of subsampled features is selected, i.e., $\sqrt{d}$. All timings were performed under MATLAB R2011b [24].

Numerical results: We report our experimental evaluations in Figure 2 and in Table 1. In the first row of Figure 2 we depict

Total number of nodes for given ϵ classification accuracy					
	ϵ	RFs	Uniform	Norm	Lev. Scores
MNIST	$\sim 7\%$	102066	125152	**92064**	93527
ORL	$\sim 12\%$	**5498**	5568	5566	5587
MADELON	$\sim 26\%$	26407	>T	>T	**4003**
PIE	$\sim 1\%$	**18198**	20632	24851	19137

Table 1: Total number of nodes using $k = 50$ features.

the classification error versus the elapsed training time for a pair of train and test data. The rationale behind this plot is to demonstrate that Algorithm 1 can achieve similar or better classification error than RFs with lower computational requirements. In all cases, LESS algorithm with $k = 50$ is superior or matches the performance of RFs. On the other hand, the performance of Algorithm 1 with $k = 10$ is not superior in all cases. This is due to the small value of k. Hence, a suggestive setting of k in Algorithm 1 is in the range $[\sqrt{d}, 2\sqrt{d}]$. However, in stark contrast with conventional wisdom, there are cases where only $k = 10$ features, selected using leverage scores, seem to be sufficient to achieve the same or even better classification accuracy in much less training time, increasing the interpretability of the result due to the limited number of used features; e.g., see Figure 2(first row) for the cases ORL and PIE. Moreover, an increased number of features usually results in an increased processing time, with no further classification error improvement. Overall, we observe that LESS trees are at least as accurate as RF, while being less computationally expensive in practice.

The second row of Figure 2 depicts the classification accuracy versus the number of trees. We observe that Algorithm 1 with $k = 50$ matches the performance of RFs in terms of number of trees. Moreover, in the MADELON dataset Algorithm 1 is superior to RFs, which in turn, RFs is superior to both uniform and norm based feature selection. On the other hand, Algorithm 1 with $k = 10$ does not perform well.

We further study the space complexity of the resulting ensembles as a function of the total number of nodes needed among all trees to achieve a predefined classification error $\epsilon > 0$. We also set up a time threshold limit value T = 3600 seconds (1 hour) per approach to achieve accuracy ϵ. Table 1 shows the reported space complexities for all test cases. As observed, both RFs and LESS trees has similar (or even better) space complexity for given ϵ. From a different perspective, in a memory-limited scenario where only a fixed number of nodes can be maintained, non-uniform feature sampling leads to equivalent, if not better, mis-classification error level, as compared to uniform feature selection and/or RFs.

7. DISCUSSION AND FUTURE WORK

In this work, we study feature selection strategies in classification, both in terms of time/space-complexity efficiency as well as of classification accuracy. Overall, results indicate that the proposed tree ensemble, based on leverage scores, might outperform the state-of-the-art RFs [3], as well as schemes where uniform weighting is applied. We observe that the proposed scheme results into low space-complexity trees for better interpretability, requires overall less training time and has at least the same accuracy, as compared to top-notch approaches.

8. REFERENCES

[1] AMIT, Y., AND GEMAN, D. Shape quantization and recognition with randomized trees. *Neural computation 9*, 7 (1997), 1545–1588.

[2] BACHE, K., AND LICHMAN, M. UCI Machine Learning Repository. `http://archive.ics.uci.edu/ml`, 2013. University of California, Irvine, School of Information and Computer Sciences.

[3] BREIMAN, L. Random forests. *Machine learning 45*, 1 (2001), 5–32.

[4] BREITENBACH, M., NIELSEN, R., AND GRUDIC, G. Z. Probabilistic random forests: Predicting data point specific misclassification probabilities. *Univ. of Colorado at Boulder, Tech. Rep. CU-CS-954-03* (2003).

[5] CHATTERJEE, S., AND HADI, A. *Sensitivity Analysis in Linear Regression*. Wiley Series in Probability and Statistics. Wiley, 1988.

[6] CORTES, C., AND VAPNIK, V. Support-vector networks. *Machine learning 20*, 3 (1995), 273–297.

[7] DASARATHY, B. V., AND SHEELA, B. V. A composite classifier system design: concepts and methodology. *Proceedings of the IEEE 67*, 5 (1979), 708–713.

[8] DIETTERICH, T. G. Ensemble methods in machine learning. In *Multiple classifier systems*. Springer, 2000, pp. 1–15.

[9] DRINEAS, P., MAHONEY, M. W., AND MUTHUKRISHNAN, S. Relative-error cur matrix decompositions. *SIAM Journal on Matrix Analysis and Applications 30*, 2 (2008), 844–881.

[10] FAN, J., AND FAN, Y. High dimensional classification using features annealed independence rules. *Annals of statistics 36*, 6 (2008), 2605.

[11] FAN, J., AND REN, Y. Statistical analysis of dna microarray data in cancer research. *Clinical Cancer Research 12*, 15 (2006), 4469–4473.

[12] FAN, W., WANG, H., YU, P. S., AND MA, S. Is random model better? on its accuracy and efficiency. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on* (2003), IEEE, pp. 51–58.

[13] FREUND, Y., AND SCHAPIRE, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci. 55*, 1 (Aug. 1997), 119–139.

[14] GUNN, S. R. Support vector machines for classification and regression. *ISIS technical report 14* (1998).

[15] HANSEN, L. K., AND SALAMON, P. Neural network ensembles. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 12*, 10 (1990), 993–1001.

[16] HE, X., CAI, D., AND NIYOGI, P. Laplacian Score for Feature Selection. In *Neural Information Processing Systems (NIPS)*, Y. Weiss, B. Schölkopf, and J. Platt, Eds. 2006, pp. 507–514.

[17] HO, T. K. Random decision forests. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on* (1995), vol. 1, IEEE, pp. 278–282.

[18] HO, T. K. The random subspace method for constructing decision forests. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 20*, 8 (1998), 832–844.

[19] JOLY, A., SCHNITZLER, F., GEURTS, P., AND WEHENKEL, L. L1-based compression of random forest models. In *20th European Symposium on Artificial Neural Networks* (2012).

[20] KLEINBERG, E. Stochastic discrimination. *Annals of Mathematics and Artificial intelligence 1*, 1 (1990), 207–239.

[21] KULKARNI, V. Y., AND SINHA, P. K. Pruning of random forest classifiers: A survey and future directions. In *Data Science & Engineering (ICDSE), 2012 International Conference on* (2012), IEEE, pp. 64–68.

[22] LIU, C., AND WECHSLER, H. Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *Image processing, IEEE Transactions on 11*, 4 (2002), 467–476.

[23] MAHONEY, M. W., DRINEAS, P., MAGDON-ISMAIL, M., AND WOODRUFF, D. P. Fast approximation of matrix coherence and statistical leverage. In *ICML* (2012).

[24] MATLAB. *7.13.0.564 (R2011b)*. The MathWorks Inc., Natick, Massachusetts, 2010.

[25] NENE, S. A., NAYAR, S. K., AND MURASE, H. Columbia university image library. `http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php`. Technical Report CUCS-005-96, February 1996.

[26] The ORL Database of Faces. `http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html`. AT&T Laborartories Cambridge, UK.

[27] PIE Database. `http://www.ri.cmu.edu/research_project_detail.html?project_id=418&menu_id=261`. Carnegie Mellon University.

[28] ROBNIK-ŠIKONJA, M. Improving random forests. In *Machine Learning: ECML 2004*. Springer, 2004, pp. 359–370.

[29] ROBNIK-ŠIKONJA, M., AND KONONENKO, I. Theoretical and empirical analysis of relieff and rrelieff. *Machine learning 53*, 1-2 (2003), 23–69.

[30] RUDELSON, M., AND VERSHYNIN, R. Sampling from large matrices: An approach through geometric functional analysis. *Journal of the ACM (JACM) 54*, 4 (2007), 21.

[31] SCHAPIRE, R. E. The boosting approach to machine learning: An overview. *Lecture notes in Statistics - New York - Springer Verlag* (2003), 149–172.

[32] WEST, M., BLANCHETTE, C., DRESSMAN, H., HUANG, E., ISHIDA, S., SPANG, R., ZUZAN, H., OLSON, J. A., MARKS, J. R., AND NEVINS, J. R. Predicting the clinical status of human breast cancer by using gene expression profiles. *Proceedings of the National Academy of Sciences 98*, 20 (2001), 11462–11467.

[33] WITTEN, D. M., AND TIBSHIRANI, R. Penalized classification using fisher's linear discriminant. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 73*, 5 (2011), 753–772.

[34] ZHANG, G. P. Neural networks for classification: a survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 30*, 4 (2000), 451–462.