Recommendation Systems for Markets with Two Sided Preferences

Anjan Goswami, *Member, IEEE,* Fares Hedayati, *Member, IEEE,* and Prasant Mohapatra, *Fellow, IEEE*

Abstract—In recent times we have witnessed the emergence of large online markets with two-sided preferences that are responsible for businesses worth billions of dollars. Recommendation systems are critical components of such markets. It is to be noted that the matching in such a market depends on the preferences of both sides, consequently, the construction of a recommendation system for such a market calls for consideration of preferences of both sides. The online dating market, and the online freelancer market are examples of markets with two-sided preferences. Recommendation systems for such markets are fundamentally different from typical rating based product recommendations. We pose this problem as a bipartite ranking problem. There has been extensive research on bipartite ranking algorithms. Typically, generalized linear regression models are popular methods of constructing such ranking on account of their ability to be learned easily from big data, and their computational simplicity on engineering platforms. However, we show that for markets with two sided preferences, one can improve the AUC (Area Under the receiver operator Curve) score by considering separate models for preferences of both the sides and constructing a two layer architecture for ranking. We call this a two-level model algorithm. For both synthetic and real data we show that the two-level model algorithm has a better AUC performance than the direct application of a generalized linear model such as L_1 logistic regression or an ensemble method such as random forest algorithm. We provide a theoretical justification of AUC optimality of two-level model and pose a theoretical problem for a more general result.

Keywords—Recommender systems, Two-sided markets.

1 Introduction

Two-sided markets [1] facilitates transaction between two distinct user groups. Today, several companies provide such two sided market platforms. Examples of such markets include e-commerce platform such as eBay, online labor markets [2] such as Elance-oDesk, online dating services such as eHarmony and match.com, and online housing rentals such as Airbnb and many others. In this paper, we consider the two sided markets where both user groups have distinct preferences such as online dating (male and female user groups) or online labor market (employee and employer user groups). We call such markets as "markets with two sided preferences". Matching of users from both sides is central to the business models of all such two-sided markets. One way of providing algorithmic matching for such markets is to build an efficient recommendation system [3]. Generally these markets with two sided preferences employ some mechanisms for promoting interactions between the two user groups. For example, Elance-oDesk's platform comprises of mechanisms such as bidding for jobs by contractors, inviting of contractors by the employers who posted jobs, rejecting or accepting of the bids submitted by the contractors, or rejecting or accepting the invites sent by the employers etc. This user behavior data (mostly counts of these events) and the attributes of the user groups can be used to model preferences of the two user groups separately. One can also construct joint features consisting of attributes of two user groups. All these features then can be used to estimate probability of matching. The probability of matching can be considered as a function of probability of preferences of the two user groups. In this paper, we pose the recommendation problem as a bipartite ranking problem and we measure the accuracy of the system using AUC (Area Under the receiver operator Curve) score. An AUC score represents the ratio of the number of pairs for which the probability of matching with a good candidate is

- *Anjan Goswami (agoswami@ucdavis.edu) and Prasant Mohapatra (pmohapatra@ucdavis.edu) are with the Department of Computer Science at UC Davis, California.*
- *Fares Hedayati (fareshed@eecs.berkeley.edu) is a senior data scientist in Elance-oDesk.com*

higher than probability of matching with an inferior candidate and the total number of all matching pairs. This guarantees that the recommended set will be ranked according to the desirability of the candidates for the purpose of matching. In the case of online labor market, there can be only one or a few matching employees or contractors for a job posted by an employer. Hence, AUC score will be high for a model if the top ranked candidates are recruited by the employers. In this paper, we show that one can exploit the special structure of typical data sets available in the markets with two sided preferences and can design a two layer architecture that can produce high AUC score compared to some popular regression based algorithms. We call this algorithm as "two-level model". In the first level this estimates the preference functions of both sides as well as a general matching function; in the second level it computes the probability of matching of two users by way of an approximate Bayesian matching technique that incorporates a binning based density estimation. We show that this simple two-level model results in a 2 to more than 10% higher AUC (area under the curve) score and similar gains in TPR (true positive rate) and TNR (true negative rate) compared to direct regression based estimation of probability of matching using L_1 regularized logistic regression [4] or random forest [5].

2 Background on recommendation systems

There has been extensive research on recommendation systems [3], [6] because of their commercial importance. Popular algorithms include nearest neighbor based collaborative filtering [7], content based filtering [8], matrix factorization [9], regression, classification [10], and learning to rank [11] based algorithms. Collaborative filtering or matrix factorization have been extensively used particularly in the e-commerce industry, however they are best to use when estimating a rating given by users. However, when the goal is to obtain a recommendation based on an n-dimension feature constructed from the attributes of two user groups, the input is a tensor [12] and the factorization problem becomes much more complex. The nearest neighbor techniques are also known to suffer from curse of dimensionality problems and not the preferred algorithm when n is high. Notably, a few researchers from the industry [13], [14], [15] have reported the use of linear models for matching in online dating and labor markets. They essentially have constructed one model for matching based on regression based techniques directly from the features created from the attributes and behaviors of two user groups on the site. Generally, the linear models [16] are also suitable for learning from big data. Furthermore, the underlying algorithms for these models can be easily scaled up to include many features, these models are also easy to deploy in engineering platforms, and they can be computed fast. These reasons make linear models very popular for building recommendation systems for two sided markets in the industry. Interestingly, very few papers have actually focussed on recommendation systems for two sided markets, barring some studies on matching in the context of online dating [17]. In one such study [17], it has been observed that the recommendation systems in such markets can have a reciprocal relationship between entities of both sides. In this paper the authors describe the characteristics of such a reciprocal relationship and present a heuristic algorithm for designing recommendation systems for online dating by computing a compatibility score for each side. Their paper discusses about the special characteristics of the problem of recommendation systems for markets with two sided preferences. Their algorithm is based on a compatibility score that is constructed from the given preferences of the users. Their compatibility score approach has been shown to improve the precision of the recommender systems but they did not report any ranking metric in the paper. The success of their algorithm depends on correctness of the specific preferences provided by the users that can often be unreliable. Our method on the other hand does not depend on user specified preferences and is much more generally applicable.

3 Problem setting

Let us assume that there are two user groups u and u' in a two-sided marketplace. We model their preference to each other by $y = f_1(\bar{x_1})$ and $y' = f_2(\bar{x_2})$ respectively. Let $m = f_3(\bar{x_3})$ denote the function that represents the matching between the two user groups, where $f_i : R^n \rightarrow \{0, 1\}\ \forall i$ is an arbitrary function and $\bar{x_i} \in R^n$ is an n-dimensional feature vector and $i \in \{1, 2, 3\}$ represents three different feature vectors. In all our experiments, we used $x_1 = x_2 = x_3$. The features are constructed from attributes of u and u' and interactions between them. The values $y = 1$ and $y' = 1$ denote positive preferences whereas the value $m = 1$ denotes a match. The probability of preferences of user groups u and u' can then be represented by $P((y = 1)|\bar{x})$ and $P((y' = 1)|\bar{x})$. Similarly, the probability of matching can be represented by $P((m = 1)|\bar{x})$.

4 Main idea

In this paper, we propose a new approach to maximize AUC scores for recommendation systems in markets with two-sided preferences via a two-level model. In the first level, we estimate the probabilities $P((y = 1)|\bar{x})$ and $P((y' = 1)|\bar{x})$ using a L_1-LR (l_1 regularized logistic regression) [4]. This corresponds to functions $f_1(\bar{x})$ and $f_2(\bar{x})$ respectively. We construct a third model using a L_1-LR that corresponds to the probability $P((m = 1)/\bar{x}) = f_3(\bar{x})$. We use a cross validation technique to choose a regularization parameter λ for each logistic model. Note, the reason we have used the third model because we have seen this to improve the AUC score although we have not reported the results without the third model. Intuitively, one can consider that the first three models capture some of the latent features from this data sets. We use a L_1-LR for the following reasons:

- It is easy to train with big data. Several efficient algorithms [18] exist to solve the optimization problems.
- L_1 minimization tends to give sparse solutions and has logarithmic sample complexity bounds [4].
- It is also an effective learner even in circumstances involving an exponential number of irrelevant features [19].
- L_1 regularization has appealing asymptotic sample consistency for feature selection [20].
- Logistic loss is proper and is known to have a better regret bound with pairwise ranking. This makes it a good choice for maximizing AUC score using a univariate loss function [21].

All the above properties ensure that we can efficiently learn the preference function of each side using large amount of historical data, and these models can have better AUC scores for matching. Furthermore, Logistic Regression or other generalized linear models are preferred for many commercial ranking and recommendation applications primarily because computing scores or probabilities in real time in production is much easier compared to more expressive and powerful ensemble models such as random forest or gradient boosted tree [22]. Noteworthy here is the fact that, although the algorithm uses three logistic models, it is applicable to a more general framework where multiple hidden states may exist to model the interactions of agents from two sides in a marketplace. In the second level, we aim to compute

$$P\left(m = 1|f_1(\bar{x}), f_2(\bar{x}), f_3(\bar{x})\right) \tag{1}$$

which is the probability of matching given the models obtained in first level. It is well known that algorithms that can estimate the optimal class probability [23] also have nice AUC properties. One way of optimally estimating class probability is to use a proper composite loss function [24]. Reid et al. defined a proper loss function as a loss where its conditional expected loss $L(p, q) := \mathcal{E}_{y\sim p}[l(y, q)]$ is minimized when $q = p$. This means that, the loss is minimum when the estimation of y is correct. Here, $y \in \{0, 1\}$ represents the class. It has been observed [24] that, it is possible to write an arbitrary loss function $\lambda(y, v)$ as a composition of a proper loss l and an invertible function ϕ^{-1}. The resulting composite function is called proper composite loss. It can be easily shown that square, exponential, or logistic losses are of this kind. However, in our case, each logistic function takes the following form:

$$f_i(\bar{x}) = P((y = 1)|\bar{x}, \theta) = \sigma(\theta^T \bar{x}) = \frac{1}{1 + \exp(-\theta^T \bar{x})}$$

We can write the loss function for equation 1 as $l(y, \bar{\sigma}(\theta^T \bar{x}))$, where $\bar{\sigma}$ denotes a real valued vector and each element $\bar{x}$ is generated by the logistic function σ. It is not obvious if an invertible function ϕ^{-1} exists for such a nonlinear loss function of $\bar{x}$. However, it is known that one can estimate this probability empirically using a density estimation technique [25]. We choose to use the simple histogram based technique with a small modification to address it's limitation with respect to continuity. These techniques are known to perform well with enough data. We first discretize each of the three probabilities and compute the empirical joint distribution of the these probabilities with the class label. Discretization is done by binning the probabilities in B number of bins. This discretization maps each probability value to a bin. The parameter B is chosen by cross validation. The bins corresponding to the probability values are represented by $f_i(\bar{x})_b$. The new probability of match is computed by using the following two entities:

$$c_0(\bar{x}) = \#(f_1(\bar{x})_b, f_2(\bar{x})_b, f_3(\bar{x})_b, 0)$$

$$c_1(\bar{x}) = \#(f_1(\bar{x})_b, f_2(\bar{x})_b, f_3(\bar{x})_b, 1)$$

If either $c_0(\bar{x})$ or $c_1(\bar{x})$ is zero we turn back to our base model for prediction as the counts are not trusted otherwise we return the following empirical distribution:

$$\hat{P}\left(m = 1|f_1(\bar{x}), f_2(\bar{x}), f_3(\bar{x})\right) = \frac{c_1(\bar{x})}{c_1(\bar{x}) + c_0(\bar{x})}$$

Algorithm 1 explains the details of this two-level method. Note that this binning strategy approximates the conditional probability of match given the three probabilities $f_1(\bar{x}), f_2(\bar{x})$, and $f_3(\bar{x})$. The the best way to combine these three probabilities, or equivalently their corresponding models, is through their joint distribution with the class label. This joint distribution is the optimal classifier in the sense that it minimizes the expected squared loss [26]. Furthermore note as B increases we use our base model more and more. The number of different combinations for $\#(f_1(\bar{x})_b, f_2(\bar{x})_b, f_3(\bar{x})_b, m)$ is $2 \times B^3$. Note that, we have found using this empirical density estimation techniques provide better AUC score compared to using another logistic model in the second layer. Using another logistic model in second layer will be very similar to a feed forward neural network. We have not reported the results with logistic model in the second layer.

Algorithm 1 Two-level Model

procedure TRAINTWOLEVEL(BaseCls, B, X, Y)
    BaseCls is the Base Classifier
    ▷ X is an $n \times d$ and Y is $n \times 3$
    ▷ $Y[\,,0]$ contains y
    ▷ $Y[\,,1]$ contains y'
    ▷ $Y[\,,2]$ contains m
    ▷ B is the number of bins

    Split (X, Y) randomly into two parts:
    (X_{train}, Y_{train}) and $(X_{validate}, Y_{validate})$
    $cls_0 \leftarrow LR.train\,(X_{train}, Y_{train}[,0])$
    $cls_1 \leftarrow LR.train\,(X_{train}, Y_{train}[,1])$
    $cls_2 \leftarrow BaseCls.train\,(X_{train}, Y_{train}[,2])$
    ▷ LR is logistic regression
    $C \leftarrow \{\}$ ▷ C will store empirical counts

    for $\bar{x}, y \in X_{validate}, Y_{validate}[\,,2]$ **do**
        $b_0 \leftarrow bin(cls_0.prob(\bar{x}), B)$
        $b_1 \leftarrow bin(cls_1.prob(\bar{x}), B)$
        $b_2 \leftarrow bin(cls_2.prob(\bar{x}), B)$
        $c \leftarrow (b_0, b_1, b_2, y)$
        if $c \in C$ **then**
            $C[c]{+}{+}$
        else
            $C[c] = 1$
        end if
    end for
end procedure

procedure BIN(p, B)
    return $floor(p \times B)$
end procedure

procedure ESTIMATEPROBABILITY($\bar{x}$)
    $b_0 \leftarrow bin(cls_0.prob(\bar{x}), B)$
    $b_1 \leftarrow bin(cls_1.prob(\bar{x}), B)$
    $b_2 \leftarrow bin(cls_2.prob(\bar{x}), B)$
    $c_0 \leftarrow (b_0, b_1, b_2, 0)$
    $c_1 \leftarrow (b_0, b_1, b_2, 1)$
    if $c_0 \in C$ and $c_1 \in C$ **then**
        return $\frac{C[c_1]}{C[c_0]+C[c_1]}$
    else
        return $cls_2.prob(\bar{x})$
    end if
end procedure

5 EXPERIMENTAL RESULTS

In order to validate our algorithm, we conducted several experiments with synthetic and real data sets.

5.1 Synthetic data generation

We draw a random real vector $\bar{x} \in R^n$ from unit cube. We then use two separate linear functions of form $f = a_0 + \Sigma_{i=1}^{i=n} a_i x_i$to generate two preference functions from two sides of the market. Let's assume that these are f_1 and f_2. The coefficients a_is are drawn randomly from two different Gaussian distributions with two different means and standard deviations. Then, we use two threshold functions of form $f > b$ to generate the values of y i.e. class labels for the preference function. Values of b are chosen such that we can obtain certain percentage of positive ($y = 1$) instances. Varying the values of b we can impose a degree of imbalance in the generated data. We define an imbalance parameter positive skew as ratio of number of positive instances and the total number of data points and call it μ. We control μ by choosing suitable values of threshold b. We then generate two kinds of functions combining f_1 and f_2. we combine two functions linearly $f_3 = c_1 f_1 + c_2 f_2$. We also define a label noise parameter $\nu_l \in \{0, 1\}$ which denotes the percentages of label that may be randomly flipped in a data set. Table 1 compares the AUC scores results on a 50 dimensional synthetic datasets of $100K$ points with 10% label noise.

Data	Metrics	L1-LR	TL
Syn $\nu_l = 0.00$ $\mu = 0.10$	AUC	0.91	0.93
	TPR	0.82	0.86
	TNR	0.83	0.86
Syn $\nu_l = 0.10$ $\mu = 0.10$	AUC	0.83	0.88
	TPR	0.74	0.82
	TNR	0.78	0.87
Real $\nu_l = 0.0$	AUC	0.87	0.89
	TPR	0.71	0.76
	TNR	0.83	0.84

TABLE 1
Comparison of 16 binned two-level model with L_1-LR with $\lambda = 0.1$.

Data	Metrics	RF	TL
Syn $\nu_l = 0.00$	AUC	0.91	0.93
	TPR	0.82	0.87
	TNR	0.83	0.85
Syn $\nu_l = 0.10$	AUC	0.82	0.89
	TPR	0.74	0.82
	TNR	0.77	0.87
Real $\nu_l = 0.0$	AUC	0.86	0.89
	TPR	0.71	0.76
	TNR	0.83	0.84

TABLE 2
Comparison of 16 binned two-level model with RF with 100 trees with 18 nodes.

5.2 Real data description

We obtain a real two-sided market data from one of the world's largest online labor market Elance.com. This data set is collected from six month's of interactions in Elance.com in 2013. At Elance.com clients post jobs and they can invite contractors. The contractors can bid for any job with or without an invite. Upon receiving job applications from several contractors for a posted job, a client can then award the job to his contractor of choice. Once the contractor accepts the awarded jobs, we can consider that a matching between the client and contractor has taken place. In our notations, invites can be considered as a signal or preference for the clients towards the contractors and a bid can be considered as a contractor's interest towards a client. An accept even indicates a match between the client and contractor. An award event does not have any special significance but it is just a state that comes before an accept state. In the data set, features have been obtained using various user interaction signals and content based signals from both clients and contractors. Each data point contains either a bid or an invite. The data set has around ten million such instances and it has 88 features. We use first three month's of data for training and validation and the next three month's data is used for testing.

5.3 Results

In table 1 we show that a two-level model with 16 bins gives 4% improvement in AUC score for no label noise and it gives 10% improvement when there is a 10% label noise compared to L_1-LR applied directly to the data to model matching. Note that the $f_1(\bar{x})$ and $f_2(\bar{x})$ are generated with $\mu = 0.35$ and $f_3(\bar{x}$ is generated with $\mu = 0.10$. The AUC score is improved by 2% for real data compared to l_1 logistic regression. It is to be noted that 2% real improvement can actually be responsible for huge amount of revenue for these sites. For synthetic data, two-level model gives 3-12$\%$ better true positive and true negative rates improvement compared to L_1-LR. It also improves those for real data but in a smaller percentage from what we observed in the synthetic data. One can argue that the L_1-LR being a generalized linear model is not the best model to capture the complex distribution of matching. Hence, we also use random forest as another alternative direct regression technique. Random forest is a powerful ensemble method and is known to be consistent [27]. In table 2, we show that two-level model has 4-10$\%$ AUC improvement and 2-10$\%$ true positive and negative rate improvement for the synthetic data. We also observe 2% AUC improvement for real data compared to random forest. Note, we cross validate to obtain the optimal λ in case of L_1-LR and we did a grid search to find the optimal number of nodes and trees for the random forest for our data sets. Overall the observation is that the two-sided model almost always improves not only the AUC score but also the TPR and TNR values in our experiments.

5.4 A theoretical question

There is an interesting empirical observation in this experiment. The following set relationship holds for any markets with two-sided preferences:

$$|\bar{x} : m = 1| \subseteq |\bar{x} : y = 1| \cup |\bar{x} : y' = 1|$$

These sets are the domains of the functions $f_1(\bar{x_1}), f_2(\bar{x_2})$ and $f_3(\bar{x_3})$ respectively. We can pose a theoretical question on this as follows: if this set relationship in the data domains always leads to a better AUC performance of two-level models compared to any direct regression algorithm with univariate loss function from the proper composite loss function family.

6 Conclusion

In this paper we introduce a novel two-level model algorithm that has a better AUC performance compared to direct applications of regression techniques such as L_1-LR or random forests for markets with two-sided preferences and provide a theoretical justification for the algorithm's better AUC performance. The results we obtained clearly demonstrate that the two-level algorithm, despite being quite simple can lead to substantially better AUC performance. This yields a better recommendation system for markets with two-sided preferences.

Acknowledgments

The authors would like to thank Dr. ChengXiang Zhai (czhai@cs.uiuc.edu) and Dr. Sayanti Banerjee (spbanerjee@scu.edu) for their comments on improving the manuscript.

References

[1] A. E. Roth and M. A. O. Sotomayor, *Two-sided matching: A study in game-theoretic modeling and analysis*. Cambridge University Press, 1992, no. 18.

[2] T. Economist, "The workforce in the cloud," The Economist, June 2013.

[3] P. Resnick and H. R. Varian, "Recommender systems," *Communications of the ACM*, vol. 40, no. 3, pp. 56–58, 1997.

[4] S.-I. Lee, H. Lee, P. Abbeel, and A. Y. Ng, "Efficient l˜ 1 regularized logistic regression," in *Proceedings of the National Conference on Artificial Intelligence*, vol. 21, no. 1. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006, p. 401.

[5] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[6] K. Wei, J. Huang, and S. Fu, "A survey of e-commerce recommender systems," in *Service Systems and Service Management, 2007 International Conference on*. IEEE, 2007, pp. 1–5.

[7] Y. Koren and R. Bell, "Advances in collaborative filtering," in *Recommender Systems Handbook*. Springer, 2011, pp. 145–186.

[8] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in *The adaptive web*. Springer, 2007, pp. 325–341.

[9] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[10] T. Zhang and V. S. Iyengar, "Recommender systems using linear classifiers," *The Journal of Machine Learning Research*, vol. 2, pp. 313–334, 2002.

[11] J. Weston, H. Yee, and R. J. Weiss, "Learning to rank recommendations with the k-order statistic loss," in *Proceedings of the 7th ACM Conference on Recommender Systems*, ser. RecSys '13. New York, NY, USA: ACM, 2013, pp. 245–248.

[12] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver, "Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering," in *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 2010, pp. 79–86.

[13] V. Petricek. (2013, Oct) Data science of love (eharmony). [Online]. Available: http://strataconf.com/stratany2013/public/schedule/detail/30229

[14] J. J. Horton, "The effects of subsidizing employer search," *Available at SSRN 2346486*, 2013.

[15] J. J. Horton, "Computer mediated matchmaking: Facilitating employer search and screening," Stern School of Business, Tech. Rep., 2013.

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

[17] L. Pizzato, T. Rej, T. Chung, I. Koprinska, and J. Kay, "Recon: a reciprocal recommender for online dating," in *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 2010, pp. 207–214.

[18] K. Koh, S.-J. Kim, and S. Boyd, "An interior-point method for large-scale l1-regularized logistic regression," *Journal of Machine Learning Research*, vol. 8, pp. 1519–1555, 2007.

[19] A. Y. Ng, "Feature selection, l 1 vs. l 2 regularization, and rotational invariance," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 78.

[20] P. Zhao and B. Yu, "On model selection consistency of lasso," *The Journal of Machine Learning Research*, vol. 7, pp. 2541–2563, 2006.

[21] W. Kotlowski, K. J. Dembczynski, and E. Huellermeier, "Bipartite ranking through minimization of univariate loss," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 1113–1120.

[22] J. Friedman, T. Hastie, R. Tibshirani *et al.*, "Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors)," *The annals of statistics*, vol. 28, no. 2, pp. 337–407, 2000.

[23] C. Scott *et al.*, "Calibrated asymmetric surrogate losses," *Electronic Journal of Statistics*, vol. 6, pp. 958–992, 2012.

[24] M. D. Reid and R. C. Williamson, "Composite binary losses," *The Journal of Machine Learning Research*, vol. 11, pp. 2387–2422, 2010.

[25] B. W. Silverman, *Density estimation for statistics and data analysis*. CRC press, 1986, vol. 26.

[26] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[27] G. Biau, L. Devroye, and G. Lugosi, "Consistency of random forests and other averaging classifiers," *The Journal of Machine Learning Research*, vol. 9, pp. 2015–2033, 2008.