Published in final edited form as:

Genet Epidemiol. 2012 February ; 36(2): 99–106. doi:10.1002/gepi.21608.

Bootstrap Aggregating of Alternating Decision Trees to Detect Sets of SNPs that Associate with Disease

Richard T. Guy^{1,4}, Peter Santiago^{2,3}, and Carl D. Langefeld¹

¹Department of Biostatistical Sciences, Division of Public Health Sciences, Center for Public Health Genomics, Wake Forest School of Medicine, Winston-Salem, NC

²Department of Biomedical Engineering, Wake Forest University Health Sciences, Winston-Salem, NC

³Virginia Tech – Wake Forest Biomedical Engineering and Sciences

Abstract

Complex genetic disorders are a result of a combination of genetic and non-genetic factors, all potentially interacting. Machine learning methods hold the potential to identify multi-locus and environmental associations thought to drive complex genetic traits. Decision trees, a popular machine learning technique, offer a computationally low complexity algorithm capable of detecting associated sets of SNPs of arbitrary size, including modern genome-wide SNP scans. However, interpretation of the importance of an individual SNP within these trees can present challenges.

We present a new decision tree algorithm denoted as Bagged Alternating Decision Trees (BADTrees) that is based on identifying common structural elements in a bootstrapped set of ADTrees. The algorithm is order nk^2 , where n is the number of SNPs considered and k is the number of SNPs in the tree constructed. Our simulation study suggests that BADTrees have higher power and lower type I error rates than ADTrees alone and comparable power with lower type I error rates compared to logistic regression. We illustrate the application of these data using simulated data as well as from the Lupus Large Association Study 1 (7822 SNPs in 3548 individuals). Our results suggest that BADTrees holds promise as a low computational order algorithm for detecting complex combinations of SNP and environmental factors associated with disease.

Keywords

Machine Learning; Genetic Association; Gene-Gene Interaction; Multi-locus Models

Introduction

The current paradigm is that complex genetic disorders are a result of combinations of genetic and non-genetic factors, many potentially interacting. Although there has been tremendous progress in the identification of genetic variation influencing complex genetic traits, there has been less success in the identification of multi-locus effects, gene-environment interactions and epistatic effects. Genetic epistasis is classically defined as a masking relationship between two loci: the effect of one is suppressed by another. Statistical

Corresponding author: Carl D. Langefeld, Medical Center Blvd., Winston-Salem, NC, 27157, Phone: 336-713-0013, clange@wfu.edu.

⁴Current affiliation: Department of Computer Science, University of Toronto, Toronto, ON, Canada

interaction occurs when the effect of one locus depends on the genotypes at other loci [Cordell, 2002; Moore and Williams, 2005].

In supervised machine learning, algorithms are developed that use existing data to “learn” how to predict complex data well. Examples include decision trees, support vector machines and many regression models. With the increased interest in multilocus models and genetic interactions, good machine learning algorithms may form a complementary approach to classic linear models in statistical genetics. A promising set of algorithms with linear computational complexity are the decision tree algorithms. Decision trees are decision structures (often binary) that iteratively subdivide the data in an attempt to classify each sample (e.g., case vs. control).

Alternating Decision Trees (ADTrees) form a subclass of decision trees that have been utilized for case-control genetic studies by [Liu et al., 2005] and [Fiaschi et al., 2009]. Two key features of ADTrees that differentiate them from typical, binary decision trees such as CART or C4.5 are: 1) ADTrees may produce nonbinary structures; and 2) traversing an ADTree to classify an individual requires following all possible paths in parallel rather than choosing only one path at each node. (See Figures 1 and 2.) ADTree classification uses a numerical score, with the sign of the score indicating the predicted class and the magnitude of the score providing a measure of confidence in prediction. Greater structural flexibility and graded predictions help ADTrees to have higher classification accuracy than a binary decision tree of the same size [Fruend and Mason, 1999; Pfahringer, Holmes, and Kirkby, 2001]. Another important advantage is that an ADTree can be created in time proportional to nk^2 for n genetic markers (or other variables) in the data set and k markers in the tree. In this paper, we utilize ADTrees in a way that encourages a small k , yielding an algorithm linear in the number of markers. Among the difficulties associated with using decision trees for case-control studies is that greedily built trees are vulnerable to noise, interpretation can be challenging, and at least a weak trend toward marginal effects must exist for an association to be detected. Herein, we present a solution to the first two problems and a brief discussion of the third. We also present a change to the ADTree algorithm to improve its power in the special case of genetic data.

While ADTrees have promise for more accurate classification compared to other decision tree algorithms, they still pose problems of interpretation and validation. In particular, how to interpret the significance of the position of a marker in an ADTree is not clear. We introduce bagging (Bootstrap Aggregating) for the detection and quantification of features in ADTrees. Bagging is a noise-reduction technique [Breiman, 1996] based on bootstrapped samples of the input data. An ADTree is built on each bootstrapped sample and we use bagging for feature detection by recording the most common features in the set of ADTree trees. Formally, a feature is defined as a set of predictors that forms a single rule or path through the ADTree.

Bagged ADTrees (BADTrees) provide a measure of feature importance: the percentage of the trees that include a feature. Two hypotheses lie at the core of the algorithm: (1) markers or sets of markers that are associated with a phenotype should be identified more often than unassociated markers by an ADTree that uses the best marker at each step; and (2) the markers that are most strongly associated with the phenotype in the data set will be associated with phenotype in the bootstrap samples more often than unassociated markers or sets of markers. Therefore, we propose an algorithm that identifies associated sets of markers and quantifies their importance by examining their prevalence in a set of trees built on the bootstrapped data.

The purpose of this study is to introduce the use of BADTrees for detecting genetic and environmental factors and their combinations that predict disease status. We compare BADTrees to the tests for single and multi-locus association using the most commonly used approach for case-control data, logistic regression. We demonstrate that the BADTrees algorithm is of low computational burden and can identify complex associations between disease status and genetic and non-genetic factors. The presentation of the approach uses single nucleotide polymorphisms (SNP) as the genetic markers but the approach is generalizable to other genetic markers (e.g., insertion/deletion, non-binary polymorphisms) and other variables (e.g., gender, admixture). The BADTrees algorithm is implemented in our analysis software, SNPLASH.

2. Materials and Methods

2.1 Algorithms

An alternating decision tree (ADTree, Figure 2) consists of alternating layers of decision and prediction nodes. In the figure, prediction nodes (oval) identify a SNP and contain two decision node descendents that partition the input into two sets. Decision nodes (square) contain a numeric score and have zero or more descendents.

Each iteration of the ADTree algorithm adds one prediction node and two decision nodes to the tree and reweights the individuals in the data set according to the current iteration's success in classifying them. For an iteration, a condition c and precondition p are chosen that minimize the value of

$$Z=2(\sqrt{W_+(p_j \wedge c_k)W_-(p_j \wedge c_k)} + \sqrt{W_+(p_j \wedge \neg c_k)W_-(p_j \wedge \neg c_k)}) + W(\neg p_j) \quad (1)$$

over all p_j and c_k . Values p_j and c_k are a precondition and condition, respectively. Conditions are Boolean expressions of the form “rs1234 > v” for a given SNP in the data set and value v that the SNP can take assuming an additive coding (0, 1, or 2 copies of the minor allele). A prediction node can be represented as a condition and a precondition consisting of all parent prediction nodes. Preconditions are conjunctions of conditions and negations of conditions. Each prediction node is the root of two subsequent decision nodes (i.e. children) that represent the two Boolean outcomes from the condition c and are given

weights $\frac{1}{2} \log \left(\frac{W_+(p \wedge c)+1}{W_-(p \wedge c)+1} \right)$ and $\frac{1}{2} \log \left(\frac{W_+(p \wedge \neg c)+1}{W_-(p \wedge \neg c)+1} \right)$ for the ‘true’ and ‘false’ result. Functions $W_+(p)$ and $W_-(p)$ represent the sum of all cases and controls, respectively, that satisfy the given conjunction of conditions while function $W(p)$ is the sum of the weights of all individuals satisfying p . In equation (1), the first term expresses the degree to which the condition and precondition separate the data for which the precondition is true (perfect separation would be a score of 0) [Schapire RE and Singer Y, 1999]. The second term penalizes a precondition that is not true for most samples. Initially, all individuals are given equal weight, but weights are updated according to whether the individual is correctly or incorrectly classified by the decision node most recently added. Weights are updated according to the equation $w := w_i e^{-R}$ where R is the score for the most recently added decision node that applies to individual i . If the individual fails to satisfy the precondition p then $R=0$ and the weight is unchanged.

Equation (1) effectively forces SNPs to conform to a dominant or recessive penetrance pattern: the choice between dominant and recessive is reflected in the condition values. In order to consider additive risk we alter the function to be minimized to:

$$\hat{z}(p, SNP) = 2 \sqrt{W_+^{AA}(p)W_-^{AA}(p)f_{AA} + W_+^{Aa}(p)W_-^{Aa}(p)f_{Aa} + W_+^{aa}(p)W_-^{aa}(p)f_{aa} + W(-p)}.$$

Here, the weights of individuals that satisfy precondition p , defined above, are summed for each possible genotype (homozygous dominant, heterozygous, homozygous recessive) and phenotype. The SNP s and precondition p that minimize $\hat{z}$ are then used to create a condition by minimizing z over all conditions that involve p and s . While the SNP s that minimizes $\hat{z}$ might differ from the SNP in condition c that minimizes z , we use s to create the condition in each step so as to maintain the form of an ADTree.

An individual is classified by following all paths through the tree in parallel and summing the scores for all decision nodes. The sign of the summed score classifies an individual as a predicted case or control [Fruend and Mason, 1999]. Pilot testing suggested that the magnitude of scores in individual nodes or sets of nodes was not necessarily informative for SNP selection. This is because scores increase deeper in the tree and decision node scores for noise SNPs are equivalent in magnitude to those for penetrant SNPs.

Bagging (Bootstrap AGGREGatING) is a noise-reduction technique that we propose to measure SNP importance in conjunction with the ADTree algorithm. Bagging creates a set of bootstrapped (sampling with replacement) samples and runs the ADTree algorithm on each set. Final classification is provided by a vote of the classification from each tree. For example, if 100 trees are constructed and an individual is classified as a case in more than 50 of them, that individual is classified as a case overall. Bagging increases the signal-to-noise ratio because effects that are consistent across bootstrap samples are found in a larger percentage of the population from which the samples are taken, while less frequent effects, unfortunately including rare variants, are found in a lower percentage of the bootstrap samples [Breiman, 1996]. Individuals are over- or underrepresented in the bootstrapped samples, which allows the greedy ADTree algorithm to build trees using SNPs that are suboptimal yet informative in the entire data set. We use bagging for feature detection by recording the most common features in the set of trees. Intuitively, a feature is defined as a set of interrelated SNPs that can be used to predict disease status. In trees, one definition of interrelatedness between a set of trees is a shared relationship in the tree structure. Thus, we define a feature in an ADTree as the SNPs in a path through the tree beginning at the root node and terminating at a leaf node. This definition best fits the concept of prediction nodes in an ADTree as a condition along with a precondition that corresponds to the path through the tree to that prediction node.

The BADTrees algorithm is as follows. One hundred bootstrapped data sets are constructed and an ADTree is built on each set. Features that appear in more than t trees, for suitable t , are recorded and the SNPs involved are removed from the data. If at least one feature was identified in t or more trees, the process is repeated with the reduced data set. If not, the algorithm ends. Features including the SNPs involved are returned with the number of trees in which they appear. Stringency of acceptance is controlled by the parameter t , and a study of the effect of t on power and type I error is presented in the Results.

2.3 Methods: Simulation studies

Bagged ADTrees and ADTrees are compared to logistic regression in a set of genetic models designed to capture a wide range of potential association models. In each case, we record the power to reject the null hypothesis of no association. For BADTrees and ADTrees, we include a test of the ability to detect a single SNP that is involved in a

multilocus association and a test of the ability to detect the entire multilocus association, where applicable.

Simulated data includes the following models which are described in the supplement: 3 models with a single associated SNP, 4 models with two associated SNPs, and 3 models with three associated SNPs. Single-locus models (i.e., allele frequency, penetrance function) were motivated in part by genes discovered in autoimmune disorder studies [e.g., Harley et al., 2008]. Multilocus models were based on combinations of common patterns of Mendelian inheritance and include a range of interactions, marginal effects, and penetrance patterns. Similar models were utilized to compare methods in [Miller et al., 2011]. All data sets include 200 SNPs, and all sets contain 2000 individuals split evenly between cases and controls. The simulated data do not contain missing data, but missing data is handled by the BADTrees algorithm.

2.4 Definitions of Power

For comparison, we include results from single-SNP logistic regression (LR) analysis, assuming an additive genetic model as implemented in our program SNPLASH available on the web at <http://www.phs.wfubmc.edu/public/bios/gene/downloads.cfm>. Significance for type I error and power measurements were defined at the $\alpha=0.05$ level. The type I error rate was estimated using 1000 iterations of 1000 cases and 1000 controls and randomly varying allele frequency. For multilocus models, we interpreted the single-locus test of association in two ways. First, we tested whether the null hypothesis of no association between a single SNP and phenotype was rejected for any single SNP in the multilocus model. Second, we tested whether the single-locus null hypothesis was rejected for all SNPs in the model. Also, for multilocus models, we include a test for the presence of interaction using LR; for multilocus models, the logistic regression did not attempt model building but assumed the proper number of SNPs. We also include the results using a single ADTree built with 10 SNPs. The optimal size of an ADTree is a matter of current research and depends on parameters that are specific to the data set including the expected number of SNPs that are associated with phenotype. Since 10 SNPs represents 5% of the data, the null hypothesis is rejected for 5% of the SNPs assuming that the data is under the null hypothesis, which is consistent with logistic regression on 200 SNPs. For multilocus models, we test the power to detect at least one SNP in the model and the power to detect the entire model in an association. Three definitions of a multilocus association in an ADTree are tested that grow progressively less stringent: (1) the SNPs must be in a single path (2) the SNPs must share a common root prediction node; and (3) the SNPs must all be in the ADTree. In total, power is defined as identifying any one of the three penetrant SNPs in the tree and as identifying all three penetrant SNPs in an association, and the latter is reported for 3 definitions of association.

The power to detect a multilocus model using BADTrees is compared using four definitions of detection. We report BADTrees' power to detect a single SNP out of a multilocus model, power to detect all SNPs in the model without reference to the relationship detected between them, and two measures that require all SNPs in the multilocus association to be identified in the same feature.

3 Results

The power and type I error for the null hypothesis of no association between a single SNP and phenotype are presented in Table 1. As expected, LR has the proper type I error rate for an individual SNP but requires an adjustment for multiple comparisons when scanning multiple SNPs (e.g., Bonferroni correction). Since the size of the ADTrees remains the same

we control for multiple correction in the BADTrees by adjusting the threshold of the number of trees in which a feature must be defined in order to count as a detected feature.

The type I error rate for a tree of size 10 is slightly lower than nominal levels for ADTrees (i.e., ADTrees are conservative). BADTrees are conservative due to the more stringent nature of the requirement for identification as a feature (i.e., repeated identification across bootstrap samples). The approach taken for this simulation study required that all ADTrees used in the BADTrees algorithm included 10 SNPs.

The power to detect a single ground truth SNP varied across the methods (Table I). Overall, logistic regression had the greatest statistical power in the absence of any multiple comparison adjustment. However, when multiple comparisons adjustment is applied, the BADTrees had the greatest statistical power across the three models. Power remains high and the type I error rate decreases if smaller trees (i.e., <10 SNPs) are used because the ground truth SNPs tended to be added early in the ADTrees iterations if they were added. Figure 3 displays the effect of the parameter t on power and type I error to detect a single SNP.

The results of the simulation study for the two and three-SNP models are presented in Tables II and III. Here, we present three tests of power: power to detect at least one SNP in the model, power to detect all SNPs in the model without reference to the relationship between those SNPs, and power to detect all SNPs in the model as a single unit. The third definition requires that the model be identified in a single rule (BADTrees) or using an LR test for interaction of the appropriate size. In these cases, ADTrees and BADTrees have greatly increased power to detect all or part of a multi-SNP model compared to LR with multiple comparisons adjustment. With one exception, BADTrees and ADTrees both have greater power to detect at least one SNP in the model, and they have greater power to detect all SNPs in the model and to correctly identify a relationship between the SNPs than corrected LR. In addition, the relative performance increases when three SNPs are included in a penetrance model as opposed to two. Bagging ADTrees universally increases power over ADTrees alone, particularly for the three SNP models. Figure 3 shows power and type I error for BADTrees on Model 4 for different thresholds of acceptance of a feature. Note that type I error decreases faster than power initially, but that a low threshold of acceptance is preferred if type I error is to be near 0.05.

A significant advantage of BADTrees is that it scales well to GWAS sized data sets. On a Xeon E5430 processor with 8 CPUs running at 2.66 GHz with 16 GB of RAM, BADTrees took 0:0:53 (hrs:min:sec) wall time or 0:6:12 CPU time per set of 200 SNPs and 2,000 individuals. Analysis of a simulated data set with 1,720 individuals and 308330 SNPs took 16:43:19 wall time or 127:19:32 CPU time with a memory footprint of 10.926 GB. All runs utilized 8 CPUs in parallel.

The International Consortium for Systemic Lupus Erythematosus Genetics (www.slegen.org) has published a genome-wide association study in Caucasian women using 317,501 SNPs and 720 systemic lupus erythematosus (SLE) cases and 2,337 controls and a similar female replication cohort of 1,846 cases and 1,825 controls genotyped on SNPs from the top 7,288 regions (Harley, et al., 2008). We analyzed these 7,288 SNP in the replication cohort using the BADTree algorithm and identified multiple associations in the HLA region and four additional regions (Table IV, Figure 2). Figure 2 is an ADTree built on the SLE data set, and table IV is the results of the BADTrees algorithm. In Figure 2, the oval nodes are prediction nodes while rectangles are decision nodes. Example features are (rs1270942) and (rs1801274, rs3843307, rs283733). Assuming that the genotype coding is 2 for AA, 3 for Aa, and 4 for aa where A is the major allele and a is the minor allele, then an

individual with genotypes (rs1270942=AA, rs9888739=Aa, rs1567190=Aa, rs1801274=aa) would be predicted to be a control ($-0.593 + 0.325 + 0.197 - 0.218 + 0.053 = -0.236$), since the sum < 0 . In this example the first several SNPs in the tree were also identified in BADTrees. Each of the regions identified by the ADTree algorithm have been confirmed as SLE-predisposing loci; albeit, the functional variant may remain unknown. Emphasizing that it certainly will not always be the case, none of the identified regions were false positives. Analysis of the 7,288 SNPs in the SLEGEN data set took 1:20:25 wall time or 10:06:52 CPU time with a total memory footprint of 0.66 GB.

4. Discussion

The current paradigm is that complex genetic traits are driven by genetic and non-genetic factors, all potentially interacting. Interestingly, the empirical evidence of this complexity has not been as pervasive as anticipated for multiple reasons. Are complex interactions and interrelationships among genetic and non-genetic factors less common than expected? Inherently, the power to detect interactions is generally believed to be lower than for main effects. In addition, tests of interactions often use SNPs that are only in linkage disequilibrium with the functionally interacting SNPs further reduces the statistical power. Finally, it is clear that better analytic methods are needed. The search for genetic variation that influences traits needs efficient methods that scale to a genome-wide level, consider multiple loci simultaneously, and test for modifiers and interactions among genetic and non-genetic factors. It is unlikely that any one approach will be optimal or near optimal for all these needs.

Compared to classic linear models, decision trees, ADTrees and BADTrees appear to be interesting complementary tools for elucidating complicated multilocus relationships. Relative to classic decision trees, ADTrees have higher classification accuracy for a comparably sized tree [Fruend and Mason, 1999]. The change to the objective function $\hat{z}$ introduced here improves ADTrees' power to detect sets of SNPs that are associated with disease status. In addition, we introduced bagging to the ADTree method to increase the robustness of these multilocus methods, as the main advantage of bagging is increased robustness in the face of noisy data [Breiman, 1996]. Interestingly, the more stringent BADTrees power definition, which requires there to be no extraneous SNPs in the final model, performs better as the model complexity increases. In addition, both ADTrees and BADTrees allow for confirmatory analysis; BADTrees can be built on a test data set and used to classify individuals in a confirmatory set. Classification accuracy can be compared using permutation and cross-validation methods to generalize the BADTree to another population.

4.1 Complementary Methods

There are a number of useful complementary algorithms to Bagged ADTrees for detecting complex relationships among predictors and traits. A general comparison of these methods is beyond the scope of this paper but some general comments are in order.

Generalized linear models provide a robust set of methods with well understood statistical properties. However, these models test a null hypothesis for each model individually, and correct interpretation of significance requires the addition of an appropriate multiple comparisons adjustment. Such adjustments can significantly reduce the effective power of the test compared to ADTrees and BADTrees. ADTrees and BADTrees approach the problem of multiple testing by capping the size of the tree or trees at a size that does not scale with the size of the data set. Optimal selection methods for the size of the tree used by BADTrees is the subject of further research and should incorporate knowledge of the expected complexity of potentially associated sets of SNPs. Our study indicates that the

BADTrees algorithm has greater power than logistic regression to detect all SNPs in an interacting pair or larger set if marginal associations exist. In higher order models, a subset of the SNPs often conveys nearly as much risk as the entire set of SNPs, and BADTrees may fail to identify the remaining modest risk factors. In models with no marginal effects where risk is defined solely by an interaction, BADTrees will fail to reliably identify these SNPs, which is caused by the algorithm's reliance on a modest marginal effects in at least a subset of SNPs. It is not clear how often in complex diseases the risk allele frequency and effect size are so balanced as to yield a model with no evidence of an association with any of the SNPs involved in the interaction.

Machine learning approaches provide a complementary set of tools to classic statistical techniques. Another resampling approach for the improvement of decision tree performance is found in the Random Forest algorithm [Breiman, 2001; Bureau et al., 2005]. The Random Forest approach uses random sub-sampling of individuals and SNPs rather than bootstrapping only individuals to create many data sets on which trees might be built. Multifactor dimensionality reduction (MDR) [Ritchie, Hahn, and Moore, 2003] uses cross-validation to generate a measure of feature confidence based on the hypothesis that a significant feature will more likely be found consistently across multiple subsets of the data. While a full comparison of these methods to BADTrees is not possible in this paper and each method has its strengths, BADTrees has a lower order runtime (linear in the number of SNPs if tree size is fixed as opposed to quadratic for Random Forest and MDR) and allows for the detection of sets of associated SNPs of arbitrary size. Analysis of 100 SNPs on 2000 individuals took 3:40:41 for RF, 0:10:26 for MDR, and 0:00:21 for BADTrees. However, doubling the number of SNPs to 200 gave runtimes of 14:16:58 seconds for RF, 1:22:27 seconds for MDR, and 0:00:53 seconds for BADTree. The Random Forrest algorithm was as implemented in FORTRAN 77 [Bureau et al., 2005] and MDR was as implemented in Java [Greene et al., 2010].

Admixture and population structure are particularly important sources of confounding in genetic association studies. Classically, estimates via admixture proportions or sources of variation (e.g., principal components) have been included as covariates in linear models. Although the optimal approach for including continuous covariates into ADTrees and BADTrees applications is a matter of current research, simple approaches are immediately available. For example, the population structure estimates can be partitioned into groups via cluster analysis, binning techniques [Dougherty, Kohavi, and Sahami; 1995] or visual inspection and these variables forced into the ADTree. The SNPs are allowed to enter into the ADTree only after adjustment for these confounders. More generally, other covariates can be handled in a similar fashion with or without forced inclusion.

4.2 Limitations

Linkage disequilibrium represents a limitation of the current method. Two SNPs that are in linkage disequilibrium with a functional variant will often be identified in disjoint trees in the set of ADTrees, falsely diluting the influence of that region. Although LD trimming may help, we are exploring expanding the concept of a feature to include SNPs that are in LD.

Another limitation is the philosophically important consideration that BADTrees relies on modest marginal effects to construct ADTree learners. By relying on marginal effects, the algorithm gives up power to detect sets of SNPs that are interacting or in an epistatic relationship but have no marginal effects. In exchange, the algorithm provides a lower computational complexity. While the importance of detecting epistatic relationships and pure interactions has been well presented [Moore and Williams, 2005; Thornton-Wells, Moore, and Haines, 2004], low complexity methods that focus on identifying sets of SNPs that include at least one SNP with a marginal effect are important.

The current version of the BADTrees removes features and not just SNPs. A feature can be a cluster of SNPs, potentially interacting. However, there are other strategies to detect interaction not considered here that merit further study.

Finally, resampling methods may not be the optimal approach for rare variants. Because each individual observation with the rare variant will have high influence and because resampling can generate samples that highly vary in the frequency of the rare polymorphism. Thus, bagging with voting may have higher variation in inference. We believe this is not limited to BADTrees and merits further study.

In conclusion, Bagged ADTrees presents a novel method for detecting associations in case-control studies. BADTrees has low computational complexity and the ability to identify higher order sets of SNPs that associate with disease. Of particular importance is the ability of the bagging algorithm to increase the robustness of the detected associations.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

We thank Haiyong Xu, Miranda C. Marion, Laurie P. Russell, Mary E. Comeau, and David John for their helpful comments. We acknowledge the National Institutes of Health (NIH) grants (R01 AR057106-01, R01 HL090567-01, R01 NS36695-10A2, R01 AR043274, and PO1 AR049084). The International SLE Genetics Consortium (SLEGEN) was funded by the Alliance for Lupus Research and the NIH (P01 AI083194-01). In particular, we thank: Drs. Marta E. Alarcon, Lindsey A. Criswell, John B. Harley, Chaim O. Jacob, Robert P. Kimberly, Kathy Moser, Betty P. Tsao and Timothy J. Vyse. Computing resources and funding for RTG were provided by the Wake Forest School of Medicine, Center for Public Health Genomics.

References

- Breiman L. Bagging Predictors. *Machine Learning*. 1996; 24:123–140.
- Bureau A, Dupuis J, Falls K, Lunetta K, Hayward H, Keith T, Eerdekewegh PV. Identifying SNPs predictive of phenotype using random forests. *Genet Epidemiol*. 2005; 28:171–182. [PubMed: 15593090]
- Cordel HJ. Epistasis: what it means, what it doesn't mean, and statistical methods to detect it in humans. *Hum Mol Genet*. 2002; 11:2463–2468. [PubMed: 12351582]
- Dougherty J, Kohavi R, Sahami M. Supervised and Unsupervised Discretization of Continuous Features. *Proc Int Conf Mach Learn*. 1995:194–202.
- Fiaschi, L.; Garibaldi, JM.; Krasnogor, N. A framework for the application of decision trees to the analysis of SNPs data. *Proc. 6th Annual IEEE conf. on Computational Intelligence in Bioinformatics and Computational Biology*; 2009. p. 106-113.
- Freund, Y.; Mason, L. *Machine Learning: Proceedings of the Sixteenth International Conference*. New York: Morgan Kaufmann; 1999. The alternating decision tree algorithm; p. 124-133.
- Greene CS, Himmelstein DS, Nelson HH, Kelsey KT, Williams SM, Andrew AS, Karagas MR, Moore JH. Enabling personal genomics with an explicit test of epistasis. *Pac Symp Biocomput*. 2010:327–36. [PubMed: 19908385]
- Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The Weka data mining software: An update. *SIGKDD Explorations*. 2009:11.
- Harley JB, Alarcon-Riquelme ME, Criswell LA, Jacob CO, Kimberly RP, Moser KL, Tsao BP, Vyse TJ, Langefeld CD. Genome-wide association scan in women with systemic lupus erythematosus identifies susceptibility variants in ITGAM, PXX, KIAA1542, and other loci. *Nat Genet*. 2008; 40:204–210. [PubMed: 18204446]
- Liu KY, Lin J, Zhou X, Wong ST. Boosting alternating decision trees modeling of disease trait information. *BMC Genet*. 2005; 6(Suppl 1):S132–S138. [PubMed: 16451591]

- Miller DJ, Zhang Y, Yu G, Liu Y, Chen L, Langefeld CD, Herrington D, Wang Y. An algorithm for learning maximum entropy probability models of disease risk that efficiently searches and sparingly encodes multilocus genomic interactions. *Bioinformatics*. 2009; 25:2478–2485. [PubMed: 19608708]
- Moore JH, Williams SM. Traversing the conceptual divide between biological and statistical epistasis: system biology and a more modern synthesis. *Bioessays*. 2005; 27:637–646. [PubMed: 15892116]
- Pfahring B, Holmes G, Kirkby R. Optimizing the induction of alternating decision trees. *Proc Fifth Pacific-Asian Conf on Adv Knowledge Discovery and Data Mining (PAKDD2001)*. 2001:477–487.
- Ritchie MD, Hahn LW, Moore JH. Power of multifactor dimensionality reduction for detecting gene-gene interactions in the presence of genotyping error, missing data, phenocopy, and genetic heterogeneity. *Genet Epidemiol*. 2003; 24:150–157.
- Schapire RE, Singer Y. Improved Boosting Algorithms Using Confidence-rated Predictions. *Machine Learning*. 1999; 37:297–336.
- Thornton-Wells TA, Moore JH, Haines JL. Genetics, statistics and human disease: analytical retooling for complexity. *Trends in Genetics*. 2004; 20:640–647. [PubMed: 15522460]

Decision Node: In an ADTree, a decision node holds a numeric value that is summed when classifying an individual.

Edge: In a tree, an edge is a connection between two nodes. They are directed and always point away from the root node.

Feature (in ADTree): In this paper, a feature is defined as a set of SNPs that forms a path from root node to leaf node.

Ground truth: In our simulation studies, between one and three SNPs out of 200 were combined with a sporadic rate to create the phenotype. The SNPs used to create phenotype are called ground truth.

Leaf Node: A node with no outgoing edges.

Node: A basic element in the tree. Nodes are connected with ordered edges.

Prediction Node: In an ADTree, a prediction node holds a condition that can be evaluated as either true or false.

Root Node: The starting node and only node in a tree with no incoming edges.

Figure 1.
Decision Tree Terminology

Figure 2.

A sample ADTree created using the SLEGEN data described in the Results section and visualized using the Weka Machine Learning Toolbox [Hall et. al., 2009]. Oval nodes are prediction nodes while rectangles are decision nodes. In this example, an individual with genotype (rs1270942=AA, rs9888739=Aa, rs1567190=Aa, rs1801274=aa) would be $(-0.593 + 0.325 + 0.197 - 0.218 + 0.053 = -0.236)$, a control. The coding in use is 2 for AA, 3 for Aa, and 4 for aa where A is the major allele and a is the minor allele. Example features in the ADTree are (rs1270942) and (rs1801274, rs3843307, rs283733).

Figure 3.

A comparison of BADTrees' power to detect a single SNP that is associated with phenotype under a dominant model for various acceptance thresholds for a path. All ADTrees used by BADTrees contained 10 SNPs.

Table I

Type I error and power for logistic regression, ADTrees, and Bagged ADTrees over models with a single ground truth SNP.

	Type 1 Error	Power		
Algorithm	Null Model	Model 1 (Dominant) ^I	Model 2 (Additive) ^I	Model 3 (Recessive) ^I
Logistic Regression (Bonferroni correction) ²	0.050	0.097	0.093	0.124
ADTrees ³	0.033	0.346	0.397	0.615
Bagged ADTree (1) ⁴	0.041	0.433	0.511	0.786
Bagged ADTree (2) ⁵	0.011	0.289	0.342	0.540

^IThe following are operating definitions for a dominant, additive, and recessive risk, which are defined in terms of the minor allele. Dominant risk is defined as risk of disease (r.o.d.) = 0.50 for individuals with genotypes aa, aA, and AA (sporadic, random disease rate of 0.40.) Recessive risk is defined as r.o.d = 0.50 for individuals with genotype aa. Additive risk implies that risk is 0.05 for AA, 0.1363 for Aa or aA, and 0.2435 for aa.

²Power for Logistic Regression after a Bonferroni correction (= 0.05/200).

³Power for ADTrees was defined as identifying the penetrant SNP in the tree.

⁴Power for BADTrees (1) was defined as identifying the SNP in a feature using definition (1), i.e. the SNP must be in a feature but the feature may contain other SNPs.

⁵Power for BADTrees (2) was defined as identifying the SNP as a feature using definition (2), i.e., the SNP must be in a feature that contains no other SNPs.

Table II

Power for logistic regression (LR), ADTrees, and BADTrees over models with two ground truth SNPs.

Algorithm	Model 4 (2-SNP) ¹	Model 5 (2-SNP) ²	Model 6 (2-SNP) ³	Model 7 (2-SNP) ⁴
LR – single SNP (Bonferroni corrected)	> 0.999	0.542	0.161	0.046
ADT ⁵ - at least one	> 0.999	0.335	0.536	0.342
BADTrees (1) ⁶ - at least one	> 0.999	0.400	0.673	0.472
LR – both SNPs (Bonferroni corrected)	> 0.999	0.005	0.003	<0.001
ADT - both (same rule)	0.576	0.061	0.320	0.056
ADT - both (same root)	0.576	0.061	0.320	0.056
BADTrees (1) – both in same tree	> 0.999	0.061	0.273	0.071
LR test for interaction (Bonferroni corrected)	<0.001	0.003	0.031	0.000
ADT - both (same tree)	> 0.999	0.066	0.333	0.059
BADTrees (1) - all in same rule	0.945	0.173	0.581	0.455
BADTrees (2) ⁷ - all in same rule	>0.999	0.774	0.841	0.718

¹Penetrance of 0.6 for at least one copy of major allele in each locus, MAFs = 0.25, sporadic rate of 0.05. See Table S.II for more information.

²Penetrance defined in Table S.II. MAFs = 0.20 and sporadic rate of 0.10.

³Penetrance of 0.35 if possess at least one copy of minor allele at each locus and 0.70 if 2 copies at each locus. MAFs = 0.15 and 0.25, and sporadic rate of 0.10.

⁴Penetrance of 0.20 if genotype is in the set (AAbb, AaBb, aaBB). MAFs = 0.20 and 0.30 with no sporadic rate.

⁵Power for ADTrees was defined as identifying at least one of the two SNPs in a tree (at least one) or identifying both SNPs in the tree in a single path (same rule), in a common subtree (same head) or simply both in the tree (same tree).

⁶Power for BADTrees (1) was defined as identifying either SNP in some path in the tree (at least one), identifying both SNPs in at least one path, and identifying both SNPs in the same path (all in same rule).

⁷BADTrees (2) measures power to identify both SNPs in a path that contains only the two ground truth SNPs and ends in a leaf node (Defn. 2 in the text).

Table III

Power for logistic regression, ADTrees, and BADTrees over models with three ground truth SNPs. Power was defined for all tests as in Table II.

Algorithms	Model 8 (3-SNP) ¹	Model 9 (3-SNP) ²	Model 10 (3-SNP) ³
LR – at least one (Bonferroni corrected.)	0.340	0.558	0.537
ADT - at least one	0.999	0.912	0.876
BADTrees (1) - at least one	>0.999	0.972	>0.999
LR – all three (bon)	<0.001	0.010	<0.001
ADT - all three (same rule)	0.215	0.790	0.071
ADT - all three (same root)	0.215	0.790	0.071
BADTrees (1) - all in each tree	0.670	0.738	0.488
LR – interaction (Bonferroni corrected.)	0.033	0.019	0.031
ADT - all three (same tree)	0.312	0.809	0.239
BADTrees (1) - all in same rule	0.140	0.768	0.120
BADTrees (2) - all in same rule	0.619	0.892	0.558

¹Penetrance of 1.00 if homozygous recessive at SNP 1 (MAF = 0.25) and heterozygous recessive at either of SNP 2 or 3 (MAFS = 0.20) with sporadic rate of 0.10.

²Penetrance that is ordinal in all tree SNPs (see S.IV) with MAFS = 0.40, 0.25, and 0.25, and a sporadic rate of 0.10.

³Penetrance that is additive in all three SNPs (Table S.V) with MAFS = 0.25, 0.20, and 0.20, and a sporadic rate of 0.10.

Table IV

Single-SNP associations identified by the Bagged ADTree (BADTree) algorithm. SNPs rs7775397, rs3132580, and rs1517352 are functional polymorphisms, with rs7775397 causing mis-sense leading to non-conservative change, rs3132580 causing mis-sense in a splicing region leading to the abolition of a protein domain, and rs1517352 is an intronic enhancer. SNPs in the table from rs2517403 down are from the second iteration of the BADTrees algorithm, meaning that they were identified only after the first seven SNPs were identified and removed. Multilocus associations were also identified between rs3131379 (in HLA region) and rs11592314 in 11% of trees and between rs4548893 and rs12537284 in 17% of trees.

% Trees	Gene	SNP	Chromosome	Position (kb)
84	IRF5/TNPO3 Region	rs12537284	7	128.717
62	HLA Region	rs3131379	6	31.829
43	ITGAM	rs4548893	16	31.272
27	IRF5 region	rs729302	7	128.569
25	STAT4	rs1517352	2	191.931
19	HLA Region	rs7775397	6	32.210
19	HLA Region	rs1270942	6	32.027
56 (second iteration)	HLA Region	rs2517403	6	31.061
45 (second iteration)		rs569269	6	12.618
24 (second iteration)	HLA Region	rs7758736	6	32.682
19 (second iteration)	HLA Region	rs3132580	6	30.910