SNPboost: interaction analysis and risk prediction on GWA data

Ingrid Brænne [1,2,3⋆], Jeanette Erdmann[2,3], and
Amir Madany Mamlouk[1,3]

1. Institute for Neuro- and Bioinformatics,
2. Medizinische Klinik II
3. Graduate School for Computing in Medicine and Life Sciences
University of Lübeck,
Ratzeburger Allee 160, 23562 Lübeck, Germany
{braenne,madany}@inb.uni-luebeck.de
http://www.inb.uni-luebeck.de

Abstract. Genome-wide association (GWA) studies, which typically aim to identify single nucleotide polymorphisms (SNPs) associated with a disease, yield large amounts of high-dimensional data. GWA studies have been successful in identifying single SNPs associated with complex diseases. However, so far, most of the identified associations do only have a limited impact on risk prediction. Recent studies applying SVMs have been successful in improving the risk prediction for Type I and II diabetes, however, a drawback is the poor interpretability of the classifier. Training the SVM only on a subset of SNPs would imply a preselection, typically by the p-values. Especially for complex diseases, this might not be the optimal selection strategy. In this work, we propose an extension of Adaboost for GWA data, the so-called SNPboost. In order to improve classification, SNPboost successively selects a subset of SNPs. On real GWA data (German MI family study II), SNPboost outperformed linear SVM and further improved the performance of a non-linear SVM when used as a preselector. Finally, we motivate that the selected SNPs can be put into a biological context.

Keywords: Genome-wide association, risk prediction, SNP-SNP interaction

1 Introduction

Genome-wide association (GWA) studies, which typically aim to identify single nucleotide polymorphisms (SNPs) associated with a disease, yield large amounts of high-dimensional data. During the last decade there has become a growing body of studies- mainly focusing on single-SNP statistics (p-values) - that have identified genetic loci (SNPs) associated to common complex diseases such as diabetes [14], myocardial infarction [11, 2], and Crohn's disease [10]. However,

⋆ corresponding author

so far these findings have only limited impact on risk assessment and clinical treatment [8, 7].

Complex diseases are caused by a variety of genetic factors. These factors, e.g. SNPs, may interact positively or negatively to increase or reduce the effect of the individual factors; indeed, an appreciable disease effect may only come about through such an interaction. Studies that focus on single locus effects alone are thus not likely to reveal the more complex genetic mechanisms underlying multifactorial traits [14, 15, 9]. Since, so far, most of the identified genetic variants have only a limited effect on disease risk, it suggests itself to analyze several SNPs simultaneously.

A prominent algorithm for classification accounting multiple factors is the so-called Support Vector Machine (SVM) [5, 6]. Recent studies using SVMs have been successful in improving the risk prediction for Type I and II diabetes [14, 16, 1]. However, the resulting classifier is using all SNPs, making it hard to interpret the resulting classifier in a biological context, especially for hundreds of thousands of SNPs. In order to allow a better interpretation of the results a feature selection approach is required. An intuitive way is to choose SNPs that are individually associated with the disease. But, clearly, SNPs might be missed that only show an effect in interaction with others.

Boosting algorithms like Adaboost might be an good alternative [4, 3]. The main idea of Boosting is to combine several weak classifiers to one strong classifier. Weak classifiers, i.e. SNPs, are added one after another in order to gain a set of classifiers that together boost the classification. With this selection strategy, one can control the number of SNPs without having to preselect a subset of SNPs and possible SNP-SNP interactions might be found between the selected SNPs.

In this work, we propose a variation of the Adaboost algorithm for an application to GWA data. The algorithm is evaluated on the german MI family study II GWA data set [2].

2 Methods and Data

2.1 Support Vector Machine (SVM)

Support vector machines (SVM) aim to determine the hyperplane that separates two given classes with maximum margin [13]. It has been applied to a broad range of classification problems and is one of the standard benchmark methods. In this work, we train the SVM on an increasing number of SNPs, where the SNP subset is selected based on the single SNP p-values. In order to measure the classification performance for each subset, we train a SVM on the genotype data of the selected SNPs of the training set. For the linear and gaussian SVM the softness of the margin and the kernel width for the gaussian SVM is adjusted by a 10-fold cross-validation on the training set.

2.2 Adaboost

As SNPboost is derived from one of the most popular boosting algorithms, Adaboost [4, 3], we will provide in the following a brief sketch of the algorithm.

In Adaboost, the classifiers are combined such that classification errors of the first single classifiers are compensated as good as possible by the second classifier etc.. Thus, the selection of the next classifier is biased in favor of the previously misclassified datapoints. This is done by applying weights to the datapoints, where the weights are updated after each step. More specific, the weights of the misclassified datapoints are increased and correspondingly decreased for the correctly classified datapoints. Consequently, step by step, the classifier gets stronger by subsequently adding weak classifiers that optimize the performance of the combined Adaboost classifier.

Let $X = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$, $\mathbf{x}_i \in \mathbb{R}^D$ be a set of given data samples and $Y = (y_1, \ldots, y_N)$, $y_i \in \{1, -1\}$ the corresponding class information. Furthermore, we define a set of weights $\mathbf{w} = (\mathbf{w}_1, \ldots, \mathbf{w}_N)$ with the two constraints $\mathbf{w} \geq 0$ and $\sum_{i=1}^{N} \mathbf{w}_i = 1$, and T be the number of weak classifiers combined to the strong classifier. Finally, let $H = (h_1, \ldots, h_M)$ describe the set of weak classifiers to choose from and $L^{h_j} = (l_1^{h_j}, ..., l_N^{h_j})$ be the classification of X by each classifier h_j. Then, Adaboost combines a desired number of T classifiers h_j to a strong new classifier according to Algorithm 1.

Algorithm 1: Basic Adaboost Algorithm

Input: Training data X, labels Y, and a set of naive classifier H
Output: List of T chosen classifier $\eta = (\eta_1, \ldots, \eta_T)$ and their significance $\alpha = (\alpha_1, \ldots, \alpha_T)$,
foreach *epoch* $t = 1, \ldots, T$ **do**

1. Find the classifier η_t that minimizes the training error ϵ

$$\eta_t = \arg\min_{h_j \in H} \epsilon_j \tag{1}$$

with

$$\epsilon_j = \sum_{i=1}^{N} |l_i^{h_j} - y_i| \quad w_i(t) \tag{2}$$

2. Choose α_t with

$$\alpha_t = 1/2 \; ln(\frac{1 - \epsilon_t}{\epsilon_t}) \tag{3}$$

3. Update the weights in $\mathbf{w}$ with a normalization constant Z_t to

$$w_i(t+1) = \frac{w_i(t)}{Z_t} \cdot \begin{cases} e^{-\alpha_t} & \text{if } l_i^{\eta_t} = y_i \\ & \text{otherwise.} \end{cases} \tag{4}$$

end
Calculate final classification with $L^{ada}(x) = sign(\sum_{t=1}^{T} \alpha_t \cdot l_x^{\eta_t})$

2.3 SNPboost: Adaboost for GWA Data

To use Adaboost with GWA data, we have to define the set of weak classifiers H. In this work, we obtain H by deriving a set of 6 classifiers out of each SNP: Each of the given SNPs consists of three discrete states (**AA**, **AB**, **BB**), which denote the possible homozygot and heterozygot genotypes an individuum can express for this location. We use these three states to derive six naive classifiers that indicate a subject expressing one of the three states or its negation (a variant might just as well be protective in its effect). Hence, let p be the predicted genotype for all D SNPs with $p = p_1, \ldots, p_M$ and $M = D \cdot 6$. Given the genotype $g_c(i)$ of the individual i we get the training error ϵ_c by:

$$\epsilon_c = \sum_{i=1}^{N} [1 - [p_c(i) \neq g_c(i)]] \quad w_t(i). \tag{5}$$

2.4 Data

We applied the SNPboost algorithm to the German MI family study II [2] in order to evaluate the performance of the algorithm. SNPs with a missing rate of ≤ 0.01, minor allele frequency of ≤ 0.01, and p-value $\leq 10^{-4}$ for deviation from Hardy-Weinberg equilibrium were excluded. After quality filtering we pruned the data for Linkage Disequilibrium (LD).The total number of SNPs left for analysis is $D = 127370$ SNPs for a total of 2032 individuals with 1222 controls and 810 cases. We filled up the remaining missings with imputed values.

For training and testing, the data was randomly divided in a training and test set with equal sample size for the cases and the controls with 405 individuals each.

3 Results and Discussion

We evaluated the performance of the SNPboost algorithm and the SVM by means of the receiver operator characteristic (ROC) obtained on the test set. We tested SNPboost against linear SVMs with p-values as a preselection method, then we evaluated the gain of using SNPboost as a feature selection method for a non-linear SVM, and finally we examined the selected SNPs for biological relevance.

3.1 Linear Classification

As shown in Figure (1), both algorithms yield a peak performance for small number of SNPs. Subsequently, the performance decrease with additional SNPs. This decrease is likely caused by overfitting of the algorithms: The more features used, the more likely the trained classifiers are adapted to the training set and hence no longer describe the test set appropriately. The maximum AUC (area under the (ROC) curve) for the SNPboost algorithm is 0.76 with a total of 5

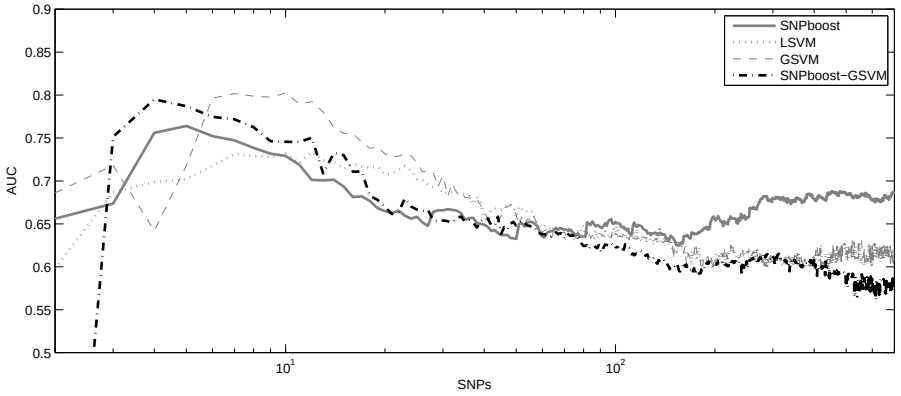


Fig. 1. Classification performance of linear and non-linear SVM with SNPboost selected SNPs vs p-value selected SNPs. A logarithmic (base 10) scale is used for the X-axis.

SNPs. For the LSVM the performance yield 0.73 for a total number of 11 SNPs. Whereas the performance of the LSVM remain weak for further increasing numbers of SNPs, the performance of the SNPboost algorithm recover. Hence, the SNP selection is no longer specific for the training set. The overall performance of the SNPboost algorithm clearly outranges the LSVM. Whereas the mean performance of the LSVM is 0.62 the mean performance of the SNPboost algorithm yield 0.67.

With SNPboost being a linear classifier we first evaluated the performance of the SNPboost algorithm and the linear kernel SVM (LSVM). The selection done by the SNPboost algorithm seems to be the better choice compared to the selection by the p-values. This might be due to the fact that the SNPboost algorithm selects SNPs one at a time in order to bit by bit increases the classification and might thus better fit together.

3.2 Non-linear Classification

The previous results indicated that the SNPs selected by the SNPboost algorithm might be more appropriate for combined classification than the SNPs selected due to the single significance values. Hence, combining the SNP selection strategy of the SNPboost algorithm with a more powerful classifier might improve the classification performance.

In order to test whether the classification performance can be further increased by applying a non-linear classifier, we trained a gaussian kernel SVM (GSVM) on the SNPs selected by the SNPboost algorithm and compared the results with the performance on the p-value selected SNPs.

The dash-dotted line in Figure 1 shows the performance of the gaussian classifiers with SNPboost as a preselector, while the dashed line depicts the

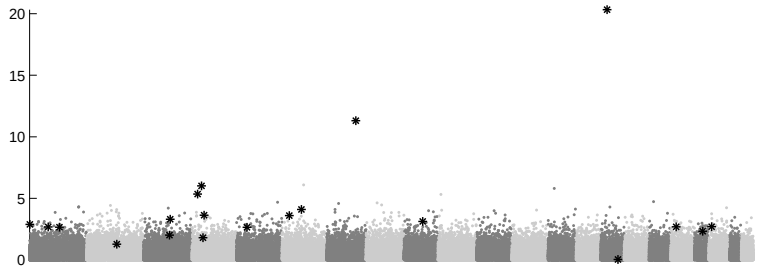


Fig. 2. Single p-values for all SNPs and the 20 first selected SNPs by the SNPboost algorithm. The SNPboost selected SNPs are shown as black dots.

performance of the gaussian kernal SVM with p-values as a feature selector. Regardless of the selection strategies, the performance primarily increases by applying GSVM. For the p-value selection strategy, the GSVM yiel better performance than the LSVM for small number of SNPs but the performances align for larger numbers of SNPs. Analogous to the p-values selection, for small number of SNPs, the GSVM on SNPboost selected SNPs improve the performance compared to the SNPboost algorithm. As for the performance of the SNPboost algorithm, also the performance of the GSVM decreases with a larger number of SNPs. However, in contrast to the SNPboost algorithm, for large number of SNPs, the performance does not recover but even drops below the performance of the LSVM and GSVM on p-value selected SNPs. This is probably due to the fact that the SNPboost algorithm selects a set of SNPs that fit well together. The GSVM further optimizes the classification with these matching SNPs and thus the classifier is too specific to the training data set.

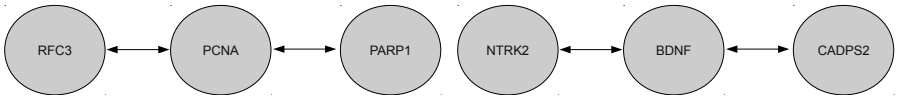
The maximum performance of the GSVM is 0.80 for both the selection strategies. The peak performance is gained with 4 and 10 SNPs for the SNPboost selection and p-values selection respectively.

3.3 Interaction Analysis

The main advantage of SNPboost is the selection strategy: If a weak classifier positively interacts with a previous selected one, this weak classifier might be chosen since an interaction might improve the classification. Thus, possible interactions might be found within the selected SNPs. Hence, it might be of value to further analyze these SNPs.

Figure (2) shows the p-values of the 20 first selected SNPs by the SNPboost algorithm. Of the top ten p-value SNPs, three are selected by the SNPboost algorithm. While the first one corresponds to the strongest single classifier, which in this case is the SNP with the highest p-value, all upcoming SNPs are selected due to their interaction with the first classifiers, independent of their p-values.

First we assigned the p-value selected SNPs to their corresponding genes. 14 SNPs were found within genes. Two of the 14 genes can be linked through



(a) SNPboost (b) p-value

Fig. 3. Gene interaction, of the 20 first selected SNPs two genes can be linked through an additional gene for both selection strategies

an additional gene. As shown in Figure (3a) RFC3 interacts with PARP1 via PCNA [12]. Next we assigned the 20 SNPs, selected by the SNPboost algorithm, to their corresponding genes. 9 out of 20 lie within genes and two of these genes can be linked through an additional gene. As shown in Figure (3b) CADPS2 interacts with NTRK2 via BDNF [12]. Hence, through SNPboost a new possible interaction might have been identified. Whether this interaction increase the risk of a disease must however be further evaluated.

4 Conclusion

In this work, we propose a boosting algorithm for classification as well as for the identification of potential SNP-SNP interactions on GWA data. The SNPboost algorithm is a modified version of the well-known adaboost algorithm. Using each possible SNP genotype as a weak classifier, we build a strong combined classifier. We evaluated SNPboost on the German MI family study II data.

Initially, the classification performance of the SNPboost algorithm clearly outperforms the linear SVM (LSVM) hence leading to the assumption that the selection strategy of the SNPboost might be more appropriate in a multivariate context than the standard selection by p-values. Training a gaussian kernel SVM (GSVM) on these SNPs further improves the classification performance, however only for small number of SNPs. Since SNPboost selects the SNPs that combined improve classification, interacting SNPs are likely to be chosen by the algorithm. In this work, we extracted the first 20 selected SNPs and mapped these to the genes. Of the 20 SNPs, 9 were found within genes. Two of these SNPs can be linked through a third gene. However, before one can state any interaction or biological plausibility, these results need to be further evaluated.

SNPboost is a very fast and memory efficient algorithm and can thus be applied even on the largest datasets without any preselection step. We would thus propose this algorithm as an promising tool for feature selection, interaction analysis and classification on GWA data.

Acknowledgements This work was supported by the Graduate School for Computing in Medicine and Life Sciences funded by Germanys Excellence Initiative [DFG GSC 235/1]

References

1. Ban, H.J., Heo, J.Y., Oh, K.S., Park, K.: Identification of type 2 diabetes-associated combination of snps using support vector machine. BMC Genetics 11(1), 26 (2010)
2. Erdmann, J., Großhennig, A., Braund, P.S., König, I.R., Hengstenberg, C., Hall, A.S., Linsel-Nitschke, P., et al.: New susceptibility locus for coronary artery disease on chromosome 3q22.3. Nat Genet 41(3), 280–282 (Mar 2009)
3. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. In: European Conference on Computational Learning Theory. pp. 23–37 (1995)
4. Freund, Y., Schapire, R.E.: A short introduction to boosting. Journal of japanese Society for Artificial Intelligence pp. 771–780 (1999)
5. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. 3, 1157–1182 (2003)
6. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Machine Learning 46, 389–422 (2002)
7. Ioannidis, J.P.: Prediction of cardiovascular disease outcomes and established cardiovascular risk factors by Genome-Wide association markers. Circ Cardiovasc Genet 2(1), 7–15 (Feb 2009)
8. Manolio, T.A., Collins, F.S., Cox, N.J., Goldstein, D.B., Hindorff, L.A., Hunter, D.J., McCarthy, M.I., et al.: Finding the missing heritability of complex diseases. Nature 461(7265), 747–753 (Oct 2009)
9. Moore, J.H.: The ubiquitous nature of epistasis in determining susceptibility to common human diseases. Human Heredity 56(1-3), 73–82 (2003)
10. Raelson, J.V., Little, R.D., Ruether, A., Fournier, H., Paquin, B., Van Eerdewegh, P., Bradley, W.E.C., et al.: Genome-wide association study for crohn's disease in the quebec founder population identifies multiple validated disease loci. Proceedings of the National Academy of Sciences 104(37), 14747–14752 (2007)
11. Samani, N.J., Erdmann, J., Hall, A.S., , Hengstenberg, C., Mangino, M., Mayer, B., Dixon, R.J., et al.: Genomewide Association Analysis of Coronary Artery Disease. N Engl J Med 357(5), 443–453 (2007)
12. Szklarczyk, D., Franceschini, A., Kuhn, M., Simonovic, M., Roth, A., Minguez, P., Doerks, T., Stark, M., Muller, J., Bork, P., Jensen, L.J., v. Mering, C.: The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored. Nucleic Acids Research 39(Database), D561–D568 (2010)
13. Vapnik, V.N.: Statistical Learning Theory. Wiley (1998)
14. Wei, Z., Wang, K., Qu, H.Q.Q., Zhang, H., Bradfield, J., Kim, C., Frackleton, E., et al.: From disease association to risk assessment: an optimistic view from genome-wide association studies on type 1 diabetes. PLoS genetics 5(10), e1000678+ (October 2009)
15. Wray, N.R., Goddard, M.E., Visscher, P.M.: Prediction of individual genetic risk of complex disease. Current Opinion in Genetics and Development 18(73), 257–263 (2008)
16. Yoon, Y., Song, J., Hong, S.H., Kim, J.Q.: Analysis of multiple single nucleotide polymorphisms of candidate genes related to coronary heart disease susceptibility by using support vector machines. Clinical Chemistry and Laboratory Medicine: CCLM / FESCC 41(4), 529–534 (Apr 2003), PMID: 12747598