ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

Combining latent tree modeling with a random forest-based approach, for genetic association studies

Christine Sinoquet[1] and Kamel Mekhnacha[2] *

1 - LS2N, UMR CNRS 6004, Université de Nantes
2 rue de la Houssinière, BP 92208, 44322 Nantes Cedex, France

2 - Probayes
180 avenue de l'Europe, Inovallée, 38330 Montbonnot, France

Abstract.
Association studies have been widely used to discover the genetic basis of complex phenotypes. However, standard univariate tests, and their alternatives, do not fully exploit the dependences between genetic markers. In this paper, we propose Sylva, a hybrid approach in which a random forest framework based on embedded trees benefits from a probabilistic graphical model. The latter is a collection of tree-shaped Bayesian networks with latent variables. We extensively compared Sylva and T-Trees, on simulated and real data. Sylva outperforms the already highly performant T-Trees, in a vast majority of cases.

1 Introduction

Genome-wide association studies (GWASs) rely on genetic markers, to help reveal the genetic architecture of complex phenotypes, such as diseases. In GWASs, hundred thousands of genetic markers, distributed throughout the genome, are observed for a population (*e.g.*, plants, animals), to detect a dependence (*i.e.*, an association) between some markers and the phenotype of interest. In this paper, we are interested in binary phenotypes observed from a cohort of affected subjects and a cohort of unaffected subjects, with a total population size in the order of a few thousands.

In the literature, several classification methods have been applied on GWAS data. These methods encompass logistic and penalized regressions, support vector machines, neural networks and random forests (RFs). A random forest (RF) [1] is a set of classification trees grown from bootstrap samples of observations. At each node, a random subset of K predictors is used to determine the best discriminative split with respect to a binary variable of interest denoted c hereafter. In a GWAS context, the learning set input to an RF consists of D, a matrix describing the p variables (*i.e.*, the genetic markers) of set V, for each of n observations (*i.e.*, subjects). RFs applied to GWAS data produce a ranking of the markers, by decreasing importances [2].

A limitation of RFs in GWASs is the lack of exploitation of linkage desequilibrium, defined as the non-random association of the variants of genetic markers,

*This research was supported by the French National Research Agency (ANR SAMOGWAS project). The software development and the realization of experiments were performed in part at the CCIPL (Centre de Calcul Intensif des Pays de la Loire, Nantes, France).

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

in a given population. In this context, it is appealing to consider latent variables that capture the dependences between the variables, to build the trees of a RF based on these variables. In this line, Botta and collaborators recently introduced the T-Trees (Trees inside Trees) model [3]. In T-Trees, the block-based modeling of dependences within variables is naive. It was therefore questionable whether (and to which extent) the high performances obtained for T-Trees could still be increased with a more refined model of dependences. In this paper, we propose Sylva, an innovative hybrid method combining T-Trees with FLTM (Forest of Latent Tree Model), a model derived from our former works [4].

The remainder of this paper is organized as follows. Section 2 briefly reviews the features of T-Trees and FLTM essential to the design of the hybrid method. The algorithms behind the hybrid approach are described in Section 3. Section 4 presents the evaluation datasets and experimental protocol, together with the results of our comparative studies of T-Trees and Sylva on simulated and real data.

2 Combining two forest models

In the GWAS context, the variables (*i.e.*, markers) are ordered along the genome. Linkage disequilibrium translates into the existence of local dependences within markers. To cope with these dependences, T-Trees slices the data into blocks of contiguous variables, of equal sizes. As will be explained later on, this data dimensionality reduction process is the key to further infer numerical latent variables.

This paradigm shift from standard RFs to T-Trees requires an adaptation to compute optimal splits for such latent variables. In the standard RF learning algorithm, for any variable v selected at random when growing a tree, all univariate split functions ($v \leq \theta_{v_r}$; $v > \theta_{v_r}$) are examined across the value domain $\mathscr{D}om_v = \{\theta_{v_r}\}_{r=1,\ldots,m_v}$. Thus can be determined the best discriminative split for this variable, to partition the current learning set. The univariate split functions used in standard RFs are replaced by non-linear multivariate split functions of contiguous variables in T-Trees. We will further detail how such a multivariate split function is itself connected to an embedded tree.

In the Sylva approach, we transform the original space of observed variables into a smaller space of discrete latent variables provided through FLTM modeling. A FLTM is a collection of latent tree models. A latent tree model is a tree-structured Bayesian network (BN) with observed discrete variables at leaf nodes, and discrete latent variables at internal nodes. The root is associated with a marginal distribution, and each other node is associated with a distribution of the corresponding variable given its parent. Learning a latent tree is challenging in the case of high dimensional data. There exist $O(2^{3p^2})$ candidate structures for a latent tree derived from p observed variables [5]. In high dimensional settings, learning a latent tree can only be efficiently addressed through iterative ascending clustering of variables [6]. In this category, the FLTM learning algorithm allows diversity for the cardinalities of latent variables and does not impose binary structure for trees, while still offering scalability.

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

3 Description of the hybrid approach

In this section, we describe the two main algorithms behind the Sylva approach.

3.1 The T-Trees-based framework

At the upper level, Sylva drives the construction of a RF as follows:

1. Initialize the RF, $\mathscr{F}_{tt}$, to the empty set.
2. Construct D_i, a bootstrap version of initial dataset D; grow tree $\mathscr{T}_i$ from learning set D_i *via* function **growTree** and add $\mathscr{T}_i$ to $\mathscr{F}_{tt}$.

The process stops when T trees have been built.

Function **growTree** expands tree $\mathscr{T}_i$ through a recursive scheme that will operate on smaller and smaller subsets of dataset D_i:

1. If a recursion termination case is detected (minimum sample size, maximum number of nodes, node purity, constance of variables), create a leaf node $\mathscr{T}$ labeled by the distribution of the binary variable of interest over observation sample D_i, and return $\mathscr{T}$.
2. Otherwise,
   2.1. Select at random K discrete latent variables from the FLTM model.
   2.2. For each discrete latent variable ℓ (corresponding to some cluster $c\ell$ of variables in V), infer a numerical latent variable ℓ' by expanding an Extra-Tree $\mathscr{E}_{c\ell}$. $\mathscr{E}_{c\ell}$ is grown from the learning set $D_{i,\ell}$, obtained as the projection of the matrix D_i (of second dimension $|\mathrm{V}|$) onto its $|\mathrm{c}\ell|$ columns related to ℓ. Compute $OS(\ell')$, the optimal split for numerical latent variable ℓ'.
   2.3. Identify the best split across all K numerical latent variables: $S^* = \underset{OS(\ell'_j), j=1,\ldots,K}{argmax}$ discriminativeScore $(OS(\ell'_j))$.
   2.4. Based on best split S^*, partition D_i into $D_{i,1}$ and $D_{i,2}$, and grow subtrees $\mathscr{T}_1$ and $\mathscr{T}_2$ from the former observation subsamples.
   2.5. Create node $\mathscr{T}$ labeled with S^*, graft $\mathscr{T}_1$ and $\mathscr{T}_2$ on $\mathscr{T}$, and return $\mathscr{T}$.

Growing an Extra-Tree is less complex than growing a tree in a standard RF: the split is selected *at random* for each of the K_e variables selected at random for some current node.

We now focus on 2.2, to explain how a numerical latent variable ℓ' is inferred. In the T-Trees framework, Extra-Trees are used to infer the novel numerical latent variables necessary to build the non-linear multivariate split functions:

1. The current learning set $D_{i,\ell}$ is distributed into $\mathscr{E}_{c\ell}$'s leaves; each leaf is then labeled with the probability that the binary variable of interest be equal to, say, 0, over the observation sample related to the leaf.
2. The value domain $\mathscr{D}om(\ell')$ of the novel numerical latent variable ℓ' is defined: for each observation o in $D_{i,\ell}$, $\ell'(o)$ is assigned the label of the leaf reached by o, and therefore $\mathscr{D}om(\ell') = \{\ell'(o),\ o \in D_{i,\ell}\}$.

Thus can be computed the optimal non-linear split $OS(\ell')$ mentioned in step 2.2.

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

3.2 Provision of precursor latent variables by FLTM

In the FLTM learning algorithm, clusters of pairwise dependent variables are subsumed through discrete latent variables, in an ascending hierarchical process. Full advantage is taken from the probabilistic graphical model framework, to validate these latent variables. The learning process is depicted as follows:

1. Initialize W, the working set of variables, to the whole set of variables in V, and initialize the forest $\mathscr{F}_{lm}$ to the collection of $|\mathrm{V}|$ univariate BNs.
2. Given some specified clustering method, partition W into non-overlapping clusters. The metrics used is the mutual information measure.
3. For each non-singleton cluster $c\ell$, try to infer a latent variable ℓ. For this purpose:
   3.1. Build a latent class model (LCM), that is a latent tree rooted in ℓ, and whose leaves are the variables in cluster $c\ell$.
   3.2. Learn the distributions associated to the nodes of this LCM using the Expectation-Maximization (EM) algorithm.
   3.3. Assess the quality of ℓ through a criterion $\mathscr{C}$, based on mutual information and entropy between ℓ and any child variable.
   3.4. If $\mathscr{C}$ is greater than a specified threshold τ, (i) in $\mathscr{F}_{lm}$'s structure, connect novel node ℓ to its child nodes as indicated by the LCM, (ii) replace the current marginal distribution of each child variable v, $\mathbb{P}(v)$, with the conditional probability $\mathbb{P}(v/\ell)$ computed with EM, and (iii) update W by discarding the child variables and including ℓ. Otherwise, keep the variables in $c\ell$ as isolated nodes in $\mathscr{F}_{lm}$.
   3.5. Repeat steps 2 and 3 until no latent variable can be inferred or a single cluster is built from W.

4 Experimental study

We compared the performances of T-Trees and Sylva on simulated and real datasets. The FLTM learning algorithm relies on ProBT, a C++ library dedicated to BNs (http://www.probayes.com/fr/recherche/probt/). The current implementation allows to choose between adapted versions of CAST, DBSCAN and the Louvain method, to cluster variables.

4.1 Experimental protocol

We used a widely-used software program, HAPGEN, to simulate realistic genotypic data harboring a dependence between one of the simulated variables and the simulated phenotype. We simulated 27 conditions and generated 100 replicate datasets (20,000 markers, 2,000 affected and 2,000 controls) under each condition. Performance is measured as the percentage of simulated causal markers identified among the variables with the highest importances, over all 100 replicates related to the same condition. We successively computed this percentage for the top 25, top 50, top 100, top 200 and top 1000 variables showing the highest importances.

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

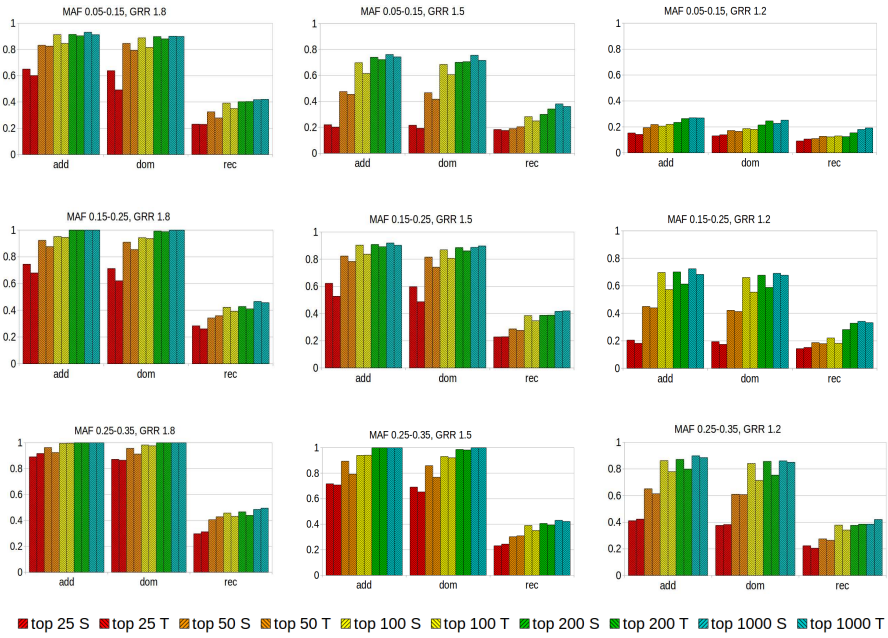

Fig. 1: Comparison of performances for T-Trees and Sylva/DBSCAN, under 27 simulated conditions (GM × GRR × MAF). GM: genetic disease model (additive, dominant, recessive); GRR: genetic relative risk; MAF: minor allele frequency of the causal marker; S: Sylva approach, T: T-Trees approach.

In complement, we ran T-Trees and Sylva on 161 large-scale real datasets provided by the Wellcome Trust Case Control Consortium) (http://www.wtccc.org.uk/). Therein, for each of seven diseases, around 4,500 to 5,000 affected and unaffected subjects are described for each of the 23 human chromosomes. Across the chromosomes, the number of markers ranges from 5,754 to 38,867. We ran Sylva for each of methods CAST, DBSCAN and Louvain. We compared the discriminative powers of T-Trees and Sylva through their areas under ROC curves.

4.2 Results

The results on simulations are displayed in figure 1. For the additive and dominant genetic models, we find that Sylva almost always outperforms T-Trees when small sets of top markers are examined. Then, sooner or later, the discrepancy between the methods diminishes. Besides, we observe a more or less smooth degradation of the performances as the MAF and GRR decrease. For the recessive model, the performances are poor for T-Trees and Sylva in all conditions.

Both methods show comparable (high) AUCs (table 1(A)) on real datasets. When CAST or DBSCAN is used, over all 161 chomosome-wide datasets, Sylva slightly outperforms T-Trees in a vast majority of cases (respectively 71.7%

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

AUC	T	SC	SD	SL
min	0.887	0.902	0.890	0.885
max	0.961	0.979	0.972	0.955
avg	0.934	**0.946**	**0.952**	**0.940**

(A)

cases for which AUC(S) > AUC(T)				
		T *vs* SC	T *vs* SD	T *vs* SL
		71.7%	75.5%	62.3%
diff	min	0.012	0.009	0.005
	max	0.037	0.048	0.055
	avg	**0.021**	**0.023**	0.017

(B)

co-occurrences of AUC(S) > AUC(T) over the 161 datasets				
T *vs* SC	T *vs* SD	T *vs* SL		p-val χ^2
x	x	x	53.5%	
x	x		61.6%	2.7e-06
x		x	62.3%	$<$ 2.2e-16
	x	x	53.5%	2.0e-04

(C)

Table 1: Comparison of the predictive powers of T-Trees and Sylva on 161 real datasets. SC: Sylva/CAST, SD: Sylva/DBSCAN; SL: Sylva/Louvain.

and 75.5%); this percentage is 62.3% for the Louvain method (table 1(B)). First, it was not a foregone result that we could improve the performance of T-Trees as it was already high. Second, we observe substantial percentages of co-occurrences for outperformance of Sylva over T-Trees, between any two methods among CAST, DBSCAN and Louvain (table 1(C)). χ^2 tests indicate that these co-occurrences are statistically significant. This latter result emphasizes the advantage of combining T-Trees with FLTM, as shown on simulated data.

5 Conclusions and future work

In this paper, we put forward the hybrid approach Sylva, that combines T-Trees with a refined modeling of dependences between variables. We showed that Sylva outperforms T-Trees for applications in association genetics, in a vast majority of cases. Our next work will introduce consensus clustering in FLTM with the aim of increasing the performance of Sylva. Besides, to apply Sylva on genome scale, efforts will be spent to increase the scalability of FLTM learning algorithm.

References

[1] L. Breiman, Random forests, *Machine Learning*, 45(1):5-32, 2001.

[2] G. Louppe, L. Wehenkel, A. Sutera and P. Geurts, Understanding variable importances in forests of randomized trees. In *proceedings of Advances in Neural Information Processing Systems 26 (NIPS 2013)*, pages 431-439, 2013.

[3] V. Botta, G. Louppe, P. Geurts and L. Wehenkel, Exploiting SNP correlations within random forest for genome-wide association studies, *PLOS ONE*, 9(4):e93379, 2014.

[4] R. Mourad, C. Sinoquet and P. Leray, A hierarchical Bayesian network approach for linkage disequilibrium modeling and data-dimensionality reduction prior to genome-wide association studies, *BMC Bioinformatics*, 12(1):16, 2011.

[5] N.L. Zhang, Hierarchical latent class models for cluster analysis, *Journal of Machine Learning Research*, 5:697-723, 2004.

[6] R. Mourad, C. Sinoquet, N.L. Zhang, T. Liu, P. Leray, A survey on latent tree models and applications, *Journal or Artificial Intelligence Research*, 47:157-203, 2013.