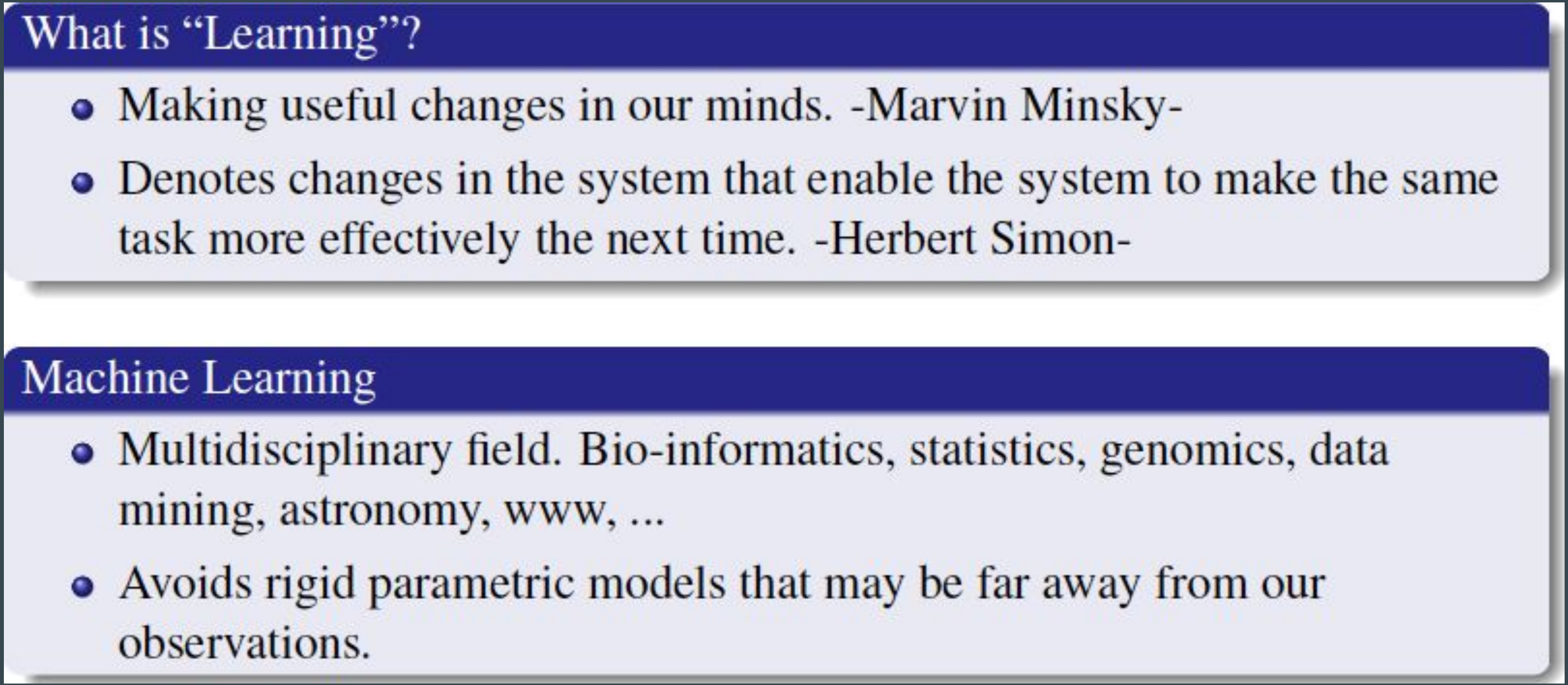
Machine Learning

Evangelina López de Maturana & Oscar González-Recio

Topics

Reasoning	Advantages	Methods	Diving-in	Meta-analysis
What is ML	ML properties	Brief review of ML algorithms used in GWP	Further details on ensemble methods	Comparison of ML methods in the literature

Machine Learning



What is “Learning”?

- Making useful changes in our minds. -Marvin Minsky-
- Denotes changes in the system that enable the system to make the same task more effectively the next time. -Herbert Simon-

Machine Learning

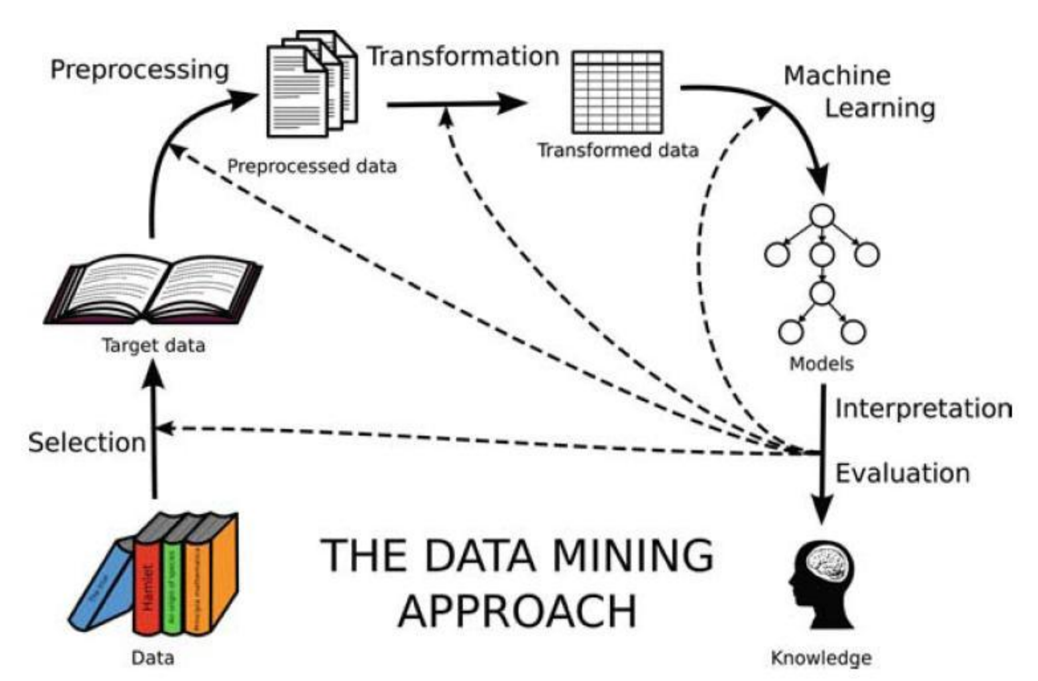
- Multidisciplinary field. Bio-informatics, statistics, genomics, data mining, astronomy, www, ...
- Avoids rigid parametric models that may be far away from our observations.

Machine Learning

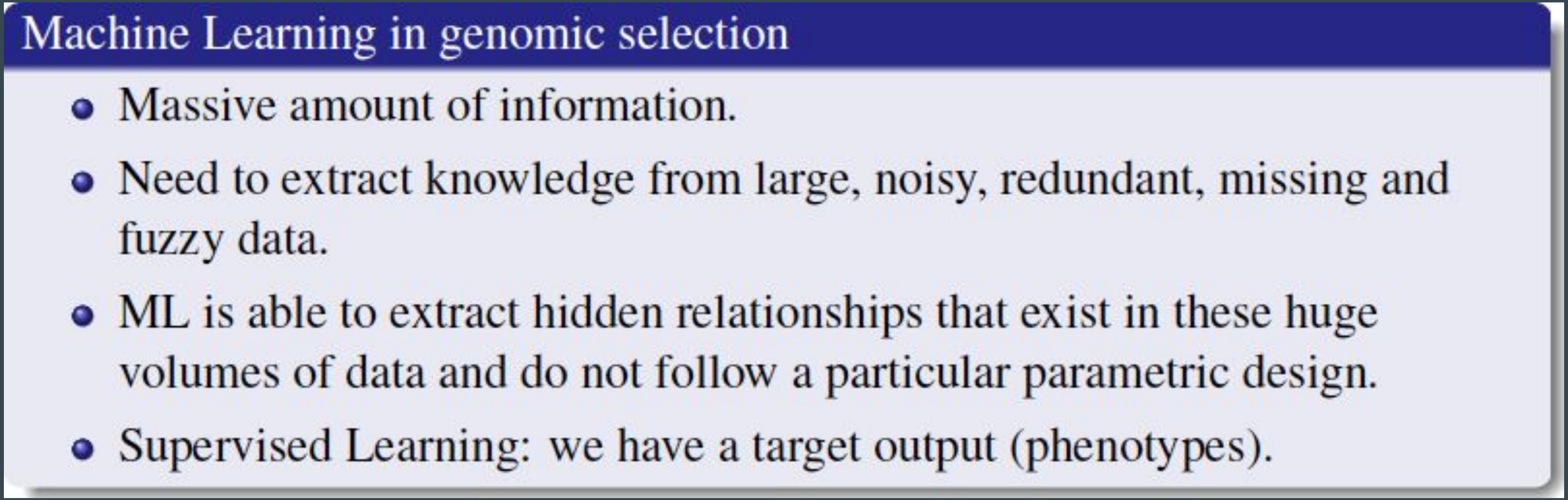
$$y=Xb+e$$

The assumption that an increment of one unit in the dose of an allele has a fixed and linear effect in the phenotype is a simplistic and unrealistic assumption.

Machine Learning



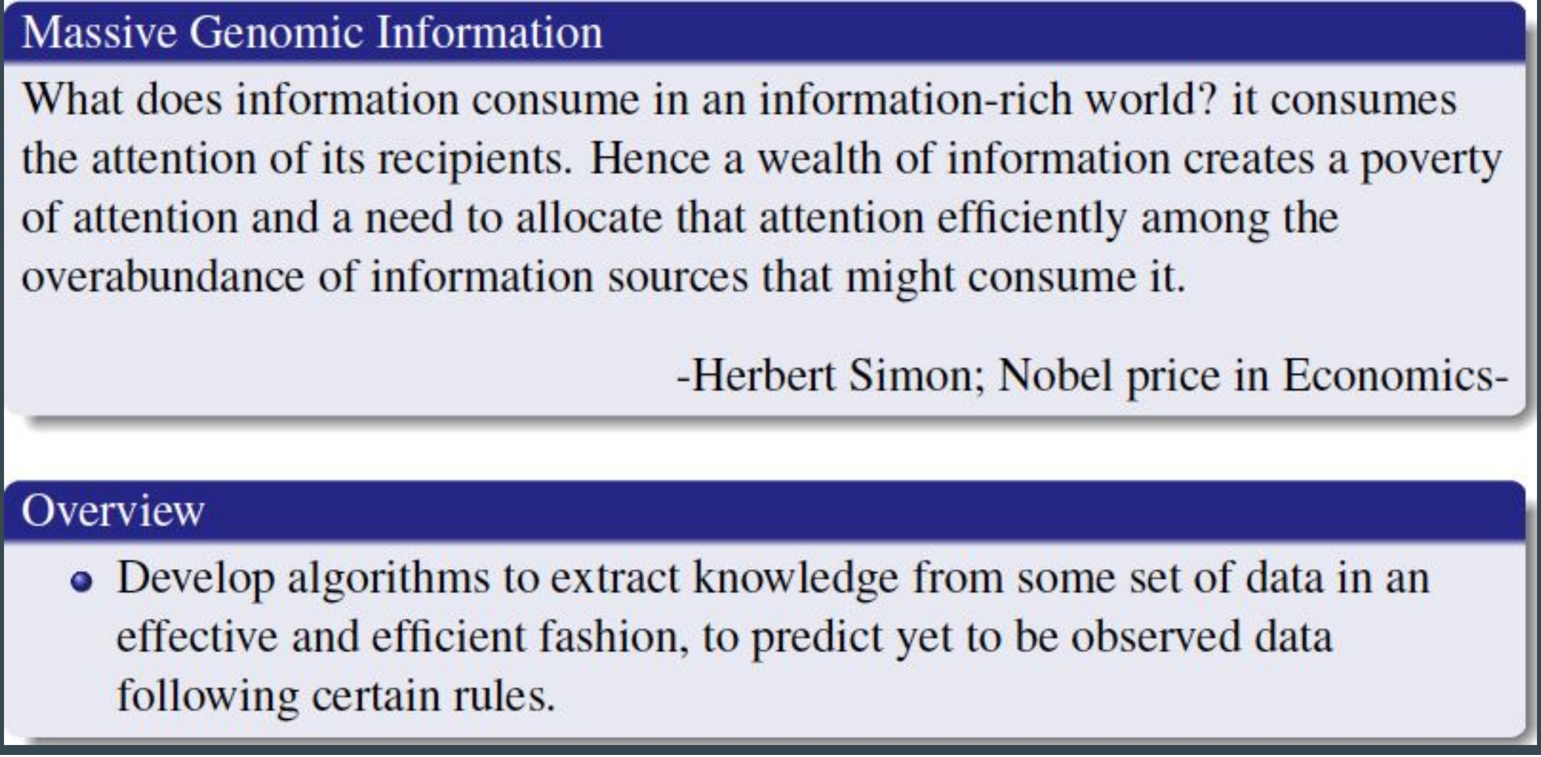
Machine Learning



Machine Learning in genomic selection

- Massive amount of information.
- Need to extract knowledge from large, noisy, redundant, missing and fuzzy data.
- ML is able to extract hidden relationships that exist in these huge volumes of data and do not follow a particular parametric design.
- Supervised Learning: we have a target output (phenotypes).

Machine Learning



Massive Genomic Information

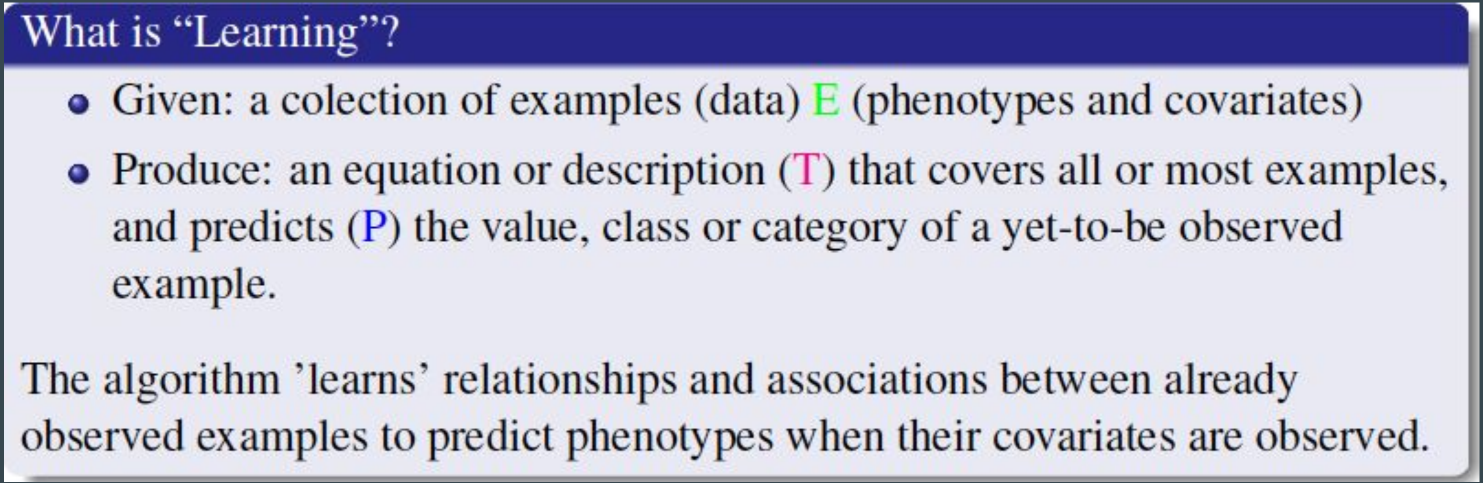
What does information consume in an information-rich world? it consumes the attention of its recipients. Hence a wealth of information creates a poverty of attention and a need to allocate that attention efficiently among the overabundance of information sources that might consume it.

-Herbert Simon; Nobel price in Economics-

Overview

- Develop algorithms to extract knowledge from some set of data in an effective and efficient fashion, to predict yet to be observed data following certain rules.

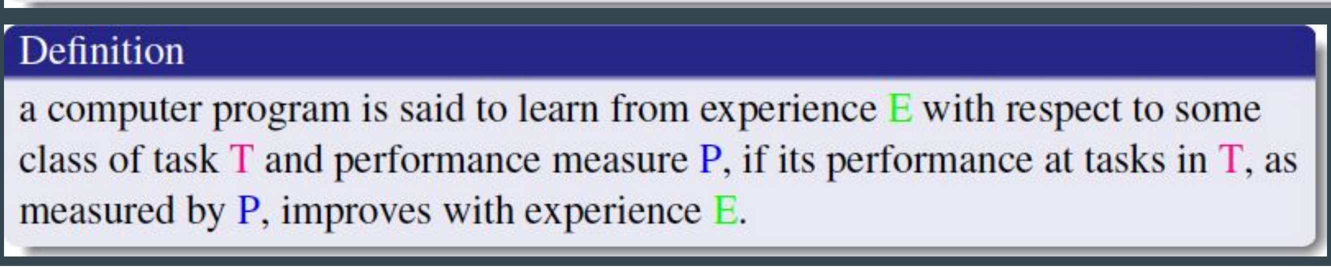
Machine Learning



What is “Learning”?

- Given: a colection of examples (data) E (phenotypes and covariates)
- Produce: an equation or description (T) that covers all or most examples, and predicts (P) the value, class or category of a yet-to-be observed example.

The algorithm ’learns’ relationships and associations between already observed examples to predict phenotypes when their covariates are observed.



Definition

a computer program is said to learn from experience E with respect to some class of task T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

Machine Learning

RKHS

Artificial Neural Networks

Bayesian Neural Networks

Random Forest

Boosting

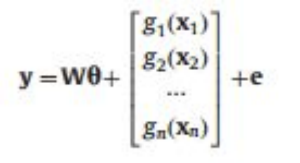
Others: radial basis functions, support vector machines, bagging, lasso

Regression problems

$$y_i = X_i b_i + Z_{ui} u_i + e'_i \quad \Rightarrow \quad \mathbf{y} = \mathbf{1}\mu + \begin{bmatrix} g_1(\mathbf{x}_1) \\ g_2(\mathbf{x}_2) \\ \cdots \\ g_n(\mathbf{x}_n) \end{bmatrix} + \mathbf{e}$$

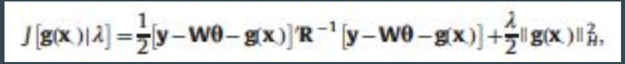
$g(\mathbf{x})$ is some function of SNP genotypes

RKHS recap

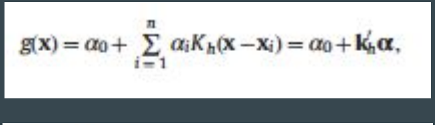


$$\mathbf{y} = \mathbf{W}\boldsymbol{\theta} + \begin{bmatrix} g_1(\mathbf{x}_1) \\ g_2(\mathbf{x}_2) \\ \cdots \\ g_n(\mathbf{x}_n) \end{bmatrix} + \mathbf{e}$$

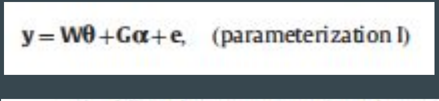
With penalized residual sum of squares →



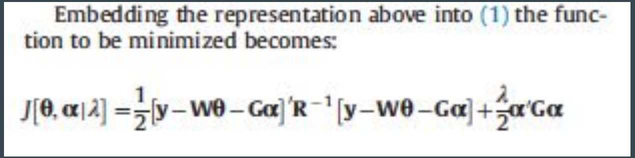
$$J[\mathbf{g}(\mathbf{x})|\lambda] = \frac{1}{2}[\mathbf{y} - \mathbf{W}\boldsymbol{\theta} - \mathbf{g}(\mathbf{x})]'\mathbf{R}^{-1}[\mathbf{y} - \mathbf{W}\boldsymbol{\theta} - \mathbf{g}(\mathbf{x})] + \frac{\lambda}{2}\|\mathbf{g}(\mathbf{x})\|^2_H,$$



$$g(\mathbf{x}) = \alpha_0 + \sum_{i=1}^{n} \alpha_i K_h(\mathbf{x} - \mathbf{x}_i) = \alpha_0 + \mathbf{k}_h'\boldsymbol{\alpha},$$

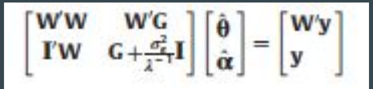


$$\mathbf{y} = \mathbf{W}\boldsymbol{\theta} + \mathbf{G}\boldsymbol{\alpha} + \mathbf{e}, \quad \text{(parameterization I)}$$



Embedding the representation above into (1) the function to be minimized becomes:

$$J[\boldsymbol{\theta}, \boldsymbol{\alpha}|\lambda] = \frac{1}{2}[\mathbf{y} - \mathbf{W}\boldsymbol{\theta} - \mathbf{G}\boldsymbol{\alpha}]'\mathbf{R}^{-1}[\mathbf{y} - \mathbf{W}\boldsymbol{\theta} - \mathbf{G}\boldsymbol{\alpha}] + \frac{\lambda}{2}\boldsymbol{\alpha}'\mathbf{G}\boldsymbol{\alpha}$$



$$\begin{bmatrix} \mathbf{W}'\mathbf{W} & \mathbf{W}'\mathbf{G} \\ \mathbf{I}'\mathbf{W} & \mathbf{G} + \frac{\sigma_e^2}{\lambda^{-1}}\mathbf{I} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\theta}} \\ \hat{\boldsymbol{\alpha}} \end{bmatrix} = \begin{bmatrix} \mathbf{W}'\mathbf{y} \\ \mathbf{y} \end{bmatrix}$$

Artificial Neural Networks

$$\mathbf{y} = \mathbf{W}\boldsymbol{\theta} + \begin{bmatrix} g_1(\mathbf{x}_1) \\ g_2(\mathbf{x}_2) \\ \cdots \\ g_n(\mathbf{x}_n) \end{bmatrix} + \mathbf{e}$$

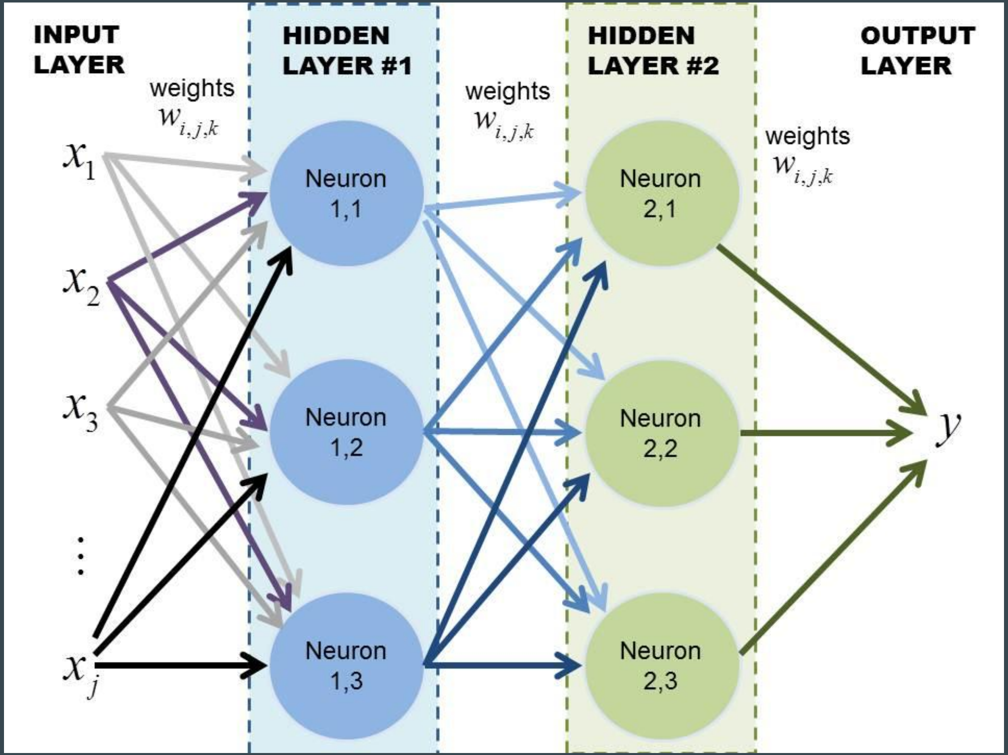
$$g_i(\mathbf{x}_{g,i}) = \beta_0 + \sum_{s=1}^{S} w_s f_s(\mathbf{w}_s; \mathbf{y}, \mathbf{x}_{g,i})$$

$f_s(\mathbf{w}_s; \mathbf{y}, \mathbf{x}_{g,i})$ is a transformation (linear or non-linear) for neuron s, with w being the vector of connection strengths between neurons

If the network is trained using Bayesian statistics, it is called Bayesian Neural Network (BNN)

Artificial Neural Networks

$$\mathbf{y} = \mathbf{W}\boldsymbol{\theta} + \begin{bmatrix} g_1(\mathbf{x}_1) \\ g_2(\mathbf{x}_2) \\ \cdots \\ g_n(\mathbf{x}_n) \end{bmatrix} + \mathbf{e}$$

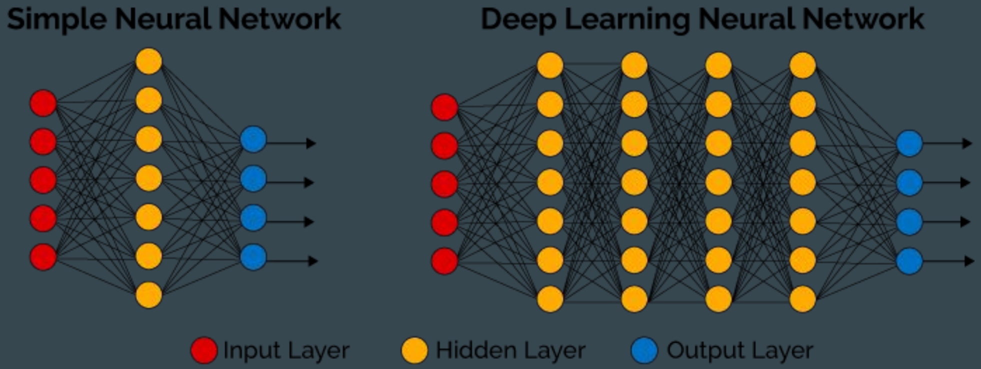


Different activity functions possibly for the neurons:

Threshold, lineal, exponential, sigmoidal, hyperbolic, …

Deep Learning

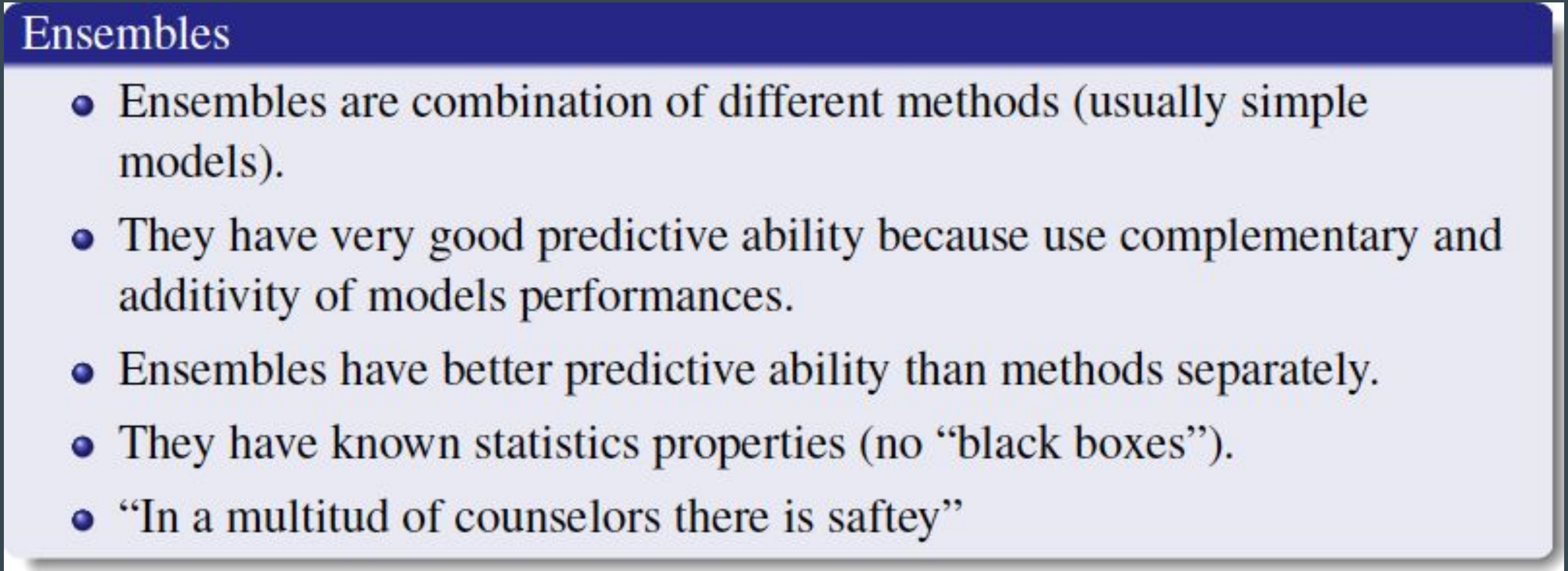
$$\mathbf{y} = \mathbf{W}\boldsymbol{\theta} + \begin{bmatrix} g_1(\mathbf{x}_1) \\ g_2(\mathbf{x}_2) \\ \cdots \\ g_n(\mathbf{x}_n) \end{bmatrix} + \mathbf{e}$$



Add more layers. More complex models, and larger number of parameters to estimate. Still not better than other methods.

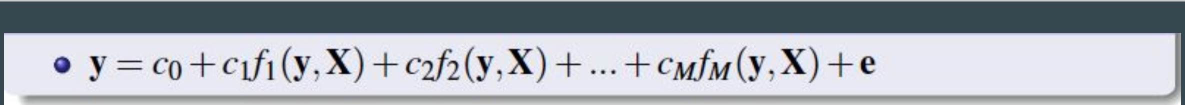
@teenybiscuit

Ensemble methods



Ensembles

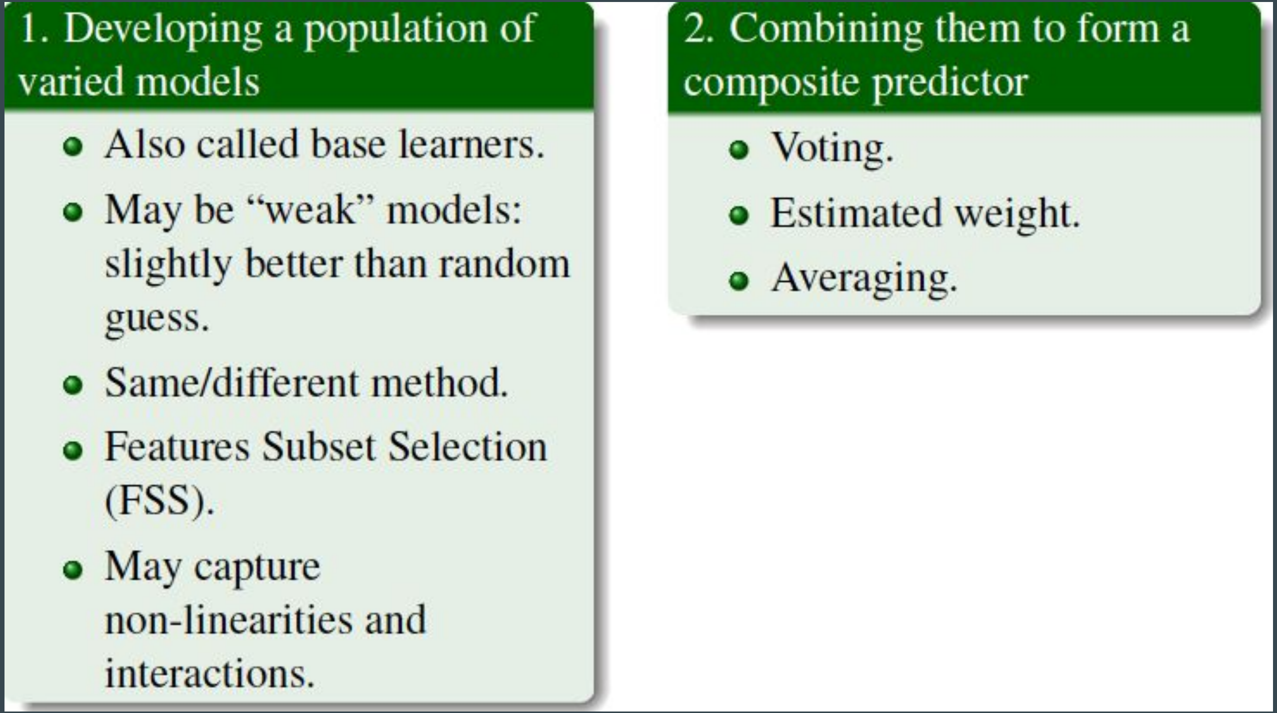
- Ensembles are combination of different methods (usually simple models).
- They have very good predictive ability because use complementary and additivity of models performances.
- Ensembles have better predictive ability than methods separately.
- They have known statistics properties (no “black boxes”).
- “In a multitud of counselors there is saftey”



- $\mathbf{y} = c_0 + c_1 f_1(\mathbf{y},\mathbf{X}) + c_2 f_2(\mathbf{y},\mathbf{X}) + \ldots + c_M f_M(\mathbf{y},\mathbf{X}) + \mathbf{e}$

Ensemble methods

Two steps



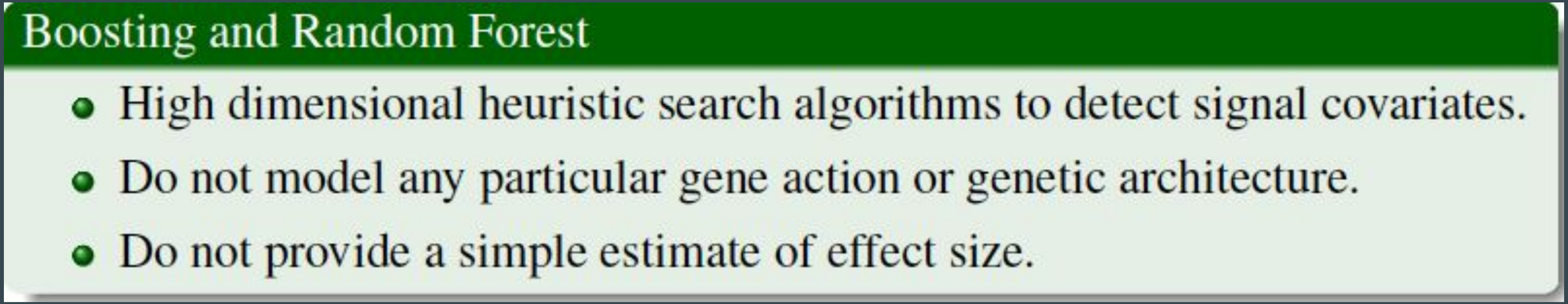
1. Developing a population of varied models

- Also called base learners.
- May be "weak" models: slightly better than random guess.
- Same/different method.
- Features Subset Selection (FSS).
- May capture non-linearities and interactions.

2. Combining them to form a composite predictor

- Voting.
- Estimated weight.
- Averaging.

Ensemble methods

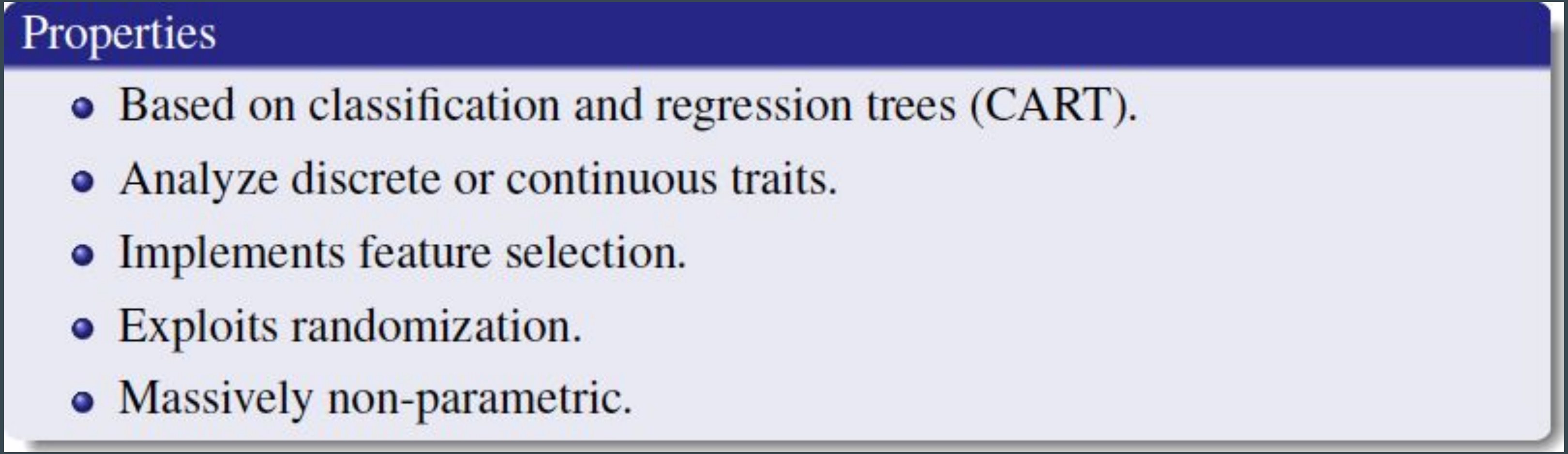


Boosting and Random Forest

- High dimensional heuristic search algorithms to detect signal covariates.
- Do not model any particular gene action or genetic architecture.
- Do not provide a simple estimate of effect size.

Random Forest

Ensemble methods



Properties

- Based on classification and regression trees (CART).
- Analyze discrete or continuous traits.
- Implements feature selection.
- Exploits randomization.
- Massively non-parametric.

Random Forest

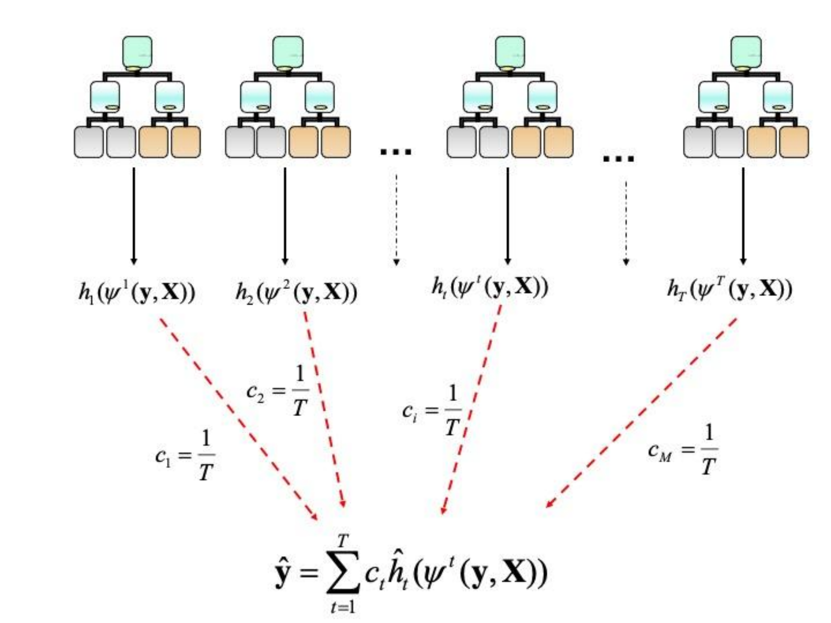
Ensemble methods

- Based on Classification And Regression Trees (CART).
- Use Randomization and Bagging.
- Performs Feature Subset Selection.
- Convenient for classification problems.
- Fast computation
- Simple interpretation of results for human minds.

Random Forest

Ensemble methods

$$\mathbf{y} = \mathbf{W}\boldsymbol{\theta} + \begin{bmatrix} g_1(\mathbf{x}_1) \\ g_2(\mathbf{x}_2) \\ \cdots \\ g_n(\mathbf{x}_n) \end{bmatrix} + \mathbf{e}$$


...
...
$h_1(\psi^1(\mathbf{y},\mathbf{X}))$
$h_2(\psi^2(\mathbf{y},\mathbf{X}))$
$h_t(\psi^t(\mathbf{y},\mathbf{X}))$
$h_T(\psi^T(\mathbf{y},\mathbf{X}))$
$c_2 = \frac{1}{T}$
$c_i = \frac{1}{T}$
$c_1 = \frac{1}{T}$
$c_M = \frac{1}{T}$
$\hat{\mathbf{y}} = \sum_{t=1}^{T} c_t \hat{h}_t(\psi^t(\mathbf{y},\mathbf{X}))$

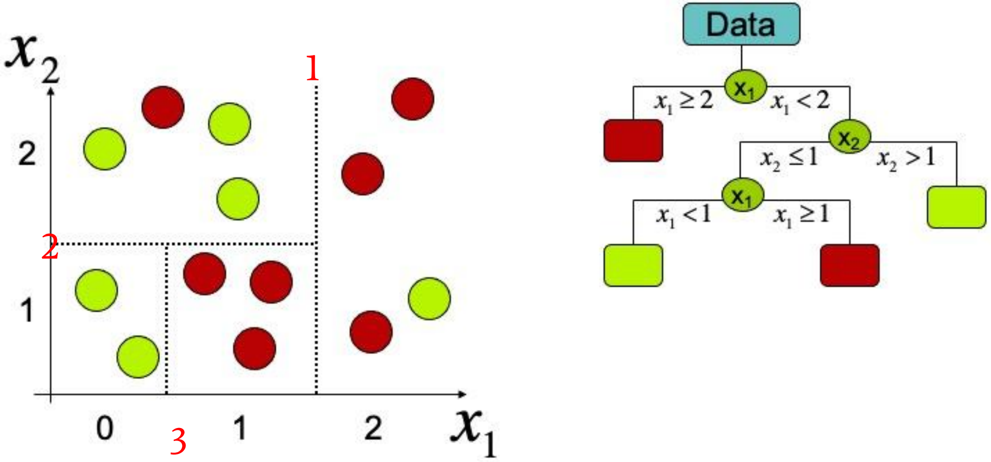
$$g(\mathbf{x}) = \frac{1}{M} \sum_{m=1}^{M} t_m(\Psi_m(\mathbf{y};\mathbf{X}))$$

t_m is a decision tree (CART) on a bootstrapped sample of the data set.

Random Forest

Ensemble methods

Classification trees

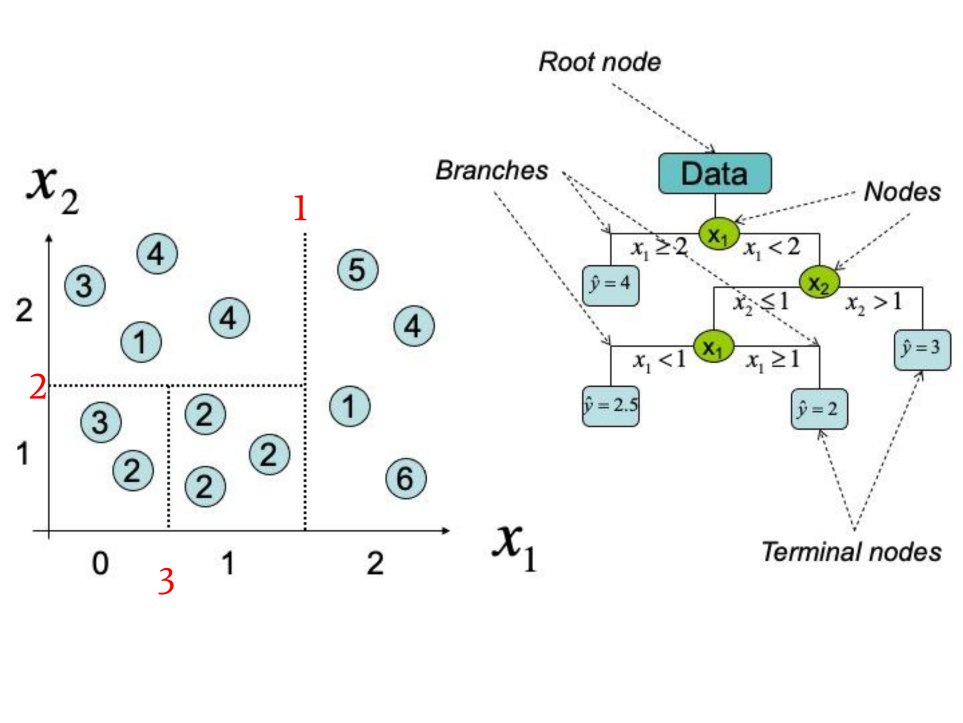


Random Forest

Ensemble methods

Regression trees

Trees are not pruned (do not care about overfitting of a single tree)

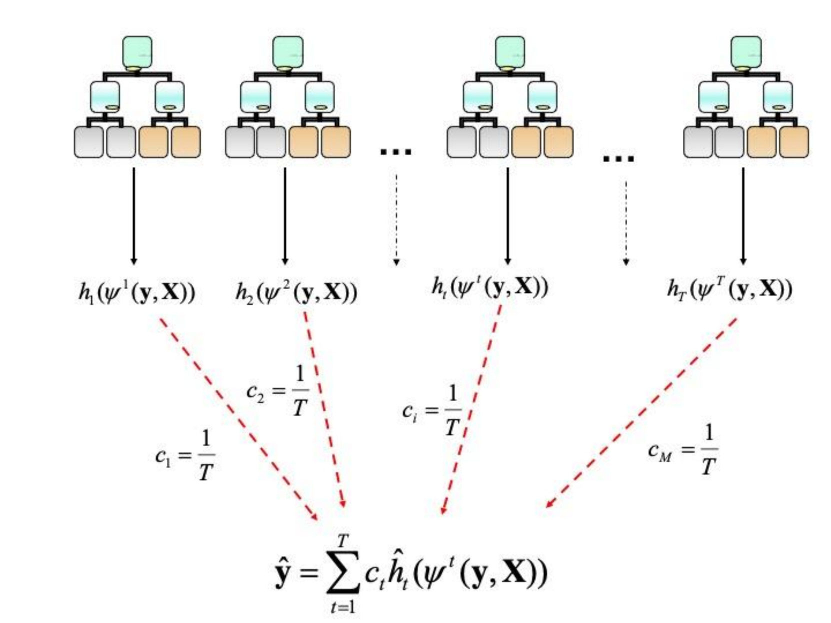


Random Forest

Ensemble methods

$$\mathbf{y} = \mathbf{W}\boldsymbol{\theta} + \begin{bmatrix} g_1(\mathbf{x}_1) \\ g_2(\mathbf{x}_2) \\ \cdots \\ g_n(\mathbf{x}_n) \end{bmatrix} + \mathbf{e}$$

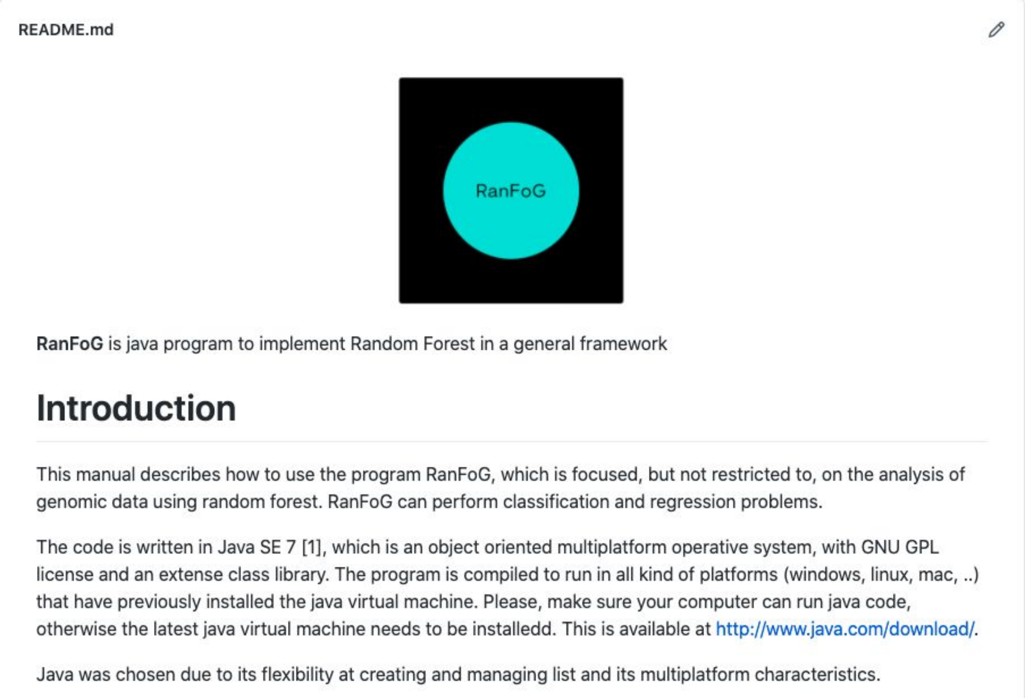
$$g(\mathbf{x}) = \frac{1}{M} \sum_{m=1}^{M} t_m(\Psi_m(\mathbf{y};\mathbf{X}))$$



...
...
$h_1(\psi^1(\mathbf{y},\mathbf{X}))$
$h_2(\psi^2(\mathbf{y},\mathbf{X}))$
$h_t(\psi^t(\mathbf{y},\mathbf{X}))$
$h_T(\psi^T(\mathbf{y},\mathbf{X}))$
$c_2 = \frac{1}{T}$
$c_i = \frac{1}{T}$
$c_1 = \frac{1}{T}$
$c_M = \frac{1}{T}$
$\hat{\mathbf{y}} = \sum_{t=1}^{T} c_t \hat{h}_t(\psi^t(\mathbf{y},\mathbf{X}))$

- t_m is a decision tree (CART) on a bootstrapped sample of the data set.
- The remaining observations are sent to an Out Of Bag (OOB) data set.
- The OOB will serve to monitor the loss function and to calculate the Variable Importance

RANFOG



README.md

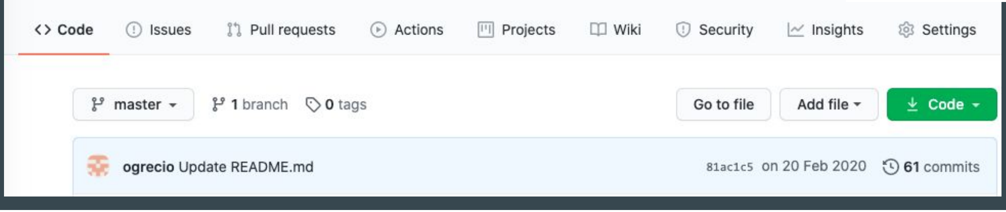
RanFoG is java program to implement Random Forest in a general framework

Introduction

This manual describes how to use the program RanFoG, which is focused, but not restricted to, on the analysis of genomic data using random forest. RanFoG can perform classification and regression problems.

The code is written in Java SE 7 [1], which is an object oriented multiplatform operative system, with GNU GPL license and an extense class library. The program is compiled to run in all kind of platforms (windows, linux, mac, ..) that have previously installed the java virtual machine. Please, make sure your computer can run java code, otherwise the latest java virtual machine needs to be installedd. This is available at http://www.java.com/download/.

Java was chosen due to its flexibility at creating and managing list and its multiplatform characteristics.

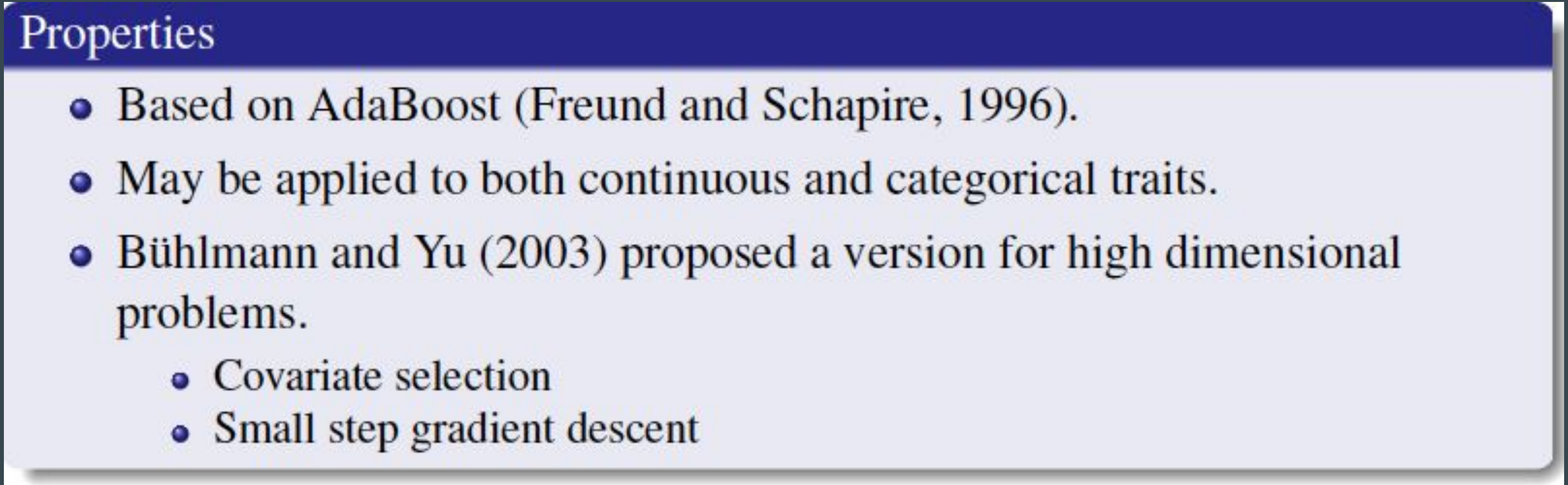


ogrecio / RanFog

ogrecio Update README.md 81ac1c5 on 20 Feb 2020 61 commits

Boosting

Ensemble methods



Properties

- Based on AdaBoost (Freund and Schapire, 1996).
- May be applied to both continuous and categorical traits.
- Bühlmann and Yu (2003) proposed a version for high dimensional problems.
  - Covariate selection
  - Small step gradient descent

Boosting

Ensemble methods

$$\mathbf{y} = \mathbf{W}\boldsymbol{\theta} + \begin{bmatrix} g_1(\mathbf{x}_1) \\ g_2(\mathbf{x}_2) \\ \cdots \\ g_n(\mathbf{x}_n) \end{bmatrix} + \mathbf{e}$$

$$g(\mathbf{x}) = \sum_{m=1}^{M} h_m(\mathbf{y};\mathbf{X})$$

h_m is a naïve predictor that minimizes a loss function (e.g. MSE), and built on the residuals from h_{m-1}

$$\hat{g}_m(\mathbf{x}) = g_{m-1}(\mathbf{x}) + \nu h_m(y_i;\mathbf{x}) \quad \text{with} \quad \nu \in (0,1)$$

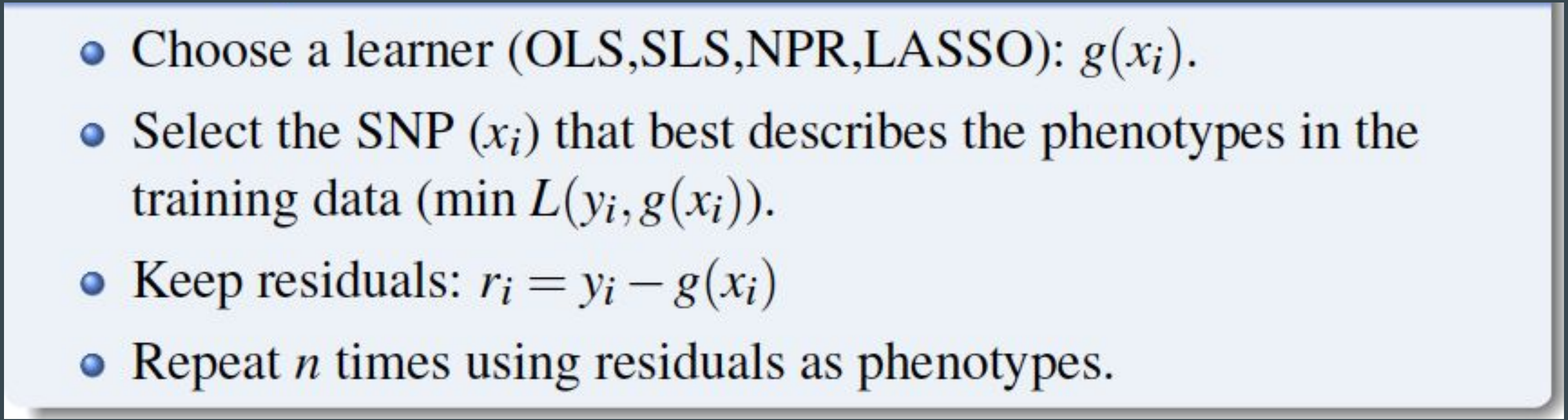
Usually, a shrinkage factor is applied on the naïve predictor, to improve convergence to a global minimum

Random Boosting: select mtry variables at each iteration to speed up the algorithm

Boosting

Ensemble methods

Brief description of the algorithm

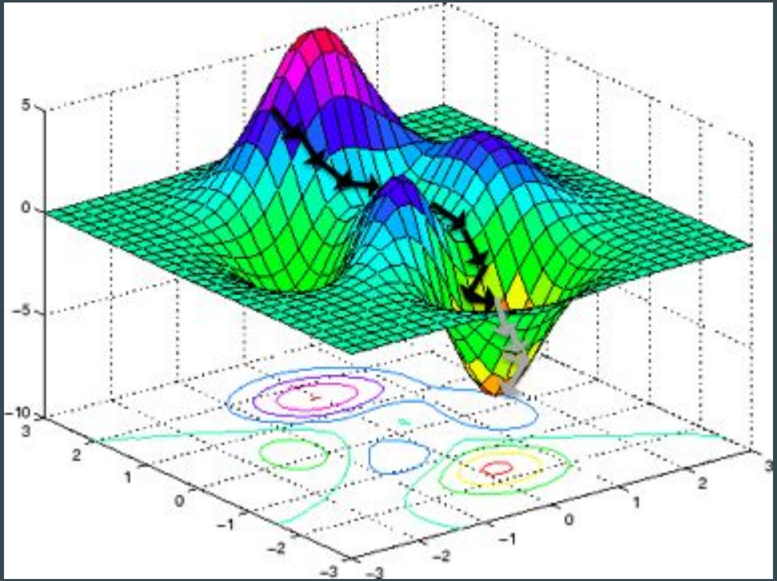


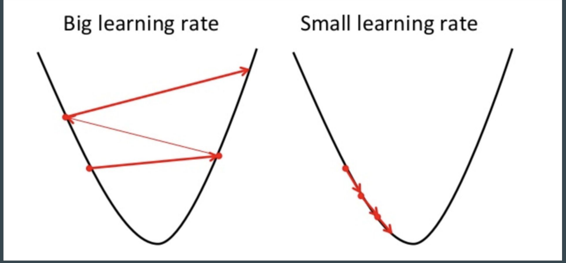
- Choose a learner (OLS,SLS,NPR,LASSO): $g(x_i)$.
- Select the SNP (x_i) that best describes the phenotypes in the training data (min $L(y_i, g(x_i))$.
- Keep residuals: $r_i = y_i - g(x_i)$
- Repeat n times using residuals as phenotypes.

Boosting

Ensemble methods

- Based on small gradients descent steps
- Performs feature subset selection
- Use simple regression
- “Highest” level of shrinkage
- Fast computation
- Any amount of data and markers.
- Tractable in “whole genome sequencing”



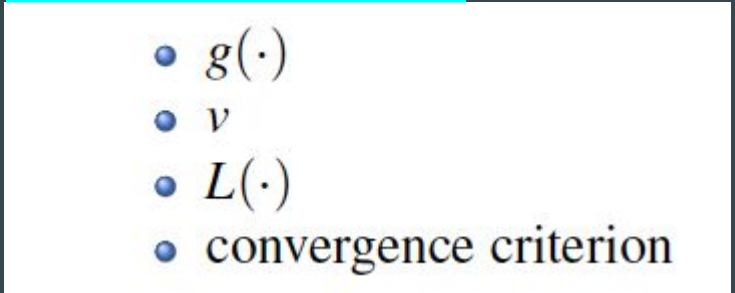


Boosting

Ensemble methods

- Based on small gradients descent steps
- Performs feature subset selection
- Use simple regression
- “Highest” level of shrinkage
- Fast computation
- Any amount of data and markers.
- Tractable in “whole genome sequencing”

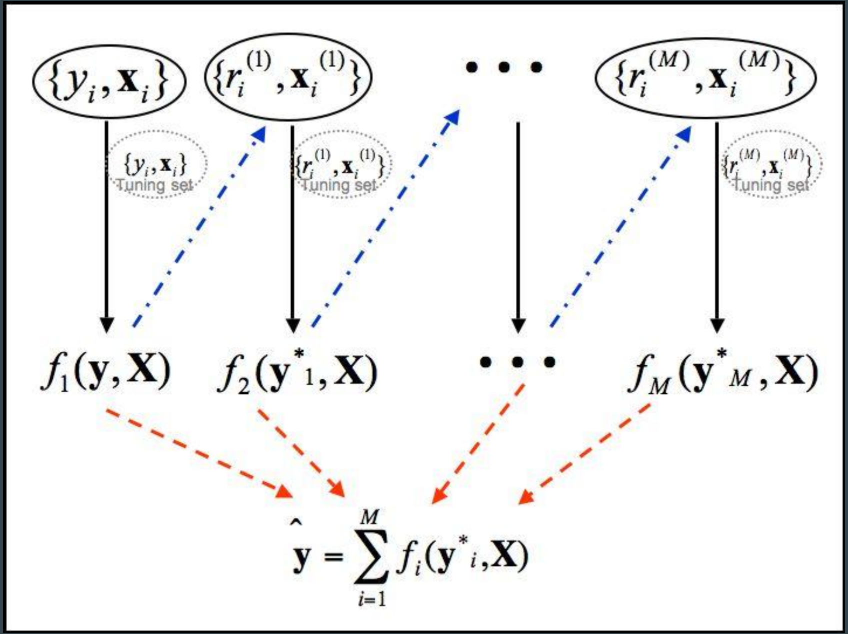
Features to be tune or chosen



- $g(\cdot)$
- ν
- $L(\cdot)$
- convergence criterion

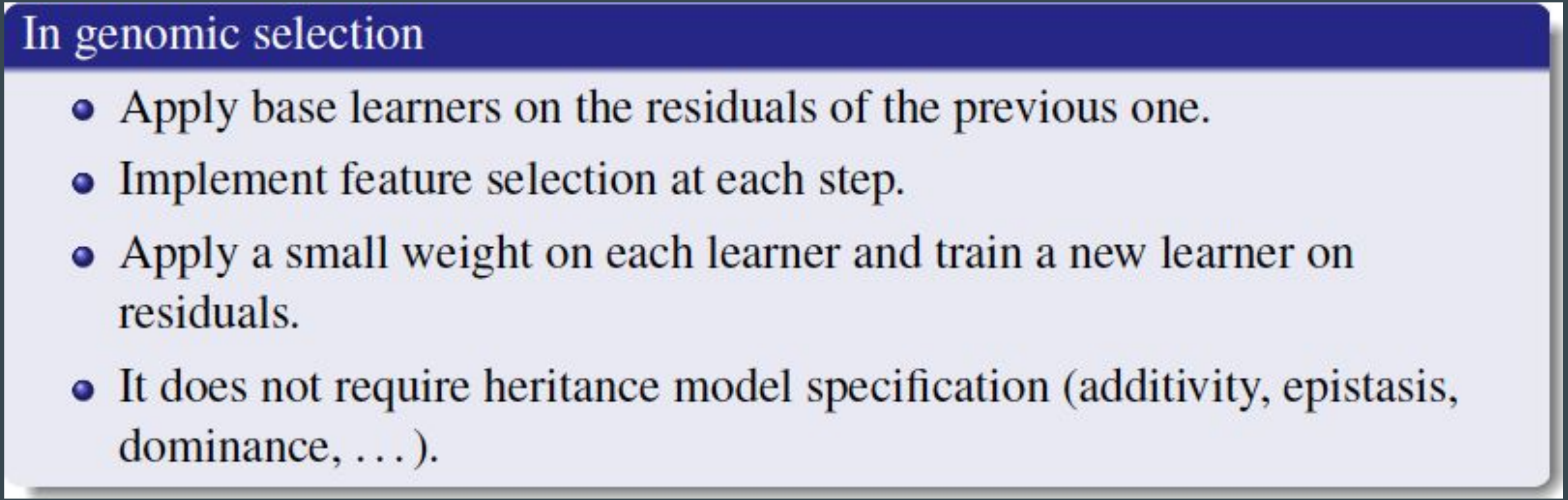
Boosting

Ensemble methods


$\{y_i, \mathbf{x}_i\}$
$\{r_i^{(1)}, \mathbf{x}_i^{(1)}\}$
$\{r_i^{(M)}, \mathbf{x}_i^{(M)}\}$
$\{y_i, \mathbf{x}_i\}$ Tuning set
$\{r_i^{(1)}, \mathbf{x}_i^{(1)}\}$ Tuning set
$\{r_i^{(M)}, \mathbf{x}_i^{(M)}\}$ Tuning set
$f_1(\mathbf{y}, \mathbf{X})$
$f_2(\mathbf{y}^*_1, \mathbf{X})$
$f_M(\mathbf{y}^*_M, \mathbf{X})$
$\hat{\mathbf{y}} = \sum_{i=1}^{M} f_i(\mathbf{y}^*_i, \mathbf{X})$

Boosting

Ensemble methods



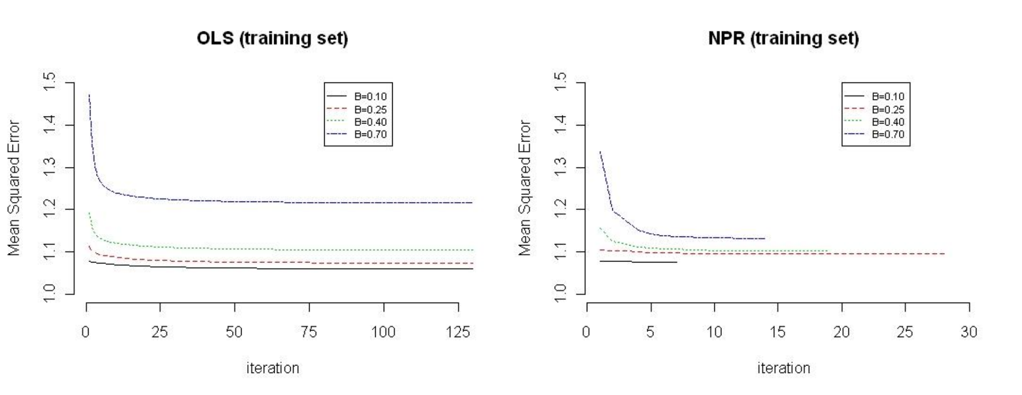
In genomic selection

- Apply base learners on the residuals of the previous one.
- Implement feature selection at each step.
- Apply a small weight on each learner and train a new learner on residuals.
- It does not require heritance model specification (additivity, epistasis, dominance, . . .).

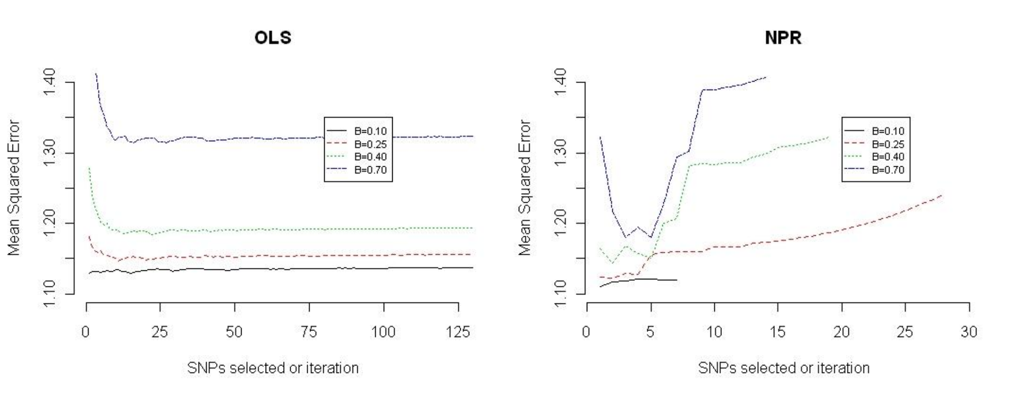
Boosting

Ensemble methods

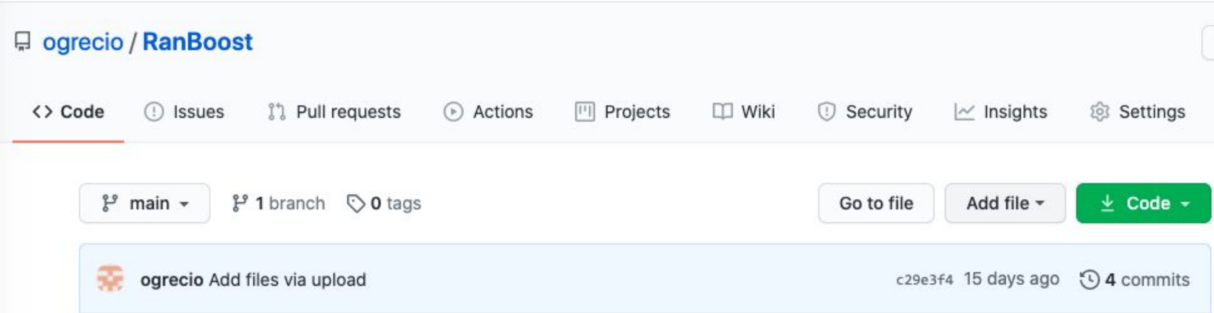
Bias-variance trade off (training set)



Bias-variance trade off (testing set)



RanBOOST



ogrecio / RanBoost

ogrecio Add files via upload c29e3f4 15 days ago 4 commits

Bagging

Ensemble methods

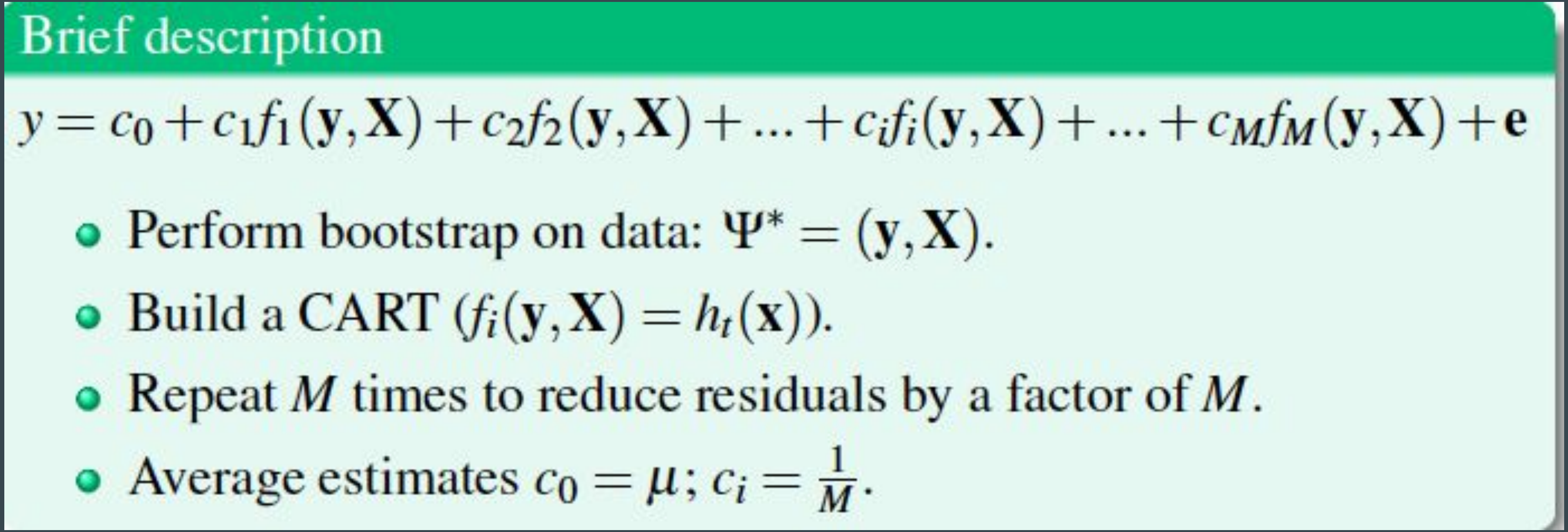
$$\mathbf{y} = \mathbf{W}\boldsymbol{\theta} + \begin{bmatrix} g_1(\mathbf{x}_1) \\ g_2(\mathbf{x}_2) \\ \cdots \\ g_n(\mathbf{x}_n) \end{bmatrix} + \mathbf{e}$$

$$g(\mathbf{x}) = \sum_{m=1}^{M} h_m(\mathbf{y}; \mathbf{X})$$

h_m is a predictor on a bootstrapped sample on the data, divided by M (averaging).

Bagging

Ensemble methods

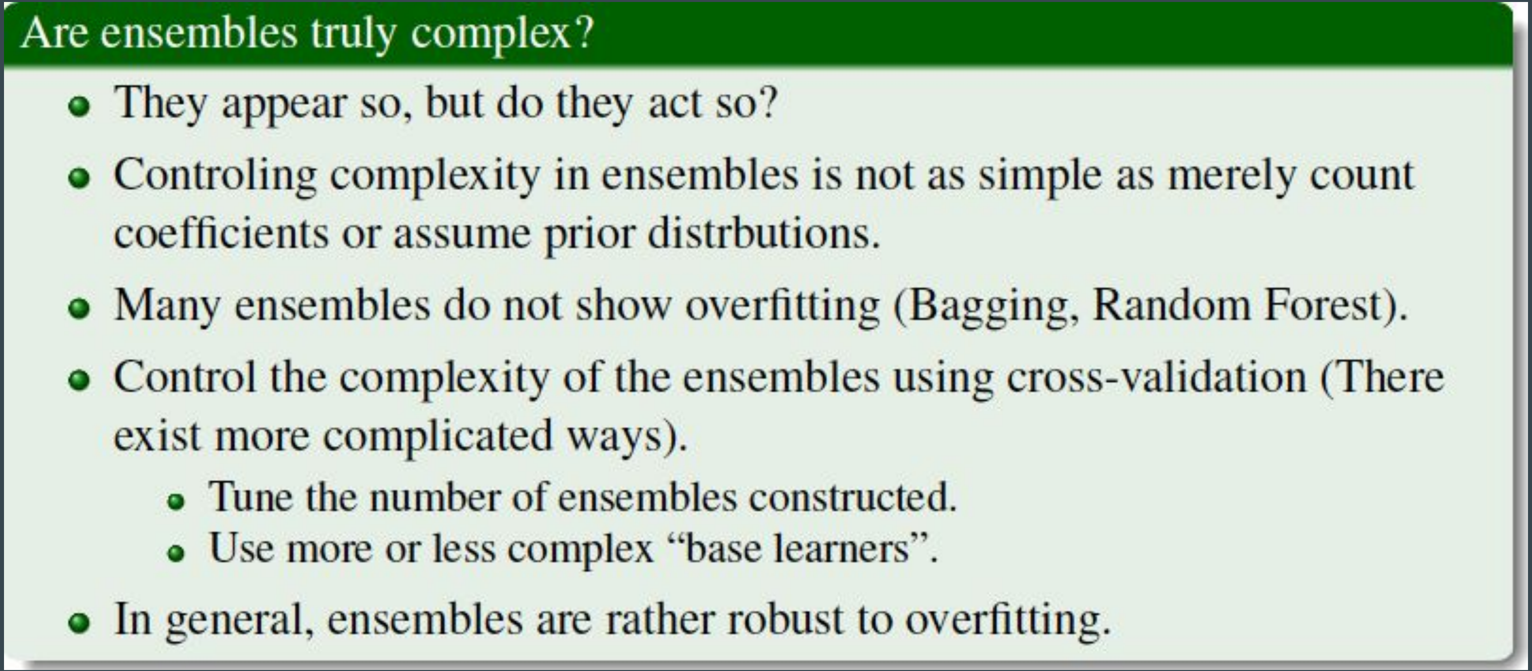


Brief description

$$y = c_0 + c_1 f_1(\mathbf{y},\mathbf{X}) + c_2 f_2(\mathbf{y},\mathbf{X}) + \ldots + c_i f_i(\mathbf{y},\mathbf{X}) + \ldots + c_M f_M(\mathbf{y},\mathbf{X}) + \mathbf{e}$$

- Perform bootstrap on data: $\Psi^* = (\mathbf{y},\mathbf{X})$.
- Build a CART ($f_i(\mathbf{y},\mathbf{X}) = h_t(\mathbf{x})$).
- Repeat M times to reduce residuals by a factor of M.
- Average estimates $c_0 = \mu$; $c_i = \frac{1}{M}$.

Ensemble methods



Are ensembles truly complex?

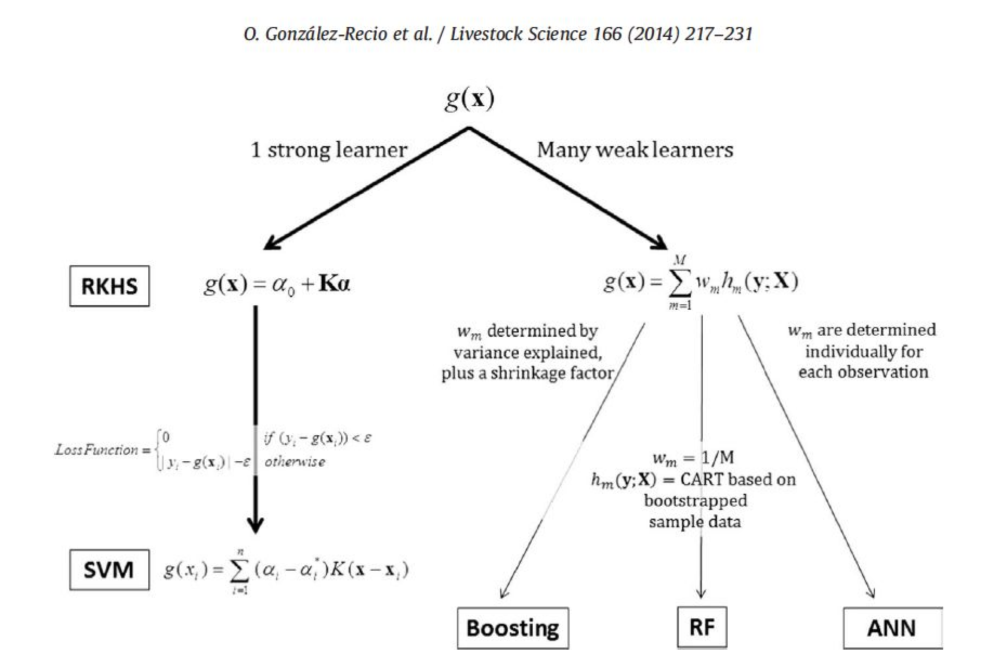
- They appear so, but do they act so?
- Controling complexity in ensembles is not as simple as merely count coefficients or assume prior distrbutions.
- Many ensembles do not show overfitting (Bagging, Random Forest).
- Control the complexity of the ensembles using cross-validation (There exist more complicated ways).
  - Tune the number of ensembles constructed.
  - Use more or less complex “base learners”.
- In general, ensembles are rather robust to overfitting.

Ensemble methods

- ✓ Use simple models.
- Use many models.
- Interpretation of many models, even simple model, may be much harder than with a single model.
- Ensembles are competitive in accuracy though at a probable loss of interpretability.
- Too complex ensembles may lead to overfitting.

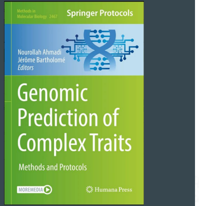
Comparison between methods

- González-Recio et al. (2014)


O. González-Recio et al. / Livestock Science 166 (2014) 217–231
$g(\mathbf{x})$
1 strong learner
Many weak learners
RKHS
$g(\mathbf{x}) = \alpha_0 + \mathbf{K}\boldsymbol{\alpha}$
$g(\mathbf{x}) = \sum_{m=1}^{M} w_m h_m(\mathbf{y};\mathbf{X})$
w_m determined by variance explained, plus a shrinkage factor
w_m are determined individually for each observation
$LossFunction = \begin{cases} 0 & if\ (y_i - g(\mathbf{x}_i)) < \varepsilon \\ \lvert y_i - g(\mathbf{x}_i) \rvert - \varepsilon & otherwise \end{cases}$
$w_m = 1/M$
$h_m(\mathbf{y};\mathbf{X})$ = CART based on bootstrapped sample data
SVM
$g(x_i) = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) K(\mathbf{x} - \mathbf{x}_i)$
Boosting
RF
ANN

Comparison between methods

- Reinoso et al. (in book “Genomic prediction of complex traits”)



Springer Protocols

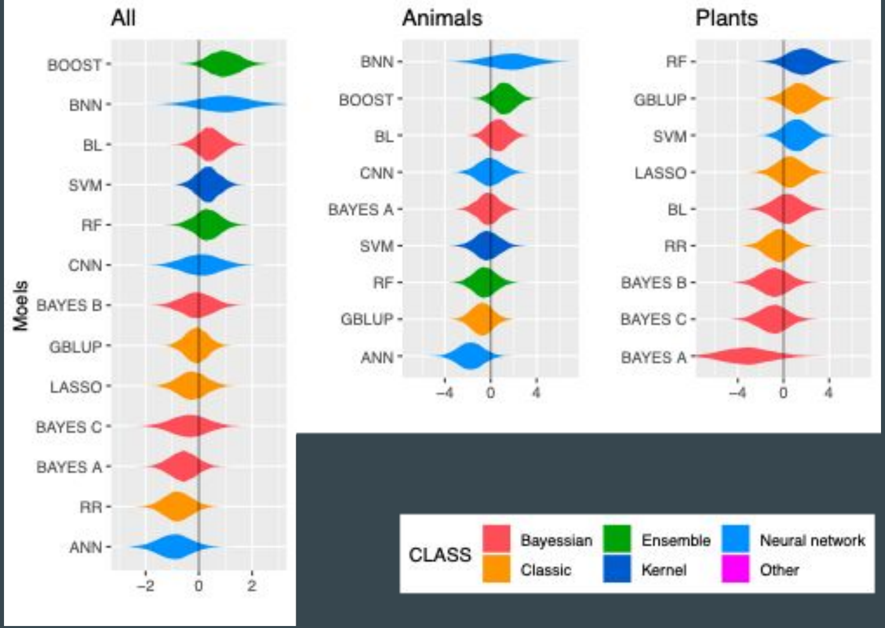
Nourollah Ahmadi
Jérôme Bartholomé
Editors

Genomic Prediction of Complex Traits

Methods and Protocols

Humana Press

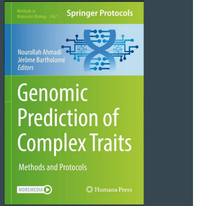
Systematic review and meta-analysis for the predictive performance (mean squared error) using a Thurstonian model.



Might be biased on researcher’s preferences biased!

Comparison between methods

- Reinoso et al. (in book “Genomic prediction of complex traits”)



Springer Protocols

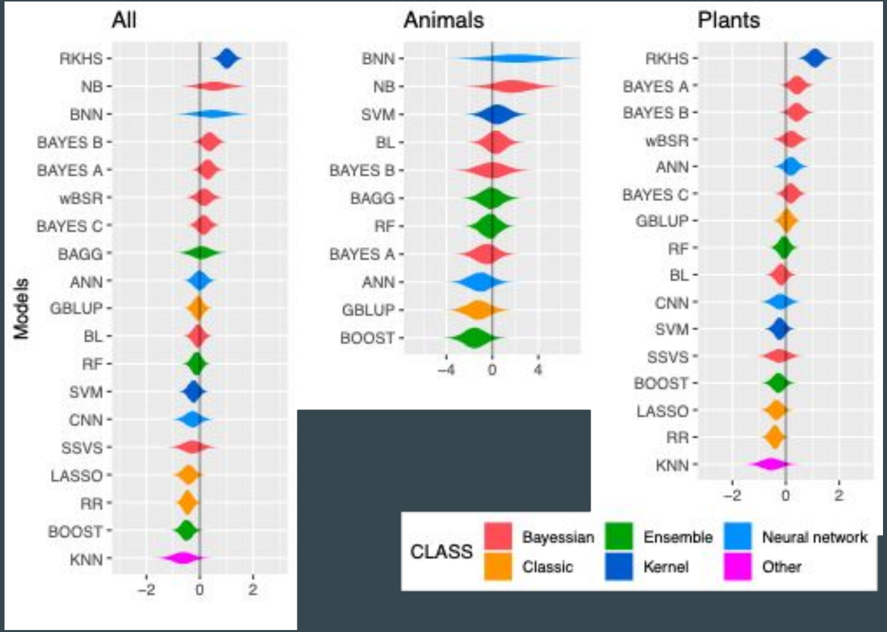
Nourollah Ahmadi
Jérôme Bartholomé
Editors

Genomic Prediction of Complex Traits

Methods and Protocols

Humana Press

Systematic review and meta-analysis for the predictive performance (pearson correlation) using a Thurstonian model.



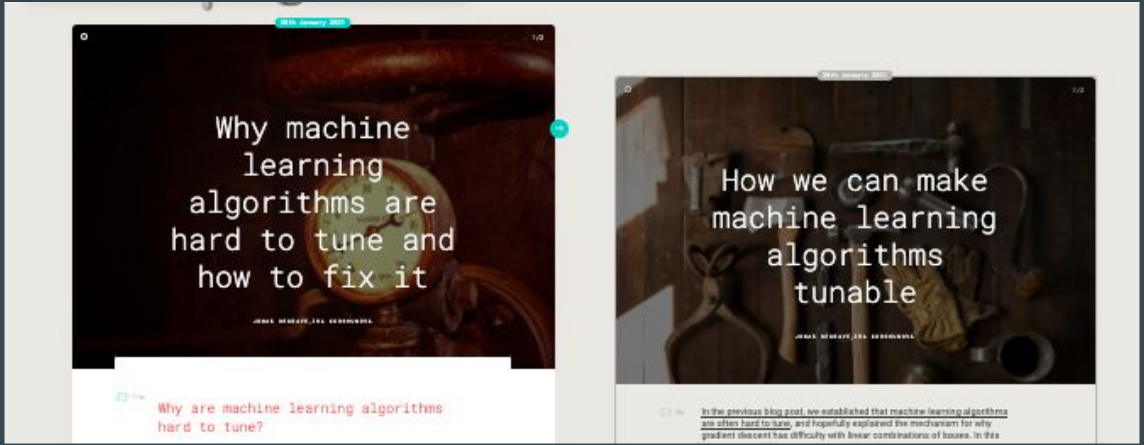
Might be biased on researcher’s preferences biased!

Considerations

- Machine Learning methods need of a thorough tuning of hyperparameters. Dedicate some time to tune them using internal and external cross-validation
- Usually, work better than ‘traditional’ models.
- Difficult to interpret from a biological point of view (but linear models are also an unrealistic simplification of biology).
- Some can be very fast, and easy computational pipelines can be implemented for genomic prediction (e.g. Random Boosting).

Considerations

- Not “one case fits all”.
- https://engraved.ghost.io/why-machine-learning-algorithms-are-hard-to-tune/



Why machine learning algorithms are hard to tune and how to fix it

Why are machine learning algorithms hard to tune?

How we can make machine learning algorithms tunable

In the previous blog post, we established that machine learning algorithms are often hard to tune, and hopefully explained the mechanism for why gradient descent has difficulty with linear combinations of losses. In this

References

- Breiman L (2001) Random forest. Machine Learning, 45:5–3.
- Goldstein et al. (2010) BMC Genetics 11:49.
- Gonzalez-Recio O, S. Forni (2011) Genome-wide prediction of discrete traits using bayesian regressions and machine learning. Genetics Selection Evolution, 43:7
- Hastie et al. (2009) Elements of Statistical Learning. 2nd Edition.
- Seni and Elder (2010) Ensemble Methods in Data Mining.
- González-Recio O., K. A. Weigel, D. Gianola, H. Naya, G.J.M. Rosa. 2010. L2-boosting algorithm applied to high dimensional problems in genomic selection. Genetics Research 92(3): 227-237.
- González-Recio O., J.A. Jiménez-Montero, R. Alenda. 2013. The gradient boosting algorithm and random Boosting for genome-assisted evaluation in large data sets. Journal of Dairy Science 96: 614-624.