Random subspace with trees for feature selection under memory constraints

Antonio Sutera

Dept. of EECS, University of Liège, Belgium

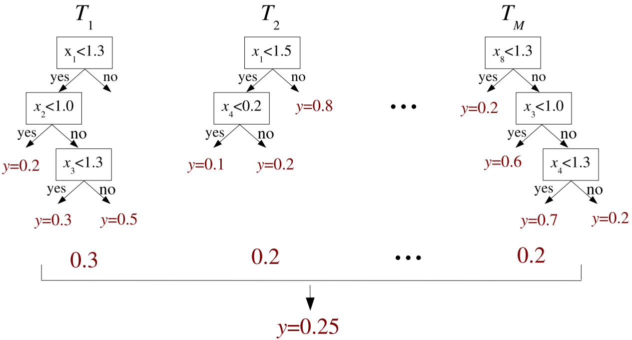
Benelearn 2016,
Kortrijk, Belgium
September 12, 2016

Pierre Geurts, Louis Wehenkel (ULg),
Gilles Louppe (CERN & NYU)
Célia Châtel (Luminy)

Université de Liège

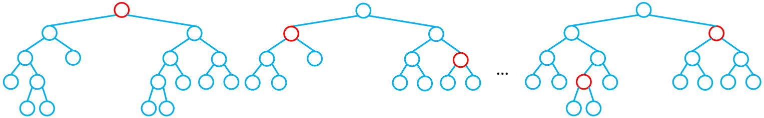
Background: Ensemble of randomized trees

✓ Good classification method



Background: Ensemble of randomized trees for feature selection

✓ Good classification method **useful for feature selection**

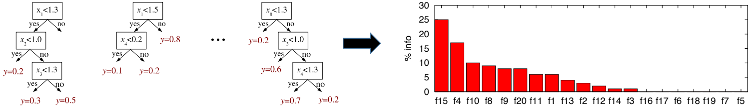


Importance of variable X_m for an ensemble of N_T trees is given by:

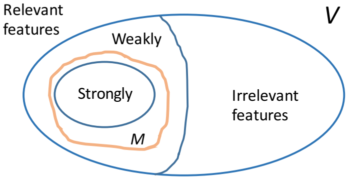
$$Imp(X_m) = \frac{1}{N_T} \sum_{T} \sum_{t \in T : v(t) = X_m} p(t) \Delta i(t)$$

where $p(t) = N_t / N$ and $\Delta i(t)$ is the impurity reduction at node t:

$$\Delta i(t) = i(t) - \frac{N_{t_L}}{N_t} i(t_L) - \frac{N_{t_r}}{N_t} i(t_R)$$



Background: Feature relevance (Kohavi and John, 1997)

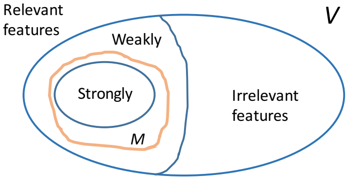


Given an output Y and a set of input variables V, $X \in V$ is

- **relevant** iff $\exists B \subseteq V$ such that $Y \not\perp\!\!\!\perp X | B$.
- **irrelevant** iff $\forall B \subseteq V$: $Y \perp\!\!\!\perp X | B$
- **strongly relevant** iff $Y \not\perp\!\!\!\perp X | V \setminus \{X\}$.
- **weakly relevant** iff X is relevant and not strongly relevant.

A **Markov boundary** is a minimal size subset $M \subseteq V$ such that $Y \perp\!\!\!\perp V \setminus M | M$.

Background: Feature selection (Nilsson et al., 2007)



Two different feature selection problems:

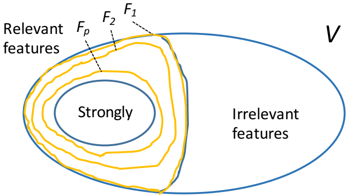
- **Minimal-optimal:** find a Markov boundary for the output Y.
- **All-relevant:** find all relevant features.

Random forests, variable importance and feature selection

Main results

In asymptotic conditions : infinite sample size and number of trees

- $K = 1$: Unpruned totally randomized trees solve the **all-relevant** feature selection problem.
- $K > 1$: In the case of stricly positive distributions, non random trees always find a superset F of the minimal-optimal solution which size decreases with K.



Motivation

Our objective: Design more efficient feature selection procedures based on random forests

- We address large-scale feature selection problems where one can not assume that all variables can be stored into memory
- We study and improve ensembles of trees grown from random subsets of features

Random subspace for feature selection

Simplistic memory constrained setting: We can not grow trees with more than q features

Straightforward ensemble solution: Random Subspace (RS)

Train each ensemble tree from a random subset of q features

1. Repeat T times:
   1.1 Let Q be a subset of q features randomly selected in V
   1.2 Grow a tree only using features in Q (with randomization K)
2. Compute importance $Imp_{q,T}(X)$ for all X

Proposed e.g. by (Ho, 1998) for accuracy improvement, by (Louppe and Geurts, 2012) for handling large datasets and by (Draminski et al., 2010, Konukoglu and Ganz, 2014) for feature selection

Let us study the population version of this algorithm.

RS for feature selection: study

Asymptotic guarantees:

- **Def.** $deg(X)$ with X relevant is the size of the smallest $B \subseteq V$ such that $Y \not\perp\!\!\!\perp X|B$
- $K = 1$: If $deg(X) < q$ for all relevant variables X: X is relevant iff $Imp_q(X) > 0$
- $K \geq 1$: If there are q or less relevant variables: X strongly relevant $\Rightarrow Imp_q(X) > 0$

Drawback: RS requires many trees to find high degree variables

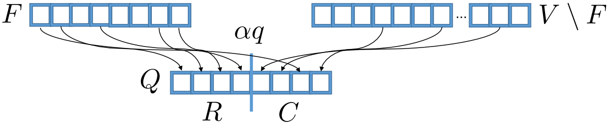
E.g.: $p = 10000, q = 50, k = 1 \Rightarrow \frac{\binom{p-k-1}{q-k-1}}{\binom{p}{q}} = 2.5 \cdot 10^{-5}$. In average, at least $T = 40812$ trees are required to find X.

Sequential Random Subspace (SRS)

Proposed algorithm:

1. Let $F = \emptyset$
2. Repeat T times:
   2.1 Let $Q = R \cup C$, where:
      - R is a subset of $\min\{\alpha q, |F|\}$ features randomly taken from F
      - C is a subset of $q - |R|$ features randomly selected in $V \setminus R$

   2.2 Grow a tree only using features in Q
   2.3 Add to F all features that get non-zero importance
3. Return F

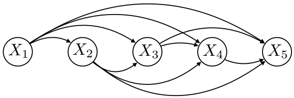


Compared to RS: *fill α% of the memory with previously found relevant variables and $(1-\alpha)$% with randomly selected variables.*

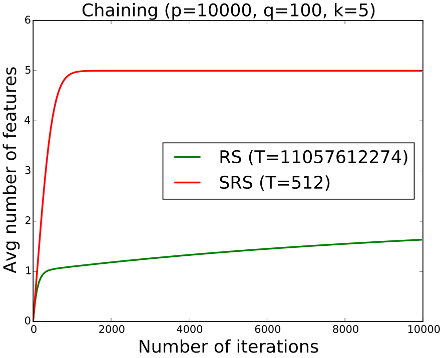
SRS for feature selection: study

Asymptotic guarantees: similar as RS if all relevant variables can fit into memory.

Convergence: SRS requires much less trees than RS in most cases. *For example,*



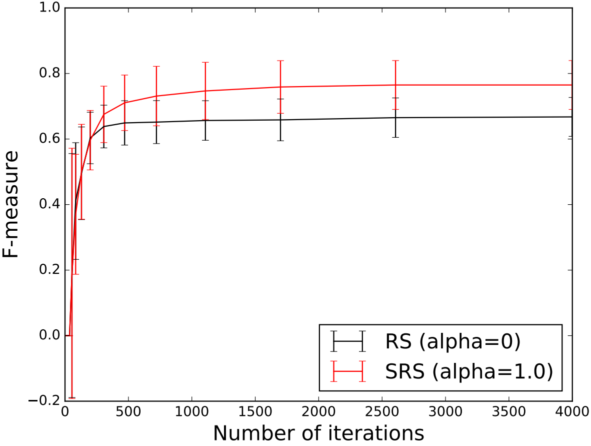
Numerical simulation



Experiments: results in feature selection

Dataset: Madelon (Guyon et al., 2007)

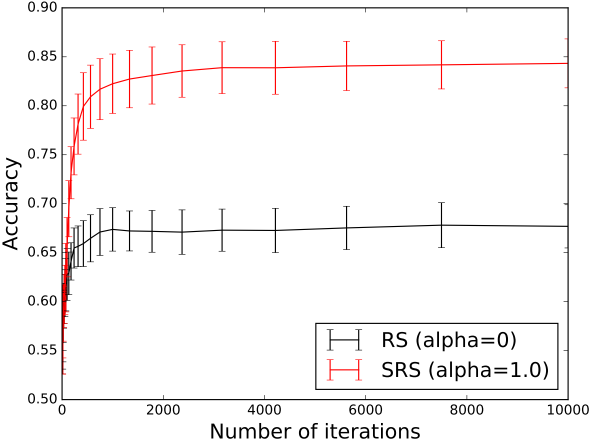
- 1500 samples (|LS|=1000, |TS|=500)
- 500 features whose 20 relevant features (5 features that define Y, 5 random linear combinations of the first 5, and 10 noisy copies of the first 10)



Parameter:

- q : 50

Experiments: results in prediction



Parameter:

- q : 50

After 10000 trees/iterations:

- RF ($K = max$): 0.81
- RF ($K = q$): 0.70
- RS : 0.68
- **SRS: 0.84**

Conclusions

Future works on SRS:

- Good performance of SRS are confirmed on other datasets but more experiments are needed.
- How to dynamically adapt K and α to improve correctness and convergence?
- Parallelization of each step or of the global procedure

Conclusion:

In most cases, accumulating relevant features speeds up the discovery of new relevant features while improving the accuracy.

References

Célia Châtel, *Sélection de variables à grande échelle à partir de forêtes aléatoires*, Master's thesis, École Centrale de Marseille/Université de Liège, 2015.

Gilles Louppe and Pierre Geurts, *Ensembles on random patches.*, ECML/PKDD (1) (Peter A. Flach, Tijl De Bie, and Nello Cristianini, eds.), Lecture Notes in Computer Science, vol. 7523, Springer, 2012, pp. 346–361.

Gilles Louppe, *Understanding random forests: From theory to practice*, Ph.D. thesis, University of Liège, 2014.

G. Louppe, L. Wehenkel, A. Sutera, and P. Geurts, *Understanding variable importances in forests of randomized trees*, Advances in neural information processing, 2013.