Random forests with random projections of the output space for high dimensional multi-label classification

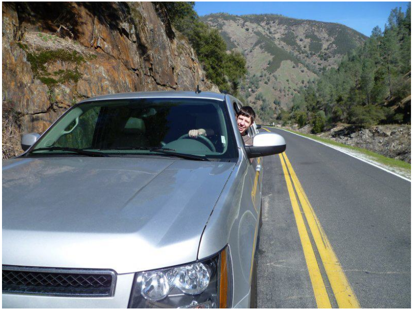
Arnaud Joly, Pierre Geurts, Louis Wehenkel

Université de Liège

Multi-label classification tasks

Many supervised learning applications in text, biology or image processing where samples are associated to sets of labels.

Input $\mathcal{X}$ 800×600 pixel



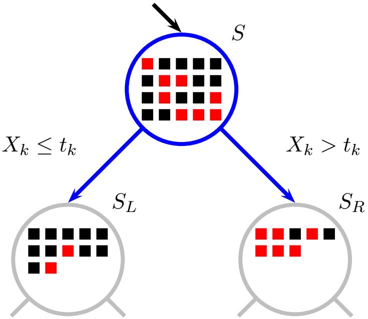
Output $\mathcal{Y}$ labels

driver, mountain, road, car, tree, rock, line, human, ...

If each label corresponds to a wikipedia article, then we have around 4 million labels.

Random forest

Randomized trees are built on a bootstrap copy of the input-output pairs $((x^i, y^i) \in (\mathcal{X} \times \mathcal{Y}))_{i=1}^{n}$ by recursively maximizing the reduction of impurity, here the variance Var. At each node, the best split is selected among k randomly selected features.



$$\mathrm{Var}(S) = 0.24$$

$$\mathrm{Var}(S_L) = 0.014$$

$$\mathrm{Var}(S_R) = 0.1875$$

$$\Delta\,\mathrm{Var}(S) = \mathrm{Var}(S) - \frac{12}{20}\,\mathrm{Var}(S_L) - \frac{8}{20}\,\mathrm{Var}(t_R)$$

$$\approx 0.16$$

When $\mathcal{Y}$ is very high dimensional, this constitutes the main bottleneck of the random tree ensemble.

The multi-output single tree algorithm requires the computation of the sum of the variance over the label space at each tree node and for each candidate split.

Multi-output regression trees in randomly projected output space

We propose to approximate the computation of the variance by using random projection of the output space.

Multi-output regression trees in randomly projected output space

We propose to approximate the computation of the variance by using random projection of the output space.

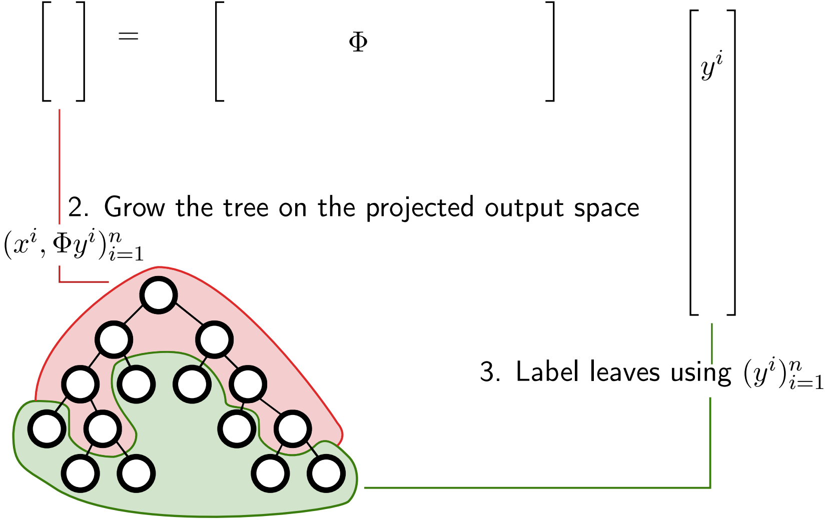
Theorem

Given $\epsilon > 0$, a sample $(y^i)_{i=1}^n$ of n points $y \in \mathbb{R}^d$, and a projection matrix $\Phi \in \mathbb{R}^{m \times d}$ such that for all pairs of points the Jonhson-Lindenstrauss lemma holds, we have also

$$(1-\epsilon)\,\mathrm{Var}\left((y^i)_{i=1}^n\right) \leq \mathrm{Var}\left((\Phi y^i)_{i=1}^n\right) \leq (1+\epsilon)\,\mathrm{Var}\left((y^i)_{i=1}^n\right).$$

Multi-output regression trees in randomly projected output space

1. Randomly project the output space

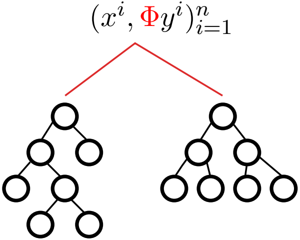


2. Grow the tree on the projected output space

3. Label leaves using $(y^i)_{i=1}^n$

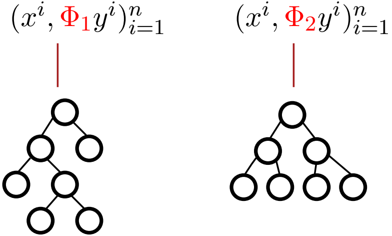
Ensemble of randomized trees

Shared subspace



$(x^i, \Phi y^i)_{i=1}^n$

Individual subspace



$(x^i, \Phi_1 y^i)_{i=1}^n$

$(x^i, \Phi_2 y^i)_{i=1}^n$

Bias-variance analysis

Averaging over the learning set LS, algorithm randomization ϵ and output subspace randomization Φ, the square error Err of t multi output tree models can be decomposed into:

Single shared subspace (Algo 1)

$$E_{LS,\Phi,\varepsilon^t}\{Err(f_1(x;LS,\Phi,\varepsilon^t))\}$$
$$= \sigma_R^2(x) + B^2(x) + V_{LS}(x) + \frac{V_{Algo}(x)}{t} + V_{Proj}(x).$$

Individual subspace (Algo 2)

$$E_{LS,\Phi^t,\varepsilon^t}\{Err(f_2(x;LS,\Phi^t,\varepsilon^t))\}$$
$$= \underbrace{\sigma_R^2(x)}_{\text{residual error}} + \underbrace{B^2(x)}_{\text{bias}} + \underbrace{V_{LS}(x) + \frac{V_{Algo}(x) + V_{Proj}(x)}{t}}_{\text{variance}}.$$

Individual subspace should always be preferred to single shared subspace.

Label ranking average precision to assess performance

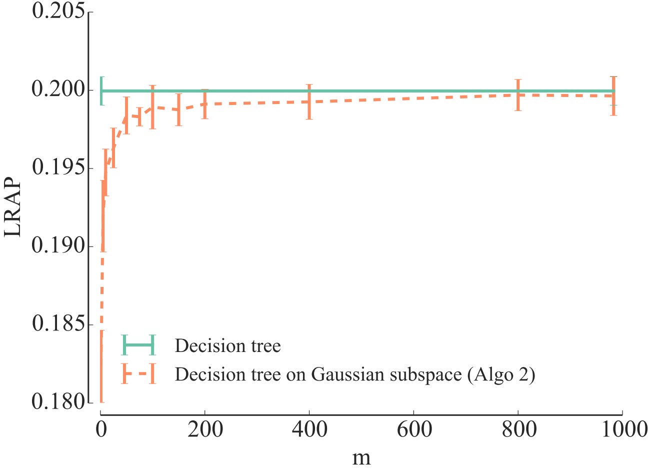
$$\mathsf{LRAP}(\hat{f}) = \frac{1}{|TS|} \sum_{i \in TS} \frac{1}{|y^i|} \sum_{j \in \{k : y^i_k = 1\}} \frac{|\mathcal{L}^i_j(y^i)|}{|\mathcal{L}^i_j(1_d)|},$$

$$\mathcal{L}^i_j(q) = \left\{ k : q_k = 1 \text{ and } \hat{f}(x^i)_k \geq \hat{f}(x^i)_j \right\}$$

where $\hat{f}(x^i)_j$ is the probability (or the score) associated to the label j by the learnt model $\hat{f}$ applied to x^i, 1_d is a d-dimensional row vector of ones.

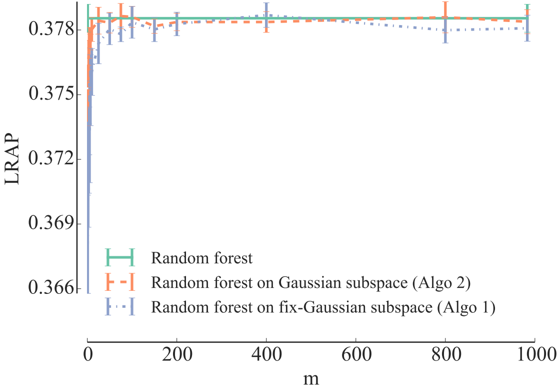
Higher score if true labels have a higher probability (score) than the false labels.

Decision tree performance converges with $m = 200$ Gaussian random output projections



Delicious dataset (983 labels)

Faster convergence with ensemble of randomized trees

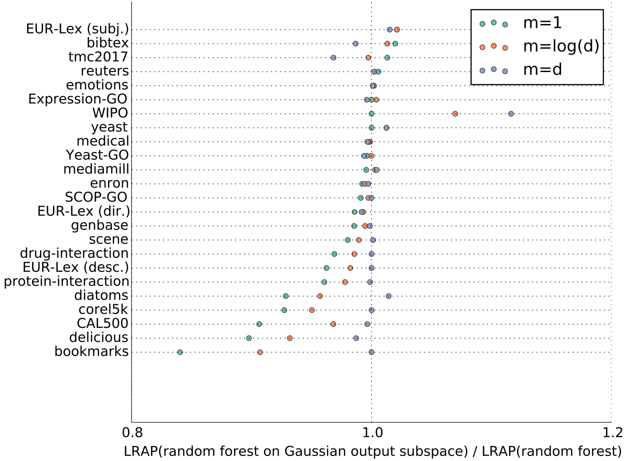


Delicious dataset (983 labels, $k = \sqrt{p}$, $t = 100$, $n_{\min} = 1$)

Randomly projecting the output space reduces computing time from 3458 seconds (no projection) to 311 seconds ($m = 25$, individual subspace) without accuracy degradation.

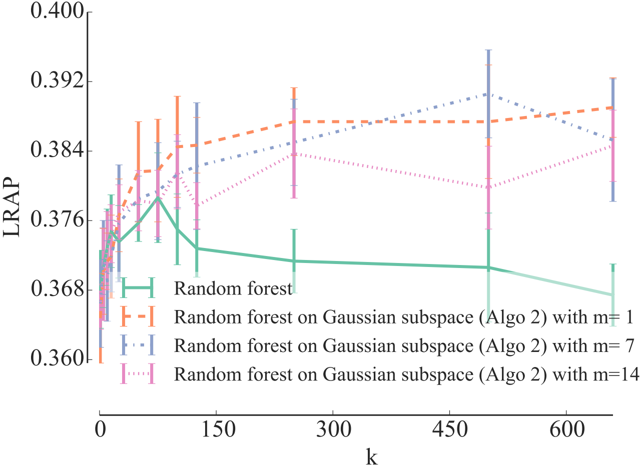
Systematic analysis on 24 datasets

Increasing m leads to convergence in LRAP



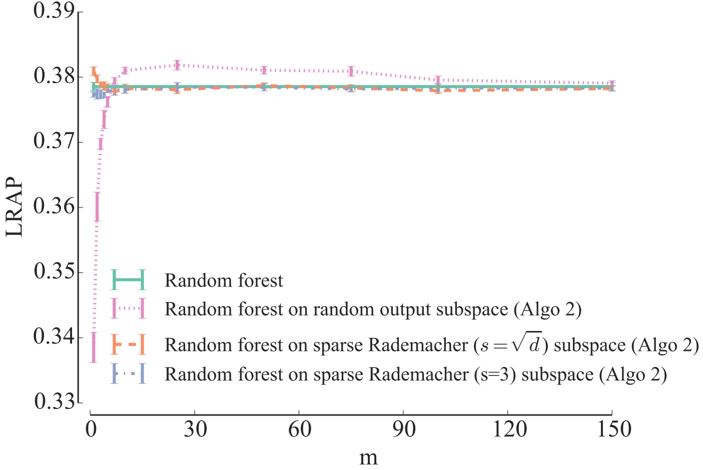
$(k = \sqrt{p}$, $t = 100$, $n_{\min} = 1$, averaged over 10 repetitions)

Output randomization could be more effective than input randomization



Drug-interaction dataset(1554 labels, $t = 100$, $n_{\min} = 1$)

Alternative random output subspace



Delicious dataset(981 labels, $k = \sqrt{p}$, $t = 100$, $n_{\min} = 1$)

Random forests with random projections of the output space for high dimensional multi-label classification

Conclusions

- Lower computing time, without affecting accuracy.
- Optimizing input and output randomization could improve prediction performance.

Future work

Efficient technique to adjust random output space parameters so as to reach the best accuracy and computing time trade-off.

Source code is available `@`
`github.com/arjoly/random-output-trees`.