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

Arnaud Joly, Pierre Geurts, Louis Wehenkel
Multi-label classification tasks

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

Input $\mathbf{X}$ $800 \times 600$ pixel

Output $\mathbf{Y}$ 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 \(\text{Var}\). At each node, the best split is selected among \(k\) randomly selected features.

\[
\begin{align*}
\text{Var}(S) &= 0.24 \\
\text{Var}(S_L) &= 0.014 \\
\text{Var}(S_R) &= 0.1875 \\
\Delta \text{Var}(S) &= \text{Var}(S) - \frac{12}{20} \text{Var}(S_L) - \frac{8}{20} \text{Var}(t_R) \\
&\approx 0.16
\end{align*}
\]
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.
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 Johnson-Lindenstrauss lemma holds, we have also

$$(1 - \epsilon) \text{Var}(y_i)_{i=1}^n \leq \text{Var}(\Phi y_i)_{i=1}^n \leq (1 + \epsilon) \text{Var}(y_i)_{i=1}^n.$$
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

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

(1 - \epsilon) \text{Var} ((y^i_{i=1}^n)) \leq \text{Var} ((\Phi y^i_{i=1}^n)) \leq (1 + \epsilon) \text{Var} ((y^i_{i=1}^n)).
\end{align*}
Multi-output regression trees in randomly projected output space

1. Randomly project the output space

\[
\begin{bmatrix}
\end{bmatrix}
= \begin{bmatrix}
\Phi
\end{bmatrix}
\]

2. Grow the tree on the projected output space

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

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 $\epsilon$ and output subspace randomization $\Phi$, the square error $Err$ of $t$ multi output tree models can be decomposed into:

**Single shared subspace (Algo 1)**

$$E_{LS, \Phi, \epsilon^t} \{Err(f_1(x; LS, \Phi, \epsilon^t))\}$$

$$= \sigma^2_R(x) + B^2(x) + V_{LS}(x) + \frac{V_{Algo}(x)}{t} + V_{Proj}(x).$$

**Individual subspace (Algo 2)**

$$E_{LS, \Phi^t, \epsilon^t} \{Err(f_2(x; LS, \Phi^t, \epsilon^t))\}$$

$$= \sigma^2_R(x) + B^2(x) + V_{LS}(x) + \frac{V_{Algo}(x) + V_{Proj}(x)}{t}.$$  

Individual subspace should always be preferred to single shared subspace.
Label ranking average precision to assess performance

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

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

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
Faster convergence with ensemble of randomized trees

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

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

$$LRAP(\text{random forest on Gaussian output subspace}) / LRAP(\text{random forest})$$

$$(k = \sqrt{p}, t = 100, n_{\text{min}} = 1, \text{averaged over 10 repetitions})$$
Output randomization could be more effective than input randomization

Drug-interaction dataset (1554 labels, $t = 100$, $n_{\text{min}} = 1$)
Alternative random output subspace

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