Random Forest

Random Forest

Random Forest – is bagging of *de-correlated* decision trees.

Algorithm: Random Forest

Input: training set $\boldsymbol{Z}=\{(\boldsymbol{x}_1, y_1),\ldots,(\boldsymbol{x}_n, y_n)\}$,
B – number of iterations

1. For b=1…B:
2. Draw a bootstrap sample $\boldsymbol{Z}^*$ of size n from training data
3. Grow a random forest (**de-correlated**) tree T_b to the $\boldsymbol{Z}^*$
4. Return: ensemble $\{T_1 \ldots T_B\}$

Prediction with decision trees:

- Regression: $\mathbf{f(x)}=\sum_{b=1}^{B} T_b(\boldsymbol{x})$
- Classification: majority vote of all decision trees predictions $T_b(\boldsymbol{x})$, $b=1 \ldots B$

How to grow a *random forest* decision tree

1. The tree is built **greedily** from top to bottom.

2. Select m ≤ p of the input variables at random as candidates for splitting.

3. Each split is selected to **maximize information gain (IG)**.

$$IG = \underbrace{\mathbf{Impurity}(Z)}_{\text{Error before split}} - \underbrace{\left(\frac{|Z_L|}{|Z|}\mathbf{Impurity}(Z_L) + \frac{|Z_R|}{|Z|}\mathbf{Impurity}(Z_R)\right)}_{\text{Error after split}}$$

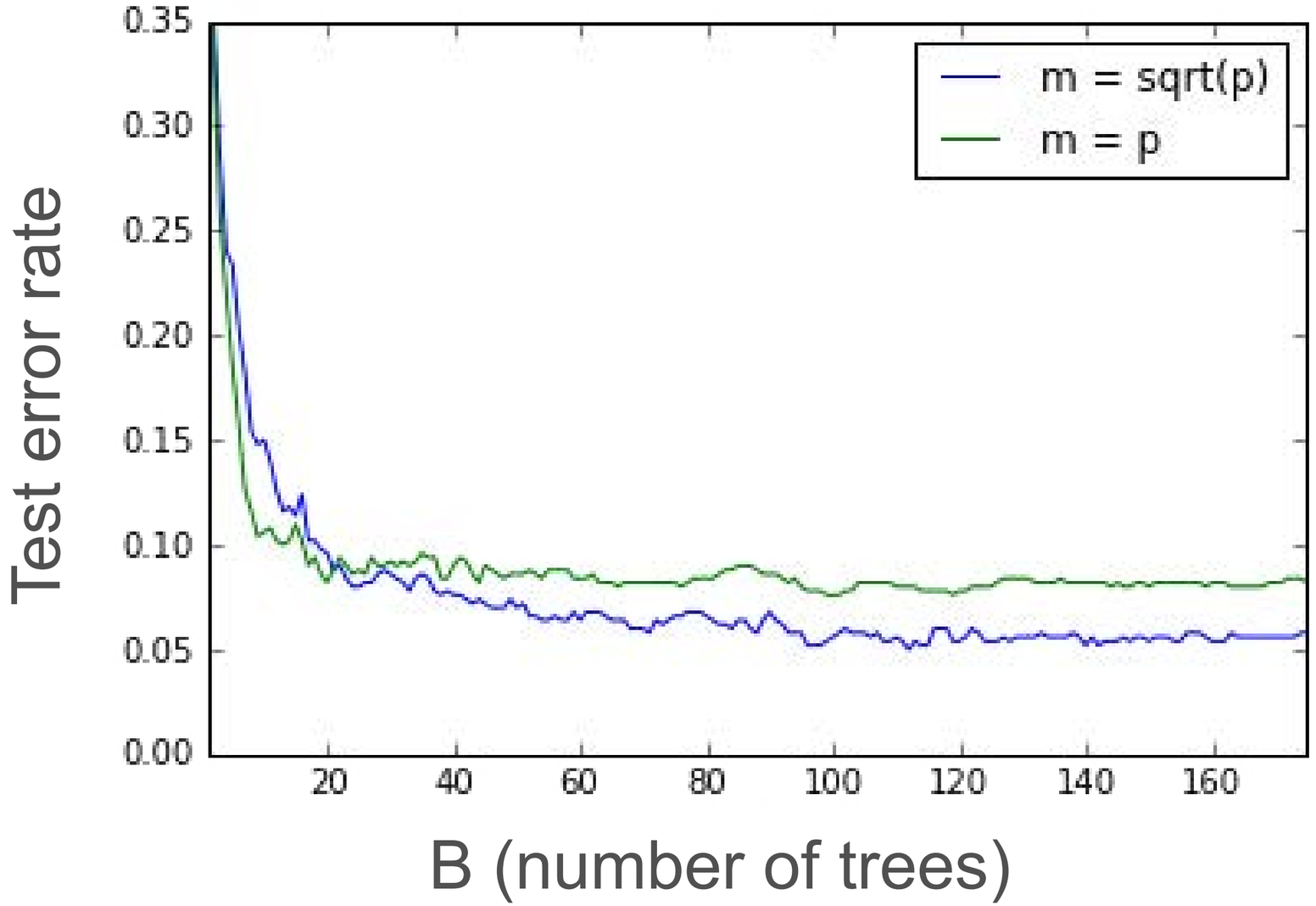
Select $m \leq p$ of the input variables at random as candidates for splitting.

Recommendations from inventors of Random Forests:

$m=\sqrt{p}$ for classification, minInstancesPerNode = 1

$m=p/3$ for regression, minInstancesPerNode = 5

Random Forest



Summary

- **Random Forest** is a good method for a general purpose classification/regression problems (typically slightly worse than gradient boosted decision trees)
- Automatically handling interactions of features
- Computational scalability (!)
- Predictive power
- Interpretability