Ensemble Methods: Bagging & Random Forests

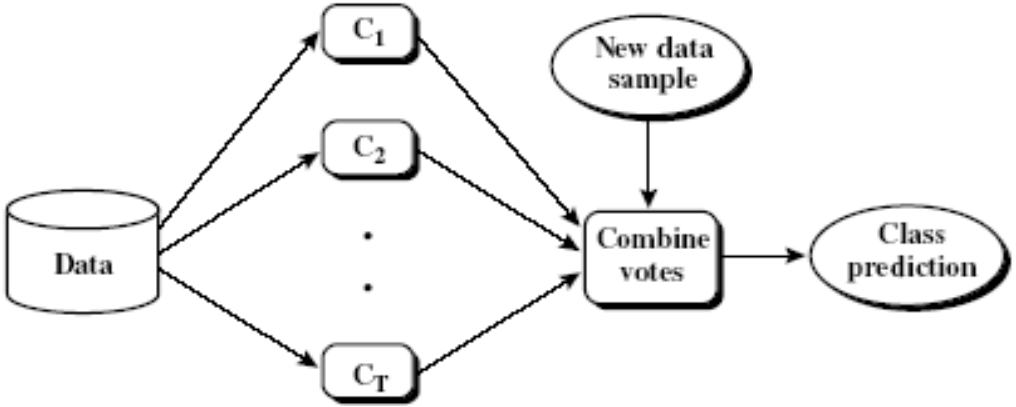
B. Ravindran

What are Ensemble Methods?

- Real - world problems often do not have a unique “correct” solution
- Multiple thought processes and teams and approaches often yield equally valid but completely different solutions.
- Combining different approaches using some heuristic could lead to a meta-learning method that utilizes the best of different approaches.
- Such a set of methods for combining many solutions (classifiers, regressors etc.) into one “better” solution are known as “**Ensemble Methods”**

Ensemble methods

- Uses a combination of N learned models: C_1, C_2, … C_N with the aim of creating an improved model C*
- Ensemble learning, more often than not, decreases variance of the predictors and allows better generalization



Paradigms of Ensemble Learning

Parallel Ensembles

- Combines several models, built separately to fit the data
- Allows naive parallelisation (many parallel models)
- Example: **Bagging** etc.

Paradigms of Ensemble Learning

Parallel Ensembles

- Combines several models, built separately to fit the data
- Allows naive parallelisation (many parallel models)
- Example: **Bagging** etc.

Sequential Ensembles

- Each model built, explicitly uses other model predictions sequentially
- Every model improves upon the weaknesses of previous models
- Example: **Boosting** etc.

Committee Methods

- Individual learners are often “high capacity”, with a tendency to overfit
- High variance in such learners reduces their generalization capability
- Ensemble methods aim to reduce variance → counter overfitting

Committee Methods

- Individual learners are often “high capacity”, with a tendency to overfit
- High variance in such learners reduces their generalization capability
- Ensemble methods aim to reduce variance → counter overfitting
- **Committees:**
  - Group of N individual models: C_1, C_2, … C_N
  - makes predictions by averaging over predictions of individual models
  - Motivated by frequentist perspective of bias and variance
- How does averaging predictions reduce variance?

Error Reduction

- Regression: Predict real-valued function **h(x)**
- Input: **x** = $(x_1, \ldots, x_p)^T$ where each $x_i \in \mathbb{R}$
- N models predicting outputs $\mathbf{y_i(x)}$ for i=1,...,N

$$y_i(\mathbf{x}) = h(\mathbf{x}) + \epsilon_i(\mathbf{x})$$

- Committee prediction: averaging

$$y_{\mathrm{COM}}(\mathbf{x}) = \sum_{i=1}^{N} y_i(\mathbf{x})$$

Error Reduction

The average sum-of-squares error of all N models:

$$E_{\text{avg}} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}_x \left[\{ h(\mathbf{x}) - y_i(\mathbf{x}) \}^2 \right]$$

$$= \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}_x \left[\epsilon_i(\mathbf{x})^2 \right]$$

Error Reduction

The sum-of-squares error for the committee prediction is:

$$E_{\mathrm{COM}} = \mathbb{E}_x\left[\left\{h(\mathbf{x}) - \frac{1}{N}\sum_{i=1}^{N} y_i(\mathbf{x})\right\}^2\right]$$

$$= \mathbb{E}_x\left[\left\{\frac{1}{N}\sum_{i=1}^{N} h(\mathbf{x}) - y_i(\mathbf{x})\right\}^2\right]$$

$$= \mathbb{E}_x\left[\left\{\frac{1}{N}\sum_{i=1}^{N} \epsilon_i(\mathbf{x})\right\}^2\right]$$

Error Reduction

- Assumption 1: Classifiers have zero bias

$$\mathbb{E}_x[y_i(\mathbf{x})] = h(\mathbf{x}) \implies \mathbb{E}_x[\epsilon_i(\mathbf{x})] = 0$$

- Assumption 2: Classifiers are uncorrelated

$$\mathbb{E}_x\left[\epsilon_i(\mathbf{x})\epsilon_j(\mathbf{x})\right] = 0 \quad \text{for all } i \neq j$$

Error Reduction

The committee sum-of-squares error is reduced to:

$$
\begin{aligned}
E_{\mathrm{COM}} &= \mathbb{E}_x\left[\left\{\frac{1}{N}\sum_{i=1}^{N}\epsilon_i(\mathbf{x})\right\}^2\right] \\
&= \mathbb{E}_x\left[\frac{1}{N^2}\cdot\left\{\sum_{i=1}^{N}\epsilon_i(\mathbf{x})^2 + \sum_{i\neq j}\epsilon_i(\mathbf{x})\epsilon_j(\mathbf{x})\right\}\right] \\
&= \frac{1}{N}\cdot\mathbb{E}_x\left[\frac{1}{N}\sum_{i=1}^{N}\epsilon_i(\mathbf{x})^2\right] \\
&= \frac{1}{N}\cdot E_{\mathrm{avg}}
\end{aligned}
$$

Error Reduction

This shows us that committees reduce variance in an ideal world using uncorrelated learners.

What if we introduce correlation? (real world)

Error Reduction

This shows us that committees reduce variance in an ideal world using uncorrelated learners.

What if we introduce correlation? (real world)

Recall, correlation from statistics:

$$\text{corr}(X, Y) = \frac{\text{cov}(X, Y)}{\sqrt{\text{var}(X) \cdot \text{var}(Y)}}$$

where: cov() and var() refer to the covariance and variance respectively

Error Reduction

The variance of classifier $\mathbf{y_i(x)}$ is given by:

$$\begin{aligned}\mathrm{var}(y_i(\mathbf{x})) &= \mathbb{E}_x\left[\left\{y_i(\mathbf{x}) - \mathbb{E}_x\left[y_i(\mathbf{x})\right]\right\}^2\right] \\ &= \mathbb{E}_x\left[\left\{y_i(\mathbf{x}) - h(\mathbf{x})\right\}^2\right] \\ &= \mathbb{E}_x\left[\epsilon_i(\mathbf{x})^2\right]\end{aligned}$$

Error Reduction

The variance of classifier $\mathbf{y_i(x)}$ is given by:

$$\begin{aligned}\text{var}(y_i(\mathbf{x})) &= \mathbb{E}_x\left[\left\{y_i(\mathbf{x}) - \mathbb{E}_x\left[y_i(\mathbf{x})\right]\right\}^2\right] \\ &= \mathbb{E}_x\left[\left\{y_i(\mathbf{x}) - h(\mathbf{x})\right\}^2\right] \\ &= \mathbb{E}_x\left[\epsilon_i(\mathbf{x})^2\right]\end{aligned}$$

Similarly, we have covariance between two classifiers:

$$\text{cov}(y_i(\mathbf{x}), y_j(\mathbf{x})) = \mathbb{E}_x\left[\epsilon_i(\mathbf{x})\ \epsilon_j(\mathbf{x})\right]$$

Error Reduction

Discard Assumption 2 (no correlation). However we make the simplifying assumption that each classifier is equivariate.

Let pairwise correlation between 2 classifiers be denoted by $\boldsymbol{\rho}$>0. Then rearranging the correlation equation we get:

$$\mathbb{E}_x\left[\epsilon_i(\mathbf{x})\ \epsilon_j(\mathbf{x})\right] = \rho \cdot \mathbb{E}_x\left[\epsilon_i(\mathbf{x})^2\right]$$

Error Reduction

Reducing the sum-of-squares error for the committee predictions:

$$
\begin{aligned}
E_{\mathrm{COM}} &= \mathbb{E}_x\left[\frac{1}{N^2}\cdot\left\{\sum_{i=1}^{N}\epsilon_i(\mathbf{x})^2+\sum_{i\neq j}\epsilon_i(\mathbf{x})\epsilon_j(\mathbf{x})\right\}\right] \\
&= \mathbb{E}_x\left[\frac{1}{N^2}\cdot\left\{\sum_{i=1}^{N}\epsilon_i(\mathbf{x})^2+\rho\cdot\sum_{i\neq j}\epsilon_i(\mathbf{x})^2\right\}\right] \\
&= \frac{1}{N}\cdot\mathbb{E}_x\left[\frac{1}{N}\sum_{i=1}^{N}\epsilon_i(\mathbf{x})^2\right]+\rho\cdot\frac{N-1}{N}\cdot\mathbb{E}_x\left[\frac{1}{N}\sum_{i=1}^{N}\epsilon_i(\mathbf{x})^2\right] \\
&= \frac{1}{N}\cdot E_{\mathrm{avg}}+\frac{N-1}{N}\cdot\rho\cdot E_{\mathrm{avg}} \\
&= \left(\rho+\frac{1-\rho}{N}\right)\cdot E_{\mathrm{avg}}
\end{aligned}
$$

Error Reduction

The sum-of-squares error in committee predictions $\mathbf{E_{COM}}$ based on N unbiased learners with average sum-of-squares error $\mathbf{E_{avg}}$ and a pairwise correlation $\boldsymbol{\rho}$>0 is:

$$E_{\mathrm{COM}} = \rho\, E_{\mathrm{avg}} + (1-\rho)\frac{E_{\mathrm{avg}}}{N}$$

- $\boldsymbol{\rho}$ = 0 : Fully uncorrelated ⇒ $\boldsymbol{\sigma}^2$/N **(Maximum Error Reduction)**
- $\boldsymbol{\rho}$ = 1 : Fully correlated ⇒ $\boldsymbol{\sigma}^2$ **(No Error Reduction)**

As N → ∞, only the first term (due to correlation) remains.

Note that for any $\boldsymbol{\rho}$:

$$E_{\mathrm{COM}} \leq E_{\mathrm{avg}}$$

Parallel Ensembles: Bagging

Modification of committee methods that uses copies of the same learner each trained on multiple bootstrapped datasets to introduce variability.

The name of this method comes from its steps: **B**ootstrap **Agg**regat**ing**

Bootstrapping + Aggregating = Bagging

- **Bootstrapping:** refers to bootstrap sampling ie. **sampling with replacement**
- **Aggregating:** refers to the aggregation of individual learner predictions into a composite prediction

Bagging

- **Bootstrapping**
  - Create a new set of N datasets: $\mathbb{D}_1$, $\mathbb{D}_2$, … $\mathbb{D}_N$ each of size n’ = n,
  - sampled randomly with replacement from the original dataset
  - Why replacement ? - Allows $\mathbb{D}_1$, … $\mathbb{D}_N$ to be independent of previous sampling

Bagging

- **Bootstrapping**
  - Create a new set of N datasets: $\mathbb{D}_1$, $\mathbb{D}_2$, … $\mathbb{D}_N$ each of size n’ = n,
  - sampled randomly with replacement from the original dataset
  - Why replacement ? - Allows $\mathbb{D}_1$, … $\mathbb{D}_N$ to be independent of previous sampling
- **Training:** Each model C_i is trained independently on bootstrapped datasets $\mathbb{D}_i$

Bagging

- **Bootstrapping**
  - Create a new set of N datasets: $\mathbb{D}_1$, $\mathbb{D}_2$, … $\mathbb{D}_N$ each of size n' = n,
  - sampled randomly with replacement from the original dataset
  - Why replacement ? - Allows $\mathbb{D}_1$, … $\mathbb{D}_N$ to be independent of previous sampling
- **Training:** Each model C_i is trained independently on bootstrapped datasets $\mathbb{D}_i$
- **Aggregation**
  - Classify a data point: it is independently classified by every C_i to give a prediction
  - C* then aggregates these predictions into a single prediction
  - Aggregating predictions: Majority voting, averaging class probabilities etc.

Bagging

- For noisy data → significantly more robust
- More **stable classification**
- **Caveat:** Bagging poor classifiers (high bias) could cause performance to become arbitrarily worse!
- Works especially well for:
  - High capacity (low bias) learners with a tendency to overfit (high variance)
  - Unstable learners which are very data-dependent
- **Examples: ?**

Bagging

- For noisy data → significantly more robust
- More **stable classification**
- **Caveat:** Bagging poor classifiers (high bias) could cause performance to become arbitrarily worse!
- Works especially well for:
  - High capacity (low bias) learners with a tendency to overfit (high variance)
  - Unstable learners which are very data-dependent
- **Examples:** DECISION TREES

Why Decision trees?

- Are high capacity learners
- Have a tendency to overfit on data
- Are difficult to regularize
- Are highly unstable learners

Bagged decision trees are themselves very useful. However, they can be improved further!

How?

Random Forests

- We have established that decision trees are naturally very good learners for use in bagging
- Bagging reduces variance ⇒ When is this most effective?

Random Forests

- We have established that decision trees are naturally very good learners for use in bagging
- Bagging reduces variance ⇒ When is this most effective?
- **When learners are not very correlated!**
- Highly correlated learners will often learn similar things and give similar predictions - bagged classifier will often also have similar predictions

Random Forests

We need to bag decision trees and attempt to reduce correlation at the same time!

Recall that we use p-dimensional data. For every data point, there are p features.

How can we reduce correlation between classifiers?

Random Forests

We need to bag decision trees and attempt to reduce correlation at the same time!

Recall that we use *p*-dimensional data. For every data point, there are *p* features.

How can we reduce correlation between classifiers?

- At every step of growing the decision tree do the following:
  - Randomly pick only *t* out of the *p* features available
  - Pick the feature out of these *t* which offers the best split
  - Use it to grow the tree.
  - Repeat again with a new randomly picked set of *t* features
  - Break when further refinement is not possible

- Typical values of *t* could be: $p/2$, $\sqrt{p}$ etc.

Random Forests

The two important parameters of the RF algorithm are:

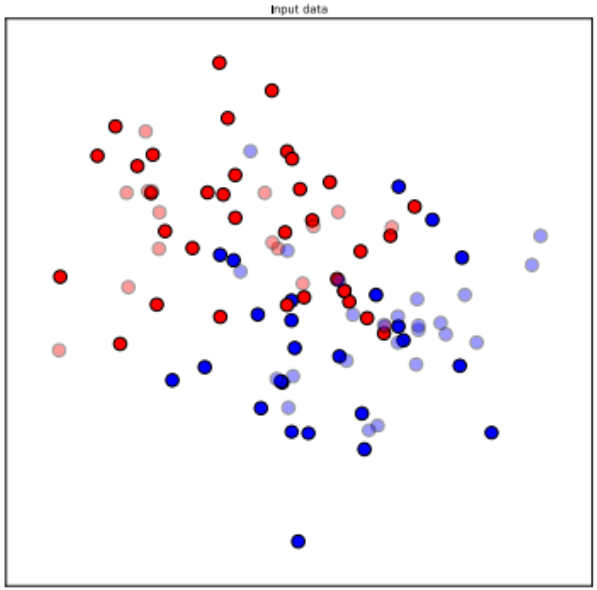
- **Number of trees (n_est)**

  RF builds a number of decision trees and fits them onto the different sampled datasets. As the number of learners used increases, the variance tends to decrease until it saturates.

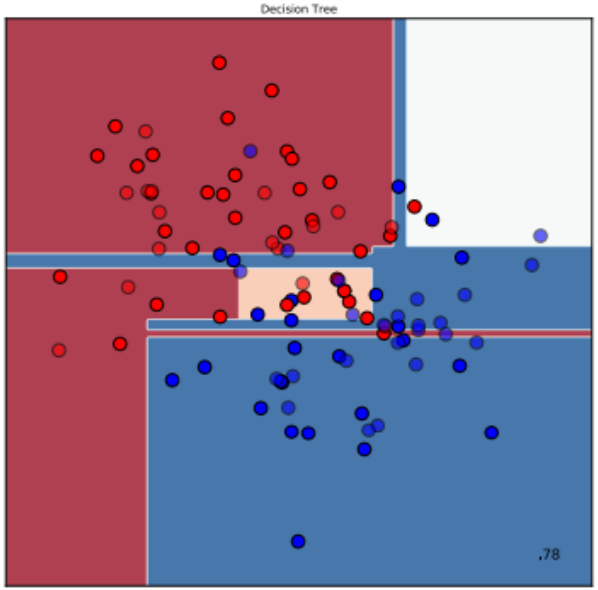
- **Maximum features at a node (t)**

  As discussed in previous slide, the reduction in number of features used for splitting the decision tree at every node leads to uncorrelated trees. Very large values of t ($t \rightarrow p$) tends to give very similar (correlated) trees.

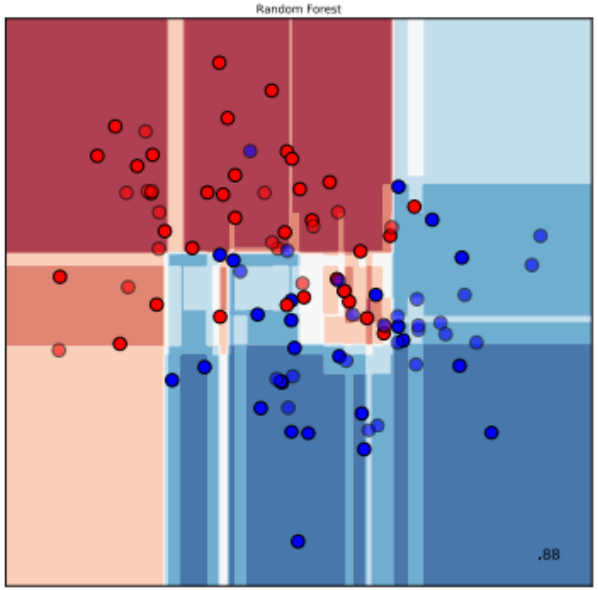
Random Forests



Decision tree



Random forest



Accuracy: **78%** **88%**

Random Forests

- RF can give competitive results with gradient boosted trees
- Number of high performance libraries made it a very popular algorithm!
- In recent years however, there is an advent of high performance gradient boosting algorithms: XGBOOST, LightGBM etc.
- Try a combination of these to see which works best for your application!

Summary

- Ensemble methods - Types - Why are they used
- Committee Methods
  - Error reduction by averaging predictions
  - Effect of correlation between learners on error reduction
- Bagging (Bootstrap Aggregating)
  - Bootstrapping to generate multiple slightly different datasets from same data
  - Aggregating classifiers trained on different to give a single prediction
  - Useful for low bias, high variance unstable learners - Decision Trees
- Random Forests
  - Bagged Decision trees with random feature selection at nodes!
  - Random Forests algorithm - building a forest of trees with less correlation