Machine Learning
Beyond Linearity

Rajesh Ranganath

Goal: Understand influence of Vitamin D on Health

Vitamin D

- Used in bone health
- Function of the immune system
- Muscle function

For the observed

$$(x_1, y_1), \ldots, (x_n, y_n)$$

There's a positive correlation.

What would a linear model trained say?

For the observed

$$(x_1,y_1),\ldots,(x_n,y_n)$$

There's a positive correlation.

What would a linear model trained say?

- More vitamin D is better
- Give all people infinite vitamin D

Deploy this model. People take lots of vitamin D

Deploy this model. People take lots of vitamin D

Too much vitamin D? Get Hypervitaminosis-D

- High blood concentrations of calcium
- Calcification of heart
- Calcification of kidney

Not great

Linear model says

- The relationship is always positive

- The relationship is always negative

- The relationship is zero

What about for vitamin D?

Linear model says

- The relationship is always positive
- The relationship is always negative
- The relationship is zero

What about for vitamin D?

Too-little is bad. Too-much is bad. Doesn't fit

Linear Model Bandaid

Linear models can be non-linear

$$y_i = \theta_1 x_{i,1} + \theta_0 + \epsilon_i$$

Linear Model Bandaid

Linear models can be non-linear

$$y_i = \theta_1 x_{i,1} + \theta_0 + \epsilon_i$$

Bandaid with more features

$$y_i = \theta_1 x_{i,1} + \theta_2 x_{i,1}^2 + \theta_0 + \epsilon_i$$

- Model now a quadratic (non-linear) function of x_1
- Still a linear problem in this new feature space
- Parameters can be learned with generalized linear models

Linear Model Bandage

Linear models can be become non-linear

1. Adding higher order features to the feature matrix
2. Estimate the parameters of the augmented model

Vitamin K also influences the blood calcium system

Goal: Understand influences of Vitamins D and K on health?

A linear model

$$y_i = \theta_1 x_{i,vitamin-d} + \theta_2 x_{i,vitamin-k} + \theta_0 + \epsilon_i$$

Can fix this model to handle non-linear vitamin-D by

$$y_i = \theta_1 x_{i,vitamin-d} + \theta_3 x_{i,vitamin-d}^2 + \theta_2 x_{i,vitamin-k} + \theta_0 + \epsilon_i$$

But what about how vitamin k and vitamin d interact?

Linear Model: More Bandages

Add a new feature by multiplying $x_{i,vitamin-d}x_{i,vitamin-k}$

$$\begin{aligned} y_i = {} & \theta_1 x_{i,vitamin-d} + \theta_3 x^2_{i,vitamin-d} + \theta_2 x_{i,vitamin-k} + \theta_0 + \epsilon_i \\ & + \theta_4 x_{i,vitamin-d} x_{i,vitamin-k} \end{aligned}$$

- Include interaction between
- Model still can be trained by generalized linear models

General Recipe

1. Precompute feature combinations
2. Train a generalized linear model

Problems?

General Recipe

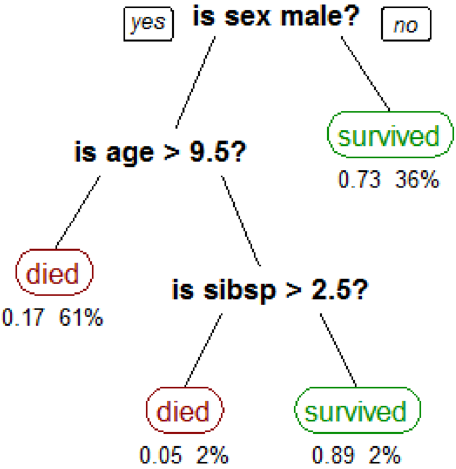
1. Precompute feature combinations
2. Train a generalized linear model

Problems?

- Which interaction terms?
- Scales very poorly with dimensionality
- Computational and statistical problems!

Want new models with non-linearities!

Decision Trees



[Wikipedia, Decision_tree_learning]

Decision Trees

- Work by dividing the feature space into different regions
- Different predictions in each region
- Can be done for classification or regression

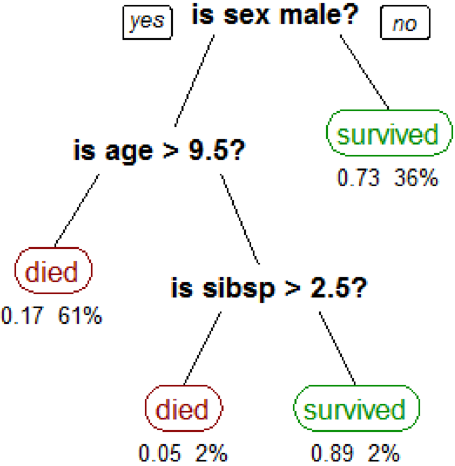
Decision Trees

- Work by dividing the feature space into different regions
- Different predictions in each region
- Can be done for classification or regression

$$pred(\mathbf{x}^*) = \sum_{i=1}^{N_{leaves}} \mathbb{1}[\mathbf{x}^* \in leaf_i]\mathbb{E}[y \mid \mathbf{x} \in leaf_i]$$

Where $leaf_i$ is intersection of sets along the path to the leaf

Decision Trees



[Wikipedia, Decision_tree_learning]

Decision Trees

How do you learn a decision tree?

Decision Trees

How do you learn a decision tree?

- Is there an optimal tree?

Decision Trees

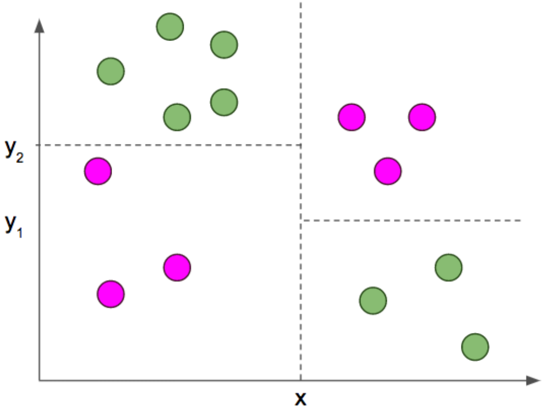
How do you learn a decision tree?

- Is there an optimal tree?

- Too many possible split orderings

Use a greedy algorithm

Greedy Tree Building: Basic Idea



[pythonmachinelearning.pro]

1. Figure out a rule to split the input space
2. Split the space according to the split from (1)
3. Start over again in each new subspace

Classification Tree

Assume all features $\mathbf{x}$ are continuous

For each feature j

1. Find a split point c_j that minimizes classification error in two subsets.

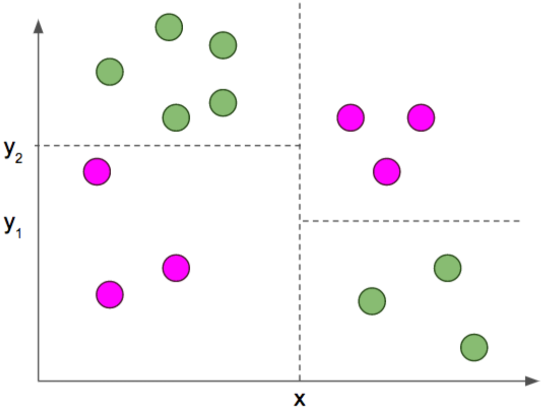
   - □ Prediction

   $$\text{prediction}(A_j) = \text{majority}(\mathbf{x}_j \in A_j)$$

   - □ Error on split

   $$\begin{aligned}\min_{c_j} \frac{1}{n} \sum_{i=1}^{n} & \mathbb{1}[\text{prediction}(x_{i,j} \leq c_j) \neq y_i]\mathbb{1}[x_{i,j} \leq c_j] \\ & + \mathbb{1}[\text{prediction}(x_{i,j} > c_j) \neq y_i]\mathbb{1}[x_{i,j} > c_j]\end{aligned}$$

2. Choose the feature j^* with the lowest classification error and split the data at c_j^*



[pythonmachinelearning.pro]

Classification Tree

Assume all features **x** are continuous

Error on split

$$\min_{c_j} \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}[\text{prediction}(x_{i,j} \leq c_j) \neq y_i] \mathbb{1}[x_{i,j} \leq c_j] \\ + \mathbb{1}[\text{prediction}(x_{i,j} > c_j) \neq y_i] \mathbb{1}[x_{i,j} > c_j]$$

Why is this the right error?

Aside: Information Measures

- Entropy:

$$\mathscr{H}(y) = -\mathbb{E}[\log p(y)]$$

  Measure of unpredictability of y

Aside: Information Measures

- Entropy:

$$\mathscr{H}(y) = -\mathbb{E}[\log p(y)]$$

  Measure of unpredictability of y

- Mutual Information:

$$MI(x;y) = \mathbb{E}_{p(x,y)}\left[\log \frac{p(x,y)}{p(x)p(y)}\right]$$

  Measure of dependence between x and y

Aside: Information Measures

Kullback-Leibler Divergence

$$KL(p||q) = \mathbb{E}_p \left[\log \frac{p}{q} \right]$$

Compares two distributions

- $KL(p||q) \geq 0$
- $KL(p||q) = 0$ if $p = q$
- Minimizing KL divergence relates to maximum likelihood

Classification error is one type of error. Many others possible

- Conditional Entropy

$$\mathscr{H}(y|\mathbb{1}[x_{i,j} \leq c_j])$$

Classification error is one type of error. Many others possible

- Conditional Entropy

$$\mathscr{H}(y|\mathbb{1}[x_{i,j} \leq c_j])$$

- Mutual Information. Define:

$$a_j = \mathbb{1}[x_{i,j} \leq c_j]$$

Compute mutual information

$$MI(y; a_j) = \mathbb{E}_{p(a_j,y)}\left[\log \frac{p(a_j,y)}{p(a_j)p(y)}\right]$$

Larger values means split has more information about labels

- Many other choices

Each choice places different model assumptions

How can we use the same ideas to do regression?

How can we use the same ideas to do regression?

Find splits that minimize the regression error in each split

What happens if we keep splitting?

What happens if we keep splitting?

- Eventually little data in each subsplit
- Predictions depend on the closest data point

Will this work well?

It will overfit. Fix by

- Requiring multiple data points in each leaf
- A test for significance in the improvement
- Pruning the tree using test data?

It will overfit. Fix by

- Requiring multiple data points in each leaf
- A test for significance in the improvement
- Pruning the tree using test data?

Is everything okay now?

Greedy algorithms can be sensitive

- Small changes in the data can create different predictions
- If two features are similar?
- Also when data in each region gets small

We want a way to learn trees that

- Less prone to overfitting
- Robust to perturbations in the data

Idea: Average lots of randomly perturbed trees together

- Robust to perturbations by averaging over them
- Adding more trees can't overfit

Random Forests

Start with a way to build a random tree

$$h(\mathbf{x}, \boldsymbol{\theta}_k), \text{ where } \boldsymbol{\theta}_k \sim q$$

How do we introduce randomness?

- Build trees from different data subsamples
- Random select the feature during construction of the tree

No explicit construction for q. However we can sample

Aside: Generalization Error

Suppose we have n data points

$$(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)$$

from p.

We train a model f on this data.

$$min_f \; Error[(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)]$$

The generalization error of f is

$$E_{(\mathbf{x}^*, y^*) \sim p}[Error(y^*, f(\mathbf{x}^*))]$$

A measure of how good f is for p

Aside: Generalization Error

How do we evaluate generalization error?

Aside: Generalization Error

How do we evaluate generalization error?

One option: Hold out some data evaluate on that!

Aside: Generalization Error

How do we evaluate generalization error?

Use the training data?

Aside: Generalization Error

How do we evaluate generalization error?

Use the training data? The generalization error of f is

$$E_{(\mathbf{x}^*, y^*) \sim p}[Error(y^*, f(\mathbf{x}^*))]$$

Aside: Generalization Error

How do we evaluate generalization error?

Use the training data? The generalization error of f is

$$\mathbb{E}_{\mathcal{D}=(\mathbf{x}_1,y_1),\dots,(\mathbf{x}_n,y_n)\sim p} E_{(\mathbf{x}^*,y^*)\sim p}[Error(y^*, f(\mathbf{x}^*;\mathcal{D}))]$$

- Double use of data is not an unbiased expectation
- Generalization bounds correct for this
- Overfitting is when train error is much lower than test

Random Forests

Suppose the random forest has K trees. Define

$$\text{margin}(\mathbf{x}, y) = \frac{1}{K}\sum_{k=1}^{K} \mathbb{1}[h(\mathbf{x}; \boldsymbol{\theta}_k) = y] - \frac{1}{K}\sum_{k=1}^{K} \mathbb{1}[h(\mathbf{x}; \boldsymbol{\theta}_k) = 1 - y]$$

The fraction of trees correct minus those that are incorrect

Random Forests

Suppose the random forest has K trees. Define

$$\text{margin}(\mathbf{x}, y) = \frac{1}{K}\sum_{k=1}^{K} \mathbb{1}[h(\mathbf{x}; \boldsymbol{\theta}_k) = y] - \frac{1}{K}\sum_{k=1}^{K} \mathbb{1}[h(\mathbf{x}; \boldsymbol{\theta}_k) = 1 - y]$$

The fraction of trees correct minus those that are incorrect

The generalization error is

$$ge = E_{(\mathbf{x}^*, y^*)\sim p}[\text{margin}(\mathbf{x}^*, y^*) < 0]$$

Random Forests

$$ge = E_{(\mathbf{x},y)\sim p}\left[\frac{1}{K}\sum_{k=1}^{K}\mathbb{1}[h(\mathbf{x};\boldsymbol{\theta}_k)=y]-\frac{1}{K}\sum_{k=1}^{K}\mathbb{1}[h(\mathbf{x};\boldsymbol{\theta}_k)=1-y]<0\right]$$

What happens when more trees are added?

$$ge = E_{(\mathbf{x},y)\sim p}\left[\mathbb{E}_{\boldsymbol{\theta}\sim q}[\mathbb{1}[h(\mathbf{x};\boldsymbol{\theta})=y]-\mathbb{1}[h(\mathbf{x};\boldsymbol{\theta})=1-y]]<0\right]$$

Random Forests

$$ge = E_{(\mathbf{x},y)\sim p}\left[\frac{1}{K}\sum_{k=1}^{K}\mathbb{1}[h(\mathbf{x};\boldsymbol{\theta}_k)=y]-\frac{1}{K}\sum_{k=1}^{K}\mathbb{1}[h(\mathbf{x};\boldsymbol{\theta}_k)=1-y]<0\right]$$

What happens when more trees are added?

$$ge = E_{(\mathbf{x},y)\sim p}\left[\mathbb{E}_{\boldsymbol{\theta}\sim q}[\mathbb{1}[h(\mathbf{x};\boldsymbol{\theta})=y]-\mathbb{1}[h(\mathbf{x};\boldsymbol{\theta})=1-y]]<0\right]$$

Adding more trees doesn't harm generalization

Random Forests

Adding more trees doesn't harm generalization

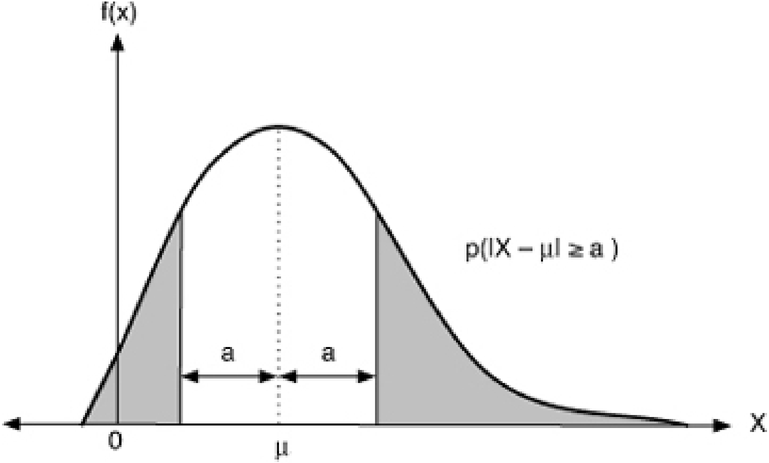
$$ge = E_{(\mathbf{x},y)\sim p}\left[\mathbb{E}_{\boldsymbol{\theta}\sim q}[\mathbb{1}[h(\mathbf{x};\boldsymbol{\theta}) = y] - \mathbb{1}[h(\mathbf{x};\boldsymbol{\theta}) = 1 - y]] < 0\right]$$

- But that's only true for a fixed choice of q
- If q learns a single tree using CART, it's the same as before?
- How do we choose random tree construction q?

Understanding generalization error gives rules of thumb

Chebyshev's Inequality

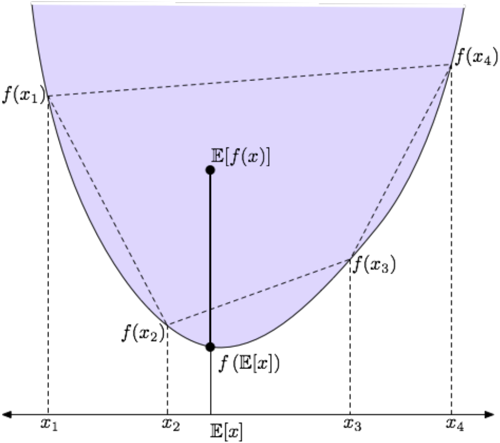
$$P(|x-\mu| \geq k\sigma) \leq \frac{1}{k^2}$$



[informit.com]

Jensen’s Inequality

$$\mathbb{E}[f(\mathbf{x})] \geq f(\mathbb{E}[\mathbf{x}])$$



[http://mark.reid.name]

Margin of a random forest assuming number of trees is large

$$\begin{aligned} \text{margin}(\mathbf{x}, y) &= \mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\mathbb{1}[h(\mathbf{x};\boldsymbol{\theta}) = y] - \mathbb{1}[h(\mathbf{x};\boldsymbol{\theta}) = 1 - y]\right] \\ &= P_{\boldsymbol{\theta}\sim q}[h(\mathbf{x};\boldsymbol{\theta}) = y] - P_{\boldsymbol{\theta}\sim q}[h(\mathbf{x};\boldsymbol{\theta}) = 1 - y] \end{aligned}$$

By substitution, for a large number of trees we have

$$\begin{aligned} ge &= E_{(\mathbf{x},y)\sim p}\Big[\mathbb{E}_{\boldsymbol{\theta}\sim q}[\mathbb{1}[h(\mathbf{x};\boldsymbol{\theta}) = y] - \mathbb{1}[h(\mathbf{x};\boldsymbol{\theta}) = 1 - y] < 0]\Big] \\ &= P_{(\mathbf{x},y)}[\text{margin}(\mathbf{x}, y) < 0] \end{aligned}$$

Define the strength of a random forest

$$s = \mathbb{E}_{\mathbf{x},y}[\text{margin}(\mathbf{x},y)]$$

Assume $s > 0$, by Chebyshev,

$$p(|x-\mu| \geq k\sigma) \leq \frac{1}{k^2}$$

we have

$$\begin{aligned}
&ge = P_{(\mathbf{x},y)}[\text{margin}(x,y) < 0] \\
=&P_{(\mathbf{x},y)}[\text{margin}(x,y) - \mathbb{E}_{\mathbf{x},y}[\text{margin}(\mathbf{x},y)] < -\mathbb{E}_{\mathbf{x},y}[\text{margin}(\mathbf{x},y)]] \\
\leq&P_{(\mathbf{x},y)}[\text{margin}(x,y) - \mathbb{E}_{\mathbf{x},y}[\text{margin}(\mathbf{x},y)] \leq -\mathbb{E}_{\mathbf{x},y}[\text{margin}(\mathbf{x},y)]] \\
&\quad + P_{(\mathbf{x},y)}[\text{margin}(x,y) - \mathbb{E}_{\mathbf{x},y}[\text{margin}(\mathbf{x},y)] \geq \mathbb{E}_{\mathbf{x},y}[\text{margin}(\mathbf{x},y)]] \\
=&P_{(\mathbf{x},y)}[|\text{margin}(x,y) - \mathbb{E}_{\mathbf{x},y}[\text{margin}(\mathbf{x},y)]| \geq \mathbb{E}_{\mathbf{x},y}[\text{margin}(\mathbf{x},y)]] \\
\leq&\frac{\mathbb{V}\text{ar}[\text{margin}]}{\mathbb{E}[\text{margin}]^2} = \frac{\mathbb{V}\text{ar}[\text{margin}]}{s^2}
\end{aligned}$$

- Uses
  - □ $P[A \leq -b] + P[A \geq b] = P[|A| \geq b]$, for $b > 0$
  - □ $k\sigma = \mathbb{E}[\text{margin}] \implies k = \frac{\mathbb{E}[\text{margin}]}{\sigma}$

Define the strength of a random forest

$$s = \mathbb{E}_{\mathbf{x},y}[\text{margin}(\mathbf{x},y)]$$

Assume $s > 0$, by Chebyshev,

$$p(|x-\mu| \geq k\sigma) \leq \frac{1}{k^2}$$

we have

$$\begin{aligned}
ge = {} & P_{(\mathbf{x},y)}[\text{margin}(x,y) < 0] \\
= & P_{(\mathbf{x},y)}[\text{margin}(x,y) - \mathbb{E}_{\mathbf{x},y}[\text{margin}(\mathbf{x},y)] < -\mathbb{E}_{\mathbf{x},y}[\text{margin}(\mathbf{x},y)]] \\
\leq & P_{(\mathbf{x},y)}[\text{margin}(x,y) - \mathbb{E}_{\mathbf{x},y}[\text{margin}(\mathbf{x},y)] \leq -\mathbb{E}_{\mathbf{x},y}[\text{margin}(\mathbf{x},y)]] \\
& + P_{(\mathbf{x},y)}[\text{margin}(x,y) - \mathbb{E}_{\mathbf{x},y}[\text{margin}(\mathbf{x},y)] \geq \mathbb{E}_{\mathbf{x},y}[\text{margin}(\mathbf{x},y)]] \\
= & P_{(\mathbf{x},y)}[|\text{margin}(x,y) - \mathbb{E}_{\mathbf{x},y}[\text{margin}(\mathbf{x},y)]| \geq \mathbb{E}_{\mathbf{x},y}[\text{margin}(\mathbf{x},y)]] \\
\leq & \frac{\mathbb{V}\text{ar}[\text{margin}]}{\mathbb{E}[\text{margin}]^2} = \frac{\mathbb{V}\text{ar}[\text{margin}]}{s^2}
\end{aligned}$$

Two sources of randomness. Trees and test data point.

Define the strength of a random forest

$$s = \mathbb{E}_{\mathbf{x},y}[\text{margin}(\mathbf{x},y)]$$

Assume $s > 0$, by Chebyshev,

$$p(|x-\mu| \geq k\sigma) \leq \frac{1}{k^2}$$

we have

$$\begin{aligned}
ge = {} & P_{(\mathbf{x},y)}[\text{margin}(x,y) < 0] \\
= & P_{(\mathbf{x},y)}[\text{margin}(x,y) - \mathbb{E}_{\mathbf{x},y}[\text{margin}(\mathbf{x},y)] < -\mathbb{E}_{\mathbf{x},y}[\text{margin}(\mathbf{x},y)]] \\
\leq & P_{(\mathbf{x},y)}[\text{margin}(x,y) - \mathbb{E}_{\mathbf{x},y}[\text{margin}(\mathbf{x},y)] \leq -\mathbb{E}_{\mathbf{x},y}[\text{margin}(\mathbf{x},y)]] \\
& + P_{(\mathbf{x},y)}[\text{margin}(x,y) - \mathbb{E}_{\mathbf{x},y}[\text{margin}(\mathbf{x},y)] \geq \mathbb{E}_{\mathbf{x},y}[\text{margin}(\mathbf{x},y)]] \\
= & P_{(\mathbf{x},y)}[|\text{margin}(x,y) - \mathbb{E}_{\mathbf{x},y}[\text{margin}(\mathbf{x},y)]| \geq \mathbb{E}_{\mathbf{x},y}[\text{margin}(\mathbf{x},y)]] \\
\leq & \frac{\mathbb{V}\text{ar}[\text{margin}]}{\mathbb{E}[\text{margin}]^2} = \frac{\mathbb{V}\text{ar}[\text{margin}]}{s^2}
\end{aligned}$$

- Two inequalities
- But big-margin trees reduce generalization error; Variance not intuitive
- A little sloppy with $\geq$ and $>$

Breaking the variance down

An identity

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\theta}\sim q}[h(\boldsymbol{\theta})]^2 &= \int h(\boldsymbol{\theta})q(\boldsymbol{\theta})d\boldsymbol{\theta} \int h(\boldsymbol{\theta}')q(\boldsymbol{\theta}')d\boldsymbol{\theta}' \\
&= \int \int h(\boldsymbol{\theta})q(\boldsymbol{\theta})h(\boldsymbol{\theta}')q(\boldsymbol{\theta}')d\boldsymbol{\theta}'d\boldsymbol{\theta} \\
&= \mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\theta}'\sim q}[h(\boldsymbol{\theta})h(\boldsymbol{\theta}')]
\end{aligned}$$

Breaking the variance down

$$\begin{aligned}\text{margin}(\mathbf{x},y) &= \mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\mathbb{1}[h(\mathbf{x};\boldsymbol{\theta})=y]-\mathbb{1}[h(\mathbf{x};\boldsymbol{\theta})=1-y]\right] \\ &:= \mathbb{E}_{\boldsymbol{\theta}\sim q}[\text{raw-margin}(\mathbf{x},y,\boldsymbol{\theta})]\end{aligned}$$

With the previous gives

$$\text{margin}(\mathbf{x},y)^2 = E_{\boldsymbol{\theta},\boldsymbol{\theta}'\sim q}[\text{raw-margin}(\mathbf{x},y,\boldsymbol{\theta})\text{raw-margin}(\mathbf{x},y,\boldsymbol{\theta}')]$$

Breaking the variance down

$$\begin{aligned}&\mathbb{V}\mathrm{ar}[\mathrm{margin}]\\&= \mathbb{E}_{\mathbf{x},y\sim p}[(\mathrm{margin}(\mathbf{x},y) - \mathbb{E}_{\mathbf{x}^*,y^*\sim p}[\mathrm{margin}(\mathbf{x}^*,y^*)])^2]\end{aligned}$$

Breaking the variance down

$$
\begin{aligned}
&\mathbb{V}\mathrm{ar}[\mathrm{margin}] \\
&= \mathbb{E}_{\mathbf{x},y\sim p}[(\mathrm{margin}(\mathbf{x},y) - \mathbb{E}_{\mathbf{x}^*,y^*\sim p}[\mathrm{margin}(\mathbf{x}^*,y^*)])^2] \\
&= \mathbb{E}_{\mathbf{x},y\sim p}[(\mathbb{E}_{\boldsymbol{\theta}\sim q}[rm(\mathbf{x},y,\boldsymbol{\theta})] - \mathbb{E}_{\mathbf{x}^*,y^*\sim p}[\mathbb{E}_{\boldsymbol{\theta}\sim q}[rm(\mathbf{x}^*,y^*,\boldsymbol{\theta})]])^2]
\end{aligned}
$$

Breaking the variance down

$$
\begin{aligned}
&\mathbb{V}\mathrm{ar}[\mathrm{margin}] \\
&= \mathbb{E}_{\mathbf{x},y\sim p}[(\mathrm{margin}(\mathbf{x},y) - \mathbb{E}_{\mathbf{x}^*,y^*\sim p}[\mathrm{margin}(\mathbf{x}^*,y^*)])^2] \\
&= \mathbb{E}_{\mathbf{x},y\sim p}[(\mathbb{E}_{\boldsymbol{\theta}\sim q}[rm(\mathbf{x},y,\boldsymbol{\theta})] - \mathbb{E}_{\mathbf{x}^*,y^*\sim p}[\mathbb{E}_{\boldsymbol{\theta}\sim q}[rm(\mathbf{x}^*,y^*,\boldsymbol{\theta})]])^2] \\
&= \mathbb{E}_{\mathbf{x},y\sim p}[(\mathbb{E}_{\boldsymbol{\theta}\sim q}[rm(\mathbf{x},y,\boldsymbol{\theta})] - \mathbb{E}_{\boldsymbol{\theta}\sim q}[\mathbb{E}_{\mathbf{x}^*,y^*\sim p}[rm(\mathbf{x}^*,y^*,\boldsymbol{\theta})]])^2]
\end{aligned}
$$

Breaking the variance down

$$
\begin{aligned}
&\mathbb{V}\mathrm{ar}[\mathrm{margin}] \\
&= \mathbb{E}_{\mathbf{x},y\sim p}[(\mathrm{margin}(\mathbf{x},y) - \mathbb{E}_{\mathbf{x}^*,y^*\sim p}[\mathrm{margin}(\mathbf{x}^*,y^*)])^2] \\
&= \mathbb{E}_{\mathbf{x},y\sim p}[(\mathbb{E}_{\boldsymbol{\theta}\sim q}[rm(\mathbf{x},y,\boldsymbol{\theta})] - \mathbb{E}_{\mathbf{x}^*,y^*\sim p}[\mathbb{E}_{\boldsymbol{\theta}\sim q}[rm(\mathbf{x}^*,y^*,\boldsymbol{\theta})]])^2] \\
&= \mathbb{E}_{\mathbf{x},y\sim p}[(\mathbb{E}_{\boldsymbol{\theta}\sim q}[rm(\mathbf{x},y,\boldsymbol{\theta})] - \mathbb{E}_{\boldsymbol{\theta}\sim q}[\mathbb{E}_{\mathbf{x}^*,y^*\sim p}[rm(\mathbf{x}^*,y^*,\boldsymbol{\theta})]])^2] \\
&= \mathbb{E}_{\mathbf{x},y\sim p}[(\mathbb{E}_{\boldsymbol{\theta}\sim q}[rm(\mathbf{x},y,\boldsymbol{\theta}) - \mathbb{E}_{\mathbf{x}^*,y^*\sim p}[rm(\mathbf{x}^*,y^*,\boldsymbol{\theta})]])^2]
\end{aligned}
$$

Breaking the variance down

$$
\begin{aligned}
&\mathbb{V}\mathrm{ar}[\mathrm{margin}]\\
&=\mathbb{E}_{\mathbf{x},y\sim p}[(\mathrm{margin}(\mathbf{x},y)-\mathbb{E}_{\mathbf{x}^*,y^*\sim p}[\mathrm{margin}(\mathbf{x}^*,y^*)])^2]\\
&=\mathbb{E}_{\mathbf{x},y\sim p}[(\mathbb{E}_{\boldsymbol{\theta}\sim q}[rm(\mathbf{x},y,\boldsymbol{\theta})]-\mathbb{E}_{\mathbf{x}^*,y^*\sim p}[\mathbb{E}_{\boldsymbol{\theta}\sim q}[rm(\mathbf{x}^*,y^*,\boldsymbol{\theta})]])^2]\\
&=\mathbb{E}_{\mathbf{x},y\sim p}[(\mathbb{E}_{\boldsymbol{\theta}\sim q}[rm(\mathbf{x},y,\boldsymbol{\theta})]-\mathbb{E}_{\boldsymbol{\theta}\sim q}[\mathbb{E}_{\mathbf{x}^*,y^*\sim p}[rm(\mathbf{x}^*,y^*,\boldsymbol{\theta})]])^2]\\
&=\mathbb{E}_{\mathbf{x},y\sim p}[(\mathbb{E}_{\boldsymbol{\theta}\sim q}[rm(\mathbf{x},y,\boldsymbol{\theta})-\mathbb{E}_{\mathbf{x}^*,y^*\sim p}[rm(\mathbf{x}^*,y^*,\boldsymbol{\theta})]])^2]\\
&=\mathbb{E}_{\mathbf{x},y\sim p}[\mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\theta}'\sim q}[(rm(\mathbf{x},y,\boldsymbol{\theta})-\mathbb{E}_{\mathbf{x}^*,y^*\sim p}[rm(\mathbf{x}^*,y^*,\boldsymbol{\theta})])\\
&\quad(rm(\mathbf{x},y,\boldsymbol{\theta}')-\mathbb{E}_{\mathbf{x}^*,y^*\sim p}[rm(\mathbf{x},y,\boldsymbol{\theta}')])]]
\end{aligned}
$$

Breaking the variance down

$$
\begin{aligned}
&\mathbb{V}\mathrm{ar}[\mathrm{margin}] \\
&= \mathbb{E}_{\mathbf{x},y\sim p}[(\mathrm{margin}(\mathbf{x},y) - \mathbb{E}_{\mathbf{x}^*,y^*\sim p}[\mathrm{margin}(\mathbf{x}^*,y^*)])^2] \\
&= \mathbb{E}_{\mathbf{x},y\sim p}[(\mathbb{E}_{\boldsymbol{\theta}\sim q}[rm(\mathbf{x},y,\boldsymbol{\theta})] - \mathbb{E}_{\mathbf{x}^*,y^*\sim p}[\mathbb{E}_{\boldsymbol{\theta}\sim q}[rm(\mathbf{x}^*,y^*,\boldsymbol{\theta})]]])^2] \\
&= \mathbb{E}_{\mathbf{x},y\sim p}[(\mathbb{E}_{\boldsymbol{\theta}\sim q}[rm(\mathbf{x},y,\boldsymbol{\theta})] - \mathbb{E}_{\boldsymbol{\theta}\sim q}[\mathbb{E}_{\mathbf{x}^*,y^*\sim p}[rm(\mathbf{x}^*,y^*,\boldsymbol{\theta})]]])^2] \\
&= \mathbb{E}_{\mathbf{x},y\sim p}[(\mathbb{E}_{\boldsymbol{\theta}\sim q}[rm(\mathbf{x},y,\boldsymbol{\theta}) - \mathbb{E}_{\mathbf{x}^*,y^*\sim p}[rm(\mathbf{x}^*,y^*,\boldsymbol{\theta})]]])^2] \\
&= \mathbb{E}_{\mathbf{x},y\sim p}[\mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\theta}'\sim q}[(rm(\mathbf{x},y,\boldsymbol{\theta}) - \mathbb{E}_{\mathbf{x}^*,y^*\sim p}[rm(\mathbf{x}^*,y^*,\boldsymbol{\theta})]) \\
&\quad (rm(\mathbf{x},y,\boldsymbol{\theta}') - \mathbb{E}_{\mathbf{x}^*,y^*\sim p}[rm(\mathbf{x},y,\boldsymbol{\theta}')])]] \\
&= \mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\theta}'\sim q}\mathbb{E}_{\mathbf{x},y\sim p}[(rm(\mathbf{x},y,\boldsymbol{\theta}) - \mathbb{E}_{\mathbf{x}^*,y^*\sim p}[rm(\mathbf{x}^*,y^*,\boldsymbol{\theta})]) \\
&\quad (rm(\mathbf{x},y,\boldsymbol{\theta}') - \mathbb{E}_{\mathbf{x}^*,y^*\sim p}[rm(\mathbf{x}^*,y^*,\boldsymbol{\theta}')])]]
\end{aligned}
$$

Breaking the variance down

$$
\begin{aligned}
&\mathbb{V}\mathrm{ar}[\mathrm{margin}]\\
&= \mathbb{E}_{\mathbf{x},y\sim p}[(\mathrm{margin}(\mathbf{x},y) - \mathbb{E}_{\mathbf{x}^*,y^*\sim p}[\mathrm{margin}(\mathbf{x}^*,y^*)])^2]\\
&= \mathbb{E}_{\mathbf{x},y\sim p}[(\mathbb{E}_{\boldsymbol{\theta}\sim q}[rm(\mathbf{x},y,\boldsymbol{\theta})] - \mathbb{E}_{\mathbf{x}^*,y^*\sim p}[\mathbb{E}_{\boldsymbol{\theta}\sim q}[rm(\mathbf{x}^*,y^*,\boldsymbol{\theta})]])^2]\\
&= \mathbb{E}_{\mathbf{x},y\sim p}[(\mathbb{E}_{\boldsymbol{\theta}\sim q}[rm(\mathbf{x},y,\boldsymbol{\theta})] - \mathbb{E}_{\boldsymbol{\theta}\sim q}[\mathbb{E}_{\mathbf{x}^*,y^*\sim p}[rm(\mathbf{x}^*,y^*,\boldsymbol{\theta})]])^2]\\
&= \mathbb{E}_{\mathbf{x},y\sim p}[(\mathbb{E}_{\boldsymbol{\theta}\sim q}[rm(\mathbf{x},y,\boldsymbol{\theta}) - \mathbb{E}_{\mathbf{x}^*,y^*\sim p}[rm(\mathbf{x}^*,y^*,\boldsymbol{\theta})]])^2]\\
&= \mathbb{E}_{\mathbf{x},y\sim p}[\mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\theta}'\sim q}[(rm(\mathbf{x},y,\boldsymbol{\theta}) - \mathbb{E}_{\mathbf{x}^*,y^*\sim p}[rm(\mathbf{x}^*,y^*,\boldsymbol{\theta})])\\
&\quad (rm(\mathbf{x},y,\boldsymbol{\theta}') - \mathbb{E}_{\mathbf{x}^*,y^*\sim p}[rm(\mathbf{x},y,\boldsymbol{\theta}')])]]\\
&= \mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\theta}'\sim q}\mathbb{E}_{\mathbf{x},y\sim p}[(rm(\mathbf{x},y,\boldsymbol{\theta}) - \mathbb{E}_{\mathbf{x}^*,y^*\sim p}[rm(\mathbf{x}^*,y^*,\boldsymbol{\theta})])\\
&\quad (rm(\mathbf{x},y,\boldsymbol{\theta}') - \mathbb{E}_{\mathbf{x}^*,y^*\sim p}[rm(\mathbf{x}^*,y^*,\boldsymbol{\theta}')])]]\\
&= \mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\theta}'\sim q}\mathbb{C}\mathrm{ov}_{\mathbf{x},y\sim p}[rm(\mathbf{x},y,\boldsymbol{\theta}), rm(\mathbf{x},y,\boldsymbol{\theta}')]
\end{aligned}
$$

Breaking the variance down

$$\begin{aligned}\mathbb{V}\mathrm{ar}[\mathrm{margin}] &= \mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\theta}'\sim q}\mathbb{C}\mathrm{ov}_{\mathbf{x},y\sim p}[rm(\mathbf{x},y,\boldsymbol{\theta}),rm(\mathbf{x},y,\boldsymbol{\theta}')] \\ &= \mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\theta}'\sim q}\Big[\rho_{\mathbf{x},y\sim p}(rm(\mathbf{x},y,\boldsymbol{\theta}),rm(\mathbf{x},y,\boldsymbol{\theta}')) \\ &\qquad \sqrt{\mathbb{V}\mathrm{ar}_{\mathbf{x},y\sim p}[rm(\mathbf{x},y,\boldsymbol{\theta})])}\sqrt{\mathbb{V}\mathrm{ar}_{\mathbf{x},y\sim p}[rm(\mathbf{x},y,\boldsymbol{\theta}')])}\Big]\end{aligned}$$

Breaking the variance down

$$\begin{aligned}\mathbb{V}\mathrm{ar}[\mathrm{margin}] &= \mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\theta}'\sim q}\mathbb{C}\mathrm{ov}_{\mathbf{x},y\sim p}[rm(\mathbf{x},y,\boldsymbol{\theta}),rm(\mathbf{x},y,\boldsymbol{\theta}')] \\ &= \mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\theta}'\sim q}\Big[\rho_{\mathbf{x},y\sim p}(rm(\mathbf{x},y,\boldsymbol{\theta}),rm(\mathbf{x},y,\boldsymbol{\theta}')) \\ &\qquad \sqrt{\mathbb{V}\mathrm{ar}_{\mathbf{x},y\sim p}[rm(\mathbf{x},y,\boldsymbol{\theta})])}\sqrt{\mathbb{V}\mathrm{ar}_{\mathbf{x},y\sim p}[rm(\mathbf{x},y,\boldsymbol{\theta}')])}\Big]\end{aligned}$$

Define average correlation $\bar{\rho}$

$$\bar{\rho} = \frac{\mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\theta}'\sim q}\Big[\rho(\boldsymbol{\theta},\boldsymbol{\theta}')\sqrt{\mathbb{V}\mathrm{ar}[\boldsymbol{\theta}]}\sqrt{\mathbb{V}\mathrm{ar}[\boldsymbol{\theta}']]}\Big]}{\mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\theta}'\sim q}\Big[\sqrt{\mathbb{V}\mathrm{ar}[\boldsymbol{\theta}]}\sqrt{\mathbb{V}\mathrm{ar}[\boldsymbol{\theta}']}\Big]}$$

Breaking the variance down

$$\begin{aligned}\mathbb{V}\mathrm{ar}[\mathrm{margin}] &= \mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\theta}'\sim q}\mathbb{C}\mathrm{ov}_{\mathbf{x},y\sim p}[rm(\mathbf{x},y,\boldsymbol{\theta}),rm(\mathbf{x},y,\boldsymbol{\theta}')] \\ &= \mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\theta}'\sim q}\Big[\rho_{\mathbf{x},y\sim p}(rm(\mathbf{x},y,\boldsymbol{\theta}),rm(\mathbf{x},y,\boldsymbol{\theta}')) \\ &\qquad \sqrt{\mathbb{V}\mathrm{ar}_{\mathbf{x},y\sim p}[rm(\mathbf{x},y,\boldsymbol{\theta})])}\sqrt{\mathbb{V}\mathrm{ar}_{\mathbf{x},y\sim p}[rm(\mathbf{x},y,\boldsymbol{\theta}')])}\Big]\end{aligned}$$

Define average correlation $\bar{\rho}$

$$\bar{\rho} = \frac{\mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\theta}'\sim q}\Big[\rho(\boldsymbol{\theta},\boldsymbol{\theta}')\sqrt{\mathbb{V}\mathrm{ar}[\boldsymbol{\theta}]}\sqrt{\mathbb{V}\mathrm{ar}[\boldsymbol{\theta}']]}\Big]}{\mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\theta}'\sim q}\Big[\sqrt{\mathbb{V}\mathrm{ar}[\boldsymbol{\theta}]}\sqrt{\mathbb{V}\mathrm{ar}[\boldsymbol{\theta}']}\Big]}$$

We get

$$\mathbb{V}\mathrm{ar}[\mathrm{margin}] = \bar{\rho}\,\mathbb{E}_{\boldsymbol{\theta}\sim q}[\sqrt{\mathbb{V}\mathrm{ar}[\boldsymbol{\theta}]}]^2 \leq \bar{\rho}\,\mathbb{E}_{\boldsymbol{\theta}\sim q}[\mathbb{V}\mathrm{ar}[\boldsymbol{\theta}]]$$

Breaking the variance down

Simplifying the variance

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\theta}\sim q}[\mathbb{V}\mathrm{ar}[\boldsymbol{\theta}]] &= \mathbb{E}_{\boldsymbol{\theta}\sim q}[\mathbb{E}[rm_{\mathbf{x},y\sim p}(\mathbf{x},y,\boldsymbol{\theta})]^2] - \mathbb{E}_{\boldsymbol{\theta}\sim q}[\mathbb{E}_{\mathbf{x},y\sim p}[rm(\mathbf{x},y,\boldsymbol{\theta})]]^2 \\ &= \mathbb{E}_{\boldsymbol{\theta}\sim q}[\mathbb{E}[rm_{\mathbf{x},y\sim p}(\mathbf{x},y,\boldsymbol{\theta})]^2] - s^2 \\ &\leq 1 - s^2\end{aligned}$$

Implies that

$$\mathbb{V}\mathrm{ar}[\mathrm{margin}] \leq \bar{\rho}(1-s^2)$$

Putting it all together

What did we just do?

Putting it all together

What did we just do?

We rewrote the variance of the margin to get an interpretable generalization error

A Generalization Error Bound For Random Forests

$$
\begin{aligned}
ge &\leq \frac{\mathbb{V}\mathrm{ar}[\mathrm{margin}]}{\mathbb{E}[\mathrm{margin}]^2} = \frac{\mathbb{V}\mathrm{ar}[\mathrm{margin}]}{s^2} \\
&\leq \frac{\bar{\rho}(1-s^2)}{s^2}
\end{aligned}
$$

Good forests have

- Low correlation between trees (ρ)
- High strength of trees (s)

Algorithms to Grow Forests

Want to design a tree sampling algorithm q where

- Two samples have low correlations

- Average margin is good

Algorithms to Grow Forests

Want to design a tree sampling algorithm q where

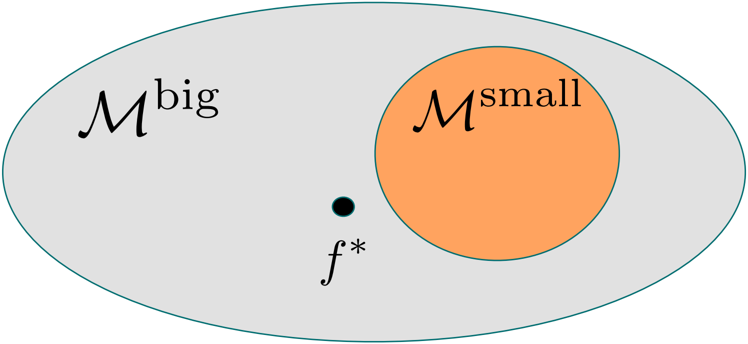
- Two samples have low correlations
- Average margin is good

Both things are in tension with each other

- Maximum margin is a “single” tree
- Good average margin means trees close to maximum margin
- Many trees close to maximum margin tree → correlated trees

A way to grow forests

1. Sample m data points from the training set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$
2. Grow a decision tree
3. Each split can only use a random subset of features


$\mathcal{M}^{\text{big}}$
$\mathcal{M}^{\text{small}}$
f^*

Random Forests: Beyond Classification

- Similar analysis holds for multiple classes

- Mean squared error based analysis for regression

Prediction with Random Forests

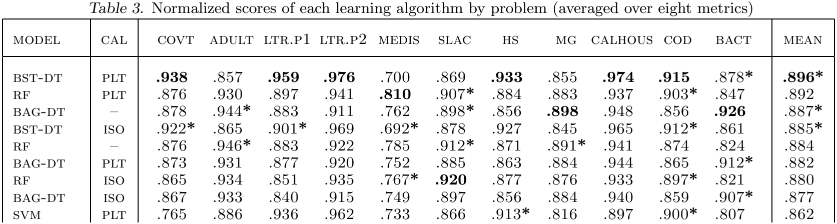


Table 3. Normalized scores of each learning algorithm by problem (averaged over eight metrics)

MODEL	CAL	COVT	ADULT	LTR.P1	LTR.P2	MEDIS	SLAC	HS	MG	CALHOUS	COD	BACT	MEAN
BST-DT	PLT	**.938**	.857	**.959**	**.976**	.700	.869	**.933**	.855	**.974**	**.915**	.878*	**.896***
RF	PLT	.876	.930	.897	.941	**.810**	.907*	.884	.883	.937	.903*	.847	.892
BAG-DT	–	.878	.944*	.883	.911	.762	.898*	.856	**.898**	.948	.856	**.926**	.887*
BST-DT	ISO	.922*	.865	.901*	.969	.692*	.878	.927	.845	.965	.912*	.861	.885*
RF	–	.876	.946*	.883	.922	.785	.912*	.871	.891*	.941	.874	.824	.884
BAG-DT	PLT	.873	.931	.877	.920	.752	.885	.863	.884	.944	.865	.912*	.882
RF	ISO	.865	.934	.851	.935	.767*	**.920**	.877	.876	.933	.897*	.821	.880
BAG-DT	ISO	.867	.933	.840	.915	.749	.897	.856	.884	.940	.859	.907*	.877
SVM	PLT	.765	.886	.936	.962	.733	.866	.913*	.816	.897	.900*	.807	.862

[Caruana; 2006]

Random Forests: Computation

Each tree in random forest requires

$$\boldsymbol{\theta}_k \sim q$$

Each sample can be drawn in parallel

Random forests *embarrassingly* parallel

Suppose

- n features
- Half the features are chosen for each split
- Combination of half the features determine class
- Very rough estimate for building correct tree is exponential in n

Suppose

- *n* features
- Half the features are chosen for each split
- Combination of half the features determine class
- Very rough estimate for building correct tree is exponential in *n*

Random forests rely on randomness to figure out what's important

Suppose

- n features
- Half the features are chosen for each split
- Combination of half the features determine class
- Very rough estimate for building correct tree is exponential in n

It’s this randomness that helps prevent overfitting with more compute

Suppose

- *n* features
- Half the features are chosen for each split
- Combination of half the features determine class
- Very rough estimate for building correct tree is exponential in *n*

It's this randomness that makes learning super fast

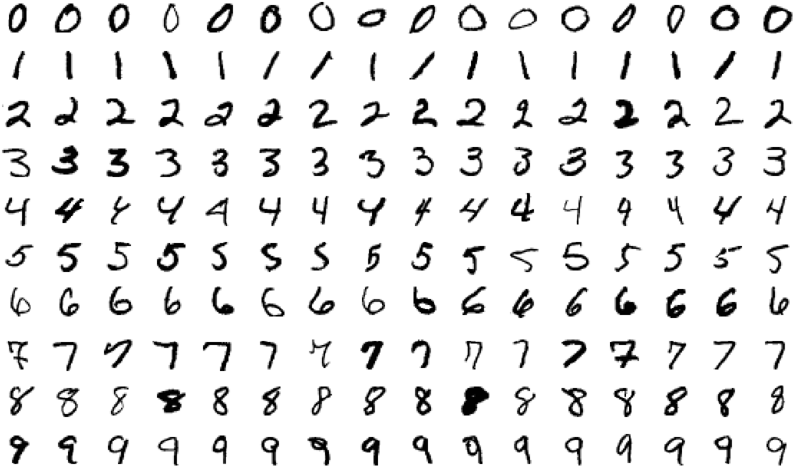
Suppose

- *n* features
- Half the features are chosen for each split
- Combination of half the features determine class
- Very rough estimate for building correct tree is exponential in *n*

It's this randomness that makes finding important corners hard

Is this rare?

Is this rare?



[Wikipedia, MNIST]

How do we move from random search?

Functional Gradients

- Gradients point in the direction of interest
- Gradients classically done for vectors
- Functions are infinite vectors

Can we do gradient optimization on functions?

Functional Gradients

$$\mathscr{L}(f) = \mathbb{E}_{y,\mathbf{x}} d(y, f(\mathbf{x})) = \mathbb{E}_{\mathbf{x}} \mathbb{E}_{y\,|\,\mathbf{x}}[d(y, f(\mathbf{x}))]$$

Suppose we have a current f^*

$$f_{m-1}(\mathbf{x}) = \sum_{i=1}^{m-1} \rho_i g_i(\mathbf{x})$$

Derivative

$$g_m(\mathbf{x}) = \left. \frac{\partial \mathbb{E}_y[d(y, f(\mathbf{x}))\,|\,\mathbf{x}]}{\partial f(\mathbf{x})} \right|_{f = f_{m-1}}$$

Interchanging differentiation and integration

$$g_m(\mathbf{x}) = \mathbb{E}_y \left. \left[\frac{\partial d(y, f(\mathbf{x}))}{\partial f(\mathbf{x})} \,|\, \mathbf{x} \right] \right|_{f = f_{m-1}}$$

Functional Gradients

Update with step size ρ_m

$$f_m(\mathbf{x}) = \sum_{i=1}^{m-1} \rho_i g_i(\mathbf{x}) - \rho_m g_m(\mathbf{x})$$

ρ_m can be set by line search to minimize loss

A quick example

$$\begin{aligned} g_m(\mathbf{x}) &= \mathbb{E}_y \left[\frac{\partial [\frac{1}{2}(y - f(\mathbf{x}))^2]}{\partial f(\mathbf{x})} \,|\, \mathbf{x} \right] \Bigg|_{f=f_{m-1}} \\ &= \mathbb{E}_y \left[f(\mathbf{x}) - y \,|\, \mathbf{x} \right] \Big|_{f=f_{m-1}} \\ &= \mathbb{E}_y \left[f_{m-1}(\mathbf{x}) - y \,|\, \mathbf{x} \right] \end{aligned}$$

Functional Gradients

Update with step size ρ_m

$$f_m(\mathbf{x}) = \sum_{i=1}^{m-1} \rho_i g_i(\mathbf{x}) - \rho_m g_m(\mathbf{x})$$

ρ_m can be set by line search to minimize loss

A quick example

$$\begin{aligned} g_m(\mathbf{x}) &= \mathbb{E}_y\left[\frac{\partial [\,\frac{1}{2}(y - f(\mathbf{x}))^2\,]}{\partial f(\mathbf{x})} \,|\, \mathbf{x} \right]\Bigg|_{f=f_{m-1}} \\ &= \mathbb{E}_y\left[f(\mathbf{x}) - y \,|\, \mathbf{x}\right]\Big|_{f=f_{m-1}} \\ &= \mathbb{E}_y\left[f_{m-1}(\mathbf{x}) - y \,|\, \mathbf{x}\right] \end{aligned}$$

Problems?

$$g_m(\mathbf{x}) = \mathbb{E}_y\left[f_{m-1}(\mathbf{x}) - y \,|\, \mathbf{x}\right]$$

Hard to compute given only finite data. Options?

- Optimize over a parametric family
- Greedy stagewise approach

Greedy Function Optimization

$$f_m(\mathbf{x}) = \sum_{i=1}^{m-1} \rho_i h_i(\mathbf{x}; a_i) + \rho_m h(\mathbf{x}; a_m)$$

Estimate h by two steps

1. Standard gradient at training points

$$-g_m(\mathbf{x}_i) = -\frac{\partial d(y_i, f(\mathbf{x}_i))}{\partial f(\mathbf{x}_i)}\bigg|_{f=f_{m-1}}$$

2. Project standard gradient to family h

$$\min_a \sum_{i=1}^{N} [-g_m(\mathbf{x}_i) - h(\mathbf{x}_i; a_m)]^2$$

Greedy Function Optimization

Update f by adding $\rho_m h(\mathbf{x}; a_m)$

$$\rho_m = \text{argmin}_{\rho} \sum_{i=1}^{n} d(y_i, f_{m-1}(\mathbf{x}_i) + \rho h(\mathbf{x}_i; a_m))$$

- A kind of projected functional gradient descent
- Step size set by line search

Example: Gradient Boosted Regression Trees

Prediction error: Mean squared error

$$d(y_i, f(\mathbf{x}_i)) = \frac{1}{2}(y_i - f(\mathbf{x}_i))^2$$

Current predictor

$$f_{m-1}(\mathbf{x}) = \sum_{i=1}^{m-1} \rho_i h_i(\mathbf{x}; a_i)$$

Compute derivative at f_{m-1}

$$-\frac{\partial d(y_i, f(\mathbf{x}_i))}{\partial f(\mathbf{x}_i)}\bigg|_{f=f_{m-1}} = (y_i - f(\mathbf{x}_i))$$

Need to project derivatives

How do we project derivatives with trees?

How do we project derivatives with trees?

$$\hat{y}_i = y_i - f(\mathbf{x}_i)$$

Fit a decision tree on $(\hat{y}_i, \mathbf{x}_i)$ pairs.

Generalizes residuals as local quantities for more losses

- Absolute deviation
- Robust losses
- Classification

What happens if you train forever?

What happens if you train forever?

- Overfits!
- Need to stop the number of iterations by checking a validation set
- Though it's possible to increase "margin" which can aid generalization

Random forests

- Don’t really overfit with more computational steps
- Embarrassingly parallel training
- Need to randomly find the right input region

Gradient Boosted Trees

- Can overfit with more computational steps
- Sequential training
- Uses gradients to focus on right regions

Real problems need non-linearities

Trees are an interpretable way to learn non-linearities

Trees can be unstable prone to overfit

Random forests stabilize by averaging trees

Randomness can be ineffective at finding regions in high dimensions

Gradients point in direction of interest

Gradient boosting finds trees with function gradients