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 $x_{i,vitamin-d}x_{i,vitamin-k}$

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

- 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

\[ \text{pred}(x^*) = \sum_{i=1}^{N_{\text{leaves}}} \mathbb{1}[x^* \in \text{leaf}_i] \mathbb{E}[y \mid x \in \text{leaf}_i] \]

Where \( \text{leaf}_i \) 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

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 $x$ are continuous

For each feature $j$

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

   - Prediction
     
     $$\text{prediction}(A_j) = \text{majority}(x_j \in A_j)$$

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

2. Choose the feature $j^*$ with the lowest classification error and split the data at $c^*_j$
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:

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

Measure of unpredictability of \( y \)
Aside: Information Measures

- **Entropy:**

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

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

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

Compared 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

\[ \mathcal{H}(y \mid 1[x_{i,j} \leq c_j]) \]
Classification error is one type of error. Many others possible

- Conditional Entropy

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

- Mutual Information. Define:

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

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

\[ h(x, \theta_k), \text{ where } \theta_k \sim q \]

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 

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

from $p$.

We train a model $f$ on this data.

$$\min_f \text{Error}[(x_1, y_1), \ldots, (x_n, y_n)]$$

The generalization error of $f$ is 

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

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

\[
E_{(x^*, y^*) \sim p}[\text{Error}(y^*, f(x^*))]
\]
Aside: Generalization Error

How do we evaluate generalization error?

Use the training data? The generalization error of $f$ is

$$
\mathbb{E}_{D}=(x_1,y_1),..., (x_n,y_n) \sim p E(x^*,y^*) \sim p [Error(y^*, f(x^*; D))]
$$

- 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

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

The fraction of trees correct minus those that are incorrect
Random Forests

Suppose the random forest has $K$ trees. Define

$$\text{margin}(x, y) = \frac{1}{K} \sum_{k=1}^{K} 1[h(x; \theta_k) = y] - \frac{1}{K} \sum_{k=1}^{K} 1[h(x; \theta_k) = 1 - y]$$

The fraction of trees correct minus those that are incorrect

The generalization error is

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

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

What happens when more trees are added?

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

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

What happens when more trees are added?

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

Adding more trees doesn’t harm generalization
Random Forests

Adding more trees doesn’t harm generalization

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

- 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

\[ p(|x - \mu| \geq k\sigma) \leq \frac{1}{k^2} \]
Jensen's Inequality

\[ \mathbb{E}[f(x)] \geq f(\mathbb{E}[x]) \]
Margin of a random forest assuming number of trees is large

\[
\text{margin}(x, y) = \mathbb{E}_{\theta \sim q}\left[\mathbb{I}[h(x; \theta) = y] - \mathbb{I}[h(x; \theta) = 1 - y]\right]
\]

\[
= P_{\theta \sim q}[h(x; \theta) = y] - P_{\theta \sim q}[h(x; \theta) = 1 - y]
\]

By substitution, for a large number of trees we have

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

\[
= p_{(x, y)}[\text{margin}(x, y) < 0]
\]
Define the strength of a random forest

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

By Chebyshev

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

we have

\[ ge = p_{(x,y)}[\text{margin}(x,y) < 0] \]
\[ = p_{(x,y)}[\text{margin}(x,y) - \mathbb{E}_{x,y}[\text{margin}(x,y)] < -\mathbb{E}_{x,y}[\text{margin}(x,y)]] \]
\[ \leq p_{(x,y)}[|\text{margin}(x,y) - \mathbb{E}_{x,y}[\text{margin}(x,y)]| \geq | - \mathbb{E}_{x,y}[\text{margin}(x,y)]|] \]
\[ \leq \frac{\text{Var}[\text{margin}]}{\mathbb{E}[\text{margin}]^2} = \frac{\text{Var}[\text{margin}]}{s^2} \]
Define the strength of a random forest

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

By Chebyshev

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

we have

\[ ge = p_{(x,y)}[\text{margin}(x,y) < 0] \]
\[ = p_{(x,y)}[\text{margin}(x,y) - \mathbb{E}_{x,y}[\text{margin}(x,y)] < -\mathbb{E}_{x,y}[\text{margin}(x,y)]] \]
\[ \leq p_{(x,y)}[|\text{margin}(x,y) - \mathbb{E}_{x,y}[\text{margin}(x,y)]| \geq | - \mathbb{E}_{x,y}[\text{margin}(x,y)]|] \]
\[ \leq \frac{\text{Var}[\text{margin}]}{\mathbb{E}[\text{margin}]^2} = \frac{\text{Var}[\text{margin}]}{s^2} \]

*Two sources of randomness. Trees and test data point.*
Define the strength of a random forest

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

By Chebyshev

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

we have

\[ ge = p_{(x,y)}[\text{margin}(x,y) < 0] \]
\[ = p_{(x,y)}[\text{margin}(x,y) - \mathbb{E}_{x,y}[\text{margin}(x,y)] < -\mathbb{E}_{x,y}[\text{margin}(x,y)]] \]
\[ \leq p_{(x,y)}[|\text{margin}(x,y) - \mathbb{E}_{x,y}[\text{margin}(x,y)]| \geq | - \mathbb{E}_{x,y}[\text{margin}(x,y)]|] \]
\[ \leq \frac{\text{Var}[\text{margin}]}{\mathbb{E}[\text{margin}]^2} = \frac{\text{Var}[\text{margin}]}{s^2} \]

- Two inequalities
- But big-margin trees reduce generalization error; Variance not intuitive
- A little sloppy with \( \geq \) and \( > \)
Breaking the variance down

An identity

\[ \mathbb{E}_{\theta \sim q}[h(\theta)]^2 = \int h(\theta)q(\theta) d\theta \int h(\theta')q(\theta') d\theta' \]

\[ = \int \int h(\theta)q(\theta)h(\theta')q(\theta') d\theta' d\theta \]

\[ = \mathbb{E}_{\theta, \theta' \sim q}[h(\theta)h(\theta')] \]
Breaking the variance down

\[
\text{margin}(x, y) = \mathbb{E}_{\theta \sim q} \left[ \mathbb{I}[h(x; \theta) = y] - \mathbb{I}[h(x; \theta) = 1 - y] \right] \\
:= \mathbb{E}_{\theta \sim q} [\text{raw-margin}(x, y, \theta)]
\]

With the previous gives

\[
\text{margin}(x, y)^2 = E_{\theta, \theta' \sim q} [\text{raw-margin}(x, y, \theta)\text{raw-margin}(x, y, \theta')]
\]
Breaking the variance down

\[ \text{Var}[\text{margin}] = \mathbb{E}_{x,y \sim p}[(\text{margin}(x,y) - \mathbb{E}_{x^*,y^* \sim p}[\text{margin}(x^*,y^*)])^2] \]
Breaking the variance down

\[ \text{Var}[\text{margin}] \\
= \mathbb{E}_{x,y \sim p}[(\text{margin}(x,y) - \mathbb{E}_{x^*,y^* \sim p}[\text{margin}(x^*,y^*)])^2] \\
= \mathbb{E}_{x,y \sim p}[(\mathbb{E}_{\theta \sim q}[\text{rm}(x,y,\theta)] - \mathbb{E}_{x^*,y^* \sim p}[\mathbb{E}_{\theta \sim q}[\text{rm}(x^*,y^*,\theta)]]))^2] \]
Breaking the variance down

$$\text{Var}[\text{margin}]$$
$$= \mathbb{E}_{x,y \sim p}[ (\text{margin}(x,y) - \mathbb{E}_{x^*,y^* \sim p}[\text{margin}(x^*,y^*)] )^2 ]$$
$$= \mathbb{E}_{x,y \sim p}[ (\mathbb{E}_{\theta \sim q}[\text{rm}(x,y,\theta)] - \mathbb{E}_{x^*,y^* \sim p}[\mathbb{E}_{\theta \sim q}[\text{rm}(x^*,y^*,\theta)]] )^2 ]$$
$$= \mathbb{E}_{x,y \sim p}[ (\mathbb{E}_{\theta \sim q}[\text{rm}(x,y,\theta)] - \mathbb{E}_{\theta \sim q}[\mathbb{E}_{x^*,y^* \sim p}[\text{rm}(x^*,y^*,\theta)]] )^2 ]$$
Breaking the variance down

\[ \text{Var}[\text{margin}] = \mathbb{E}_{x,y \sim p}(\text{margin}(x, y) - \mathbb{E}_{x^*, y^* \sim p}[\text{margin}(x^*, y^*)])^2 \]

\[ = \mathbb{E}_{x,y \sim p}((\mathbb{E}_{\theta \sim q}[rm(x, y, \theta)]) - \mathbb{E}_{x^*, y^* \sim p}[\mathbb{E}_{\theta \sim q}[rm(x^*, y^*, \theta)]])^2 \]

\[ = \mathbb{E}_{x,y \sim p}((\mathbb{E}_{\theta \sim q}[rm(x, y, \theta)]) - \mathbb{E}_{\theta \sim q}[\mathbb{E}_{x^*, y^* \sim p}[rm(x^*, y^*, \theta)]]))^2 \]

\[ = \mathbb{E}_{x,y \sim p}((\mathbb{E}_{\theta \sim q}[rm(x, y, \theta)]) - \mathbb{E}_{x^*, y^* \sim p}[rm(x^*, y^*, \theta)])^2 \]
Breaking the variance down

\[ \text{Var}[\text{margin}] \]
\[ = \mathbb{E}_{x,y \sim p}[(\text{margin}(x, y) - \mathbb{E}_{x^*, y^* \sim p}[\text{margin}(x^*, y^*)])^2] \]
\[ = \mathbb{E}_{x,y \sim p}[(\mathbb{E}_{\theta \sim q}[\text{rm}(x, y, \theta)] - \mathbb{E}_{x^*, y^* \sim p}[\mathbb{E}_{\theta \sim q}[\text{rm}(x^*, y^*, \theta)]]]^2] \]
\[ = \mathbb{E}_{x,y \sim p}[(\mathbb{E}_{\theta \sim q}[\text{rm}(x, y, \theta)] - \mathbb{E}_{x^*, y^* \sim p}[\mathbb{E}_{\theta \sim q}[\text{rm}(x^*, y^*, \theta)]]]^2] \]
\[ = \mathbb{E}_{x,y \sim p}[(\mathbb{E}_{\theta \sim q}[\text{rm}(x, y, \theta)] - \mathbb{E}_{x^*, y^* \sim p}[\text{rm}(x^*, y^*, \theta)])^2] \]
\[ = \mathbb{E}_{x,y \sim p}[\mathbb{E}_{\theta, \theta' \sim q}[(\text{rm}(x, y, \theta) - \mathbb{E}_{x^*, y^* \sim p}[\text{rm}(x^*, y^*, \theta))] (\text{rm}(x, y, \theta') - \mathbb{E}_{x^*, y^* \sim p}[\text{rm}(x, y, \theta')])] \]
Breaking the variance down

\[ \text{Var}[\text{margin}] = \mathbb{E}_{x,y \sim p} [ (\text{margin}(x, y) - \mathbb{E}_{x^*, y^* \sim p} [\text{margin}(x^*, y^*)])^2 ] \]

\[ = \mathbb{E}_{x,y \sim p} [ (\mathbb{E}_{\theta \sim q} [rm(x, y, \theta)]) - \mathbb{E}_{x^*, y^* \sim p} [\mathbb{E}_{\theta \sim q} [rm(x^*, y^*, \theta)]]]^2 ] \]

\[ = \mathbb{E}_{x,y \sim p} [ (\mathbb{E}_{\theta \sim q} [rm(x, y, \theta)]) - \mathbb{E}_{x^*, y^* \sim p} [rm(x^*, y^*, \theta)]]^2 ] \]

\[ = \mathbb{E}_{x,y \sim p} [ \mathbb{E}_{\theta, \theta' \sim q} [rm(x, y, \theta) - \mathbb{E}_{x^*, y^* \sim p} [rm(x^*, y^*, \theta)]](rm(x, y, \theta') - \mathbb{E}_{x^*, y^* \sim p} [rm(x, y, \theta')])] \]

\[ = \mathbb{E}_{\theta, \theta' \sim q} \mathbb{E}_{x,y \sim p} [rm(x, y, \theta) - \mathbb{E}_{x^*, y^* \sim p} [rm(x^*, y^*, \theta)](rm(x, y, \theta') - \mathbb{E}_{x^*, y^* \sim p} [rm(x^*, y^*, \theta')])]] \]
Breaking the variance down

\[ \text{Var}[	ext{margin}] = \mathbb{E}_{x,y \sim p}[(\text{margin}(x,y) - \mathbb{E}_{x^*,y^* \sim p}[\text{margin}(x^*,y^*)])^2] \]

\[ = \mathbb{E}_{x,y \sim p}[(\mathbb{E}_{\theta \sim q}\text{rm}(x,y,\theta)) - \mathbb{E}_{x^*,y^* \sim p}[\mathbb{E}_{\theta \sim q}\text{rm}(x^*,y^*,\theta)])^2] \]

\[ = \mathbb{E}_{x,y \sim p}[(\mathbb{E}_{\theta \sim q}\text{rm}(x,y,\theta)) - \mathbb{E}_{\theta \sim q}\mathbb{E}_{x^*,y^* \sim p}[\text{rm}(x^*,y^*,\theta)])^2] \]

\[ = \mathbb{E}_{x,y \sim p}[(\mathbb{E}_{\theta \sim q}\text{rm}(x,y,\theta)) - \mathbb{E}_{x^*,y^* \sim p}[\text{rm}(x^*,y^*,\theta)])^2] \]

\[ = \mathbb{E}_{x,y \sim p}[\mathbb{E}_{\theta,\theta' \sim q}(\text{rm}(x,y,\theta) - \mathbb{E}_{x^*,y^* \sim p}[\text{rm}(x^*,y^*,\theta)])] \]

\[ (\text{rm}(x,y,\theta') - \mathbb{E}_{x^*,y^* \sim p}[\text{rm}(x,y,\theta')]) \]

\[ = \mathbb{E}_{\theta,\theta' \sim q}\mathbb{E}_{x,y \sim p}[(\text{rm}(x,y,\theta) - \mathbb{E}_{x^*,y^* \sim p}[\text{rm}(x^*,y^*,\theta)])] \]

\[ (\text{rm}(x,y,\theta') - \mathbb{E}_{x^*,y^* \sim p}[\text{rm}(x^*,y^*,\theta')]) \]

\[ = \mathbb{E}_{\theta,\theta' \sim q}\text{Cov}_{x,y \sim p}[\text{rm}(x,y,\theta), \text{rm}(x,y,\theta')] \]
Breaking the variance down

\[
\text{Var}[\text{margin}] = \mathbb{E}_{\theta, \theta' \sim q} \text{Cov}_{x,y \sim p}[\text{rm}(x, y, \theta), \text{rm}(x, y, \theta')] \\
= \mathbb{E}_{\theta, \theta' \sim q} \left[ \rho_{x,y \sim p}(\text{rm}(x, y, \theta), \text{rm}(x, y, \theta')) \right] \\
\sqrt{\text{Var}_{x,y \sim p}[\text{rm}(x, y, \theta)]} \sqrt{\text{Var}_{x,y \sim p}[\text{rm}(x, y, \theta')]} \right]
\]
Breaking the variance down

\[ \text{Var[margin]} = \mathbb{E}_{\theta, \theta' \sim q} \text{Cov}_{x, y \sim p}[\text{rm}(x, y, \theta), \text{rm}(x, y, \theta')] \]

\[ = \mathbb{E}_{\theta, \theta' \sim q} \left[ \rho_{x, y \sim p}(\text{rm}(x, y, \theta), \text{rm}(x, y, \theta')) \right] \]

\[ \sqrt{\text{Var}_{x, y \sim p}[\text{rm}(x, y, \theta)]} \sqrt{\text{Var}_{x, y \sim p}[\text{rm}(x, y, \theta')]} \]

Define average correlation \( \bar{\rho} \)

\[ \bar{\rho} = \frac{\mathbb{E}_{\theta, \theta' \sim q} \left[ \rho(\theta, \theta') \sqrt{\text{Var}[\theta]} \sqrt{\text{Var}[\theta']} \right]}{\mathbb{E}_{\theta, \theta' \sim q} \left[ \sqrt{\text{Var}[\theta]} \sqrt{\text{Var}[\theta']} \right]} \]
Breaking the variance down

\[ \mathbb{V} \text{ar}[\text{margin}] = \mathbb{E}_{\theta, \theta' \sim q} \mathbb{C} \text{ov}_{x,y \sim p}[r_{m}(x,y, \theta), r_{m}(x,y, \theta')] \]

\[ = \mathbb{E}_{\theta, \theta' \sim q} \left[ \rho_{x,y \sim p}(r_{m}(x,y, \theta), r_{m}(x,y, \theta')) \frac{\sqrt{\mathbb{V} \text{ar}_{x,y \sim p}[r_{m}(x,y, \theta)]}}{\sqrt{\mathbb{V} \text{ar}_{x,y \sim p}[r_{m}(x,y, \theta')]}]} \right] \]

Define average correlation \( \bar{\rho} \)

\[ \bar{\rho} = \frac{\mathbb{E}_{\theta, \theta' \sim q} \left[ \rho(\theta, \theta') \sqrt{\mathbb{V} \text{ar}[\theta]} \sqrt{\mathbb{V} \text{ar}[\theta']} \right]}{\mathbb{E}_{\theta, \theta' \sim q} \left[ \sqrt{\mathbb{V} \text{ar}[\theta]} \sqrt{\mathbb{V} \text{ar}[\theta']} \right]} \]

We get

\[ \mathbb{V} \text{ar}[\text{margin}] = \bar{\rho} \mathbb{E}_{\theta \sim q} [\sqrt{\mathbb{V} \text{ar}[\theta]}]^2 \leq \bar{\rho} \mathbb{E}_{\theta \sim q} [\mathbb{V} \text{ar}[\theta]] \]
Breaking the variance down

Simplifying the variance

\[
\mathbb{E}_{\theta \sim q}[\text{Var}[\theta]] = \mathbb{E}_{\theta \sim q}[\mathbb{E}[\text{rm}_{x,y \sim p}(x, y, \theta)]^2] - \mathbb{E}_{\theta \sim q}[\mathbb{E}_{x,y \sim p}[\text{rm}(x, y, \theta)]]^2 \\
= \mathbb{E}_{\theta \sim q}[\mathbb{E}[\text{rm}_{x,y \sim p}(x, y, \theta)]^2] - s^2 \\
\leq 1 - s^2
\]

Implies that

\[
\text{Var}[\text{margin}] \leq \bar{\rho}(1 - s^2)
\]
Putting it all together

What we just did?
Putting it all together

What we just did?

We rewrote the variance of the margin to get an interpretable generalization error
A Generalization Error Bound For Random Forests

\[ ge \leq \frac{\text{Var}[\text{margin}]}{\mathbb{E}[\text{margin}]^2} = \frac{\text{Var}[\text{margin}]}{s^2} \leq \bar{\rho}(1 - s^2) \]

Good forests have
- Low correlation between trees (\(\rho\))
- 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 $\rightarrow$ correlated trees
A way to grow forests

1. Sample $m$ data points from the training set $(x_1, y_1), \ldots, (x_n, y_n)$
2. Grow a decision tree
3. Each split can only use a random subset of features
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)

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

[Caruana; 2006]
Random Forests: Computation

Each tree in random forest requires

\[ \theta_k \sim q \]

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 $2^{n-1}$
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 $2^{n-1}$

*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 $2^{n-1}$

*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 $2^{n-1}$

*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 $2^{n-1}$

*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

\[ \mathcal{L}(f) = \mathbb{E}_{y,x} d(y,f(x)) = \mathbb{E}_x \mathbb{E}_y |_x [d(y,f(x))] \]

Suppose we have a current \( f^* \)

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

Derivative

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

Interchanging differentiation and integration

\[ g_m(x) = \mathbb{E}_y \left[ \frac{\partial d(y,f(x))}{\partial f(x)} \bigg|_x \right]_{f=f_{m-1}} \]
Functional Gradients

Update with step size $\rho_m$

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

$\rho_m$ can be set by line search to minimize loss

A quick example

$$g_m(x) = \mathbb{E}_y \left[ \frac{\partial \left[ \frac{1}{2} (y - f(x))^2 \right]}{\partial f(x)} \bigg| x \right]_{f=f_{m-1}}$$

$$= \mathbb{E}_y [f(x) - y \bigg| x]_{f=f_{m-1}}$$

$$= \mathbb{E}_y [f_{m-1}(x) - y \bigg| x]$$
Functional Gradients

Update with step size $\rho_m$

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

$\rho_m$ can be set by line search to minimize loss

A quick example

$$g_m(x) = \mathbb{E}_y \left[ \frac{\partial \left[ \frac{1}{2} (y - f(x))^2 \right]}{\partial f(x)} \bigg| x \right]_{f=f_{m-1}}$$

$$= \mathbb{E}_y [f(x) - y \big| x]_{f=f_{m-1}}$$

$$= \mathbb{E}_y [f_{m-1}(x) - y \big| x]$$

Problems?
\[ g_m(x) = \mathbb{E}_y [f_{m-1}(x) - y | x] \]

Hard to compute given only finite data. Options?

- Optimize over a parametric family
- Greedy stagewise approach
Greedy Function Optimization

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

Estimate \( h \) by two steps

1. Standard gradient at training points

\[ -g_m(x_i) = -\frac{\partial d(y_i, f(x_i))}{\partial f(x_i)} \bigg|_{f=f_{m-1}} \]

2. Project standard gradient to family \( h \)

\[ \min_a \sum_{i=1}^{N} \left[ -g_m(x_i) - h(x_i; a_m) \right]^2 \]
Greedy Function Optimization

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

$$\rho_m = \arg\min_{\rho} \sum_{i=1}^{n} d(y_i, f_{m-1}(x_i) + \rho h(x_i; a_m))$$

- A kind of projected functional gradient descent
- Step size set by line search
Example: Gradient Boosted Regression Trees

Prediction error: Mean squared error

\[
d(y_i, f(x_i)) = \frac{1}{2} (y_i - f(x_i))^2
\]

Current predictor

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

Compute derivative at \( f_{m-1} \)

\[
-\left. \frac{\partial d(y_i, f(x_i))}{\partial f(x_i)} \right|_{f=f_{m-1}} = (y_i - f(x_i))
\]

Need to project derivatives
How do we project derivatives with trees?
How do we project derivatives with trees?

\[ \hat{y}_i = y_i - f(x_i) \]

Fit a decision tree on \((\hat{y}_i, x_i)\) 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
Random forests
- Don’t really overfit with more computational steps
- Embarrassingly parallel training
- Need to randomly find the right input region

Gradient Boosted Trees
- Overfits 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