Ensemble Methods: Bagging

Instructor: Jessica Wu -- Harvey Mudd College

The instructor gratefully acknowledges Eric Eaton (UPenn), Jenna Wiens (UMich), Tommi Jaakola (MIT), David Kauchak (Pomona), David Sontag (NYU), Piyush Rai (Utah), and the many others who made their course materials freely available online.

Ensemble Methods Basics

Learning Goals

- Describe the goal of ensemble methods

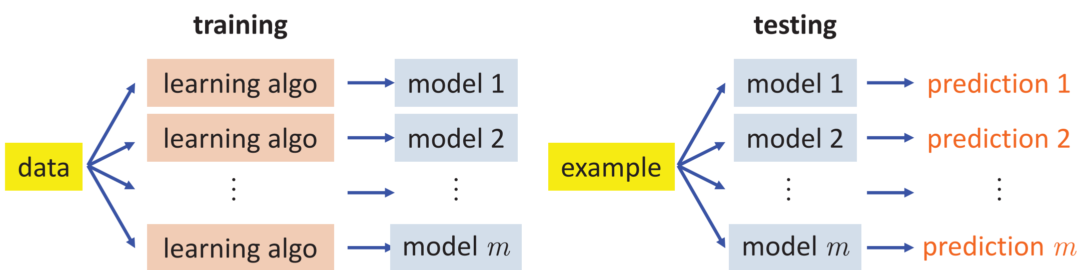
Basic Idea

If one classifier works well…
why not use multiple classifiers!?!

Ensemble: "All the parts of a thing taken together so that each part is considered in relation to the whole."

ML Ensemble:

Combine multiple classifiers to improve prediction

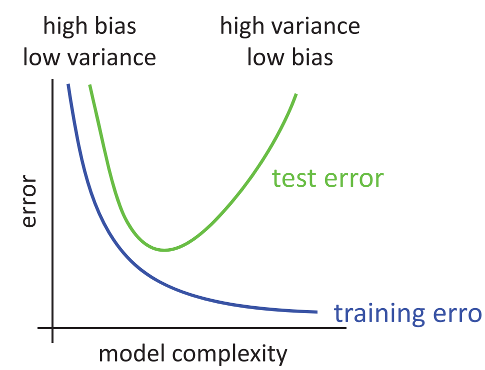


Based on slide by David Kauchak

Bias vs Variance

Recall ML balances

- estimation error (variance)
  - precision of match
  - sensitivity to training data
- structural error (bias)
  - distance from true relationship



Goals

- Reduce variance without increasing bias
- Reduce bias and variance

Benefits of Ensemble Learning

Setup (for example)

- Assume a binary classification problem
- Suppose that we have trained 3 classifiers $h^1(\boldsymbol{x}),\ h^2(\boldsymbol{x}),\ h^3(\boldsymbol{x})$, each with a misclassification rate of 0.4
- Assume that decisions made between classifiers are independent
- Take majority vote of classifiers

What is the probability that we make a mistake?

model 1	model 2	model 3	probability of this combination
c	c	c	

Benefits of Ensemble Learning

In general, for m classifiers with r probability of mistake,

$$h(\boldsymbol{x}) = \operatorname*{arg\,max}_{y \in \{+1,-1\}} \sum_{b=1}^{m} \mathbb{I}[[h^b(\boldsymbol{x}) = y]]$$

Let $B = \sum_{b=1}^{m} \mathbb{I}[[h^b(\boldsymbol{x}) = \tilde{y}]]$ be # of "votes" for wrong class.

Then $B \sim \text{binomial}(m,\ r)$.

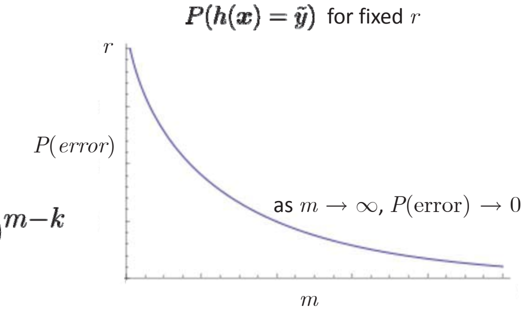
$$P(k; m, r) = \binom{m}{k} r^k (1-r)^{m-k}$$

$$P(h(\boldsymbol{x}) = \tilde{y}) = P(k > \tfrac{m}{2}; m, r)$$

[cumulative probability distribution for binomial]

$$= \sum_{k=\frac{m+1}{2}}^{m} \binom{m}{k} r^k (1-r)^{m-k}$$

$P(h(\boldsymbol{x}) = \tilde{y})$ for fixed r



Based on example by David Kauchak

Bagging and Random Forests

Learning Goals

- Describe bagging
  - How does bagging improve performance?
- Describe random forests
  - How is random forest related to DTs and bagging?

Combining Classifiers

- averaging
  - final hypothesis is simple vote of members
- weighted averaging
  - coefficients of individual members trained using validation set
- stacking
  - predictions of one layer used as input to next layer

averaging reduce variance (without increasing bias)

$$\mathrm{var}(\bar{X}) = \frac{\mathrm{var}(X)}{m}$$

(when predictions are **independent**)

But how do we get independent classifiers?

Based on slides by Eric Eaton and David Sontag

Obtaining Independent Classifiers

Idea: Use different learning methods

Pros
- Lots of existing classifiers
- Can work well for some problems

Cons
- Often classifiers are not independent (they make same mistakes!)
  - E.g. many classifiers are linear models
  - Voting will not help if they make the same mistakes

Successful ensembles require **diversity**!

achieving diversity

cause of mistake	diversification strategy
difficult pattern	
overfitting	
noisy features	

Based on slides by David Kauchak and Eric Eaton

Obtaining Independent Classifiers

Idea: Split up training data

(use same learning algorithm but train of different parts of the data)

Pros
- Learning from different data so cannot overfit to same examples
- Easy to implement
- Fast

Cons
- Each classifier only trained on small amount (e.g. 10%) of data
- Not clear why this would do better than training on full data and using good regularization

Based on slide by David Kauchak

Obtaining Independent Classifiers

Idea: **Bagging** (Bootstrap Aggregation)

Bootstrap Sampling

- Use training data as proxy for data-generating distribution
- Given S_n with n training examples, create S_n' by sampling with replacement n examples

Bagging

- Create m bootstrap samples $S_n^{(1)}, \ldots, S_n^{(m)}$
- Train distinct classifier on each $S_n^{(i)}$
- Classify new example by majority vote / averaging

Based on notes and slides by Eric Eaton and David Sontag

Bagging Best Case Scenario

$$\mathrm{var}(bagging(L(h, S_n))) = \frac{\mathrm{var}(L(h, S_n))}{m}$$

In practice:

- models correlated so reduction smaller than $1/m$
- model trained on fewer training samples so variance usually somewhat larger

Based on slide by David Sontag

Bagging Concerns

Will bootstrap samples all be basically the same?

For data set of size n, what is the probability that a given example will <u>not</u> be selected in a bootstrap sample?

probability that example is not chosen the first time = _____

probability that example is not chosen any of n times = ________

for large n, $\exp(x) = \lim_{n \to \infty} \left(1 + \frac{x}{n}\right)^n$

so ________ converges to ________

probability that example is chosen in any of n times = __________

on average,
a bootstrap sample contains _____ of training examples

Based on notes and slides by Jenna Wiens and David Kauchak

When does Bagging work?

- Bagging tends to reduce the variance of the classifier
  - By voting, the ensemble classifier is more robust to noisy examples
- Bagging is most useful for classifiers that are
  - Unstable
    - small changes in training set produce very different models
  - Prone to overfitting

What classifiers have we seen so far with these properties?

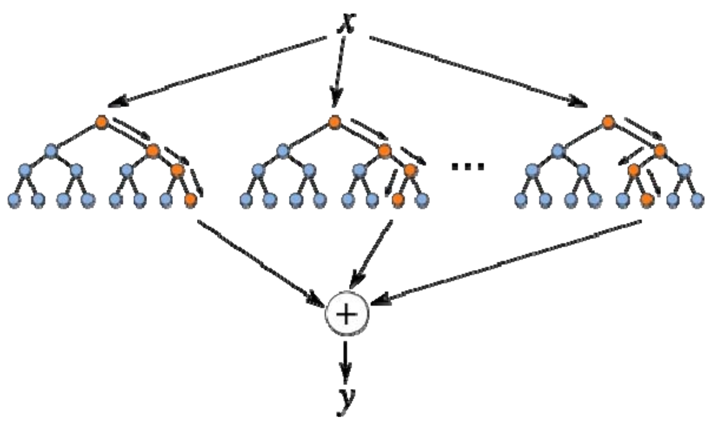
- Often has similar effect to regularization

Based on slide by David Kauchak

Random Forests

Introduces two sources of randomness

- bagging
- random feature subset
  - at each node, best split is chosen from random subset of $k < d$ features



Based on notes by Jenna Wiens [Image by Léon Bottou]

Random Forest Procedure

for $b = 1, \ldots, m$

  draw bootstrap sample $S_n^{(b)}$ of size n from S_n

  grow random forest DT$^{(b)}$

output ensemble

[subprocedure for growing DT$^{(b)}$]

until stopping criteria are met, recursively repeat following steps for each node of tree

  i) select k features at random from d features

  ii) pick best feature to split on using info gain

  iii) split node into children

Based on notes by Jenna Wiens