Machine Learning

CS311
David Kauchak
Spring 2013

Some material borrowed from:
Sara Owsley Sood and others

Admin

Two talks this week
- Tuesday lunch
- Thursday lunch

Midterm exam posted later today

Assignment 4
- How's it going?
- Due Friday at 6pm

No office hours on Friday

Classifiers so far

Naïve Bayes

k-nearest neighbors (k-NN)

Both fairly straightforward to understand and implement

How do they work?

Many, many classifiers

- Linear classifiers
  - Fisher's linear discriminant
  - Logistic regression
  - Naive Bayes classifier
  - Perceptron
- Support vector machines
  - Least squares support vector machines
- Quadratic classifiers
- Kernel estimation
  - k-nearest neighbor
- Boosting (meta-algorithm)
- Decision trees
  - Random forests
- Neural networks
- Gene Expression Programming
- Bayesian networks
- Hidden Markov models
- Learning vector quantization
- Proaftn

http://en.wikipedia.org/wiki/Statistical_classification

- AODE
- Artificial neural network
  - Backpropagation
- Bayesian statistics
  - Naive Bayes classifier
  - Bayesian network
  - Bayesian knowledge base
- Case-based reasoning
- Decision trees
- Inductive logic programming
- Gaussian process regression
- Gene expression programming
- Group method of data handling (GMDH)
- Learning Automata
- Learning Vector Quantization
- Logistic Model Tree
- Minimum message length (decision trees, decision graphs, etc.)
- Lazy learning
- Instance-based learning
  - Nearest Neighbor Algorithm
  - Analogical modeling
- Probably approximately correct learning (PAC) learning
- Ripple down rules, a knowledge acquisition methodology
- Symbolic machine learning algorithms
- Subsymbolic machine learning algorithms
- Support vector machines
- Random Forests
- Ensembles of classifiers
  - Bootstrap aggregating (bagging)
  - Boosting (meta-algorithm)
- Ordinal classification
- Regression analysis
- Information fuzzy networks (IFN)

http://en.wikipedia.org/wiki/List_of_machine_learning_algorithms

Many, many classifiers: what we will cover

Problem setup
- Training vs. testing
- Feature-based learning
- Evaluation

Introduction to a few models

Model comparison
- When to apply one vs the other (maybe…)
- How models differ
- Pros/cons

Many, many classifiers: what we won't cover

Quite a few models

Won't dive too much into the theoretical underpinnings

meta-learning (i.e. combining classifiers)

recommender systems aka collaborative filtering (but can be viewed as a classification problem)

Bias/Variance

Bias: How well does the model predict the training data?
- high bias – the model doesn't do a good job of predicting the training data (high training set error)
- The model predictions are *biased by the model*

Variance: How sensitive to the training data is the learned model?
- high variance – changing the training data can drastically change the learned model

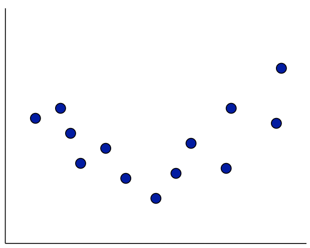
Bias/Variance

Another way to think about it is model complexity

Simple models
- may not model data well
- high bias

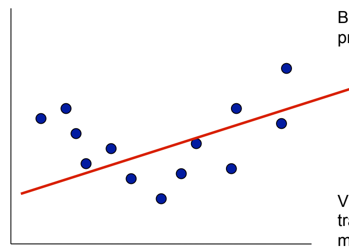
Complicated models
- may overfit to the training data
- high variance

Bias/variance trade-off



We want to fit a polynomial to this, which one should we use?

Bias/variance trade-off



High variance OR high bias?

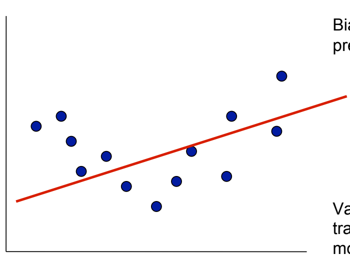
Bias: How well does the model predict the training data?

- high bias – the model doesn't do a good job of predicting the training data (high training set error)
- The model predictions are biased by the model

Variance: How sensitive to the training data is the learned model?

- high variance – changing the training data can drastically change the learned model

Bias/variance trade-off



High bias

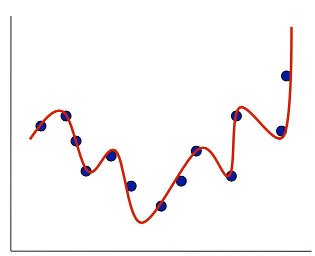
Bias: How well does the model predict the training data?

- high bias – the model doesn't do a good job of predicting the training data (high training set error)
- The model predictions are biased by the model

Variance: How sensitive to the training data is the learned model?

- high variance – changing the training data can drastically change the learned model

Bias/variance trade-off



High variance OR high bias?

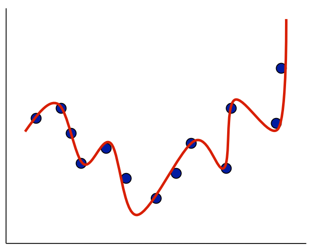
Bias: How well does the model predict the training data?

- high bias – the model doesn't do a good job of predicting the training data (high training set error)
- The model predictions are biased by the model

Variance: How sensitive to the training data is the learned model?

- high variance – changing the training data can drastically change the learned model

Bias/variance trade-off

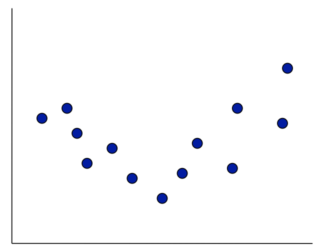


High variance

Bias: How well does the model predict the training data?
- high bias – the model doesn't do a good job of predicting the training data (high training set error)
- The model predictions are biased by the model

Variance: How sensitive to the training data is the learned model?
- high variance – changing the training data can drastically change the learned model

Bias/variance trade-off

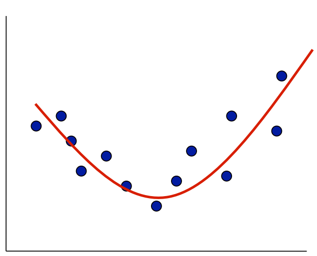


What do we want?

Bias: How well does the model predict the training data?
- high bias – the model doesn't do a good job of predicting the training data (high training set error)
- The model predictions are biased by the model

Variance: How sensitive to the training data is the learned model?
- high variance – changing the training data can drastically change the learned model

Bias/variance trade-off



Compromise between bias and variance

Bias: How well does the model predict the training data?
- high bias – the model doesn't do a good job of predicting the training data (high training set error)
- The model predictions are biased by the model

Variance: How sensitive to the training data is the learned model?
- high variance – changing the training data can drastically change the learned model

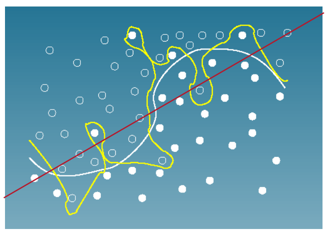
k-NN vs. Naive Bayes

How do k-NN and NB sit on the variance/ bias spectrum?

k-NN has high variance and low bias.
- more complicated model
- can model any boundary
- but very dependent on the training data

NB has low variance and high bias.
- Decision surface has to be linear (more on this later)
- Cannot model all data
- but, less variation based on the training data

Bias vs. variance: Choosing the correct model capacity



Which separating line should we use?

Playing tennis

You want to decide whether or not to play tennis today
- Outlook: Sunny, Overcast, Rain
- Humidity: High, normal
- Wind: Strong, weak

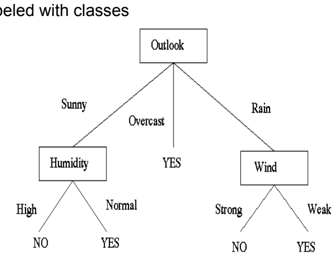
Tell me what you're classifier should do?

Decision tree is an intuitive way of representing a decision

Tree with internal nodes labeled by features

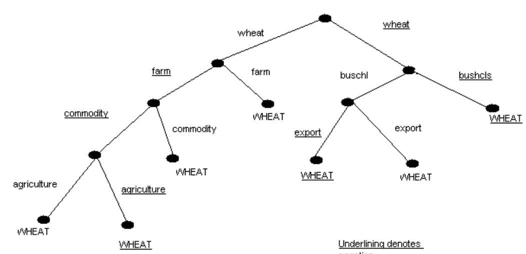
Branches are labeled by tests on that feature
- outlook = sunny
- x > 100

Leaves labeled with classes



Another decision tree

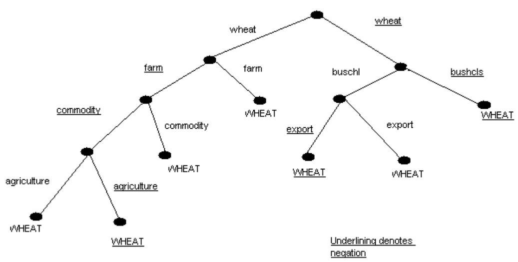
Document classification: wheat or not wheat?



Another decision tree

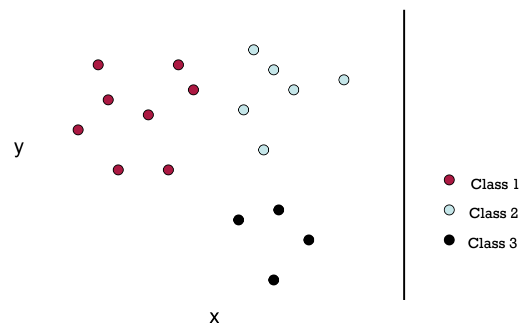
Document: wheat, agriculture, buschl

Which category?



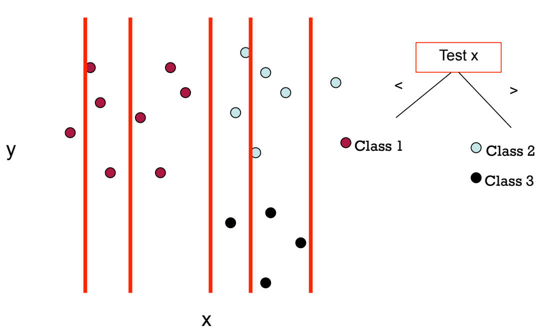
Decision tree learning

What does a decision node look like in the feature space?



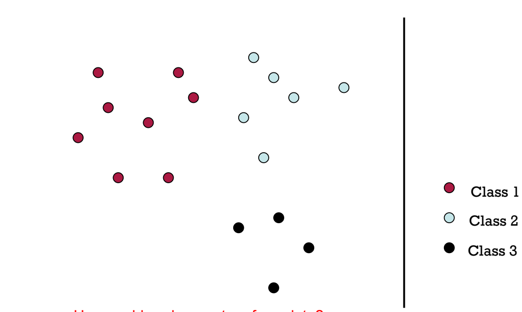
Decision tree learning

A node in the tree is threshold on that dimension



Decision tree learning

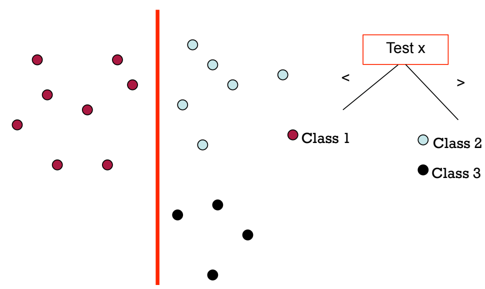
Features are x and y
A node in the tree is threshold on that dimension



How could we learn a tree from data?

Decision tree learning

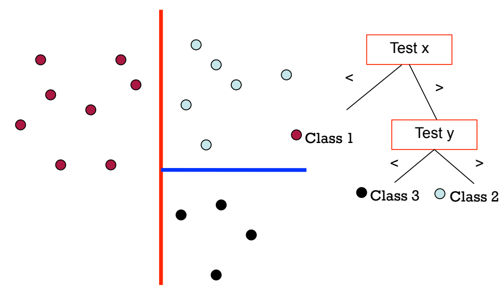
Features are x and y
A node in the tree is threshold on that dimension



How could we learn a tree from data?

Decision tree learning

Features are x and y
A node in the tree is threshold on that dimension



How could we learn a tree from data?

Decision Tree Learning

Start at the top and work our way down

- Examine all of the features to see which feature best separates the data
- Split the data into subsets based on the feature test
- Test the *remaining* features to see which best separates the data in each subset
- Repeat this process in all branches until:

Decision Tree Learning

Start at the top and work our way down

- Examine all of the features to see which feature best separates the data
- Split the data into subsets based on the feature test
- Test the *remaining* features to see which best separates the data in each subset
- Repeat this process in all branches until:
  - all examples in a subset are of the same type
  - there are no examples left (or some small number left)
  - there are no attributes left

Decision Tree Learning

Start at the top and work our way down

- Examine all of the features to see **which feature best separates the data**
- Split the data into subsets based on the feature test
- Test the *remaining* features to see which best separates the data in each subset
- Repeat this process in all branches until:
  - all examples in a subset are of the same type
  - there are no examples left
  - there are no attributes left

Ideas?

KL-Divergence

Given two probability distributions P and Q

$$D_{KL}(P \parallel Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$$

When is this large? small?

KL-Divergence

Given two probability distributions P and Q

$$D_{KL}(P \parallel Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$$

When P = Q, $D_{KL}(P||Q) = 0$

KL-Divergence

Given two probability distributions P and Q

$$D_{KL}(P \parallel Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$$

P(1) = 0.999	Q(1) = 0.001
P(2) = 0.001	Q(2) = 0.999

$D_{KL}(P||Q) = 6.89$

KL-divergence is a measure of the distance between two probability distributions (though it's not a distance metric!)

Information Gain

$$D_{KL}(P(class \mid f) \parallel P(class)) = \sum_{c \in class} P(c \mid f) \log \frac{P(c \mid f)}{P(c)}$$

What is this asking?

KL-divergence is a measure of the distance between two probability distributions (though it's not a distance metric!

Information Gain

$$D_{KL}(P(class \mid f) \parallel P(class)) = \sum_{c \in class} P(c \mid f) \log \frac{P(c \mid f)}{P(c)}$$

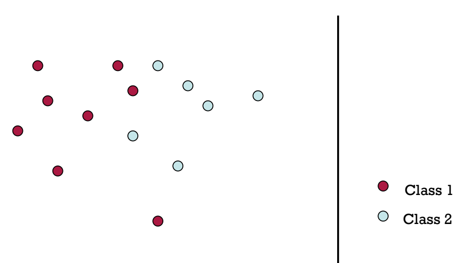
What is the distance from the probability of a class (i.e. the prior) and the probability of that class conditioned on *f* ?

What information do we gain about the class decision, given the feature *f* ?

Use information gain to decide the most informative feature to split on

Decision tree learning

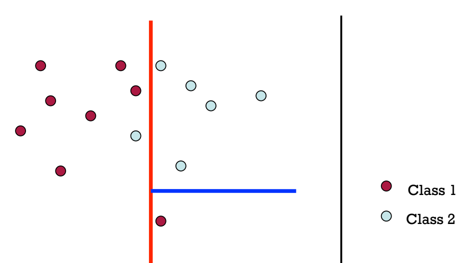
Features are x and y
A node in the tree is threshold on that dimension



What would be the learned tree?

Decision tree learning

Features are x and y
A node in the tree is threshold on that dimension



Do you think this is right?

Overfitting

Decision trees: high-variance or high-bias?

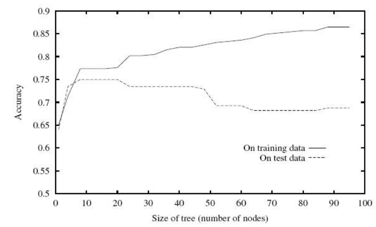
Bias: How well does the model predict the training data?
- high bias – the model doesn't do a good job of predicting the training data (high training set error)
- The model predictions are biased by the model

Variance: How sensitive to the training data is the learned model?
- high variance – changing the training data can drastically change the learned model

Overfitting

Decision trees can have a high variance

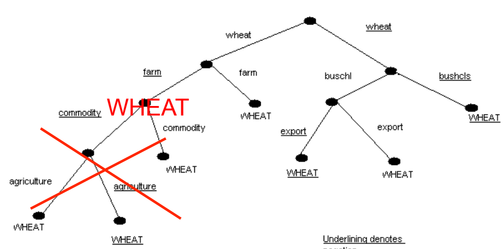
The model can be too complicated and *overfit* to the training data



Ideas?

Pruning

Ideas?



Pruning

Measure accuracy on a hold-out set (i.e. not used for training)
- Stop splitting when when accuracy decreases
- Prune tree from the bottom up, replacing split nodes with majority label, while accuracy doesn't decrease

Other ways look at complexity of the model with respect to characteristics of the training data
- Don't split if information gain gets below a certain threshold
- Don't split if number of examples is below some threshold
- Don't split if tree depth goes beyond a certain depth
- …

Decision trees: good and bad

Good
- Very human friendly
  - easy to understand
  - people can modify
- fairly quick to train

Bad
- overfitting/pruning can be tricky
- greedy approach: if you make a split you're stuck with it
- performance is ok, but can do better

Midterm

Open book
- still only 2 hours, so don't rely on it too much

Anything we've talked about in class or read about is fair game

Written questions are a good place to start

Review

Intro to AI
- what is AI
- goals
- challenges
- problem areas

Review

Uninformed search
- reasoning through search
- agent paradigm (sensors, actuators, environment, etc.)
- setting up problems as search
  - state space (starting state, next state function, goal state)
  - actions
  - costs
- problem characteristics
  - observability
  - determinism
  - known/unknown state space
- techniques
  - BFS
  - DFS
  - uniform cost search
  - depth limited search
  - Iterative deepening

Review

Uninformed search cont.

- things to know about search algorithms
  - time
  - space
  - completeness
  - optimality
  - when to use them
- graph search vs. tree search

Informed search

- heuristic function
  - admissibility
  - combining functions
  - dominance
- methods
  - greedy best-first search
  - A*

Review

Adversarial search

- game playing through search
  - ply
  - depth
  - branching factor
  - state space sizes
  - optimal play
- game characteristics
  - observability
  - # of players
  - discrete vs. continuous
  - real-time vs. turn-based
  - determinism

Review

Adversarial search cont

- minimax algorithm
- alpha-beta pruning
  - optimality, etc.
- evalution functions (heuristics)
  - horizon effect
- improvements
  - transposition table
  - history/end-game tables
- dealing with chance/non-determinism
  - expected minimax
- dealing with partially observable games

Review

Local search

- when to use/what types of problems
- general formulation
- hill-climbing
  - greedy
  - random restarts
  - randomness
  - simulated annealing
  - local beam search
- genetic algorithms

Review

Basic probability

- why probability (vs. say logic)?
- vocabulary
  - experiment
  - sample
  - event
  - random variable
  - probability distribution
- unconditional/prior probability
- joint distribution
- conditional probability
- Bayes rule
- estimating probabilities

Review

Machine learning (up through last Thursday)

- Bayesian classification
  - problem formulation, argmax, etc.
  - NB model
- k-nearest neighbor
- training, testing, evaluation
- bias vs. variance
- model characteristics