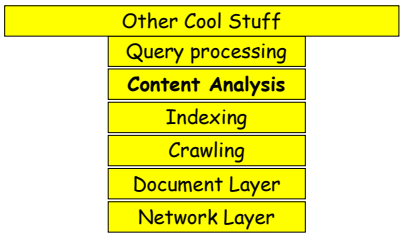
Machine Learning

CSE 454

Administrivia

- PS1 due next tues 10/13
- Project proposals also due then
- Group meetings with Dan
  Signup out shortly

Class Overview



Today's Outline

- Brief supervised learning review
- Evaluation
- Overfitting
- Ensembles
  Learners: The more the merrier
- Co-Training
  (Semi) Supervised learning with few labeled training ex

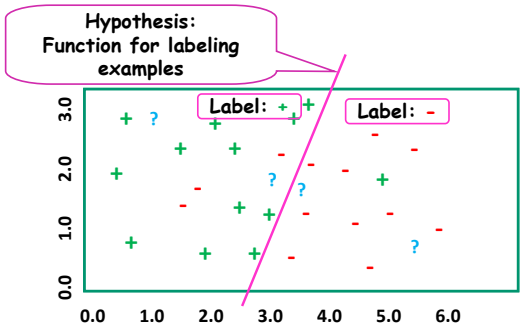
Types of Learning

- **Supervised (inductive) learning**
  Training data includes desired outputs
- **Semi-supervised learning**
  Training data includes a ***few*** desired outputs
- **Unsupervised learning**
  Training data ***doesn't*** include desired outputs
- **Reinforcement learning**
  Rewards from sequence of actions

Supervised Learning

- **Inductive learning** or "**Prediction**":
  Given examples of a function *(X, F(X))*
  Predict function *F(X)* for new examples *X*
- Classification
  F(X) = Discrete
- Regression
  F(X) = Continuous
- Probability estimation
  F(X) = Probability*(X):*

Classifier



Bias

- Which hypotheses ***will you consider***?
- Which hypotheses do you ***prefer***?

Naïve Bayes

- Probabilistic classifier:
  $P(C_i \mid \text{Example})$
- Bias?
- Assumes all features are conditionally independent given class

$$P(E \mid c_i) = P(e_1 \wedge e_2 \wedge \cdots \wedge e_m \mid c_i) = \prod_{j=1}^{m} P(e_j \mid c_i)$$

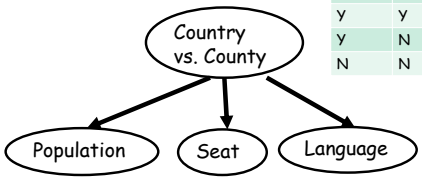
- Therefore, we then only need to know $P(e_j \mid c_i)$ for each feature and category

Naïve Bayes for Text

- Modeled as generating a bag of words for a document in a given category
- Assumes that word order is unimportant,
  only cares whether word appears in document
- Smooth probability estimates with Laplace *m*-estimates
  assuming uniform distribution over words ($p = 1/|V|$) and $m = |V|$
  Equivalent to a virtual sample of seeing each word in each category exactly once.

Naïve Bayes

Pop.	Seat	Lang.	Class
Y	Y	N	County
Y	Y	Y	County
Y	N	Y	Country
N	N	Y	Country

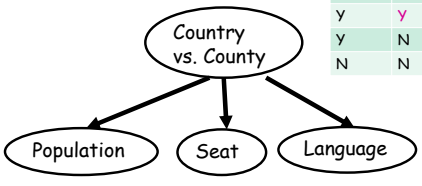


Probability(Seat | County) = ??

Probability(Seat | Country) = ??

Naïve Bayes

Pop.	Seat	Lang.	Class
Y	Y	N	County
Y	Y	Y	County
Y	N	Y	Country
N	N	Y	Country

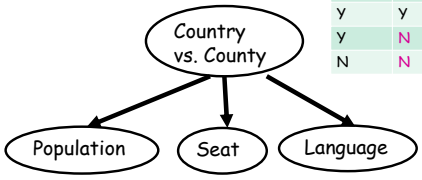


Probability(Seat | County) = 2 + 1 / 2 + 1 = 1.0

Probability(Seat | Country) = ??

Naïve Bayes

Pop.	Seat	Lang.	Class
Y	Y	N	County
Y	Y	Y	County
Y	N	Y	Country
N	N	Y	Country



Probability(Seat | County) = 2 + 1/ 2 + 2 = 0.75

Probability(Seat | Country) = 0 + 1 / 2 + 2 = 0.25

Probabilities: Important Detail!

- $P(spam \mid E_1 \ldots E_n) = \prod_i P(spam \mid E_i)$

Any more potential problems here?

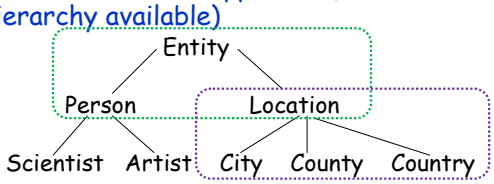
- **We are multiplying lots of small numbers**
  Danger of underflow!
  - $0.5^{57} = 7\ E\ -18$
- **Solution? Use logs and add!**
  - $p_1 * p_2 = e^{\log(p1)+\log(p2)}$
  - **Always keep in log form**

Multi-Class Categorization

- Pick the category with max probability
- Create many 1 vs other classifiers
  Classes = City, County, Country
  Classifier 1 = {City} {County, Country}
  Classifier 2 = {County} {City, Country}
  Classifier 3 = {Country} {City, County}

Multi-Class Categorization

- Use a hierarchical approach (wherever hierarchy available)



Today's Outline

- Brief supervised learning review
- Evaluation
- Overfitting
- Ensembles
  Learners: The more the merrier
- Co-Training
  (Semi) Supervised learning with few labeled training ex

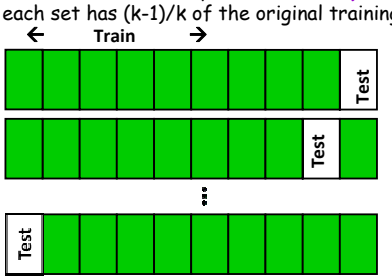
Experimental Evaluation

Question: How do we estimate the performance of classifier on unseen data?

- Can't just at accuracy on training data – this will yield an over optimistic estimate of performance
- Solution: Cross-validation
- Note: this is sometimes called estimating how well the classifier will generalize

Evaluation: Cross Validation

- Partition examples into *k* disjoint sets
- Now create *k* training sets
  Each set is union of all equiv classes ***except one***
  So each set has (k-1)/k of the original training data



Cross-Validation (2)

- Leave-one-out
  Use if < 100 examples (rough estimate)
  Hold out one example, train on remaining examples
- 10-fold
  If have 100-1000's of examples
- M of N fold
  Repeat M times
  Divide data into N folds, do N fold cross-validation

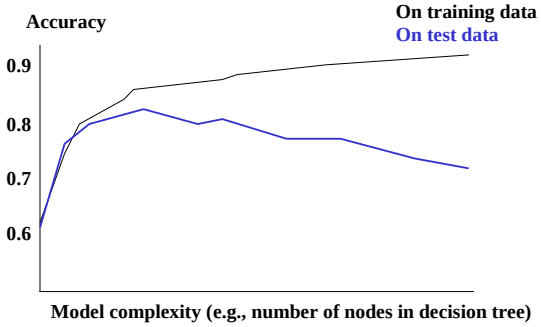
Today's Outline

- Brief supervised learning review
- Evaluation
- Overfitting
- Ensembles
  Learners: The more the merrier
- Co-Training
  (Semi) Supervised learning with few labeled training ex
- Clustering
  No training examples

Overfitting Definition

- Hypothesis H is *overfit* when ∃ H' and
  H has ***smaller*** error on training examples, but
  H has ***bigger*** error on test examples
- Causes of overfitting
  Noisy data, or
  Training set is too small
  Large number of features
- Big problem in machine learning
- One solution: Validation set

Overfitting

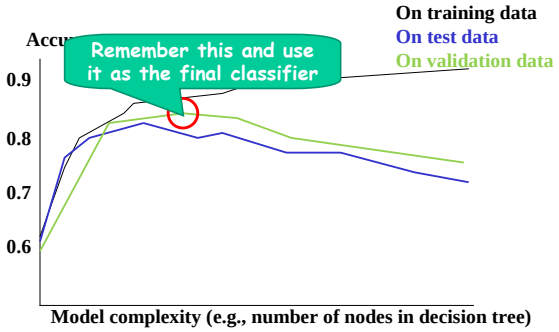


Validation/Tuning Set

- Split data into train and validation set

- Score each model on the tuning set, use it to pick the 'best' model

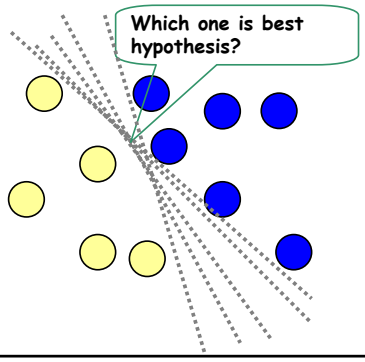
Early Stopping



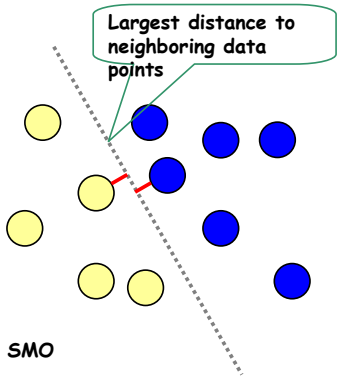
Extra Credit Ideas

- Different types of models
  - Support Vector Machines (SVMs), widely used in web search
  - Tree-augmented naïve Bayes
- Feature construction

Support Vector Machines



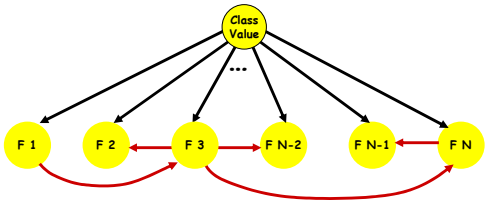
Support Vector Machines



SVMs in Weka: SMO

Tree Augmented Naïve Bayes (TAN)

[Friedman,Geiger & Goldszmidt 1997]



Models limited set of dependencies
Guaranteed to find best structure
Runs in polynomial time

Construct Better Features

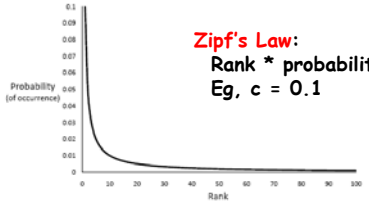
- Key to machine learning is having good features
- In industrial data mining, large effort devoted to constructing appropriate features
- Ideas??

Possible Feature Ideas

- Look at capitalization (may indicated a proper noun)
- Look for commonly occurring sequences
  - E.g. New York, New York City
  - Limit to 2-3 consecutive words
  - Keep all that meet minimum threshold (e.g. occur at least 5 or 10 times in corpus)

Properties of Text

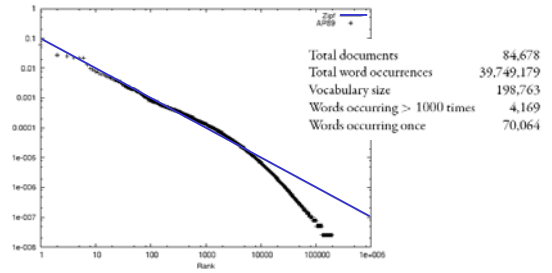
- Word frequencies - skewed distribution
- `The' and `of' account for 10% of all words
- Six most common words account for 40%



Zipf's Law:
Rank * probability = c
Eg, c = 0.1

From [Croft, Metzler & Strohman 2010]

Associate Press Corpus `AP89'



From [Croft, Metzler & Strohman 2010]

Middle Ground

- Very common words → bad features
- Language-based **stop list**:
  words that bear little meaning
  20-500 words
  http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/stop_words
- Subject-dependent stop lists

- Very rare words ***also*** bad features
  Drop words appearing less than k times / corpus

Stop lists

- Language-based stop list:
  words that bear little meaning
  20-500 words
  http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/stop_words
- Subject-dependent stop lists

From Peter Brusilovsky Univ Pittsburg INFSCI 2140 35

Stemming

- Are there different index terms?
  retrieve, retrieving, retrieval, retrieved, retrieves…
- Stemming algorithm:
  (retrieve, retrieving, retrieval, retrieved, retrieves) ⇨ retriev
  Strips prefixes of suffixes (-s, -ed, -ly, -ness)
  Morphological stemming

Stemming Continued

- Can reduce vocabulary by ~ 1/3
- C, Java, Perl versions, python, c#
  www.tartarus.org/~martin/PorterStemmer
- Criterion for removing a suffix
  Does "a document is about w_1" mean the same as a "a document about w_2"
- Problems: sand / sander & wand / wander
- Commercial SEs use giant in-memory tables

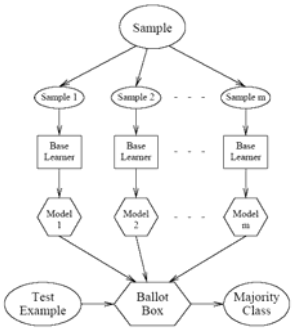
Today's Outline

- Brief supervised learning review
- Evaluation
- Overfitting
- Ensembles
  Learners: The more the merrier
- Co-Training
  (Semi) Supervised learning with few labeled training ex

Ensembles of Classifiers

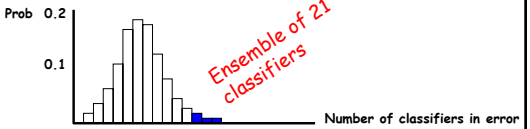
- Traditional approach: Use one classifier
- Alternative approach: Use lots of classifiers
- Approaches:
  - Cross-validated committees
  - Bagging
  - Boosting
  - Stacking

Voting



Ensembles of Classifiers

- Assume
  Errors are independent (suppose 30% error)
  Majority vote
- Probability that majority is wrong…
  = area under binomial distribution

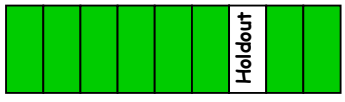


- If individual area is 0.3
- **Area under curve for ≥11 wrong is 0.026**
- Order of magnitude improvement!

Constructing Ensembles

Cross-validated committees

- Partition examples into *k* disjoint equiv classes
- Now create *k* training sets
  Each set is union of all equiv classes ***except one***
  So each set has (k-1)/k of the original training data
- Now train a classifier on each set



Ensemble Construction II

Bagging

- Generate k sets of training examples
- For each set
  - Draw m examples randomly (with replacement)
  - From the original set of m examples
- Each training set corresponds to
  - 63.2% of original (+ duplicates)
- Now train classifier on each set
- Intuition: Sampling helps algorithm become more robust to noise/outliers in the data

Ensemble Creation III

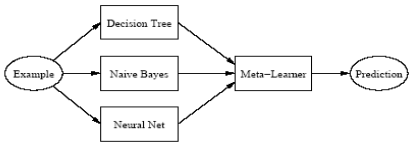
Boosting

- Maintain prob distribution over set of training ex
- Create k sets of training data iteratively:
- On iteration *i*
  - Draw m examples randomly (like bagging)
  - But use probability distribution to bias selection
  - Train classifier number *i* on this training set
  - Test partial ensemble (of *i* classifiers) on all training exs
  - Modify distribution: increase P of each error ex
- Create harder and harder learning problems...
- "Bagging with ***optimized*** choice of examples"

Ensemble Creation IV

Stacking

- Train several base learners
- Next train meta-learner
  - Learns when base learners are right / wrong
  - Now meta learner arbitrates



Train using cross validated committees
- Meta-L inputs = base learner predictions
- Training examples = 'test set' from cross validation

Today's Outline

- Brief supervised learning review
- Evaluation
- Overfitting
- Ensembles
  - Learners: The more the merrier
- Co-Training
  - (Semi) Supervised learning with few labeled training ex

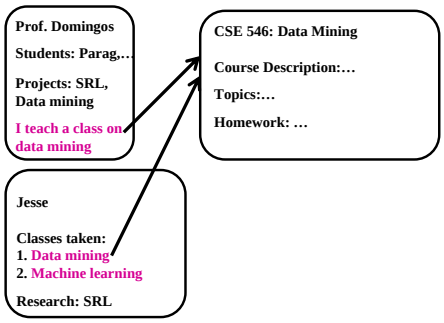
Co-Training Motivation

- Learning methods need labeled data
  - Lots of <x, f(x)> pairs
  - Hard to get... (who wants to label data?)
- But unlabeled data is usually plentiful...
  - Could we use this instead??????
- Semi-supervised learning

Co-training

Suppose
- Have ***little*** labeled data + ***lots*** of unlabeled
- Each instance has two parts:
  - x = [x1, x2]
  - x1, x2 conditionally independent given f(x)
- Each half can be used to classify instance
  - $\exists f1, f2$ such that $f1(x1) \sim f2(x2) \sim f(x)$
- Both f1, f2 are learnable
  - $f1 \in H1$, $f2 \in H2$, $\exists$ learning algorithms A1, A2

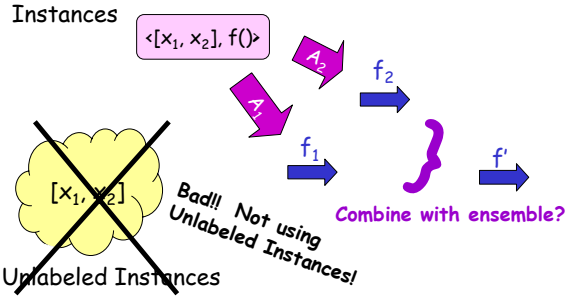
Co-training Example



Without Co-training

$f_1(x_1) \sim f_2(x_2) \sim f(x)$

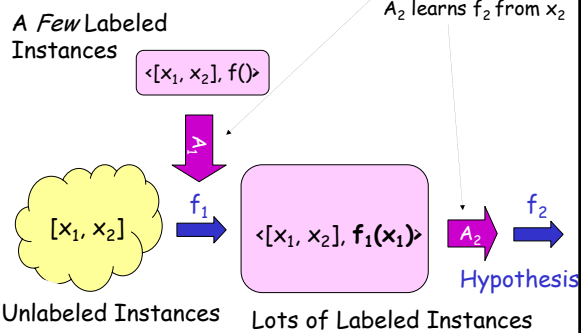
A_1 learns f_1 from x_1
A_2 learns f_2 from x_2



Co-training

$f_1(x_1) \sim f_2(x_2) \sim f(x)$

A_1 learns f_1 from x_1
A_2 learns f_2 from x_2



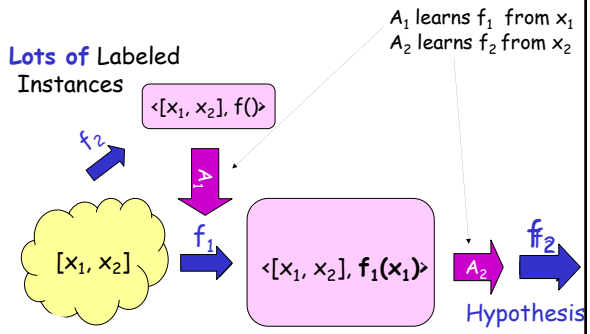
Observations

- Can apply A_1 to generate as much training data as one wants
  If x_1 is conditionally independent of x_2 / $f(x)$, then the error in the labels produced by A_1 ***will look like random noise to A_2 !!!***
- Thus ***no limit*** to quality of the hypothesis A_2 can make

Co-training

$f_1(x_1) \sim f_2(x_2) \sim f(x)$

A_1 learns f_1 from x_1
A_2 learns f_2 from x_2



It really works!

- Learning to classify web pages as course pages
  x1 = bag of words on a page
  x2 = bag of words from all anchors pointing to a page
- Naïve Bayes classifiers
  12 labeled pages
  1039 unlabeled

	Page-based classifier	Hyperlink-based classifier	Combined classifier
Supervised training	12.9	12.4	11.1
Co-training	6.2	11.6	5.0

Table 2: Error rate in percent for classifying web pages as course home pages. The top row shows errors when training on only the labeled examples. Bottom row shows errors when co-training, using both labeled and unlabeled examples.