ENSEMBLE METHODS

Today

- Reading
  - AIMA 18.10-18.11
- Goals
  - Ensembles of classifiers
  - (Supervised learning: putting it all together)

Which classifier should I use?

- Is there a classifier that is optimal for all classification problems?
- Factors to take into account:
  - How much training data is available?
  - How simple/complex is the problem? (linear vs. nonlinear decision boundary)
  - How noisy/skewed is the training data?
  - How stable is the problem over time?
  - Is it a singly-labeled or multi-labeled problem? Are the labels correlated?

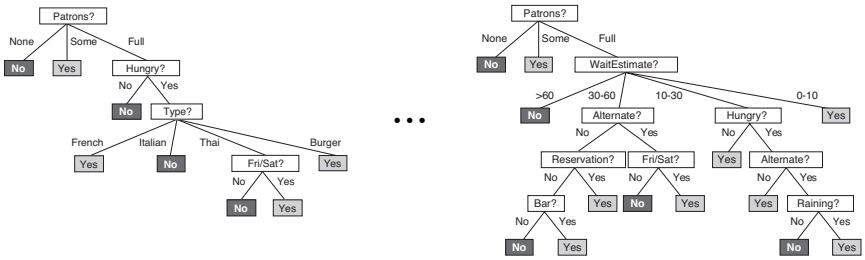
How Much Data?

- Learning theory (PAC learning)
  - Gives theoretical bounds on how much training data you need for a given accuracy (AIMA 18.5)
- Very Little
  - There are empirical results that naïve Bayes should do well in such circumstances (Ng and Jordan 2002 NIPS)
  - The interesting theoretical answer is to explore semi-supervised training methods: Bootstrapping, EM over unlabeled documents, …
  - The practical answer is to get more labeled data as soon as you can
- A reasonable amount of data
  - Start with SVMs
- A lot of data?
  - expensive methods like SVMs (train time) or kNN (test time) are quite impractical
  - Naïve Bayes! - with lots of data, simple methods work well

Ensembles of Classifiers

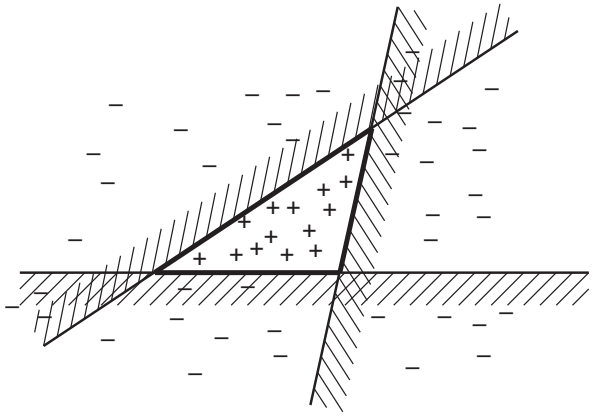
- Ensemble - A group of items viewed as a whole rather than individually
- An ensemble of classifiers – A group of classifiers whose predictions are combined to produce one final prediction
- Benefits
  - Harder to make a wrong prediction
  - More expressive hypothesis

Ensemble of decision trees



- Combine the prediction of each decision tree using majority vote
- Variation of this called a Random Forest

Ensemble of linear classifiers



- More expressive than any one linear classifier by itself

Ensemble Schemes

- Multi-expert combination methods
  - Global - All classifiers generate a prediction and all predictions are used in some way
    - e.g. weighting, voting, averaging
  - Local – A gating model chooses one (or very few) of the classifiers responsible for generating the prediction for a specific input
    - e.g. mixture of experts
- Multi-stage combination
  - Classifiers are trained with, or tested on, only the instances where the previous classifiers are not accurate enough
    - e.g. cascading

Boosting

- Boosting is one of the most common forms of constructing an ensemble of classifiers
  - Learn a series of weak classifiers, i.e. classifiers whose performance is slightly better than random chance
  - Weight each weak classifier to create a final strong classifier
  - Often the weight for each classifier is proportional to its accuracy
- A well-known boosting algorithm is AdaBoost short for "Adaptive Boosting" (Freund and Schapire 1995)

AdaBoost

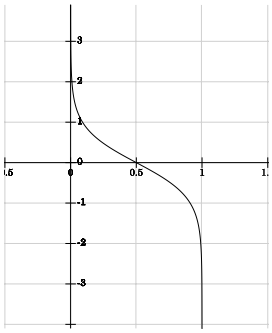
function ADABOOST($examples, L, K$) **returns** a weighted-majority hypothesis
  inputs: $examples$, set of N labeled examples $(x_1, y_1), \ldots, (x_N, y_N)$
    L, a learning algorithm
    K, the number of hypotheses in the ensemble
  local variables: **w**, a vector of N example weights, initially $1/N$
    h, a vector of K hypotheses
    z, a vector of K hypothesis weights

  for $k = 1$ **to** K **do**
    $\mathbf{h}[k] \leftarrow L(examples, \mathbf{w})$
    $error \leftarrow 0$
    for $j = 1$ **to** N **do**
      if $\mathbf{h}[k](x_j) \neq y_j$ **then** $error \leftarrow error + \mathbf{w}[j]$
    for $j = 1$ **to** N **do**
      if $\mathbf{h}[k](x_j) = y_j$ **then** $\mathbf{w}[j] \leftarrow \mathbf{w}[j] \cdot error/(1 - error)$
    $\mathbf{w} \leftarrow$ NORMALIZE(**w**)
    $\mathbf{z}[k] \leftarrow \log(1 - error)/error$
  return WEIGHTED-MAJORITY(**h**, **z**)

AdaBoost

- Generates a sequence of weak classifiers each focusing on the errors of the previous classifier
- AdaBoost returns a strong classifier, i.e. a classifier that can perfectly classify the training data for large enough K
- To classify a new example x:

$$h(x) = \text{sign}\left(\sum_{k=1}^{K} \mathbf{z}[k]\, h_k(x)\right) \quad \text{where} \quad \mathbf{z}[k] = \log\left(\frac{1 - error}{error}\right)$$

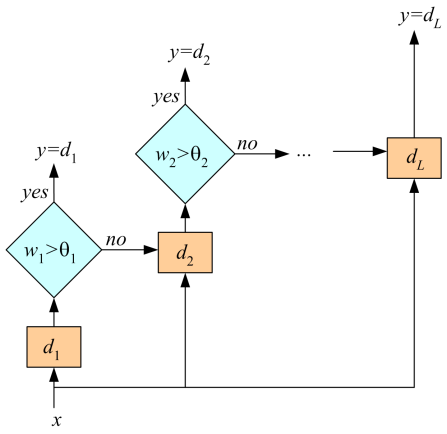


Bagging

- Short for "Bootstrap aggregating"
- Given training set D
  - Generate M new training sets D_i where $|D_i| < |D|$ by sampling from D with replacement
  - This is a statistical technique known as bootstrapping
  - Train a classifier on each of the M new training sets
  - Combine output of M classifiers using averaging or voting
- Random Forests (Breimen, 2001)
  - Bagged decision trees

Cascading classifiers

- Order classifiers by complexity, e.g. representational complexity
- Use i^{th} classifier d_i only if previous classifiers are not confident
- Good with high precision/ low recall classifiers



Ensemble methods

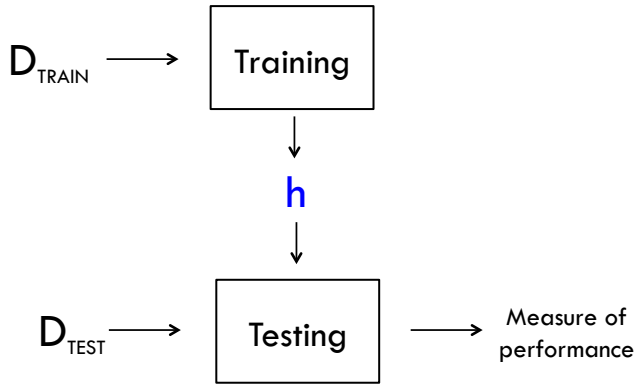
- Boosting
  - Weighted training sets
  - Ex: Adaboost
- Bagging
  - Resampled training sets
  - Ex: Random forests
- Cascading
  - Ordered collection of classifiers

Putting it all together

- Supervised learning: putting it all together
  - Step 1: Formulating the problem
  - Step 2: Exploring the data
  - Step 3: Feature Selection
  - Step 4: Training
  - Step 5: Testing

The first 4 steps are not necessarily done in a strict linear progression

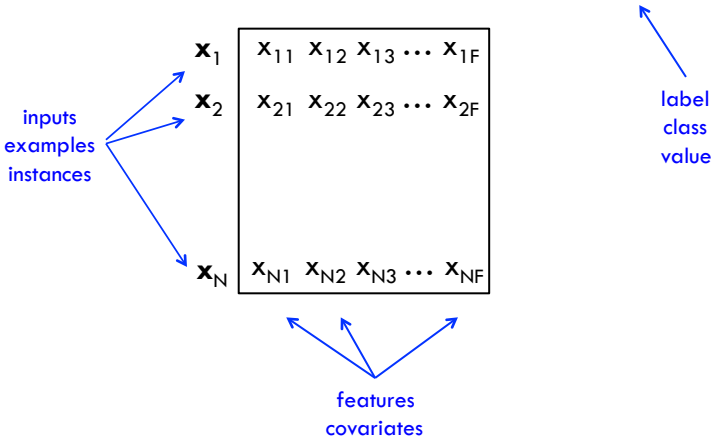
Overview



$$D = \{(\mathbf{x}_i, y_i) \mid i = 1, \ldots, N\} \quad \text{where} \quad f(\mathbf{x}_i) = y_i$$

Overview

$$D = \{(\mathbf{x}_i, y_i) \mid i = 1, \ldots, N\} \quad \text{where} \quad f(\mathbf{x}_i) = y_i$$

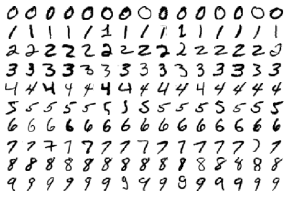


Step 1: Formulate the problem

- What quantity are you predicting?
  - Regression
    - Range? Changing over time?
  - Classification
    - Binary classification? Multi-class classification?
    - Singly-labeled? Multi-labeled?
    - For multi-labeled classification tasks, how correlated are the labels?
- What data do you have?
  - Where to get labeled data? (Amazon mechanical turk)
  - How much labeled data?
  - What is the quality of the labeled data?
  - Are the labels learnable given the data?
  - Is the distribution of labels in the data skewed/imbalanced?

Multi-class Classification

- Generalization of binary classification to more than 2 classes
- One-versus-all
  - Train C independent binary classifiers: one for each label
  - For classifier c
    - Examples with label c are positive examples
    - All other examples are negative examples
  - At prediction time, choose label whose corresponding classifier has highest "confidence"
- One-versus-one
  - Train C(C-1)/2 binary classifiers
  - At prediction time, each classifier votes for a label



NN, NNP, VBZ, DT, RB,…

Multi-class Classification

One-vs-All

x1	c1
x2	c3
x3	c1
x4	c2

original training data

x1	1
x2	-1
x3	1
x4	-1

c1 vs. all

x1	-1
x2	-1
x3	-1
x4	1

c2 vs. all

x1	-1
x2	1
x3	-1
x4	-1

c3 vs. all

One-vs-One

x1	c1
x2	c3
x3	c1
x4	c2

original training data

x1	1
x3	1
x4	-1

c1 vs. c2

x1	1
x2	-1
x3	1

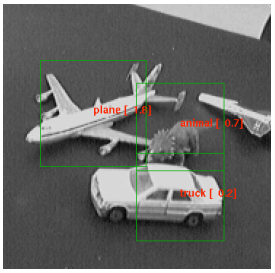
c1 vs. c3

x2	-1
x4	1

c2 vs. c3

Multi-label Classification

- Each example can be labeled with multiple labels
  - Don't confuse this with multi-class classification!
  - Common for document classification or object recognition



- One-vs-all
- One classifier for every possible combination of labels
  - Combinatorial explosion
  - Limited training data

Step 2: Exploratory Data Analysis

- Look at the data. It's surprising how often we forget to actually do this!
- Exploratory Data Analysis (EDA) is a statistical mindset
  - Box plots, histograms, scatter plots, mean, mode, deviations
  - Can guide the modeling process by
    - give you insight into the data
    - help (in)validate your assumptions
    - detect outliers

Step 3: Feature Selection

- What features should I use?
  - Dimensionality reduction if exist time/space constraints
  - Reduce noise in the data (irrelevant or redundant features)
- Dimensionality reduction
  - Principal component analysis (PCA)
  - Singular value decomposition (SVD)
  - Canonical correlation analysis (CCA)
- Regularization
  - Use every feature but penalize classifiers that are overly complex

$$\text{Error}(w) = \sum_{i=1}^{N}(y_i - h_w(x_i)) + \lambda||w||^2$$

encourages sparse weight vectors