SL: PUTTING IT ALL TOGETHER

Today

- Reading
  - AIMA 18.4

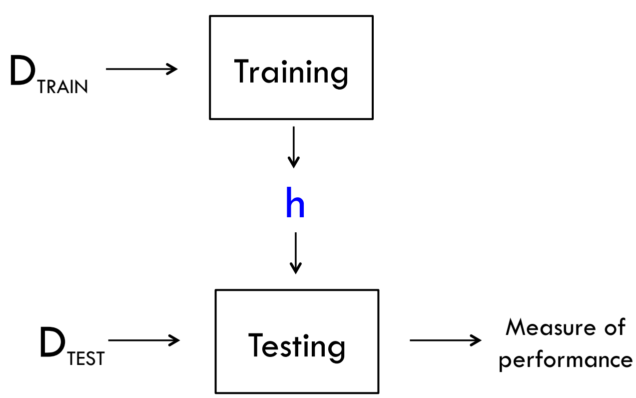
- Goals
  - Step 1: Formulating the problem
  - Step 2: Exploring the data
  - Step 3: Feature Selection
  - Step 4: Training
  - Step 5: Testing

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

Recap: Machine Learning

- The goal of machine learning is to learn from data
  - We might use machine learning to
    - learn the probabilities for a Bayesian network
    - learn the topology of a Bayesian network
- Three types of learning
  - Supervised learning – learning with labels
  - Unsupervised learning – learning without labels
  - Reinforcement learning – learning with rewards

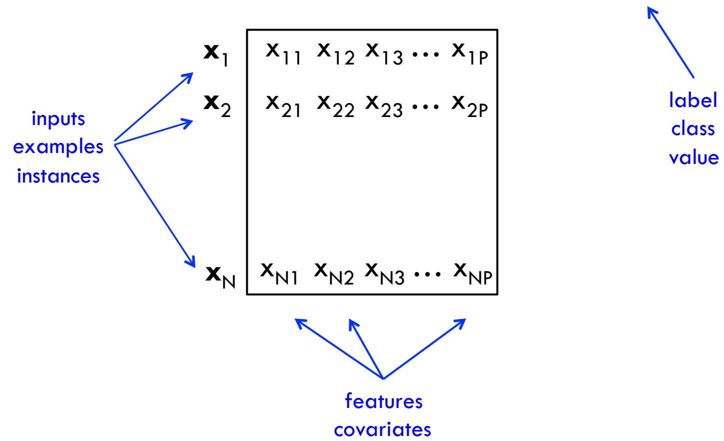
Overview



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

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

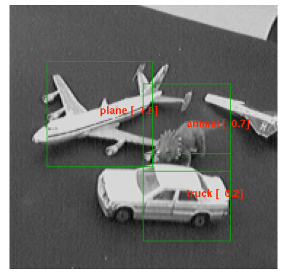
c1 vs. c3

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

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

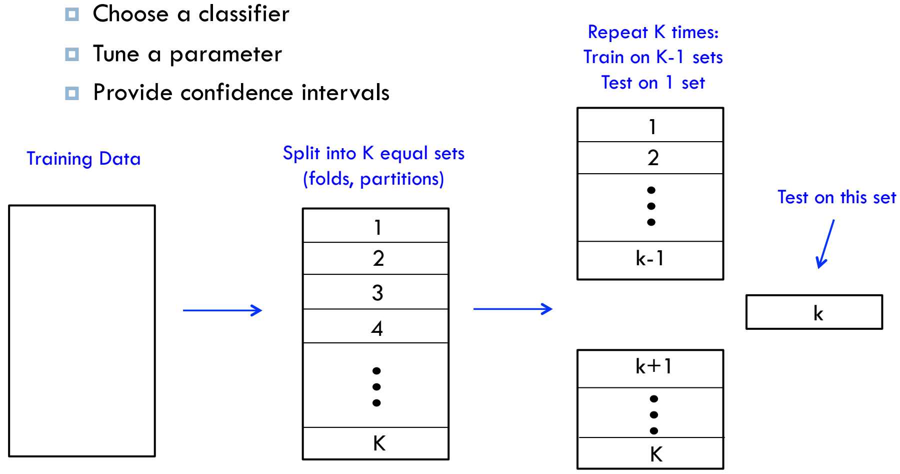
encourages sparse weight vectors

Step 4: Training

- Pick your classifier
  - Decision tree, perceptron, neural network, SVM, linear regression, logistic regression, random forests, ensembles, Gaussian process regression, hidden Markov models, conditional random field, Bayesian networks,…
- Your choice is informed by all of the previous steps
  - Formulating the problem
  - EDA
- Often there are parameters that must be tuned…

Step 4: Training

- K-fold cross validation
  - Choose a classifier
  - Tune a parameter
  - Provide confidence intervals



Step 4: Training

```
function CROSS-VALIDATION-WRAPPER(Learner, k, examples) returns a hypothesis

  local variables: errT, an array, indexed by size, storing training-set error rates
                   errV, an array, indexed by size, storing validation-set error rates
  for size = 1 to ∞ do
      errT[size], errV[size] ← CROSS-VALIDATION(Learner, size, k, examples)
      if errT has converged then do
          best_size ← the value of size with minimum errV[size]
          return Learner(best_size, examples)

function CROSS-VALIDATION(Learner, size, k, examples) returns two values:
          average training set error rate, average validation set error rate

  fold_errT ← 0; fold_errV ← 0
  for fold = 1 to k do
      training_set, validation_set ← PARTITION(examples, fold, k)
      h ← Learner(size, training_set)
      fold_errT ← fold_errT + ERROR-RATE(h, training_set)
      fold_errV ← fold_errV +ERROR-RATE(h, validation_set)
  return fold_errT/k, fold_errV/k
```

Step 5: Testing

- We have a final hypothesis
- We now use our hypothesis to predict on new (unseen) examples from the test set.
  - There's no going back and tweaking the classifier based on its test set performance!
- Where do these new unseen examples come from?
  - External source
  - Set aside from training data

Binary Classification: Measures of Performance

- Let $D_{TEST} = \{ (x_i, y_i) \mid i=1\ldots N\}$ be our test set and $\{ h_i \}$ be the set of predicted values
- The contingency table is given by:

	y = 1	y = 0
h = 1	TP	FP
h = 0	FN	TN

  - TP is the number of *true positives*
  - FP is the number of *false positives*
  - FN is the number of *false negatives*
  - TN is the number of *true negatives*

Binary Classification: Measures of Performance

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{F}_1\text{-score} = 2 \cdot \frac{\text{Prec} \cdot \text{Recall}}{\text{Prec} + \text{Recall}}$$

	y = 1	y = 0
h = 1	TP	FP
h = 0	FN	TN

Contingency Table

Binary Classification: Measures of Performance

$$\text{Accuracy} = \frac{7+8}{7+8+2+3} = \frac{15}{20} = .75$$

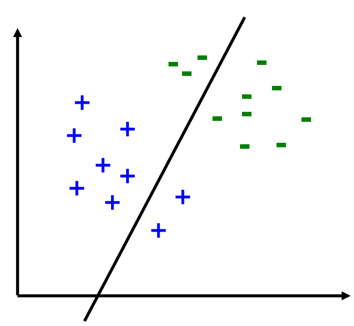
$$\text{Precision} = \frac{7}{7+3} = .70$$

$$\text{Recall} = \frac{7}{7+2} = .78$$

$$\text{F}_1\text{-score} = 2\left(\frac{.70 \cdot .78}{.70 + .78}\right) = 2\left(\frac{.546}{1.48}\right) = .74$$

	y = 1	y = 0
h = 1	7	3
h = 0	2	8

Contingency Table



Multi-class Classification: Measures of performance

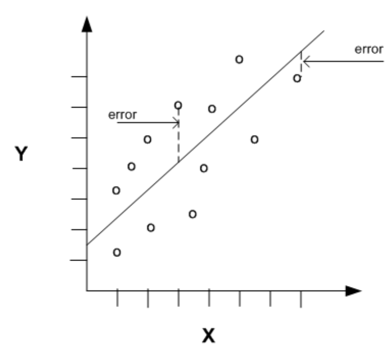
- Evaluate each label separately using a "one-vs-all" approach
  - Macro-averaging
    - Compute the measure (precision, recall, F_1) for each class
    - Average across all C classes
    - Gives equal weight to all classes
  - Micro-averaging
    - Pool the TP, FP, FN, TN for all C classes
    - Compute the measure (precision, recall, F1)
    - Weighted towards performance of most likely class

	$y_c = 1$	$y_c = 0$
$h_c = 1$	TP_c	FP_c
$h_c = 0$	FN_c	TN_c

Contingency Table

Regression: Measures of performance

- Mean-squared error
- Root mean-squared error
- Mean absolute error
- Mean absolute percentage
- …



Summary

- Overview
  - Step 1: Formulate the problem
  - Step 2: Explore the data
  - Step 3: Feature Selection
  - Step 4: Training
  - Step 5: Testing

- Lessons
  - Choose supervised over unsupervised learning
  - Reproducibility
  - Think of how you would justify each decision you made
  - Start simple and iterate