# Machine Learning Methodology

Dan Sheldon

November 12, 2012

# Overview

Some different reasons for ML experiments

- Evaluation
  - How accurate is your classifier?
- Model selection
  - E.g., find best $C$, $\gamma$ for Gaussian-kernel SVM
- Algorithm comparison
  - SVMs are better than logistic regression for hand-written digit classification.

# Evaluation

Simplest question: how accurate is my classifier?

Nuances:

- Accuracy on training set?
- Accuracy on test set?

Real goal: estimate accuracy on *unseen* future data.

- Test set is a proxy for this, sometimes imperfect

# Data-generating mechanism

- Assumption: training data is representative of test data
- Distinction
  - Test data = unseen future data
  - Test set = subset of training data, proxy for test data
- Formally, training examples and test examples drawn *independently* from same probability distribution $\mathcal{P}$

$$(\mathbf{x}_i, y_i) \sim \mathcal{P}$$
$$(\mathbf{x}, y) \sim \mathcal{P}$$

# Data-generating mechanism

- How to think of this
  - Huge bag of input-output pairs $(\mathbf{x}, y)$
  - $N$ training examples drawn randomly from bag
  - Future data drawn randomly from same bag
- Example: hand-written digits

# Data-generating mechanism

Example: hand-written digits

Discussion: can you describe a scenario where this assumption is violated?

# Dilemma: Size of Train vs. Test Sets

Back to question: how to evaluate accuracy of a classifier?

Need to split $N$ examples into train/test sets
(Examples)

Dilemma

- More training data $\rightarrow$ more accurate classifier
- More test data $\rightarrow$ better estimate of accuracy on future data

What should we do?

# Cross-Validation

Split data into $k$ equally sized *folds*

For $i = 1$ to $k$

- Test on fold $i$
- Train on all other folds

Average performance over the $k$ trials

**Example**

| | Train folds | Test folds | Accuracy |
|---|---|---|---|
| 12345 | 2,3,4,5 | 1 | 85% |
| 12345 | 1,3,4,5 | 2 | 83% |
| 12345 | 1,2,4,5 | 3 | 91% |
| 12345 | 1,2,3,5 | 4 | 88% |
| 12345 | 1,2,3,4 | 5 | 84% |

Average accuracy = 88.2%

# Cross-Validation

Practical tips

- Use cross-validation
- Choose $k \in \{10, 5, 2\}$ depending on how much running time you can afford
- For deployment, retrain model on entire labeled set

# Don't cheat

What is wrong with this example?

**Example**

Run two cross-validation experiments

- Model 1: SVM with $C = 1 \rightarrow 93\%$ accuracy
- Model 2: SVM with $C = 10 \rightarrow 94\%$ accuracy

Therefore, by selecting Model 2, I can predict with 94% accuracy on future data!

Selecting the *best* model on test data $\rightarrow$ optimistic estimate of performance.

# Model Selection

Example: select $(C, \gamma)$ for Gaussian SVM

- Approach: run grid search on parameters

| | $\gamma = 10^{-2}$ | $\gamma = 10^{-1}$ | $\gamma = 10^{0}$ |
|---|---|---|---|
| $C = 10^{-1}$ | | | |
| $C = 10^{0}$ | | | |
| $C = 10^{1}$ | | | |

- Select combination with best performance on test set
- Use simple train/test split
- Or cross-validation

# Validation Set

To do model selection *and* estimate future performance, need separate test data.

- Validation set—tune parameters (model selection)
- Test set—estimate performance

Various setups

- Simple train/validation/test split
- Nested cross-validation

# Performance Measures

Accuracy is not always the best performance measure.

Other choices

- Precision
- Recall

# Confusion Matrix

Many measures based on *confusion* matrix

| | | Predicted label: 1 | Predicted label: 0 |
|---|---|---|---|
| Actual label | 1 | TP | FN |
| | 0 | FP | TN |

- TP: true positive
- FP: false positive
- TN: true negative
- FN: false negative

# Performance Measures

| | | Predicted label: 1 | Predicted label: 0 |
|---|---|---|---|
| Actual label | 1 | TP | FN |
| | 0 | FP | TN |

$$\text{Accuracy} = \frac{\text{TP}+\text{TN}}{\text{total}}$$ What percent do I get right?

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$ Of predicted positives, what fraction are correct?

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$ Of actual positives, what fraction do I detect?

# Example: Spam classification

| | | Predicted | |
|---|---|---|---|
| | | spam | non-spam |
| | spam | 22 | 6 |
| | non-spam | 4 | 68 |

# Precision-Recall Curves

- In practice, may want to favor precision over recall, or vice versa
- E.g., spam classification: precision is important (don't want to miss real emails)
- Most classifiers have a threshold you can tune to trade off precision vs. recall
- Test all possible thresholds $\rightarrow$ precision-recall curve

Example on board