Methods and Criteria for Model Selection I

CS57300 Data Mining
Fall 2016

Instructor: Bruno Ribeiro
Goal

- Predictive Model Assessment
- Model Selection
In Training Phase There is Often a Better Model
Testing Classification Accuracy
**Classification Error**

- **True positive (TP):** positive prediction that is correct
- **True negative (TN):** negative prediction that is correct
- **False positive (FP):** positive prediction that is incorrect
- **False negative (FN):** negative prediction that is incorrect

- **Accuracy** = \( \frac{TP + TN}{TP + TN + FP + FN} \) % predictions that are correct
- **Misclassification** = \( \frac{FP + FN}{TP + TN + FP + FN} \) % predictions that are incorrect
- **Recall (Sensitivity)** = \( \frac{TP}{TP + FN} \) % positive instances that are predicted positive
- **Precision** = \( \frac{TP}{TP + FP} \) % positive predictions that are correct
- **Specificity** = \( \frac{TN}{TN + FP} \) % negative instances that are predicted negative
- **F1** = \( \frac{2 \cdot P \cdot R}{P + R} \) harmonic mean of precision and recall
Precision and Recall

Precision: \( \frac{a}{a + b} \)

Recall: \( \frac{a}{a + c} \)

(a) Precision and recall

(b) Precision = 1

(c) Recall = 1
More score functions

- Absolute loss: \( \frac{1}{n} \sum_{i=1}^{n} |p(y_i = t_i) - 1.0| \) where \( t \) is true label

- Squared loss: \( \frac{1}{n} \sum_{i=1}^{n} [p(y_i = t_i) - 1.0]^2 \) where \( t \) is true label

- Likelihood/conditional likelihood: \( \prod_{i=1}^{n} p(y_i = t_i) \) where \( t \) is true label

Area under the ROC curve

- Receiver Operating Characteristic (ROC) curve

- For classifier that a classification threshold can be defined (e.g. SVMs [vary hyperplane offset b],...)

- Plots the true positive rate against the false positive rate for different classification thresholds
ROC curves

- Evaluates performance over varying costs and class distributions
  - Can summarize with area under the curve (AUC)
  - AUC of 0.5 is random
  - AUC of 1.0 is perfect
# How to compute ROC curve

<table>
<thead>
<tr>
<th>$P(Y)$</th>
<th>True class</th>
<th>Predict class</th>
<th>$P(Y)$</th>
<th>True class</th>
<th>Predict class</th>
<th>$P(Y)$</th>
<th>True class</th>
<th>Predict class</th>
<th>$P(Y)$</th>
<th>True class</th>
<th>Predict class</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.94</td>
<td>+</td>
<td>+</td>
<td>0.94</td>
<td>+</td>
<td>+</td>
<td>0.94</td>
<td>+</td>
<td>+</td>
<td>0.94</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>0.84</td>
<td>-</td>
<td></td>
<td>0.84</td>
<td>-</td>
<td></td>
<td>0.84</td>
<td>-</td>
<td></td>
<td>0.84</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>0.67</td>
<td>+</td>
<td></td>
<td>0.67</td>
<td>+</td>
<td></td>
<td>0.67</td>
<td>+</td>
<td></td>
<td>0.67</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>0.58</td>
<td>+</td>
<td></td>
<td>0.58</td>
<td>-</td>
<td></td>
<td>0.58</td>
<td>-</td>
<td></td>
<td>0.58</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>0.51</td>
<td>+</td>
<td></td>
<td>0.51</td>
<td>+</td>
<td></td>
<td>0.51</td>
<td>+</td>
<td></td>
<td>0.51</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>0.42</td>
<td>+</td>
<td></td>
<td>0.42</td>
<td>+</td>
<td></td>
<td>0.42</td>
<td>+</td>
<td></td>
<td>0.42</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>0.16</td>
<td>-</td>
<td></td>
<td>0.16</td>
<td>-</td>
<td></td>
<td>0.16</td>
<td>-</td>
<td></td>
<td>0.16</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>-</td>
<td></td>
<td>0.1</td>
<td>-</td>
<td></td>
<td>0.1</td>
<td>-</td>
<td></td>
<td>0.1</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>0.07</td>
<td>-</td>
<td></td>
<td>0.07</td>
<td>-</td>
<td></td>
<td>0.07</td>
<td>-</td>
<td></td>
<td>0.07</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

FPR = 0/5  TPR = 1/4  FPR = 1/5  TPR = 1/4  FPR = 1/5  TPR = 2/4
ROC curve
Issues with conventional score functions

- Assumes errors for all individuals are equally weighted
  - May want to weight recent instances more heavily
  - May want to include information about reliability of sets of measurements

- Assumes errors for all contexts are equally weighted
  - May want to weigh false positive and false negatives differently
  - May be costs associated with acting on particular classifications (e.g., marketing to individuals)
Examples

- Loan decisions
  - Cost of lending to a defaulter is much greater than the lost business of refusing loan to a non-defaulter

- Oil-slick detection
  - Cost of failing to detect an environmental disaster is far less than the cost of a false alarm

- Promotional mailing
  - Cost of sending junk mail to a household that doesn't respond is far less that the lost business of not sending it to a household that would have responded
Cost-sensitive models

- Define a score function based on a cost matrix
- If $\sim y$ is the predicted class and $y$ is the true class, then need to define a matrix of costs $C(\sim y,y)$
- Reflects the severity of classifying an instance with true class $y$ to class $\sim y$
Predictive modeling: summary

- Classification is an extensively studied problem
- Probably the most widely used data mining technique
  - Clearly defined task
  - Well-defined evaluation functions to guide search
  - Framework to understand model performance
- Other issues:
  - Scalability, understandability, non-standard data representations
Assessing Error with Training Data
The Problem of Overfitting

![Graph showing the behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error, while the light red curves show the conditional test error. The solid curves show the expected test error and the expected training error. High Bias and Low Bias are indicated, as well as Low Variance and High Variance.](image)

Test error, also referred to as generalization error, is the prediction error over an independent test sample:

\[ \text{Err}_T = E \left[ L(Y, \hat{f}(X)) \right] | T \]  

(7.2)

where both \( X \) and \( Y \) are drawn randomly from their joint distribution (population). Here the training set \( T \) is fixed, and test error refers to the error for this specific training set. A related quantity is the expected prediction error (or expected test error):

\[ \text{Err} = E \left[ L(Y, \hat{f}(X)) \right] = E[\text{Err}_T] \]  

(7.3)

Note that this expectation averages over everything that is random, including the randomness in the training set that produced \( \hat{f} \).

Figure 7.1 shows the prediction error (light red curves) \( \text{Err}_T \) for 100 simulated training sets each of size 50. The lasso (Section 3.4.2) was used to produce the sequence of fits. The solid red curve is the average, and hence an estimate of \( \text{Err} \).

Estimation of \( \text{Err}_T \) will be our goal, although we will see that \( \text{Err} \) is more amenable to statistical analysis, and most methods effectively estimate the expected error. It does not seem possible to estimate conditional training error.
Linear Regression

\[ y = ax + b \]
Cubic Regression

A simple, a complex and a trade-off (3rd degree) polynomial.

This intuition is confirmed by numerous experiments on real-world data from a broad variety of sources [Rissanen 1989; Vapnik 1998; Ripley 1996]: if one naively fits a high-degree polynomial to a small sample (set of data points), the model does not obtain a very good fit to the data. Yet if one tests the inferred polynomial on a second set of data coming from the same source, it typically fits this test data very badly in the sense that there is a large distance between the polynomial and the new data points. We say that the polynomial overfits the data. Indeed, all model selection methods that are used in practice either implicitly or explicitly choose a trade-off between goodness-of-fit and complexity of the models involved. In practice, such trade-off slightly outperforms better predictions of test data than one would get by adopting the 'simplest' (one degree) or most 'complex' ($3^{rd}$-degree) polynomial. MDL provides one particular means of achieving such a trade-off.

It will be useful to make a precise distinction between 'model' and 'hypothesis':

**Models vs. Hypotheses**

We use the phrase point hypothesis to refer to a single probability distribution or function. An example is the polynomial $5x^2 + 4x + 3$. A point hypothesis is also known as a 'simple hypothesis' in the statistical literature.

We use the word model to refer to a family (set) of probability distributions or functions with the same functional form. An example is the set of false second-degree polynomials. A model is also known as a 'composite hypothesis' in the statistical literature.

We use hypothesis as a generic term, referring to both point hypotheses and models.

In our terminology, the problem described in Example 1.2 is a 'hypothesis selection problem' if we are interested in selecting both the degree of a polynomial and the corresponding coefficients.

Better predictions of test data than one would get by adopting the 'simplest' or most 'complex' polynomial. MDL provides one particular means of achieving such a trade-off.

$$y = ax^3 + bx^2 + cx + d$$
Degree $n-1$ Polynomial

$y = ax^{n-1} + bx^{n-2} + \ldots$
Model Selection and Assessment

- **Model Selection**: Estimating performances of different models to select the best one
- **Model Assessment**: Having chosen a model, estimating the prediction error on new data

© 2015 Bruno Ribeiro
Model Selection and Assessment

- In data-rich scenarios split the data:

  - Train
  - Validation
  - Test

- In data-poor scenarios: approximate validation step
  - analytically
    - AIC, BIC, MDL
  - via sample re-use
    - cross-validation
      - Leave-one-out
      - K-fold
    - bootstrap

© 2015 Bruno Ribeiro
Model vs Hypotheses

- Simple hypothesis: single probability distributions (or functions)
  - Model with specific parameters

- Composite hypothesis: family of probability distributions (or functions)
  - A Model
Decomposing Test Error

True target Predicted target

Model: \( t = f(x_0; \theta^*) + \epsilon; \)

\[ E[\epsilon] = 0 \quad \text{Var}(\epsilon) = \sigma_\epsilon^2 \]

For squared-error loss with additive noise:

\[
\text{Err}(x_0) = E[(t - f(x_0; \hat{\theta}))^2 | X = x_0] \\
= \sigma_\epsilon^2 + \left( E f(x_0; \hat{\theta}) - f(x_0; \theta^*) \right)^2 + E \left[ f(x_0; \hat{\theta}) - E f(x_0; \hat{\theta}) \right]^2 \\
= \sigma_\epsilon^2 + \text{Bias}^2(f(x_0; \hat{\theta})) + \text{Var}(f(x_0; \hat{\theta}))
\]
Findings

- **Bias**
  - Often related to size of model space
  - More complex models tend to have lower bias

- **Variance**
  - Often related to size of dataset and quality of information
  - When data is large enough to estimate true parameters ($\theta^*$) well we get lower variance

- With big data simple models can perform surprisingly well due to lower variance

\[
\text{Err}(x_0) = E[(t - f(x_0; \hat{\theta}))^2 | X = x_0]
\]

\[
= \sigma_e^2 + \left( E f(x_0; \hat{\theta}) - f(x_0; \theta^*) \right)^2 + E \left[ f(x_0; \hat{\theta}) - E f(x_0; \hat{\theta}) \right]^2
\]

\[
= \sigma_e^2 + \text{Bias}^2(f(x_0; \hat{\theta})) + \text{Var}(f(x_0; \hat{\theta}))
\]
Optimism of the Training Error Rate

- Typically: training error rate < true error
  (same data is being used to fit the method and assess its error)

\[
Err_{\text{train}} = \frac{1}{N} \sum_{i=1}^{N} L(t(i), f(x_0(i); \hat{\theta})) < Err(X) = E[L(t, f(X; \hat{\theta}))]
\]

overly optimistic
Estimating Test Error

- Can we estimate the discrepancy between $\text{Err}(x_0)$ and $\text{Err}(X)$?
- $D$ – old observations; $D^{\text{new}}$ – new observations
- Suppose we observed new values of label $t(i)$ for same input $x(i)$

$$E_{D_{\text{in}}} = \frac{1}{N} \sum_{i=1}^{N} E_{D} E_{D^{\text{new}}} L(t^{\text{new}}(i), f(x_0(i); \hat{\theta}))$$

$$= E_{D}[\text{Err}_{\text{train}}] + \frac{2}{N} \sum_{i=1}^{N} \text{Cov}(\hat{t}_i, t_i)$$

Adjustment for optimism of training error

Expectation over N new responses at each $x_i$

© 2015 Bruno Ribeiro
Measuring Optimism

If model: \( t = f(X; \theta) + \epsilon; \) \( E[\epsilon] = 0 \) \( \text{Var}(\epsilon) = \sigma^2 \epsilon \)
with \(|\theta| = d\)

Then, if \( f \) is linear, \( f(x; \theta) = x^T \theta \) then

\[
E_D[Err_{in}] = E_D[Err_{train}] + \frac{2}{N} d \sigma^2 \epsilon
\]

- Optimism grows linearly with model dimension
- Optimism decreases as training sample size increases

© 2015 Bruno Ribeiro
Ways to Estimate Prediction Error

- In-sample error estimates:
  - AIC
  - BIC
  - MDL

- Extra-sample error estimates:
  - Cross-Validation
    - Leave-one-out
    - K-fold
  - Bootstrap

© 2015 Bruno Ribeiro
Estimates of In-Sample Prediction Error

- General form of the in-sample estimate:
  \[ Err_{in} = Err_{train} + \text{optimism error} \]

- For additive error:
  \[ C_p = Err_{train} + \frac{2d}{N} \hat{\sigma}_\epsilon^2 \]
  Known as the Marllow’s $C_p$ statistic
**AIC & BIC**

**Akaike Information Criterion (AIC)**

\[
\text{AIC} = -\frac{2}{N} \ln P[t = f(x_0; \hat{\theta})] + 2 \frac{d}{N}
\]

AIC does not assume model is correct  
Better for small # samples  
May not converge to "correct model"  
(as \(N \to \infty\) model bias \(\geq 0\))

**Bayesian Information Criterion (BIC)**

\[
\text{BIC} = -\frac{2}{N} \ln P[t = f(x_0; \hat{\theta})] + \frac{d}{N} \log N
\]

BIC assumes model is correct  
Converges to correct parameterization (\(\theta^*\)) of model  
(as \(N \to \infty\) model bias = 0)
MDL (Minimum Description Length)

- Find hypothesis (model) that minimizes
  - \( H(M) + H(D|M) \), where
    - \( H(M) \) – length in bits of description of model \( M \)
    - \( H(D|M) \) – length in bits of description of data encoded by model \( M \)

- If model probabilistic:
  \[
  \text{length} = - \ln P[t = f(x_0; \hat{\theta})|M] - \ln P[\hat{\theta}|M]
  \]

  Length of transmitting the discrepancy given the model + optimal coding under the given model

  Description of the model under optimal coding

**MDL principle:** choose the model with the minimum description length

**Equivalent to maximizing the posterior:**

\[
P[t = f(x_0; \hat{\theta})|M] P[\hat{\theta}|M]
\]

© 2015 Bruno Ribeiro
Estimation of Extra-Sample Err

- Cross Validation
- Bootstrap
Cross-Validation

For the $k$th part (third above), we fit the model to the other $K - 1$ parts of the data, and calculate the prediction error of the fitted model when predicting the $k$th part of the data. We do this for $k = 1, 2, \ldots, K$ and combine the $K$ estimates of prediction error.

Here are more details. Let $\kappa: \{1, \ldots, N\} \rightarrow \{1, \ldots, K\}$ be an indexing function that indicates the partition to which observation $i$ is allocated by the randomization. Denote by $\hat{f}^{(-k)}(x)$ the fitted function, computed with the $k$th part of the data removed. Then the cross-validation estimate of prediction error is

$$CV(\hat{f}) = \frac{1}{N} \sum_{i=1}^{N} L(y(i), f(x^{(-K)}(i); \hat{\theta}))$$

Typical choices of $K$ are 5 or 10 (see below). The case $K = N$ is known as leave-one-out cross-validation. In this case $\kappa(i) = i$, and for the $i$th observation the fit is computed using all the data except the $i$th.

Given a set of models $f(x, \alpha)$ indexed by a tuning parameter $\alpha$, denote by $\hat{f}^{(-k)}(x, \alpha)$ the $\alpha$th model fit with the $k$th part of the data removed. Then for this set of models we define

$$CV(\hat{f}, \alpha) = \frac{1}{N} \sum_{i=1}^{N} L(y(i), f(x^{(-K)}(i); \alpha))$$

The function $CV(\hat{f}, \alpha)$ provides an estimate of the test error curve, and we find the tuning parameter $\hat{\alpha}$ that minimizes it. Our final chosen model is $f(x, \hat{\alpha})$, which we then fit to all the data.

It is interesting to wonder about what quantity $K$-fold cross-validation estimates. With $K = 5$ or 10, we might guess that it estimates the expected error $Err$, since the training sets in each fold are quite different from the original training set. On the other hand, if $K = N$ we might guess that cross-validation estimates the conditional error $Err_T$. It turns out that cross-validation only estimates effectively the average error $Err$, as discussed in Section 7.12.

What value should we choose for $K$? With $K = N$, the cross-validation estimator is approximately unbiased for the true (expected) prediction error, but can have high variance because the $N$ “training sets” are so similar to one another. The computational burden is also considerable, requiring $N$ applications of the learning method. In certain special problems, this computation can be done quickly—see Exercises 7.3 and 5.13.
How many folds?

- Variance decreases
- Bias decreases
- Computation increases

k fold → Leave-one-out

k increases
Cross-Validation: Choosing K

Popular choices for K: 5, 10, N=leave one out
Evaluating classification algorithms A and B

- Use k-fold cross-validation to get k estimates of error for algorithms A and B

Set of errors estimated over the test set folds provides empirical estimate of sampling distribution

- Mean is estimate of expected error

- Is the mean error significant between A and B?
Assessing significance

- Use **paired t-test** to assess whether the two distributions of errors are statistically different from each other.

  - Error A.1  Error B.1
  - Error A.2  Error B.2
  - Error A.3  Error B.3
  - Error A.4  Error B.4
  - Error A.5  Error B.5

- Takes into account both the difference in means and the variability of the scores.
Bootstrapping
Bootstrapping

- Bootstrap is a powerful statistical tool to quantify the uncertainty associated with a given estimator or statistical learning method.

- E.g.: It can provide an estimate of the standard error of a coefficient, or a confidence interval for that coefficient.
Why Bootstrapping?

- The term *bootstrap* derives from “*to pull oneself up by one’s bootstraps*”, likely based on one of the eighteenth century novel “The Surprising Adventures of Baron Munchausen” by Rudolph Erich Raspe:
  - The Baron had fallen to the bottom of a deep lake. Just when it looked like all was lost, he thought to pick himself up by his own bootstraps.

- It is not the same as the term “bootstrap” used in computer science meaning to “boot” a computer from a set of core instructions, though the derivation is similar.
Model Assessment and Selection

Bootstrap

replications

samples

Training

Z = (z_1, z_2, \ldots, z_N)

FIGURE 7.12.

Schematic of the bootstrap process. We wish to assess the statistical accuracy of a quantity $S(Z)$ computed from our dataset.

$Z^*_{\text{1}}$  $Z^*_{\text{2}}$  $Z^*_{\text{B}}$

$S(Z^*_{\text{1}})$  $S(Z^*_{\text{2}})$  $S(Z^*_{\text{B}})$

Step 1: Draw $N$ samples with replacement from training data

Step 2: Learn model $S(.)$ over Bootstrap samples

Step 3: Compute variance of estimated parameters & error over validation data

Bootstrap replications

Bootstrap samples

Estimating Prediction Error with Bootstrap

One approach would be to fit the model in question on a set of bootstrap samples, and then keep track of how well it predicts the original training set. If $\hat{f}^*_{\text{b}}(x_i)$ is the predicted value at $x_i$ from the model fitted to the $b$th bootstrap dataset, our estimate is

$$\hat{\text{Err}}_{\text{boot}} = \frac{1}{B} \sum_{b=1}^{B} \sum_{i=1}^{N} L(y_i, \hat{f}^*_{\text{b}}(x_i)).$$

However, it is easy to see that $\hat{\text{Err}}_{\text{boot}}$ does not provide a good estimate in general. The reason is that the bootstrap datasets are acting as the training samples, while the original training set is acting as the test sample, and these two samples have observations in common. This overlap can make overfit predictions look unrealistically good, and is the reason that cross-validation explicitly uses non-overlapping data for the training and test samples.
Wrong Way to Do Cross-validation

- Use cross-validation to estimate anything in the model or for feature selection
- Use cross-validation to hunt for new models

The human regression machine:

© 2015 Bruno Ribeiro