Architecture of Data Mining for Business Intelligence

This lecture
Defining a Mining Structure

Data Source

Mining Structure

- cereal name
- mtr
- calories
- fiber
- rating

Model J
filter by calories < 100

Model K
filter by rating > 50

Data Mining Engine

Pattern Analysis

Learning Algorithms

Knowledge Base

Choose the best model from evaluating them

Choose the best model from evaluating them

Testing Dataset

Whole Dataset

Training Dataset

Validation Dataset

Choose the best model from evaluating them
Why Evaluate?

• Multiple classifiers and predictive methods are available to classify or predict
• For each method, multiple choices are available for settings
• To choose best model, need to assess each model’s performance
Training Dataset, Validation Dataset, Test Dataset

• **Training dataset is used to select the parameters of the learning model**
  - The trained parameters give the minimize error on training dataset

• **Validation dataset, if there is one, is used to minimize overfitting**
  - Performance improvement over the training dataset should yield an increase in accuracy over a separate validation dataset to learning model
  - Otherwise stop the training
  - Optimized parameters are tuned from

• **Testing dataset is used to test the final solution in order to evaluate how good the prediction model is**
Cross Validation

• A scheme of evaluating and comparing learning algorithms by dividing data into two separate groups:
  ➢ One is used to learn or train a model and
  ➢ The other is used to validate the model
• Typically, the training and validation sets must be cross-over in successive rounds
  ➢ Each data point is given a chance to be validated against
• Basic form of cross-validation
  ➢ k-fold cross-validation
Cross Validation

Two possible goals in cross-validation:

- To estimate performance of the learned model from available data against a typical algorithm
- To compare the performance of two or more different learning algorithms with parameters tuned
  - find out the best algorithm for the available data with optimized parameters
Classification Measurements

• Classification Accuracy (%) – two types:
  ➢ Cross validation
  ➢ Testing accuracy

• True classified instance
  ➢ Classifying an instance to one class, it turns out it belongs to

• Accuracy

\[
accuracy = \frac{T}{N} \times 100
\]

T: the number of true classified instances
N: total number of instances in the dataset
Types of Cross Validation

• Resubstitution Validation
  - The model is trained from all available data and tested on the same data

• Hold-Out Validation
  - Separate the dataset into training and testing
  - Learn or train a model from training dataset
  - Estimate performance of the learned model in the testing dataset

• K-Fold Cross-Validation
  - Separate the dataset into $K$ subsets, where $K < N$ (the total number of data points in the dataset)
  - Repeat Hold-Out $K$ times, each time take one of the $k$ subsets as the test set and the other $K-1$ subsets combined as training set together
Types of Cross-Validation

• Leave-One-Out Cross-Validation
  ➢ Extreme version of K-Fold
  ➢ Repeat Hold-Out $N$ times, each time the training model is trained on all the data except for one data point and a prediction is made for that point

• Repeated K-Fold Cross-Validation
  ➢ Perform K-fold cross-validation multiple times
    o e.g. run 10-fold CV 100 times
  ➢ The data is reshuffled and re-stratified before each round to give a large number of estimates out of $K$ subsets

• In all types of validation schemes, the mean error is computed and used to estimate
Performance Metrics

• Prediction tasks
  ➢ Average error
    o Gives an idea of systematic over- or under-prediction
  ➢ Mean absolute percentage error (MAPE)
  ➢ RMSE (root-mean-squared-error)
    o Square the errors, find their average, take the square root
  ➢ Total SSE: total sum of squared error

• Classification tasks
  ➢ Classification matrix
  ➢ ROC Curve
  ➢ Count for misclassification costs
    o Total sum of squared error
Example of MES

MES (Mean Squared Error)

\[ \text{MSE} = \frac{(y_{1p} - y_{1o})^2 + (y_{2p} - y_{2o})^2 + (y_{3p} - y_{3o})^2}{3} \]

Linear learning model built from training set

Linear function example

• Problem: how to tell how close the estimates are
Predicting Performance

• Problem

  ➢ Assume the estimated success rate 75%, how close is this to the true success rate \( p \) ?

• Confidence intervals (Confidence limits)

  ➢ \( p \) lies within a certain specified interval with a certain specified confidence

  ➢ Example 1: \( S=750 \) success in \( N=1000 \) trials
    
    o Estimated success rate: \( f = 75\% \)
    
    o With 80% confidence true success rate \( p \) in \([73.2, 76.7]\)

  ➢ Example 2: \( S=75 \), and \( N=100 \)

    o Estimated success rate : \( f = 75\% \)
    
    o With 80% confidence \( p \) in \([69.1, 80.1]\)
Confidence Limits

- Confidence limits for the normal distribution with 0 mean and a variance of 1:

<table>
<thead>
<tr>
<th>Pr[(X \geq z])</th>
<th>(z)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1%</td>
<td>3.09</td>
</tr>
<tr>
<td>0.5%</td>
<td>2.58</td>
</tr>
<tr>
<td>1%</td>
<td>2.33</td>
</tr>
<tr>
<td>5%</td>
<td>1.65</td>
</tr>
<tr>
<td>10%</td>
<td>1.28</td>
</tr>
<tr>
<td>20%</td>
<td>0.84</td>
</tr>
<tr>
<td>40%</td>
<td>0.25</td>
</tr>
</tbody>
</table>

\[Pr[X \geq 1.65] = 5% \rightarrow \text{there is a 5\% chance } X \text{ lies more than 1.65 SD above the mean}\]

- Thus:

\[Pr[-1.65 \leq X \leq 1.65] = 1 - 2 \times Pr[X \geq 1.65] = 90\%\]

- To use this, have to reduce random variable \(f\) to have 0 mean and unit variance
Transforming $f$

- **Transformed value for $f$**:
  - Subtract the mean and divide by the standard deviation
    \[
    \frac{f - \rho}{\sqrt{\rho(1 - \rho)/N}}
    \]

- **Resulting equation**:
  \[
  Pr[-z \leq \frac{f - \rho}{\sqrt{\rho(1 - \rho)/N}} \leq z] = c
  \]

- **Solving for $\rho$**:
  \[
  \rho = (f + \frac{z^2}{2N} + Z\sqrt{\frac{f}{N} - \frac{f^2}{N} + \frac{z^2}{4N^2}})/(1 + \frac{z^2}{N})
  \]
Comparing Mining Models

• Question: if model A is better than model B over the given domain

• Solution I: assume infinite amount of data
  ➢ Sample infinitely many dataset of specified size
  ➢ Obtain cross-validation estimate on each dataset for each scheme
  ➢ Check if mean accuracy for scheme A is better than mean accuracy for scheme B

• Solution II: with limited data in practice
  ➢ paired t-test
    o Perform the same cross-validation twice against model A and B
    o Decision can be drawn if the difference is significant
Distribution of Means

• Let \( x_1, x_2, \ldots, x_k \) and \( y_1, y_2, \ldots, y_k \) the 2\( k \) samples for the \( k \) separate datasets.

• \( m_x, m_y \) are the means; \( \mu_x, \mu_y \) are the true means.

• With enough samples (\( k > 100 \)), the mean of a set of independent samples is normally distributed.

• Estimated variances of the means are \( \sigma_x^2/k \) and \( \sigma_y^2/k \).

• Then \( m_x \) and \( m_y \) are \textit{approximately} normally distributed with mean 0, variance 1.

\[
\frac{m_x - \mu_x}{\sqrt{\sigma_x^2/k}} \quad \frac{m_y - \mu_y}{\sqrt{\sigma_y^2/k}}
\]
Student’s distribution

- With small samples ($k < 100$) the mean follows *Student’s distribution* with $k-1$ degrees of freedom rather than normal distribution

- Confidence limits:

<table>
<thead>
<tr>
<th>Pr[$X \geq z$]</th>
<th>$z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1%</td>
<td>4.30</td>
</tr>
<tr>
<td>0.5%</td>
<td>3.25</td>
</tr>
<tr>
<td>1%</td>
<td>2.82</td>
</tr>
<tr>
<td>5%</td>
<td>1.83</td>
</tr>
<tr>
<td>10%</td>
<td>1.38</td>
</tr>
<tr>
<td>20%</td>
<td>0.88</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pr[$X \geq z$]</th>
<th>$z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1%</td>
<td>3.09</td>
</tr>
<tr>
<td>0.5%</td>
<td>2.58</td>
</tr>
<tr>
<td>1%</td>
<td>2.33</td>
</tr>
<tr>
<td>5%</td>
<td>1.65</td>
</tr>
<tr>
<td>10%</td>
<td>1.28</td>
</tr>
<tr>
<td>20%</td>
<td>0.84</td>
</tr>
</tbody>
</table>

Assuming 10 estimates

*Student’s distribution* with 9 degrees of freedom

*normal distribution*
Distribution of the differences

- Let the difference of the means \( m_d = m_x - m_y \)
- Then \( m_d \) also has a Student’s distribution with \( k-1 \) degrees of freedom
- Let \( \sigma_d^2 \) be the estimated variance of the difference
- \( t \)-statistic (the standardized version of \( m_d \)) is:
  \[
  t = \frac{m_d}{\sqrt{\sigma_d^2 / K}}
  \]
- Use \textit{t-statistic} to perform the \( t \)-test (also called Student’s test)
  - For a given confidence level, check whether the actual difference exceeds the confidence limit
Performing the test

- For a fixed significance level $\alpha$ (5% or 1% in practice)
  - If a difference is significant at the $\alpha\%$ level, there is a (100-$\alpha\%$) chance that the true means differ
- Divide the significance level by two because the test is two-tailed
  - I.e. the true difference can be +ve or – ve
- Look up the value for $z$ that corresponds to $\alpha/2$ from the table of Student’s distribution
  - I.e. For 1% test, use $z$ corresponding to 0.5%, i.e. $z=3.25$
- If $t \leq -z$ or $t \geq z$ then the difference is significant
  - I.e. the null hypothesis (that the difference is zero) can be rejected
Performance Metrics

• **Prediction tasks**
  - Average error
    - Gives an idea of systematic over- or under-prediction
  - Mean absolute percentage error (MAPE)
  - RMSE (root-mean-squared-error)
    - Square the errors, find their average, take the square root
  - Total SSE

• **Classification tasks**
  - Classification matrix
  - ROC Curve
  - Life Charts
  - Count for misclassification costs
    - Total sum of squared error
Misclassification Error

- Error = classifying a record as belonging to one class when it belongs to another class

- Error rate = percent of misclassified records out of the total records in the validation data

- Problem: any minimal probability of misclassification required for a classifier?
Naïve Rule

Naïve rule: classify all records as belonging to the most prevalent class

• Often used as benchmark: we hope to do better than that

• Exception: when goal is to identify high-value but rare outcomes, we may do well by doing worse than the naïve rule (see “lift” – later)
Confusion Matrix
(Classification Matrix)

<table>
<thead>
<tr>
<th>Actual Class</th>
<th>Predicted Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>25</td>
</tr>
<tr>
<td>0</td>
<td>2689</td>
</tr>
</tbody>
</table>

- **201** 1’s correctly classified as “1”
- **85** 1’s incorrectly classified as “0”
- **25** 0’s incorrectly classified as “1”
- **2689** 0’s correctly classified as “0”
n = 3000, total number of instances

**Overall error rate** = \( \frac{n_{0,1} + n_{1,0}}{n} = \frac{25 + 85}{3000} = 3.67\% \)

**Accuracy** = \( 1 - \text{error} = \frac{n_{0,0} + n_{1,1}}{n} = \frac{201 + 2689}{3000} = 96.33\% \)

If multiple classes, error rate is:

\[
\frac{\text{(sum of misclassified records)}}{\text{(total records)}}
\]
Cutoff for classification

• For each record, most DM algorithms classify via a 2-step process:
  1. Compute *probability of belonging to class “1”*
  2. Compare to cutoff value, and classify accordingly

• **Default cutoff value is 0.50**
  
  If >= 0.50, classify as “1”
  If < 0.50, classify as “0”

• **Can use different cutoff values**

• **Typically, error rate is lowest for cutoff = 0.50**
<table>
<thead>
<tr>
<th>Actual Class</th>
<th>Prob. of &quot;1&quot;</th>
<th>Actual Class</th>
<th>Prob. of &quot;1&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.996</td>
<td>1</td>
<td>0.506</td>
</tr>
<tr>
<td>1</td>
<td>0.988</td>
<td>0</td>
<td>0.471</td>
</tr>
<tr>
<td>1</td>
<td>0.984</td>
<td>0</td>
<td>0.337</td>
</tr>
<tr>
<td>1</td>
<td>0.980</td>
<td>1</td>
<td>0.218</td>
</tr>
<tr>
<td>1</td>
<td>0.948</td>
<td>0</td>
<td>0.199</td>
</tr>
<tr>
<td>1</td>
<td>0.889</td>
<td>0</td>
<td>0.149</td>
</tr>
<tr>
<td>1</td>
<td>0.848</td>
<td>0</td>
<td>0.048</td>
</tr>
<tr>
<td>0</td>
<td>0.762</td>
<td>0</td>
<td>0.038</td>
</tr>
<tr>
<td>1</td>
<td>0.707</td>
<td>0</td>
<td>0.025</td>
</tr>
<tr>
<td>1</td>
<td>0.681</td>
<td>0</td>
<td>0.022</td>
</tr>
<tr>
<td>1</td>
<td>0.656</td>
<td>0</td>
<td>0.016</td>
</tr>
<tr>
<td>0</td>
<td>0.622</td>
<td>0</td>
<td>0.004</td>
</tr>
</tbody>
</table>

- If cutoff is 0.50: eleven records are classified as “1”
- If cutoff is 0.80: seven records are classified as “1”
### Confusion Matrix for Different Cutoffs

#### Classification Confusion Matrix

<table>
<thead>
<tr>
<th>Actual Class</th>
<th>owner</th>
<th>non-owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>owner</td>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td>non-owner</td>
<td>2</td>
<td>10</td>
</tr>
</tbody>
</table>

Cut off Prob.Val. for Success (Updatable) **0.25**

<table>
<thead>
<tr>
<th>Predicted Class</th>
<th>Actual Class</th>
<th>owner</th>
<th>non-owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>owner</td>
<td>11</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>non-owner</td>
<td>4</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

Cut off Prob.Val. for Success (Updatable) **0.75**

<table>
<thead>
<tr>
<th>Predicted Class</th>
<th>Actual Class</th>
<th>owner</th>
<th>non-owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>owner</td>
<td>7</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>non-owner</td>
<td>1</td>
<td>11</td>
<td></td>
</tr>
</tbody>
</table>

Confusion matrices with cutoffs as 0.25, 0.5 and 0.75
The Variation of Accuracy and Overall error with Cutoff

<table>
<thead>
<tr>
<th>cutoff</th>
<th>accuracy</th>
<th>overall error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>0.05</td>
<td>0.75</td>
<td>0.25</td>
</tr>
<tr>
<td>0.1</td>
<td>0.75</td>
<td>0.25</td>
</tr>
<tr>
<td>0.15</td>
<td>0.791666667</td>
<td>0.208333333</td>
</tr>
<tr>
<td>0.2</td>
<td>0.833333333</td>
<td>0.166666667</td>
</tr>
<tr>
<td>0.25</td>
<td>0.791666667</td>
<td>0.208333333</td>
</tr>
<tr>
<td>0.3</td>
<td>0.791666667</td>
<td>0.208333333</td>
</tr>
<tr>
<td>0.35</td>
<td>0.833333333</td>
<td>0.166666667</td>
</tr>
<tr>
<td>0.4</td>
<td>0.833333333</td>
<td>0.166666667</td>
</tr>
<tr>
<td>0.45</td>
<td>0.833333333</td>
<td>0.166666667</td>
</tr>
<tr>
<td>0.5</td>
<td>0.875</td>
<td>0.125</td>
</tr>
<tr>
<td>0.55</td>
<td>0.833333333</td>
<td>0.166666667</td>
</tr>
<tr>
<td>0.6</td>
<td>0.833333333</td>
<td>0.166666667</td>
</tr>
<tr>
<td>0.65</td>
<td>0.875</td>
<td>0.125</td>
</tr>
<tr>
<td>0.7</td>
<td>0.791666667</td>
<td>0.208333333</td>
</tr>
<tr>
<td>0.75</td>
<td>0.75</td>
<td>0.25</td>
</tr>
<tr>
<td>0.8</td>
<td>0.791666667</td>
<td>0.208333333</td>
</tr>
<tr>
<td>0.85</td>
<td>0.75</td>
<td>0.25</td>
</tr>
<tr>
<td>0.9</td>
<td>0.708333333</td>
<td>0.291666667</td>
</tr>
<tr>
<td>0.95</td>
<td>0.666666667</td>
<td>0.333333333</td>
</tr>
<tr>
<td>1</td>
<td>0.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>

- Accuracy level is stable around 0.8 for cutoff varies from 0.2 to 0.8
- Adjust the cutoff value so that the learning model classifies more records as the high-value class
When One Class is More Important

• In many cases it is more important to identify members of one class than the other
  ➢ Tax fraud
  ➢ Credit default
  ➢ Response to promotional offer
  ➢ Detecting electronic network intrusion
  ➢ Predicting delayed flights

• In such cases, we would tolerate greater overall error, in return for better identifying the important class for further attention
Alternate Accuracy Measures

If “C₁” is the important class

- **Sensitivity** = % of “C₁” class correctly classified
- **Specificity** = % of “C₀” class correctly classified
- **False positive rate** = % of predicted “C₁’s” that were not “C₁’s”
- **False negative rate** = % of predicted “C₀’s” that were not “C₀’s”
- Plot \{sensitivity, 1-specificity\} versus the cutoff values help find a cutoff value balancing these measures
ROC Curve

• Receiver operating characteristic (ROC)
  
  ➢ Plots the pairs \{sensitivity, 1\text{-specificity}\} as the cutoff value increases from 0 to 1
  
  ➢ Better performance reflected from the top left corner of the curve (see next slide)
ROC Curve

![ROC Curve Graph]

- Sensitivity vs. 1-Specificity
- ROC (Red) and Random (Dashed Red)

COMP4332 Spring 2016  Performance Evaluation
Lift and Decile Charts

- The cases when a particular class is relatively rare and of much more interest than the other class
  - Tax cheats
  - Debt defaulters
  - Responders to a mailing
    - Given a scored dataset with each instance an estimated probability of responding,
    - Lift curve identifies the likely responders to a mailing

- Lift chart & Decile chart
Lift and Decile Charts – Cont.

- Compare performance of DM model to benchmark
  - In baseline, assume “no model but pick instances randomly”

- Measures ability of DM model to identify the important class, relative to its average prevalence

- Charts give explicit assessment of results over a large number of cutoffs
Lift and Decile Charts: How to Use

Compare lift to “no model” baseline

In lift chart: compare step function to straight line

In decile chart compare to ratio of 1
Lift and Decile Charts: How to Use

• Compare lift to baseline
  ➢ In lift chart: compare step function to straight line
  ➢ In decile chart compare to ratio of 1
After examining (e.g.,) 10 cases (x-axis), 9 owners (y-axis) have been correctly identified.

Verse in baseline, only 5 have been correctly identified.

The model gives a “lift” in predicting class 1 of $9/5 = 1.8$.
Decile Chart

- The same data can be plotted as a decile chart, see next slide
- Widely used in direct marketing predictive modeling
- The bars show the factor by which the model outperforms a random assignment of 0’s and 1’s
In the most left decile, model is twice as likely to identify the important class (compared to avg. prevalence)
Lift Charts: How to Compute

• Using the model’s classifications, sort records from most likely to least likely members of the important class

• Compute lift: Accumulate the correctly classified “important class” records (Y axis) and compare to number of total records (X axis)
Lift vs. Decile Charts

• Both embody concept of “moving down” through the records, starting with the most probable
• Decile chart does this in decile chunks of data
  ➢ Y axis shows ratio of decile mean to overall mean
• Lift chart shows continuous cumulative results
  ➢ Y axis shows number of important class records identified
Asymmetric Costs
Misclassification Costs May Differ

• The cost of making a misclassification error may be higher for one class than the other(s)
• Looked at another way, the benefit of making a correct classification may be higher for one class than the other(s)
Example – Response to Promotional Offer

Suppose we send an offer to 1000 people, with 1% average response rate ("1" = response, "0" = nonresponse)

• “Naïve rule” (classify everyone as “0”) has error rate of 1% (seems good)

• Using DM we can correctly classify eight 1’s as 1’s

   It comes at the cost of misclassifying twenty 0’s as 1’s and two 0’s as 1’s.
## The Confusion Matrix

<table>
<thead>
<tr>
<th>Predict as 1</th>
<th>Predict as 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual 1</td>
<td>8</td>
</tr>
<tr>
<td>Actual 0</td>
<td>20</td>
</tr>
</tbody>
</table>

Error rate = \( \frac{2 + 20}{\text{total}} = 2.2\% \) (higher than naïve rate)
Introducing Costs & Benefits

Suppose:
- Profit from a “1” is $10
- Cost of sending offer is $1

Then:
- Under naïve rule, all are classified as “0”, so no offers are sent: no cost, no profit
- Under DM predictions, 28 offers are sent.
  - 8 respond with profit of $10 each
  - 20 fail to respond, cost $1 each
  - 972 receive nothing (no cost, no profit)

- Net profit = $60
## Profit Matrix

<table>
<thead>
<tr>
<th></th>
<th>Predict as 1</th>
<th>Predict as 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual 1</td>
<td>$80</td>
<td>0</td>
</tr>
<tr>
<td>Actual 0</td>
<td>($20)</td>
<td>0</td>
</tr>
</tbody>
</table>
Lift (again)

- Adding costs to the mix, as above, does not change the actual classifications.

- Better: Use the lift curve and change the cutoff value for “1” to maximize profit.
Generalize to Cost Ratio

Sometimes actual costs and benefits are hard to estimate

• Need to express everything in terms of costs (i.e., cost of misclassification per record)
• Goal is to minimize the average cost per record
• A good practical substitute for individual costs is the ratio of misclassification costs (e.g., “misclassifying fraudulent firms is 5 times worse than misclassifying solvent firms”)

COMP4332 Spring 2016
Performance Evaluation
Minimizing Cost Ratio

$q_1 = \text{cost of misclassifying an actual “1”}$,
$q_0 = \text{cost of misclassifying an actual “0”}$

- Minimizing the cost ratio $q_1/q_0$ is identical to
- minimizing the average cost per record

Software* may provide option for user to specify cost ratio

*Currently unavailable in XLMiner
Note: Opportunity costs

- As we see, best to convert everything to costs, as opposed to a mix of costs and benefits
- E.g., instead of “benefit from sale” refer to “opportunity cost of lost sale”
- Leads to same decisions, but referring only to costs allows greater applicability
Cost Matrix (inc. opportunity costs)

<table>
<thead>
<tr>
<th>Predict as 1</th>
<th>Predict as 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual 1</td>
<td>$8</td>
</tr>
<tr>
<td>Actual 0</td>
<td>$20</td>
</tr>
</tbody>
</table>

Recall original confusion matrix (profit from a “1” = $10, cost of sending offer = $1):

<table>
<thead>
<tr>
<th>Predict as 1</th>
<th>Predict as 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual 1</td>
<td>8</td>
</tr>
<tr>
<td>Actual 0</td>
<td>20</td>
</tr>
</tbody>
</table>
Multiple Classes

For $m$ classes, confusion matrix has $m$ rows and $m$ columns

- Theoretically, there are $m(m-1)$ misclassification costs, since any case could be misclassified in $m-1$ ways
- Practically too many to work with
- In decision-making context, though, such complexity rarely arises – one class is usually of primary interest
Adding Cost/Benefit to Lift Curve

• Sort records in descending probability of success
• For each case, record cost/benefit of actual outcome
• Also record cumulative cost/benefit
• Plot all records
  ➢ X-axis is index number (1 for 1st case, n for nth case)
  ➢ Y-axis is cumulative cost/benefit
  ➢ Reference line from origin to \( y_n \) (\( y_n \) = total net benefit)
Lift Curve May Go Negative

• If total net benefit from all cases is negative, reference line will have **negative slope**

• Nonetheless, goal is still to use cutoff to select the point where net benefit is at a maximum
Negative slope to reference curve

Lift Curve Incorporating Costs
Oversampling and Asymmetric Costs
Rare Cases

Asymmetric costs/benefits typically go hand in hand with presence of rare but important class

- Responder to mailing
- Someone who commits fraud
- Debt defaulter

• Often we oversample rare cases to give model more information to work with
• Typically use 50% “1” and 50% “0” for training
Example

• Following graphs show optimal classification under three scenarios:
  ➢ assuming equal costs of misclassification
  ➢ assuming that misclassifying “o” is five times the cost of misclassifying “x”
  ➢ Oversampling scheme allowing DM methods to incorporate asymmetric costs
Classification: equal costs

Classification Assuming equal costs of misclassification
Classification: Unequal costs

Classification Assuming unequal costs of misclassification
Oversampling Scheme

Oversample “o” to appropriately weight misclassification costs
An Oversampling Procedure

1. Separate the responders (rare) from non-responders
2. Randomly assign half the responders to the training sample, plus equal number of non-responders
3. Remaining responders go to validation sample
4. Add non-responders to validation data, to maintain original ratio of responders to non-responders
5. Randomly take test set (if needed) from validation
Classification Using Triage

Take into account a gray area in making classification decisions

• Instead of classifying as $C_1$ or $C_0$, we classify as
  $C_1$
  $C_0$
  Can’t say

• The third category might receive special human review
Summary

- Apply different evaluation metrics when comparing across DM models, choosing the optimized parameters of a specific DM model, and comparing to the baseline.
- Use major metrics when applicable:
  - confusion matrix
  - error rate
  - predictive error
- Use other metrics like lift, decile, cost matrix when:
  - one class is more important
  - asymmetric costs
- Use oversampling when important class is rare.
- In all cases, metrics computed from validation dataset.