Model Evaluation

Jun Du
The University of Western Ontario
jdu43@uwo.ca
Outline

• Metrics for Performance Evaluation
  – How to evaluate the performance of a model?

• Methods for Performance Evaluation
  – How to obtain reliable estimates?

• Methods for Model Comparison
  – How to compare the relative performance among competing models?
Metrics for Performance Evaluation

• Focus on the predictive capability of a model
  – Rather than how fast it takes to classify or build models, scalability, etc.

• Confusion Matrix:

<table>
<thead>
<tr>
<th>ACTUAL CLASS</th>
<th>PREDICTED CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class=Yes</td>
<td>Class=Yes</td>
</tr>
<tr>
<td>Class=Yes</td>
<td>Class=No</td>
</tr>
<tr>
<td>Class=No</td>
<td>Class=Yes</td>
</tr>
<tr>
<td>Class=No</td>
<td>Class=No</td>
</tr>
</tbody>
</table>

TP: true positive
FN: false negative
FP: false positive
TN: true negative
## Metrics for Performance Evaluation

- **Most widely-used metric:**

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

<table>
<thead>
<tr>
<th>ACTUAL CLASS</th>
<th>PREDICTED CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Class=Yes</td>
</tr>
<tr>
<td>Class=Yes</td>
<td>TP</td>
</tr>
<tr>
<td>Class=No</td>
<td>FP</td>
</tr>
</tbody>
</table>
Limitation of Accuracy

• Consider a 2-class problem
  – Number of Class 0 examples = 9,990
  – Number of Class 1 examples = 10

• If model predicts everything to be class 0, accuracy is
  \( \frac{9990}{10000} = 99.9 \% \)
  – Accuracy is misleading because model does not detect any class 1 example
Cost Matrix

<table>
<thead>
<tr>
<th>ACTUAL CLASS</th>
<th>PREDICTED CLASS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C(i</td>
<td>j)</td>
</tr>
<tr>
<td>Class=Yes</td>
<td>C(Yes</td>
<td>Yes)</td>
</tr>
<tr>
<td>Class=No</td>
<td>C(Yes</td>
<td>No)</td>
</tr>
</tbody>
</table>

$C(i | j)$: Cost of misclassifying class j example as class i
Computing Cost of Classification

<table>
<thead>
<tr>
<th>Cost Matrix</th>
<th>PREDICTED CLASS</th>
<th>ACTUAL CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>C(i</td>
<td>j)</td>
<td>+</td>
</tr>
<tr>
<td>+</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>-</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

E.g.,
Cancer patient diagnosed as non-cancer v.s.
Non-cancer patient diagnosed as cancer

<table>
<thead>
<tr>
<th>Confusion Matrix</th>
<th>PREDICTED CLASS</th>
<th>ACTUAL CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>+</td>
<td>150</td>
<td>40</td>
</tr>
<tr>
<td>-</td>
<td>60</td>
<td>250</td>
</tr>
</tbody>
</table>

Model 1: Accuracy = 80%
Cost = 4060

<table>
<thead>
<tr>
<th>Confusion Matrix</th>
<th>PREDICTED CLASS</th>
<th>ACTUAL CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>+</td>
<td>250</td>
<td>45</td>
</tr>
<tr>
<td>-</td>
<td>5</td>
<td>200</td>
</tr>
</tbody>
</table>

Model 2: Accuracy = 90%
Cost = 4505
Misclassification Cost

• Different classification mistakes yield different cost
  – Misclassification cost (instead of accuracy) is usually used to evaluate the predictive model (to be minimized)
  – Cost matrix is usually required (according to domain knowledge)

• Most traditional classification algorithms aim to minimize error rate (maximize accuracy)
  – New algorithms have to be developed
  – Cost-sensitive learning
Precision and Recall, and F-measure

• **Precision**: exactness – what % of examples that the classifier labeled as positive are actually positive

\[
\text{precision} = \frac{TP}{TP + FP}
\]

• **Recall**: completeness – what % of positive examples did the classifier label as positive?

\[
\text{recall} = \frac{TP}{TP + FN}
\]

• **F measure ($F_1$ or F-score)**: harmonic mean of precision and recall

\[
F = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}
\]

• **Question**: What are the perfect scores for precision, recall and F measure? Why?
**Evaluation Metrics: Example**

<table>
<thead>
<tr>
<th>ACTUAL CLASS</th>
<th>PREDICTED CLASS</th>
<th>Cancer = yes</th>
<th>Cancer = no</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cancer = yes</td>
<td>90</td>
<td>210</td>
<td>300</td>
<td></td>
</tr>
<tr>
<td>Cancer = no</td>
<td>140</td>
<td>9560</td>
<td>9700</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>230</td>
<td>9770</td>
<td>10000</td>
<td></td>
</tr>
</tbody>
</table>

- **Accuracy** = \( \frac{90 + 9560}{10000} = 96.4\% \)
- **Precision** = \( \frac{90}{230} = 39.13\% \)
- **Recall** = \( \frac{90}{300} = 30.00\% \)
- ......
Other Metrics

• Time Complexity (speed)
  – time to construct the model (training time)
  – time to use the model (classification/prediction time)

• Robustness
  – handling noise and missing values

• Scalability
  – efficiency in handling large scale data

• Interpretability
  – understanding and insight provided by the model

• ......
Summary

- Confusion matrix is used to calculate all metrics
- **Accuracy / error rate** is the most common one
- When data is imbalanced (or errors have non-uniform costs), **misclassification cost** can be applied
- Other common metrics: precision, recall, F measure
Outline

• Metrics for Performance Evaluation
  – How to evaluate the performance of a model?

• Methods for Performance Evaluation
  – How to obtain reliable estimates?

• Methods for Model Comparison
  – How to compare the relative performance among competing models?
What Matters

• When building predictive models, what really matters is the performance of the models on the future unseen data.
  – i.e., the performance of the model when we make actual predictions.

• Given only the training data and the model built upon it, how can we reliably estimate the performance on future predictions.
Evaluation on Training Data

• The simplest way is to **directly apply the model back to the training data**, and estimate the performance.

• In this case, **we assume that the actual predictive performance is the same as the performance on training data**.

  – Can we?

• Recall

**Quick Questions:**

1) What model(s) have 100% accuracy on training data?

2) When can models **never** have 100% accuracy on training data?
Training vs. Testing --- Basic

• Building model on training set
• Testing model on independent test set
  – Data in test set plays no part in building model.

• Assumption:
  • Data from both training and test sets are i.i.d. (independently drawn from identical distribution)
  • i.e., both training and test data are representative samples of the underline problem.
  • Counterexample: ...

15
Training vs Testing --- Parameter Turning

- Some learning algorithms operate in two stages:
  - Stage 1: build the basic model structure
  - Stage 2: optimize parameter settings
- Example: $K$NN
  - Parameter $K$ has to be set up
- Proper procedure
  - Use three sets: training data, validation data, and test data
  - Validation data is used to optimize parameters
- Why would we do this? Why can’t directly use test data to determine the parameters?
Training vs. Testing --- Most Data

• Once evaluation is complete, all the data can be used to build the final classifier.

• Generally,
  – The larger the training set the better the model
  – The larger the test set the more reliable the accuracy estimate

• Dilemma: ideally both training set and test set should be large!
Holdout Methods

• Holdout
  – Given data is randomly partitioned into two independent sets
    • Training set (e.g., 2/3) for model construction
    • Test set (e.g., 1/3) for accuracy estimation
  – Estimation is more reliable (on the separate data set)
    • What if training or test data happen to be NOT representative?

• Random subsampling --- Repeated holdout
  – Repeat holdout $k$ times, accuracy = avg. of the accuracies obtained
    • More reliable accurate estimate
  – Still not optimal: the different test sets overlap
Cross-validation (1)

• $k$-fold Cross-validation
  – Randomly partition the data into $k$ \textit{mutually exclusive} subsets, each approximately equal size
  – At $i$-th iteration, use $i$-th subset as test set, and others together as training set

• Stratified cross-validation: (recall \textit{stratified sampling})
  – Folds are stratified so that class dist. in each fold is approx. the same as that in the initial data
Cross-validation (2)

• Most commonly used method for evaluation
  – Stratified 10-fold cross-validation
  – Extensive empirical and theoretical studies have shown that this is a very good choice to get an accurate estimate

• Even better: repeated stratified cross-validation
  – E.g. 10-fold cross-validation is repeated 10 times and results are averaged
Leave-one-out

• A special case of cross-validation --- \( k = \# \) of examples
  – Good --- makes best use of the data
  – Bad --- computationally expensive
• Stratification is not possible
  – Only one example in the test set
• Extreme example
  – 100 training examples, 50 positive, 50 negative
  – Naïve learner: always predicts majority class
  – What is the actual predictive accuracy?
  – What is the predictive accuracy estimated by LOO?
Summary

• Holdout: training set + test set
  – Basic method
  – A separate validation set can be apply for parameter estimation

• Repeated holdout
  – More reliable

• Cross validation, Stratified cross validation
  – Even more reliable, very commonly used

• Leave one out
  – Expensive
  – Possible to make big mistakes
Outline

• Metrics for Performance Evaluation
  – How to evaluate the performance of a model?

• Methods for Performance Evaluation
  – How to obtain reliable estimates?

• Methods for Model Comparison
  – How to compare the relative performance among competing models?
Model Comparison

• Model comparison strongly relies on some statistics knowledge, including:
  – Mean, standard deviation, variance,
  – Bernoulli trial, binomial distribution,
  – Confidence interval, hypothesis test, t test
• If you are familiar with these terms, try to derive the formulas in the next few slides
• If not, you still can directly apply the formulas
• Given two models:
  – Model $M_1$: accuracy = 85%, tested on 30 instances
  – Model $M_2$: accuracy = 85%, tested on 5000 instances

• 85% is the estimated accuracy for $M_1$ and $M_2$, how reliable is this estimation for $M_1$ and $M_2$?
  – Which one is more reliable? What is your gut-feeling?
  – Can we quantify such confidence of the estimation?
Confidence Interval (2)

Formula:

- Given \( n \) test instances and the estimated accuracy \( Acc \)
- With certain probability, the true accuracy lies in the interval

\[
Acc \pm z_n \sigma_{Acc} = Acc \pm z_n \times \sqrt{\frac{Acc \times (1 - Acc)}{n}}
\]

- \( Z_n \) is determined by \( n \) and the probability (according to \text{T distribution table}; \( z_n \approx 1.96 \) when \( n \geq 30 \) and probability = 95%).
- \( \sigma_{Acc} \) is the standard deviation of the accuracy;

\[
\sigma_{Acc} = \sqrt{\frac{Acc \times (1 - Acc)}{n}}
\]
Confidence Interval (3)

Example:

• Model \( M_1 \): accuracy = 85\%, tested on 30 instances
  – With 95\% probability, the true accuracy lies in the interval
    \[
    0.85 \pm 1.96 \times \sqrt{\frac{0.85 \times 0.15}{30}} \rightarrow [0.72, 0.98]
    \]

• Model \( M_2 \): accuracy = 85\%, tested on 5,000 instances
  – With 95\% probability, the true accuracy lies in the interval
    \[
    0.85 \pm 1.96 \times \sqrt{\frac{0.85 \times 0.15}{5000}} \rightarrow [0.84, 0.86]
    \]

• Accuracy estimation (85\%) on \( M_2 \) is more reliable
Comparing 2 Models (1)

• Given two models, say $M_1$ and $M_2$, which is better?
  – Model $M_1$: accuracy = $acc_1$, tested on $n_1$ instances
  – Model $M_2$: accuracy = $acc_2$, tested on $n_2$ instances

• Basic idea:
  – Consider the performance difference: $d = acc_1 - acc_2$
  – Calculate the confidence interval of $d$: $d \pm z_n \sigma_d$
  – If the confidence interval $[d - z_n \sigma, d + z_n \sigma]$ contains 0, the performance difference is not statistically significant;
  – Otherwise, difference is significant (i.e., one is better than the other)

• Key issue: calculating the confidence interval: $d \pm z_n \sigma_d$
  – Calculating the standard deviation of $d$: $\sigma_d$
Comparing 2 Models (2)

• What we know:
  – Model $M_1$ (accuracy = $acc_1$ on $n_1$ instances):
  – Model $M_2$ (accuracy = $acc_2$ on $n_2$ instances):

• Performance difference: $d = acc_1 - acc_2$

\[
\sigma_d = \sqrt{\sigma_1^2 + \sigma_2^2} = \sqrt{\frac{Acc_1 \times (1 - Acc_1)}{n_1} + \frac{Acc_2 \times (1 - Acc_2)}{n_2}}
\]

• With 95% probability, the true performance difference lies in the interval:

\[
d \pm z \sigma_d = d \pm 1.96 \sqrt{\frac{Acc_1 \times (1 - Acc_1)}{n_1} + \frac{Acc_2 \times (1 - Acc_2)}{n_2}}
\]
Comparing 2 Models (3)

Examples:

- Model $M_1$: $acc_1 = 85\%$, tested on $n_1=30$ instances
- Model $M_2$: $acc_2 = 75\%$, tested on $n_2=5000$ instances

Performance difference:

- $d = acc_1 - acc_2 = 0.1$;
- $\sigma_d = \sqrt{\frac{0.85 \times 0.15}{30} + \frac{0.75 \times 0.26}{5000}} = 0.0665$
- With 95\% probability, the true performance difference lies in the interval
  \[ d \pm 1.96 \sigma_d = 0.1 \pm 1.96 \times 0.0665 = 0.1 \pm 0.128 \]

- Such interval contains 0, we conclude:
  - The performance difference between $M_1$ and $M_2$ is not statistically significant as a 95\% confidence level.
Comparing 2 Algorithms (1)

• How “algorithms” differ from “models”?
  – “Algorithms” refer to the leaning techniques, such as decision tree, naïve bayes, etc.
  – “Models” refer to the specific predictive models built on given training data, according to certain learning algorithms.
  – Given different training data, multiple models can be built from same learning algorithm.

• $K$-fold cross validation is most commonly used to compare two algorithms (given the same data)
  – i.e., $k$ models are built for each algorithm
Comparing 2 Algorithms (2)

Basic Idea

• Conduct $k$-fold cross validation for the two algorithms
  – Algorithm 1: $M_{11}, M_{12}, M_{13}, ..., M_{1k}$
  – Algorithm 2: $M_{21}, M_{22}, M_{23}, ..., M_{2k}$
  – $M_{1i}$ and $M_{2i}$ are paired, as being built on the same training set, and tested on the same test set
  – Denote $acc_{1i}$ and $acc_{2i}$ the accuracy for $M_{1i}$ and $M_{2i}$ respectively

• Calculate performance difference for each model pair:
  \[ d_i = acc_{1i} - acc_{2i} \quad (i=1, ..., k) \]
  – We have $k$ such performance differences
Comparing 2 Algorithms (3)

Basic Idea (continue)

• Calculate confidence interval for the mean of $d$

\[ \bar{d} \pm z_k \sigma_{\bar{d}} = \bar{d} \pm z_k \sqrt{\frac{1}{k(k-1)} \sum_{i=1}^{k} (d_i - \bar{d})^2} \]

- $Z_k$ is determined by $k$ and the probability according to T distribution table;
- $\sigma_{\bar{d}}$ is the standard deviation of the mean of $d$;

\[ \sigma_{\bar{d}} = \sqrt{\frac{1}{k(k-1)} \sum_{i=1}^{k} (d_i - \bar{d})^2} \]

• Check if the confidence interval contains 0.
Summary

• “Comparing two algorithms” based on cross-validation is very commonly used in data mining

• You should know how to do it in Excel, Weka, or other data mining / analytics software packages
Demonstration

• T-test
  – T-test in Excel: comparing two group of numbers

• Algorithm comparison in Weka
  – “Experimenter” module
  – Comparing two (or more) algorithms on multiple data sets