Evaluation Measures

Sebastian Pölsterl

Computer Aided Medical Procedures | Technische Universität München

April 28, 2015
Outline

1 Classification
   1. Confusion Matrix
   2. Receiver operating characteristics
   3. Precision-Recall Curve

2 Regression

3 Unsupervised Methods

4 Validation
   1. Cross-Validation
   2. Leave-one-out Cross-Validation
   3. Bootstrap Validation

5 How to Do Cross-Validation
Performance Measures: Classification

Confusion Matrix

Deterministic Classifiers

Multi-class
- No Change Correction
  - Accuracy
  - Error Rate
  - Micro/Macro Average
- Change Correction
  - Cohen's Kappa
  - Fleiss' Kappa

Single-class
- TP/FP Rate, Precision, Recall, Sensitivity, Specificity, $F_1$-Measure, Dice, Geometric Mean

Scoring Classifiers

Graphical Measures
- ROC Curves
- PR Curves
- Lift Charts
- Cost Curves

Summary Statistics
- Area under the curve
  - H Measure
Let us consider a binary classification problem:

- **True Positive (TP)** = positive sample **correctly classified** as belonging to the positive class
- **False Positive (FP)** = negative sample **misclassified** as belonging to the positive class
- **True Negative (TN)** = negative sample **correctly classified** as belonging to the negative class
- **False Negative (FN)** = positive sample **misclassified** as belonging to the negative class
Confusion Matrix I

<table>
<thead>
<tr>
<th>Prediction</th>
<th>Ground Truth</th>
<th>Class A</th>
<th>Class B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class A</td>
<td>True positive</td>
<td></td>
<td>False positive</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Type I Error ($\alpha$)</td>
</tr>
<tr>
<td>Class B</td>
<td>False negative</td>
<td></td>
<td>True negative</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Type II Error ($\beta$)</td>
</tr>
</tbody>
</table>

- Let class A indicate the positive class and class B the negative class.
- Accuracy = \( \frac{TP + TN}{TP + FP + TN + FN} \)
- Error rate = 1 - Accuracy
# Confusion Matrix II

<table>
<thead>
<tr>
<th></th>
<th>Ground Truth</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Class A</td>
</tr>
<tr>
<td><strong>Pred.</strong></td>
<td><strong>Class A</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Class B</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Sensitivity</th>
<th>Specificity</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>False negative rate</strong></td>
<td>$\frac{FN}{FN+TP} = 1 - \text{Sensitivity}$</td>
<td>$\frac{TN}{TN+FP} = 1 - \text{Specificity}$</td>
</tr>
</tbody>
</table>

- Sensitivity/True positive rate/Recall = $\frac{TP}{TP+FN}$
- Specificity/True negative rate = $\frac{TN}{TN+FP}$
- False negative rate = $\frac{FN}{FN+TP} = 1 - \text{Sensitivity}$
- False positive rate = $\frac{FP}{FP+TN} = 1 - \text{Specificity}$
### Confusion Matrix III

<table>
<thead>
<tr>
<th>Pred.</th>
<th>Ground Truth</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Class A</td>
<td>Class B</td>
<td></td>
</tr>
<tr>
<td>Class A</td>
<td>TP</td>
<td>FP</td>
<td>Positive predictive value</td>
</tr>
<tr>
<td>Class B</td>
<td>FN</td>
<td>TN</td>
<td>Negative predictive value</td>
</tr>
</tbody>
</table>

- Positive predictive value (PPV)/Precision = \(\frac{TP}{TP+FP}\)
- Negative predictive value (NPV) = \(\frac{TN}{TN+FN}\)
Multiple Classes – One vs. One

<table>
<thead>
<tr>
<th>Prediction</th>
<th>Ground Truth</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Class A</td>
</tr>
<tr>
<td>Class A</td>
<td>Correct</td>
</tr>
<tr>
<td>Class B</td>
<td>Wrong</td>
</tr>
<tr>
<td>Class C</td>
<td>Wrong</td>
</tr>
<tr>
<td>Class D</td>
<td>Wrong</td>
</tr>
</tbody>
</table>

• With $k$ classes confusion matrix becomes a $k \times k$ matrix.
• No clear notion of positives and negatives.
## Multiple Classes – One vs. All

### Table:

<table>
<thead>
<tr>
<th>Ground Truth</th>
<th>Class A</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pred. Class A</td>
<td>True positive</td>
<td>False positive</td>
</tr>
<tr>
<td>Other</td>
<td>False negative</td>
<td>True negative</td>
</tr>
</tbody>
</table>

- Choose one of $k$ classes as positive (here: class A).
- Collapse all other classes into negative to obtain $k$ different $2 \times 2$ matrices.
- In each of these matrices the number of true positives is the same as in the corresponding cell of the original confusion matrix.
Micro and Macro Average

- **Micro Average:**
  1. Construct a single $2 \times 2$ confusion matrix by summing up TP, FP, TN and FN from all $k$ one-vs-all matrices.
  2. Calculate performance measure based on this average.

- **Macro Average:**
  1. Obtain performance measure from each of the $k$ one-vs-all matrices separately.
  2. Calculate average of all these measures.
\[ F_1 \text{-Measure} \]

\( F_1 \)-measure is the harmonic mean of positive predictive value and sensitivity:

\[ F_1 = \frac{2 \cdot \text{PPV} \cdot \text{sensitivity}}{\text{PPV} + \text{sensitivity}} \quad (1) \]

- Micro Average \( F_1 \)-Measure:
  1. Calculate sums of TP, FP, and FN across all classes
  2. Calculate \( F_1 \) based on these values
- Macro Average \( F_1 \)-Measure:
  1. Calculate PPV and sensitivity for each class separately
  2. Calculate mean PPV and sensitivity
  3. Calculate \( F_1 \) based on mean values
1 Classification
   1. Confusion Matrix
   2. Receiver operating characteristics
   3. Precision-Recall Curve

2 Regression

3 Unsupervised Methods

4 Validation
   1. Cross-Validation
   2. Leave-one-out Cross-Validation
   3. Bootstrap Validation

5 How to Do Cross-Validation
Receiver operating characteristics (ROC)

- **Binary classifier** returns **probability** or **score** that represents the degree to which class an instance belongs to.

- The ROC plot compares **sensitivity** (y-axis) with **false positive rate** (x-axis) for all possible **thresholds** of the classifier’s score.

- It visualizes the **trade-off** between benefits (sensitivity) and costs (FPR).
ROC Curve

- Line from the lower left to upper right corner indicates **random classifier**.
- Curve of **perfect classifier** goes through the upper left corner at \((0, 1)\).
- A single confusion matrix corresponds to one point in ROC space.
- It is insensitive to changes in class distribution or changes in error costs.
Area under the ROC curve (AUC)

- The **AUC** is equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance (Mann-Whitney *U* test).

- The **Gini coefficient** is twice the area that lies between the diagonal and the ROC curve:

\[
\text{Gini coefficient} + 1 = 2 \cdot \text{AUC}
\]
Averaging ROC curves I

- **Merging**: Merge instances of $n$ tests and their respective scores and sort the complete set
- **Vertical averaging**:
  1. Take vertical samples of the ROC curves for fixed false positive rate
  2. Construct confidence intervals for the mean of true positive rates

![Vertical Average](chart.png)
Averaging ROC curves II

**Threshold averaging:**
1. Do merging as described above
2. Sample based on thresholds instead of points in ROC space
3. Create confidence intervals for FPR and TPR at each point
Disadvantages of ROC curves

• ROC curves can present an overly optimistic view of an algorithm’s performance if there is a large skew in the class distribution, i.e. the data set contains much more samples of one class.

• A large change in the number of false positives can lead to a small change in the false positive rate (FPR).

\[
FPR = \frac{FP}{FP + TN}
\]

• Comparing false positives to true positives (precision) rather than true negatives (FPR), captures the effect of the large number of negative examples.

\[
\text{Precision} = \frac{TP}{FP + TP}
\]
1 Classification
   1. Confusion Matrix
   2. Receiver operating characteristics
   3. Precision-Recall Curve

2 Regression

3 Unsupervised Methods

4 Validation
   1. Cross-Validation
   2. Leave-one-out Cross-Validation
   3. Bootstrap Validation

5 How to Do Cross-Validation
**Precision-Recall Curve**

- Compares precision ($y$-axes) to recall ($x$-axes) at different thresholds.
- PR curve of optimal classifier is in the upper-right corner.
- One point in PR space corresponds to a single confusion matrix.
- **Average precision** is the area under the PR curve.
Relationship to Precision-Recall Curve

• Algorithms that optimize the area under the ROC curve are not guaranteed to optimize the area under the PR curve
• **Example**: Dataset has 20 positive examples and 2000 negative examples.
1. Classification
   1. Confusion Matrix
   2. Receiver operating characteristics
   3. Precision-Recall Curve

2. Regression

3. Unsupervised Methods

4. Validation
   1. Cross-Validation
   2. Leave-one-out Cross-Validation
   3. Bootstrap Validation

5. How to Do Cross-Validation
Evaluating Regression Results

- Remember that the predicted value is **continuous**.
- Measuring the performance is based on comparing the actual value $y_i$ with the predicted value $\hat{y}_i$ for each sample.
- Measures are either the sum of squared or absolute differences.
Regression – Performance Measures

- Sum of absolute error (SAE):
  \[ \sum_{i=1}^{n} |y_i - \hat{y}_i| \]

- Sum of squared errors (SSE):
  \[ \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \]

- Mean squared error (MSE): \( \frac{1}{n} \text{SSE} \)
- Root mean squared error (RMSE): \( \sqrt{\text{MSE}} \)
1 Classification
   1. Confusion Matrix
   2. Receiver operating characteristics
   3. Precision-Recall Curve

2 Regression

3 Unsupervised Methods

4 Validation
   1. Cross-Validation
   2. Leave-one-out Cross-Validation
   3. Bootstrap Validation

5 How to Do Cross-Validation
Unsupervised Methods

• **Problem**: Ground truth is usually not available or requires manual assignment

• Without ground truth (*internal* validation):
  ○ Cohesion
  ○ Separation
  ○ Silhouette Coefficient

• With ground truth (*external* validation):
  ○ Jaccard index
  ○ Dice’s coefficient
  ○ (Normalized) mutual information
  ○ (Adjusted) rand index
Cohesion and Separation

- Requires definition of *proximity* measure, such as distance or similarity

\[
\text{cohesion}(C_i) = \sum_{x,y \in C_i} \text{proximity}(x, y)
\]

\[
\text{separation}(C_i, C_j) = \sum_{x \in C_i, y \in C_j} \text{proximity}(x, y)
\]
**Silhouette Coefficient**

- $a(i)$ is the mean distance between the $i$-th sample and all other points in the same class
- $b(i)$ the mean distance to all other points in the *next nearest cluster*
- The silhouette coefficient $s(i) \in [-1; 1]$ is defined as

\[
s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}
\]

- $s(i) = 1$ if the clustering is dense and well separated
- $s(i) = -1$ if the $i$-th sample was assigned incorrectly
- $s(i) = 0$ if clusters overlap
Jaccard Index and Dice’s Coefficient

- Consider two sets $S_1, S_2$ where one set is used as ground truth and the other was predicted.
- **Example**: Pixels in image classification or segmentation.
- **Jaccard Index**
  \[
  \text{Jaccard}(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|} \in [0; 1]
  \]
- **Dice’s coefficient**
  \[
  \text{Dice}(S_1, S_2) = \frac{2|S_1 \cap S_2|}{|S_1| + |S_2|} \in [0; 1]
  \]
1 Classification
   1. Confusion Matrix
   2. Receiver operating characteristics
   3. Precision-Recall Curve

2 Regression

3 Unsupervised Methods

4 Validation
   1. Cross-Validation
   2. Leave-one-out Cross-Validation
   3. Bootstrap Validation

5 How to Do Cross-Validation
Validation

- **Test error**: Prediction error over an independent sample.
- **Training error**: Average loss over the training samples

\[
\frac{1}{n} \sum_{i=1}^{n} L(y_i, \hat{f}(x_i))
\]

- As the model gets more complex it infers more information from the training data to represent more complicated underlying structures.
Validation – Training Error

- Training error consistently decreases with increasing model complexity, whereas Test error starts to increase again.
- Training error is not a good measure of performance.
• **Overfitting**: A model with zero or very low training error is likely to perform well on the training data but generalize badly (model too complex).

• **Underfitting**: Model does not capture the underlying structure and hence performs poorly (model too simple).
Validation – Ideal Situation

- Assume we have access to large amount of data.
- Construct three different sets
  1. **Training set**: Used to fit the model.
  2. **Validation set**: Estimate prediction error to choose best model (e.g. different costs $C$ for SVMs).
  3. **Test set**: Used to assess how well final model generalizes.
Cross-Validation

- **Cross-validation**: Split data set into $k$ equally large parts.
- **Stratified cross-validation**: Ensures that the ratio between classes is the same in each fold as in the complete dataset.
Leave-one-out Cross-Validation

- Use all but one sample for training and assess performance on the excluded sample.
- For a data set with $n$ samples, leave-one-out cross-validation is equivalent to $n$-fold cross-validation.
- Not suitable if data set is very large and/or training the classifier takes a long time.
Bootstrap Sampling

- The **bootstrap** is a general tool for assessing statistical accuracy.
- **Assumption**: Our data set is a representative portion of the overall population.
- **Bootstrap sampling**: Randomly draw samples with replacement from the original data set to generate new data sets of the same size.
Bootstrap Validation

• Bootstrap sampling is repeated $B$ times and samples not included in each bootstrap sample are recorded.

• Train model on each of the $B$ bootstrap samples.

• For each sample of the original data set, assess performance only on bootstrap samples not containing this sample:

$$\frac{1}{n} \sum_{i=1}^{n} \frac{1}{|C^{-i}|} \sum_{b \in C^{-i}} L(y_i, \hat{f}_b(x_i))$$
1 Classification
   1. Confusion Matrix
   2. Receiver operating characteristics
   3. Precision-Recall Curve

2 Regression

3 Unsupervised Methods

4 Validation
   1. Cross-Validation
   2. Leave-one-out Cross-Validation
   3. Bootstrap Validation

5 How to Do Cross-Validation
A Typical Strategy

1. Find a “good” subset of features that show fairly strong (univariate) correlation with the class labels
2. Using just this subset of features, build a multivariate classifier
3. Use cross-validation to estimate the unknown hyper-parameters and to estimate the prediction error of the final model.
A Typical Strategy

1. Find a “good” subset of features that show fairly strong (univariate) correlation with the class labels
2. Using just this subset of features, build a multivariate classifier
3. Use cross-validation to estimate the unknown hyper-parameters and to estimate the prediction error of the final model.

Is this the correct way to do cross-validation?
Scenario

- Consider a data set with 50 samples in two equal-sized classes and 5000 features that are independent of the class labels
- The true test error rate of any classifier is 50%
- Example:
  1. Choose 100 predictors with highest correlation with class labels
  2. Use a 1-Nearest Neighbor classifier based on these 100 features
  3. **Result**: Doing 50 simulations in this setting, yielded an average CV error rate of 1.4%
What Happened?

- Classifier had an **unfair advantage** because features were selected based on all **samples**
- This validates the requirement that the test set is **completely independent** of the training set, because the classifier has already “seen” the samples in the test set
What Happened?

Wrong

Correct

Correlations of Selected Features with Label

Frequency

-1.0 -0.5 0.0 0.5 1.0

0 100 300

-0.5 0.0 0.5 1.0

0 200 400
How to Do It Right?

1. Divide data set into $K$ folds at random
2. For each fold
   2.1 Find a subset of “good” features
   2.2 Using this subset, build a multivariate classifier, using all samples except those in fold $k$
   2.3 Use the classifier to predict the class label of samples in fold $k$
How to Do It Right?

1. Divide data set into $K$ folds at random
2. For each fold
   2.1 Find a subset of “good” features
   2.2 Using this subset, build a multivariate classifier, using all samples except those in fold $k$
   2.3 Use the classifier to predict the class label of samples in fold $k$

Result

The estimated mean error rate is 51.2%, which is much closer to the true test error rate.
How to Do It Right?

- Cross-validation must be applied to the **entire sequence of modeling steps**
- **Examples:**
  - Selection of features
  - Tuning of hyper-parameters
Conclusion

- Many different performance measures for classification exist.
- ROC and Precision-Recall curves can be applied for binary classifiers which return probabilities or scores.
- Cross-Validation is the most commonly used validation scheme.
- Bootstrap cannot only be used for validation, it can be used in many more applications as well (e.g. bagging).

Important

Every performance measure has its advantages and its disadvantages. **There is no best measure.** Therefore, you have to consider multiple measures to evaluate your model.
References (1)


References (2)