Announcements

Submission instructions for Mini-project #1 are available:

Everyone should create an individual account on CMT. Only one person (per team) submits the report and code. Collaborators should be listed as “authors” on the submission. Use the same email address for all interactions with CMT: firstname.lastname@mail.mcgill.ca

Late submissions should be submitted by email directly to me (-30%).
Terminology

- **Type of classification outputs:**
  - **True positive** (m11): Example of class 1 predicted as class 1.
  - **False positive** (m01): Example of class 0 predicted as class 1. Type I error.
  - **True negative** (m00): Example of class 0 predicted as class 0.
  - **False negative** (m10): Example of class 1 predicted as class 0. Type II error.

- **Total number of instances:** \( m = m_{00} + m_{01} + m_{10} + m_{11} \)

- **Error rate:** \( \frac{(m_{01} + m_{10})}{m} \)
  - If the classes are imbalanced (e.g. 10% from class 1, 90% from class 0), one can achieve low error (e.g. 10%) by classifying everything as coming from class 0!
Confusion matrix

- Many software packages output this matrix.

\[
\begin{bmatrix}
  m_{00} & m_{01} \\
  m_{10} & m_{11}
\end{bmatrix}
\]

- Be careful! Sometimes the format is slightly different.

(E.g. http://en.wikipedia.org/wiki/Precision_and_recall#Definition_classification_context)

Common measures

- **Accuracy**  
  \[
  \text{Accuracy} = \frac{(TP + TN)}{(TP + FP + FN + TN)}
  \]

- **Precision**  
  \[
  \text{Precision} = \frac{\text{True positives}}{\text{Total number of declared positives}} = \frac{TP}{(TP + FP)}
  \]

- **Recall**  
  \[
  \text{Recall} = \frac{\text{True positives}}{\text{Total number of actual positives}} = \frac{TP}{(TP + FN)}
  \]

- **False positive rate**  
  \[
  \text{False positive rate} = \frac{FP}{(FP + TN)}
  \]

- **Sensitivity** is the same as recall.

- **Specificity**  
  \[
  \text{Specificity} = \frac{\text{True negatives}}{\text{Total number of actual negatives}} = \frac{TN}{(FP + TN)}
  \]

- **F1 measure**  
  \[
  F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}
  \]
Trade-off

• Often have a trade-off between false positives and false negatives.

E.g. Consider 30 different classifiers trained on a class. Classify a new sample as positive if K classifiers output positive. Vary K between 0 and 30.

Receiver-operator characteristic (ROC) curve

• Characterizes the performance of a binary classifier over a range of classification thresholds

Data from 4 prediction results:

Example from: http://en.wikipedia.org/wiki/Receiver_operating_characteristic
Understanding the ROC curve

- Consider a classification problem where data is generated by 2 Gaussians (blue = negative class; red = positive class).
- Consider the decision boundary (shown as a vertical line on the left figure), where you predict Negative on the left of the boundary and predict Positive on the right of the boundary.
- Changing that boundary defines the ROC curve on the right.

Building the ROC curve

- In many domains, the empirical ROC curve will be non-convex (red line). Take the convex hull of the points (blue line).
Using the ROC curve

• To compare 2 algorithms over a range of classification thresholds, consider the Area Under the Curve (AUC).
  – A perfect algorithm has AUC=1.
  – A random algorithm has AUC=0.5.
  – Higher AUC doesn’t mean all performance measures are better.

Lessons for evaluating ML algorithms

• Always compare to a simple baseline:
  – In classification:
    • Classify all samples as the majority class.
    • Classify with a threshold on a single variable.
  – In regression:
    • Predict the average of the output for all samples.
    • Compare to a simple linear regression.

• Use K-fold cross validation to properly estimate the error. If necessary, use a validation set to estimate hyper-parameters.

• Consider appropriate measures for fully characterizing the performance: Accuracy, Precision, Recall, F1, AUC.