Lecture 20 - ROC Curves; k-Nearest Neighbors

ROC Curves

Receiver Operating Characteristic (ROC) curve is the plot between the true positive rate (TPR) and the false positive rate (FPR), where the TPR is defined as the y-axis and FPR is defined as the x-axis.

- ROC curves were first developed for RADAR systems, hence the name.
- Given a binary classifier and its threshold, the (x,y) coordinates of ROC space can be calculated from all the prediction result. You trace out a ROC curve by varying the threshold to get all of the points on the ROC.
- The diagonal between (0,0) and (1,1) separates the ROC space into two areas, which are left up area and right bottom area. The points above the diagonal represent good classification (better than random guess) which below the diagonal represent bad classification (worse than random guess).
- *What is the perfect prediction point in a ROC curve?*

Area Under the Curve (AUC)

Area Under Curve (AUC) is a common measure of how good a test is. It is simply the area under the ROC curve. Random guessing can achieve the diagonal line, so the minimum AUC is 1/2. The maximum AUC is 1, which is achieved by a test that is always right; the ROC curve is along the left and top axes.

Example

1. Suppose you have a target detection task that you would like to evaluate using ROC curve analysis. You emplaced 10 targets and collected aerial hyperspectral imagery over 10 km^2 . Then, suppose you ran a set of alarm generation and target detection algorithms over the collected data. Your algorithms produced the following list of alarm confidence values. You have already matched each of these alarms to a location on the ground and compared them with you ground truth. True targets, based on your ground truth, are marked with a "T" in the second column. Draw the associated ROC cure for these results.

Alarm confidence values	0.91	0.90	0.80	0.79	0.77	0.75	0.50	0.40	0.39	0.38	0.37	0.25	0.10	0.09	0.01
Ground truth	T	T		T				T						T	

1. Suppose you were segmenting a data set into three classes (e.g., vegetation, man-made materials, sand) and wanted to evaluate your results. Would using a ROC curve be an appropriate method for evaluation? Why or why not?

Non-parametric Generative Classification

We have introduced the probabilistic generative classifier, and, as we discussed, the probabilistic generative classifier requires us to assume a parametric form for each class (e.g., each class is represented by a multivarite Gaussian distribution, etc.). Because of this, the probabilistic generative classifier is a *parametric* approach.

- Parametric approaches have the drawback that the functional parametric form needs to be decided/assumed in advance and, if chosen poorly, might a poor model of the distribution that generates the data resulting in poor performance.

Non-parametric methods are those that do not assume a particular generating distribution for the data. The **K-nearest nerighbors (K-NN)** algorithm is one example of a non-parametric classifier.

K-Nearest Neighbors Classifier

Nearest neighbors methods compare a test point to the k nearest training data points and then estimate an output value based on the desired/true output values of the k nearest training points.

- Essentially, there is no "training" other than storing the training data points and their desired outputs
- In test, you need to:

  1. Determine which k neighbors in the training data are closest to the test point; and,
  2. Determine the output value for the test point.

In order to find the k *nearest-neighbors* in the training data, you need to define a **similarity measure** or a **dissimilarity measure**. The most common dissimilarity measure is Euclidean distrance:

- Euclidean distance: $d_E(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{(\mathbf{x}_1 - \mathbf{x}_2)^T(\mathbf{x}_1 - \mathbf{x}_2)}$
- City-block distance: $d_{CB}(\mathbf{x}_1, \mathbf{x}_2) = \sum_{i=1}^n \left|\mathbf{x}_{1i} - \mathbf{x}_{2i}\right|$
- Mahalanobis distance: $d_M(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{(\mathbf{x}_1 - \mathbf{x}_2)^T\Sigma^{-1}(\mathbf{x}_1 - \mathbf{x}_2)}$
- Cosine angle similarity: $\cos(\theta) = \frac{\mathbf{x}_1^T\mathbf{x}_2}{\|\mathbf{x}_1\|_2^2\|\mathbf{x}_2\|_2^2}$
- and many more.

If you are doing classification, once you find the k nearest neighbors to your test point in the training data, then you can determine the class label of your test point using (most commonly) **majority vote**.

- If there are ties, they can be broken randomly or using schemes like applying the label to the closest data point in the neighborhood.

- What happens when there are imbalanced classes?
- Is k-NN sensitive to data scaling?

```
In [ ]:
```