ROC Graphs: Notes and Practical Considerations for Data Mining Researchers

Tom Fawcett[1]

[1]Intelligent Enterprise Technologies Laboratory, HP Laboratories Palo Alto

Outline

ROC Graphs

- Receiver Operating Characteristics (ROC) graphs are a useful technique for organizing classifiers and visualizing their performance.

- An ROC graph is a technique for visualizing, organizing and selecting classifiers based on their performance. ROC graphs have long been used in signal detection theory to depict the trade-off between hit rates and false alarm rates of classifiers.

ROC Graphs

- Although ROC graphs are apparently simple, there are some common misconceptions and pitfalls when using them in practice.
- One of the earliest adopters of ROC graphs in machine learning was Spackman (1989), who demonstrated the value of ROC curves in evaluating and comparing algorithms.
- ROC graphs are conceptually simple, but there are some non-obvious complexities that arise when they are used in research.

Confusion matrix and performance metrics

Hypothesized class \ True class	p	n
Y	True Positives	False Positives
N	False Negatives	True Negatives
Column totals:	P	N

$$\text{FP rate} = \frac{FP}{N} \qquad \text{TP rate} = \frac{TP}{P} = \text{Recall}$$

$$\text{Precision} = \frac{TP}{TP + FP} \qquad \text{F-score} = \text{Precision} \times \text{Recall}$$

$$\text{Accuracy} = \frac{TP + TN}{P + N}$$

Figure: A confusion matrix and several common performance metrics that can be calculated from it.

Classification model

- Considering a two-class classification problem: each instance I is mapped to one element of the set $\{p, n\}$ of positive and negative class labels.
- A *classification model* is a mapping from instances to predicted classes.
- To distinguish between the actual class and the predicted class we use the labels $\{Y, N\}$ for the class predictions produced by a model.

TP, FP, TN and FN

- Given a classifier and an instance, there are four possible outcomes:
  - TP: if a positive instance is classified as positive.
  - FN: if a positive instance is classified as negative.
  - TN: if a negative instance is classified as negative.
  - FN: if a negative instance is classified as positive.

- A two-by-two *confusion matrix* (or contingency matrix) can be constructed representing the dispositions of the set of instances.

Metrics from the confusion matrix

- True Positive rate (also called *hit rate* or *recall*):

$$TP\ rate \approx \frac{positives\ correctly\ classified}{total\ positives}$$

- False Positive rate (also called *false alaram*):

$$FP\ rate \approx \frac{negatives\ incorrectly\ classified}{total\ negatives}$$

- Additional terms associated with ROC curves:

$$Sensitivity = Recall$$

$$Specificity = \frac{TN}{FP + TN} = 1 - FP\ rate$$

$$Positive\ predictive\ value = Precision$$

ROC Graphs

- ROC graphs are two-dimensional graphs in which *TP rate* is plotted on the Y axis and *FP rate* is plotted on the X axis.
- A ROC graph depicts relative trade-offs between benefits (true positives) and costs (false positives).
- A *discrete* classifier is one that outputs only a class label. Each discrete classifier produces an *(FP rate, TP rate)* pair, which corresponds to a single point in ROC space.

An example of a ROC graph

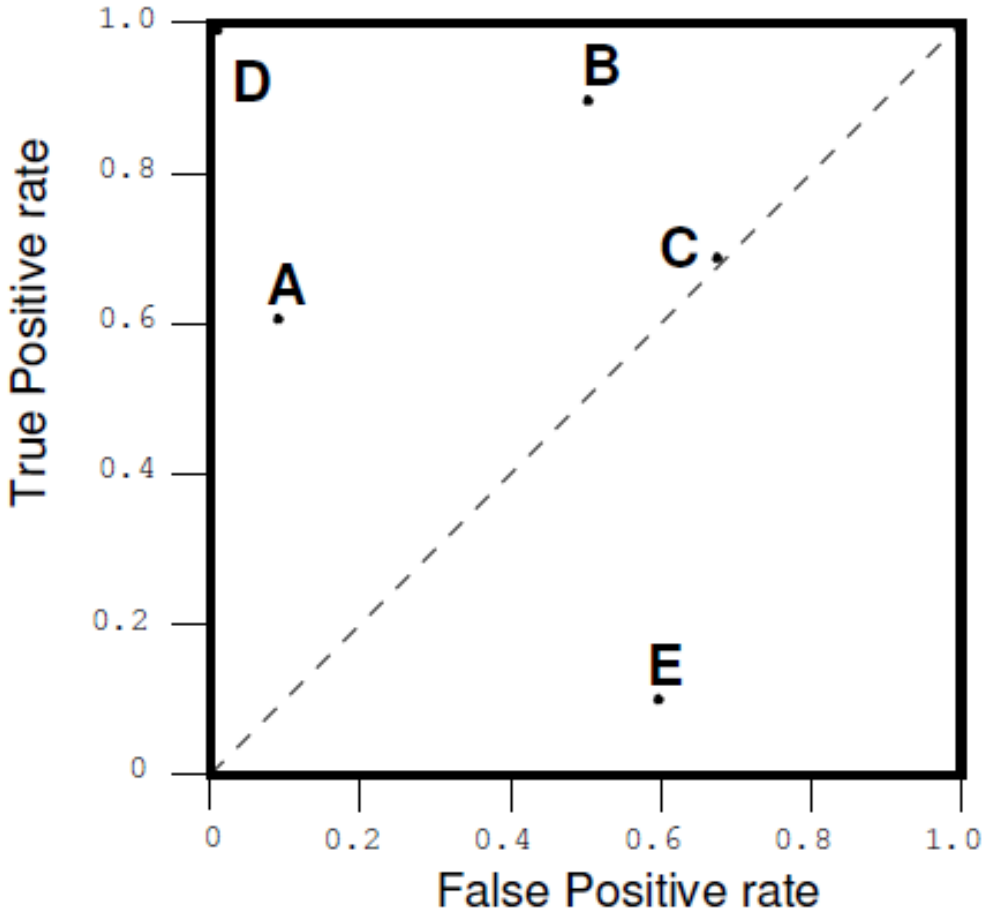

Figure: A basic ROC graph showing five discrete classifiers.

Important Points in ROC space

- The lower left point (0,0) represents the strategy of never issuing a positive classification.
- The opposite strategy is represented by the upper right point (1,1).
- The point (0,1) represents perfect classification.

Important Points in ROC space

- Informally, one point in ROC space is better than another if it is to the northwest (TP rate is higher, FP rate is lower, or both) of the first.
- Classifiers appearing on the left hand-side of an ROC graph, near the X axis, may be thought of as "conservative": they make positive classifications only with strong evidence so they make few false positive errors, but they often have low true positive rates as well.
- Classifiers on the upper right-hand side of an ROC graph may be thought of as "liberal".

Outline

Random Performance

- The diagonal line y = x represents the strategy of randomly guessing a class.
- A random classifier will produce an ROC point that "slides" back and forth on the diagonal based on the frequency with which it guesses the positive class. In order to get away from this diagonal into the upper triangular region, the classifier must exploit some information in the data.
- Any classifier that appears in the lower right triangle performs worse than random guessing. This triangle is therefore usually empty in ROC graphs.

Curves

- Some classifiers yield an instance *probability* or *score*, a numeric value that represents the degree to which an instance is a member of a class.

- Such a ranking or scoring classier can be used with a threshold to produce a discrete (binary) classifier, we may imagine varying a threshold from $-\infty$ to $+\infty$ and tracing a curve through ROC space. But this is a poor way of generating a ROC curve.

Outline

Relative versus absolute scores

- A consequence of relative scoring is that classifier scores should not be compared across model classes. One model class may be designed to produce scores in the range [0,1] while another produces scores in [-1,+1] or [1,100]. Comparing model performance at a common threshold will be meaningless.

- An important point about ROC graphs is that they measure the ability of a classifier to produce good *relative* instance scores.

Outline

Class skew

- ROC curves have an attractive property: they are insensitive to changes in class distribution.
- Any performance metric that uses values from both columns of the confusion matrix will be inherently sensitive to class skews. Metrics such as accuracy, precision, lift and F scores use values from both columns of the confusion matrix.
- ROC graphs are based upon TP rate and FP rate, in which each dimension is a strict columnar ratio, so do not depend on class distributions.

Outline

Creating scoring classifiers

- Even if a classifier only produces a class label, an aggregation of them may be used to generate a score. MetaCost (Domingos, 1999) employs bagging to generate an ensemble of discrete classifiers, each of which produces a vote. The set of votes could be used to generate a score.

- Finally, some combination of scoring and voting can be employed. For example, rules can be provide basic probability estimates, which may then be used in weighted voting.

Efficient generation of ROC curves

- Given a test set, we often want to generate an ROC curve efficiently from it.
- Exploiting the monotonicity of thresholded classifications a much better algorithm can be created: any instance that is classified positive with respect to a given threshold will be classified positive for all lower thresholds as well. Therefore, we can simply sort the test instances by f scores, from highest to lowest, and move down the list, processing one instance at a time and updating TP and FP as we go. In this way we can create an ROC graph from a linear scan.
- Let n be the number of points in the test set. This algorithm requires an O(n log n) sort followed by an O(n) scan down the list, resulting in O(n log n) total complexity.

Practical method for calculating an ROC curve from a test set

Algorithm 2 Practical method for calculating an ROC curve from a test set

Inputs: L, the set of test instances; $f(i)$, the probabilistic classifier's estimate that instance i is positive.
Outputs: R, a list of ROC points from (0,0) to (1,1)

```
1: L_sorted ⇐ L sorted decreasing by f scores
2: FP ⇐ 0
3: TP ⇐ 0
4: R ⇐ ⟨⟩
5: f_prev ⇐ −∞
6: for i ∈ L_sorted do
7:    if f(i) ≠ f_prev then
8:       ADD_POINT((FP/N, TP/P), R)
9:       f_prev ⇐ f(i)
10:   if i is a positive example then
11:      TP ⇐ TP + 1
12:   else                          /* i is a negative example, so this is a false positive */
13:      FP ⇐ FP + 1
14: ADD_POINT((FP/N, TP/P), R)
15: end
```

```
1: subroutine ADD_POINT(P, R)
2: push P onto R
3: end subroutine
```

Figure: Calculating a ROC curve.

Outline

Equally scored instances

- Steps 7-9 of algorithm 2 are necessary in order to correctly handle sequences of equally scored instances.
- The sort in line 1 of algorithm 2 does not impose any specific ordering on these instances since their f scores are equal.
- We want the ROC curve to represent the *expected* performance of the classifier, which, lacking any other information, is the average of the pessimistic and optimistic.

Outline

Creating convex ROC curves

Algorithm 3 Modifications to algorithm 2 to avoid introducing concavities.

```
1: subroutine ADD_POINT(P, R)
2: loop
3:    if |R| < 2 then
4:       push P onto R
5:       return
6:    else
7:       T ⇐ pop(R)
8:       T2 ⇐ top_of_stack(R)
9:       if SLOPE(T2, T) < SLOPE(T, P) then
10:         push T onto R
11:         push P onto R
12:         return
13: end subroutine
```

Figure: Modications to algorithm 2 to avoid introducing concavities

Area under an ROC Curve (AUC)

- An ROC curve is a 2D depiction of classifier performance.
- To compare classifiers we may want to reduce ROC performance to a single scalar value representing expected performance. A common method is to calculate the area under the ROC curve (AUC).
- Since AUC is a portion of the area of the unit square, its value will always be between 0 and 1.0.
- No realistic classifier should have an AUC less than 0.5.

Area under an ROC Curve (AUC)

- IMPORTANT: AUC of a classifier is equivalent to the probability that the classifier will rank a randomly chose positive instance higher than a randomly chosen negative instance.
- The AUC is also closely related to the Gini index: $Gini + 1 = 2 \times AUC$
- In practice the AUC performs very well and is often used when a general measure of predictiveness is desired.

Calculating the AUC

Algorithm 4 Calculating the area under an ROC curve

Inputs: L, the set of test instances; $f(i)$, the probabilistic classifier's estimate that instance i is positive.
Outputs: A, the area under the ROC curve.

```
1:  L_sorted ⇐ L sorted decreasing by f scores
2:  FP ⇐ TP ⇐ 0
3:  FP_prev ⇐ TP_prev ⇐ 0
4:  A ⇐ 0
5:  f_prev ⇐ −∞
6:  for i ∈ L_sorted do
7:      if f(i) ≠ f_prev then
8:          A ⇐ A + TRAP_AREA(FP, FP_prev, TP, TP_prev)      /* See A.3 for TRAP_AREA */
9:          f_prev ⇐ f(i)
10:         FP_prev ⇐ FP
11:         TP_prev ⇐ TP
12:     if i is a positive example then
13:         TP ⇐ TP + 1
14:     else
15:         FP ⇐ FP + 1
16: A ⇐ A + TRAP_AREA(1, FP_prev, 1, TP_prev)
17: A ⇐ A/(P · N)                    /* scale from P × N onto the unit square */
18: end
```

Figure: Algorithm to calculate the AUC

Averaging ROC Curves

- Although ROC curves may be used to evaluate classifiers, care should be taken when using them to make conclusions about classifier superiority.
- Some researchers have assumed that an ROC graph may be used to select the best classifiers simply by graphing them in ROC space and seeing which ones dominate. This is misleading; it is analogous to taking the maximum of a set of accuracy figures from a single test set. Without a measure of variance we cannot easily compare the classifiers.
- Averaging ROC curves is easy if the original instances are available.
- This section presents two methods for averaging ROC curves: vertical and threshold averaging.

Outline

Vertical Averaging

- Vertical Averaging takes vertical samples of the ROC curves for fixed FP rates and averages the corresponding TP rates.
- Such averaging is appropriate when the FP rate can indeed be fixed or when a 1D measure of variation is desired.

Method of Vertical Averaging

- Each ROC curve is treated as a function, R_i, such that $TP = R_i(FP)$.
- This is done by choosing the maximum TP for each FP and interpolating between points when necessary.
- The averaged ROC curve is the function $\hat{R}(FP) = mean[R_i(FP)]$.
- To plot an average ROC curve we can sample from $\hat{R}$ at points regularly spaced along the FP -axis. Confidence intervals of the mean of TP are computed using the common assumption of a binomial distribution.

Algorithm for Vertical Averaging

Algorithm 5 Vertical averaging of ROC curves.

Inputs: *samples*, the number of FP samples; *nrocs*, the number of ROC curves to be sampled, $ROCS[nrocs]$, an array of *nrocs* ROC curves; $npts[m]$, the number of points in ROC curve m. Each ROC point is a structure of two members, FP and TP, whose values are referenced by subscripts here.

Output: Array $TPavg$, containing the vertical (TP) averages.

```
1: s ⇐ 1
2: for FP_sample = 0 to 1 by 1/samples do
3:    TPsum ⇐ 0
4:    for i = 1 to nrocs do
5:       TPsum ⇐ TPsum + TP_FOR_FP(FP_sample, ROCS[i], npts[i])
6:    TPavg[s] ⇐ TPsum/i
7:    s ⇐ s + 1
8: end
1: function TP_FOR_FP(FP_sample, ROC, npts)
2: i ⇐ 1
3: while i < npts and ROC[i + 1]_FP ≤ FP_sample do
4:    i ⇐ i + 1
5: if ROC[i]_FP = FP_sample then
6:    return ROC[i]_TP
7: else if ROC[i]_FP < FP_sample then
8:    return INTERPOLATE(ROC[i], ROC[i + 1], FP_sample)
9: end function
```

Figure: Vertical averaging of ROC curves

Outline

Threshold Averaging

- Vertical averaging has the advantage that averages are made of a single dependent varaible (FP-rate). This simplifies computing confidence intervals. However FP-rate is often not under the direct control of the researcher.
- Threshold averaging, instead of sampling points based on their positions in ROC space, as vertical averaging does, it samples based on the thresholds that produced these points.
- It generates an array T of classifier scores which are sorted from largest to smallest and used as the set of thresholds. These thresholds are sampled at fixed intervals determined by the number of samples desired. For a given threshold, the algorithm selects from each ROC curve the the point of greatest score less than or equal to the threshold. These points are then averaged separately along their X and Y axes, with the center point returned in the *Avg* array.

Algorithm for Threshold Averaging

Algorithm 6 Threshold averaging of ROC curves.

Inputs: *samples*, the number of threshold samples; *nrocs*, the number of ROC curves to be sampled; $ROCS[nrocs]$, an array of *nrocs* ROC curves; $npts[m]$, the number of points in ROC curve m. Each ROC point is a structure of three members, FP, TP and Score, whose values are referenced by subscripts here.
Output: *Avg*, an array of (X,Y) points constituting the average ROC curve.

1: $T \Leftarrow$ all Scores of all ROC points
2: sort T in descending order
3: $s \Leftarrow 1$
4: **for** $tidx = 1$ **to** $length(T)$ **by** $int(length(T)/samples)$ **do**
5: $\quad FPsum \Leftarrow 0$
6: $\quad TPsum \Leftarrow 0$
7: $\quad$ **for** $i = 1$ **to** nrocs **do**
8: $\quad\quad p \Leftarrow$ POINT_AT_THRESH$(ROCS[i], npts[i], T[tidx])$
9: $\quad\quad FPsum \Leftarrow FPsum + p_{FP}$
10: $\quad\quad TPsum \Leftarrow TPsum + p_{TP}$
11: $\quad Avg[s] \Leftarrow (FPsum/i\ , TPsum/i)$
12: $\quad s \Leftarrow s + 1$
13: **end**

1: **function** POINT_AT_THRESH$(ROC, npts, thresh)$
2: $i \Leftarrow 1$
3: **while** $i < npts$ **and** $ROC[i]_{Score} > thresh$ **do**
4: $\quad i \Leftarrow i + 1$
5: **return** $ROC[i]$
6: **end function**

Figure: Threshold averaging or ROC curves

Outline

The ROC convex hull

- One advantage of ROC graphs is that they enable visualizing and organizing classifier performance without regard to class distributions or error costs.
- A researcher can graph the performance of a set of classifiers, and that graph will remain invariant with respect to the operating conditions (class skews and error costs).
- A set of operating conditions may be transformed into a so-called *iso-performance line* in ROC space. All classifiers corresponding to points on a line of slope m have the same expected cost.
- Generally a classifier is potentially optimal if and only if it lies on the convex hull of the set points in ROC space, the *ROC convex hull* (ROCCH).
- The operating conditions of the classifier may be translated into an iso-performance line, which in turn may be used to identify a portion of the ROCCH. As conditions change, the hull itself does not change;

Outline

When we have more than two classes

- The two axes represent trade-offs between errors (false positives) and benefits (true positives) that a classifier makes between two classes. Much of the analysis is straight-forward because of the symmetry that exists in the two-class problem.

Multi-class ROC graphs

- With n classes the confusion matrix becomes an $n \times n$ matrix containing the n correct classifications and $n^2 - n$ possible errors. Instead of managing trade-offs between TP and FP, we have n benefits and $n^2 - n$ errors.
- One method is to produce n different ROC graphs, one for each class: the **class reference** formulation.
- Specifically, if C is the set of all classes, ROC graph i plots the classication performance using class c_i as the positive class and all other classes as the negative class.
- This formulation compromises one of the attractions of ROC graphs, namely that they are insensitive to class skew (see section 4.2). Because each N_i comprises the union of $n-1$ classes, changes in prevalence within these classes may alter the c_i's ROC graph. However, this method can work in practice and provide reasonable flexibility in evaluation.

Multi-class AUC

- The AUC is a measure of discriminability of a pair of classes.
- In a two-class problem AUC is a single scalar value, but a multi-class problem introduces the issue of combining multiple pairwise discriminability values.

Provost and Domingo's Multi-class AUC approach

- One approach is to generate each class reference ROC curve in turn, measure the AUC, then assuming the AUCs weighted by the reference class's prevalence in the data. More precisely:

$$AUC_{total} = \sum_{c_i \in C} AUC(c_i) \cdot p(c_i)$$

- The disadvantage is that the class reference ROC is sensitive to class distributions and error costs, so this formulation of AUC_{total} is as well.

Hand and Till Multi-class AUC approach

- They desired a measure that is insensitive to class distribution and error costs.
- The derivation is too detailed to summarize here, but it is based upon the fact that the AUC is equivalent to the probability that the classier will rank a randomly chosen positive instance higher than a randomly chosen negative instance. From this probabilistic form, they derive a formulation that measures the unweighted *pairwise* discriminability of classes.
- While Hand and Till's formulation is well justified and is insensitive to changes in class distribution, there is no easy way to visualize the surface whose area is being calculated.

Outline

Interpolating classifiers

Logically combining classifiers

Chaining classifiers

Outline

DET curves

- DET graphs are not so much an alternative to ROC curves as an alternative way of presenting them.
- There are two dierences:
  - DET graphs plot false negatives on the Y axis instead of true positives, so they plot one kind of error against another.
  - DET graphs are log scaled on both axes so that the area of the lower left part of the curve (which corresponds to the upper left portion of an ROC graph) is expanded.
- The log scaling of a DET graph gives the lower left region of the graph greater surface area and allows these classifiers with low false positive rates and/or low false negative rates to be compared more easily.

Cost curves

- On a cost curve, the X axis ranges from 0 to 1 and measures the proportion of positives in the distribution. The Y axis, also from 0 to 1, is the relative expected misclassification cost. A perfect classifier is a horizontal line from (0; 0) to (0; 1). Cost curves are a point-line dual of ROC curves: a point (i.e., a discrete classifier) in ROC space is represented by a line in cost space, with the line designating the relative expected cost of the classifier. For any X point, the corresponding Y points represent the expected costs of the classifiers. Thus, while in ROC space the convex hull contains the set of lowest-cost classifiers, in cost space the lower envelope represents this set.

Relative superiority graphs and the LC index

- LC index is a transformation of ROC curves that facilitates comparing classifiers by cost.
- Adams and Hand's method maps the ratio of error costs onto the interval (0,1). It then transforms a set of ROC curves into a set of parallel lines showing which classifier dominates at which region in the interval. An expert provides a sub-range of (0,1) within which the ratio is expected to fall, as well as a most likely value for the ratio. This serves to focus attention on the interval of interest.
- The relative superiority graphs may be seen as a binary version of cost curves, in which we are only interested in which classifier is superior. The LC index (for loss comparison) is thus a measure of confidence of superiority rather than of cost difference.

Conclusion

- ROC graphs are a very useful tool for visualizing and evaluating classifiers.

- They are able to provide a richer measure of classification performance than accuracy or error rate can, and they have advantages over other evaluation measures such as precision-recall graphs and lift curves.

Questions?

Thank you for your attention.