Computer Sciences Department

The Relationship Between Precision-Recall and ROC Curves

Jesse Davis
Mark Goadrich

Technical Report #1551

January 2006

UNIVERSITY OF
WISCONSIN
MADISON

The Relationship Between Precision-Recall and ROC Curves

Jesse Davis JDAVIS@CS.WISC.EDU
Mark Goadrich RICHM@CS.WISC.EDU
Department of Computer Sciences and Department of Biostatistics and Medical Informatics, University of Wisconsin-Madison, 1210 West Dayton Street, Madison, WI, 53706 USA

Abstract

Receiver Operator Characteristic (ROC) curves and Precision-Recall (PR) curves are commonly used to present results for binary decision problems in machine learning. When the class distribution is close to being uniform, ROC curves have many desirable properties. However, when dealing with a highly skewed dataset, PR curves give a more accurate picture of an algorithm's performance. We show that a deep connection exists between ROC space and PR space. We prove that a curve dominates in ROC space if and only if it dominates in PR space. An important corollary to this proof is the notion of an achievable PR curve, and we show an efficient algorithm for computing the achievable PR curve. While it cannot be called a convex hull, this curve has properties much like the convex hull in ROC space. Finally, we show that differences in the two types of curves are significant for algorithm design. For example, in PR space it is incorrect to linearly interpolate between point. Furthermore, an algorithm which optimizes the area under the ROC curve is not guaranteed to optimize the area under the PR curve.

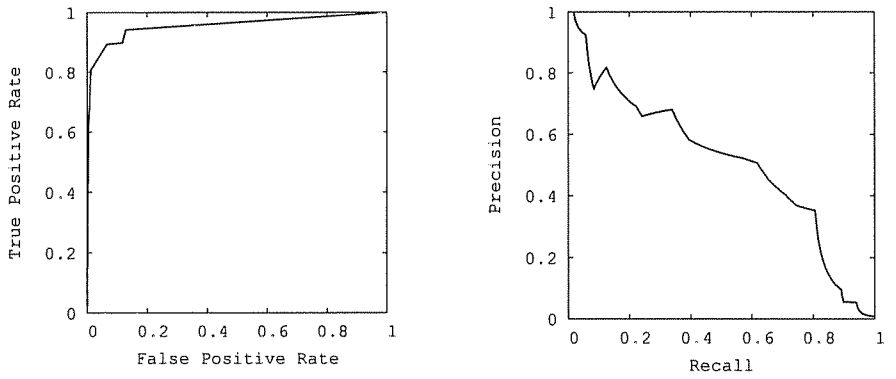
1. Introduction

In machine learning, current research has shifted away from simply presenting accuracy results when performing an empirical validation of new algorithms. This is especially true when evaluating algorithms that output probabilities of class values. Provost et al. have argued that simply using accuracy results can be misleading, as one can often use thresholds on the output of an algorithm to alter its performance (Provost et al., 1998). They recommended using Receiver Operator Characteristic (ROC) curves when evaluating binary decision problems; however, ROC curves can present an overly optimistic view of an algorithm's performance if there is a large skew in the class distribution. Drummond and Holte (Drummond & Holte, 2000; Drummond & Holte, 2004) have recommended using cost curves to address this issue, yet it is difficult to always quantify the misclassification cost for a given task.

Precision-Recall (PR) curves, often used in Information Retrieval (Manning & Schutze, 1999; Raghavan et al., 1989), have been cited as an alternative to ROC curves for tasks with a large skew in the class distribution (Bockhorst & Craven, 2005; Davis et al., 2005; Goadrich et al., 2004; Kok & Domingos, 2005; Singla & Domingos, 2005). An important difference between ROC space and PR space is the visual representation of the curves. For a sample ROC curve see Figure 1(a) and for a sample PR curve see Figure 1(b). These curves, taken from the same learned model on a highly-skewed biomedical information extraction dataset, highlight the visual difference between these spaces (Goadrich et al., 2004). The goal in ROC space is to be in the upper-left-hand corner, and when one looks at the ROC curve in Figure 1(a) it appears to be fairly close to optimal. In PR space the goal is to be in the upper-right-hand corner, and the PR curve in Figure1(b) shows that there is still significant room for improvement. Section 2 defines precision and recall for the reader unfamiliar with these terms.

Furthermore, looking at PR curves can accentuate differences between algorithms that are not apparent in ROC space. For example, let us consider a cancer detection task. Mammography is the only the proven method for early detection of Breast Cancer. Figure 2(a) shows the ROC curves for two different algorithms that look at a radiologist's interpretation of a mammogram and predict whether an abnormality is benign or malignant. The performances of the algo-

Technical Report from the University of Wisconsin - Madison Computer Science Department.



(a) Sample ROC curve

(b) Sample PR curve

Figure 1. The same curve shown in both ROC and PR space

rithms appear to be comparable in ROC space. However, Figure 2(b) shows the same curves in PR space. In PR space we can see that Algorithm 2 has a clear advantage over Algorithm 1 (Davis et al., 2005). This difference exists because in this domain the number of negative examples greatly exceeds the number positives instances. Consequently, a large change in the number of false positives can lead to a small change in the false positive rate used in ROC analysis. Precision, on the other hand, by comparing false positives to true positives rather than true negatives, captures the effect of the large number of negative examples has on the algorithms performance.

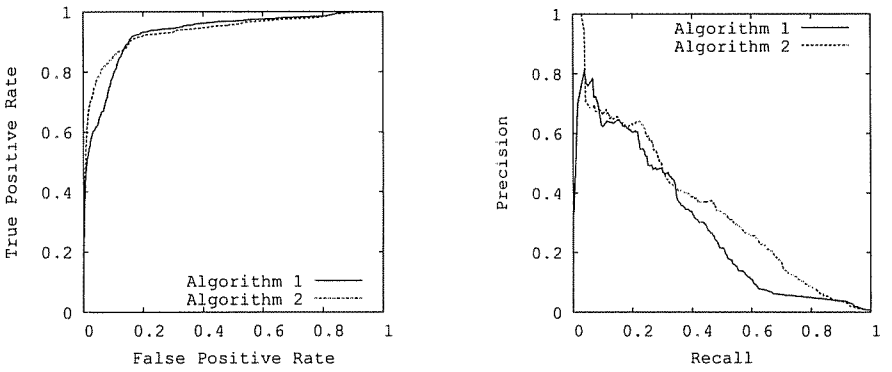
Finally, when comparing two curves it is important to note the curve with the best area under the curve in ROC space is not guaranteed to have the best area under the curve in PR space. Figure 3(a) shows ROC curves for two classifiers on a dataset. Figure 3(b) show PR curves for same two classifiers on the same dataset. In Figure 3(a), curve II has a larger area than curve I. However, in PR space (Figure 3(b)), curve I has a substantially larger area than curve II. This will be discussed in greater detail later.

We show that for any dataset, and hence a fixed number of positive and negative examples, the ROC curve and PR curve for a given algorithm contain the "same points." Hence PR curve I and PR curve II in Figure 3(b) are, in a sense that we formally define, equivalent to the ROC curve I and ROC curve II, respectively in Figure 3(a). Based on this equivalence for ROC and PR curves, we show that a curve dominates in ROC space if and only if it dominates in PR space. Second, we introduce the PR space analog to the convex hull in ROC space. We refer to the analog as the achievable PR curve. We show that due to the equivalence of these two spaces we can efficiently compute the achievable PR curve. Third we demonstrate that in PR space it is insufficient to linearly interpolate between points. Finally, we show that an algorithm that optimizes the area under the ROC curve is not guaranteed to optimize the area under the PR curve.

2. Review of ROC and Precision-Recall

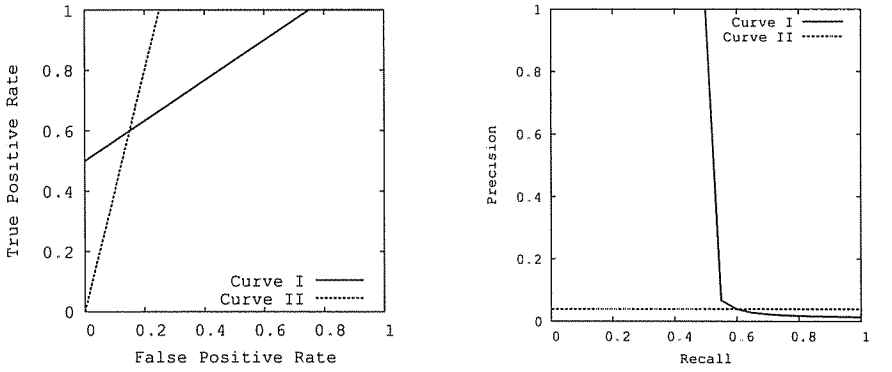
In a binary decision problem, a classifier labels examples as either positive or negative. The decision made by the classifier can be represented in a structure known as a confusion matrix or contingency table. The confusion matrix has four categories: True positives (TP) are examples correctly labeled as positives. False positives (FP) refer to negative examples incorrectly labeled as positive. True negatives (TN) correspond to negatives correctly labeled as negative. Finally, false negatives (FN) refer to positive examples incorrectly labeled as negative.

A confusion matrix is shown in Figure 4(a). The confusion matrix can be used to construct a point in either ROC space or PR space. Given the confusion matrix, we are able to define the metrics used in each space as in Figure 4(b). In ROC space, one plots the False Positive Rate (FPR) on the x-axis and the True Positive Rate (TPR) on the y-axis. The FPR measures the fraction of negative examples that are misclassified as positive. The TPR measures the fraction of positive examples that are correctly labeled. In PR space, one plots Recall on the x-axis and Precision on the y-axis. Recall is the same as TPR, whereas precision measures that fraction of examples classified as positive that are truly positive. Figure 4(b) gives the definitions for



(a) Comparison in ROC space (b) Comparison in PR space

Figure 2. The difference between comparing algorithms in ROC vs PR space



(a) Comparison in ROC space (b) Comparison in PR space

Figure 3. Difference in optimizing area under the curve in each space

each metric. We will treat the metrics as functions that act on the underlying confusion matrix that defines a point in either ROC space or PR space. Thus, given a confusion matrix A, $RECALL(A)$ returns the recall associated with A.

3. Relationship between ROC Space and PR Space

Despite the differences between ROC space and PR space previously discussed, there still exists a relationship between the two spaces.

Theorem 3.1. *For a fixed number of positive and negative examples, there exists a one-to-one correspondence between a curve in ROC space and a curve in PR space, such that the curves contain exactly the same confusion matrices.*

Proof. Note that a confusion matrix defines a unique point in ROC space. Since we ignore FN in PR space, one might worry that each point may correspond to multiple confusion matrices. However, with a fixed number of positive and negative examples, given the other three entries in a matrix, FN is uniquely determined. Consequently, we have a one-to-one mapping between confusion matrices and points in PR space. This implies that we also have a one-to-one mapping between points (each defined by a confusion matrix) in ROC space and PR space; hence, we can translate a curve in ROC space to PR space and vice-versa. □

One important definition we need for our next theorem is the notion that one curve dominates another curve. We use Provost et al.'s definition and say that curve I dominates curve II if curve II is always equal to or below curve I (Provost et al., 1998).

	actual positive	actual negative
predicted positive	TP	FP
predicted negative	FN	TN

(a) Confusion Matrix

Recall	=	$\frac{TP}{TP+FN}$
Precision	=	$\frac{TP}{TP+FP}$
True Positive Rate	=	$\frac{TP}{TP+FN}$
False Positive Rate	=	$\frac{FP}{FP+TN}$

(b) Definitions of metrics

Figure 4. Common machine learning evaluation metrics

Theorem 3.2. *For a fixed number of positive and negative examples, one curve dominates a second curve in ROC space if and only if the first dominates the second in Precision-Recall space.*

Proof.

Claim 1 ($\Rightarrow$): If a curve dominates in ROC space then it dominates in PR space. Proof by contradiction. Suppose we have curve I and curve II, such that curve I dominates in ROC space, yet, once we translate these curves in PR space, curve I no longer dominates. Since curve I does not dominate in PR space, there exists some point A on curve II such that the point B on curve I with identical recall has lower precision. In other words, $PRECISION(A) > PRECISION(B)$ yet $RECALL(A) = RECALL(B)$. Since $RECALL(A) = RECALL(B)$ and Recall is identical to TPR, we have that $TPR(A) = TPR(B)$. Since curve I dominates in curve II in ROC space $FPR(A) \geq FPR(B)$. Remember that total positives and total negatives are fixed and since $TPR(A) = TPR(B)$:

$$TPR(A) = \frac{TP_A}{\text{Total Positives}}$$
$$TPR(B) = \frac{TP_B}{\text{Total Positives}}$$

We now have $TP_A = TP_B$ and thus denote both as TP. Since $FPR(A) \geq FPR(B)$ and

$$FPR(A) = \frac{FP_A}{\text{Total Negatives}}$$
$$FPR(B) = \frac{FP_B}{\text{Total Negatives}}$$

This implies that $FB_A \leq FP_B$ and now we see that

$$PRECISION(A) = \frac{TP}{FP_A + TP}$$
$$PRECISION(B) = \frac{TP}{FP_B + TP}$$

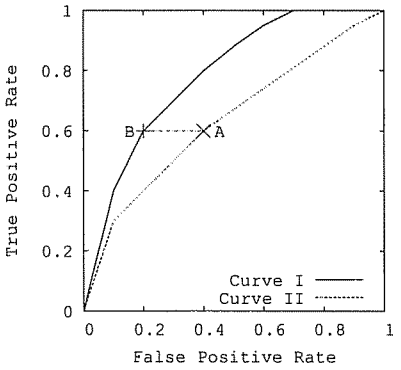
We now have that $PRECISION(A) \leq PRECISION(B)$. But this contradicts our original assumption that $PRECISION(A) > PRECISION(B)$.

Claim 2 ($\Leftarrow$): If a curve dominates in PR space then it dominates in ROC space. Proof by contradiction. Suppose we have curve I and curve II such that in that curve I in PR space, but once translated in ROC space curve I no longer dominates. Since curve I does not dominate in ROC space, there exists some point A on curve II such that the point B on curve I with identical TPR yet $FPR(A) < TPR(B)$. Since $RECALL$ and TPR are the same, we get that $RECALL(A) = RECALL(B)$. Because curve I dominates in PR space we know that $PRECISION(A) \leq PRECISION(B)$, as illustrated in Figure 6(a). Since $RECALL(A) = RECALL(B)$ and

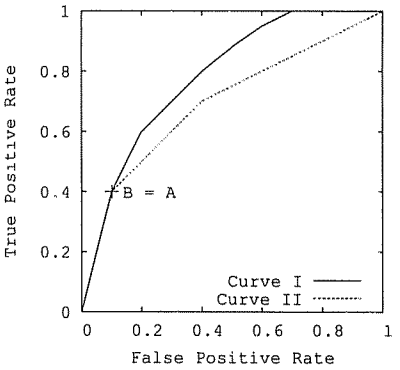
$$RECALL(A) = \frac{TP_A}{\text{Total Positives}}$$
$$RECALL(B) = \frac{TP_B}{\text{Total Positives}}$$

We know that $TP_A = TP_B$, so we will now denote them simply as TP. Because $PRECISION(A) \leq PRECISION(B)$ and

$$PRECISION(A) = \frac{TP}{TP + FP_A}$$
$$PRECISION(B) = \frac{TP}{TP + FP_B}$$

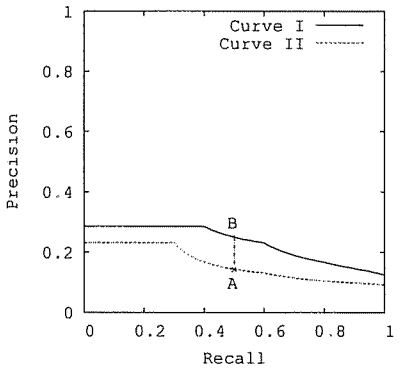


(a) Case 1: $FPR(A) > FPR(B)$

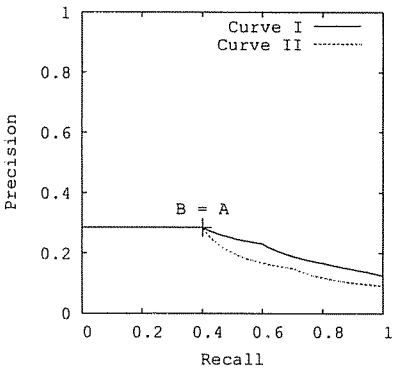


(b) Case 2: $FPR(A) = FPR(B)$

Figure 5. Two cases for Claim 1 of Theorem 3.2



(a) Case 1: $PRECISION(A) < PRECISION(B)$



(b) Case 2: $PRECISION(A) = PRECISION(B)$

Figure 6. Two cases of Claim 2 of Theorem 3.2

We get that $FP_A \geq FP_B$. Now we get that

$$FPR(A) = \frac{FP_A}{\text{Total Negatives}}$$

$$FPR(B) = \frac{FP_B}{\text{Total Negatives}}$$

This implies that $FPR(A) \geq FPR(B)$ and this contradicts our original assumption that $FPR(A) < FPR(B)$. □

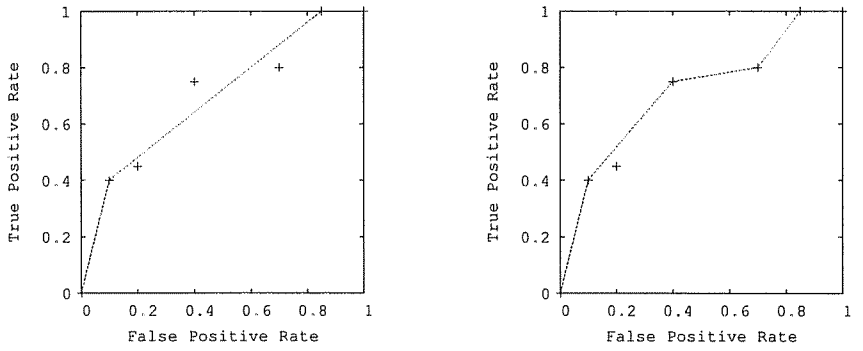
An interesting question to ask is if the convex hull in ROC space has an analog in PR space. The issue of dominance in ROC space is directly related to the convex hull. Given a set of points in ROC space, the convex hull must meet the following three criteria:

1. Linear interpolation is used between adjacent points.
2. No point lies above the final curve.
3. For any pair of points used to construct the curve, the line segment connecting them is equal to or below the curve.

Figure 7 shows several incorrect examples of constructing a convex hull. Figure 7(a) violates condition 2, whereas Figure 7(b) violates condition 3. For a detailed algorithm of how to efficiently construct the convex hull, see Cormen et al. (Cormen et al., 1990).

Corollary 3.1. *Given a set of points in PR space, there exists an analogous curve to the convex hull in ROC space, which we call the achievable PR curve.*

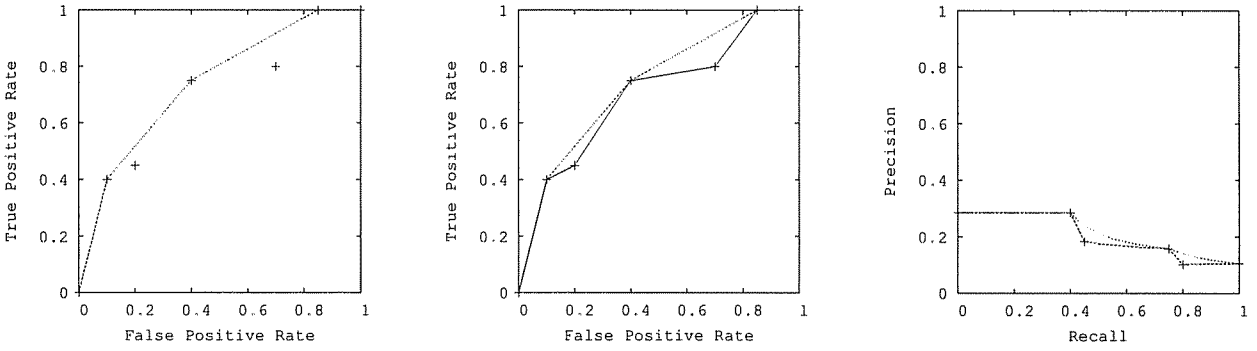
Proof. Figure 8(a) shows an example of a convex hull in ROC space. By definition, the convex hull dominates all other curves that could be constructed with



(a) Point above the curve

(b) Curve is not convex

Figure 7. Mistakes in constructing ROC convex hulls



(a) Convex hull in ROC space

(b) Curves in ROC space

(c) Equivalent curves in PR space

Figure 8. Convex hull and its PR analog dominate the naive method for curve construction in each space

those points when using linear interpolation between the points. Thus converting the points of the ROC convex hull into PR space will yield a curve that dominates in PR space as shown in Figure 8. This follows from Theorem 3.2. Interestingly the achievable PR curve will exclude exactly those points beneath the convex hull in ROC space. □

An important methodological issue must be addressed when building a convex hull in ROC space or an achievable curve in PR space. When constructing a ROC curve (or PR curve) from an algorithm that outputs a probability, the following approach is usually taken: first find the probability that each test set example is positive, next sort this list and then traverse the sorted list in descending order. To simplify the discussion, we will refer let $class(i)$ refer to the true classification of the example at position i in the array and $prob(i)$ refer to the probability that the example at position i is positive. For each i such that $class(i) \neq class(i+i)$ and $prob(i) < prob(i+i)$, create a classifier by calling every example j such that $j \geq i+1$ positive and all other examples negative.

Thus each point in ROC space or PR space represents a specific classifier, with a threshold for a calling an example positive. Building the convex hull can be seen as constructing a new classifier, as one picks the best points. Therefore it would be methodologically incorrect to construct a convex hull or achievable PR curve by looking at performance on the test data and then constructing a convex hull. To combat this problem, the convex hull must be constructed using a tuning a set as follows: First, use the method described above to find a candidate set of thresholds on the tuning data. Then, build a convex hull over the tuning

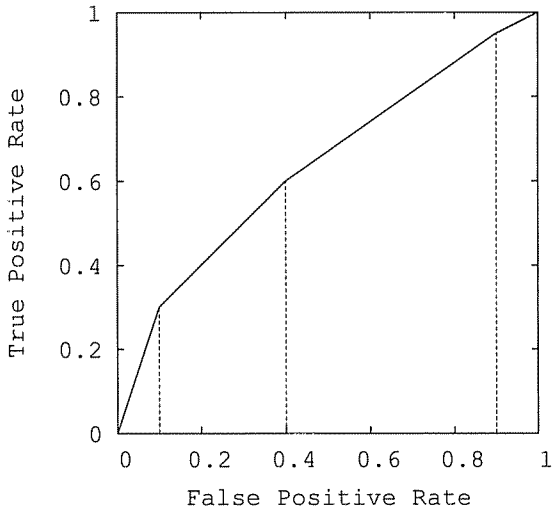


Figure 9. How to Calculate Area Under ROC Curve

data. Finally use the thresholds selected on the tuning data, when building an ROC or PR curve for the test data. While this test-data curve is not guaranteed to be a convex hull, it preserves the split between training data and testing data.

4. Interpolation and AUC

A key issue to address is how to interpolate between points in each space. It is straightforward to interpolate between points in ROC space by simply drawing a straight line connecting the two points. One can achieve any level of performance on this line by flipping a weighted coin to decide between the classifiers that the two end points represent. Often, the area under the curve is used as a simple metric to define how an algorithm performs over the whole space (Bradley, 1997; Davis et al., 2005; Goadrich et al., 2004; Kok & Domingos, 2005; Macskassy & Provost, 2005; Singla & Domingos, 2005). The area under the curve (AUC-ROC) can be calculated by using the polygonal areas created between each ROC point. For an example of how to create the polygons, see Figure 9.

However, in Precision-Recall space, interpolation is more complicated. As the level of recall varies, the precision does not necessarily change linearly due to a factor in the denominator of the precision metric. In these cases, linear interpolation is a mistake that yields an overly-optimistic estimate of performance. Goadrich et al. proposed the following method to approximate the interpolation between two points in PR space (Goadrich et al., 2004).

To construct the curve, we first standardize our precision-recall curves to always cover the full range of recall values and then interpolate between the points. From the first point, which we designate (R_{first}, P_{first}), the curve is extended horizontally to the point $(0, P_{first})$. This new point is achievable since we could randomly discard a fraction, f, of the extracted relations and expect the same precision but smaller recall on the remaining examples; the setting of f would determine the recall. An ending point of $(1, \frac{\text{Total Pos}}{\text{Total Pos + Total Neg}})$ can always be found by classifying everything as positive. This will give us a continuous curve extending from 0 to 1 along the recall dimension.

First, remember that any point A on a precision-recall curve is generated from the underlying true positive (TP_A) and false positive (FP_A) counts. Suppose we have two points, A and B which are far apart in precision-recall space. To find some intermediate values for our curve, we must interpolate between their counts TP_A and TP_B, and FP_A and FP_B. First, we find out how many negative examples it takes to equal one positive, or the local skew, defined by $\frac{FP_B - FP_A}{TP_B - TP_A}$. Now we can create new points with recall $TP_A + x$ for all integer values of x such that $0 \leq x \leq B - A$, i.e. $TP_A + 1, TP_A + 2, ..., TP_B - 1$, and calculate precision by linearly increasing the false positives for each new point by the local skew. Our resulting precision-recall points will be

$$\left(\frac{TP_A + x}{\text{Total Pos}}, \frac{TP_A + x}{TP_A + x + FP_A + \frac{FP_B - FP_A}{TP_B - TP_A} x} \right).$$

With these new points, we can now use the polygon method previously discussed to calculate the area under the curve (AUC-PR).

For example, suppose we have a dataset with 20 positive examples and 2000 negative examples. Let $TP_A = 5$, $FP_A = 5$, $TP_B = 10$, and $FP_B = 30$. Table 1 shows the proper interpolation of the intermediate points between A and B, with the local skew of 5 negatives for every 1 positive. Notice how the resulting precision interpolation is not linear between 0.50 and 0.25.

The graphical interpolation for the precision-recall curve is different than for an ROC curve; whereas the ROC interpolation would be a linear connection between the two points, in precision-recall space the connection can be curved, depending on the actual number of positive and negative examples covered by each point. The curve is especially pronounced when two points are far away in recall and precision and the local skew is high. Consider a curve (Figure 10) constructed from a single point of $(0.02, 1)$, and extended to the endpoints of $(0, 1)$ and $(1, 0.008)$ as described

Table 1. Correct interpolation between two points in PR space for a dataset with 20 positive and 2000 negative examples

	TP	FP	REC	PREC
A	5	5	0.25	0.500
.	6	10	0.30	0.375
.	7	15	0.35	0.318
.	8	20	0.40	0.286
.	9	25	0.45	0.265
B	10	30	0.50	0.250

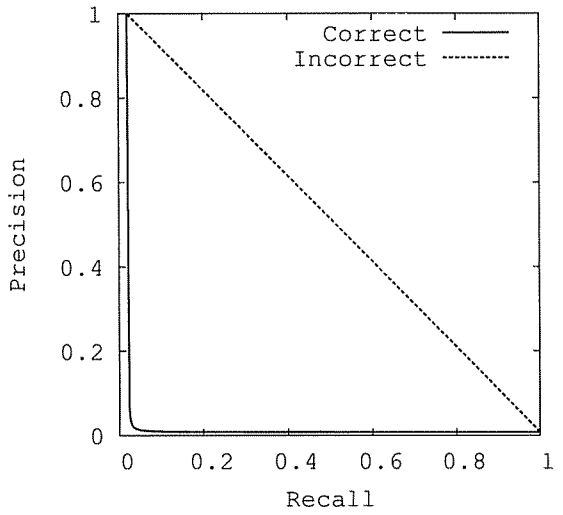


Figure 10. The effect of incorrect interpolation in PR space

above (for this example, our dataset contains 433 positives and 56,164 negatives). Interpolating as we have described would have an AUC-PR of 0.031; a linear connection would severely overestimate with an AUC-PR of 0.50.

In order to find the achievable curve in precision-recall space we merely have to find the convex hull in ROC space. Next, for each point selected by the algorithm to be included in the hull, we use the confusion matrix that defines that point to construct the corresponding point in PR space. Finally, we will have to perform the correct interpolation between the newly created PR points.

5. Optimizing Area Under the Curve.

Since the introduction of AUC-ROC as an evaluation metric, several researchers have investigated using AUC-ROC to inform the search heuristics of their algorithms. Ferri et al. (Ferri et al., 2002) alter decision trees to use the AUC-ROC as their splitting criterion, Cortes and Mohri (Cortes & Mohri, 2003) show that the boosting algorithm RankBoost (Freund et al., 1998), is also well-suited to optimize the AUC-ROC, Joachims (Joachims, 2005) presents a generalization of Support Vector Machines which can optimize AUC-ROC among other ranking metrics, Prati and Flach (Prati & Flach, 2005) use a rule selection algorithm to directly create the convex hull in ROC space, and both Yan et al. (Yan et al., 2003) and Herschtal and Raskutti (Herschtal & Raskutti, 2004) explore ways to optimize the AUC-ROC within neural networks. Also, ILP algorithms such as Aleph (Srinivasan, 2003) can be changed to use heuristics related to ROC or PR space, at least in relation to an individual rule.

Knowing that a convex hull in ROC space can be translated into the achievable curve in precision-recall space leads to another open question: do algorithms which optimize the AUC-ROC also optimize the AUC-PR? Unfortunately, the answer generally is no, and we prove this by the following counter-example. Figure 11 shows two overlapping curves in ROC space for a domain with 20 positive examples and 2000 negative examples, where each curve individually is a convex hull. The AUC-ROC for curve I is 0.813 and the AUC-ROC for curve II is 0.875, so an algorithm optimizing the AUC-ROC and choosing between these two rankings would choose curve II. However, Figure 12 shows the same curves translated into PR space, and the difference here is drastic. The AUC-PR for curve I is now 0.514 due to the high ranking of over half of the positive examples, while the AUC-PR for curve II is far less at 0.038, so the direct opposite choice of curve I should be made to optimize the AUC-PR.

In ROC space, the primary contribution to the area under the curve comes from the region with higher false positive rates and higher true positive rates. However, in PR space the main contribution comes from achieving a lower recall range with higher precision. Unfortunately, precision at low levels of recall can have a higher variance than high recall levels because of the variation in the denominator of precision. Nevertheless, based on Theorem 3.2 ROC curves are useful in an algorithm that optimizes AUC-PR. An algorithm can find the convex hull in ROC space, convert that curve to PR space for an achievable PR curve, and score the classifier by the area under this achievable PR curve.

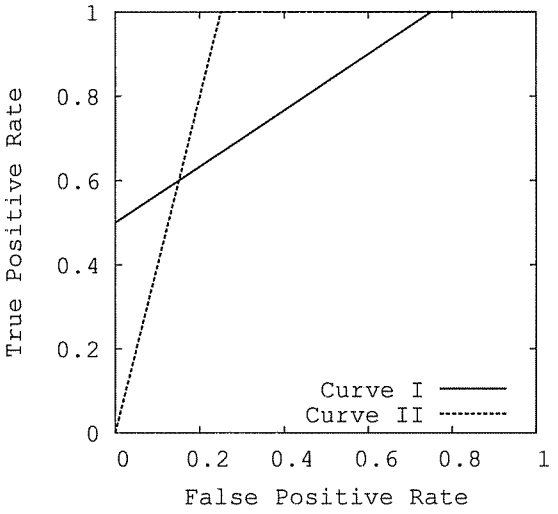


Figure 11. Comparing AUC-ROC for Two Algorithms

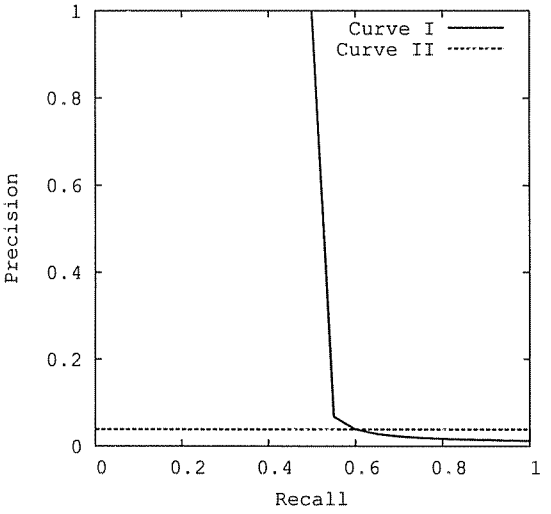


Figure 12. Comparing AUC-PR for Two Algorithms

6. Conclusions

This work makes four important contributions. First, for any dataset, the ROC curve and PR curve for a given algorithm contain the same points. This equivalence, leads to the surprising theorem that a curve dominates in ROC space if and only if it dominates in PR space. Second, as a corollary to the theorem we show the existence of the PR space analog to the convex hull in ROC space, which we call achievable PR curve. Remarkably, when constructing the achievable PR curve one discards exactly the same points omitted by the convex hull in ROC space. Consequently, we can efficiently compute the achievable PR curve. Third, we show that simple linear interpolation is insufficient between points in PR space. Finally, we show that an algorithm that optimizes the area under the ROC curve is not guaranteed to optimize the area under the PR curve.

7. Acknowledgements

A Java program for calculating the AUC-PR can be found at `http://www.cs.wisc.edu/~richm/programs/AUC/`
We gratefully acknowledge the funding from USA NLM Grant 5T15LM007359-02 and USA Air Force Grant F30602-01-2-0571, Vítor Santos Costa, and our advisors David Page and Jude Shavlik for their helpful comments and suggestions.

References

Bockhorst, J., & Craven, M. (2005). Markov networks for detecting overlapping elements in sequence data. *To appear in Advances in Neural Information Processing Systems 17 (NIPS)*. MIT Press.

Bradley, A. (1997). The Use of the Area Under the ROC Curve in the Evaluation of Machine Learning Algorithms. *Pattern Recognition, 30*, 1145–1159.

Cormen, T. H., Leiserson, Charles, E., & Rivest, R. L. (1990). *Introduction to algorithms*. MIT Press.

Cortes, C., & Mohri, M. (2003). AUC Optimization vs. Error Rate Minimization. *Neural Information Processing Systems (NIPS)*. MIT Press.

Davis, J., Burnside, E., Dutra, I., Page, D., Ramakrishnan, R., Costa, V. S., & Shavlik, J. (2005). View learning for statistical relational learning: With an application to mammography. *Proceeding of the 19th International Joint Conference on Artificial Intelligence*. Edinburgh, Scotland.

Drummond, C., & Holte, R. (2000). Explicitly representing expected cost: an alternative to roc representation. *Proceeding of Knowledge Discovery and Datamining* (pp. 198–207).

Drummond, C., & Holte, R. C. (2004). What roc curves can't do (and cost curves can). *ROCAI* (pp. 19–26).

Ferri, C., Flach, P., & Henrandez-Orallo, J. (2002). Learning decision trees using area under the ROC curve. *Proceedings of the 19th International Conference on Machine Learning* (pp. 139–146). Morgan Kaufmann.

Freund, Y., Iyer, R., Schapire, R., & Singer, Y. (1998). An efficient boosting algorithm for combining preferences. *Proceedings of the 15th International Conference on Machine Learning* (pp. 170–178). Madison, US: Morgan Kaufmann Publishers, San Francisco, US.

Goadrich, M., Oliphant, L., & Shavlik, J. (2004). Learning Ensembles of First-Order Clauses for Recall-Precision Curves: A Case Study in Biomedical Information Extraction. *Proceedings of the 14th International Conference on Inductive Logic Programming (ILP)*. Porto, Portugal.

Herschtal, A., & Raskutti, B. (2004). Optimising area under the ROC curve using gradient descent. *Proceedings of the 21st International Conference on Machine Learning* (p. 49). New York, NY, USA: ACM Press.

Joachims, T. (2005). A Support Vector Method for Multivariate Performance Measures. *Proceedings of the 22nd International Conference on Machine Learning*. ACM Press.

Kok, S., & Domingos, P. (2005). Learning the structure of Markov Logic Networks. *Proceedings of 22nd International Conference on Machine Learning* (pp. 441–448). ACM Press.

Macskassy, S., & Provost, F. (2005). Suspicion scoring based on guilt-by-association, collective inference, and focused data access. *International Conference on Intelligence Analysis*.

Manning, C., & Schutze, H. (1999). *Foundations of statistical natural language processing*. MIT Press.

Prati, R., & Flach, P. (2005). Roccer: an algorithm for rule learning based on roc analysis. *Proceeding of the 19th International Joint Conference on Artificial Intelligence*. Edinburgh, Scotland.

Provost, F., Fawcett, T., & Kohavi, R. (1998). The case against accuracy estimation for comparing induction algorithms. *Proceeding of the 15th International Conference on Machine Learning* (pp. 445–453). Morgan Kaufmann, San Francisco, CA.

Raghavan, V., Bollmann, P., & Jung, G. S. (1989). A critical investigation of recall and precision as measures of retrieval system performance. *ACM Trans. Inf. Syst.*, *7*, 205–229.

Singla, P., & Domingos, P. (2005). Discriminative training of Markov Logic Networks. *Proceedings of the 20th National Conference on Artificial Intelligene (AAAI)* (pp. 868–873). AAAI Press.

Srinivasan, A. (2003). The Aleph Manual Version 4. *http://web.comlab.ox.ac.uk/ oucl/ research/ areas/ machlearn/ Aleph/*.

Yan, L., Dodier, R., Mozer, M., & Wolniewicz, R. (2003). Optimizing classifier performance via the wilcoxon-mann-whitney statistics. *Proceedings of the 20th International Conference on Machine Learning*.