

# Análise ROC

Felipe Castro da Silva

`felipe@dpi.inpe.br`

São José dos Campos, 6 de Dezembro de 2006

## 1 Introdução

ROC<sup>1</sup> é uma técnica para visualizar, avaliar, organizar e selecionar classificadores baseado em suas performances. Para realizar estas análises, gráficos ROC podem mostrar o limiar entre taxas de acertos e alarmes falsos (taxas de erros) dos classificadores.

Um dos primeiros usos dos gráficos ROC em aprendizado por máquina foi desenvolvido por [Spackman, 1989], que demonstrou a utilidade das curvas ROC na avaliação e comparação entre algoritmos. A análise ROC também tem sido de grande utilidade para visualizar e analisar o comportamento de sistemas de diagnóstico [Swets, 1988], principalmente na medicina. Em [Vilarino et al., 2006] os gráficos ROC são utilizados para otimizar o limiar que classifica classes de contração ou não contração intestinal em imagens geradas através no exame de endoscopia. Atualmente há um grande crescimento no uso de gráficos ROC na comunidade de aprendizado por máquina, em parte pelo uso desta técnica na análise de classificadores.

Gráficos ROC são conceitualmente simples, mas existem algumas complexidades que surgem quando utilizados. Existem alguns conceitos vagos e dificuldades quando são usados na prática. Todas as definições descritas neste trabalho foram baseadas no trabalho de [Fawcett, 2005].

## 2 Performance dos Classificadores

Alguns classificadores geram como saída valores contínuos (*e.g.*, um classificador que gera a probabilidade de uma instância pertencer a uma dada classe), para os quais diferentes limiares podem ser aplicados para gerar diferentes conjuntos de saída (em um classificador contínuo pode-se aplicar um limiar para binarizá-lo, *i.e.* gerar duas classes). Outros, geram resultados discretos indicando somente a classe.

Considera-se um conjunto de amostras, onde uma instância<sup>2</sup>  $I$  pode assumir valores no conjunto  $p, n$ , *positive* e *negative* respectivamente. Tendo-se um classificador e uma instância pode-se ter 4

---

<sup>1</sup>Sigla para *Receiver Operating Characteristics*.

<sup>2</sup>Neste contexto, instância representa um elemento que não se sabe a qual classe pertence e deseja-se classificá-lo.

situações. Se a instância é *positive* e é classificada como *positive*, conta-se como *true positive*; se é classificada como *negative*, conta-se como *false negative*. Se a instância é *negative* e é classificada como tal, conta-se como *true negative*; se é classificada como *positive*, conta-se como *false positive*. Portanto, tendo um classificador e um conjunto de instâncias pode-se construir uma matriz de confusão de 2 por 2, no caso de 2 classes. Esta matriz serve como base para muitas métricas que podem ser aplicadas a classificação. A Figura 1 mostra a matriz de confusão.

		Classe Verdadeira	
		p	n
Classificada Como	p	<i>True Positive</i>	<i>False Positive</i>
	n	<i>False Negative</i>	<i>True Negative</i>
Totais		P	N

Figura 1: Matriz de Confusão gerada apartir de um conjunto de instâncias.

O conjunto de Equações 1 demonstra as métricas que podem ser calculadas a partir da matriz de confusão. Destacam-se as métricas *tp rate* (também chamado de taxa de acerto do produtor) e *fp rate* (também chamado de falso alarme do produtor) que servirão como base para a construção do espaço ROC. A métrica *precision* é a taxa de acerto do usuário, isto é, quantas instâncias de uma determinada classe o classificador acertou. A métrica *accuracy* representa a taxa de acerto de todo o classificador, isto é, a razão entre a soma dos acertos das 2 classes e o número total de instâncias.

$$\begin{aligned}
 tp\ rate &= \frac{TP}{P} & fp\ rate &= \frac{FP}{N} \\
 precision &= \frac{TP}{TP+FP} & recall &= tp\ rate = \frac{TP}{P} \\
 accuracy &= \frac{TP+TN}{P+N} & Fmeasure &= \frac{2(precision+recall)}{precision+recall+1}
 \end{aligned} \tag{1}$$

### 3 Gráficos ROC

Antes de definir como são construídos os gráficos ROC, tem-se que definir o espaço em que estas curvas serão representadas. Além disso, apresenta-se um algoritmo para gerar as curvas ROC.

#### 3.1 Espaço ROC

Os gráficos ROC são bidimensionais, onde no eixo Y plota-se o valor de *tp rate* e no eixo X o valor de *fp rate*. A Figura 2 mostra um gráfico ROC simples, somente com classificadores discretos. Classificadores discretos são aqueles que geram como saída somente uma classe. Estes classificadores fornecem um par (*fp rate*, *tp rate*) correspondendo a um ponto no espaço ROC.

Muitos pontos são importantes no espaço ROC. O ponto inferior esquerdo (0,0) representa uma estratégia que nunca gera uma classificação positiva; como um classificador que não comete

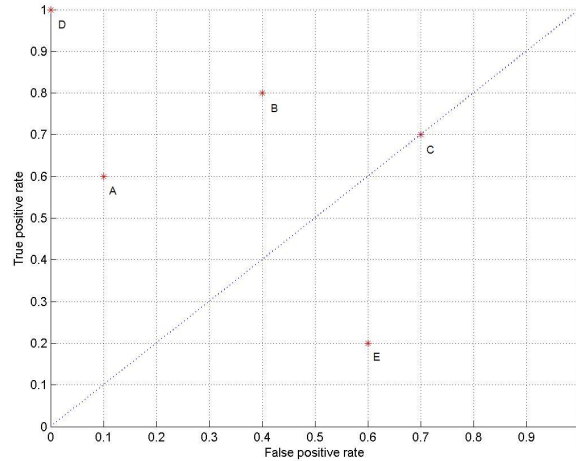


Figura 2: Gráfico ROC mostrando 5 classificadores discretos.  $A$  é dito um classificador “conservador”,  $B$  é o inverso de  $E$ ,  $D$  é um classificador perfeito e  $C$  é dito aleatório.

erros *false positives*, mas também não classifica nenhum *true positives*. A estratégia oposta, de incondicionalmente gerar *true positives* é representada pelo ponto superior direito  $(1, 1)$ .

O ponto  $(0, 1)$  representa uma classificação perfeita. Este caso pode ser representado pelo classificador  $D$ . De maneira geral, um ponto, no espaço ROC, é melhor que outro se ele está mais à noroeste (*tp rate* é maior e/ou *fp rate* é menor). Classificadores no lado esquerdo do gráfico ROC (perto do eixo Y) são ditos “conservadores”, pois fazem classificações positivas somente com uma evidência forte, portanto comentem poucos erros *false positives*. Classificadores no lado direito são ditos “liberais”, pois fazem classificações positivas com pouca evidência, mas comentem muitos erros *false positives*. Na Figura 2,  $A$  é mais conservador que  $B$ .

Qualquer classificador que está abaixo da linha diagonal que vai do ponto  $(0, 0)$  ao ponto  $(1, 1)$  é dito pior do que um classificador aleatório. Por isso, o triângulo inferior de um gráfico ROC está geralmente vazio. Pode-se ressaltar que se um classificador produz pontos abaixo da diagonal, pode-se negá-lo para produzir pontos acima dela. Na Figura 2,  $B$  é igual  $E$  negado.

### 3.2 Curvas no Espaço ROC

Muitos classificadores, tais como árvores de decisão ou conjunto de regras, são desenvolvidos para produzir uma única classe, *i.e.*  $p$  ou  $n$ , para cada instância. Portanto, um classificador discreto gera apenas um ponto no espaço ROC.

Alguns classificadores, tais como *Naive Bayes* ou redes neurais, naturalmente produzem uma probabilidade ou valor para cada instância que representa o grau de representatividade da classe sobre aquela instância. A partir destes classificadores, pode-se gerar classificadores discretos com uma simples aplicação de um limiar. Cada limiar produzirá um conjunto distinto de pontos no espaço ROC.

A Figura 3 mostra um exemplo de uma curva ROC produzida a partir de 20 instâncias descritas na Tabela 1.

Inst	Classe	Valor	Inst	Classe	Valor
1	p	0.9	11	p	0.4
2	p	0.8	12	n	0.39
3	n	0.7	13	p	0.38
4	p	0.6	14	n	0.37
5	p	0.55	15	n	0.36
6	p	0.54	16	n	0.35
7	n	0.53	17	p	0.34
8	n	0.52	18	n	0.33
9	p	0.51	19	p	0.30
10	n	0.505	20	n	0.1

Tabela 1: Instâncias geradas por um classificador probabilístico.

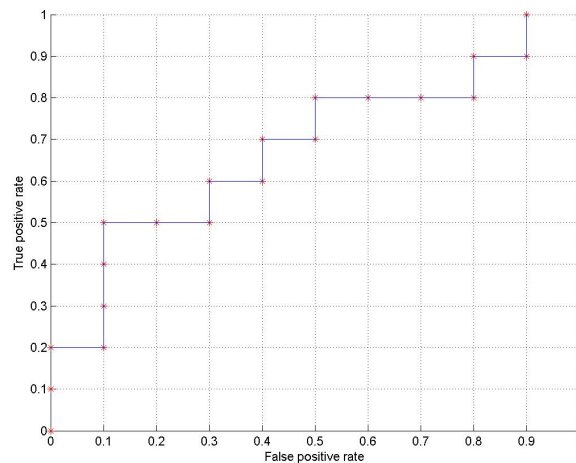


Figura 3: Uma curva ROC criada pela limiarização do conjunto de teste (Tabela 1).

Qualquer curva ROC que é gerada por um número finito de instância, produz uma “função degrau” no espaço ROC. Quanto maior o número de instâncias, mais contínua a curva fica. No Anexo 1 deste trabalho apresenta-se uma função para geração dos pontos no espaço ROC. Esta função foi adaptada para gerar curvas ROC e foi escrita no MatLab.

É importante ressaltar que cada ponto criado no tempo  $t$  do processo de geração da curva ROC, depende do ponto  $t - 1$ . Caso isso não seja considerado, a curva gerará uma nuvem de pontos e não ponto de uma curva.

## 4 Área Abaixo da Curva ROC

Uma curva ROC é uma demonstração bidimensional da performance de um classificador. Para comparar classificadores é preciso reduzir a curva ROC a um valor escalar. Um método comum para realizar esta redução é calcular a área abaixo da curva ROC (AUC<sup>3</sup>). Como a AUC é uma porção da área do quadrado unitário (espaço ROC) seus valores vão de 0.0 à 1.0. Entretanto, como classificadores piores que os aleatórios não são encontrados no espaço ROC, não existem classificadores com AUC menor que 0.5 (0.5 é a área de um classificador aleatório).

A Figura 4 mostra a área abaixo de duas curvas ROC, *A* e *B*. O classificador *B* possui uma área maior e, portanto, tem uma melhor performance média. É possível que em algumas regiões do espaço ROC um classificador seja melhor que outro. Na Figura 4 o classificador *B* é geralmente melhor que *A* exceto em  $fp\ rate > 0.6$  onde o *A* leva uma pequena vantagem. No Anexo 2 deste trabalho apresenta-se uma função para calcular AUC. Esta função foi escrita no MatLab e retorna um valor escalar correspondente a AUC.

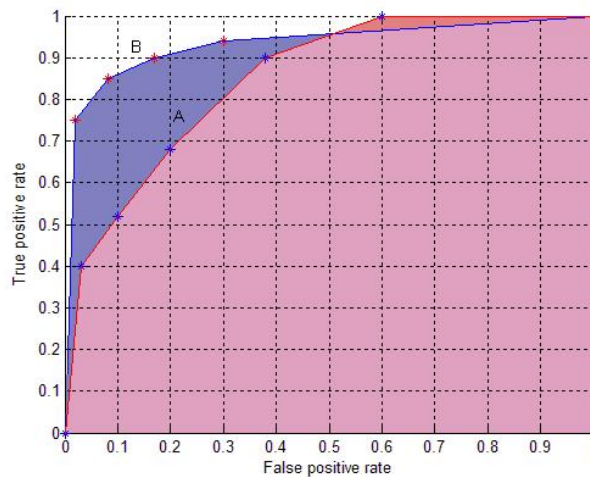


Figura 4: Área abaixo das curvas ROC *A* e *B*. Em média o classificador *B* é melhor que *A*, porém em alguns pontos *A* tem melhor performance que *B*.

## 5 Média entre Curvas ROC

Alguns pesquisadores assumem que um gráfico ROC pode ser usado para selecionar o melhor classificador, simplesmente plotando-os no espaço ROC e observando qual deles é dominante. Isto não é verdade. Sem uma medida de variância não é possível comparar os classificadores.

Fazer a média de curvas ROC é uma operação simples. Toda a média é feita no espaço unidimensional, porém as curvas ROC encontram-se no espaço ROC que, por sua vez é bidimensional.

---

<sup>3</sup>Sigla de *Area Under Curve*

Por isso, as curvas ROC podem ser projetadas em uma dimensão, sendo possível o cálculo da média. A questão é, qual a projeção é apropriada? Apresenta-se a seguir dois métodos para realizar a média entre curvas ROC.

O primeiro método é a média vertical das curvas ROC. Esse processo é muito simples, e consiste em tomar um passo constante no eixo *fp rate* e calcular a média, entre as curvas ROC, dos respectivos valores do eixo *tp rate*. Essa medida é adequada quando o passo do eixo *fp rate* pode ser fixado, ou quando uma simples medida de variância é desejada. A Figura 5 mostra o cálculo da média vertical entre as curvas ROC *A* e *B*.

O segundo método é a média dos limiares. Ao invés da média ser baseada na posição dos pontos no espaço ROC, como é o caso da vertical, ela é baseada nas amostras geradas por diferentes limiares. Para cada limiar acha-se o ponto correspondente em cada curva ROC e assim faz-se a média desses pontos.

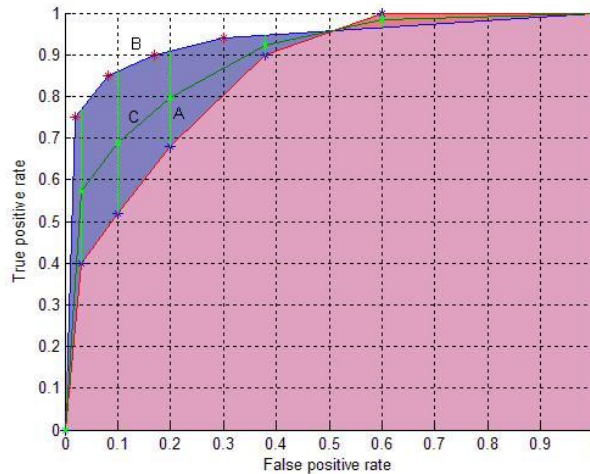


Figura 5: A curva ROC *C* representa a média vertical entre as curvas ROC *A* e *B*.

## 6 ROC com Múltiplas Classes

Até agora discutiu-se o problema da análise ROC sobre 2 classes. Na literatura, a maioria dos trabalhos são realizados utilizando somente 2 classes. Os dois eixos do espaço ROC representam fronteiras entre erros (*false positives*) e benefícios (*true positives*) que um classificador faz entre 2 classes.

Com  $n$  classes a matriz de confusão torna-se uma matriz de dimensão  $n \times n$ , contendo uma diagonal principal de tamanho  $n$  (acertos) e  $n^2 - n$  possíveis erros. Uma maneira para manipular  $n$  classes é produzir  $n$  gráficos ROC diferentes, um para cada classe. Se  $C$  é o conjunto de todas as classes, o gráfico ROC  $i$  plota a performance do classificador usando a classe  $c_i$  como a classe *positive* (Equação 2), e todas as outras classes como *negatives* (Equação 3).

$$P_i = c_i \quad (2)$$

$$N_i = \bigcup_{j \neq i} c_j \in C \quad (3)$$

A área abaixo da curva ROC também irá mudar quando temos  $n$  classes. Uma alternativa para o cálculo da área é apresentada por [Provost and Domingos, 2001]. Eles calcularam as AUCs para o problema de múltiplas classes gerando uma curva para cada classe, medindo a área abaixo da curva, e então somando as AUCs de todas as classes, dando peso maior para a classe que prevalece sobre as outras (Equação 4).

$$AUC_{total} = \sum_{c_i \in C} AUC(c_i)p(c_i) \quad (4)$$

onde  $AUC(c_i)$  é a área abaixo da curva da classe  $c_i$ . Esta definição precisa apenas de  $|C|$  cálculos de AUC.

Em [Hand and Till, 2001] é apresentado uma simples generalização para o problema do cálculo da AUC para múltiplas classes. Eles desejavam uma medida que não dependesse da distribuição das classes e dos custos dos erros (Equação 5).

$$AUC_{total} = \frac{2}{|C|(|C| - 1)} \sum_{(c_i, c_j) \in C} AUC(c_i, c_j) \quad (5)$$

onde  $AUC(c_i, c_j)$  é a área abaixo da curva ROC das classes  $c_i$  e  $c_j$ .

Em [Everson and Fieldsend, 2006] é apresentado um novo e recente método para a análise multiclass.

## 7 Conclusão

Uma visão sobre o que é a análise ROC foi apresentada neste trabalho. Apesar de ser um método relativamente simples, a análise ROC é amplamente utilizada para visualizar, avaliar, organizar e selecionar classificadores baseado em suas performances.

Existe uma limitação no processo apresentado, o fato de trabalhar-se sempre com duas classes para avaliar os classificadores. Existem na literatura algumas propostas de realizar a análise multiclass. Algumas delas foram apresentadas e são baseadas em tomar-se pares de classes e depois combiná-las.

Outra possibilidade que pode-se aplicar, é a interpolação de classificadores. Isso pode ser necessário quando a performance desejada é obtida com a combinação de mais de um classificador. Em [van der Putten and van Someren, 2000] é apresentado um exemplo de como resolver este problema.

## Referências

- [Everson and Fieldsend, 2006] Everson, R. M. and Fieldsend, J. E. (2006). Multi-class roc analysis from a multi-objective optimisation perspective. *Pattern Recognition Letters*, 27:918–927.
- [Fawcett, 2005] Fawcett, T. (2005). An introduction to roc analysis. *Pattern Recognition Letters*, pages 861–874.
- [Hand and Till, 2001] Hand, D. J. and Till, R. J. (2001). A simple generalization of the area under the roc curve to multiple class classification problem. *Machine Learning*, 45(2):171–186.
- [Provost and Domingos, 2001] Provost, F. and Domingos, P. (2001). Well-trained: Pets: Improving probability estimation trees. *CeDER Working Paper*.
- [Spackman, 1989] Spackman, K. A. (1989). Signal detection theory: Valuable tools for evaluating inductive learning. *Proc. Sixth Internat. Workshop on Machine Learning*, pages 160–163.
- [Swets, 1988] Swets, J. (1988). Measuring the accuracy of diagnostic systems. *Science*, 240:1285–1293.
- [van der Putten and van Someren, 2000] van der Putten, P. and van Someren, M. (2000). Coil challenge 2000: The insurance company case. Technical report, Institute of Advanced Computer ScienceS, Universiteit van Leiden.
- [Vilarino et al., 2006] Vilarino, F., Kuncheva, L. I., and Radeva, P. (2006). Roc curves and video analysis optimization in intestinal capsule endoscopy. *Pattern Recognition Letters*, pages 875–881.



## Anexo 1

```
function graph = ROC_Graph(L_class, f)
tam_L_class = size(L_class);
% sort L_class and f
for i=1:tam_L_class(2)
    m = 0.0;
    m_index = i;
    for j=i:tam_L_class(2)
        if (f(j) > m)
            m = f(j);
            m_index = j;
        end;
    end;
    aux = f(m_index);
    f(m_index) = f(i);
    f(i) = aux;
    aux = L_class(m_index);
    L_class(m_index) = L_class(i);
    L_class(i) = aux;
end;
%count number of P and N
P = 0;
N = 0;
for i=1:tam_L_class(2)
    if (L_class(i) == 'p')
        P = P + 1;
    else
        N = N + 1;
    end;
end;
TP = 0;
FP = 0;
R_index = 0;
f_prev = -1;
i = 1;
while (i <= tam_L_class(2))
    if (~(f(i) == f_prev))
        R_index = R_index + 1;
        R(R_index, 1) = FP / N;
        R(R_index, 2) = TP / P;
        f_prev = f(i);
    end;
    if (L_class(i) == 'p')
        TP = TP + 1;
    else
        FP = FP + 1;
    end;
    i = i + 1;
end;
R_index = R_index + 1;
R(R_index, 1) = FP / N;
R(R_index, 2) = TP / P;
hold on;
plot(R(:, 1), R(:, 2), 'r');
plot(R(:, 1), R(:, 2));
```

## Anexo 2

```
function area = ROC_Area(L_class, f)
tam_L_class = size(L_class);
% sort L_class and f
for i=1:tam_L_class(2)
    m = 0.0;
    m_index = i;
    for j=i:tam_L_class(2)
        if (f(j) > m)
            m = f(j);
            m_index = j;
        end;
    end;
    aux = f(m_index);
    f(m_index) = f(i);
    f(i) = aux;
    aux = L_class(m_index);
    L_class(m_index) = L_class(i);
    L_class(i) = aux;
end;
%count number of P and N
P = 0;
N = 0;
for i=1:tam_L_class(2)
    if (L_class(i) == 'p')
        P = P + 1;
    else
        N = N + 1;
    end;
end;
TP = 0;
FP = 0;
TP_prev = 0;
FP_prev = 0;
f_prev = -1;
i = 1;
A = 0;
while (i <= tam_L_class(2))
    if (~(f(i) == f_prev))
        A = A +
            Trapezoid_Area(FP, FP_prev, TP, TP_prev);
        f_prev = f(i);
        FP_prev = FP;
        TP_prev = TP;
    end;
    if (L_class(i) == 'p')
        TP = TP + 1;
    else
        FP = FP + 1;
    end;
    i = i + 1;
end;
A = A + Trapezoid_Area(N, FP_prev, N, TP_prev);
A = A / (P * N);
area = A;
```