Reconhecimento de Padrões em Imagens
Critérios de Avaliação

Dainf - UTFPR

Leyza Baldo Dorini - Rodrigo Minetto

Como escolher o classificador

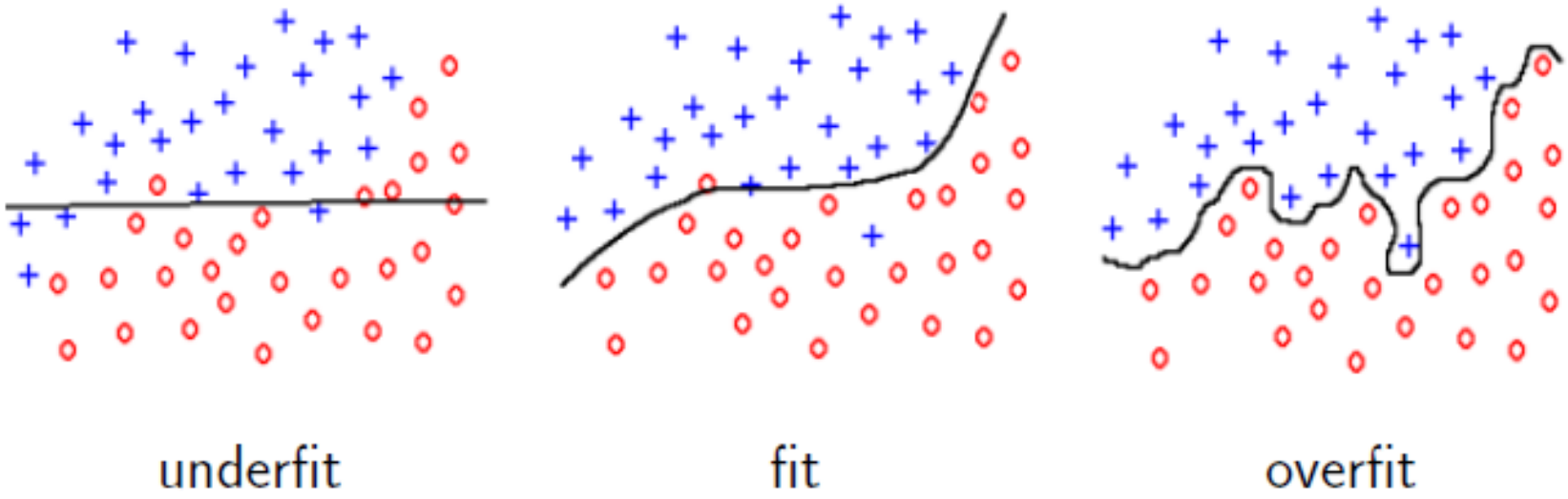
- Dependendo da característica do problema, alguns classificadores podem ser mais adequados. Por exemplo, SVM é mais apropriado do que kNN para alta dimensionalidade.
- Se todos os passos do algoritmo precisam estar explícitos, uma árvore de decisão pode ser mais apropriada que uma rede neural.
- De forma geral, diferentes abordagens são comparadas, bem como a mesma abordagem com diferentes parâmetros.

Avaliação de classificadores

Avaliação

Após o processo de treinamento, é preciso avaliar quão bem o classificador resultante consegue categorizar novas amostras.

Não se esqueça: aprender os dados de treinamento com muita precisão pode conduzir ao *overfitting*.



Com isso, a capacidade de generalização (ou seja, de classificação de novos dados) é comprometida.

Avaliação de classificadores

Classificadores são treinados em um conjunto finito de amostras, denominado **conjunto de treinamento**.

- Para verificar a capacidade de generalização do classificador, ele precisa ser testado experimentalmente com um **conjunto de testes**, o qual possui amostras diferentes daquelas presentes no conjunto de treinamento.
- São necessários critérios de avaliação que permitam comparar o desempenho de diferentes classificadores: estimar o grau de confiabilidade.
- Critérios alternativos podem ser considerados:
  - Tempo de treinamento.
  - Tempo de classificação.
  - Requisitos de armazenamento.
  - Robustez a ruídos, dados faltantes, rótulos incorretos, etc.
  - Interpretabilidade.

Dados: treinamento × testes

Problema: dados finitos estão disponíveis. Como decidir o que será usado para treinamento?

- Mais dados para treinamento conduzem a uma melhor generalização.
- Mais dados para testes resultam em uma melhor estimativa da probabilidade de erro.

Solução (?): particionamento dos dados (conjunto de treinamento e conjunto de testes).

- *Hold out.*
- *Cross validation.*
- *Bootstrap.*

O particionamento também pode ser útil para estimar parâmetros de modelos específicos (ex.: kNN).

Método *hold out*

Os dados são aleatoriamente particionados em dois conjuntos independentes:

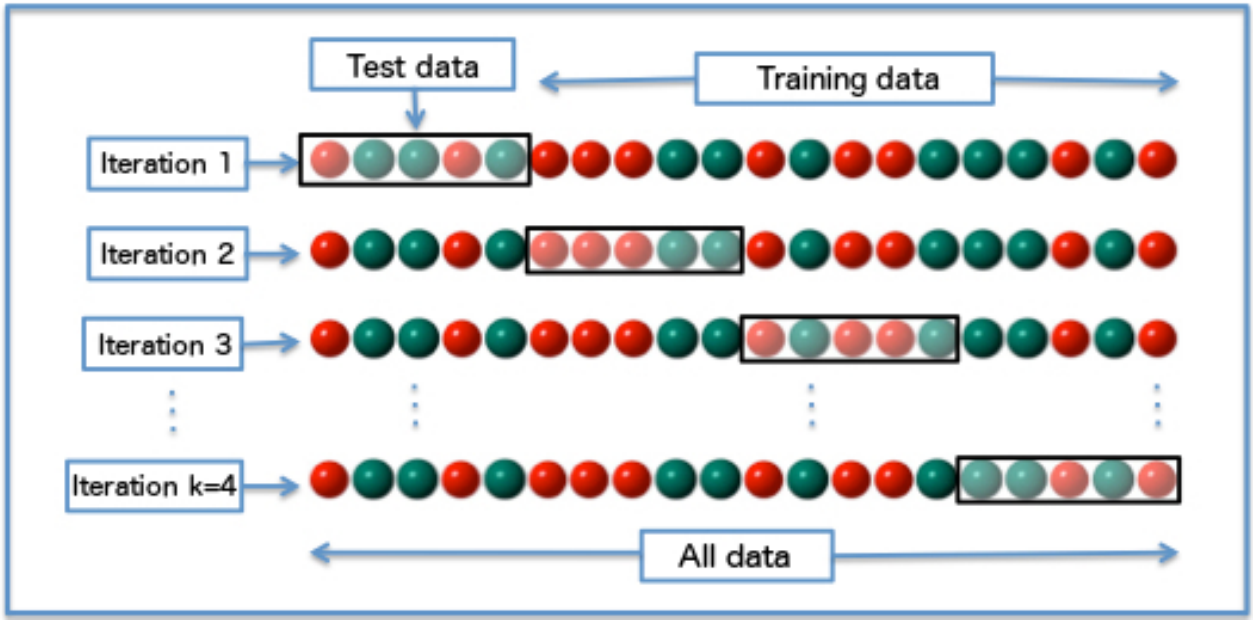
- Conjunto de treinamento: por exemplo, com 2/3 dos dados.
- Conjunto de testes: por exemplo, com o 1/3 restante dos dados.

Desvantagens: não possibilita prever o impacto de diferentes dados de treinamento.

O *Random Sampling* é uma variação, em que o método *hold out* é aplicado k vezes e a precisão média obtida (e variância) é considerada.

Método *K-fold cross validation*

- O conjunto de treinamento é dividido em K subconjuntos distintos, de mesmo tamanho e com aproximadamente a mesma distribuição de classes.



- O classificador é treinado K vezes, sendo que em cada uma delas um subconjunto é utilizado para validação. Erro: média dos K erros.
- Quanto maior o K, menor o bias (ie, menor a chance de não aprender relações relevantes entre características/classes) e maior a variância (ie, maior a chance de suposições erradas - *overfitting*).

Método *Leave-one-out*

É um caso especial do método *K-fold cross validation*, em que n experimentos são realizados utilizando $n - 1$ amostras para treinamento e a única amostra restante para teste.

- É caro computacionalmente. Mas seu uso se justifica para dados muito esparsos, por exemplo.
- Não garante a mesma distribuição entre as classes.
- Caso extremo: *50% class A, 50% class B. Predict majority class label in the training data. True error 50%; Leave-one-out error estimate 100%!*

Método *Bootstrap Aggregating (bagging)*

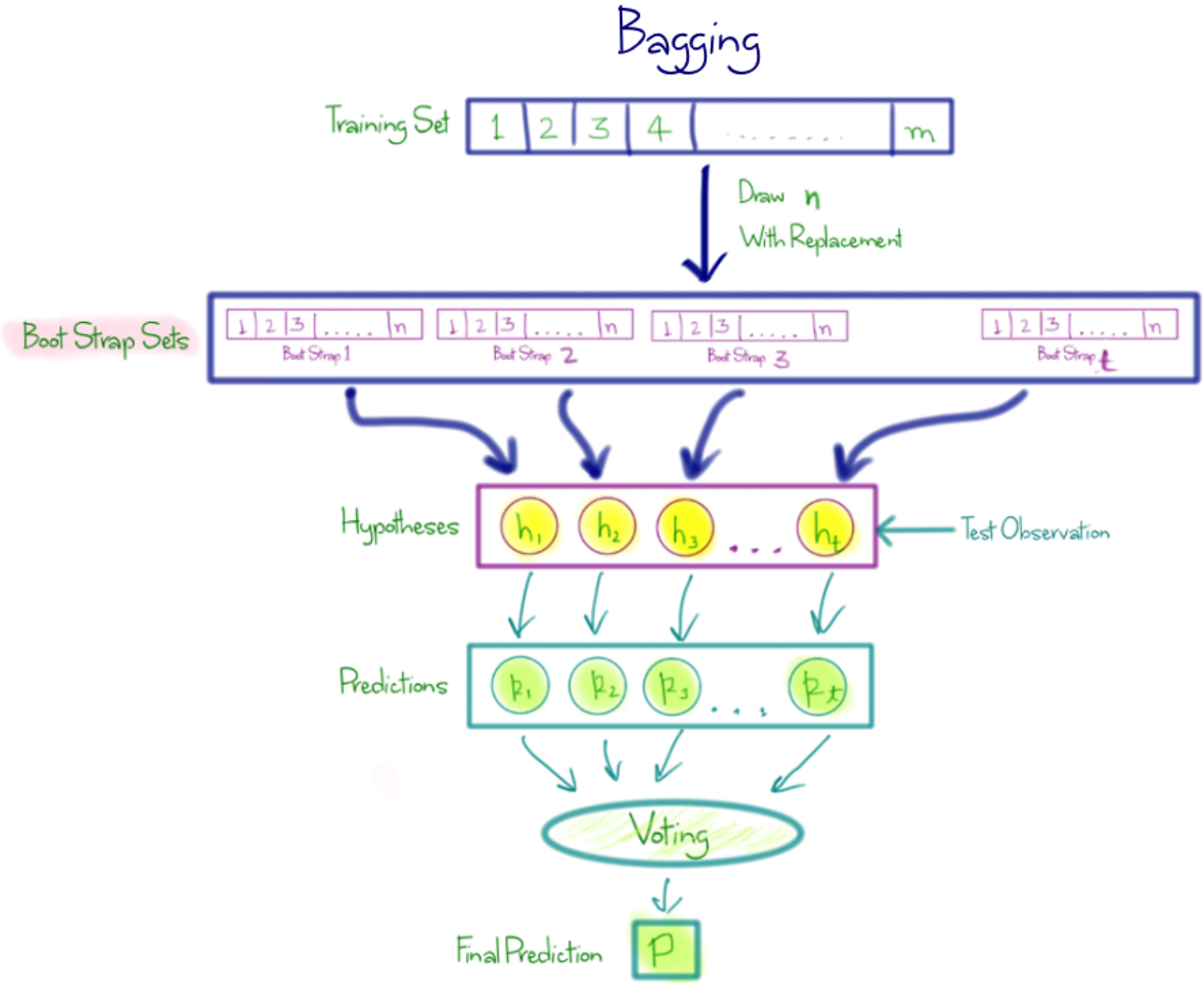
Utiliza amostragem com substituição para compor o conjunto de treinamento[1].

- Seja um conjunto de treinamento T com n amostras.
- Utilizando amostragem de T com substituição, o *bagging* gera m novos subconjuntos T_i, cada um com tamanho $n' < n$. Como consequência, algumas amostras aparecem repetidas em T_i [2].
- O processo de aprendizagem é realizado utilizando estes m subconjuntos e os modelos estatísticos são combinados (média ou votação).

---

[1] Ver Breiman, Leo (1996). Bagging predictors. Machine Learning.

[2] Para um n grande, quando $n' = n$, T_i possui aproximadamente 63.2% de amostras únicas (as restantes são duplicadas).


Bagging
Training Set
1
2
3
4
m
Draw n
With Replacement
Boot Strap Sets
Boot Strap 1
Boot Strap 2
Boot Strap 3
Boot Strap t
Hypotheses
Test Observation
Predictions
Voting
Final Prediction
P

Como lidar com dados não-balanceados?

Em diversos casos, as classes não possuem a mesma frequência de amostras. Por exemplo:

- Diagnóstico médico: 95% saudáveis.
- E-commerce: 92% não compram.
- Segurança: 99.99% das pessoas são não terroristas.

Uma alternativa consiste em construir bases de treinamento e teste balanceadas (mesma quantidade de amostras). Para que este processo não introduza um bias, deve-se considerar uma alteração na função de penalidade.

Critérios de validação: *error metrics*

A medida de erro é dada por:

$$err(\hat{f}) = \frac{1}{m} \sum_{i=1}^{m} I(y_i \neq \hat{f}(x_i)),$$

em que $\hat{f}(x_i)$ denota a predição do classificador $\hat{f}$ para a amostra x_i, m é a quantidade de amostras e y_i é o rótulo real. $I(k) = 1$ se $k == TRUE$ e 0 caso contrário.

Dela, deriva-se a medida de acurácia (*accuracy*):

$$acc(\hat{f}) = 1 - err(\hat{f}).$$

Critérios de validação: matriz de confusão

Em uma matriz de confusão M, a posição $M(i, j)$ indica a quantidade de amostras da classe i que são classificadas como pertencentes à classe j.

	class j predicted by a classifier										
true class i	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'	'R'
'0'	97	0	0	0	0	0	1	0	0	1	1
'1'	0	98	0	0	1	0	0	1	0	0	0
'2'	0	0	96	1	0	1	0	1	0	0	1
'3'	0	0	2	95	0	1	0	0	1	0	1
'4'	0	0	0	0	98	0	0	0	0	2	0
'5'	0	0	0	1	0	97	0	0	0	0	2
'6'	1	0	0	0	0	1	98	0	0	0	0
'7'	0	0	1	0	0	0	0	98	0	0	1
'8'	0	0	0	1	0	0	1	0	96	1	1
'9'	1	0	0	0	3	1	0	0	0	95	0

Exemplo: classificação de dígitos (R = *reject*).

Matriz de confusão

		predicted	
		negative	positive
actual examples	negative	a TN - True Negative correct rejections	b FP - False Positive false alarms type I error
	positive	c FN - False Negative misses, type II error overlooked danger	d TP - True Positive hits

- *Accuracy* = (a + d)/(a + b + c + d) = (TN + TP)/total
- *Error rate* = 1 - *accuracy*
- *Precision, predicted positive value* = d/(b + d) = TP/positivos preditos
- *True positive rate, recall, sensitivity* = d/(c + d) = TP/positivos reais
- *Specificity, true negative rate* = a/(a + b) = TN/negativos reais
- *False positive rate, false alarm* = b/(a + b) = FP/negativos reais = 1 - *specificity*
- *False negative rate* = c/(c + d) = FN/positivos reais
- $F\text{-measure} = 2\,\dfrac{precision \times recall}{precision + recall}$

Diferentes custos para classificações erradas

A *Accuracy*, que mede a porcentagem de classificações corretas, tem a desvantagem de considerar que todos os erros possuem o mesmo custo. Exemplos:

- Diagnóstico médico: o custo de indicar erroneamente um câncer em um paciente é menor do que o custo de não detectar uma pessoa doente.
- Defesa contra mísseis: o custo de não detectar um ataque real é muito maior do que um alarme falso.

O risco Bayesiano é capaz de representar estes diferentes custos.

Limitações com as medidas

As medidas anteriores podem não ser suficientes quando são necessárias outras informações:

- Como os erros estão distribuídos entre as classes?
- Como é o desempenho do classificador quando as condições de teste mudam (por exemplo, diferentes custos ou distribuições das amostras entre as classes).

Receiver Operating Characteristic (ROC)

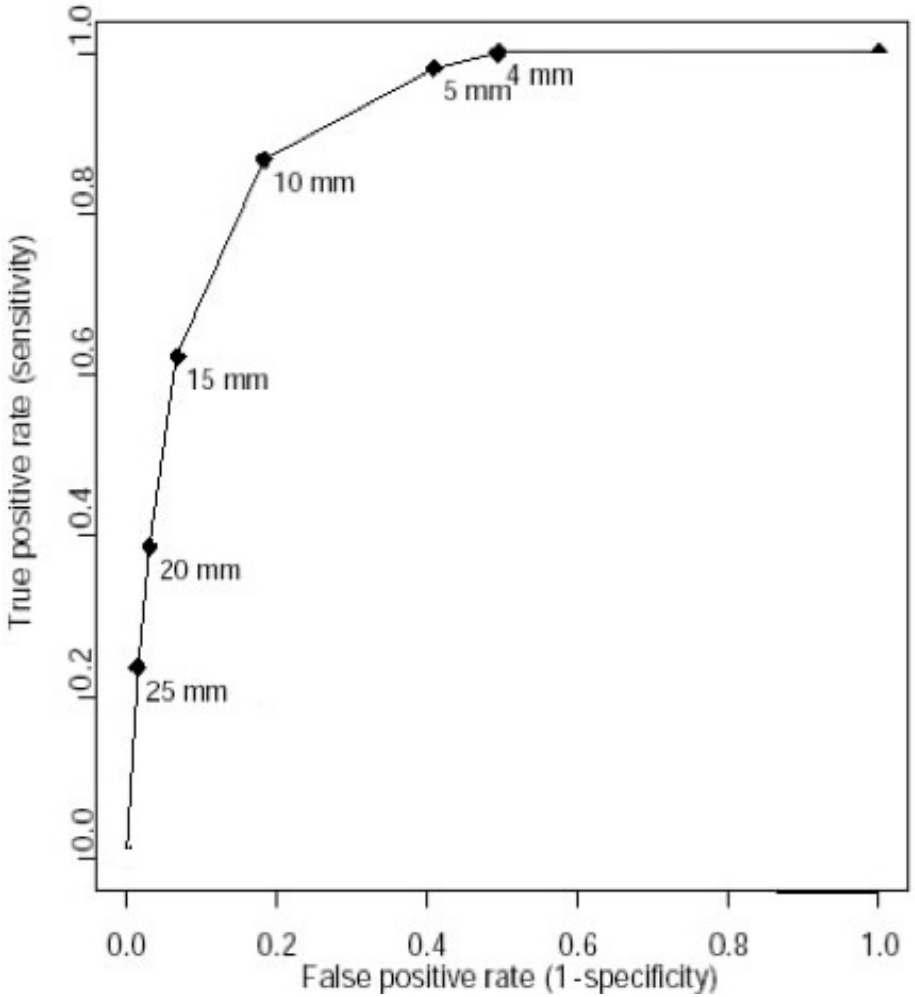
A combinação dos valores de *true positive rate* e *false positive rate* fornecem informações relevantes. Eles podem ser melhor visualizados em uma curva ROC, a qual fornece informações sobre:

- Desempenho para todos os possíveis custos de classificações incorretas.
- Desempenho para todas as possíveis taxas de distribuição entre classes.
- Em que condições o classificador c_1 supera o c_2?

A abordagem foi desenvolvida para classificadores binários! E para problemas com múltiplas classes?

Curvas ROC

Desta forma, é possível visualizar a relação entre a probabilidade de classificar corretamente os exemplos positivos contra a taxa de classificar incorretamente os exemplos negativos.



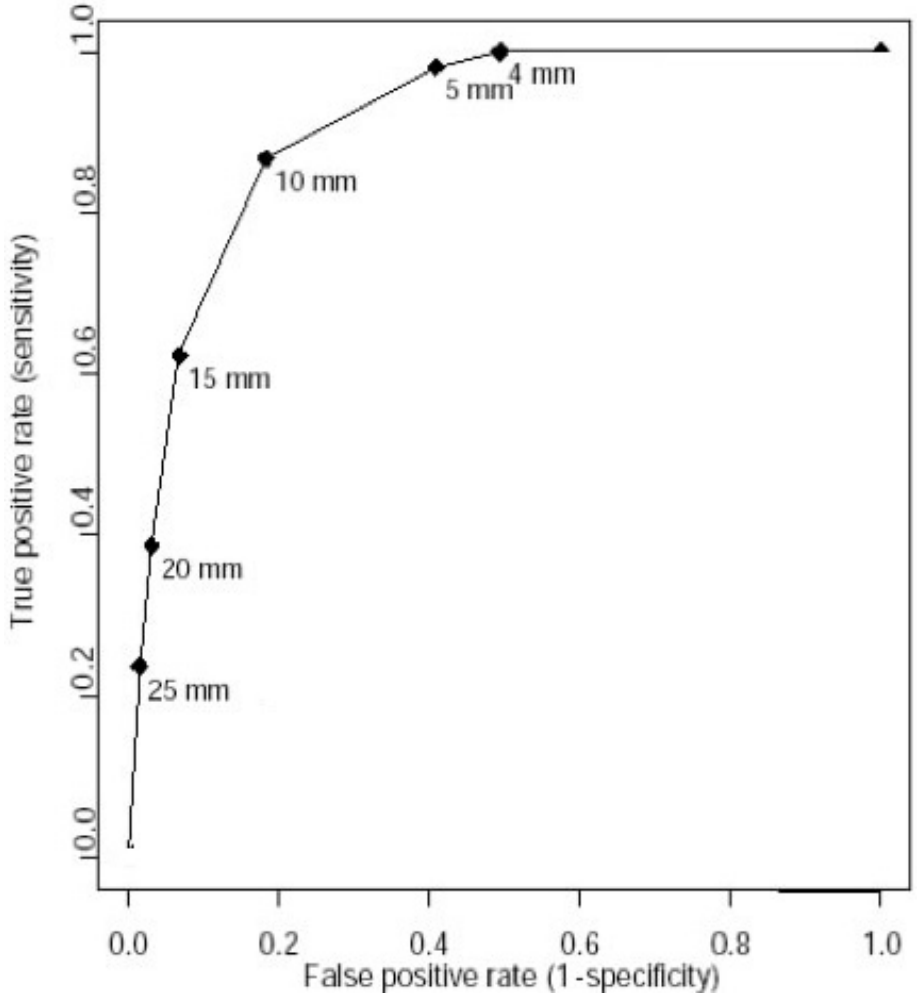
"One can interpret this curve as a comparison of the classifier across the entire range of class distributions and error costs."

Exemplo: ultrassom

Ultrassom endometrial: o exame de ultrassom pode ser utilizado para detectar o espessamento do revestimento do útero, o que pode ser um indicativo de câncer. Caso algo fora do normal seja detectado, indica-se uma biópsia ou mesmo procedimento cirúrgico. Ambos são invasivos e apresentam risco. A meta é maximizar a quantidade de *true positives* (câncer corretamente detectado) com um número aceitável de falsos positivos (*false alarms*).

Exemplo: ultrassom

Cutoff	*Sens.*	*Spec.*	*1-Spec.*
$>$ 4mm	99	50	50
$>$ 5mm	97	61	39
$>$ 10mm	83	80	20
$>$ 15mm	60	90	10
$>$ 20mm	40	95	5
$>$ 25mm	20	98	2



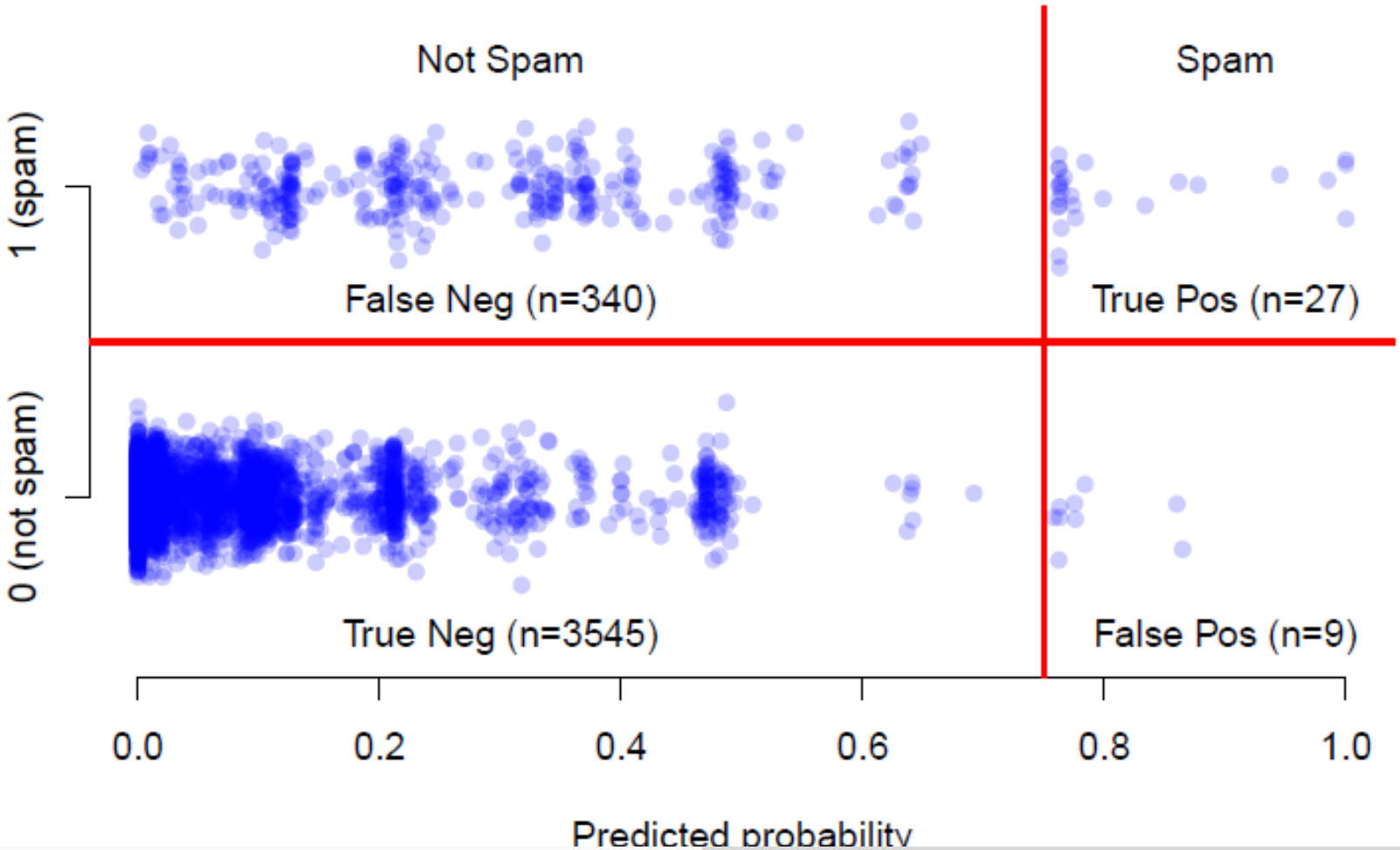
Sensitivity: TPR, ie TP/(FP+TP)

Specificity: TNR, ie TN/(TN+FN) [3]

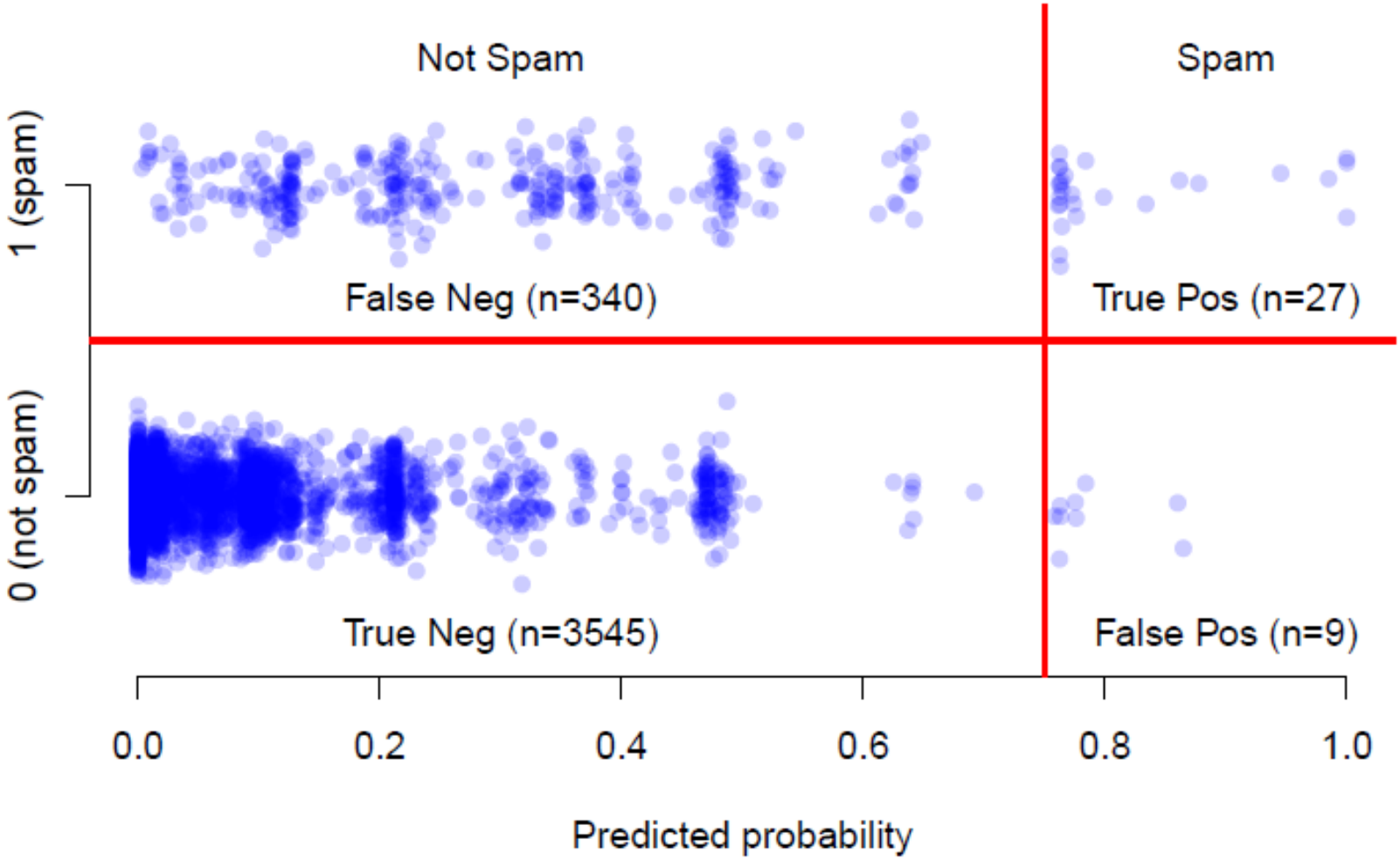
[3] Mais exemplos: http://ebp.uga.edu/courses/Chapter%204%20-%20Diagnosis%20I/8%20-%20ROC%20curves.html

Exemplo: spam

Suponha um modelo (por exemplo, baseado em regressão logística) que determine a probabilidade de um e-mail ser spam. Um limiar determina a partir de qual probabilidade considera-se spam. Para 0.75, por exemplo:



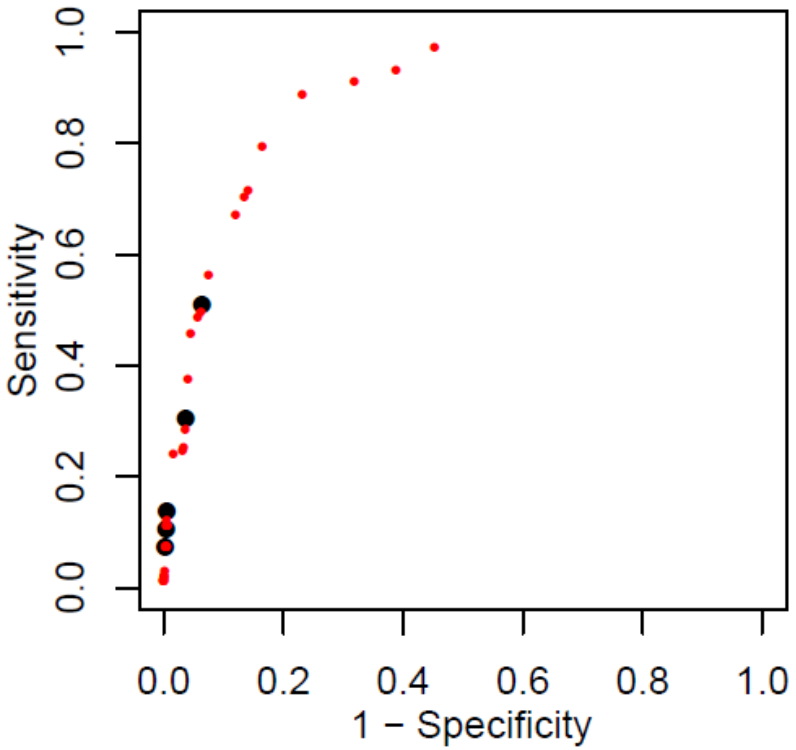
Exemplo: spam

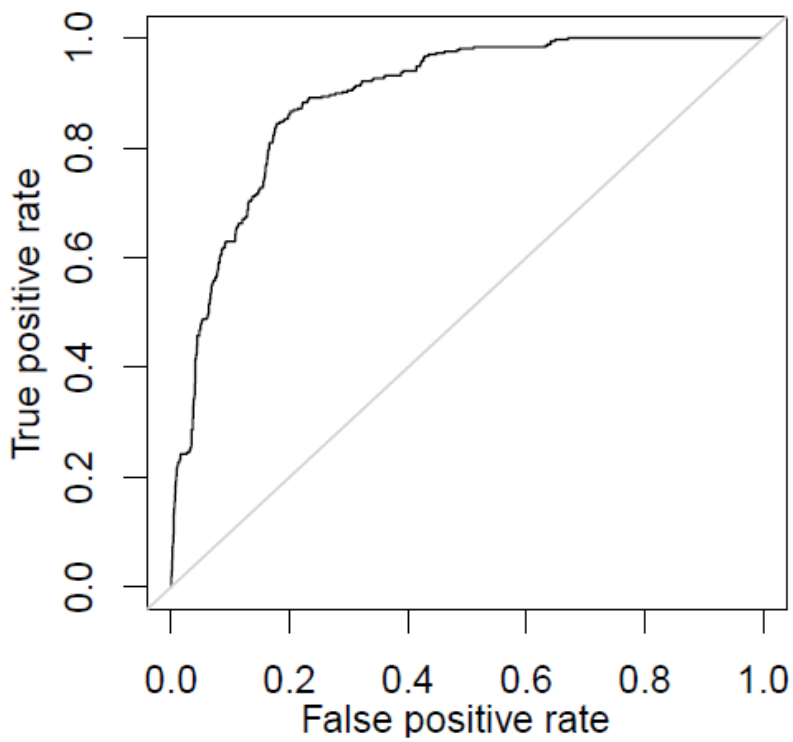


Limiar	0.75	0.625	0.5	0.375	0.25
Sensitivity (TPR)	0.074	0.106	0.136	0.305	0.510
Specificity (TNR)	0.997	0.995	0.995	0.963	0.963

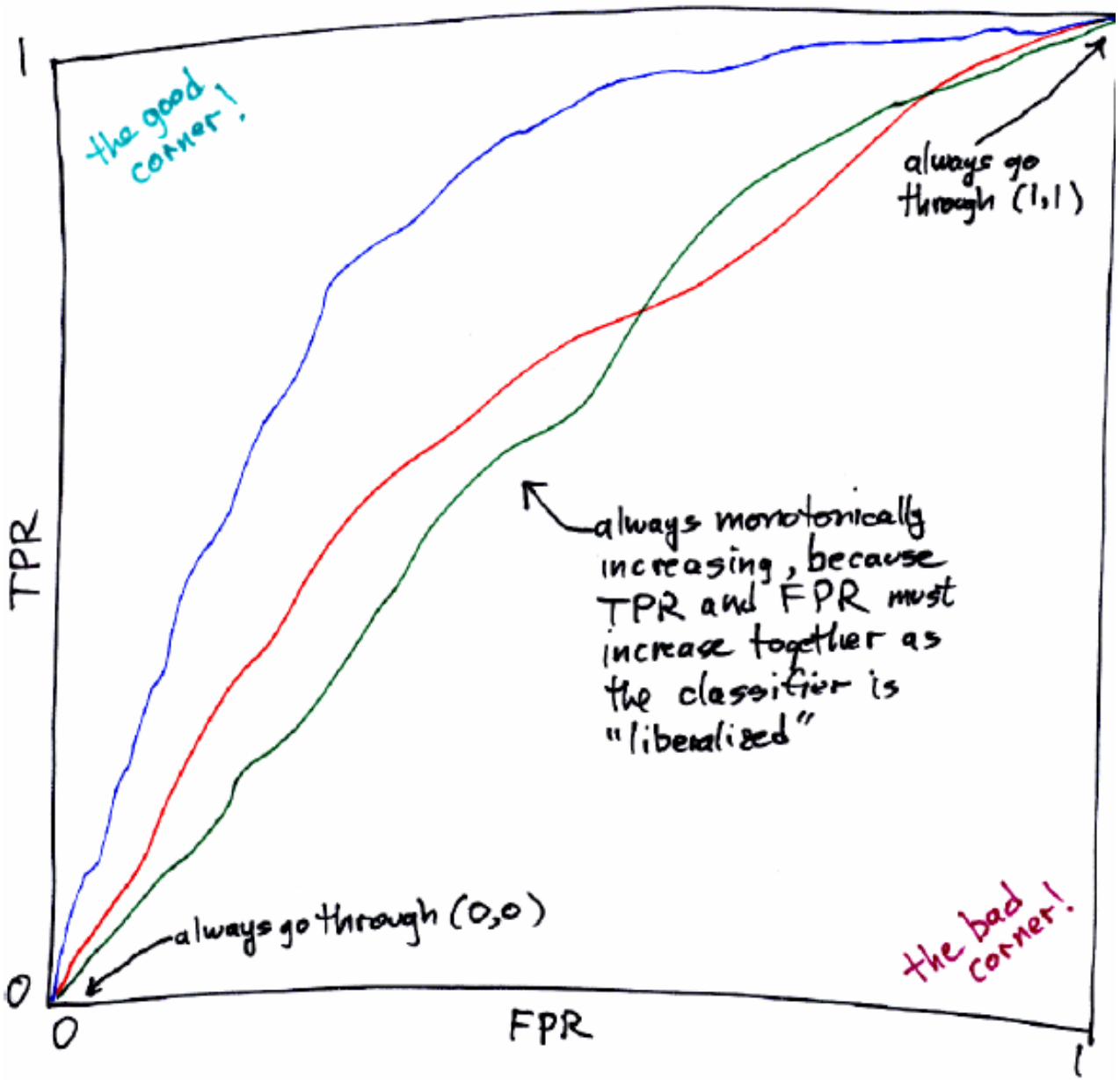
Exemplo: spam

Limiar	0.75	0.625	0.5	0.375	0.25
Sensitivity (TPR)	0.074	0.106	0.136	0.305	0.510
Specificity (TNR)	0.997	0.995	0.995	0.963	0.963

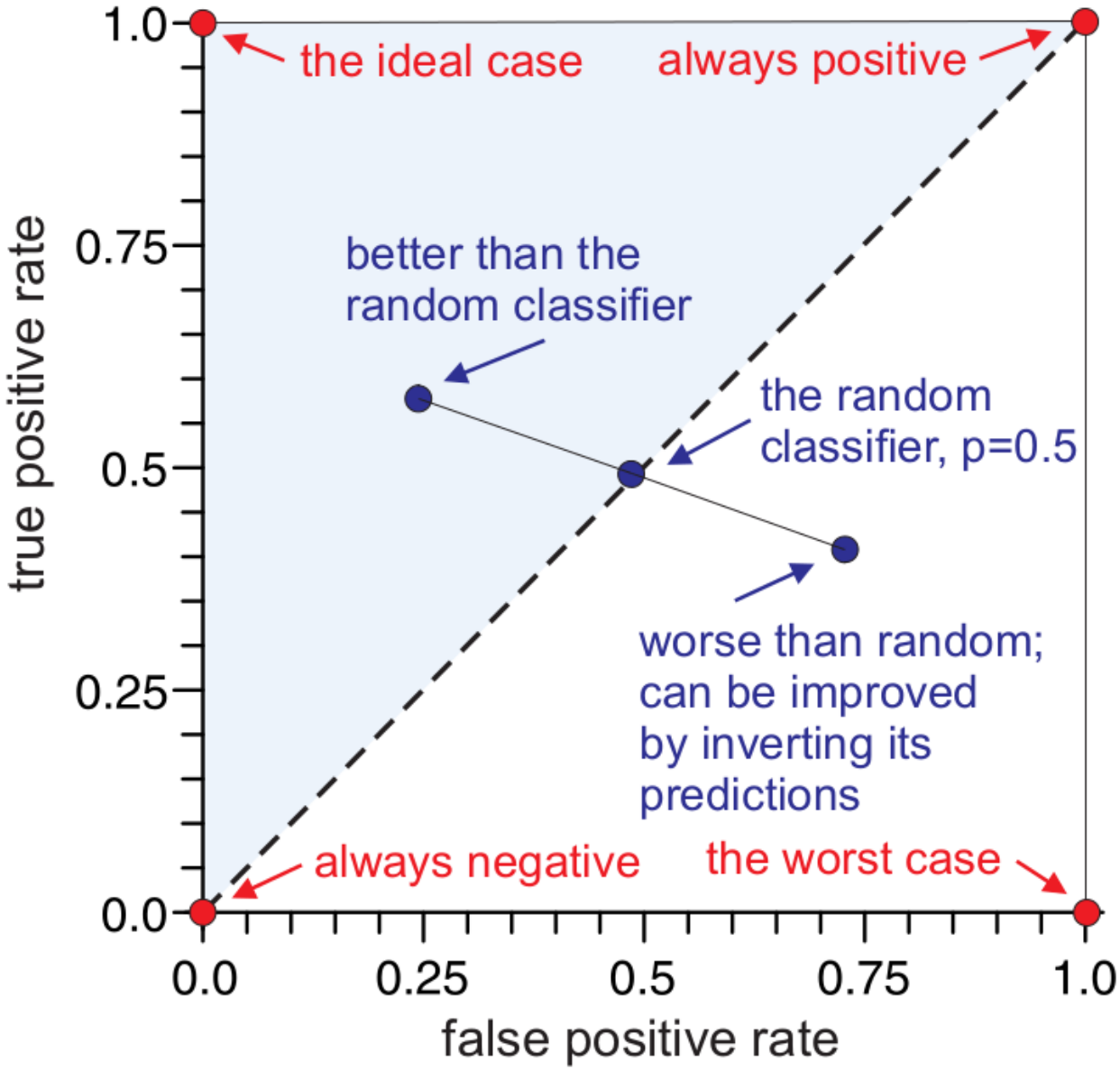




ROC *space*


1
TPR
0
0
FPR
1
the good corner!
always go through (1,1)
always monotonically increasing, because TPR and FPR must increase together as the classifier is "liberalized"
always go through (0,0)
the bad corner!

ROC *space*

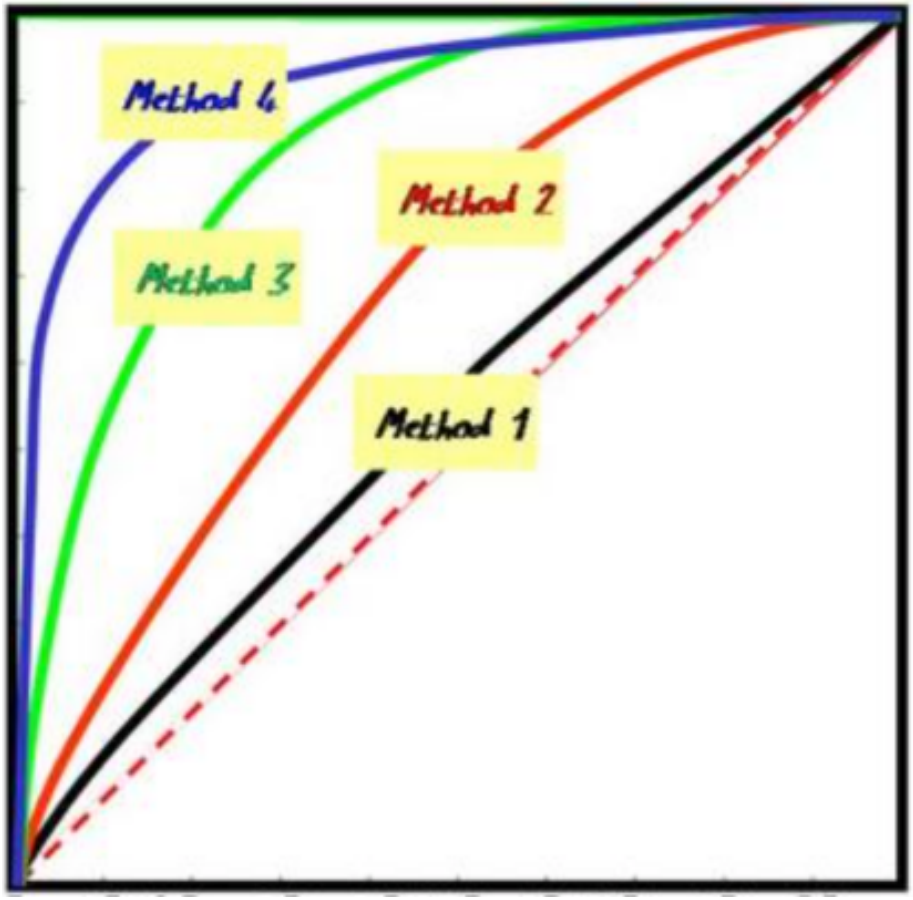

1.0
0.75
0.5
0.25
0.0
true positive rate
0.0
0.25
0.5
0.75
1.0
false positive rate
the ideal case
always positive
better than the
random classifier
the random
classifier, p=0.5
worse than random;
can be improved
by inverting its
predictions
always negative
the worst case

ROC: história

You may be wondering where the name "Reciever Operating Characteristic" came from. ROC analysis is part of a field called "Signal Dectection Theory" developed during World War II for the analysis of radar images. Radar operators had to decide whether a blip on the screen represented an enemy target, a friendly ship, or just noise. Signal detection theory measures the ability of radar receiver operators to make these important distinctions. Their ability to do so was called the Receiver Operating Characteristics. It was not until the 1970's that signal detection theory was recognized as useful for interpreting medical test results.

AUC: *Area Under the Curve*

A área sob a curva (AUC) é considerada como medida de acurácia.



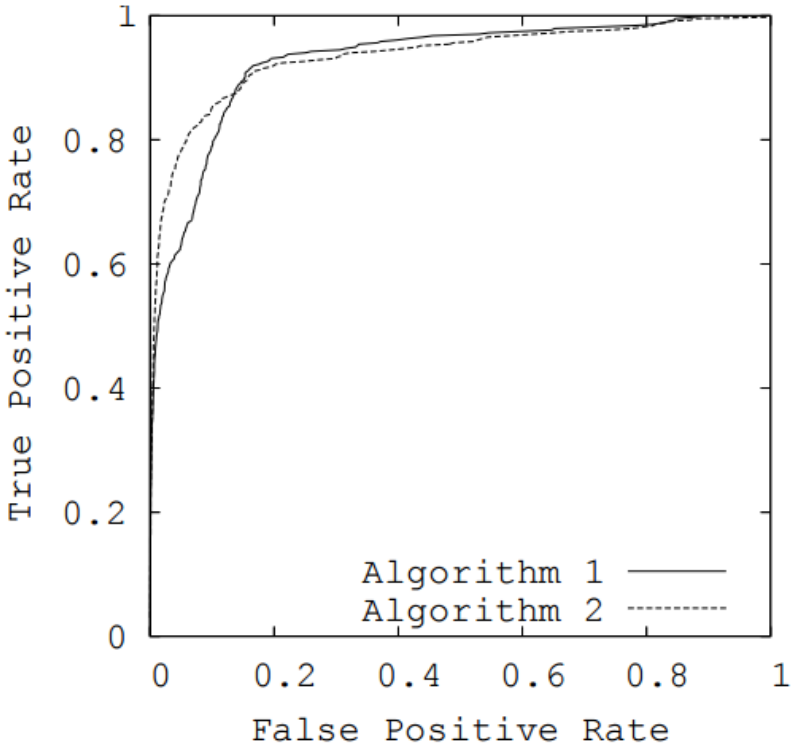
Tipicamente:

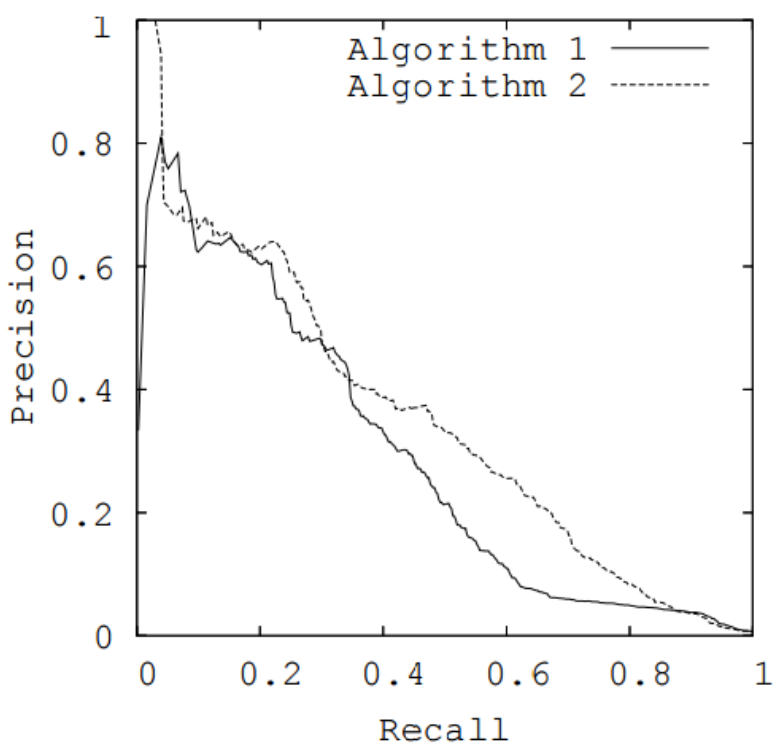
- 0.90 - 1.00 = *excellent*
- 0.80 - 0.90 = *good*
- 0.70 - 0.80 = *fair*
- 0.60 - 0.70 = *poor*
- 0.50 - 0.60 = *fail*

Precision - Recall (PR)

As curvas PR são consideradas quando:

- Conjuntos de dados com muito desbalanceamento entre as classes.
- A quantidade de exemplos negativos excede significativamente a de positivos (com isso, mesmo uma grande alteração no número de falsos positivos não terá impacto na curva ROC).





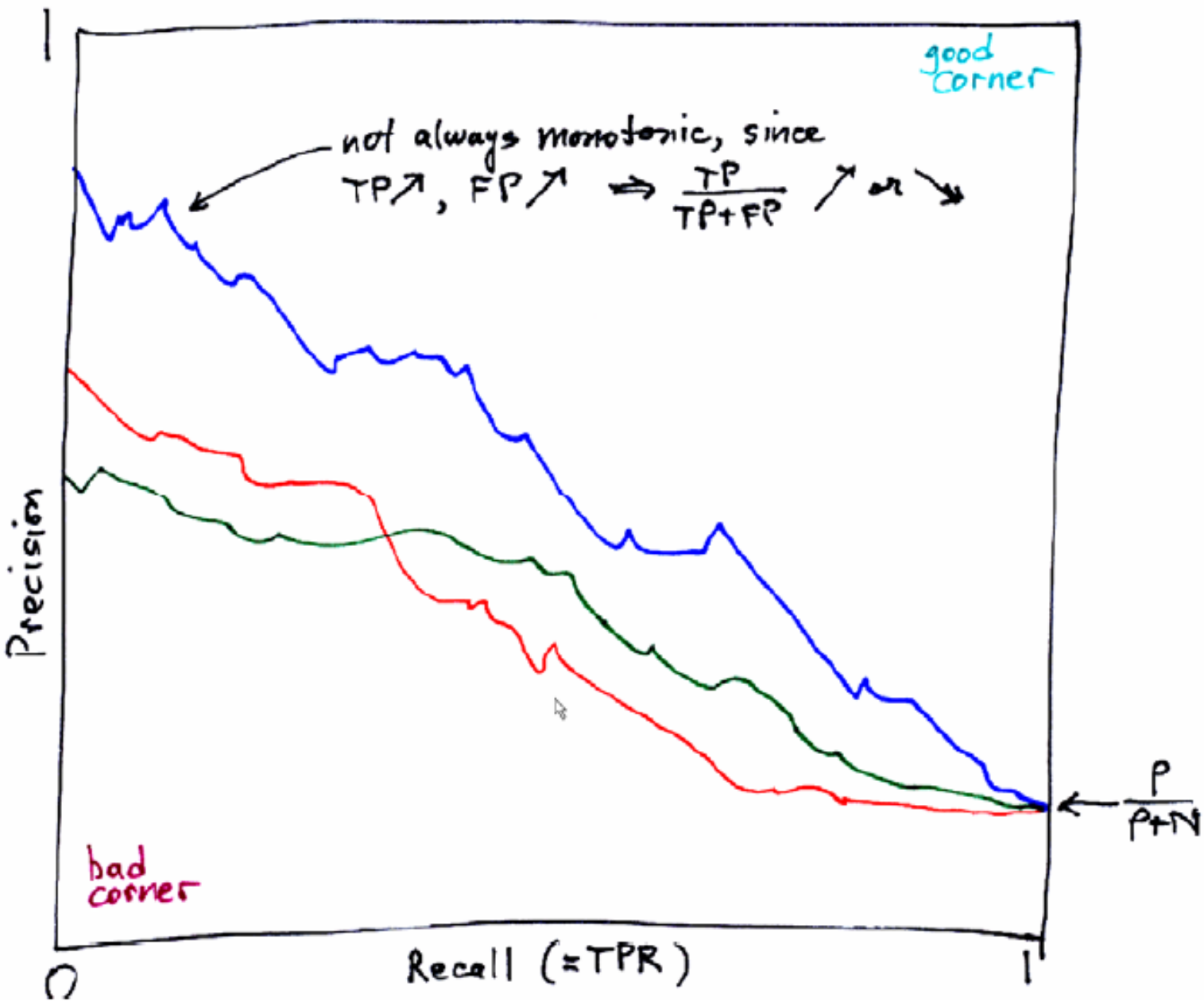
Curvas PR

Por exemplo, considere um problema de detecção de fraudes, onde exemplos negativos (não-fraude) são muito mais abundantes que positivos.

- Algoritmo 1: identificados 90 relevantes de 100.
  - ROC: TPR=90/100=0.9, FPR= 10/1,999,900=0.00000500025
  - PR: precision=0.9, recall=0.9
- Algoritmo 2: identificados 90 relevantes de 1000.
  - ROC: TPR=90/100=0.9, FPR=910/1,999,900=0.00045502275
  - PR: precision=90/1000=0.09, recall= 0.9

A diferença ROC é de 0.0004500225, enquanto a diferença PR é de 0.81.

PR *space*


1
good corner
not always monotonic, since
TP↗, FP↗ ⇒ TP/(TP+FP) ↗ or ↘
Precision
P/(P+N)
bad corner
0
Recall (=TPR)
1

Curvas PR: The Berkeley Segmentation Dataset and Benchmark

Dependendo do problema em questão, o uso de curvas PR pode ser mais apropriado. Por exemplo, o *The Berkeley Segmentation Dataset and Benchmark* considera curvas PR pelo fato de a quantidade de falsos positivos não ser tão relevante. Além disso, não é independente da resolução da imagem.

Mais sobre *precision* (precisão) e *recall* (revocação)

A precisão pode ser vista como uma medida de **quão bom** é um modelo, ao passo que a revocação é uma medida da sua **completude.** O que significa (e qual a informação que falta):

- Precisão==1?
- Revocação==1?

E se o desempenho do algoritmo não for satisfatório?

Alternativas:

- Obter mais dados para treinamento: contra variância alta.
- Selecionar as características mais significativas: contra variância alta.
- Adicionar mais características: contra bias alto.
- Aumentar a complexidade da combinação de características: contra bias alto.

Ao avaliar o desempenho de um classificador, é importante avaliar também a variância (e não apenas a média). Pode indicar instabilidade nos parâmetros.