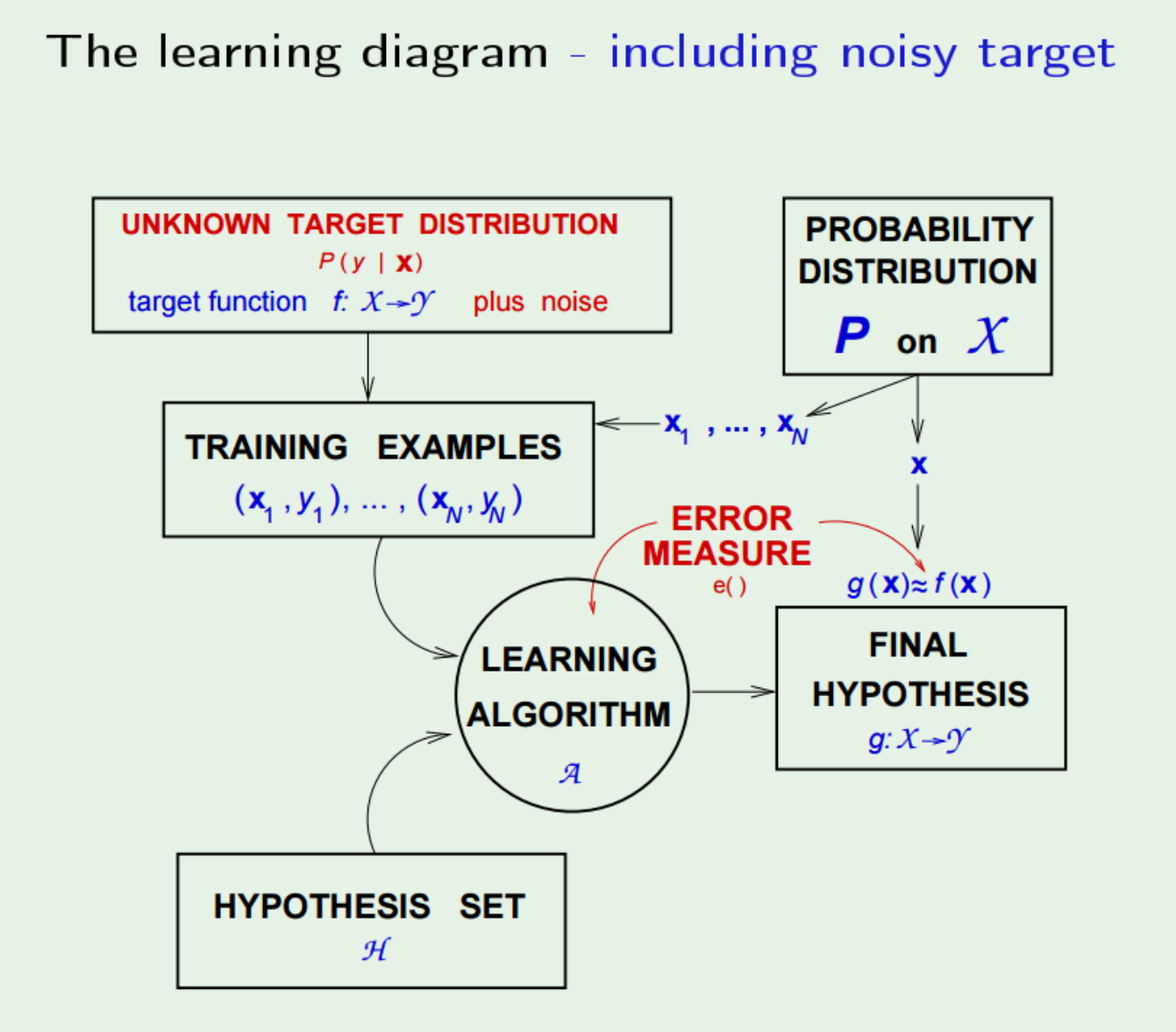
MAC 0459 / 5865

Data Science and Engineering

R. Hirata Jr. (hirata@ime.usp.br)

Class 20 (2020)

Stating the problem



Training a Classifier

Estimate or infer:

probability distributions, or

target functions

from **samples and prior knowledge** available.

Suppose for instance you already induced a decision tree or estimated the probability density using as input the training set.

What now?

Avaliação de classificadores

A avaliação de classificadores pode ter dois propósitos:

- **avaliação do desempenho do classificador**
  Quão bem um classificador generaliza a classificação/predição?
  (Quão bem ele funciona no "mundo real" ?)

- **escolha de um melhor classificador**
  Qual classificador é o melhor?

Critérios para escolha de um classificador

Critérios para escolha de um classificador

Diferentes aspectos podem ser considerados na avaliação de um classificador:

- taxa de acerto (ou probabilidade de erro)
- facilidade de interpretação
- tempo de treinamento
- tempo de aplicação
- robustez
- etc

Desempenho do classificador em termos de taxa de acertos é certamente um dos principais critérios. **Como avaliar desempenho de um classificador ?**

Terminologias no contexto de medidas de desempenho

Classificação geral (multi-classes)

- **Taxa de acerto**
- **Probabilidade de erro**
- **Matriz de confusão**

Classificação binária

- **TP, TN, FP, FN** (matriz de confusão no caso binário)
- **Sensibilidade e especificidade**
- **Erros do tipo I e do tipo II (Type I e Type II errors)**
- **Recall (sensibilidade) e precision**
- **Acurácia e precisão**
- **F-score**

Erros em problemas de classificação binária

		Condition (as determined by "Gold standard")		
		Condition Positive	Condition Negative	
Test Outcome	Test Outcome Positive	**True Positive**	**False Positive** (Type I error)	Positive predictive value = $\frac{\Sigma \text{ True Positive}}{\Sigma \text{ Test Outcome Positive}}$
	Test Outcome Negative	**False Negative** (Type II error)	**True Negative**	Negative predictive value = $\frac{\Sigma \text{ True Negative}}{\Sigma \text{ Test Outcome Negative}}$
		Sensitivity = $\frac{\Sigma \text{ True Positive}}{\Sigma \text{ Condition Positive}}$	**Specificity** = $\frac{\Sigma \text{ True Negative}}{\Sigma \text{ Condition Negative}}$	

Fonte: Wikipedia

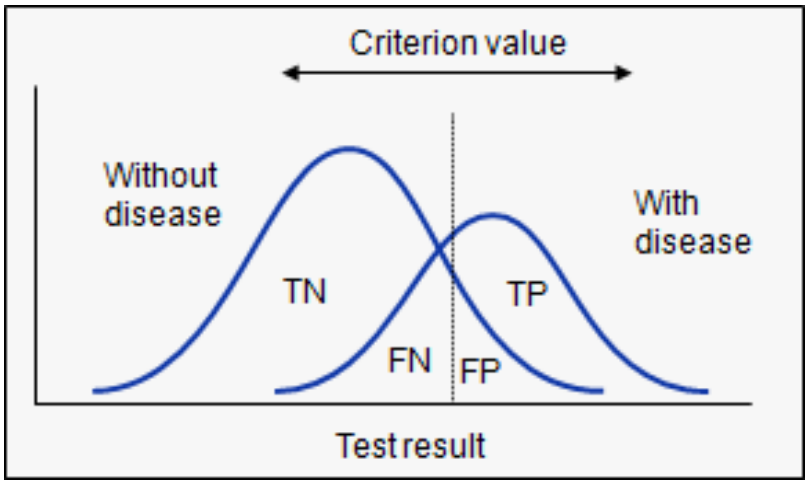
Erros em problemas de classificação binária

Classes

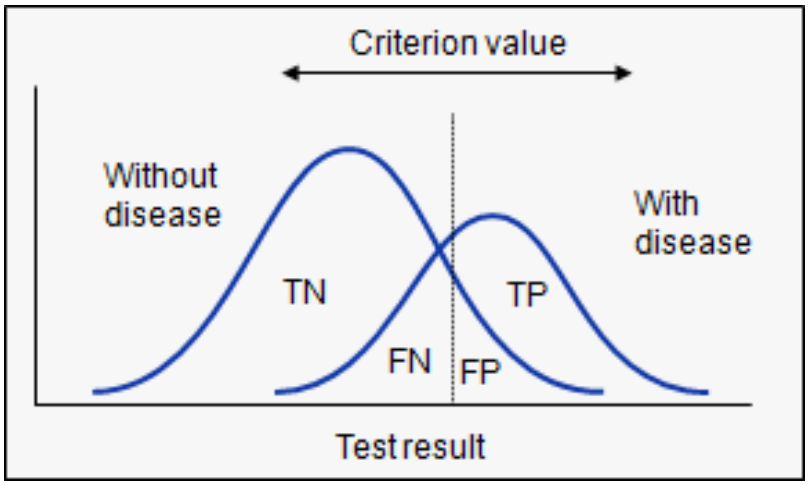
- POSITIVO
- NEGATIVO

Quatro possíveis diagnósticos:

- Falso-positivo (FP)
- Falso-negativo (FN)
- Verdadeiro-positivo (TP)
- Verdadeiro-negativo (TN)



Erros em problemas de classificação binária



Sensibilidade
(Prob. verdadeiro-positivo)

$$\frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

Especificidade
(Prob. verdadeiro-negativo)

$$\frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}$$

Revocação (recall) e Precisão (precision)

Usado em recuperação de informação:
N: **conjunto dos items esperados**
R: **conjunto de items recuperados**

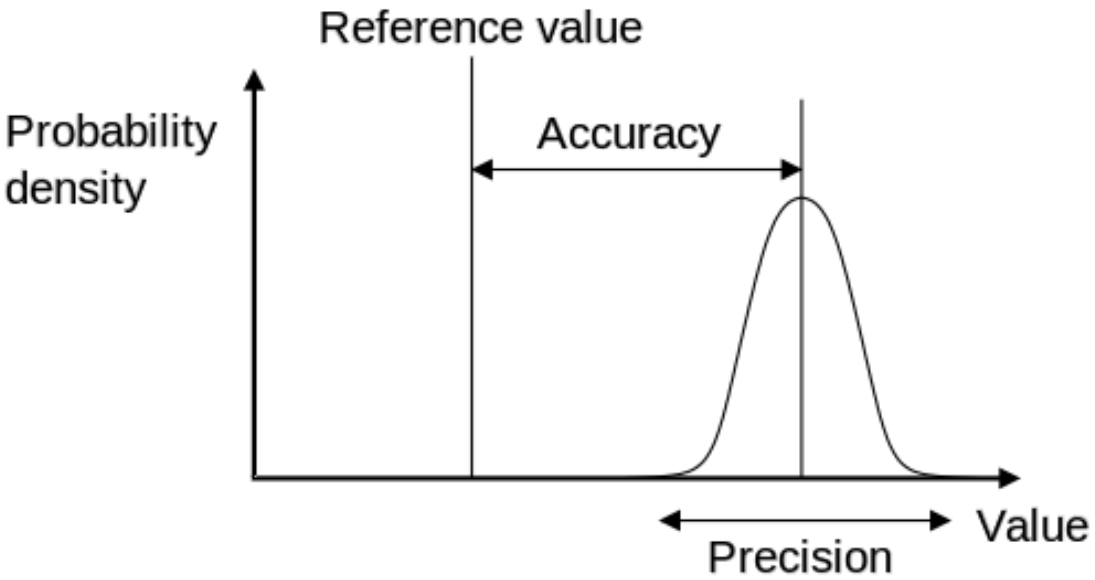
Revocação: proporção de itens esperados que são recuperados

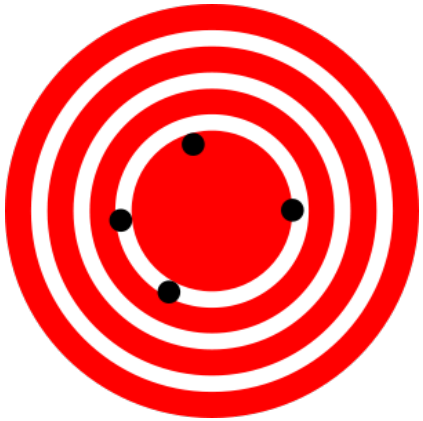
$$Recall = \frac{|N \cap R|}{|N|}$$

Precisão: proporção de itens recuperados que são itens esperados

$$Precision = \frac{|N \cap R|}{|R|}$$

Acurácia e precisão





(alta acurácia, baixa precisão)

(alta precisão, baixa acurácia)

Fonte das imagens: Wikimedia Commons

Medida F_1

Média harmônica entre a sensibilidade e especificidade.

$$F_1 = \frac{2 * \text{Especificidade} * \text{Sensibilidade}}{\text{Especificidade} + \text{Sensibilidade}}$$

$$F_1 = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Visão geral sobre estimação de erro

Preditor: $f : \mathrm{X} \rightarrow Y$

Visão geral sobre estimação de erro

Preditor: $f : X \rightarrow Y$

- **se o domínio das instâncias X e as respectivas saídas em Y são conhecidos, então pode-se calcular o erro exato (verdadeiro) de f**

Visão geral sobre estimação de erro

Preditor: $f : X \rightarrow Y$

- **se o domínio das instâncias X e as respectivas saídas em Y são conhecidos, então pode-se calcular o erro exato (verdadeiro) de f**
- **como isso em geral não é o caso, erros são estimados a partir de amostras desse domínio**

Visão geral sobre estimação de erro

Preditor: $f : X \rightarrow Y$

- **se o domínio das instâncias X e as respectivas saídas em Y são conhecidos, então pode-se calcular o erro exato (verdadeiro) de f**
- **como isso em geral não é o caso, erros são estimados a partir de amostras desse domínio**
- **a partir disso, a questão principal é saber o quão confiável é a estimativa. Calcular, por exemplo, intervalos de confiança**

Visão geral sobre estimação de erro

Preditor: $f : X \rightarrow Y$

- **se o domínio das instâncias X e as respectivas saídas em Y são conhecidos, então pode-se calcular o erro exato (verdadeiro) de f**
- **como isso em geral não é o caso, erros são estimados a partir de amostras desse domínio**
- **a partir disso, a questão principal é saber o quão confiável é a estimativa. Calcular, por exemplo, intervalos de confiança**
- **como comparar dois preditores para os quais têm-se apenas um erro estimado ? Usar, por exemplo, teste de hipóteses**

Visão geral sobre estimação de erro

Preditor: $f : X \rightarrow Y$

- **se o domínio das instâncias X e as respectivas saídas em Y são conhecidos, então pode-se calcular o erro exato (verdadeiro) de f**
- **como isso em geral não é o caso, erros são estimados a partir de amostras desse domínio**
- **a partir disso, a questão principal é saber o quão confiável é a estimativa. Calcular, por exemplo, intervalos de confiança**
- **como comparar dois preditores para os quais têm-se apenas um erro estimado ? Usar, por exemplo, teste de hipóteses**
- **existem formas melhores para estimar o erro? Fazer, por exemplo, validação cruzada**

Erro de um preditor

Seja $f(\mathrm{x})$ a predição e y o valor-alvo. O **erro de predição** pode ser caracterizado por uma **função de perda**

$$L(y, f(\mathrm{x})) = \begin{cases} (y - f(\mathrm{x}))^2, & \text{erro quadrático} \\ |y - f(\mathrm{x})|, & \text{erro absoluto,} \\ I(y \neq f(\mathrm{x})), & \text{perda zero-um.} \end{cases}$$

O erro (verdadeiro) de um preditor f é dado pelo valor esperado:

$$\mathrm{Erro}(f) = E[L(y, f(\mathrm{x}))]$$

calculado com respeito à distribuição D associado ao espaço $\mathrm{X} \times Y$.

Erro de um classificador

Erro de classificação: função de perda zero-um

(OBS.: nem todas as funções de perda fazem sentido em qualquer problema de classificação)

Para a função de perda zero-um, temos

$$Erro(f) = P_{x \in X}(y \neq f(x))$$

Isto é a probabilidade de classificação incorreta.

Daqui em diante, os conceitos são apresentados tendo-se em mente problemas de classificação. Porém, vários conceitos podem ser estendidos/aplicados para os casos nos quais a função f é geral.

- **Se fosse possível calcular o erro $Erro(f)$, a escolha do "melhor" classificador seria fácil. Além disso, saberíamos qual seria o desempenho do classificador escolhido "no mundo real".**

- **Se fosse possível calcular o erro $Erro(f)$, a escolha do "melhor" classificador seria fácil. Além disso, saberíamos qual seria o desempenho do classificador escolhido "no mundo real".**

- **Na prática, a distribuição associada ao espaço X das instâncias a serem classificadas não é conhecida.**
  Supõe-se apenas que há uma distribuição conjunta $p(\mathrm{x}, y)$ (desconhecida) associada ao espaço $\mathrm{X} \times Y$, $Y = \{1, 2, \ldots, c\}$ e que temos uma amostra obtida segundo essa distribuição:

$$A = \{(\mathrm{x}_1, y_1), (\mathrm{x}_2, y_2), \ldots, (\mathrm{x}_m, y_m)\}$$

- **Se fosse possível calcular o erro $Erro(f)$, a escolha do "melhor" classificador seria fácil. Além disso, saberíamos qual seria o desempenho do classificador escolhido "no mundo real".**

- **Na prática, a distribuição associada ao espaço X das instâncias a serem classificadas não é conhecida.**
  Supõe-se apenas que há uma distribuição conjunta $p(\mathrm{x}, y)$ (desconhecida) associada ao espaço $\mathrm{X} \times Y$, $Y = \{1, 2, \ldots, c\}$ e que temos uma amostra obtida segundo essa distribuição:

$$A = \{(\mathrm{x}_1, y_1), (\mathrm{x}_2, y_2), \ldots, (\mathrm{x}_m, y_m)\}$$

- **Como devemos estimar o erro de um classificador g?**

  O que podemos dizer dessas estimativas?

Erro de treinamento

Erro de treinamento (ou resubstitution error): erro calculado sobre um conjunto de treinamento $S \subseteq A$

$$Erro_S(g) = \frac{1}{|S|} \sum_{i=1}^{|S|} \delta(y_i, g(x_i))$$

na qual δ é a função delta de Kronecker dada por

$$\delta(a, b) = \begin{cases} 1, & \textbf{se } a \neq b, \\ 0, & \textbf{se } a = b. \end{cases}$$

$Erro_S(g)$ **= proporção de exemplos de S classificados incorretamente por g**

$Erro_S(g)$ **é uma estimação super-otimista de** $Erro(g)$

Erro de treinamento – Overfitting (1)

Ajuste excessivo aos dados de treinamento

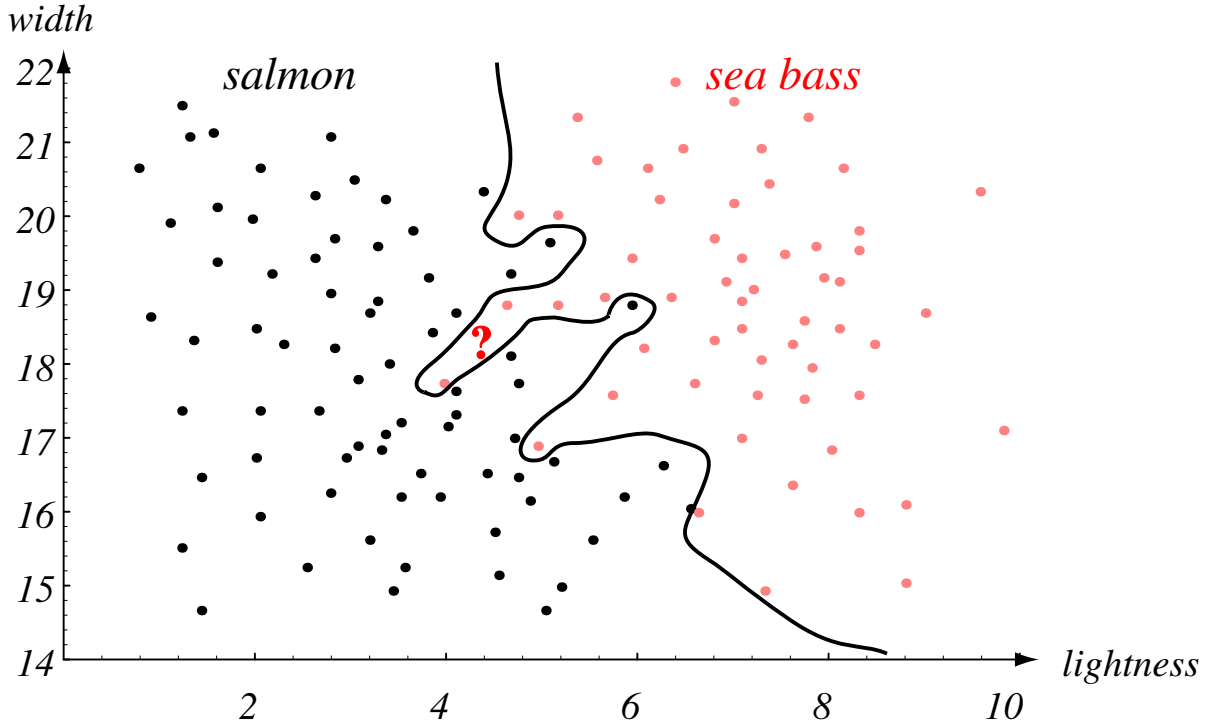


FIGURE 1.5. Overly complex models for the fish will lead to decision boundaries that are complicated. While such a decision may lead to perfect classification of our training samples, it would lead to poor performance on future patterns. The novel test point marked **?** is evidently most likely a salmon, whereas the complex decision boundary shown leads it to be classified as a sea bass. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Erro de treinamento – Overfitting (2)

Mesmos dados da página anterior; superfície de decisão mais simples

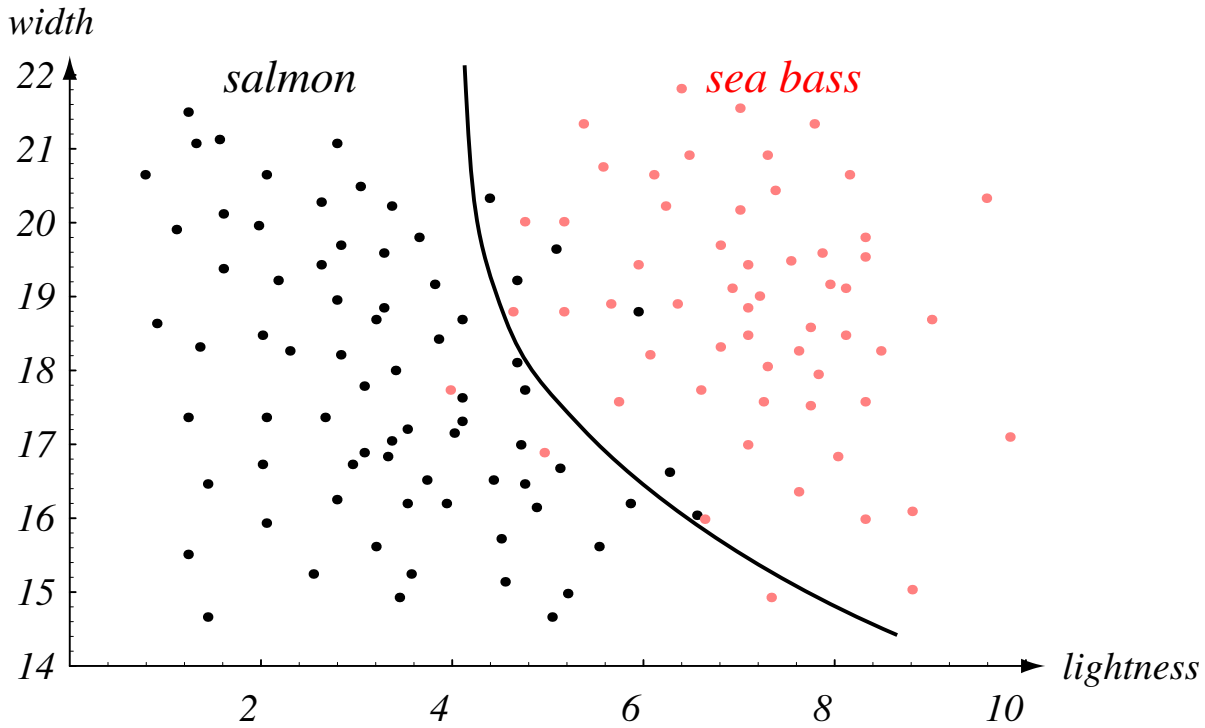
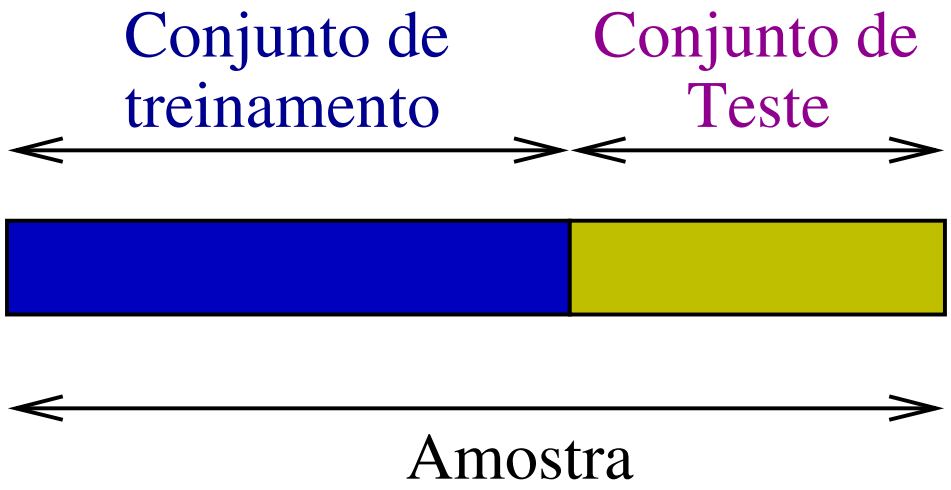


FIGURE 1.6. The decision boundary shown might represent the optimal tradeoff between performance on the training set and simplicity of classifier, thereby giving the highest accuracy on new patterns. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Método Holdout (1)

Dividir a amostra (A) em conjunto de treinamento (S) e conjunto de teste (T), na proporção 2:1, por exemplo.



Método Holdout (2)

- **usa-se o conjunto de treinamento para treinar um classificador**

Método Holdout (2)

- usa-se o conjunto de treinamento para treinar um classificador
- usa-se o conjunto de teste para estimar o erro do classificador

Método Holdout (2)

- **usa-se o conjunto de treinamento para treinar um classificador**
- **usa-se o conjunto de teste para estimar o erro do classificador**

- **Erro amostral (ou holdout)**: Erro estimado de g com respeito ao conjunto de teste $|T|$:

$$Erro_T(g) = \frac{1}{|T|} \sum_{i=1}^{|T|} \delta(y_i, g(x_i))$$

na qual $\delta(a, b) = 1$ se $a \neq b$ e $\delta(a, b) = 0$ se $a = b$.

(proporção de exemplos em T classificados incorretamente por g)

Em geral, o erro de teste é maior que o erro de treinamento

$$Erro_T(g) \geq Erro_S(g)$$

Holdout error

O que pode dar errado:

Holdout error

O que pode dar errado:

- **quando o conjunto A é pequeno, ambos os conjuntos S e T são menores ainda ... e estimações feitas sobre conjuntos muito pequenos não são confiáveis ...**

Holdout error

O que pode dar errado:

- **quando o conjunto A é pequeno, ambos os conjuntos S e T são menores ainda ... e estimações feitas sobre conjuntos muito pequenos não são confiáveis ...**

- **Como o erro estimado é calculado numa única divisão (S, T) da amostra, essa partição poderia, por coincidência, ser a pior ou melhor possível em termos de estimação de erro**

Validação cruzada (1)

As desvantagens da técnica holdout podem ser compensadas usando-se técnicas de reamostragem, ao custo de aumento no tempo computacional

- **Validação cruzada (reamostragem sem reposição)**

- **Bootstrap (reamostragem com reposição)**

Validação cruzada (1)

As desvantagens da técnica holdout podem ser compensadas usando-se técnicas de reamostragem, ao custo de aumento no tempo computacional

- **Validação cruzada (reamostragem sem reposição)**
  - **amostragem aleatória**
  - **k-fold cross-validation**
  - **leave-one-out cross-validation**
  - **2-fold cross-validation**

- **Bootstrap (reamostragem com reposição)**

Validação cruzada (2)

Idéia: Em vez de se considerar apenas uma partição (S, T) da amostra, consideram-se k partições (S_j, T_j).

Para cada partição (S_j, T_j), faz-se o treinamento com S_j e estima-se o erro sobre T_j. Isto resulta em k erros $Erro_{T_j}$

Validação cruzada (2)

Idéia: Em vez de se considerar apenas uma partição (S, T) da amostra, consideram-se k partições (S_j, T_j).

Para cada partição (S_j, T_j), faz-se o treinamento com S_j e estima-se o erro sobre T_j. Isto resulta em k erros $Erro_{T_j}$

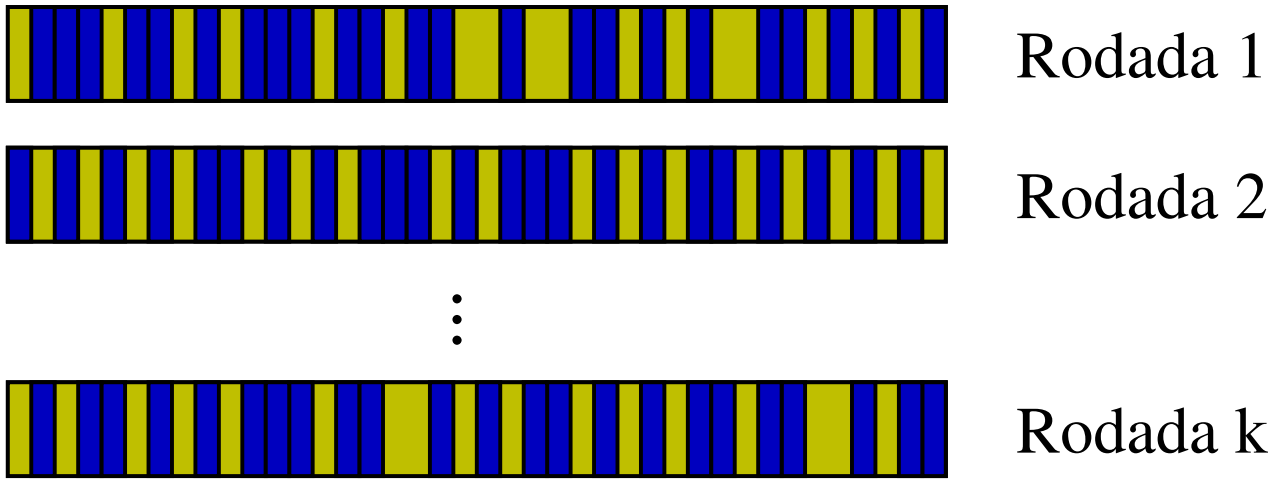
O erro de validação cruzada é dado pela média dos erros $Erro_{T_j}$:

$$Erro = \frac{1}{k} \sum_{j=1}^{k} Erro_{T_j}$$

Validação cruzada (3) – Reamostragem aleatória

Tamanho do conjunto de treinamento fixo em $m < n$

Repetir k rodadas de treinamento; para cada rodada, sortear aleatoriamente m exemplos da amostra.

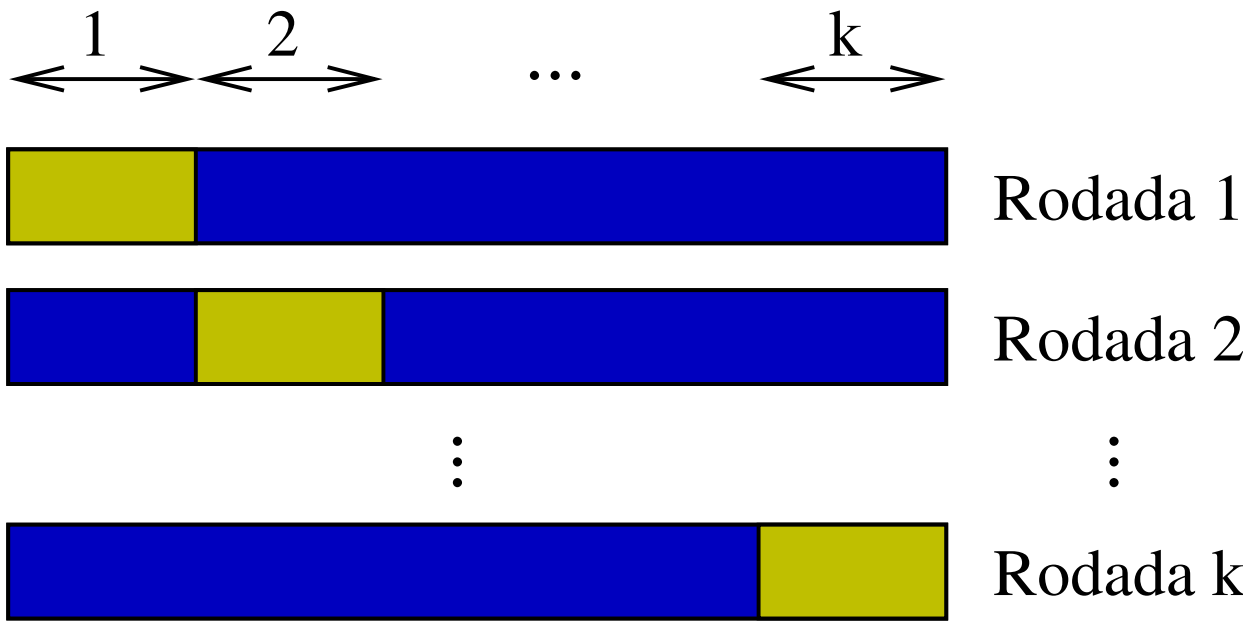


Conjunto de treinamento: exemplos em azul
Conjunto de teste: exemplos em amarelo

Validação cruzada (4) – k-fold cross validation

Dividir a amostra em k partes de tamanhos (aproximadamente) iguais.

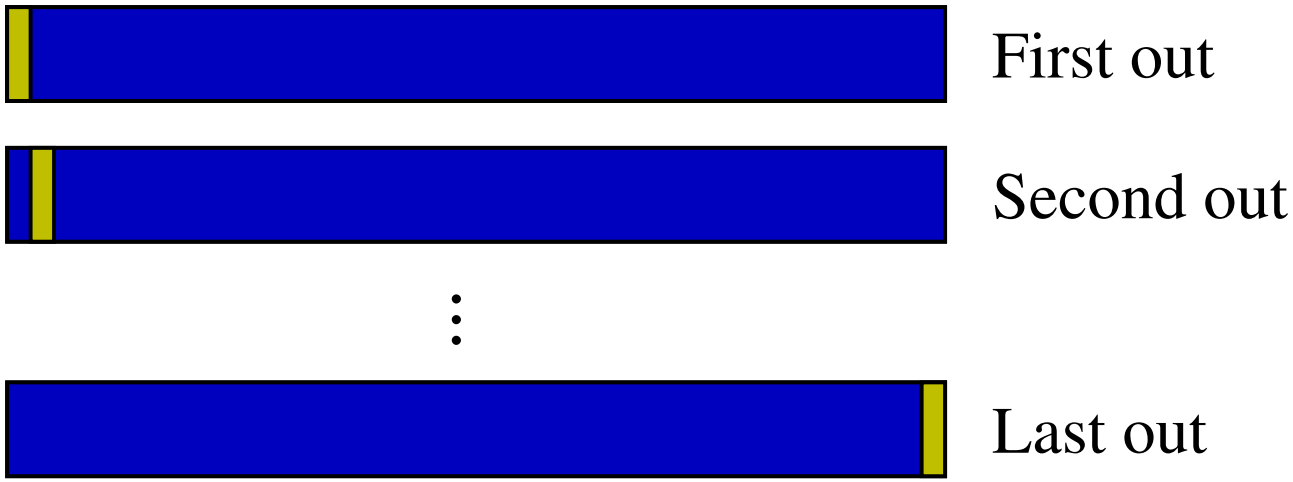
Repetir k rodadas de treinamento, deixando alternadamente uma das partes para validação em cada rodada



Validação cruzada (5) – Leave-one-out cross validation

Caso degenerado do k-fold cross validation

Caso no qual $k = n$ (ou seja, apenas um exemplo de validação em cada rodada)



Geralmente usado quando a amostra é pequena.

Validação cruzada (5) – 2-fold cross validation

2-fold cross validation é um caso particular do k-fold cross validation (caso $k = 2$).

Pode ser visto também como uma variação do método holdout (na qual o processo de treinamento e estimação de erro é repetido duas vezes trocando-se o papel dos conjuntos de treinamento/validação e o erro é calculado como a média dos erros obtidos).

Boostrap (reamostragem com reposição)

Idéia:

similar à reamostragem aleatória, com a diferença de que no bootstrap a reamostragem é com reposição

(um mesmo exemplo pode ser sorteado mais de uma vez e portanto aparecer mais de uma vez num conjunto de treinamento)

Boostrap e cálculo de erro

Sejam k conjuntos de treinamento, $Z_1, Z_2, \ldots, Z_k$, todos com tamanho n e obtidos por reamostragem com reposição

Boostrap e cálculo de erro

Sejam k conjuntos de treinamento, $Z_1, Z_2, \ldots, Z_k$, todos com tamanho n e obtidos por reamostragem com reposição

Calcula-se o erro para cada classificador sobre a amostra toda (cada um deles treinados usando um Z_j).

O erro bootstrap é a média dos erros dos classificadores.

Como a reamostragem foi com reposição, significa que há exemplos que estão tanto no conjunto de treinamento como no de validação.

Boostrap e cálculo de erro

Sejam k conjuntos de treinamento, $Z_1, Z_2, \ldots, Z_k$, todos com tamanho n e obtidos por reamostragem com reposição

Calcula-se o erro para cada classificador sobre a amostra toda (cada um deles treinados usando um Z_j).

O erro bootstrap é a média dos erros dos classificadores.

Como a reamostragem foi com reposição, significa que há exemplos que estão tanto no conjunto de treinamento como no de validação.

O erro bootstrap tende a subestimar o verdadeiro erro.

Leave one out Bootstrap

Alternativa:

usar o leave one out Bootstrap error, dado por:

$$Erro_{Boot} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|C^{-i}|}\sum_{j\in C^{-i}}\delta(g_j(x_i), y_i)$$

na qual C^{-i} é o conjunto de índices j em $\{1, 2, \ldots, k\}$ cuja amostra Z_j não contém o exemplo x_i.

(em outras palavras, para o classificador g_j, treinado com o conjunto Z_j, o erro é calculado somente sobre os exemplos da amostra que não aparecem em Z_j)

Leave one out Bootstrap

- **Cada conjunto de treinamento** Z_j tem n exemplos, mas como foi obtido por reamostragem com reposição, **contém apenas aproximadamente** 63% **dos exemplos** do conjunto original (o correspondente conjunto de validação contém aproximadamente 37% deles).

Leave one out Bootstrap

- **Cada conjunto de treinamento** Z_j tem n exemplos, mas como foi obtido por reamostragem com reposição, **contém apenas aproximadamente** 63% **dos exemplos** do conjunto original (o correspondente conjunto de validação contém aproximadamente 37% deles).

- **Treinar com 63% dos dados pode resultar em classificadores com acurácia bem inferior que treinar com 100% dos dados.**

Leave one out Bootstrap

- **Cada conjunto de treinamento** Z_j tem n exemplos, mas como foi obtido por reamostragem com reposição, **contém apenas aproximadamente** 63% **dos exemplos** do conjunto original (o correspondente conjunto de validação contém aproximadamente 37% deles).

- **Treinar com 63% dos dados pode resultar em classificadores com acurácia bem inferior que treinar com 100% dos dados.**

- O **leave one out Bootstrap error** tende a ser pessimista

0.632 Bootstrap

- O **leave one out Bootstrap error** tende a ser pessimista

0.632 Bootstrap

- O **leave one out Bootstrap error** tende a ser pessimista

  Para compensar isso, sugere-se o **0.632 Bootstrap estimator** dado por

$$Erro_{0.632\,Boot} = 0.632 Erro_{Boot} + 0.368 Erro_{train}$$

  na qual $Erro_{train}$ **é a média dos erros sobre os conjuntos de treinamento**.

  A idéia é que $Erro_{train}$ do segundo termo, que é otimista, puxe o erro pessimista $Erro_{Boot}$ para baixo.

Erro médio (erro esperado)

Em vez de considerar a probabilidade de erro, podemos considerar o **erro médio**

Dado um algoritmo de treinamento, para cada conjunto de treinamento S, temos um classificador $g(S)$ e um erro $Erro(g(S))$

O erro médio do algoritmo é a média dos erros calculados sobre cada possível conjunto de treinamento de mesmo tamanho n.

Para diminuir a variância amostral, é comum serem considerados m repetições do k-fold cross validation e serem utilizados as m médias dessas m repetições.

O erro de treinamento $Erro_S(g)$ **é uma estimação super-otimista de** $Erro(g)$ pois

os algoritmos de treinamento tendem a gerar classificadores que se ajustam aos dados de treinamento

O ajuste demasiado aos dados de treinamento é conhecido por overfitting. Quanto mais flexível é a classe de classificadores utilizados (ou seja, maior o número de parâmetros), maior é a tendência de overfitting.

Técnicas de validação

Muitas vezes, os dados de teste são usados para validação.

Exemplos:

- o número de iterações no treinamento da rede neural pode ser interrompido a partir do momento em que o erro relativo ao conjunto de dados não usado no treinamento começa a aumentar
- o número de vizinhos k ideal na técnica k-NN pode ser escolhido como sendo aquele que produz menor erro no conjunto de dados não usados para classificação.
- etc

Nesses casos, o erro estimado para o modelo selecionado tenderá a ser subestimado.

Conjuntos de treinamento, validação e de teste

Nos casos em que deseja-se fazer a seleção de um modelo e em seguida estimar o erro do modelo selecionado, a amostra deve ser dividida em três partes:

- **Conjunto de treinamento**
- **Conjunto de validação**
- **Conjunto de teste**

- **Conjunto de treinamento**: usado para construir um classificador (ajuste de parâmetros)

- **Conjunto de validação**: usado para selecionar um classificador (escolher os parâmetros ótimos)

- **Conjunto de teste**: usado para estimar o erro do classificador escolhido

- **Conjunto de treinamento: usado para construir um classificador (ajuste de parâmetros)**
  No caso do classificador k-NN, esse conjunto será usado para classificar um elemento

- **Conjunto de validação: usado para selecionar um classificador (escolher os parâmetros ótimos)**
  No caso do classificador k-NN, esse conjunto pode ser usado para escolher o "melhor" k

- **Conjunto de teste: usado para estimar o erro do classificador escolhido**
  A probabilidade de erro é estimada no conjunto de teste que não foi utilizado no treinamento e seleção do modelo

quantidade de exemplos é grande: abordagem holdout
quantidade de exemplos pequena: abordagem cruzada (cross-validation, bootstrap)

Qual o número ideal de partes ($k =?$) na validação cruzada?

- **Número grande de partes**

- **Número pequeno de partes**

Qual o número ideal de partes ($k =?$) na validação cruzada?

- **Número grande de partes**
  - **Estimativa de erro mais precisa**
  - **porém, variância maior**
  - **maior tempo de treinamento (várias rodadas de treinamento)**
- **Número pequeno de partes**

Qual o número ideal de partes ($k = ?$) na validação cruzada?

- **Número grande de partes**
  - **Estimativa de erro mais precisa**
  - **porém, variância maior**
  - **maior tempo de treinamento (várias rodadas de treinamento)**
- **Número pequeno de partes**
  - **menos rodadas de treinamento; portanto, tempo de processamento menor**
  - **variância dos erros pequena**
  - **estimativa do erro menos precisa**

Validação cruzada (sem reposição)

- **Escolha do número de partes deve considerar o tamanho da amostra**
  - **Amostra grande**
  - **Amostra pequena**
- **Em geral, usa-se $k = 10$ (ou $k = 5$)**

Validação cruzada (sem reposição)

- **Escolha do número de partes deve considerar o tamanho da amostra**
  - **Amostra grande: número de partes pode ser pequena; por exemplo, 3**
  - **Amostra pequena**
- **Em geral, usa-se $k = 10$ (ou $k = 5$)**

Validação cruzada (sem reposição)

- **Escolha do número de partes deve considerar o tamanho da amostra**
  - **Amostra grande: número de partes pode ser pequena; por exemplo, 3**
  - **Amostra pequena: recomendável que o número de partes seja grande; no caso extremo, leave-one-out cross validation**
- **Em geral, usa-se $k = 10$ (ou $k = 5$)**

Treinamento, validação e teste

Na prática, pode-se seguir o seguinte procedimento

Treinamento, validação e teste

Na prática, pode-se seguir o seguinte procedimento

1. **Separar a amostra em três partes (conjunto de treinamento, validação e teste)**

Treinamento, validação e teste

Na prática, pode-se seguir o seguinte procedimento

1. **Separar a amostra em três partes (conjunto de treinamento, validação e teste)**
2. **Seleção do modelo: repetir um número desejado de vezes**
   1. escolher os parâmetros de treinamento
   2. treinar com o conjunto de treinamento
   3. validar com o conjunto de validação

Treinamento, validação e teste

Na prática, pode-se seguir o seguinte procedimento

1. **Separar a amostra em três partes (conjunto de treinamento, validação e teste)**
2. **Seleção do modelo: repetir um número desejado de vezes**
   1. escolher os parâmetros de treinamento
   2. treinar com o conjunto de treinamento
   3. validar com o conjunto de validação
3. **Retreinar o modelo selecionado: para os parâmetros de treinamento que resultou no menor erro sobre o conjunto de validação, treinar com os exemplos no conjunto de treinamento mais no de validação**

Treinamento, validação e teste

Na prática, pode-se seguir o seguinte procedimento

1. **Separar a amostra em três partes (conjunto de treinamento, validação e teste)**
2. **Seleção do modelo: repetir um número desejado de vezes**
   1. escolher os parâmetros de treinamento
   2. treinar com o conjunto de treinamento
   3. validar com o conjunto de validação
3. **Retreinar o modelo selecionado: para os parâmetros de treinamento que resultou no menor erro sobre o conjunto de validação, treinar com os exemplos no conjunto de treinamento mais no de validação**
4. **Estimar o erro sobre o conjunto de teste**

Treinamento, validação e teste

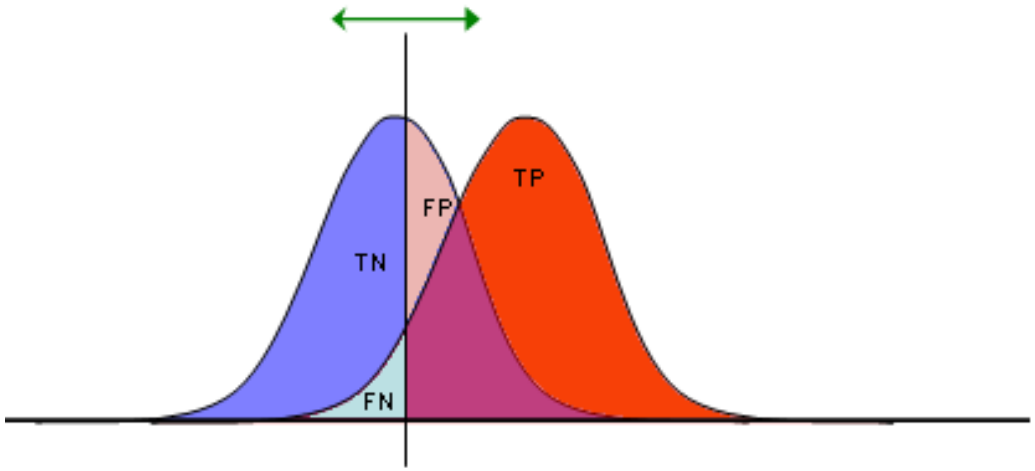
Se considerarmos validação cruzada: (conjuntos de treinamento e validação são um conjunto só – o processo de treinamento faz o particionamento)

Amostras são divididas em duas partes (treinamento e teste)

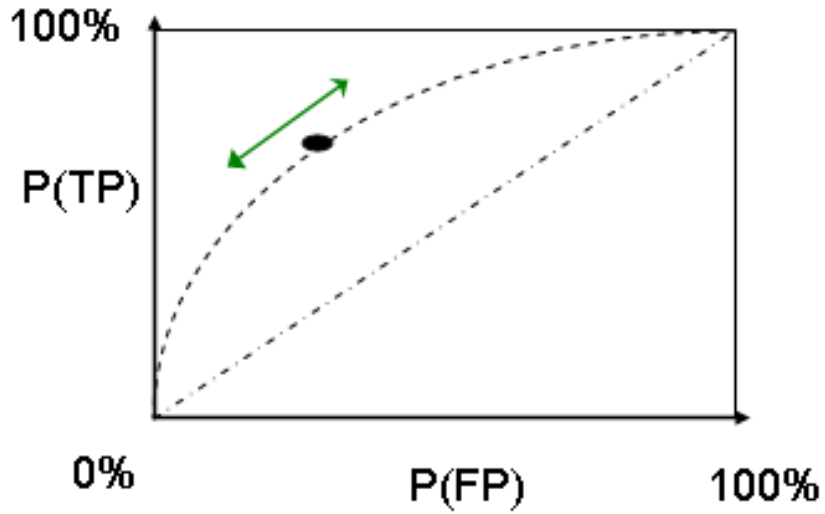
Seleção do modelo: repetir um número desejado de vezes

1. **escolher os parâmetros de treinamento**
2. **realizar treinamento e estimação de erro da validação cruzada.**

Curva ROC (Receiving Operating Characteristic)



TP	FP
FN	TN
1	1



Curva ROC (Receiving Operating Characteristic)

Curvas ROC podem ser desenhadas quando se tem um classificador binário com algum parâmetro ordenado; por exemplo, quando aumentar o valor desse parâmetro implica em aumentar a quantidade de exemplos aceitos (isso aumenta taxa de TP, mas também aumenta taxa de FP).

Olhando a curva, pode-se escolher o parâmetro ótimo.

Ou então, no caso de dois classificadores, pode-se escolher aquele com a melhor curva.

AUC — área sob a curva ROC

Como usar curva ROC para avaliar classificadores?

- Desempenho de um classificador é expresso pela área sob a curva ROC (AUC).
- Logo, para comparar classificadores binários, pode-se comparar as respectivas AUC.
- Observe que, se a prob de verdadeiros-positivos é igual à prob. de falsos-positivos, a curva ROC coincide com a diagonal (pior desempenho possível). Por outro lado, se a prob. de verdadeiros-positivos é 1 quando a prob. de falsos-positivos é zero, temos a situação em que a classificação está 100 % correta e a área sob a curva ROC é 1.
- Escolher o classificador com a maior AUC.