MAC 0460 / 5832

Aprendizagem Computacional
Modelos, algoritmos e aplicações

Nina Hirata (nina@ime.usp.br)
Sala 6 - bloco C

Monitor: Igor

Aula 7 (2012)

Classificador / função-alvo

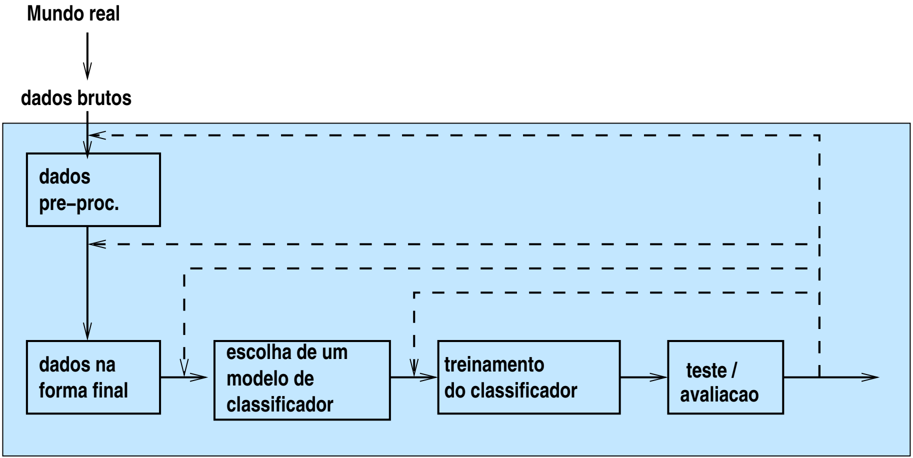
Na prática

não temos conhecimento das distribuições de probabilidade $P(X, y)$**,**

nem conhecemos a função alvo $f : \mathcal{X} \rightarrow Y$

Geralmente temos:

- **uma amostra do espaço** $\mathcal{X} \times Y$
- **conhecimentos a priori (sobre o domínio dos dados)**



Fluxo de um processo de treinamento de classificador

Treinamento de classificador

Estimar/inferir

as distribuições de probabilidade, ou

função-alvo

a partir das **amostras e conhecimentos a priori** disponíveis.

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
- robustez
- etc

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 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

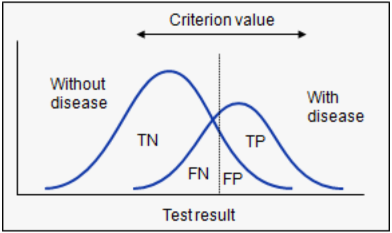
Erros em problemas de classificação binária

Classes

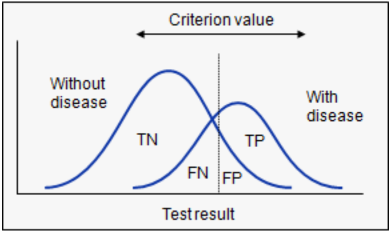
- POSITIVO
- NEGATIVO

Quatro possíveis diagnósticos:

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



Erros em problemas de classificação binária



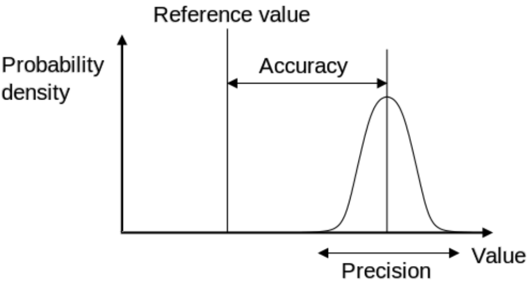
Sensibilidade
(Prob. verdadeiro-positivo)

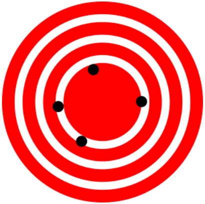
$$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$$

Especificidade
(Prob. verdadeiro-negativo)

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

Acurácia e precisão





(alta acurácia, baixa precisão)

(alta precisão, baixa acurácia)

Fonte das imagens: Wikimedia Commons

Erro de um classificador

Seja $f(\mathbf{x})$ a classificação e y a classe-alvo. O **erro de classificação** pode ser caracterizado através de uma **função de perda**

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

O erro (verdadeiro) do classificador f é dado pelo valor esperado:

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

calculado com respeito à distribuição D no espaço de características.

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

Erro de um classificador

Para a função de perda zero-um, temos

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

Isto é a probabilidade de classificação incorreta.

- **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 D não é conhecida; temos apenas uma amostra dos dados.**

- **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 D não é conhecida; temos apenas uma amostra dos dados.**

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

  O que podemos dizer dessas estimativas?

Treinamento e avaliação

- O espaço de características X possui uma distribuição de probabilidade D (desconhecida, em geral)
- a classe dos objetos $\mathbf{x} \in X$ é definida por uma distribuição conjunta $p(\mathbf{x}, y)$ ($y \in Y = \{1, 2, \ldots, c\}$)
- um classificador é um mapemento do tipo $f : X \to Y$
- nos problemas de classificação supervisionada, temos uma amostra de exemplos pré-classificados

$$A = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$$

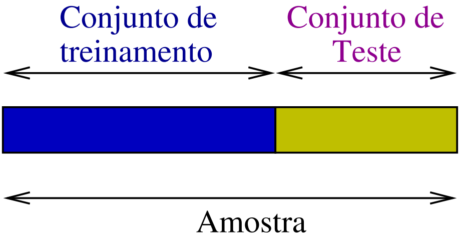
Treinamento, validação e teste

Na prática a amostra é dividida em três partes:

- **treinamento** - para ajustar parâmetros
- **validação** - para controlar o processo de ajuste
- **teste** - verificar desempenho do classificador obtido

Holdout (1)

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



Holdout (2)

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

Holdout (2)

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

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 estimado de** g (com respeito a um conjunto de teste T com n elementos):

$$Erro_T(g) = \frac{1}{n} \sum_{\mathbf{x}_i \in T} \delta(y_i, g(\mathbf{x}_i))$$

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

(proporção de exemplos classificados incorretamente por g)

Erro de treinamento

Erro de treinamento (ou resubstitution error): erro calculado sobre o conjunto de treinamento $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$

$$Erro_S(g) = \frac{1}{n} \sum_{i=1}^{n} \delta(y_i, g(\mathbf{x}_i))$$

na qual δ é a função delta de Kronecker dada por

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

$Erro_S(g)$ **= 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

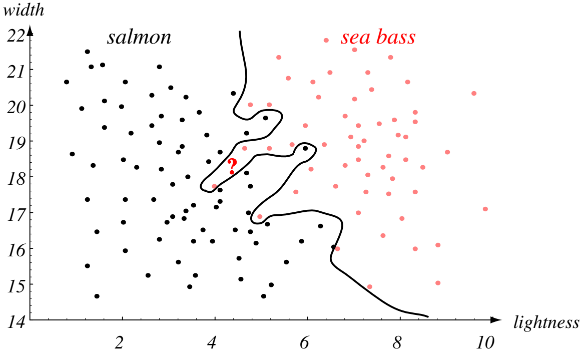


FIGURE 1.5. Overly complex models for the fish will lead to decision boundaries that are complicated. While such a decision may lead to perfect classification of our training samples, it would lead to poor performance on future patterns. The novel test point marked **?** is evidently most likely a salmon, 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

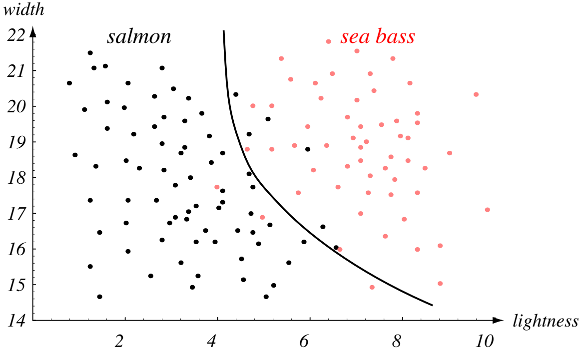


FIGURE 1.6. The decision boundary shown might represent the optimal tradeoff between performance on the training set and simplicity of classifier, thereby giving the highest accuracy on new patterns. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001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

Erro de teste (amostral)

Erro amostral (ou holdout): erro estimado sobre um conjunto de teste T, independente do conjunto de treinamento S.

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

Erro de teste (amostral)

Erro amostral (ou holdout): erro estimado sobre um conjunto de teste T, independente do conjunto de treinamento S.

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

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)**

Valiadação 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**

- **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_i, T_i).**

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

Validação cruzada (2)

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

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

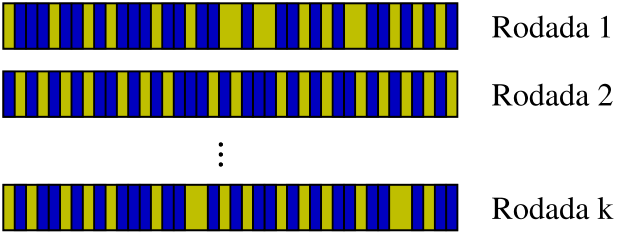
O erro de validação cruzada é dado pela média dos erros $Erro_{T_i}$:

$$Erro = \frac{1}{k} \sum_{i=1}^{k} Erro_{T_i}$$

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.

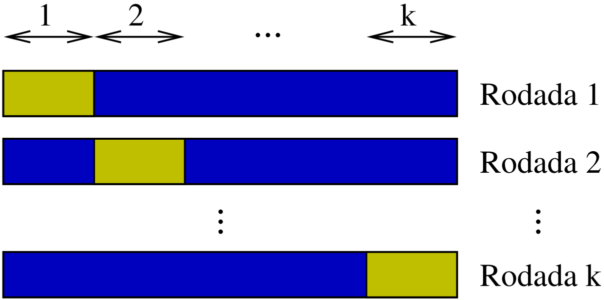


Conjunto de treinamento: exemplos em azul
Conjunto de teste: exemplos em amarelo

Validação cruzada (4) – *k*-fold cross validation

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

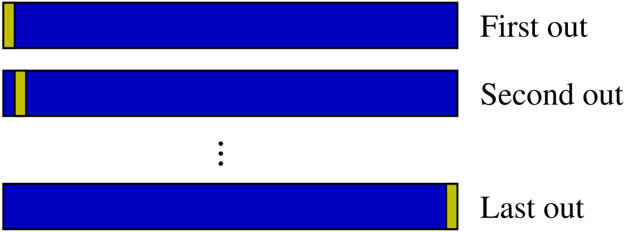
Repetir k rodadas de treinamento, deixando alternadamente uma das partes para teste em cada rodada



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 teste em cada rodada)



Geralmente usado quando a amostra é pequena.

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 B **conjuntos de treinamento,** $Z_1, Z_2, \ldots, Z_B$, todos com tamanho n e obtidos por reamostragem com reposição

Boostrap e cálculo de erro

Sejam B **conjuntos de treinamento,** $Z_1, Z_2, \ldots, Z_B$, 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_b).

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 teste.

Boostrap e cálculo de erro

Sejam B conjuntos de treinamento, $Z_1, Z_2, \ldots, Z_B$, 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_b).

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 teste.

O erro bootstrap tende a subestimar o verdadeiro erro.

Leave one out Bootstrap

Alternativa:

usar o leave one out Bootstrap error, dado por:

$$Erro_{Boot} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|C^{-i}|}\sum_{b\in C^{-i}}\delta(g_b(\mathbf{x}_i), y_i)$$

na qual C^{-i} é o conjunto de índices b em $\{1, 2, \ldots, B\}$ cuja amostra Z_b não contém o exemplo $\mathbf{x}_i$.

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

Leave one out Bootstrap

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

Leave one out Bootstrap

- **Cada conjunto de treinamento** Z_b tem n exemplos, mas como foi obtido por reamostragem com reposição, **contém apenas aproximadamente** 63% **dos exemplos** do conjunto original (o correspondente conjunto de teste 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_b tem n exemplos, mas como foi obtido por reamostragem com reposição, **contém apenas aproximadamente** 63% **dos exemplos** do conjunto original (o correspondente conjunto de teste 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.

Pontos pendentes ...

O quanto pode-se confiar nas estimativas de erro?

Isso relaciona-se com a escolha do “melhor” classificador

E os problemas de outro tipo, que não são de classificação??

Intervalo de confiança para a estimativa de erro

O que podemos dizer sobre a estimativa $Erro_T(g)$?
O quão preciso ele é?

Fazendo algumas aproximações, podemos dizer que o $Erro(g)$, encontra-se com $N\%$ de probabilidade no intervalo

$$Erro_T(g) \pm z_N \sqrt{\frac{Erro_T(g)(1 - Erro_T(g))}{n}}$$

Valores de z_N para diferentes valores de N:

Nível de confiança $N\%$	50%	68%	80%	90%	95%	98%	99%
Constante z_N	0.67	1.00	1.28	1.64	1.96	2.33	2.58

Como foi determinado o intervalo de confiança?

- suponha $p = Erro(g)$
- suponha que as n amostras no conjunto de teste são uma v.a. i.i.d
- seja r a quantidade de amostras de T classificadas incorretamente por g
- então $Erro_T(g) = r/n$
- r/n pode ser vista como um estimador de p
- Quando calculamos $Erro_T(g)$, podemos considerar que estamos realizando n ensaios de Bernoulli, um para cada $\mathbf{x}_i$ em T
- se $g(\mathbf{x}_i) \neq y_i$, temos um sucesso. Caso contrário, um fracasso.
- p é a probabilidade (desconhecida) de sucesso — que queremos estimar.
- r (número de sucessos) segue uma distribuição binomial com parâmetros n e p.

- r (número de sucessos) segue uma **distribuição binomial** com parâmetros n e p.
- ou seja, $E[r] = np$ e $Var[r] = np(1-p)$
- Logo
$$E\left[\frac{r}{n}\right] = \frac{1}{n}E[r] = p$$
e
$$Var[r/n] = \frac{1}{n^2}VAR[r] = \frac{1}{n}p(1-p)$$
(pois $VAR(ax+b) = a^2 VAR(x)$).
- Note que $Erro_T(g) = r/n$ é um estimador não viciado de p
- não conhecemos p
- usamos r/n no lugar de p
- assim temos desvio padrão
$$DP[r/n] \approx \sqrt{\frac{Erro_T(g)\,(1 - Erro_T(g))}{n}}$$
- a distribuição de $Erro_T(g)$ é uma binomial com média p e desvio padrão DP acima

- $Erro_T(g) = r/n$ possui uma distribuição binomial com média p e desvio padrão DP acima
- $Erro_T(g) = r/n$ estima bem $p = Erro(g)$?
- Calculemos um **intervalo de confiança** para o estimador $Erro_T(g) = r/n$
- Para simplificar, aproximamos a binomial por uma normal
- usando a tabela de coeficientes para intervalos de confiança de uma normal padrão, temos então que $Erro(g)$ está, com $N\%$ de probabilidade no intervalo

$$Erro_T(g) \pm z_N \sqrt{\frac{Erro_T(g)(1 - Erro_T(g))}{n}}$$

Comparação de dois classificadores

Sejam g_1 e g_2 dois classificadores. **Qual escolher?**

Comparação de dois classificadores

Sejam g_1 e g_2 dois classificadores. **Qual escolher?**

- Calcular os erros $Erro_{T_1}(g_1)$ e $Erro_{T_2}(g_2)$ sobre conjuntos de teste (não necessariamente iguais).

Comparação de dois classificadores

Sejam g_1 e g_2 dois classificadores. **Qual escolher?**

- Calcular os erros $Erro_{T_1}(g_1)$ e $Erro_{T_2}(g_2)$ sobre conjuntos de teste (não necessariamente iguais).
- Escolher o de menor erro?

Comparação de dois classificadores

Sejam g_1 e g_2 dois classificadores. **Qual escolher?**

- Calcular os erros $Erro_{T_1}(g_1)$ e $Erro_{T_2}(g_2)$ sobre conjuntos de teste (não necessariamente iguais).
- Escolher o de menor erro?
- E se os valores forem próximos?

Comparação de dois classificadores

Sejam g_1 e g_2 dois classificadores. **Qual escolher?**

- Calcular os erros $Erro_{T_1}(g_1)$ e $Erro_{T_2}(g_2)$ sobre conjuntos de teste (não necessariamente iguais).
- Escolher o de menor erro?
- E se os valores forem próximos?
- Escolher aquele com menor intervalo de confiança?

Comparação de dois classificadores

Considerar a diferença

$$\hat{d} = Erro_{T_1}(g_1) - Erro_{T_2}(g_2)$$

como estimador para a diferença verdadeira $d = Erro(g_1) - Erro(g_2)$.

Comparação de dois classificadores

Considerar a diferença

$$\hat{d} = Erro_{T_1}(g_1) - Erro_{T_2}(g_2)$$

como estimador para a diferença verdadeira $d = Erro(g_1) - Erro(g_2)$.

Aproximando a distribuição de cada um deles por uma normal, a distribuição de $\hat{d}$ pode ser aproximada por uma normal (pois a diferença de duas normais é uma normal).

De forma similar ao caso anterior, um **intervalo de confiança** para $\hat{d}$, é

$$\hat{d} \pm \sqrt{\frac{Erro_{T_1}(g_1)(1 - Erro_{T_1}(g_1))}{n_1} + \frac{Erro_{T_2}(g_2)(1 - Erro_{T_2}(g_2))}{n_2}}$$

Melhor parar por aqui por hoje ....