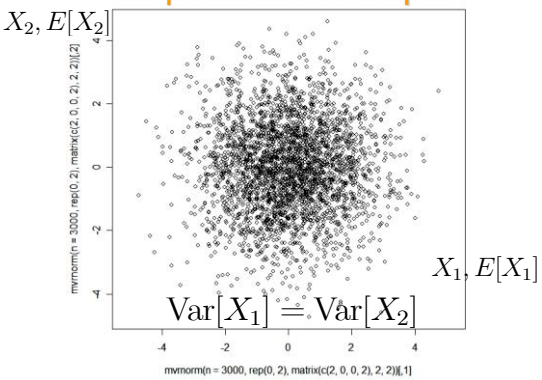
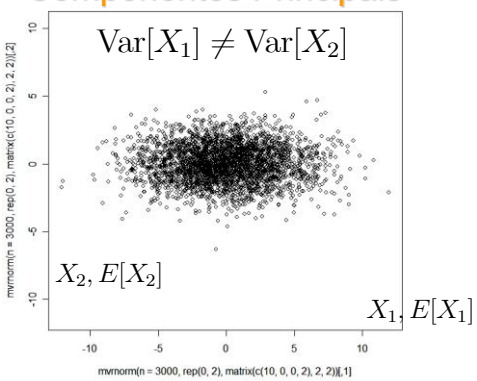
Aprendizado estatístico e computacional

Análise de componentes principais ou Principal components analysis

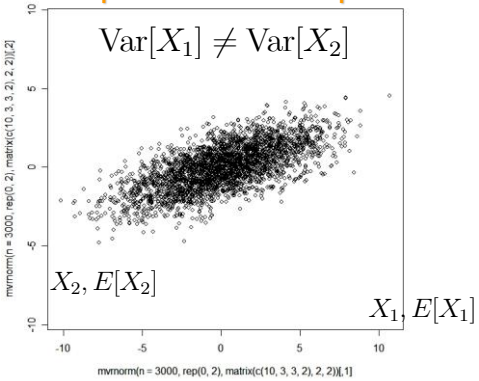
Componentes Principais



Componentes Principais



Componentes Principais



Componentes Principais

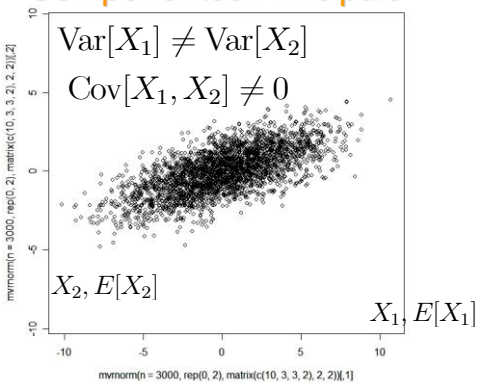
- Variáveis independentes

$$\mathrm{Var}[X_k] = \frac{\sum_{i=1}^{n}(X_k^i - E[X_k])(X_k^i - E[X_k])}{n-1}$$

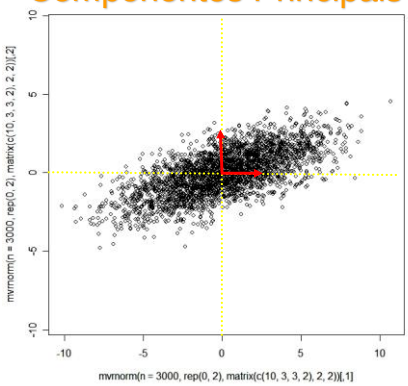
- Variáveis dependentes

$$\mathrm{Cov}[X_l, X_k] = \frac{\sum_{i=1}^{n}(X_l^i - E[X_l])(X_k^i - E[X_k])}{n-1}$$

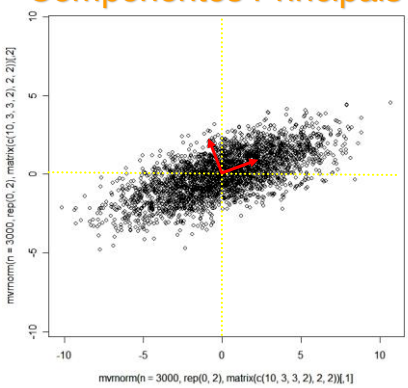
Componentes Principais



Componentes Principais



Componentes Principais



Componentes Principais

- Matriz de covariância

$$\Sigma_{\mathbf{X}} = E[(\mathbf{X} - \mu)(\mathbf{X} - \mu)^T]$$

$$\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$$

$$\mu = \{\overline{X}_1, \overline{X}_2, \ldots, \overline{X}_n\}$$

Componentes Principais

- Transformação de Karhunen-Loève (KL)

$$\Sigma_{\mathbf{Y}} = A^T \Sigma_{\mathbf{X}} A = \Lambda$$

- Assim, temos um problema de diagonalização de uma matriz positiva, que envolve achar os autovalores e os autovetores de $\Sigma_{\mathbf{X}}$

Componentes Principais

- Da álgebra linear,

$$A\mathbf{x} = \lambda\mathbf{x}$$

- Se A é uma matriz simétrica quadrada (n x n), ela tem n pares de autovalores e autovetores

Componentes Principais

$$A = \begin{bmatrix} 1 & -5 \\ -5 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & -5 \\ -5 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix} = 6 \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}$$

- Para λ = 6, o autovetor é $\begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}$

Componentes Principais

$$A = \begin{bmatrix} 1 & -5 \\ -5 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & -5 \\ -5 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = 4 \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$

- Para λ = 4, o autovetor é $\begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$

Componentes Principais

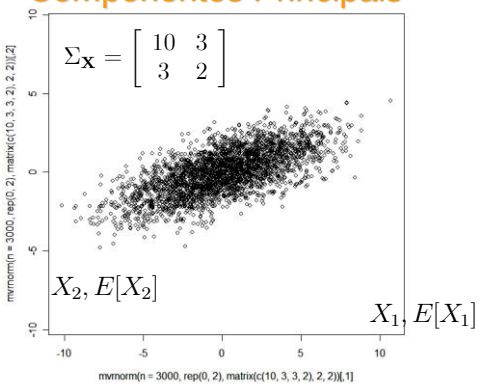
- Da álgebra linear, os autovalores são a solução da equação:

$$|A - \lambda\mathbf{I}| = 0$$

- Ou, no nosso caso:

$$|\Sigma_{\mathbf{X}} - \lambda\mathbf{I}| = 0$$

Componentes Principais



Componentes Principais

- Exemplo:

$$\begin{vmatrix} 10-\lambda & 3 \\ 3 & 2-\lambda \end{vmatrix} = (10-\lambda)(2-\lambda) - 9 = 0$$

$$\lambda^2 - 12\lambda + 11 = 0$$

$$\lambda_1 = 11 \qquad \lambda_2 = 1$$

Componentes Principais

- Para encontrar o autovetor correspondente, aplica-se a equação:

$$A\mathbf{x} = \lambda\mathbf{x}$$

- Substituindo-se os valores de lambda

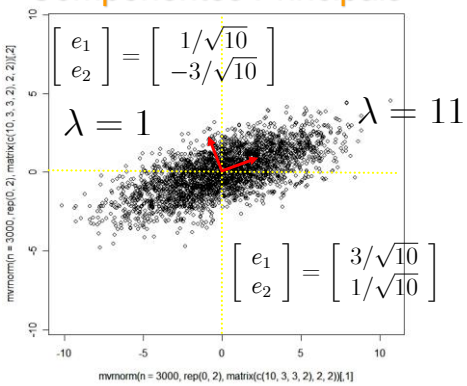
Componentes Principais

$$\begin{bmatrix} 10 & 3 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 11 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

$$\mathbf{e} = \mathbf{x}/\sqrt{\mathbf{x}'\mathbf{x}} \qquad \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} 3/\sqrt{10} \\ 1/\sqrt{10} \end{bmatrix}$$

$$\begin{bmatrix} 10 & 3 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} 1/\sqrt{10} \\ -3/\sqrt{10} \end{bmatrix}$$

Componentes Principais



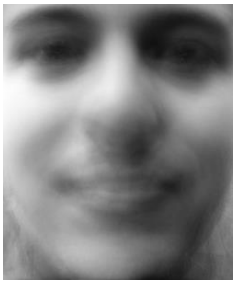
Componentes Principais

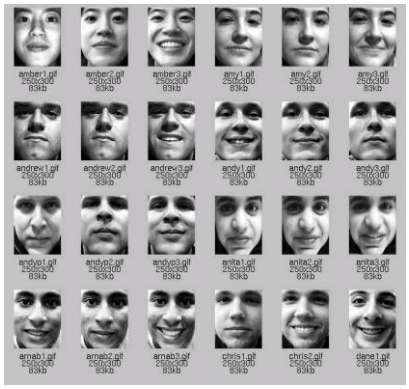
- Possíveis aplicações:
  - Compressão de imagens
  - Representação de imagens
  - Classificação
- Eigenfaces:
  - Kirby and Sirovich
  - Pentland, Turk, Moghaddam, and Starner

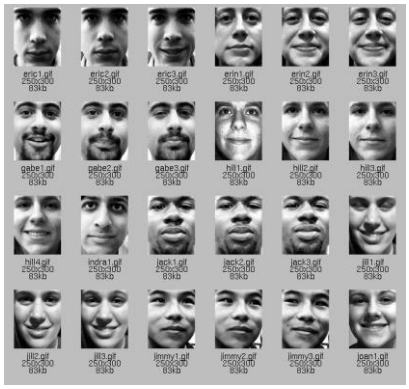
Eigenfaces

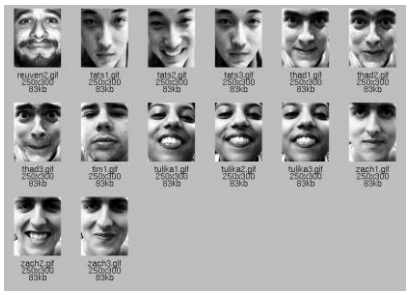
- Recortam-se várias faces de modo que fiquem os olhos e o queixo
- Converte-se o recorte em um vetor
- Empacota-se os vetores como colunas de uma matriz
- Adicionam-se zeros suficientes para a matriz ficar quadrada
- Calculam-se os autovalores e os autovetores da matriz

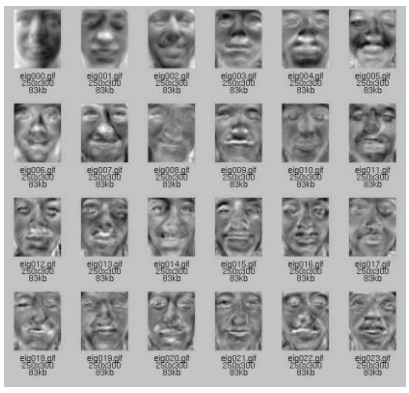
Eigenfaces

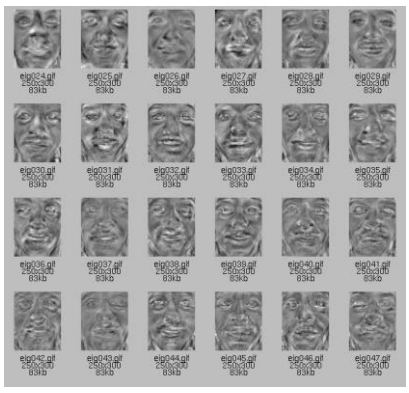


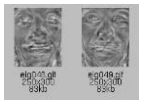




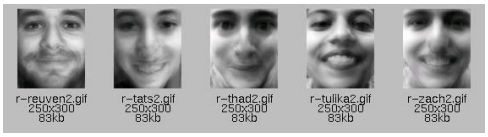


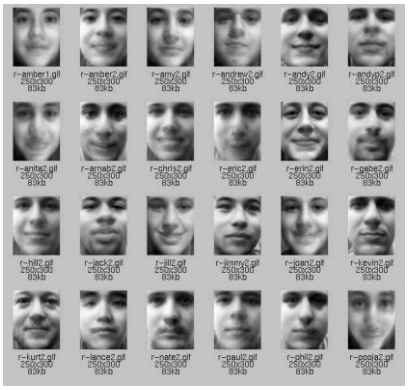




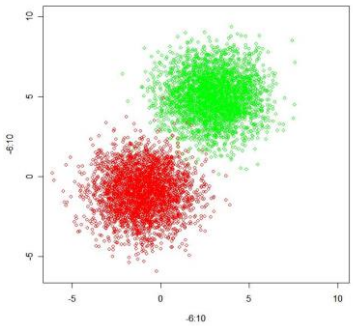


Algumas faces reconstruídas

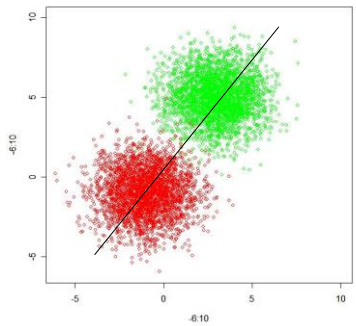




Classificação usando PCA - idéia



Classificação usando PCA - idéia



Classificação usando PCA - idéia

- Comandos do R
- data1 = mvrnorm(n=3000, c(-1,-1), matrix(c(2,0,0,2),2,2))
- data2 = mvrnorm(n=3000, c(3,5), matrix(c(2,0,0,2),2,2))
- points(data2,col="green")
- points(data1,col="red")

Classificação usando PCA - idéia

- str(data1)
- num [1:3000, 1:2] -1.996 -0.387 -0.908 -0.564 -1.907 ...
- str(data2)
- num [1:3000, 1:2] 2.82 4.01 2.51 1.75 2.32
- datat=rbind(data1,data2)
- str(datat)
- num [1:6000, 1:2] -1.996 -0.387 -0.908 -0.564 -1.907 ...

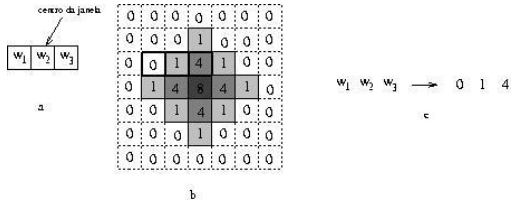
Classificação usando PCA - idéia

- Exercício – usando o comando princomp do pacote MASS (ou equivalente de outro pacote, ou feito à mão), calcule as componentes principais do dataset
- Usando o comando abline (ou equivalente), desenhe uma linha correspondente à direção da componente principal de maior autovalor

Aprendizado estatístico e computacional

Análise de aglomerados

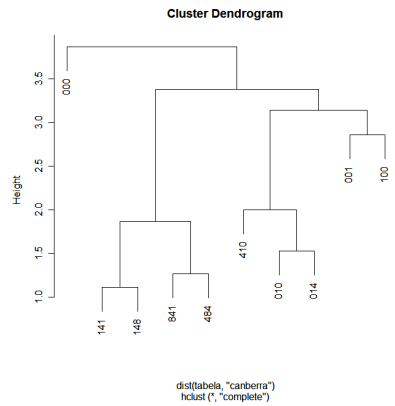
- Exemplo usando o valor da imagem



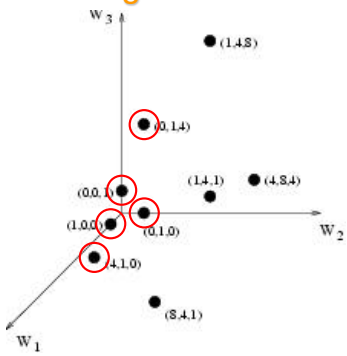
Análise de aglomerados

- Tomando como características os valores vistos através de W, mais o número de vezes que eles apareceram, formamos o conjunto de dados de treinamento

W_1	W_2	W_3	N
0	0	0	14
0	0	1	3
0	1	0	4
1	0	0	4
0	1	4	3
4	1	0	3
1	4	1	2
1	4	8	1
8	4	1	1
4	8	4	1



Análise de aglomerados



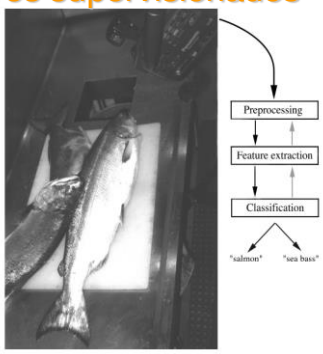
Análise de aglomerados

- Retornando ao conjunto de dados, teremos uma nova coluna com os rótulos das regiões

W_1	W_2	W_3	N	Label
0	0	0	14	Fundo
0	0	1	3	Borda
0	1	0	4	Borda
1	0	0	4	Borda
0	1	4	3	Borda
4	1	0	3	Borda
1	4	1	2	Figura
1	4	8	1	Figura
8	4	1	1	Figura
4	8	4	1	Figura

Classificadores supervisionados

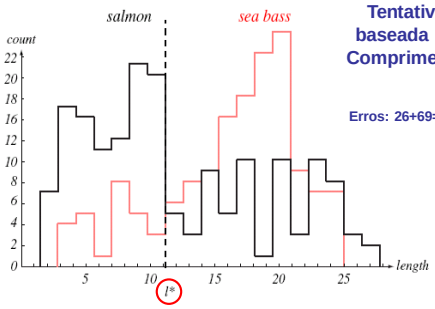
Um problema difícil de visão computacional é a classificação de objetos.



Classificadores supervisionados

- A construção de classificadores também depende da escolha de bons atributos
- Depende da quantidade de exemplos de treinamento (para a construção do classificador)
- Depende da qualidade desses exemplos, ie., o quão fiéis os exemplos são em relação a distribuição de sua população

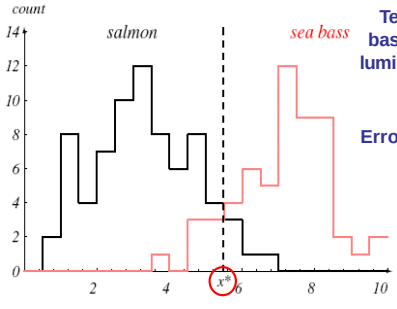
Classificadores supervisionados



Tentativa baseada no Comprimento

Erros: 26+69=95

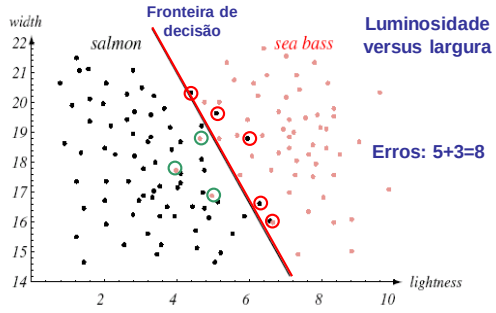
Classificadores supervisionados



Tentativa baseada na luminosidade

Erros: 7+5=12

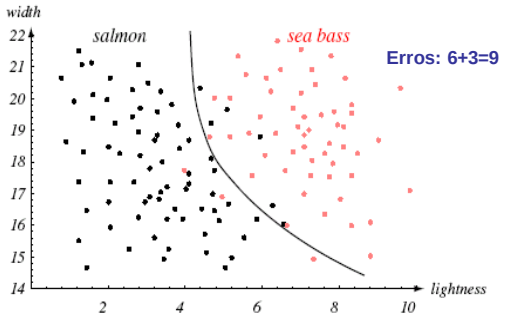
Classificadores supervisionados



Classificadores supervisionados

- Muitas vezes temos mais atributos que são necessários
- Nem sempre é fácil escolher e combinar bons atributos
- Quanto mais atributos, em geral, mais fácil encontrar uma separação natural das classes
- Porém, perde-se robustez. Também com o aumento da complexidade do operador

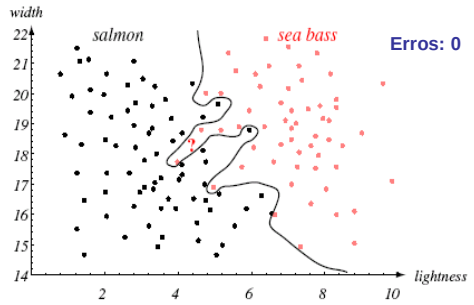
Classificadores supervisionados



Classificadores supervisionados

- Quando a complexidade aumenta demais, o classificador fica ajustado demais aos dados
- Em inglês, temos o famoso problema de "overfitting"
- Nesse caso, o erro nos dados de treinamento é zero, ou próximo
- Os erros nos dados de teste podem ser grandes

Classificadores supervisionados



Teoria de decisão - Bayes

- Abordagem puramente estatística para o problema de classificação
- Supões conhecidas as distribuições de probabilidade envolvidas no problema
- Ex: probabilidade a priori que o pescado é um salmão, ou uma robalo
- Caso essa fosse a única probabilidade conhecida, escolhe-se a classe com a maior priori

Teoria de decisão - Bayes

- Seja $P(w_1)$ a probabilidade a priori que o pescado é uma robalo
- Seja $P(w_2)$ a probabilidade a priori que o pescado é um salmão
- Decida que o pescado é um salmão se

$$P(w_2) > P(w_1)$$

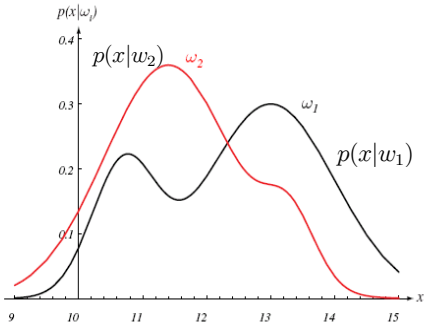
- Ou uma robalo caso contrário.

Teoria de decisão - Bayes

- Porém, esse seria um critério de decisão bem ruim, principalmente se as probabilidades são parecidas
- Caso usemos a luminosidade, podemos considerar a distribuição de probabilidade da luminosidade do pescado dado que ele seja de um certo tipo

$$p(x|w)$$

Probabilidade condicional



Teoria de decisão - Bayes

- Sabendo que:

$$p(w_j, x) = P(w_j|x)p(x) = p(x|w_j)P(w_j)$$

- Manipulando, temos a Fórmula de Bayes

$$P(w_j|x) = \frac{p(x|w_j)P(w_j)}{p(x)}$$

- onde

$$p(x) = \sum_{j=1}^{2} p(x|w_j)P(w_j)$$

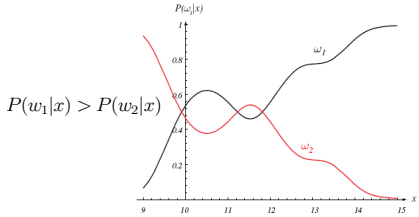
Teoria de decisão - Bayes

- Em português, ela poderia ser escrita como:
- Posteriori=verossimilhança*priori/evidência

$$P(w_j|x) = \frac{p(x|w_j)P(w_j)}{p(x)}$$

Teoria de decisão - Bayes

- Assim, se observamos x podemos transformar a probabilidade a priori $P(w_j)$ na probabilidade a posteriori $P(w_j|x)$



$P(w_1|x) > P(w_2|x)$

$P(w_1) = \frac{2}{3}$

$P(w_2) = \frac{1}{3}$

Teoria de decisão - Bayes

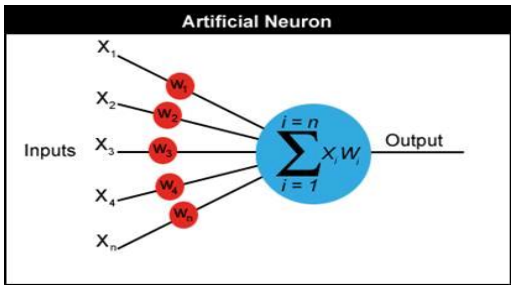
- A decisão poderia ser tomada por:
- se $P(w_2, x) > P(w_1, x)$
- Escolha w_2 senão, escolha w_1
- Isso minimiza o erro:

$$P(\text{erro}|x) = \begin{cases} P(w_1|x) & \text{Decide-se por } w_2 \\ P(w_2|x) & \text{Decide-se por } w_1 \end{cases}$$

Redes neurais

- Conjunto de métodos de aprendizado (supervisionado, ou não) que teve grande interesse na década de 60 e na década de 90
- A inspiração desse método vem das redes neuronais, do funcionamento de um neurônio e da relação entre eles

Neurônio artificial



Redes neurais

- Chamando de net_j a contribuição de cada neurônio, e **w** o conjunto de pesos para cada entrada,

$$\text{net}_j = \sum_{i=1}^{d} x_i w_{ji} + w_{j0} = \mathbf{w}_j^t \mathbf{x}$$

- A saída do neurônio vai depender, ainda, de uma função de ativação, em geral, não linear

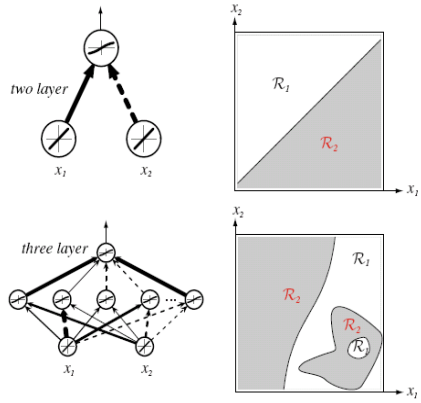
Redes neurais

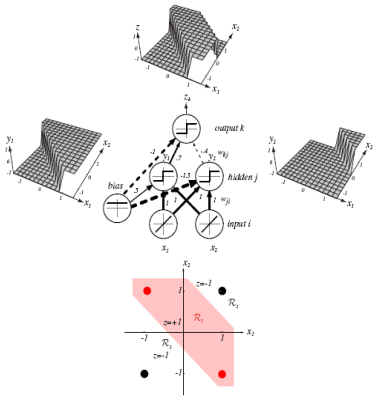
- Para cada neurônio,

$$y_j = f(\text{net})$$

- Exemplo de uma função de ativação baseada no sinal da soma

$$f(\text{net}) = \text{Sgn}(\text{net}) = \begin{cases} 1 & \text{se net} \geq 0 \\ -1 & \text{se net} < 0 \end{cases}$$





Árvores de Decisão

- Indução da árvore
  - Crie uma fila com o conjunto de exemplos rotulados e o nó raiz para representá-los
  - enquanto a fila não estiver vazia
    - **ache uma partição do conjunto que seja mais "pura"**
    - **Associe o critério ao nó atual**
    - **para cada parte, crie um novo nó na árvore**
    - **adicione cada parte impura criada na fila**

Árvores de Decisão

- Perguntas a serem respondidas
  - Partição binária/não-binária?
  - Qual critério de impureza?
  - Folha pura/impura?
  - E se a árvore ficar muito grande? Poda?
  - Se a folha for impura, qual o rótulo dela?
  - Missing data?

Critérios de impureza

- Objetivo: simplicidade
- Seja $i(N)$a impureza do nó *N*

$$i(N) = -\sum_i P(\omega_j) log_2 P(\omega_j)$$

- $P(\omega_j)$: fração dos padrões da categoria j

Critérios de impureza

- Objetivo: simplicidade
- Impureza de informação (entropia)
- Seja $i(N)$a impureza do nó *N*

$$i(N) = -\sum_i P(\omega_j) log_2 P(\omega_j)$$

- $P(\omega_j)$: fração dos padrões da categoria j

Critérios de impureza

- Objetivo: simplicidade
- Impureza de classificação

$$i(N) = 1 - \max_j P(\omega_j)$$

- Menor probabilidade que um padrão seja mal-classificado em *N*

Métodos de indução

- Twoing
  - Usado para múltiplas classes
  - Testa todas as possíveis partições e acha a que melhor separa grupos de *c* categorias segundo um critério de impureza
- Em geral, surpreendentemente, o critério de impureza não parece influenciar o erro do classificador final

Métodos de indução

- Pode-se / deve-se parar a indução?
- Algumas vezes, induzir uma árvore perfeita (sem impurezas) pode levar ao "overfitting"
- Critérios:
  - Quando o decréscimo de impureza for menor que um certo limiar
  - Quando o erro sob validação cruzada for minimizado

Overfitting

- Dado um espaço de hipóteses H, diz-se que uma hipótese h super-ajusta o conjunto de treinamento se existir uma hipótese h´ de H que error(h) < error(h´) sobre os exemplos de treinamento, mas error(h´) < error(h) para o conjunto de todas as instâncias.

Overfitting e DTs

- Por construção, árvores de decisão super-ajustam os dados.
- Soluções:
  - Pare o processo antes
  - Pode a árvore

Overfitting e DTs

- Problema:
  - Qual o critério de parada?
- Por isso, em geral, faz-se a poda.

Poda

- Uma forma de podar é ir das folhas para a raiz, juntando-se folhas do mesmo pai, desde que isso não comprometa demais o erro do classificador.

Poda

- Outra forma:
  - escrever todas as regras representadas pela árvore e,
  - para cada uma delas, remover as pré-condições que resultam em aumento da acurácia. Então,
  - ordenar as regras pela acurácia e usá-las nessa ordem para classificar.

Outros algoritmos

- Classificador de Fisher
- KNN (K-Nearest Neighbor)
- ODT (Oblique DT)
- SVM (Support Vector Machine)
- Deep learning
- Combinação de classificadores
- ...