Aprendizagem Automática

Aula Prática

Métricas de Distância

Classificadores Baseados em Distâncias

𝔾. Marques

1. Métricas de Distâncias
2. Classificadores Baseados em Distâncias
    1. Classificador de distância ao centroide
    2. Classificador dos *k*-vizinhos mais próximos

Métricas de Distância

- Para os humanos, o conceito de distância está intrinsecamente relacionado com a percepção do espaço Euclideano tridimensional, e traduz o grau de proximidade entre objetos, pontos, etc. Do ponto de vista matemático, distância é conceito mais geral e abstrato, que abrange não só a distância Euclideana, bem como um grande número de outras mediadas (métricas).
- Para uma função $\mathcal{D}(\mathbf{x}, \mathbf{y})$, com $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ ser uma métrica de distância entre os vectores d-dimensionais $\mathbf{x}$ e $\mathbf{y}$, necessita de satisfazer as seguintes quatro propriedades:
  1. Não-Negatividade: $\mathcal{D}(\mathbf{x}, \mathbf{y}) \geq 0$
  2. Identidade: $\mathcal{D}(\mathbf{x}, \mathbf{y}) = 0$ se e só se $\mathbf{x} = \mathbf{y}$
  3. Simetria: $\mathcal{D}(\mathbf{x}, \mathbf{y}) = \mathcal{D}(\mathbf{y}, \mathbf{x})$
  4. Desigualdade Triangular: $\mathcal{D}(\mathbf{x}, \mathbf{y}) \leq \mathcal{D}(\mathbf{x}, \mathbf{z}) + \mathcal{D}(\mathbf{z}, \mathbf{y})$ com $\mathbf{z} \in \mathbb{R}^d$

Métricas de Distância

- Algumas métricas de distância habitualmente usadas no contexto de aprendizagem automática (para vectores $\mathbf{x} \in \mathbb{R}^d$, $\mathbf{x} = [x_1, x_2, \ldots, x_d]^\top$).
  - Distância Euclideana:

    $$\mathcal{D}_{\ell_2}(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\| = \left(\sum_{k=1}^{d} (x_k - y_k)^2 \right)^{\frac{1}{2}}$$

  - Distância de City-block ou de Manhattan:

    $$\mathcal{D}_{\ell_1}(\mathbf{x}, \mathbf{y}) = |\mathbf{x} - \mathbf{y}| = \sum_{k=1}^{d} |x_k - y_k|$$

  - Distância de cosseno:

    $$\mathcal{D}_{\cos}(\mathbf{x}, \mathbf{y}) = 1 - \frac{\mathbf{x}^\top \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|} = 1 - \frac{\sum_{k=1}^{d} x_k y_k}{(\sum_{k=1}^{d} x_k^2)^{\frac{1}{2}} (\sum_{k=1}^{d} y_k^2)^{\frac{1}{2}}} = 1 - \cos(\theta)$$

    θ: ângulo formado pelos dois vectores

Métricas de Distância

- Métricas de distância são uma ferramenta essencial para diversos tópicos de aprendizagem automática, entre os quais técnicas de regressão, classificação, modelos probabilísticos, e métodos de agrupamentos.
- Matrizes de distância:
  - Dado um conjunto de N vectores, $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, com $\mathbf{x} \in \mathbb{R}^d$, a matriz de distâncias é uma matriz quadrada de $N \times N$ em que cada elemento (linha i, coluna j) é a distância, $\mathcal{D}(\mathbf{x}_i, \mathbf{x}_j)$, entre os vectores $\mathbf{x}_i$ e $\mathbf{x}_j$.
  - Em classificação, a matriz de distâncias dos pontos ordenados por classe permite ter uma percepção visual da separabilidade entre classes.
  - A matriz de distâncias permite igualmente ter uma ideia geral de qual métrica de distância e/ou qual tipo de pré-processamento de dados pode ser mais adequados.

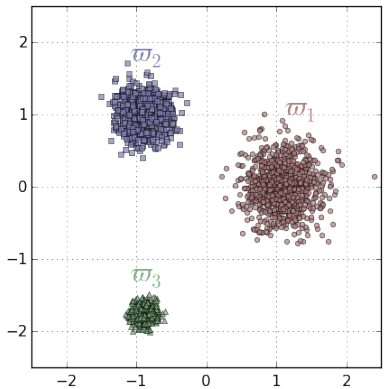
Métricas de Distância

Exemplo com dados sintéticos `LAB2distancias001.p`

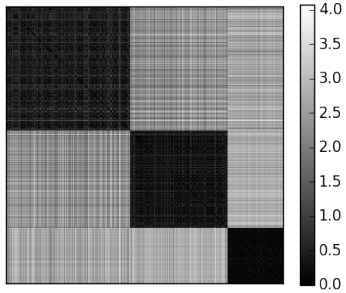
- $\mathcal{X}$, conjunto de pontos 2D dividido em três classes $\Omega = \{\varpi_1, \varpi_2, \varpi_3\}$.
  (Nº total de pontos: $N = 2000$)
- Probabilidades a priori: $p(\varpi_1) = 0.45, p(\varpi_2) = 0.35, p(\varpi_1) = 0.20$
- Probabilidades condicionadas gaussianas: $p(\mathbf{x}|\varpi_i) = \mathcal{N}(\mu_i, \Sigma_i)$

$$\mu_1 = \begin{bmatrix} 1.1 \\ 0.0 \end{bmatrix}, \quad \mu_2 = \begin{bmatrix} -0.9 \\ 1.0 \end{bmatrix}, \quad \mu_3 = \begin{bmatrix} -0.9 \\ -1.7 \end{bmatrix}$$

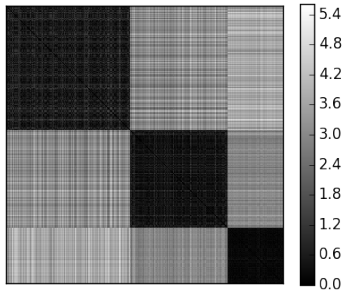
$$\Sigma_i = \begin{bmatrix} \sigma_i^2 & 0 \\ 0 & \sigma_i^2 \end{bmatrix} \text{ com } \sigma_{1,2,3} = [0.3, 0.2, 0.1]$$



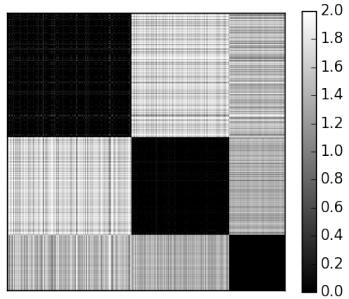
Dados



$\mathcal{D}_{\ell_2}$



$\mathcal{D}_{\ell_1}$



$\mathcal{D}_{\cos}$

Métricas de Distância

Exemplo dígitos MNIST

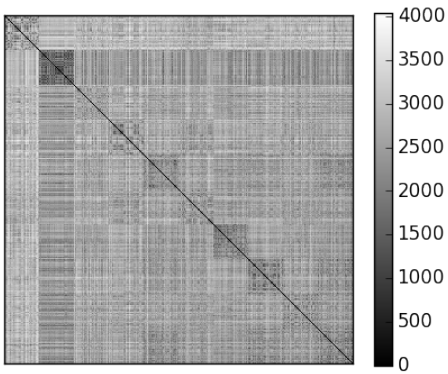
Dados disponibilizados:

- Ficheiro "pickle" `MNISTsmall.p`
- Dados guardados num dicionário.
- Chaves do dicionário:
  - `X`: 15000 imagens de dígitos manuscritos (matriz de 784×15000)
  - `trueClass`: array de 15000 entradas com classe dos dado (classes de `0` a `9`)
  - `foldTrain`: array de 15000 entradas com dados de treino (`True`: treino)
  - `foldTest`: array de 15000 entradas com dados de teste (`True`: teste)
- Cada dígito é uma imagem em tons de cinzento `(uint8)` de 28×28 pixeis, representados vetorialmente: vetores de $784 = 28^2$ dimensões. As primeiras 28 correspondem aos pixeis da 1ª coluna, as segundas 28 dimensões aos da 2ª coluna, e por aí em diante.

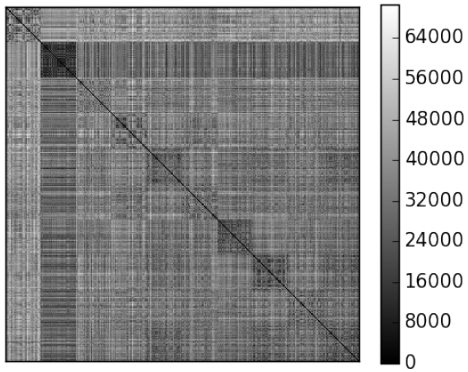
Métricas de Distância

Exemplo dígitos MNIST

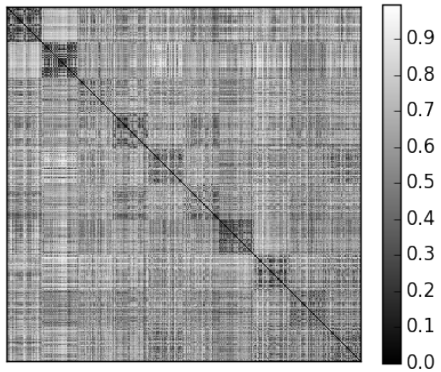
- Selecionados os 200 primeiros exemplos de cada dígito de treino
- Nº total de pontos: $N = 2000$
- Dados em "bruto": vectores de 784×1



$\mathcal{D}_{\ell_2}$



$\mathcal{D}_{\ell_1}$



$\mathcal{D}_{\cos}$

- Em Python usar módulo `scipy.spatial.distance`:

```
# X - matriz de dígitos (784×2000)
>>> import scipy.spatial.distance as spd
# usar 'euclidean','cityblock', e 'cosine'
>>> D=spd.squareform(spd.pdist(X.T,'euclidean'))
```

Classificadores Baseados em Distâncias

Várias técnicas de classificação são direta ou indiretamente baseadas em medidas de distância. Dois dos métodos de classificação mais simples são:

- **Classificador de distância ao centroide:**
  Este método classifica uma nova observação (novo vector) baseado na distâncias às médias (centroides) das classes no conjunto de treino. A classe atribuída é a do centroide que estiver mais próximo do vector.
- **Classificador do k vizinhos mais próximos (k-NN):**
  Este método classifica uma nova observação baseado nas classes dos k vizinhos mais próximos do conjunto de treino. A classe atribuída por votação - classe maioritária nos k vizinhos.

Classificadores Baseados em Distâncias

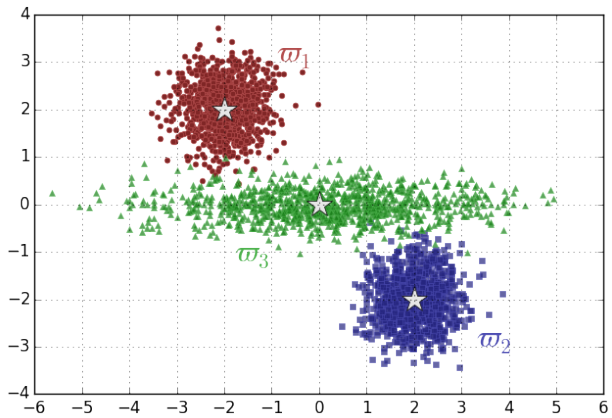
Classificador de Distância ao Centroide

Dados Sintéticos (`LAB2distancias002.p`)

- $\mathcal{X}$, conjunto de pontos 2D dividido em três classes $\Omega = \{\varpi_1, \varpi_2, \varpi_3\}$.
  (Nº total de pontos: $N = 3000$)
- Probabilidades a priori: $p(\varpi_1) = p(\varpi_2) = p(\varpi_3)$
- Probabilidades condicionadas gaussianas: $p(\mathbf{x}|\varpi_i) = \mathcal{N}(\mu_i, \Sigma_i)$

$$\mu_1 = \begin{bmatrix} -2 \\ +2 \end{bmatrix}, \quad \mu_2 = \begin{bmatrix} +2 \\ -2 \end{bmatrix}, \quad \mu_3 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\Sigma_1 = \begin{bmatrix} \frac{1}{4} & 0 \\ 0 & \frac{1}{4} \end{bmatrix} \quad \Sigma_2 = \begin{bmatrix} \frac{1}{4} & 0 \\ 0 & \frac{1}{4} \end{bmatrix} \quad \Sigma_3 = \begin{bmatrix} 3 & 0 \\ 0 & \frac{1}{10} \end{bmatrix}$$



Classificadores Baseados em Distâncias

Classificador de Distância ao Centroide

Dados Sintéticos (`LAB2distancias002.p`)

- $\mathcal{X}$, conjunto de pontos 2D dividido em três classes $\Omega = \{\varpi_1, \varpi_2, \varpi_3\}$.
  (Nº total de pontos: $N = 3000$)
- Probabilidades a priori: $p(\varpi_1) = p(\varpi_2) = p(\varpi_3)$
- Probabilidades condicionadas gaussianas: $p(\mathbf{x}|\varpi_i) = \mathcal{N}(\mu_i, \Sigma_i)$

Distância Euclideana:

- Para um conjunto $\mathcal{X}$, a distância Euclideana dum vector $\mathbf{x}$ à média é:
  $\mathcal{D}_{\ell_2}(\mathbf{x}, \boldsymbol{\mu}_\mathbf{x}) = \sqrt{(\mathbf{x} - \boldsymbol{\mu}_\mathbf{x})^\top (\mathbf{x} - \boldsymbol{\mu}_\mathbf{x})} = \sqrt{(x_1 - \mu_{x_1})^2 + \ldots + (x_d - \mu_{x_d})^2}$

- Classificação:
  - Calcular $\mathcal{D}_{\ell_2}(\mathbf{x}, \boldsymbol{\mu}_i) = \sqrt{(\mathbf{x} - \boldsymbol{\mu}_i)^\top (\mathbf{x} - \boldsymbol{\mu}_i)}$, para $i = 1, 2, 3$
  - $\mathbf{x} \in \hat{\varpi}_j$, se $\mathcal{D}_{\ell_2}(\mathbf{x}, \boldsymbol{\mu}_j) \leq \mathcal{D}_{\ell_2}(\mathbf{x}, \boldsymbol{\mu}_i)$

Classificadores Baseados em Distâncias

Classificador de Distância ao Centroide

Dados Sintéticos (`LAB2distancias002.p`)

- $\mathcal{X}$, conjunto de pontos 2D dividido em três classes $\Omega = \{\varpi_1, \varpi_2, \varpi_3\}$.
  (Nº total de pontos: $N = 3000$)
- Probabilidades a priori: $p(\varpi_1) = p(\varpi_2) = p(\varpi_3)$
- Probabilidades condicionadas gaussianas: $p(\mathbf{x}|\varpi_i) = \mathcal{N}(\mu_i, \Sigma_i)$

Distância Euclideana:

- Para um conjunto $\mathcal{X}$, a distância Euclideana dum vector $\mathbf{x}$ à média é:
  $\mathcal{D}_{\ell_2}(\mathbf{x}, \boldsymbol{\mu}_\mathbf{x}) = \sqrt{(\mathbf{x} - \boldsymbol{\mu}_\mathbf{x})^\top (\mathbf{x} - \boldsymbol{\mu}_\mathbf{x})} = \sqrt{(x_1 - \mu_{x_1})^2 + \ldots + (x_d - \mu_{x_d})^2}$
- Em Python: (ex: cálculo das distâncias à classe ϖ_1)

```
# X matriz com pontos (2×3000), m1= μ1 (2×1)
>>> Xn=X-m1
>>> D1=np.sqrt(np.sum(Xn*Xn,axis=0)) #D1, array de (3000,)
```

Classificadores Baseados em Distâncias

Classificador de Distância ao Centroide

Dados Sintéticos (`LAB2distancias002.p`)

- $\mathcal{X}$, conjunto de pontos 2D dividido em três classes $\Omega = \{\varpi_1, \varpi_2, \varpi_3\}$.
  (Nº total de pontos: $N = 3000$)
- Probabilidades a priori: $p(\varpi_1) = p(\varpi_2) = p(\varpi_3)$
- Probabilidades condicionadas gaussianas: $p(\mathbf{x}|\varpi_i) = \mathcal{N}(\mu_i, \Sigma_i)$

Distância Euclideana:

- Para um conjunto $\mathcal{X}$, a distância Euclideana dum vector $\mathbf{x}$ à média é:
  $\mathcal{D}_{\ell_2}(\mathbf{x}, \boldsymbol{\mu}_\mathbf{x}) = \sqrt{(\mathbf{x} - \boldsymbol{\mu}_\mathbf{x})^\top (\mathbf{x} - \boldsymbol{\mu}_\mathbf{x})} = \sqrt{(x_1 - \mu_{x_1})^2 + \ldots + (x_d - \mu_{x_d})^2}$
- Construir matriz de distâncias a todas as classes e ver qual a menor

```
>>> Dtotal=np.vstack((D1,D2,D3))#Dtotal, matriz de 3×3000
>>> estClass=np.argmin(Dtotal,axis=0)
```

Classificadores Baseados em Distâncias

Classificador de Distância ao Centroide

Dados Sintéticos (`LAB2distancias002.p`)

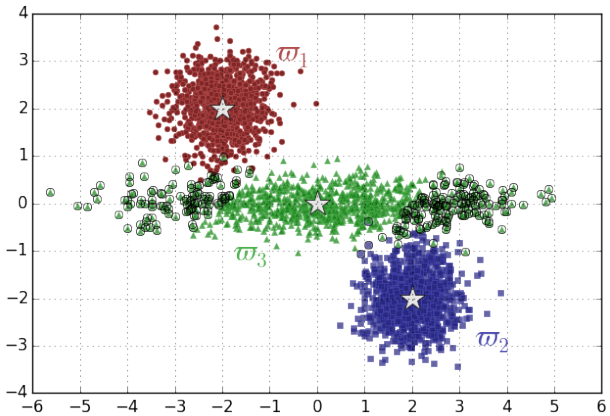
- $\mathcal{X}$, conjunto de pontos 2D dividido em três classes $\Omega = \{\varpi_1, \varpi_2, \varpi_3\}$. (Nº total de pontos: $N = 3000$)
- Probabilidades a priori: $p(\varpi_1) = p(\varpi_2) = p(\varpi_3)$
- Probabilidades condicionadas gaussianas: $p(\mathbf{x}|\varpi_i) = \mathcal{N}(\mu_i, \Sigma_i)$

Classificação:

- Distância Euclideana: $\mathcal{D}_{\ell_2}$
- $\mathbf{x} \in \hat{\varpi}_j$ se $\mathcal{D}_{\ell_2}(\mathbf{x}, \boldsymbol{\mu}_j) \leq \mathcal{D}_{\ell_2}(\mathbf{x}, \boldsymbol{\mu}_i),\ i, j = 1, 2, 3$

Resultados:

$$P = \begin{bmatrix} 1000 & 0 & 0 \\ 0 & 997 & 3 \\ 111 & 148 & 741 \end{bmatrix} \qquad \text{Prob.Erro} = \frac{262}{3000}$$



Classificadores Baseados em Distâncias

Classificador de Distância ao Centroide

Dados Sintéticos (`LAB2distancias002.p`)

- $\mathcal{X}$, conjunto de pontos 2D dividido em três classes $\Omega = \{\varpi_1, \varpi_2, \varpi_3\}$.
  (Nº total de pontos: $N = 3000$)
- Probabilidades a priori: $p(\varpi_1) = p(\varpi_2) = p(\varpi_3)$
- Probabilidades condicionadas gaussianas: $p(\mathbf{x}|\varpi_i) = \mathcal{N}(\mu_i, \Sigma_i)$

Distância de Mahalanobis:

- Para um conjunto $\mathcal{X}$ com média $\boldsymbol{\mu}_\mathbf{x}$ e matriz de convariância $\Sigma_\mathbf{x}$, a distância de Mahalanobis dum vector $\mathbf{x}$ ao conjunto é: $\mathcal{D}_\mathcal{M}(\mathbf{x}, \boldsymbol{\mu}_\mathbf{x}) = \sqrt{(\mathbf{x} - \boldsymbol{\mu}_\mathbf{x})^\top \Sigma_\mathbf{x}^{-1} (\mathbf{x} - \boldsymbol{\mu}_\mathbf{x})}$

- Classificação:
  - Calcular $\mathcal{D}_\mathcal{M}(\mathbf{x}, \boldsymbol{\mu}_i) = \sqrt{(\mathbf{x} - \boldsymbol{\mu}_i)^\top \Sigma_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i)}$, para $i = 1, 2, 3$
  - $\mathbf{x} \in \hat{\varpi}_j$, se $\mathcal{D}_\mathcal{M}(\mathbf{x}, \boldsymbol{\mu}_j) \leq \mathcal{D}_\mathcal{M}(\mathbf{x}, \boldsymbol{\mu}_i)$

Classificadores Baseados em Distâncias

Como calcular transformações do tipo $\mathbf{x}^\top \mathbf{A}\mathbf{x}$ para um conjunto, $\mathbf{X}$, de N vectores de d dimensões?

Exemplo com N pontos a 2 dimensões:

$$\mathbf{x}^\top A\mathbf{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \underbrace{\begin{bmatrix} a_{11}x_1 + a_{12}x_2 \\ a_{21}x_1 + a_{22}x_2 \end{bmatrix}}_{2\times 1} = \underbrace{\begin{matrix} x_1(a_{11}x_1 + a_{12}x_2) \\ + \\ x_2(a_{21}x_1 + a_{22}x_2) \end{matrix}}_{\text{escalar}}$$

$$A\mathbf{X} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1[1] & x_1[2] & \cdots & x_1[N] \\ x_2[1] & x_2[2] & \cdots & x_2[N] \end{bmatrix}$$

$$= \begin{bmatrix} a_{11}x_1[1] + a_{12}x_2[1] & a_{11}x_1[2] + a_{12}x_2[2] & \cdots & a_{11}x_1[N] + a_{12}x_2[N] \\ a_{21}x_1[1] + a_{22}x_2[1] & a_{21}x_1[2] + a_{22}x_2[2] & \cdots & a_{21}x_1[N] + a_{22}x_2[N] \end{bmatrix}$$

- Em NumPy $A\mathbf{X}$ $\Leftrightarrow$ `np.dot(A,X)`

Classificadores Baseados em Distâncias

Como calcular transformações do tipo $\mathbf{x}^\top \mathbf{A}\mathbf{x}$ para um conjunto, $\mathbf{X}$, de N vectores de d dimensões?

Exemplo com N pontos a 2 dimensões:

$$\mathbf{x}^\top A\mathbf{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \underbrace{\begin{bmatrix} a_{11}x_1 + a_{12}x_2 \\ a_{21}x_1 + a_{22}x_2 \end{bmatrix}}_{2\times 1} = \underbrace{\begin{array}{c} x_1(a_{11}x_1 + a_{12}x_2) \\ + \\ x_2(a_{21}x_1 + a_{22}x_2) \end{array}}_{\text{escalar}}$$

- Em `NumPy X*(np.dot(A,X))` $\Leftrightarrow$

$$\begin{bmatrix} x_1[1](a_{11}x_1[1]+a_{12}x_2[1]) & x_1[2](a_{11}x_1[2]+a_{12}x_2[2]) & \cdots & x_1[N](a_{11}x_1[N]+a_{12}x_2[N]) \\ x_2[1](a_{21}x_1[1]+a_{22}x_2[1]) & x_2[2](a_{21}x_1[2]+a_{22}x_2[2]) & \cdots & x_2[N](a_{21}x_1[N]+a_{22}x_2[N]) \end{bmatrix}$$

- Basta somar as colunas e obtém-se os valores $\mathbf{x}_n^\top A\mathbf{x}_n$ para $n = 1, \ldots, N$

Classificadores Baseados em Distâncias

Classificador de Distância ao Centroide

Dados Sintéticos (`LAB2distancias002.p`)

- $\mathcal{X}$, conjunto de pontos 2D dividido em três classes $\Omega = \{\varpi_1, \varpi_2, \varpi_3\}$.
  (Nº total de pontos: $N = 3000$)
- Probabilidades a priori: $p(\varpi_1) = p(\varpi_2) = p(\varpi_3)$
- Probabilidades condicionadas gaussianas: $p(\mathbf{x}|\varpi_i) = \mathcal{N}(\mu_i, \Sigma_i)$

Distância de Mahalanobis:

- Para um conjunto $\mathcal{X}$ com média $\boldsymbol{\mu}_\mathbf{x}$ e matriz de convariância $\Sigma_\mathbf{x}$, a distância de Mahalanobis dum vector $\mathbf{x}$ ao conjunto é: $\mathcal{D}_\mathcal{M}(\mathbf{x}, \boldsymbol{\mu}_\mathbf{x}) = \sqrt{(\mathbf{x} - \boldsymbol{\mu}_\mathbf{x})^\top \Sigma_\mathbf{x}^{-1} (\mathbf{x} - \boldsymbol{\mu}_\mathbf{x})}$
- Em Python: (ex: cálculo das distâncias à classe ϖ_1)

```
# X matriz com pontos (2×3000), SI1= Σ1^-1 (2×2), m1= μ1 (2×1)
>>> Xn=X-m1
>>> D1=np.sqrt(np.sum(Xn*np.dot(SI1,Xn),axis=0))
```

Classificadores Baseados em Distâncias

Classificador de Distância ao Centroide

Dados Sintéticos (`LAB2distancias002.p`)

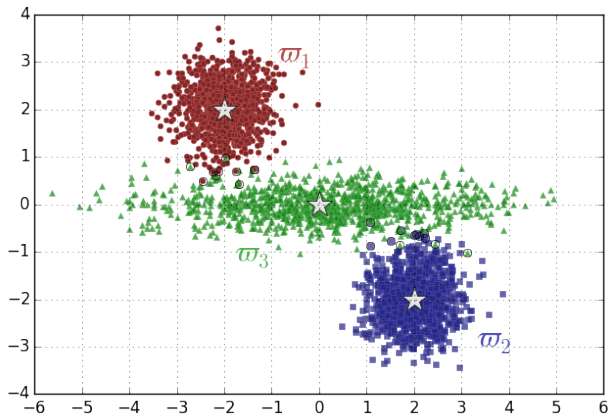
- $\mathcal{X}$, conjunto de pontos 2D dividido em três classes $\Omega = \{\varpi_1, \varpi_2, \varpi_3\}$. (Nº total de pontos: $N = 3000$)
- Probabilidades a priori: $p(\varpi_1) = p(\varpi_2) = p(\varpi_3)$
- Probabilidades condicionadas gaussianas: $p(\mathbf{x}|\varpi_i) = \mathcal{N}(\mu_i, \Sigma_i)$

Classificação:

- Distância de Mahalanobis: $\mathcal{D}_{\mathcal{M}}$
- $\mathbf{x} \in \hat{\varpi}_j$, se $\mathcal{D}_{\mathcal{M}}(\mathbf{x}, \boldsymbol{\mu}_j) \leq \mathcal{D}_{\mathcal{M}}(\mathbf{x}, \boldsymbol{\mu}_i)$, $i, j = 1, 2, 3$

Resultados:

$$P = \begin{bmatrix} 992 & 0 & 8 \\ 0 & 991 & 9 \\ 2 & 3 & 995 \end{bmatrix} \qquad \text{Prob.Erro} = \frac{22}{3000}$$



Classificadores Baseados em Distâncias

Classificador de Distância ao Centroide

Dados Reais: Iris Dataset

- Comandos:

  Importar dados de `scikit-learn`

  ```
  >>> from sklearn import datasets
  ```

  Carregar os dados do dataset "Iris"
  (íris são plantas com flor, vulgarmente designadas por lírios)

  ```
  >>> Iris=datasets.load_iris()
  ```

  `Iris`: variável do tipo `dictionary`, com vários campos:

  ```
  >>> Iris.keys() # ver os campos do dicionário
  ['target_names', 'data', 'target', 'DESCR', 'feature_names']
  ```

- Dados – `X` é um `np.array` de `(150,4)`:

  ```
  >>> X=Iris.data
  ```

- Classe dos dados – `trueClass` é um `np.array` de `(150,)`:

  ```
  >>> trueClass=Iris.target
  ```

Classificadores Baseados em Distâncias

Classificador de Distância ao Centroide

Dados Reais: Iris Dataset

- Variável `X` é uma matriz de 150×4 (transpor matriz para ficar $d \times N$)

```
>>> X=X.T
```

- Calcular os centroides (médias das classes):

```
>>> m0=np.mean(X[:,trueClass==0],axis=1) # array de (4,)
>>> m1=np.mean(X[:,trueClass==1],axis=1)
>>> m2=np.mean(X[:,trueClass==2],axis=1)
```

- Calcular calcular distâncias das 3 médias a todos os pontos:

```
>>> X0=X-m0[:,np.newaxis] #m0, agora com dim. (4,1)
>>> D0=np.sqrt(np.sum(X0*X0,axis=0)) ...
```

- Construir matriz de distâncias (3×150)

```
>>> Dtotal=np.vstack((D0,D1,D2))
```

- Classificar:

```
>>> estClass=np.argmin(Dtotal,axis=0)
```

Classificadores Baseados em Distâncias

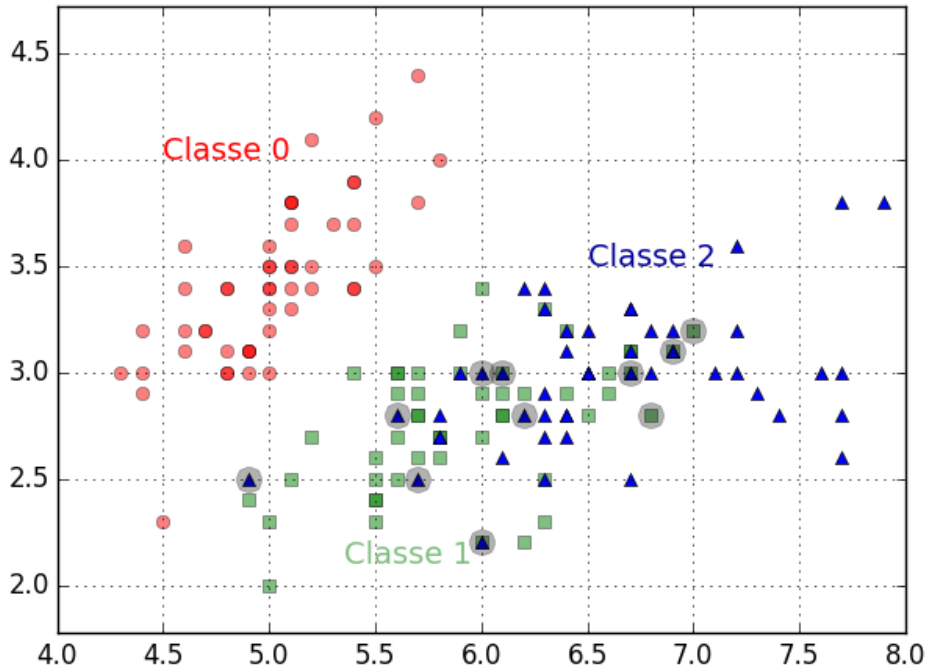
Classificador de Distância ao Centroide

Dados Reais: Iris Dataset

Resultados:

$$P = \begin{bmatrix} 50 & 0 & 0 \\ 0 & 46 & 4 \\ 0 & 7 & 43 \end{bmatrix}$$

Prob. Total de Erro= $\frac{11}{150} \approx 7.33\%$



2 primeiras dimensões dos dados (erros - pts cinza)

- Atenção:Modelo e avaliação estimados com todo o conjunto de dados. Para ter uma medida fidedigna do desempenho, é necessário avaliar o classificador com dados que não foram usados para estimar o modelo. Neste caso, devido à simplicidade do modelo (a classificação é feita com distâncias a 3 médias), a estimativa do desempenho não é tão enviesada como em outros classificadores mais complexos.

Classificadores Baseados em Distâncias

Classificador de Distância ao Centroide

Dígitos manuscritos

- Conjunto de dígitos manuscritos (10 classes).
  - Nº de pontos treino: 1000 pts por classe
  - Nº de pontos teste: 500 pts por classe

Classificação: Distância Euclideana

Matriz de confusão -**Dados de treino**:

$$\begin{bmatrix} 873 & 0 & 8 & 9 & 2 & 62 & 25 & 6 & 14 & 1 \\ 0 & 976 & 3 & 0 & 0 & 14 & 0 & 2 & 3 & 2 \\ 14 & 80 & 763 & 28 & 28 & 10 & 25 & 13 & 32 & 7 \\ 6 & 39 & 27 & 762 & 2 & 69 & 10 & 20 & 39 & 26 \\ 1 & 24 & 7 & 0 & 813 & 5 & 16 & 5 & 8 & 121 \\ 27 & 65 & 6 & 163 & 24 & 646 & 22 & 5 & 16 & 26 \\ 16 & 50 & 22 & 0 & 20 & 23 & 866 & 0 & 3 & 0 \\ 8 & 55 & 6 & 0 & 27 & 3 & 1 & 836 & 7 & 57 \\ 8 & 67 & 16 & 87 & 13 & 47 & 12 & 4 & 710 & 36 \\ 17 & 21 & 9 & 14 & 74 & 8 & 3 & 46 & 15 & 793 \end{bmatrix}$$

Probabilidade Total de Erro= $\frac{1962}{10000} \approx 19.62\%$

Classificadores Baseados em Distâncias

Classificador de Distância ao Centroide

Dígitos manuscritos

- Conjunto de dígitos manuscritos (10 classes).
  - Nº de pontos treino: 1000 pts por classe
  - Nº de pontos teste: 500 pts por classe

Classificação: Distância Euclideana

Matriz de confusão -**Dados de teste**:

$$\begin{bmatrix} 421 & 0 & 5 & 0 & 0 & 50 & 16 & 0 & 7 & 1 \\ 0 & 477 & 1 & 4 & 0 & 13 & 3 & 0 & 2 & 0 \\ 8 & 70 & 331 & 32 & 9 & 2 & 7 & 15 & 24 & 2 \\ 2 & 16 & 3 & 393 & 1 & 46 & 4 & 12 & 16 & 7 \\ 0 & 10 & 2 & 0 & 389 & 4 & 13 & 1 & 4 & 77 \\ 8 & 16 & 2 & 89 & 17 & 322 & 10 & 9 & 11 & 16 \\ 10 & 22 & 21 & 0 & 36 & 33 & 377 & 0 & 1 & 0 \\ 1 & 51 & 9 & 1 & 15 & 2 & 0 & 387 & 4 & 30 \\ 5 & 25 & 4 & 66 & 8 & 30 & 7 & 7 & 322 & 26 \\ 4 & 12 & 8 & 9 & 63 & 10 & 1 & 10 & 3 & 380 \end{bmatrix}$$

Probabilidade Total de Erro= $\frac{1201}{5000} \approx 24.02\%$

Classificadores Baseados em Distâncias

Classificador de Distância ao Centroide

Dígitos manuscritos

- Conjunto de dígitos manuscritos (10 classes).
  - Nº de pontos treino: 1000 pts por classe
  - Nº de pontos teste: 500 pts por classe

Classificação: Distância de Mahalanobis

Matriz de confusão -**Dados de treino**:

$$\begin{bmatrix} 995 & 0 & 1 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 847 & 24 & 3 & 9 & 0 & 0 & 1 & 114 & 2 \\ 0 & 0 & 986 & 12 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 2 & 994 & 0 & 0 & 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 40 & 957 & 0 & 0 & 0 & 1 & 2 \\ 1 & 0 & 0 & 213 & 0 & 783 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 115 & 0 & 4 & 875 & 0 & 6 & 0 \\ 0 & 0 & 1 & 4 & 1 & 0 & 0 & 988 & 6 & 0 \\ 0 & 0 & 1 & 49 & 0 & 0 & 0 & 0 & 950 & 0 \\ 0 & 0 & 0 & 4 & 2 & 0 & 0 & 2 & 10 & 982 \end{bmatrix}$$

Probabilidade Total de Erro= $\frac{643}{10000} \approx 6.43\%$

Classificadores Baseados em Distâncias

Classificador de Distância ao Centroide

Dígitos manuscritos

- Conjunto de dígitos manuscritos (10 classes).
  - Nº de pontos treino: 1000 pts por classe
  - Nº de pontos teste: 500 pts por classe

Classificação: Distância de Mahalanobis

Matriz de confusão -**Dados de teste**:

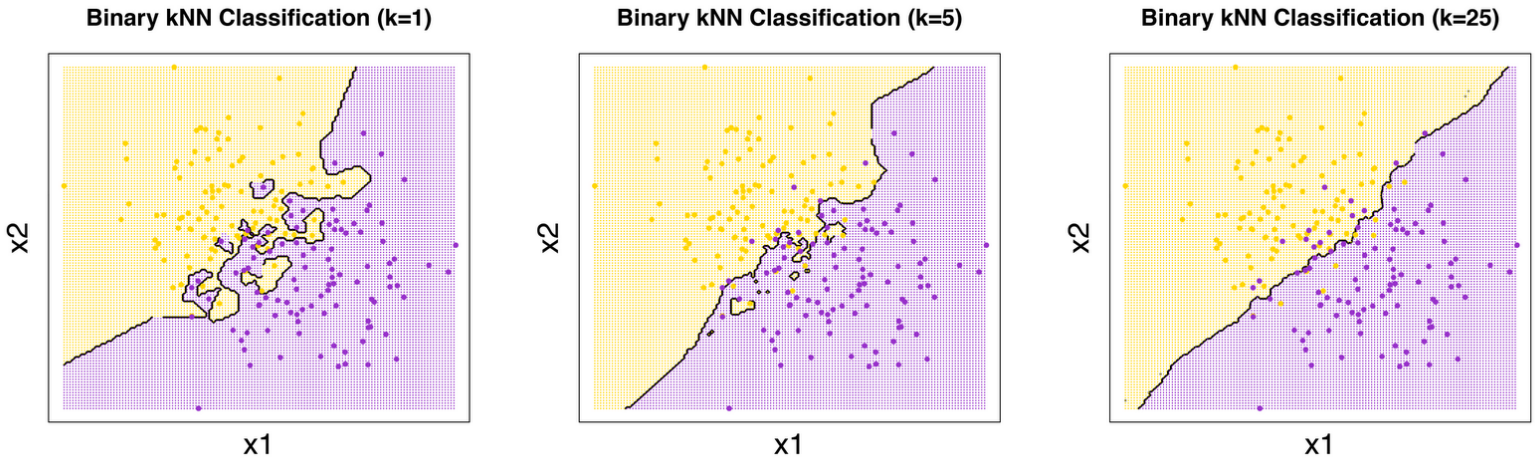
$$\begin{bmatrix} 457 & 0 & 25 & 5 & 2 & 0 & 2 & 0 & 9 & 0 \\ 0 & 385 & 26 & 2 & 7 & 0 & 3 & 0 & 77 & 0 \\ 11 & 0 & 458 & 9 & 4 & 1 & 1 & 2 & 14 & 0 \\ 3 & 0 & 46 & 415 & 3 & 1 & 0 & 3 & 27 & 2 \\ 5 & 0 & 38 & 18 & 411 & 0 & 1 & 3 & 16 & 8 \\ 17 & 0 & 22 & 149 & 12 & 218 & 1 & 1 & 79 & 1 \\ 19 & 0 & 41 & 55 & 16 & 13 & 336 & 0 & 20 & 0 \\ 3 & 0 & 42 & 20 & 39 & 0 & 0 & 358 & 19 & 19 \\ 10 & 0 & 40 & 57 & 7 & 6 & 1 & 4 & 374 & 1 \\ 3 & 0 & 14 & 11 & 70 & 0 & 0 & 9 & 31 & 362 \end{bmatrix}$$

Probabilidade Total de Erro= $\frac{1226}{5000} \approx 24.52\%$

Classificadores Baseados em Distâncias

Classificador dos *k*-Vizinhos Mais Próximos (*k*-NN)

- *k*-NN é um classificador não-paramétrico.
- Não existe fase de treino para este classificador.
- A classificação é baseada nos exemplos de treino. A classe atribuída a um dado objecto (ponto/vector não classificado) é a classe maioritária entre os *k*-vizinhos mais próximos do objecto.
- O valor óptimo para *k* é dependente do problema. Valores pequenos de *k* dão origem a zonas e fronteiras de decisão irregulares (efeito de sobre-aprendizagem). Valores muitos elevados podem resultar em regiões e fronteiras de decisão demasiado regulares.



Exemplo tirado da página de Burton DeWilde

(http://bdewilde.github.io/blog/blogger/2012/10/26/classification-of-hand-written-digits-3/)

Classificadores Baseados em Distâncias

Classificador dos k-Vizinhos Mais Próximos (k-NN)

Em Python usando k-NN implementado em scikit-learn

- Comandos:

  Importar classificador de vizinho mais próximo de `scikit-learn`

  ```
  >>> from sklearn.neighbors import KNeighborsClassifier
  ```

  Instanciar classificador k-NN com $k = 5$

  ```
  >>> kNN=KNeighborsClassifier(n_neighbors=5,weights='uniform')
  ```

  Treino:

  ```
  >>> kNN.fit(trainData,trainClasses) # dados de treino
  ```

  - `trainData`: dados de treino, matriz $N \times d$ (N nº de exemplos, d dimensão)
  - `trainClasses`: N índices das classes (números inteiros)

- Classificar:

  ```
  >>> resultados=kNN.predict(testData)
  ```

  - `testData`: dados de teste, matriz $M \times d$ (M nº de exemplos, d dimensão)
  - `resultados`: M estimativas dos índices das classes

Classificadores Baseados em Distâncias

Classificador dos *k*-Vizinhos Mais Próximos (*k*-NN)

Em Python usando *k*-NN implementado em scikit-learn

- Ver parametros da função `KNeighborsClassifier`:
  `n_neighbors, weights, algorithm, leaf_size, metric`, e outros.
- Ver métodos associado a `KNeighborsClassifier`:

```
kNN.fit()
kNN.predict()
kNN.predict_proba()
kNN.score()
kNN.kneighbors()
kNN.kneighbors_graph()
kNN.get_params()
kNN.set_params()
```

Classificadores Baseados em Distâncias

Classificador dos k-Vizinhos Mais Próximos (k-NN)

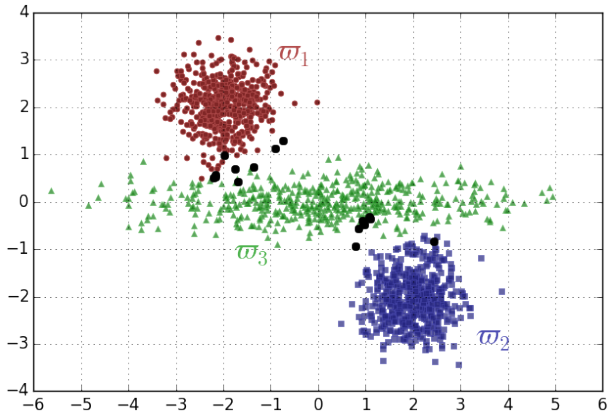
Dados Sintéticos (`LAB2distancias002.p`)

- $\mathcal{X}$, conjunto de pontos 2D dividido em três classes $\Omega = \{\varpi_1, \varpi_2, \varpi_3\}$.
  - Nº de pontos treino: 100 pts por classe
  - Nº de pontos teste: 500 pts por classe
- Probabilidades a priori: $p(\varpi_1) = p(\varpi_2) = p(\varpi_3)$
- Probabilidades condicionadas gaussianas: $p(\mathbf{x}|\varpi_i) = \mathcal{N}(\mu_i, \Sigma_i)$

Classificação ($k = 1$):

Matriz de Confusão:

$$\begin{bmatrix} 495 & 0 & 5 \\ 0 & 500 & 0 \\ 4 & 8 & 488 \end{bmatrix} \quad \text{Prob.Erro} = \frac{17}{1500}$$



Classificadores Baseados em Distâncias

Classificador dos k-Vizinhos Mais Próximos (k-NN)

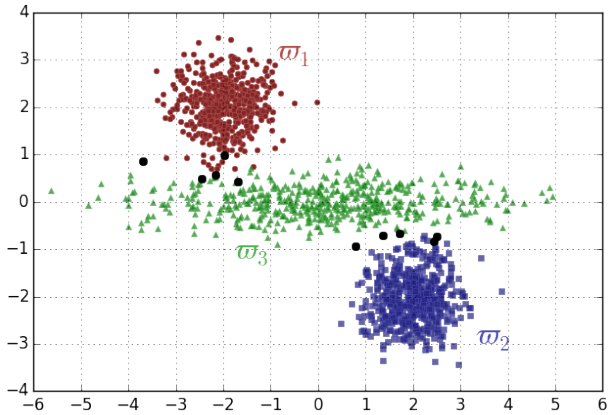
Dados Sintéticos (`LAB2distancias002.p`)

- $\mathcal{X}$, conjunto de pontos 2D dividido em três classes $\Omega = \{\varpi_1, \varpi_2, \varpi_3\}$.
  - Nº de pontos treino: 100 pts por classe
  - Nº de pontos teste: 500 pts por classe
- Probabilidades a priori: $p(\varpi_1) = p(\varpi_2) = p(\varpi_3)$
- Probabilidades condicionadas gaussianas: $p(\mathbf{x}|\varpi_i) = \mathcal{N}(\mu_i, \Sigma_i)$

Classificação ($k = 5$):

Matriz de Confusão:

$$\begin{bmatrix} 498 & 0 & 2 \\ 0 & 499 & 1 \\ 3 & 4 & 493 \end{bmatrix} \quad \text{Prob.Erro} = \frac{10}{1500}$$



Classificadores Baseados em Distâncias

Classificador dos k-Vizinhos Mais Próximos (k-NN)

Dígitos Manuscritos

- $\mathcal{X}$, conjunto de dígitos - dados em bruto ($d = 784$).
  - Nº de pontos treino: 1000 pts por classe
  - Nº de pontos teste: 500 pts por classe

Classificação ($k = 1$):

Matriz de Confusão:

$$\begin{bmatrix} 496 & 0 & 1 & 0 & 0 & 1 & 2 & 0 & 0 & 0 \\ 0 & 495 & 1 & 2 & 0 & 0 & 2 & 0 & 0 & 0 \\ 7 & 11 & 452 & 7 & 1 & 0 & 3 & 18 & 1 & 0 \\ 0 & 1 & 1 & 461 & 1 & 13 & 2 & 9 & 7 & 5 \\ 0 & 5 & 0 & 0 & 460 & 0 & 3 & 2 & 1 & 29 \\ 3 & 3 & 0 & 13 & 2 & 466 & 3 & 2 & 3 & 5 \\ 8 & 3 & 0 & 0 & 3 & 4 & 481 & 0 & 1 & 0 \\ 0 & 22 & 3 & 2 & 4 & 1 & 0 & 456 & 0 & 12 \\ 5 & 3 & 6 & 27 & 4 & 16 & 6 & 7 & 421 & 5 \\ 3 & 6 & 1 & 6 & 12 & 3 & 1 & 11 & 2 & 455 \end{bmatrix}$$

$$\text{Prob.Erro} = \frac{357}{5000} = 7.140\%$$

Classificadores Baseados em Distâncias

Classificador dos k-Vizinhos Mais Próximos (k-NN)

Dígitos Manuscritos

- $\mathcal{X}$, conjunto de dígitos - dados em bruto ($d = 784$).
  - Nº de pontos treino: 1000 pts por classe
  - Nº de pontos teste: 500 pts por classe

Classificação ($k = 5$):

Matriz de Confusão:

$$\begin{bmatrix} 494 & 0 & 0 & 0 & 0 & 1 & 5 & 0 & 0 & 0 \\ 0 & 495 & 2 & 2 & 0 & 0 & 1 & 0 & 0 & 0 \\ 9 & 16 & 439 & 6 & 0 & 0 & 6 & 21 & 3 & 0 \\ 1 & 3 & 3 & 465 & 1 & 7 & 2 & 8 & 7 & 3 \\ 0 & 4 & 1 & 0 & 462 & 0 & 5 & 1 & 0 & 27 \\ 3 & 4 & 0 & 12 & 6 & 461 & 5 & 1 & 3 & 5 \\ 9 & 5 & 0 & 0 & 4 & 1 & 481 & 0 & 0 & 0 \\ 0 & 31 & 1 & 1 & 5 & 1 & 0 & 445 & 0 & 16 \\ 8 & 4 & 7 & 22 & 4 & 16 & 5 & 7 & 418 & 9 \\ 4 & 7 & 2 & 8 & 11 & 2 & 1 & 12 & 2 & 451 \end{bmatrix}$$

$$\text{Prob.Erro} = \frac{389}{5000} = 7.780\%$$