

Relatório do Trabalho de Aprendizagem de Máquina

Maria Aparecida Amorim Sibaldo, Tiago Buarque Assunção de Carvalho

Centro de Informática – Universidade Federal de Pernambuco (UFPE)

Caixa Postal 7851 – 50.732-970 – Recife – PE – Brasil.

{maas2, tbac}@cin.ufpe.br

Recife, 11 de maio de 2012

Resumo. *Este trabalho é um estudo prático de vários modelos de aprendizagem de máquina. O mesmo compreende implementação e validação de vários modelos dessa área. Começando com a geração de padrões seguindo distribuições especificadas. Em seguida esses dados são agrupados em clusters através do algoritmo K-Means e avaliado suas taxas de erro por classe e global, e pelo índice Rand corrigido. É realizada classificação pela regra de decisão bayesiana através de métodos paramétricos (cujos parâmetros são estimados pela Máxima Verossimilhança e EM) e não paramétricos (k-NN e Janela de Parzen) e também é utilizada a combinação de classificadores. Estes classificadores são testados através da repetição de hold-out e comparados pela média dos erros e pelo intervalo de confiança.*

1. Introdução

Aprendizagem de máquina é uma área multidisciplinar vasta que envolve Computação, Matemática, Estatística, Psicologia e Neurociências. O objetivo da aprendizagem de máquina é desenvolver algoritmos que permitam ao computador aprender de forma autônoma a partir de um conjunto de dados. Os algoritmos de aprendizagem se dividem em dois grupos, a depender da forma como eles aprendem: aprendizado supervisionado e não-supervisionado. Em ambos os casos os algoritmos aprendem a partir de um conjunto de padrões. Cada padrão é um vetor de características quantitativas e/ou qualitativas.

No caso do aprendizado supervisionado cada elemento do conjunto de treinamento (utilizado para aprender) possui uma variável resposta, que é aquela que se pretende estimar após o aprendizado para padrões não vistos anteriormente. Depois do treinamento, um padrão é passado como entrada para o modelo que responde com a estimativa da variável resposta. Existem dois modelos de aprendizagem supervisionada: classificador, quando a variável resposta é qualitativa, e o regressor, quando a variável resposta é quantitativa. Um exemplo de classificador é um sistema de reconhecimento de faces, que recebe como entrada a imagem de uma face e dá como saída a identificação da pessoa que possui aquela face. Um exemplo de regressor é um sistema que prevê o preço de ações na bolsa de valores.

O aprendizado não supervisionado visa realizar agrupamentos. Um conjunto de valores (sem variável resposta) é passado para o modelo que tenta realizar o melhor agrupamento para aqueles dados. Agrupamento é muito útil em várias áreas tais como a Biologia, para gerar taxinomias.

Além de criação desses modelos é preciso definir como avaliá-los para saber quão bem eles resolvem o problema. Um classificador, por exemplo, pode ser avaliado em termos de taxa de erro de classificação, robustez a dados ruidosos, tempo gasto para realizar uma classificação, tempo gasto para aprender etc.

Depois de avaliados os modelos eles devem ser comparados para verificar qual deles é mais adequado ao problema. Para se escolher qual o melhor modelo, é definido o critério a ser avaliado, e partindo do conjunto do resultado de várias avaliações realiza-se um teste estatístico. Por exemplo, avaliando qual entre dois classificadores tem a menor taxa de erro: para se fazer um teste t-student emparelhado, calcula-se diversas taxas de erros para ambos os classificadores sob as mesmas condições [Triola 2005]. Outra forma de ser avaliar é através do intervalo de confiança, como será visto mais adiante.

Este trabalho é um estudo prático de vários algoritmos de aprendizagem de máquina. Na seção 2 tem-se os experimentos consistem em um agrupamento em dois grupos de três conjuntos de dados gerados com distribuições normais bivariadas de um conjunto de parâmetros fixados. Aplicou-se a esses padrões o algoritmo K-Means com 2 clusters 100 vezes e foi selecionado o melhor resultado dessas 100 aplicações segundo o critério de adequação entre os clusters e seus representantes.

O critério de adequação utilizado foi a soma das distâncias entre todos os padrões de cada classe ao seu centróide. O melhor resultado foi o que teve menor critério de aplicação. Para essa melhor execução foram calculadas as taxas de erro de classificação global e por classe. Além disso, foi calculado o índice de *Rand* corrigido, que retorna um valor entre 0 e 1, onde o 0 significa que os dois clusters não concordam em nenhum par de padrões, e 1 indica que os clusters são exatamente iguais.

Na seção 2 é exposta a regra de decisão bayesiana utilizada pelos classificadores, depois é revisada a distribuição normal bivariada. É explanado o método de máxima verossimilhança para estimação de parâmetros a partir dos dados de uma distribuição normal. Expõe-se algoritmo EM é um método iterativo para estimação dos parâmetros de cada componente e dos parâmetros da mistura de uma mistura de gaussianas. Depois, o método de kernel ou janela de parzen que é um método não paramétrico para a estimativa de densidade em um ponto específico. O método de k-vizinhos estima a posteriori diretamente. E a combinação de classificadores pelo método da soma.

Na seção 3 são analisados os desempenhos dos classificadores em relação ao erro de classificação. Os testes dos classificadores foram realizados usando 100 repetições de hold-out estratificado, gerando o total de 100 medidas de erro de classificação para cada classificador sob as mesmas condições. Foi escolhido calcular várias taxas de erro aleatória através da repetição do hold-out de modo que se possa calcular também o desvio padrão do erro de cada classificador. A partir desse desvio padrão, pode-se calcular o intervalo de confiança que é o intervalo onde se espera que o verdadeiro valor do erro esteja contido. A primeira fase dos testes é a escolha dos parâmetros para os algoritmos de janela de Parzen e k-vizinhos. Depois são analisadas as médias de erros

de cada classificador e finalmente são comparados os classificadores através do seu intervalo de confiança. E, por fim, na seção 4 tem-se as conclusões encontradas após a avaliação do trabalho.

2. Dados e Agrupamento

2.1 Base de Dados

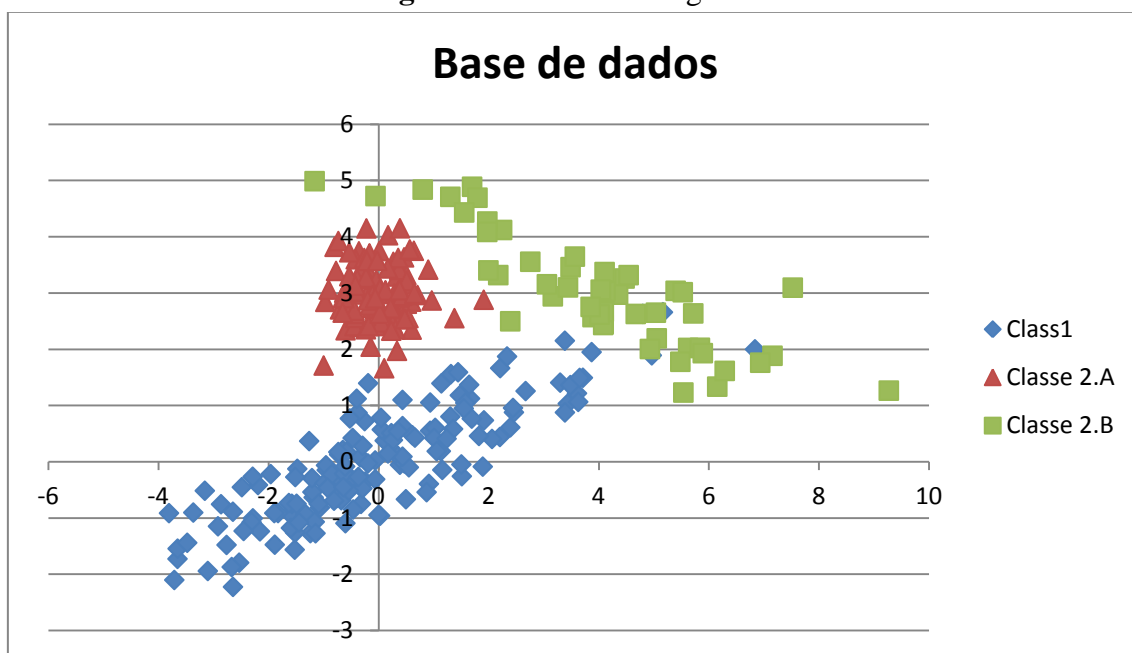
Conforme solicitado, para a criação de nossa amostra foi considerado um conjunto de 300 padrões formando 2 classes, cada classe com 150; As segunda classe é formado por uma mistura de gaussianas, 100 padrões provenientes de um componente 50 padrões provenientes também de uma distribuição normal com parâmetros distintos do primeiro componente da mistura. A base de dados criada está apresentada no gráfico da Figura 1. Os padrões de cada classe foram gerados a partir de distribuições normais bi-variadas segundo os seguintes parâmetros:

Classe 1: $\mu_1 = 0, \mu_2 = 0, \sigma_1 = 2, \sigma_2 = 1, \sigma_{12} = 1.7, \rho_{12} = 0.85$

Classe 2 (componente A): $\mu_1 = 0, \mu_2 = 3, \sigma_1 = 0.5, \sigma_2 = 0.5, \sigma_{12} = 0.0, \rho_{12} = 0.0$

Classe 2 (componente B): $\mu_1 = 4, \mu_2 = 3, \sigma_1 = 2, \sigma_2 = 1, \sigma_{12} = -1.7, \rho_{12} = -0.85$

Figura1. Base de dados gerada



2.2 K-Médias

O algoritmo K-Médias com 2 clusters foi implementado em Java e seguiu o seguinte conjunto de instruções:

- 1) *Fornecer valores para os centróides*

Inicialmente, escolhemos aleatoriamente dois padrões diferentes para serem os centróides dos dois clusters que devem ser formados.

- 2) *Gerar uma matriz de distâncias entre cada padrão e os centróides*

Para cada padrão da amostra foi calculado a distância entre esse padrão e ambos os centróides. Tal distância foi calculada através do quadrado da distância euclidiana.

$$D^2(\mathbf{x}, \mathbf{y}) = \left(\sum_{a=1}^n (x_a - y_a)^2 \right)$$

- 3) *Colocar cada padrão nas classes de acordo com a sua distância do centróide da classe*

Para se determinar com qual centróide cada padrão formaria um cluster, comparou-se o quadrado da distância euclidiana entre um padrão e cada um dos dois centróides. Caso a distância de um determinado padrão para o centróide 1 fosse menor ou igual ao quadrado da distância euclidiana deste padrão para o centróide 2, então o padrão faria parte do centróide 1. Caso contrário, faria parte do centróide 2.

- 4) *Calcular os novos centróides para cada classe*

Recalcular os centróides. Aqui, cada centróide é calculado realizando-se a média dos padrões inseridos em seu cluster. $\mathbf{x} = [x_1, x_2]^t$ é um padrão; i é o índice do cluster, $i = 1$ ou $i = 2$; e n é o número de padrões contidos naquele cluster.

$$c_i = \frac{1}{n_i} \sum_{j=1}^n x_j, x_j \text{ pertence à } c_i$$

- 5) *Repetir até a convergência*

Os passos do 2, 3 e 4 se repetem até que haja a convergência dos valores dos dois centróides, ou seja, até que eles sejam os mesmos por pelo menos duas iterações consecutivas.

Após a construção do algoritmo K-Means, ele foi executado 100 vezes e foi escolhido os resultados de uma destas 100 execuções: a que obteve um menor Critério de Adequação. O Critério de Adequação (a) para cada execução do K-Means foi encontrado fazendo-se a soma de todos os quadrados das distâncias euclidianas entre os padrões no cluster 1 e seu centróide, acrescido da soma de todas as distâncias euclidianas ao quadrado entre os padrões do cluster 2 e seu centróide, como apresentado a seguir.

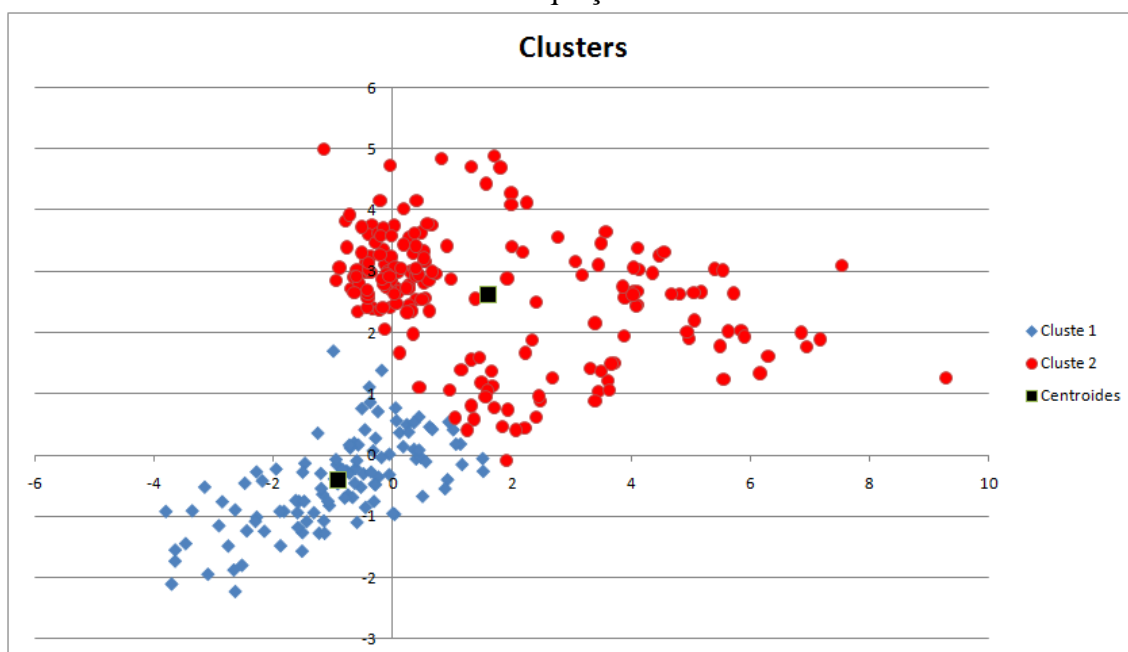
$$a = \sum_{k=1}^{n_1} D^2(c_1, x_k) + \sum_{k=1}^{n_2} D^2(c_2, x_k),$$

em que n_1 é a quantidade de padrões no cluster 1 e n_2 é a quantidade de padrões no cluster 2.

O menor valor do Critério de Adequação nos informa qual a melhor clusterização feita entre essas 100 execuções.

O melhor resultado encontrado foi o que teve como Critério de Adequação o valor: 1288.85374738435, como centróide 1: (-0.9237171247886998, -0.41580094531581757) e como centróide 2: (1.60070159180712, 2.6251529361378605). A apresentação gráfica destes clusters estão apresentados na Figura 2:

Figura 2. Agrupamentos formados pelo K-Means que obteve o menor critério de adequação.



Utilizando-se destas informações, o K-Means foi executado com estes centróides para encontrarmos as taxas de erro por classe e global, assim, encontramos os seguintes erros em cada classe:

Erros no cluster 1: 40 padrões, de 189 padrões, classificados como do cluster 1, mas eram da classe 2;

Erros no cluster 2: 1 padrão, de 111 padrões, classificados como do cluster 2, mas eram da classe 1.

Portanto, encontramos as seguintes taxas de erro:

- **Taxas de erro por classe**

Taxa de erro encontrado no **cluster 1**: **21,16%**(calculado por $(40/189) * 100$);

Taxa de erro encontrado no **cluster 2**: **0,9%** (calculado por $(1/111) * 100$).

- **Taxa de erro global**

13,67%, que foi calculado através da equação: $100 * (\text{erros no cluster 1} + \text{erros no cluster 2}) / \text{soma dos números de padrões no cluster 1 e no cluster 2} = 100 * (40 + 1) / 300$.

Índice de Rand Corrigido

O índice de Rand corrigido tem um valor entre 0 e 1, com 0 indicando que os dois clusters não concordam em nenhum par de padrões e 1 indicando que os dados dos clusters são exatamente iguais.

Para calcular o índice de Rand corrigido temos que $X = \{X1, X2\}$ é a partição com as classes reais e $Y = \{Y1, Y2\}$ é a partição calculada pelo k-Means. O índice de Rand corrigido é dado por ARI:

$$ARI = \frac{\sum_{ij} \binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}] / \binom{n}{2}}{\frac{1}{2} [\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}] / \binom{n}{2}}$$

Onde cada combinação é dada por

$$C_s^n = \binom{n}{s} = \frac{n!}{s! \cdot (n-s)!}$$

comb(n, s) =

E n_{ij} , a_i , b_j e n são apresentados abaixo.

Dado n_{11} , n_{12} , n_{21} e n_{22} , temos que n_{ij} é o número de padrões que estão ao mesmo tempo em X_i e Y_j . O valores encontrados para a execução acima citada são:

$n_{11}= 149$; $n_{12}= 1$; $n_{21}= 40$; $n_{22}= 110$

Abaixo é apresentado como a_i, b_j e n são formados, e seus valores já calculados.

$a_1 = n_{11} + n_{12}$; $a_1 = 150$

$a_2 = n_{21} + n_{22}$; $a_2 = 150$

$b_1 = n_{11} + n_{21}$; $b_1 = 189$

$b_2 = n_{12} + n_{22}$; $b_2 = 111$

$n = n_{11} + n_{12} + n_{21} + n_{22}$; $n = 300$

A fórmula de cálculo de uma combinação, e seus respectivos valores, são os seguintes:

$\text{index} = \text{comb}(n_{11}, 2) + \text{comb}(n_{12}, 2) + \text{comb}(n_{21}, 2) + \text{comb}(n_{22}, 2)$; $\text{index} = 17802$

$\text{somaAi} = \text{comb}(a_1, 2) + \text{comb}(a_2, 2)$; $\text{somaAi} = 22350$

$\text{somaBj} = \text{comb}(b_1, 2) + \text{comb}(b_2, 2)$; $\text{somaBj} = 23871$

$\text{expectedIndex} = (\text{somaAi} * \text{somaBj}) / \text{comb}(n, 2)$; $\text{expectedIndex} = 11816,5415$

$\text{maxIndex} = 0.5 * (\text{somaAi} + \text{somaBj})$; $\text{maxIndex} = 23110,5$

Por fim, temos que

$ARI = (\text{index} - \text{expectedIndex}) / \text{maxIndex} - \text{expectedIndex}$

Resultando em

$ARI = 5985,4585 / 11293,9585$

$ARI = 0,52997$

Observando que o valor de ARI está em um valor intermediário entre 0 e 1, concluímos que a formação de clusters realizado aqui atingiu um resultado intermediário entre o melhor e o pior desempenho.

3. Classificadores

3.1 Regra de decisão

A regra de decisão bayesiana para todos os algoritmos avaliados é: atribua a x à classe ω_j com maior probabilidade posteriori:

$$j = \operatorname{argmax}_i [P(\omega_i|x, \theta_i)]$$

com

$$P(\omega_i|x, \theta_i) = \frac{P(x|\omega_i, \theta_i)P(\omega_i)}{\sum_{j=1}^c P(x|\omega_j, \theta_j)P(\omega_j)}$$

$i = 1$ ou $i = 2$, e $c = 2$ é o número de classes. $P(\omega_i|x, \theta_i)$ é a probabilidade a posteriori de uma classe i dado um padrão (vetor de atributos) desconhecido x e um vetor de parâmetros θ_i . $P(\omega_i)$ é a probabilidade a priori da classe ω_i . E $P(x|\omega_i, \theta_i)$ é a densidade de x dado ω_i, θ_i . A função de densidade utilizada é densidade normal bivariada, que será vista a seguir.

$P(\omega_i)$ é estimado da seguinte forma:

$$P(\omega_i) = \frac{n_i}{n}$$

em que n_i é o número de padrões da classe ω_i no conjunto de treinamento e n é o número total de padrões no conjunto de treinamento.

No primeiro classificador a densidade da classe 1 é estimada pelo método da máxima verossimilhança e da classe 2 pelo algoritmo EM. No segundo classificador a densidade é estimada pelo método não paramétrico de kernel ou janela de Parzen. No terceiro classificador a posteriori é estimada diretamente através do método não k-vizinhos (k-NN). O quarto e último classificador utiliza uma regra de decisão derivada desta vista acima, ele realiza a combinação das posteriores dos três primeiros classificadores pelo método da soma.

3.2 Distribuição normal bivariada

O primeiro classificador utiliza os métodos paramétricos de máxima verossimilhança e de EM. Estes métodos assumem que os dados seguem determinada distribuição. Aqui é assumido que essa distribuição é Normal (Gaussiana) bivariada, que é um caso mais restrito de uma distribuição normal multivariada.

Uma distribuição normal multivariada com os parâmetros vetor de médias μ e matriz de variâncias e covariâncias Σ é uma função $p(x)$:

$$p(x) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp \left[-\frac{1}{2} (x - \mu)^t \Sigma^{-1} (x - \mu) \right]$$

se a distribuição tem m variáveis, x e μ são um vetores $m \times 1$ e Σ é uma matriz $m \times m$.

No caso bivarado \mathbf{x} é um vetor 2×1 , o vetor de médias $\boldsymbol{\mu}$ também é 2×1 e a matriz de covariância $\boldsymbol{\Sigma}$ é uma matriz 2×2 . Alguns simplificações são feitas nesse caso:

- o coeficiente de correlação é

$$\rho = \frac{\sigma_{12}}{\sigma_1 \sigma_2}$$

- a matriz de covariância é

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & \rho \sigma_1 \sigma_2 \\ \rho \sigma_1 \sigma_2 & \sigma_2^2 \end{bmatrix}$$

- o determinante de $\boldsymbol{\Sigma}$ é

$$|\boldsymbol{\Sigma}| = \sigma_1^2 \sigma_2^2 (1 - \rho^2)$$

- e a inversa de $\boldsymbol{\Sigma}$ é

$$\begin{aligned} \boldsymbol{\Sigma}^{-1} &= \frac{1}{\sigma_1^2 \sigma_2^2 (1 - \rho^2)} \begin{bmatrix} \sigma_2^2 & -\rho \sigma_1 \sigma_2 \\ -\rho \sigma_1 \sigma_2 & \sigma_1^2 \end{bmatrix} \\ &= \frac{1}{1 - \rho^2} \begin{bmatrix} 1/\sigma_1^2 & -\rho/(\sigma_1 \sigma_2) \\ -\rho/(\sigma_1 \sigma_2) & 1/\sigma_2^2 \end{bmatrix}. \end{aligned}$$

Desta forma uma distribuição normal bivariada com os parâmetros vetor de médias $\boldsymbol{\mu} = [\mu_1, \mu_2]t$ e matriz de variâncias e covariâncias $\boldsymbol{\Sigma}$ é uma função em $\mathbf{x} = [x_1, x_2]t$, $p(\mathbf{x}) = p_{x_1 x_2}(x_1, x_2)$:

$$\begin{aligned} p_{x_1 x_2}(x_1, x_2) &= \frac{1}{2\pi \sigma_1 \sigma_2 \sqrt{1 - \rho^2}} \times \\ &\exp \left[-\frac{1}{2(1 - \rho^2)} \left[\left(\frac{x_1 - \mu_1}{\sigma_1} \right)^2 - 2\rho \left(\frac{x_1 - \mu_1}{\sigma_1} \right) \left(\frac{x_2 - \mu_2}{\sigma_2} \right) + \left(\frac{x_2 - \mu_2}{\sigma_2} \right)^2 \right] \right] \end{aligned}$$

$p_{x_1 x_2}(x_1, x_2)$ é a densidade $P(\mathbf{x}|\omega_i, \theta_i)$ na regra de decisão quando os parâmetros $\theta_i = [\mu_1, \mu_2, \sigma_1, \sigma_2, \rho]$ são estimados a partir do padrões do conjunto de treinamento que pertencem à classe ω_i . Esta estimação pode ser realizada tanto pelo método da máxima verossimilhança, quanto todos os padrões da classe seguem uma mesma distribuição normal, como pelo algoritmo EM, quando os padrões de uma classe tem origem de duas ou mais distribuições normais distintas.

3.3 Máxima verossimilhança

Este método é utilizado para estimar os parâmetros $\boldsymbol{\theta}_1 = [\mu_1, \mu_2, \sigma_1, \sigma_2, \rho]$ da classe ω_1 a partir dos padrões do conjunto de treinamento que pertencem a esta classe. A partir destes parâmetros pode-se estimar $P(\mathbf{x}|\omega_1, \boldsymbol{\theta}_1)$ através de $p_{x_1 x_2}(x_1, x_2)$, e por conseguinte estimar $P(\omega_1|\mathbf{x}, \boldsymbol{\theta}_1)$ para ser utilizado na regra de decisão.

A estimativa de máxima verossimilhança encontra o vetor de parâmetros que melhor se ajuste aos dados. Esses estimadores são $\hat{\mu}$ e $\hat{\Sigma}$

$$\hat{\mu} = \frac{1}{n} \sum_{k=1}^n \mathbf{x}_k$$

e

$$\hat{\Sigma} = \frac{1}{n} \sum_{k=1}^n (\mathbf{x}_k - \hat{\mu})(\mathbf{x}_k - \hat{\mu})^t$$

Abaixo os parâmetros estimados através do método de máxima verossimilhança a partir de todos os 150 exemplos da classe 1. Observa-se que os parâmetros estimados são muito próximos daqueles utilizados para gerar os dados.

$$\mu_1 = 3.552713678800501\text{E-}17$$

$$\mu_2 = 9.62193288008469\text{E-}18$$

$$\rho = 0.85000000000000001$$

$$\sigma_1 = 1.9933221850301404$$

$$\sigma_{12} = 1.6886666666666668$$

$$\sigma_2 = 0.9966610925150702$$

3.4 Expectation-Maximization (EM)

Quando se diz que os dados são provenientes de misturas de gaussianas isto é equivalente a dizer que alguns dados de determinada classe seguem determinada distribuição normal e outros dados da mesma classe seguem outra distribuição normal com parâmetros distintos daquela, cada uma dessas distribuições são chamados componentes da mistura e cada componente tem seu vetor parâmetros próprio. Cada componente é como uma subclasse de determinada classe, contudo os rótulos dessas subclasses não são conhecidos.

O algoritmo EM é um método iterativo para estimação dos parâmetros θ_i de cada componente ω_i e dos parâmetros da mistura $P(\omega_i)$, que são a priori de cada componente. Para tanto é preciso especificar para o EM o número de componentes da mistura. O algoritmo é inicializado com uma heurística para os valores iniciais dos parâmetros então esses parâmetros são atualizados iterativamente minimizando determinada métrica de adequação.

No caso da classe 2 os dados são originários de um mistura de duas gaussianas, cada componente é chamado A e B respectivamente. A heurística para $P(\omega_i), i \in \{A, B\}$, é $P(\omega_i) = 0,5$. A heurística para o componente A são os parâmetros estimados pela máxima verossimilhança para a primeira metade do conjunto de treino. A heurística para o componente B são os parâmetros estimados pela máxima verossimilhança para os padrões outra metade do conjunto de treino.

A métrica de adequação escolhida aqui é a log-verossimilhança l:

$$l = \sum_{k=1}^n \ln \left\{ \sum_{i=1}^c P(\omega_i) N(\mathbf{x}_k | \omega_i, \theta_i) \right\}$$

em que $N(\mathbf{x}_k | \omega_i, \theta_i) = p_{x_1 x_2}(x_1, x_2)$. Quanto maior o valor da log-verossimilhança mais os parâmetros estão adequados aos dados. Portanto o algoritmo iterativo de EM executa enquanto o valor da log-verossimilhança cresce em relação à iteração anterior. Quando o valor da log-verossimilhança converge, o algoritmo chega ao fim.

Outro ponto crucial do EM é que sua etapa iterativa é dividida em duas partes: etapa E e etapa M. A etapa E estima a posteriori $P(\omega_i | \mathbf{x}_k, \theta)$ de cada componente $i \in \{A, B\}$ para cada padrão k :

$$P(\omega_i | \mathbf{x}_k, \theta) = \frac{P(\omega_i) N(\mathbf{x}_k | \omega_i, \mu_i, \Sigma_i)}{\sum_{j=1}^c P(\omega_j) N(\mathbf{x}_k | \omega_j, \mu_j, \Sigma_j)}$$

A etapa M estima novamente os parâmetros através do método da máxima verossimilhança. Os estimadores utilizados na etapa M, para cada componente $i \in \{A, B\}$, são o estimador da nova média

$$\mu_i^{new} = \frac{\sum_{k=1}^n P(\omega_i | \mathbf{x}_k, \mu_i, \Sigma_i) \mathbf{x}_k}{\sum_{k=1}^n P(\omega_i | \mathbf{x}_k, \mu_i, \Sigma_i)}$$

o estimador da nova matriz de covariância

$$\Sigma_i^{new} = \frac{\sum_{k=1}^n P(\omega_i | \mathbf{x}_k, \mu_i, \Sigma_i) (\mathbf{x}_k - \mu_i^{new})(\mathbf{x}_k - \mu_i^{new})^T}{\sum_{k=1}^n P(\omega_i | \mathbf{x}_k, \mu_i, \Sigma_i)}$$

e os parâmetros da mistura

$$P(\omega_i)^{new} = \frac{1}{n} \sum_{k=1}^n P(\omega_i | \mathbf{x}_k, \mu_i, \Sigma_i)$$

O algoritmo do EM é

1. Inicialize os parâmetros dos componentes (como descrito anteriormente);
2. l_{new} = valor inicial a log-verossimilhança;
3. $l_{old} = l_{new} - 1$;
4. while ($l_{new} > l_{old}$)
 - 4.1. etapa E: estima a posteriori $P(\omega_i | \mathbf{x}_k, \theta)$ de cada componente $i \in \{A, B\}$ para cada padrão k ;
 - 4.2. etapa M:
 - 4.2.1. estima a nova média μ_i^{new} para cada componente $i \in \{A, B\}$;

- 4.2.2. estima a nova matriz de covariância Σ_i^{new} para cada componente $i \in \{A, B\}$;
- 4.2.3. estima os novo parâmetros da mistura $P(\omega_i)^{new}$ para cada componente $i \in \{A, B\}$;
- 4.3. $l_{old} = l_{new}$;
- 4.4. l_{new} = log-verossimilhança calcula a partir dos novo parâmetros;

Após a convergência, quando a log-verossimilhança não mais aumenta, os parâmetros estão estimados e pode-se calcular a densidade da mistura a partir da densidade de cada componente $p(\mathbf{x}|\omega_j, \theta_j)$ e dos parâmetros da mistura $P(\omega_j)$:

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{j=1}^c p(\mathbf{x}|\omega_j, \theta_j) P(\omega_j)$$

Abaixo os parâmetros estimados para as misturas a partir de todos os 150 exemplos da classe 2 através do algoritmo EM. Observa-se que os parâmetros estimados são muito próximos daqueles utilizados para gerar os dados.

	Parâmetros iniciais	Parâmetros finais
A	A	A
$P(\omega_A)$	0.5	0.6502615480254033
μ_{1A}	-0.01711877734686451	-0.027324525997496473
μ_{2A}	3.047001415166429	2.993095446431221
ρ_A	-0.011523442398073912	0.0026010724341935347
σ_{1A}	0.4745963920317533	0.4556335310871572
σ_{12A}	-0.0027220131240903484	5.880863012334208E-4
σ_{2A}	0.49771822911874475	0.4962184860150349
B	B	B
$P(\omega_B)$	0.5	0.34973845197459685
μ_{1B}	2.6837854440135316	3.8631766518073993
μ_{2B}	2.952998584833571	3.012837495181641
ρ_B	-0.4614201557423857	-0.8267915478345379
σ_{1B}	2.486314723166334	2.0361447000003494
σ_{12B}	-0.9809983329096588	-1.6388909447011584
σ_{2B}	0.8550974398750941	0.9735211457149779

Abaixo os valores da log-verossimilhança para o experimento acima do início até a convergência:

log-verossimilhança na iteração 1 = -399.11697865204
log-verossimilhança na iteração 2 = -387.0186569058696
log-verossimilhança na iteração 3 = -381.175451626438
log-verossimilhança na iteração 4 = -379.0809140855965
log-verossimilhança na iteração 5 = -378.32891843537783
log-verossimilhança na iteração 6 = -378.0561834154566
log-verossimilhança na iteração 7 = -377.9697492500277

log-verossimilhança na iteração 8 = -377.946158456243
 log-verossimilhança na iteração 9 = -377.94035279417255
 log-verossimilhança na iteração 10 = -377.9390037301561
 log-verossimilhança na iteração 11 = -377.93869912023007
 log-verossimilhança na iteração 12 = -377.938631284792
 log-verossimilhança na iteração 13 = -377.9386162770212
 log-verossimilhança na iteração 14 = -377.93861296706086
 log-verossimilhança na iteração 15 = -377.93861223812615
 log-verossimilhança na iteração 16 = -377.93861207770874
 log-verossimilhança na iteração 17 = -377.9386120424172
 log-verossimilhança na iteração 18 = -377.9386120346544
 log-verossimilhança na iteração 19 = -377.93861203294676
 log-verossimilhança na iteração 20 = -377.93861203257114
 log-verossimilhança na iteração 21 = -377.9386120324887
 log-verossimilhança na iteração 22 = -377.9386120324704
 log-verossimilhança na iteração 23 = -377.9386120324662
 log-verossimilhança na iteração 24 = -377.93861203246564
 log-verossimilhança na iteração 25 = -377.93861203246524

3.5 Janela de Parzen

O método de kernel ou janela de parzen é um método não paramétrico para a estimativa de densidade em um ponto específico. É semelhante ao método de histograma, porém o histograma oferece uma estimativa para todos os valores em um intervalo enquanto a janela de parzen combina os valores da amostra de treino para estimar a densidade em um ponto específico. Para estimar a densidade em um ponto \mathbf{x} mede a distância de cada variável x_j para cada X_{ij} dos n padrões do conjunto de treinamento, essa distância é então dividida por h (parâmetro de suavização) e serve de entrada para a função de kernel $K(\cdot)$. Quanto menor a diferença $x_j - X_{ij}$ maior será a resposta do kernel. O somatório das respostas dos kernels dividido por nh^p , é a densidade no ponto \mathbf{x} – n é o número de padrões da classe ω_a no conjunto de treinamento e p é o número de dimensões no problema, para a base em questão $p = 2$. Quanto mais pontos estiverem próximos a \mathbf{x} , maior é a densidade naquele ponto pois as respostas do kernel será maior para os vizinhos de \mathbf{x} . A densidade de \mathbf{x} na classe ω_a utilizando-se o método de kernel é $P(\mathbf{x}|\omega_a)$:

$$P(\mathbf{x}|\omega_a) = \frac{1}{nh^p} \sum_{i=1}^n \prod_{j=1}^p K\left(\frac{x_j - X_{ij}}{h}\right)$$

Assumindo-se que as variáveis são independentes a combinação da resposta do kernel é realizada através do produtório das respostas do kernel para cada variável, desta forma pode-se utilizar um kernel univariado. Este trabalho utiliza o kernel gaussiano univariado:

$$K(x) = \frac{1}{\sqrt{2\pi}} \exp[-0,5x^2]$$

O parâmetro de suavização h ajusta se a estimativa é mais suave ou mais detalhada. Quando h é grande a estimativa fica mais suave, mas pode se afastar muito da densidade real. Por outro lado quando h é pequeno a estimativa é mais detalhada, porém mais sensível a ruídos e pode não aproximar bem a densidade real se não possuir amostras suficientes no conjunto de treinamento.

3.6 K-vizinhos (k-NN)

O método de k-vizinhos estima a posteriori, $p(\omega_i | \mathbf{x})$, diretamente. Ele se baseia em encontrar uma quantidade de vizinhos mais próximos, ou seja, com a menor distância para o padrão de teste \mathbf{x} . Entre os k vizinhos mais próximos (com menor distância) de \mathbf{x} , é realizada uma votação para saber qual classe é a maioria entre os k padrões da amostra de treinamento, cujas classes dos elementos são conhecidas. Atribui-se a \mathbf{x} a classe de maior frequência entre os k vizinhos.

A probabilidade a posteriori no método de k-vizinhos é calculada como segue:

$$p(\omega_i | \mathbf{x}) = \frac{k_i}{k}$$

onde k_i é o número de padrões da classe ω_i e k é o número de vizinhos, k é um parâmetro ajustável do método.

O algoritmo de k-vizinhos é fortemente influenciado pela escolha do valor k e pela função de distância utilizada. As funções de distâncias podem ser entre vetores de características puramente quantitativos, puramente qualitativos ou mistos. Para vetores de atributos puramente quantitativos a distância mais comumente empregada é a distância Euclidiana, mas existem várias outras como: city-block (ou Manhattan), Chebychev, Camberra etc. Estas distâncias estão descritas abaixo

Distância Euclidiana:

$$D(\mathbf{x}, \mathbf{y}) = \left(\sum_{a=1}^n (x_a - y_a)^2 \right)^{1/2}$$

onde \mathbf{x} e \mathbf{y} são vetores de atributos quantitativos de mesmo comprimento, e x_a e y_a são posições equivalentes dos dois vetores. A distância euclidiana é distância efetivamente utilizada nos experimentos a seguir. Abaixo outras funções de distâncias.

City-block:

$$D(\mathbf{x}, \mathbf{y}) = \sum_{a=1}^n |x_a - y_a|$$

Chebychev:

$$D(\mathbf{x}, \mathbf{y}) = \max_{a=1, \dots, n} |x_a - y_a|$$

Camberra:

$$D(\mathbf{x}, \mathbf{y}) = \sum_{a=1}^n \frac{|x_a - y_a|}{|x_a + y_a|}$$

3.7 Combinação de classificadores pelo método da soma

A combinação de classificadores é uma técnica utilizada para melhorar os desempenhos de classificadores. Tal combinação pode se dar de várias formas, mas a ideia entre elas é a mesma: usando a “opinião” de vários classificadores, obtém-se uma estimativa mais confiável. Contudo a combinação só gera resultados se os diversos classificadores combinados errarem em regiões distintas do espaço dos dados, pois, se os classificadores combinados responderem sempre igual para o mesmo conjunto de padrões, a combinação não resultará em qualquer melhoria.

A regra de decisão para todos eles é: atribua a \mathbf{x} a classe ω_i se $g_i(\mathbf{x}) > g_j(\mathbf{x})$, para todo $i \neq j$, onde $g_i(\mathbf{x})$, $i = 1, \dots, c$, a função discriminante e c é o número de classes, depende do método de combinação. O método de combinação descrito adiante baseia-se na combinação das probabilidades a posteriori estimadas pelos classificadores.

A função discriminante para combinar L classificadores pela regra da soma é dada por:

$$g_i(\mathbf{x}) = (1 - L)p(\omega_i) + \sum_{j=1}^L p_j(\omega_i|\mathbf{x})$$

onde L é o número de classificadores que está sendo combinado, $p(\omega_i)$ é a probabilidade a priori da classe ω_i e $p_j(\omega_i|\mathbf{x})$ é a probabilidade a posteriori estimada pelo j -ésimo classificador.

4. Testes dos classificadores

Nessa seção são analisados os desempenhos dos classificadores em relação ao erro de classificação. O erro de classificação é definido como: o número de erros no conjunto de teste dividido pelo tamanho do conjunto de testes.

$$\text{erro de classificação} = \frac{n_e}{n}$$

onde n_e é o número de instâncias classificadas erradas no conjunto de testes e n é o número total de instâncias no conjunto de testes.

Os testes dos classificadores foram realizados usando 100 repetições de hold-out estratificado, gerando o total de 100 medidas de erro de classificação para cada classificador sob as mesmas condições. Para cada taxa de erro medida os classificadores foram treinados com o mesmo conjunto de treinamento e avaliados no mesmo conjunto de teste.

O experimento do tipo hold-out divide aleatoriamente a base de dados em dois conjuntos: treino e teste. Estes dois conjuntos são partições, isto é, não há intercessão entre os

conjuntos de treino e teste. Foi definido para os experimentos que seguem que conjunto de teste contém 1/3 dos padrões e o conjunto de treino contém o restante dos padrões da base que não foram selecionados para o conjunto de testes, no caso 2/3 dos padrões. Pelo fato de os dados serem selecionados aleatoriamente cada repetição de hold-out gera uma nova taxa de erro para cada classificador testado. O hold-out estratificado garante que cada partição tenha a mesma proporção de cada classe, no caso do problema em questão 50% de cada classe no conjunto de treino e 50% de cada classe no conjunto de teste. Cada vez que o hold-out é repetido o mesmo conjunto de treino e teste é utilizado para todos os classificadores, fazendo com que as taxas de erros sejam emparelhadas.

A taxa de erro em um único conjunto de teste gera uma estimativa pontual. Foi escolhido calcular várias taxas de erro aleatória através da repetição do hold-out de modo que se possa calcular também o desvio padrão do erro de cada classificador. A partir desse desvio padrão, pode-se calcular o intervalo de confiança que é o intervalo onde se espera que o verdadeiro valor do erro esteja contido. Para se construir esse valor assume-se que o erro segue uma distribuição t-Student e precisa-se definir um nível de confiança. Nestes experimentos foi escolhido o nível de confiança de 95%, dessa forma o intervalo de confiança é calculado da seguinte forma:

$$[\bar{x} - 1,984 \times s; \bar{x} + 1,984 \times s]$$

onde \bar{x} é a média do erro e s é o desvio padrão do erro.

Se não há interseção entre o intervalo de confiança do erro de dois classificadores tem-se uma certeza, proporcional ao nível de confiança utilizado para se construir o intervalo, de que o erro real desses classificadores sejam distintos, e o classificador com menor taxa de erro é aquele com menor média de erro. Quanto maior a interseção menores são as evidências para se comparar a média dos erros dos classificadores.

A primeira fase dos testes é a escolha dos parâmetros para os algoritmos de janela de Parzen e k-vizinhos. Depois são analisadas as médias de erros de cada classificador e finalmente são comparados os classificadores através do seu intervalo de confiança. Tanto para a seleção dos parâmetros quanto para os testes finais, os experimentos são idênticos: 100 repetições do hold-out. A aleatorização dos dados foi realizada utilizando uma semente igual ao número da iteração para que se possa obter os mesmos resultados em uma auditoria dos experimentos.

Para escolher o parâmetro h da janela de parzen, o experimento foi realizado para $h=1,0; 0,5; 0,25; 0,125; 0,0625; 2,0; 4,0; 8,0$ e $16,0$. Abaixo são exibidas a média, desvio padrão e intervalo de confiança para cada experimento. O valor final de h escolhido foi $h = 0,5$. Analisando os intervalos de confiança percebe-se que o erro é semelhante e que há interseção entre os intervalos de confiança para $h=1,0; 0,5; 0,25; 0,125; 0,0625$, que é o menor erro. Quanto maior o valor de h , maior a média do erro, porém em todos os casos há interseção entre os intervalos de confiança, desta forma não se pode comparar os erros reais dos classificadores, mas apenas suas médias de erros. Percebe-se também que o desvio padrão cresce juntamente com o parâmetro h , portanto

a estimativa da densidade pelo método de kernel é mais estável para h pequeno. Este é outro motivo de ter escolhido $h = 0,5$.

Janela de Parzen, $h = 1.0$

Erro médio (+-) desvio padrão = 0.022900000000000004(+-)0.012855738018488093

Intervalo de 95% confiança = [0.04840578422868038; -0.002605784228680373]

Janela de Parzen, $h = 0.5$

Erro médio (+-) desvio padrão = 0.022400000000000007(+-)0.012685424707119588

Intervalo de 95% confiança = [0.047567882618925264; -0.002767882618925254]

Janela de Parzen, $h = 0.25$

Erro médio (+-) desvio padrão = 0.029799999999999997(+-)0.016624078921853074

Intervalo de 95% confiança = [0.06278217258095649; -0.0031821725809564987]

Janela de Parzen, $h = 0.125$

Erro médio (+-) desvio padrão = 0.029499999999999988(+-)0.016417977951014545

Intervalo de 95% confiança = [0.06207326825481285; -0.0030732682548128727]

Janela de Parzen, $h = 0.0625$

Erro médio (+-) desvio padrão = 0.029499999999999988(+-)0.016417977951014545

Intervalo de 95% confiança = [0.06207326825481285; -0.0030732682548128727]

Janela de Parzen, $h = 2.0$

Erro médio (+-) desvio padrão = 0.04339999999999998(+-)0.027823012058366358

Intervalo de 95% confiança = [0.09860085592379883; -0.011800855923798877]

Janela de Parzen, $h = 4.0$

Erro médio (+-) desvio padrão = 0.07089999999999996(+-)0.05395247909039953

Intervalo de 95% confiança = [0.17794171851535262; -0.03614171851535271]

Janela de Parzen, $h = 8.0$

Erro médio (+-) desvio padrão = 0.07809999999999996(+-)0.06099303238895404

Intervalo de 95% confiança = [0.19911017625968477; -0.04291017625968485]

Janela de Parzen, $h = 16.0$

Erro médio (+-) desvio padrão = 0.08079999999999997(+-)0.06364243867106285

Intervalo de 95% confiança = [0.2070665983233887; -0.04546659832338874]

Para escolher o valor de k , o experimento foi realizado para $k = 1; 3; 5; 7; 9; 11; 13; 15; 17$. Só valores ímpares foram selecionados para evitar empates na classificação. O desvio padrão foi semelhante para todos os casos, porém a média de erro caiu mais para $k = 5$ e este foi o valor escolhido. Percebe-se que apesar de haver interseção entre os intervalos de confiança, para $k = 5; 7$ foi onde ocorreu o menor valor para no intervalo de confiança. Como o intervalo de confiança tem 95% de conter o erro real, tem-se mais chance de se obter o menor erro real para $k = 5; 7$ do que para os outros valores de k . As médias, desvios padrões e intervalos de confiança desses experimentos estão descrito abaixo.

k-vizinhos, k = 17

Erro médio (+-) desvio padrão = 0.023399999999999997(+-)0.0130430057885443

Intervalo de 95% confiança = [0.04927732348447189; -0.002477323484471893]

k-vizinhos, k = 15

Erro médio (+-) desvio padrão = 0.023100000000000006(+-)0.012928650354928784

Intervalo de 95% confiança = [0.048750442304178715; -0.002550442304178703]

k-vizinhos, k = 13

Erro médio (+-) desvio padrão = 0.022500000000000006(+-)0.012718097341976901

Intervalo de 95% confiança = [0.04773270512648218; -0.002732705126482167]

k-vizinhos, k = 11

Erro médio (+-) desvio padrão = 0.022200000000000008(+-)0.012622202660391737

Intervalo de 95% confiança = [0.047242450078217216; -0.0028424500782171964]

k-vizinhos, k = 9

Erro médio (+-) desvio padrão = 0.020200000000000013(+-)0.012154011683390795

Intervalo de 95% confiança = [0.04431355917984735; -0.003913559179847325]

k-vizinhos, k = 7

Erro médio (+-) desvio padrão = 0.018800000000000001(+-)0.012014990636700468

Intervalo de 95% confiança = [0.042637741423213735; -0.005037741423213716]

k-vizinhos, k = 5

Erro médio (+-) desvio padrão = 0.018500000000000013(+-)0.012006248373242994

Intervalo de 95% confiança = [0.04232039677251412; -0.005320396772514089]

k-vizinhos, k = 3

Erro médio (+-) desvio padrão = 0.0235(+-)0.013082430966758433

Intervalo de 95% confiança = [0.049455543038048735; -0.002455543038048732]

k-vizinhos, k = 1

Erro médio (+-) desvio padrão = 0.029499999999999988(+-)0.016417977951014545

Intervalo de 95% confiança = [0.06207326825481285; -0.0030732682548128727]

Finalmente os testes com todos os classificadores e com a sua combinação pela soma. As médias, desvios padrões e intervalos de confiança desses experimentos estão descrito abaixo. Não se pode afirmar que algum desses classificadores tem o menor erro para o problema em questão pois os intervalos de confiança são praticamente os mesmos para todos os classificadores. Percebe-se que a média do erro dos métodos paramétricos EM e Máxima Verossimilhança foi a menor de todas e muito próxima da média do erro para o método de k-vizinhos. Percebe-se também, que o método de combinação pela soma gera uma média de erro menor do que qualquer classificador isoladamente, confirmando a hipótese de que classificadores diferentes erram em regiões distintas do espaço e que a taxa de erro diminui ao combiná-los, muito embora a diminuição observada não tenha sido significativa, isto é, não se pode afirmar que o erro real diminuiu apenas afirma-se que a média dos erros foi reduzida.

EM e Máxima Verossimilhança

Erro médio (+-) desvio padrão = 0.018300000000000001(+-)0.012004582458378137

Intervalo de 95% confiança = [0.04211709159742223; -0.005517091597422214]

Janela de Parzen, $h = 0.5$

Erro médio (+-) desvio padrão = 0.022400000000000007(+-)0.012685424707119588

Intervalo de 95% confiança = [0.047567882618925264; -0.002767882618925254]

k-vizinhos, $k = 5$

Erro médio (+-) desvio padrão = 0.0185000000000000013(+-)0.012006248373242994

Intervalo de 95% confiança = [0.04232039677251412; -0.005320396772514089]

Combinação pela soma

Erro médio (+-) desvio padrão = 0.0179000000000000013(+-)0.012011244731500562

Intervalo de 95% confiança = [0.04173030954729713; -0.005930309547297102]

5. Conclusão

Este trabalho faz um estudo de vários algoritmos de aprendizagem de máquina. Na primeira parte é usado um algoritmo de agrupamento. Na segunda parte são testados e comparados vários classificadores individualmente e combinados.

O agrupamento parte de um conjunto de dados gerado aleatoriamente a partir de três distribuições de dados bivariados. Nessa amostra de dados foi aplicado o algoritmo do K-Means com 2 clusters 100 vezes e escolhido, destas 100 execuções, o que teve um melhor (nesse caso, o menor) critério de adequação, tendo no grupo 1, 189 padrões, e no grupo 2, 111 padrões. Para esse melhor resultado foram calculadas as taxas de erro por classe e global. No grupo 1 foi encontrado a taxa de erro igual a 21,16%; e no grupo 2 a taxa de erro foi de 0,9%. Já a taxa de erro global foi de 13,67%. Após isso, foi calculado o índice de *Rand* corrigido, que resultou no valor 0,53 (aproximadamente), o que representa o quanto os grupos formados se adequam à amostra real.

Os experimentos foram realizados calculando a média, desvio padrão e intervalo de confiança da taxa de erro de 100 repetições de hold-out. Após a escolha dos melhores valores de h para a janela de parzen e k para o k-NN, todos os classificadores apresentaram desempenho equivalente com aproximadamente 2% de erro, e com intervalo de confiança [5%,0], para um nível de 95% de confiança o erro real desses classificadores é menor que 5%.

Os classificadores de abordagem paramétrica foram tão bons quanto aqueles que abordagem não paramétrica. Atribui-se esse fato aos dados utilizados para testar seguirem uma distribuição normal. Testes em bases de dados reais possivelmente mostrariam uma melhor performance do métodos não paramétricos.

A combinação pela soma, apesar de não ser significativamente melhor que os métodos individuais, apresentou uma menor taxa de erro, o que é um indicativo de que de que classificadores diferentes erram em regiões distintas do espaço e que a taxa de erro diminui ao combiná-los.

Referências

- Triola. Introdução à Estatística. LTC, 2005.
- Josef Kittler, Mohamad Hatef, Robert P.W. Duin, and Jiri Matas (1998) “On Combining Classifiers”, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 20, NO. 3, MARCH 1998
- Richard O. Duda, Peter E. Hart and David G. Stork (2000) “Pattern Classification, 2nd edition”, Wiley
- Tiago B. A. de Carvalho (2007) “Um Estudo Sobre Funções De Distancia Aplicadas A Algoritmos De Aprendizagem De Maquina”, Universidade Federal de Pernambuco – Centro de informática – Trabalho de Graduação, <http://www.cin.ufpe.br/~tg/2007-1/tbac.pdf>
- D. R. Wilson and T. R. Martinez (1997) “Improved Heterogeneous Distance Functions”, J. Artificial Intelligence Research, vol.6, pp.1- 34
- Francisco de Assis Tenório de Carvalho (2012). Notas de aula da disciplinas de aprendizagem de máquina. <http://cin.ufpe.br/~fatc/AM/>