

Universidade Federal de Pernambuco

Centro de Informática

Relatório do Projeto de Aprendizagem de Máquina

Autor: Mário Barbosa e José Júnior de Oliveira Silva

Professor: Francisco de Assis Tenório Carvalho

Maio de 2012

[1. Introdução 3](#_Toc324366911)

[2. Classificadores 4](#_Toc324366912)

[2.1. Estimação Paramétrica 4](#_Toc324366913)

[2.1.1. Máximo de Verossimilhança (Maximum Likelihood) 5](#_Toc324366914)

[2.1.2. Estimação Bayesiana 5](#_Toc324366915)

[2.1.3. Densidades de Misturas 5](#_Toc324366916)

[2.1.3.1. Expectation Maximization 6](#_Toc324366917)

[2.2. Janela de Parzen 6](#_Toc324366918)

[2.3. K-Vizinhos 6](#_Toc324366919)

[3. Combinação de Classificadores 7](#_Toc324366920)

[3.1. Regra do produto 7](#_Toc324366921)

[3.2. Regra da Soma 7](#_Toc324366922)

[3.3. Regra do Máximo 7](#_Toc324366923)

[3.4. Regra do Mínimo 8](#_Toc324366924)

[3.5. Regra da Mediana 8](#_Toc324366925)

[4. Experimentos 9](#_Toc324366926)

[4.1. Conjunto de Dados Artificiais 9](#_Toc324366927)

[4.2. Resultados 10](#_Toc324366928)

[Questão 1 10](#_Toc324366929)

[Questão 2.a (Estimação Paramétrica) 11](#_Toc324366930)

[Questão 2.b (Janela Parzen) 12](#_Toc324366931)

[Questão 2.c (K-NN) 13](#_Toc324366932)

[Questão 2.d (Soma) 14](#_Toc324366933)

[Questão 3 (Avaliação dos Classificadores) 15](#_Toc324366934)

[5. Conclusão 16](#_Toc324366935)

[6. Referências 17](#_Toc324366936)

1. Introdução

Este trabalho implementa três classificadores estatísticos amplamente conhecidos (Estimação Paramétrica de Misturas, k-NN, Janelas de Parzen). Esses classificadores são executados sobre um conjunto de dados artificiais, a fim de avaliar os resultados obtidos. Tal avaliação foi feita de dois modos: verificando os resultados obtidos individualmente por cada classificador, e pela combinação deles segundo regras conhecidas da literatura.

Este relatório se divide da seguinte forma: na Seção 2 será feita uma descrição dos classificadores utilizados, na Seção 3 faz-se uma breve descrição de como combinar classificadores, na Seção 4 serão mostrados os experimentos realizados, bem como os resultados obtidos, e finalmente na Seção 5, algumas considerações são apresentadas.

1. Classificadores

A classificação é a atividade de relacionar um exemplo, submetido ao sistema (classificador), a uma classe a que ele possa pertencer, partindo apenas de conhecimento alcançado por experiência nas observação e análise de dados de treinamento num processo de aprendizagem. Neste projeto foi utilizado o modelo de aprendizagem supervisionada, onde os valores de entrada são associados a uma classe previamente conhecida. Foram implementados os seguintes algoritmos de classificação: K-vizinhos, janela de Parzen e estimador paramétrico de misturas utilizando o método da máxima verossimilhança e *expectation maximization*. A seguir será feita uma breve descrição dos conceitos dos classificadores utilizados.

# Estimação Paramétrica

Um classificador ótimo necessitaria conhecer de antemão as probabilidades a priori e a densidade condicional das classes, porém num caso real típico isso raramente acontece. Na maioria dos casos tem-se apenas um conhecimento muito geral da situação e um certo número de padrões (conjunto de treinamento) que deseja-se classificar.

A técnica de estimação paramétrica visa estimar as probabilidades a priori e densidade condicional das classes extraindo esses dados a partir dos padrões de treinamento, e utiliza essas estimativas como se fosse os valores de distribuições reais dos padrões.

Essa abordagem possuir algumas dificuldades. A estimação da função de densidade costuma não ser tão precisa quando o número de padrões de treinamento são pequenos, e principalmente quando a dimensionalidade do vetor de atributos é grande. Para simplifica o problema podemos supor que:

Onde os parâmetros e são desconhecidos, e em vez de estimar função , estimamos e .

Existem duas principais abordagens para a estimação de parâmetros: Máximo de verossimilhança e estimação bayesiana.

## Máximo de Verossimilhança (*Maximum Likelihood*)

O máximo de verossimilhança visa estimar os parâmetros que maximizam a verossimilhança entre os padrões de treinamento. Procurando os melhores parâmetros capazes de representar a distribuição.

Supondo:

onde então

Utilizamos informações das amostras para estimar , supondo que as classes sejam independentes, e considera-se que o valor que se quer estimar é fixo e existente.

## Estimação Bayesiana

Na estimação bayesiana os parâmetros que deseja-se estimar são obtidos utilizando todo o conhecimento disponível. A informações para estimar os parâmetros pode estar no conhecimento a priori do problema, que pode ser a forma funcional das densidades ou intervalo dos valores dos parâmetros desconhecidos. Outra parte pode estar nas amostras de treinamento. Se representa um conjunto de amostras, então enfatiza-se o papel das amostras dizendo que o objetivo é computar . Dessa forma a fórmula de Bayes torna-se

Como a equação sugere, a informação advinda do conjunto de exemplos pode ser usada para determinar as densidades condicionais e as probabilidades a priori.

Diferente da estimação paramétrica, onde os parâmetros são desconhecidos e estáticos, na estimação bayesiana os parâmetros são desconhecidos porém são dinâmicos.

A Máximo de Verossimilhança é mais simples, e apresenta boas propriedades de convergência à medida que o número de exemplos aumenta, diferente da estimação bayesiana, sendo mais complexa computacionalmente e menos interpretável.

## Densidades de Misturas

A densidade de misturas são aplicados quando a estrutura de probabilidades do problema já é conhecido, exceto os valores dos parâmetros. As probabilidades a priori e densidade de probabilidade condicional das classes são conhecidos, mas não os rótulos dos exemplos.

A função de densidade para os exemplos é composto por:

onde

### Expectation Maximization

Algoritmo que permite aprender os parâmetros de uma distribuição a partir de amostras de treinamento que não estejam completas.

# Janela de Parzen

Janela de Parzen constitui um método para estimação de densidade. Para isto, assume-se uma função de kernel, *K(x)*,que pode ser de diferentes tipos. Essa função determina uma janela centrada em *x* com largura *h*, que é utilizada para controlar a suavização. Para obter a função de densidade de probabilidade de uma amostra estimada por este método é necessário adicionar os valores de cada kernel, e escalar o valor para que o integral da função dê 1.

Para estimar a densidade utilizando este classificador, fixa-se o volume, a partir da função de kernel, determinando *k* a partir dos dados de aprendizagem. Dois fatores influenciam esta abordagem, um é a escolha da função kernel e o outro é a largura da janela, pois, se *h* for muito pequeno as fronteiras serão muito especializadas e se *h* for muito grande irá generalizar demais. Portanto, encontrar um valor ideal para *h* não é uma tarefa simples, mas pode ser conseguido utilizando-se uma base de validação.

# K-Vizinhos

O *k-Nearest Neighbors* atua considerando a densidade das classes no local do padrão de teste fornecido como entrada ao sistema. Dado um padrão *X*, são observados seus *k* vizinhos mais próximos, utilizando para isso alguma medida de distância previamente determinada, e a classe mais predominante dentre eles é dita como sendo a do padrão *X* de entrada. Dessa forma, pode-se observar que a estimação da densidade é baseada na probabilidade a posteriori. O classificador de K-vizinhos pode ser visto como uma janela de Parzen uniforme onde a largura da janela é ajustada automaticamente. A medida de distância mais comum para este classificador é a distância euclidiana.

1. Combinação de Classificadores

A combinação de classificadores visa a união do melhor entre dois (ou mais) "mundos", onde os "mundos", nesse caso, são os classificadores. Classificadores diferem no momento em que classificam um mesmo padrão de forma divergente. Portanto, a combinação de classificadores pode ser útil para driblar limitações de algumas técnicas, cada uma podendo fornecer informações complementares na tarefa de classificação.

Será apresentadas as cinco regras de classificação: regra do produto, soma, máximo, mínimo e mediana. A seguir será descrito brevemente o conceito relativo a cada uma delas.

Neste projeto foi feita a combinação de classificadores utilizando a regra da soma.

# Regra do produto

A combinação através da regra do produto é realizada pela multiplicação dos valores das probabilidades a posteriori obtidas pelos classificadores. Para uma dada categoria, a que obtiver maior valor da multiplicação é escolhida para a rotulação do dado de entrada. Possui maior adequação quando a aplicação é realizada quando cada classificador recebe a entrada a partir de sensores diferentes.

# Regra da Soma

A regra da soma combina os classificadores através da soma dos valores das probabilidades a posterior obtidas pelos classificadores combinados. Para cada categoria, a que possuir a maior soma é escolhida para classificar o dado de entrada. A regra da soma não é muito sensível a erros na estimação de densidade, por isso, essa regra é adequada para classificadores com a mesma entrada de dados. Também é possível aplicar um sistema de pesos para a regra da soma.

# Regra do Máximo

A regra do máximo pode ser obtida como uma aproximação, limite superior, da regra da soma. Neste caso, a regra de decisão torna-se atribuir o elemento de entrada a uma determinada classe, dependendo do valor máximo dos classificadores combinados.

# Regra do Mínimo

A regra do mínimo pode ser obtida como uma aproximação da regra do produto. Representa um limite superior da regra do produto. A regra de decisão é determinada por atribuir o elemento de entrada ao maior mínimo dentre os resultados oriundos dos classificadores.

# Regra da Mediana

A combinação pode ser obtida lembrando que a regra da soma calcula a media das saídas dos classificadores e que uma estimativa robusta da média é a mediana.

1. Experimentos

Os experimentos foram realizados com dados gerados artificialmente, que serão descritos mais detalhadamente a seguir. A técnica de validação cruzada *v-fold-* *cross-validation* estratificada foi empregada nos testes, consiste em dividir os padrões em grupos de números iguais, tomando cuidado para que a quantidade de padrões de cada classe sejam igualmente distribuídos entre os grupos, e a classificação é realizada v vezes, onde um grupo é selecionado como teste e o restante como treinamento, permitindo uma melhor taxa de erro considerando toda a distribuição de padrões fornecidos.

# Conjunto de Dados Artificiais

Os dados consistem em 300 padrões, distribuídos igualmente entre 2 classes. Os padrões são distribuídos em 2 variáveis quantitativas, a média e a covariância. A primeira classe é composta apenas por uma distribuição normal bi-variada. O segundo padrão segue uma mistura de duas distribuições normais bi-variadas. Segue os parâmetros das distribuições:

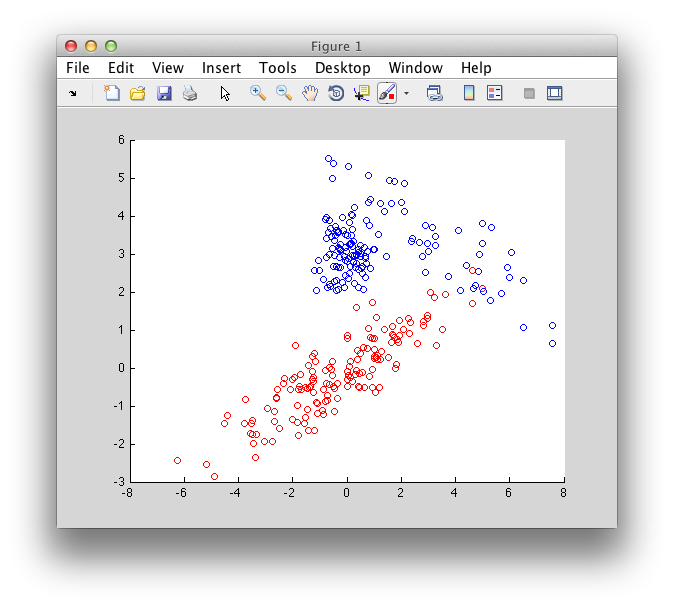
Classe 1

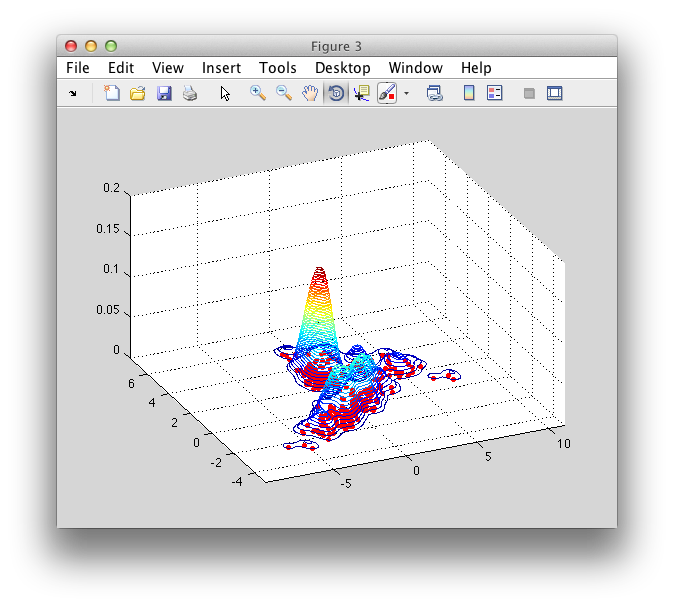
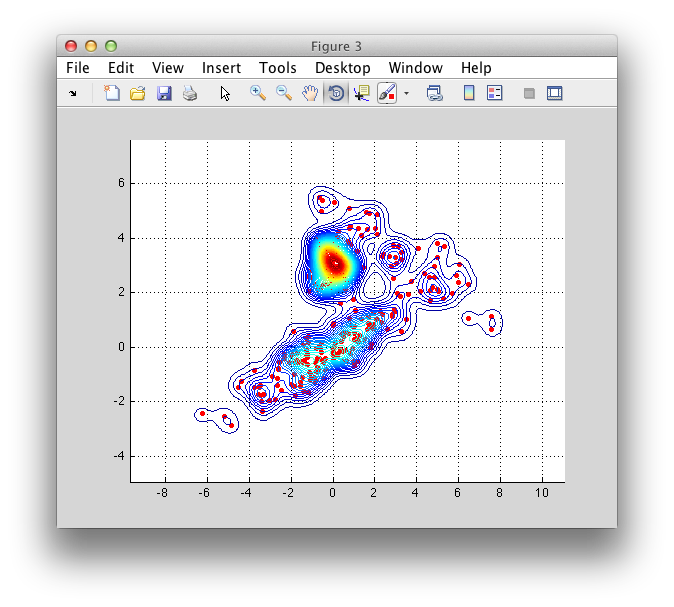
= 0, = 0, , ,

Classe 2

= 0, = 3, ,

= 4, = 3, ,

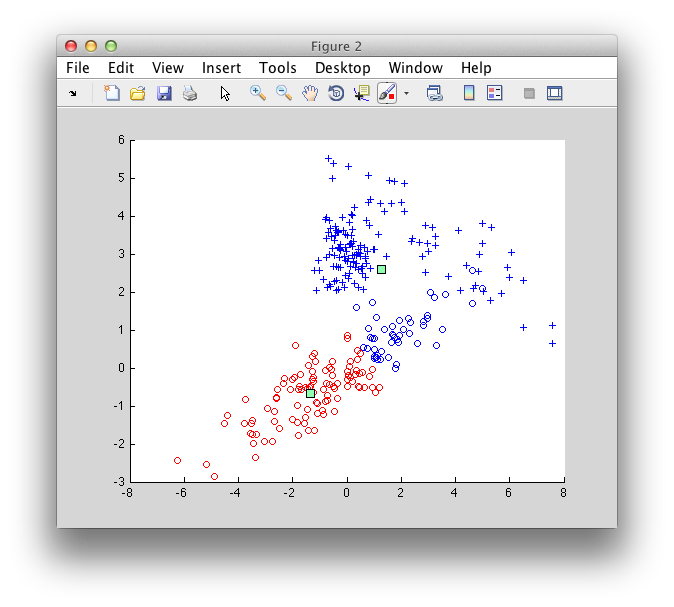
Aplicando os parâmetros das distribuições obteve-se o seguinte gráfico:



# Resultados

## Questão 1

Aplicar uma o algoritmo de *clustering* k-médias com 2 clusters 100 vezes selecionando o melhor resultado seguindo o critério de adequação entre os clusters e seus representantes.



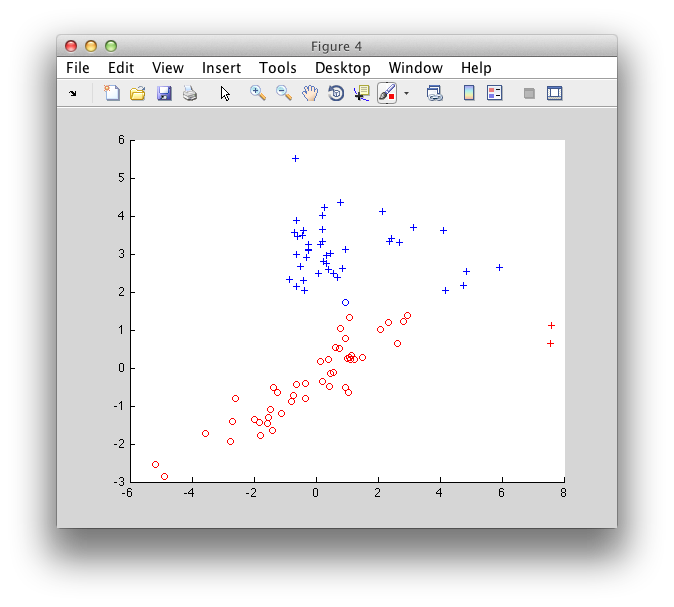
k-médias

Os padrões são divididos em duas classes representadas por (+ e O), as cores azul e vermelho são os resultados obtidos pelo agrupamento com o algoritmo de k-médias.

|  |  |
| --- | --- |
| Resultados Obtidos |  |
| Índice de Rand Corrigido | 0.4699 |
| Erro de Classificação Global | 0.1567 |
| Erro de Classificação na Classe 1 | 0.3133 |
| Erro de Classificação na Classe 2 | 0.0000 |

## Questão 2.a (Estimação Paramétrica)

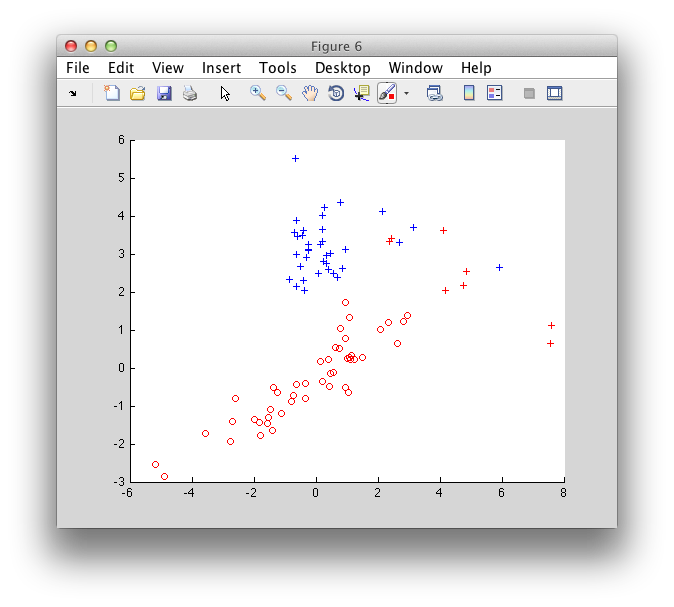
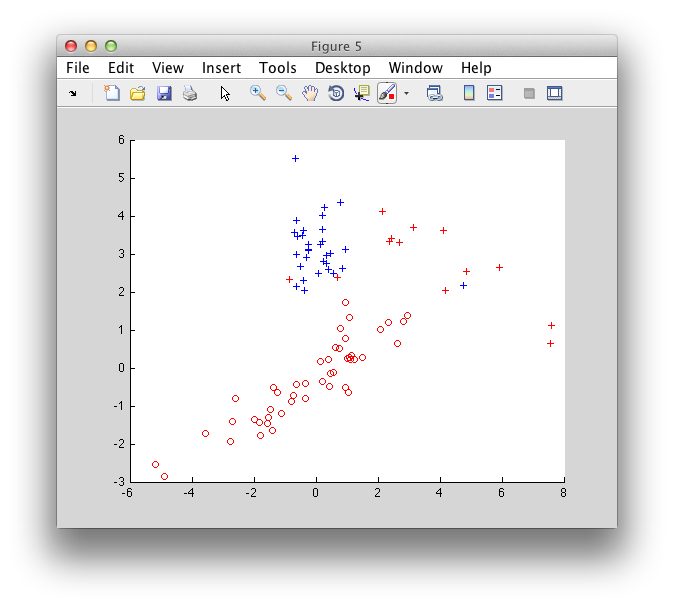
Estimar a Classe 1 utilizando o método da máxima verossimilhança e estimar a Classe 2 usando o algoritmo de *Expectation Maximization* supondo uma mistura de distribuições multivariadas.



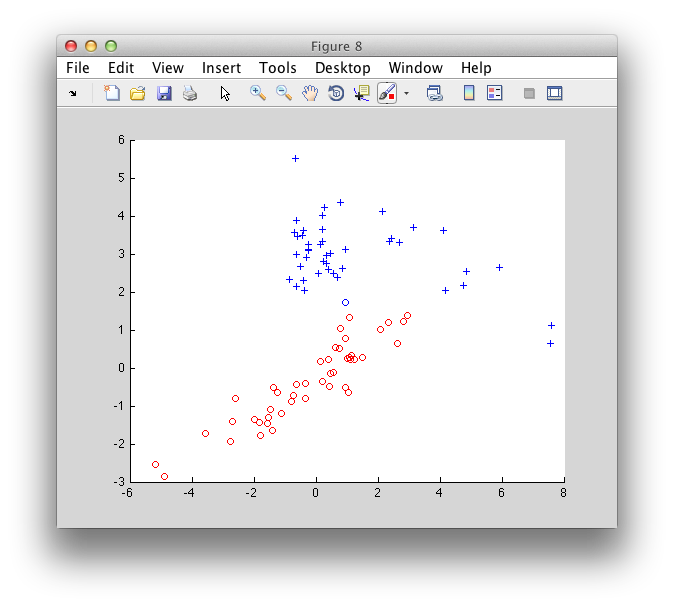
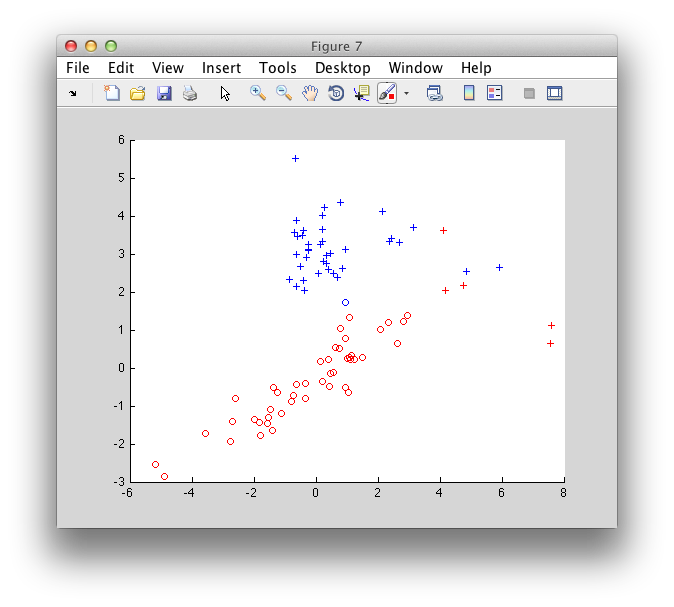
|  |  |
| --- | --- |
| Resultados Obtidos |  |
| Erro de Classificação Global | 0.0333 |
| Erro de Classificação na Classe 1 | 0.0217 |
| Erro de Classificação na Classe 2 | 0.0455 |

## Questão 2.b (Janela Parzen)

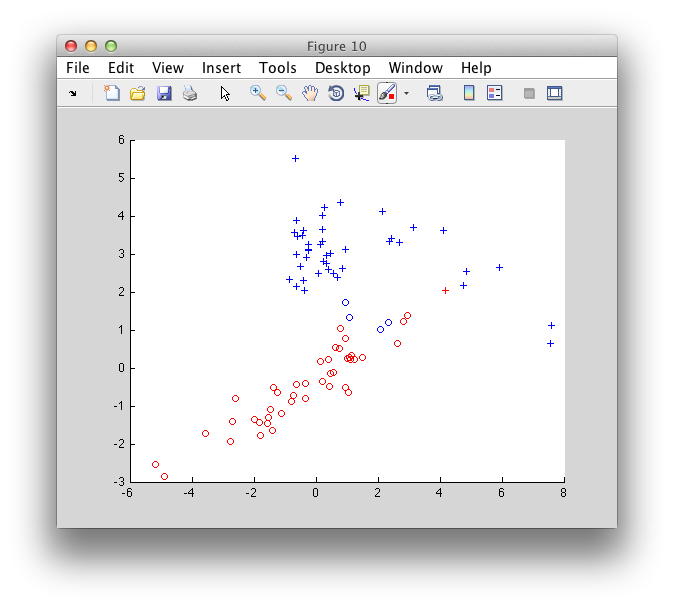
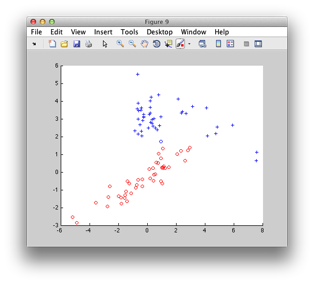
Classificar utilizando a Janela Parzen variando o parâmetro de suavização h



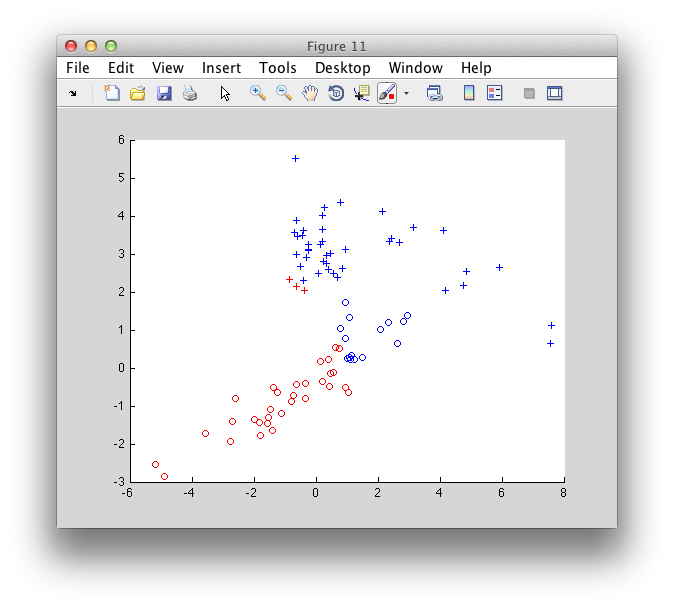
h = 0.1 h = 0.4



h = 1 h = 5



h = 10 h = 20

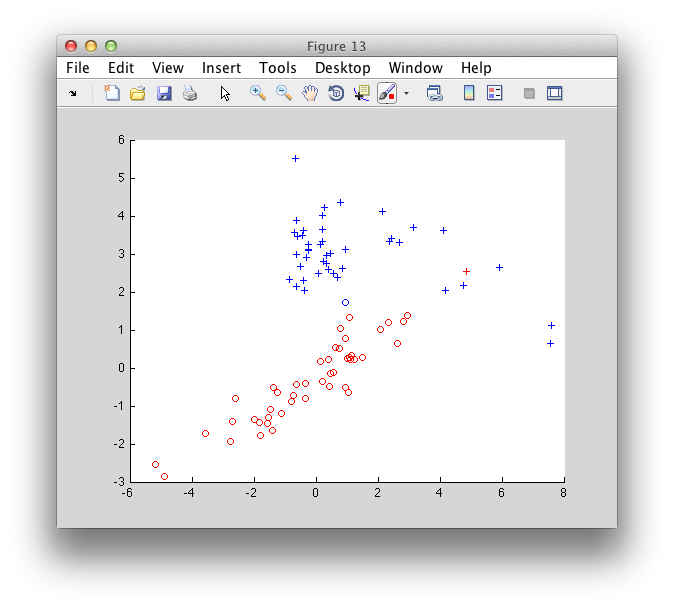
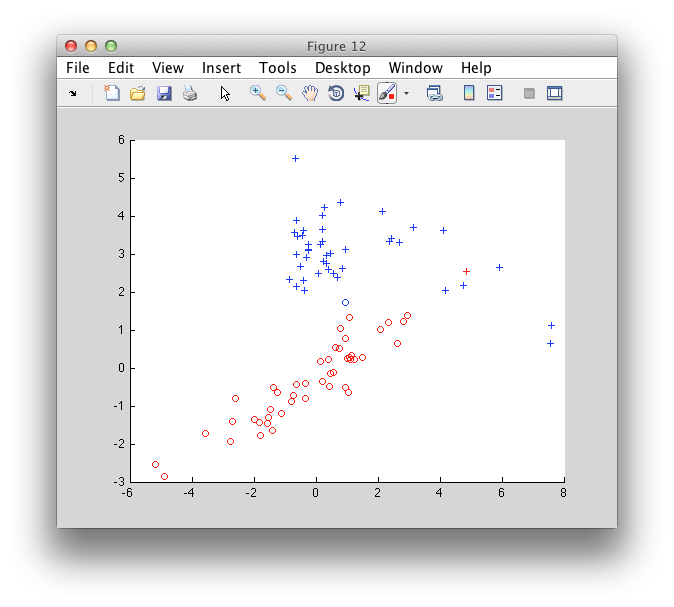


h = 50

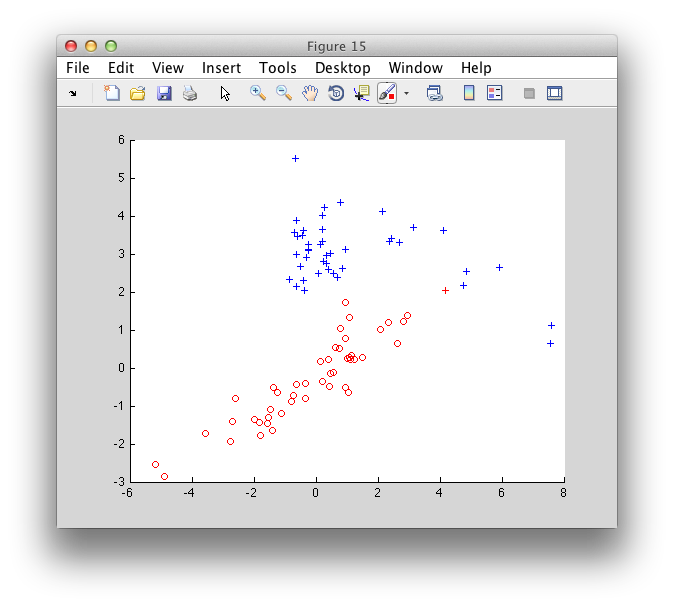
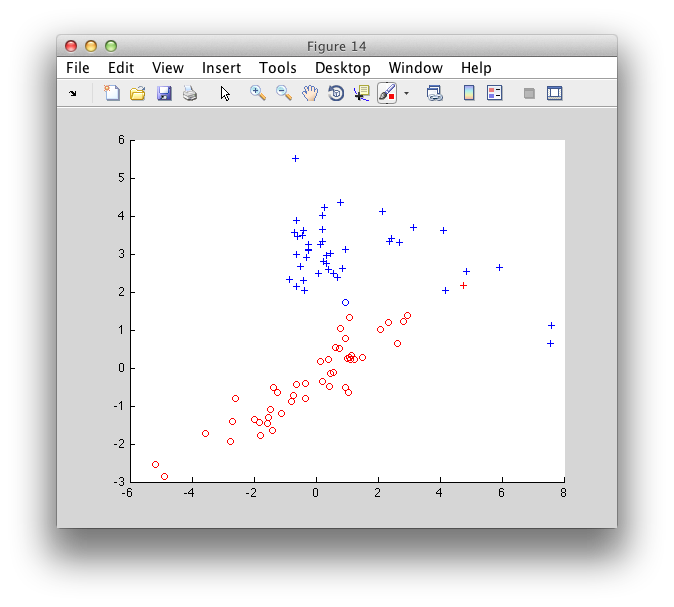
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Resultados (h) | 0.1 | 0.4 | 1 | **5** | **10** | 20 | 50 |
| Erro Global | 0.1444 | 0.0889 | 0.0667 | **0.0111** | **0.0111** | 0.0556 | 0.2000 |
| Erro Classe 1 | 0.0000 | 0.0000 | 0.0217 | **0.0217** | **0.0217** | 0.0870 | 0.3261 |
| Erro Classe 2 | 0.2955 | 0.1818 | 0.1136 | **0.0000** | **0.0000** | 0.0227 | 0.0682 |

## Questão 2.c (K-NN)

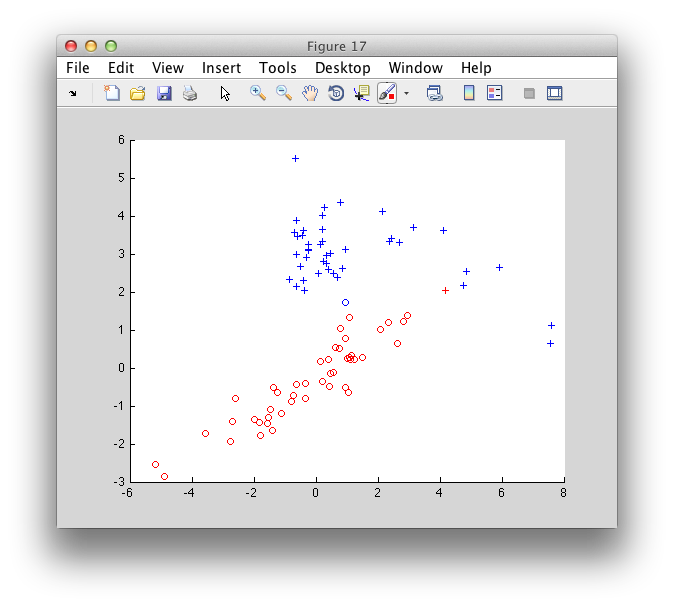
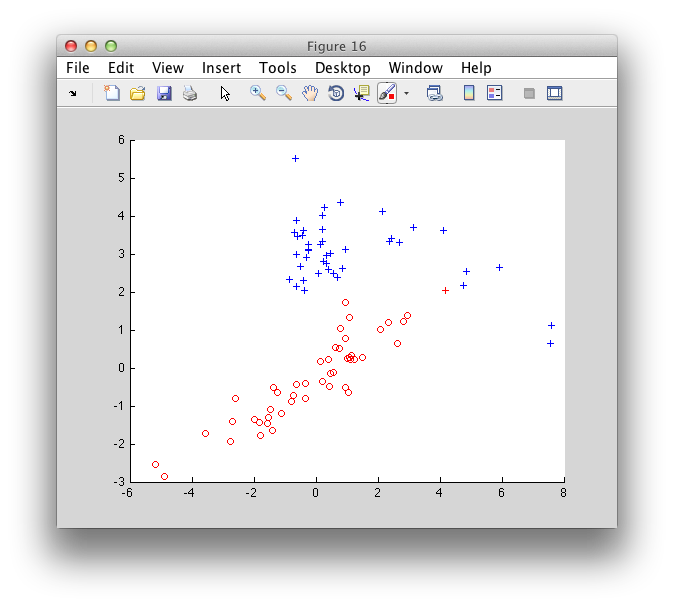
Classificar utilizando a distância euclidiana variando os vizinhos.



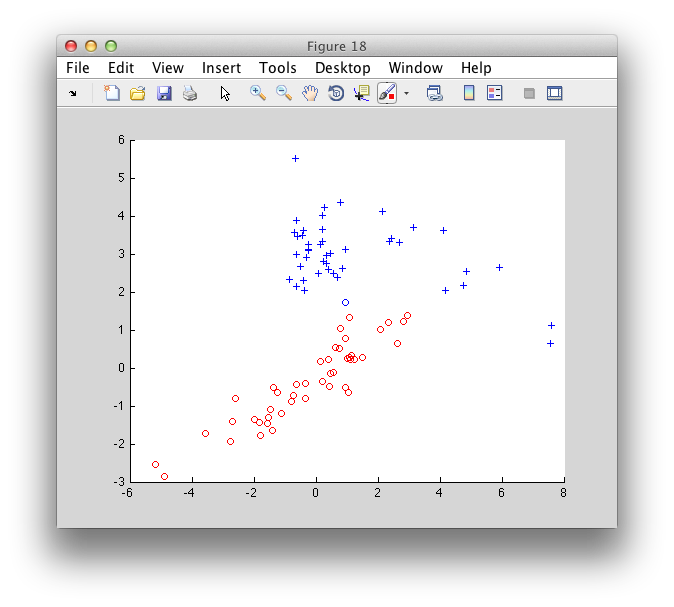
k = 1 k = 3



k=5 k=8



k = 10 k = 15



k = 20

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Resultados (knn) | 1 | 3 | 5 | 8 | 10 | 15 | 20 |
| Erro Global | 0.0222 | 0.0222 | 0.0222 | 0.0111 | 0.0111 | 0.0222 | 0.0111 |
| Erro Classe 1 | 0.0217 | 0.0217 | 0.0217 | 0.0000 | 0.0000 | 0.0217 | 0.0000 |
| Erro Classe 2 | 0.0227 | 0.0227 | 0.0227 | 0.0217 | 0.0217 | 0.0227 | 0.0217 |

## Questão 2.d (Soma)

Combinação dos classificadores anteriores utilizando o critério de soma. Janela de Parzen configurado com h=0.4 e Knn configurado com 3 vizinhos.

|  |  |
| --- | --- |
| Resultados |  |
| Erro Global | 0.0111 |
| Erro Classe 1 | 0.0217 |
| Erro Classe 2 | 0.0000 |

## Questão 3 (Avaliação dos Classificadores)

Utilizando a avaliação de *v-fold-cross-validation* com 10 grupos estratificados.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Resultados | Paramétrico | Parzen | KNN | Soma |
| Fold 1 | 0.0667 | 0.0333 | 0.0667 | 0.0667 |
| Fold 2 | 0 | 0.0333 | 0 | 0 |
| Fold 3 | 0 | 0 | 0 | 0 |
| Fold 4 | 0 | 0 | 0 | 0 |
| Fold 5 | 0 | 0 | 0 | 0 |
| Fold 6 | 0.0667 | 0.0667 | 0.0667 | 0.0667 |
| Fold 7 | 0 | 0.0333 | 0 | 0 |
| Fold 8 | 0.667 | 0.1 | 0.0333 | 0.0333 |
| Fold 9 | 0.0667 | 0.1 | 0.0667 | 0.0667 |
| Fold 10 | 0.0333 | 0.1333 | 0 | 0 |
| **Média** | **0.03** | **0.05** | **0.0233** | **0.0233** |

1. Conclusão

Observando os dados mostrados na Seção 4, pôde-se notar que os classificadores apresentaram diferentes taxas de erro. Com essas informações é possível fazer a escolha por um classificador que melhor solucione o problema. No entanto nesse tipo de abordagem os exemplos classificados incorretamente podem diferir de um classificador para outro. Isso pode fazer com que a escolha de um classificador em detrimento de um outro não seja a melhor solução para o problema.

Para solucionar esse problema pode ser usado a combinação dos classificadores a fim de fornecer informações complementares para a solução. Fazendo uso dessa técnica a taxa de erro tende a diminuir.

Neste projeto foi feita a combinação dos classificadores, de estimação paramétrica, Janela de Parzen com h=0.4 e Knn configurado com 3 vizinhos, utilizando o critério de soma. Os resultados obtidos comprovam que utilizar a combinação dos classificadores para esse caso é a abordagem mais adequada para o problema tendo em vista a diminuição das taxas de erro.

6. Referências

[1]Carvalho, F. (2012). Aprendizagem de Máquina  [http://www/~fatc/AM/]. Universidade Federal de Pernambuco, Curso de Mestrado em Ciência da Computação.

[2] Distribuição normal multivariada (http://en.wikipedia.org/wiki/Multivariate\_normal\_distribution)

[3] Índice de Rand (<http://en.wikipedia.org/wiki/Rand_index>)

[4] Máximo Verossimilhança (<http://en.wikipedia.org/wiki/Maximum_likelihood>)

[5] Máximo Verossimilhança (<http://www.math.utah.edu/~levin/M5080/mle.pdf>)

[6] Expectation Maximization (<http://mi.eng.cam.ac.uk/~mjfg/local/4F10/lect4.pdf>)

[7] Expectation Maximization Algorithm (http://en.wikipedia.org/wiki/Expectation%E2%80%93maximization\_algorithm)

[8] Density Estimation (<http://www.answers.com/topic/density-estimation>)

[9] Parzen Algorithm (<http://mi.eng.cam.ac.uk/~mjfg/local/4F10/lect13.pdf> )

[10]Duda, R. O., Hart, P. E., & Stork, D. G. (2001). Pattern classification (2nd ed.). New York: Wiley–Interscience.

[11] Mathlab Library

* http://mi.eng.cam.ac.uk/~mjfg/local/4F10/index.html
* http://cmp.felk.cvut.cz/cmp/software/stprtool/index.html
* http://cmp.felk.cvut.cz/cmp/courses/recognition/Labs/index\_en.html
* http://homepages.cae.wisc.edu/~ece738/matlab/index.html