

Universidade Federal de Pernambuco

Centro de Informática

Relatório do Projeto de Aprendizagem de Máquina

Autor: Mário Barbosa

Professor: Francisco de Assis Tenório Carvalho

Maio de 2012

1. Introdução

Este trabalho visa a execução de três classificadores estatísticos amplamente conhecidos (Estimação Paramétrica de Misturas, k-NN, Janelas de Parzen), sobre um conjunto de dados artificiais, para realização de avaliação de resultados. Além da avaliação dos resultados desses classificadores, será avaliado a combinação deles segundo regras conhecidas da literatura.

Primeiramente será mostrado as definições relativas ao problema abordado. Posteriormente será descrito como foi realizada a geração dos dados artificiais. Depois, será mostrado os resultados e a conclusão do problema abordado.

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. No caso deste projeto, será utilizado aprendizagem supervisionada. 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 não é isso que acontece. Tendo 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.

* Máxima Verossimilhança
* Estimação Bayesiana

## Máximo Verossimilhança (*Maximum Likelihood*)

A máximo verossimilhança visa estimar os parâmetros que maximize a verossimilhança entre os padrões de treinamento. Procurando os melhores parâmetros capaz 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 seque uma distribuição normal, onde o objetivo é encontrar o vetor de parâmetros que maximize a probabilidade de observar a amostra.

A Máximo 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 amostrar de treinamento que não estejam completas.

# Janela de Parzen

Janela de Parzen constitui um método para estimação de densidade. Para isto, assumi-se uma função de kernel, *K(x)*, existem diferentes tipos de funções de kernel. 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. Dessa maneira, podemos observar que este classificador para estimar a densidade, fixa-se o volume, a partir da função de kernel, e determina *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 é a distância euclidiana.

1. Combinação de Classificadores

A combinação de classificadores visa a união do melhor entre dois (ou mais) mundos, em que tais 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.

No projeto será implementado apenas 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 mínino 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 detalahdamente 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 consiste 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 seguindo padrão segue uma mistura de duas distribuições normais bi-variadas. Segue os parâmetros das distribuições:

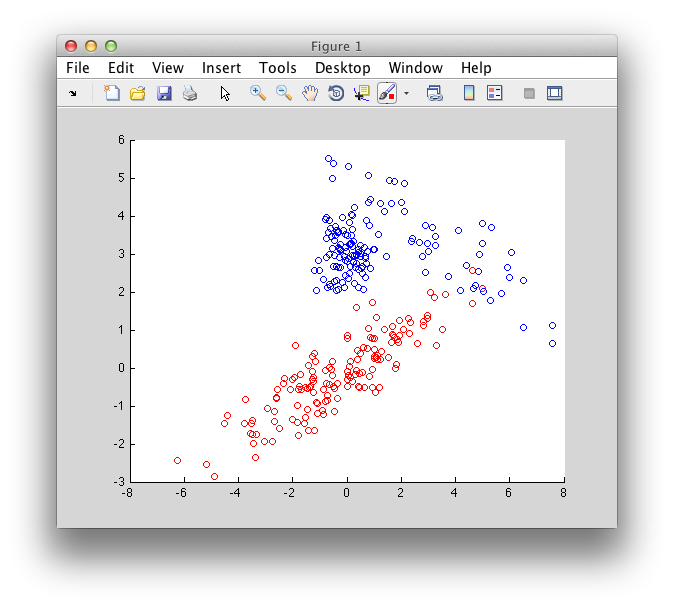
Classe 1

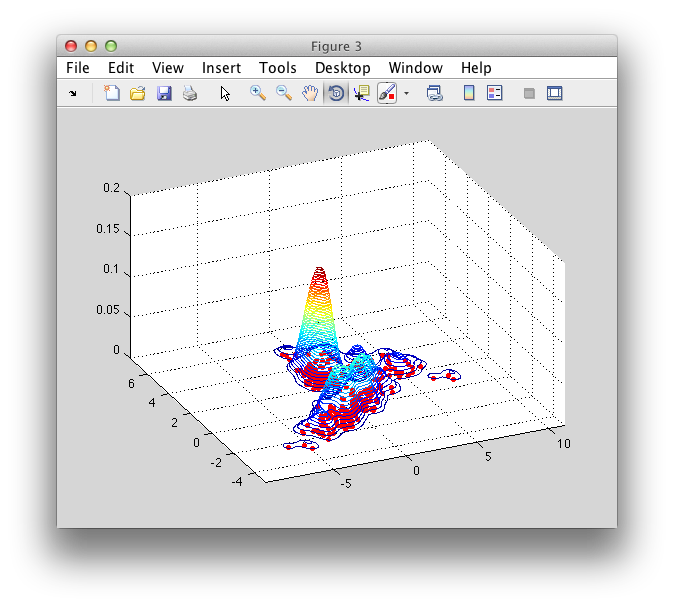
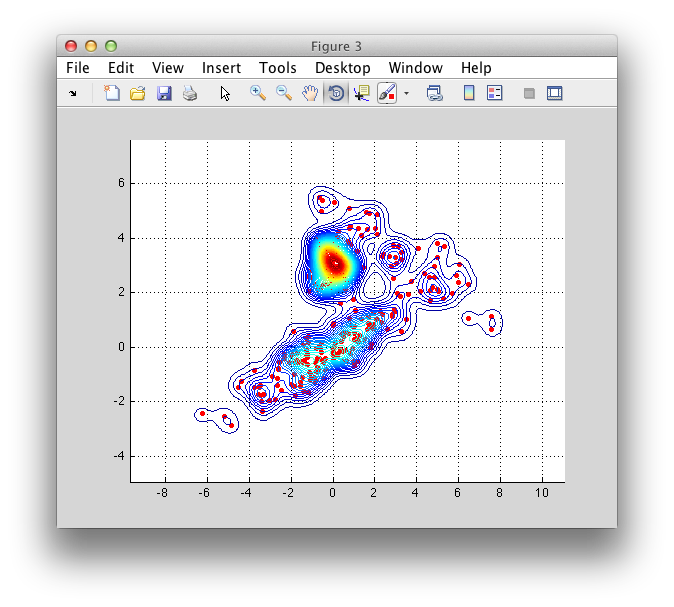
= 0, = 0, , ,

Classe 2

= 0, = 3, ,

= 4, = 3, ,

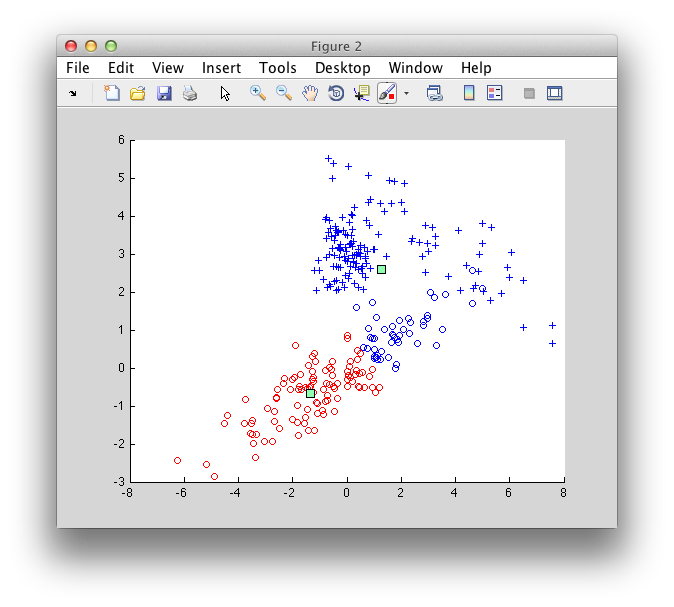
Aplicando os parâmetros das distribuições obteve-se o seguinte gráfico:



# 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.



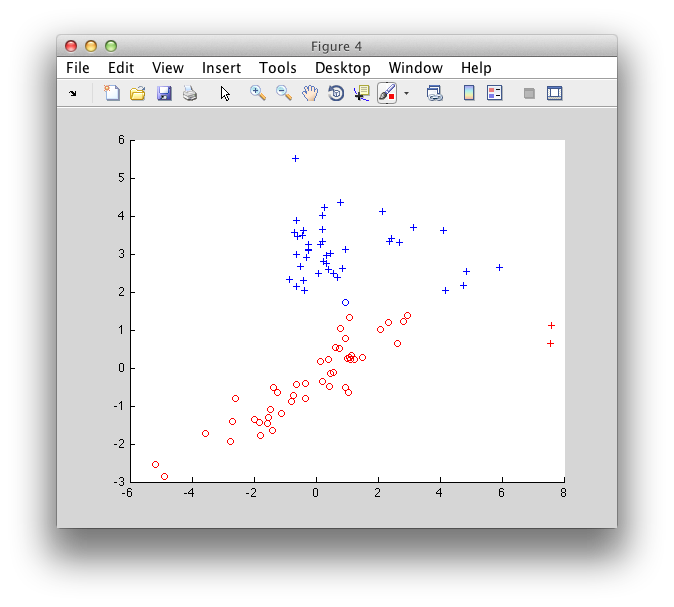
k-médias

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)

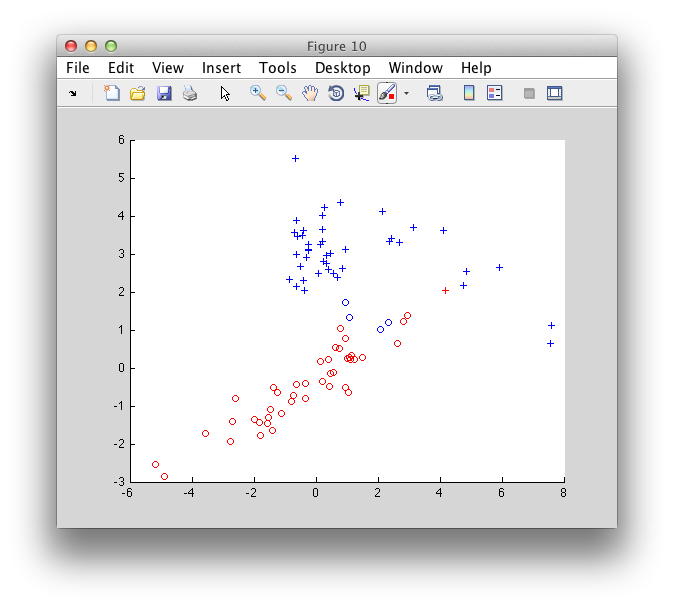
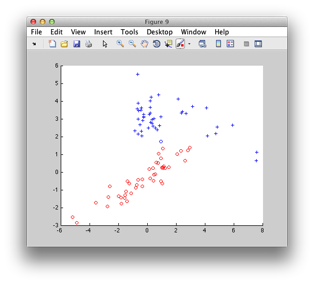
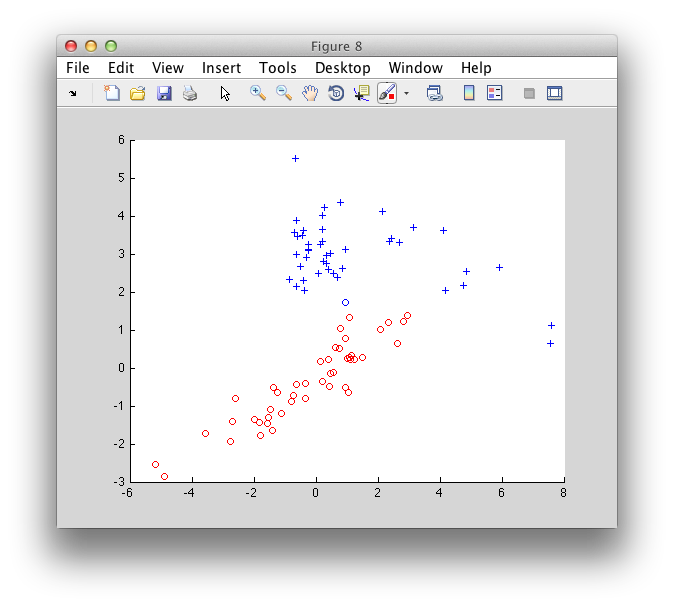
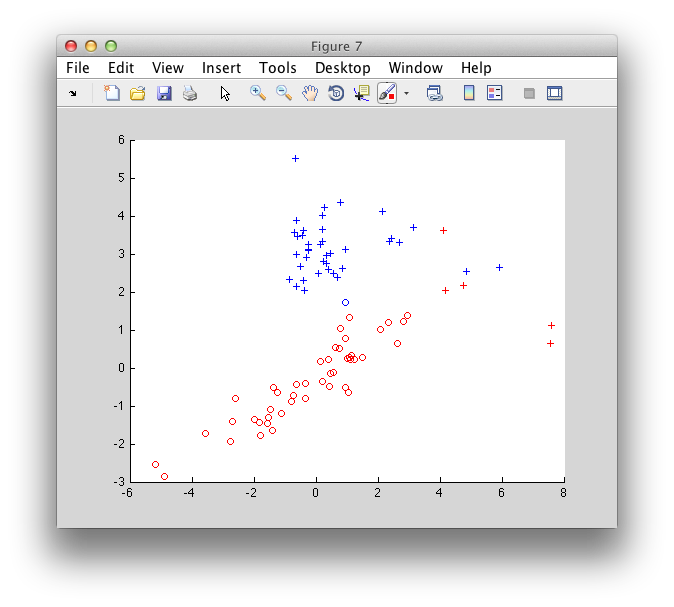
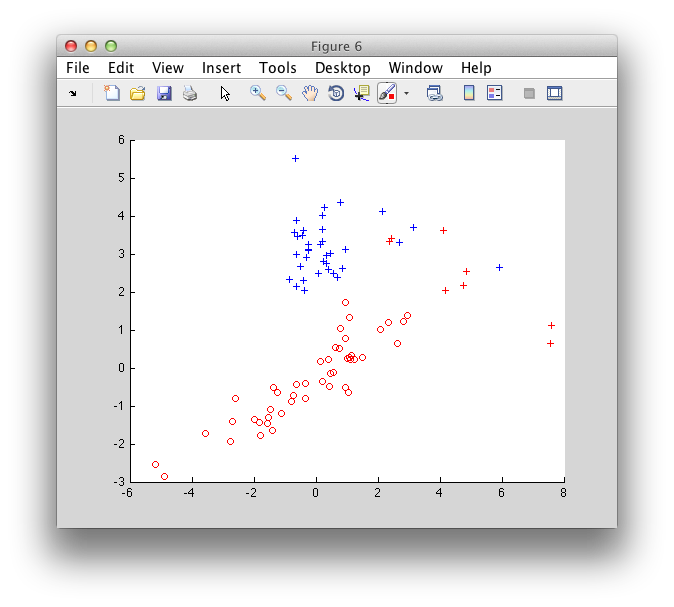
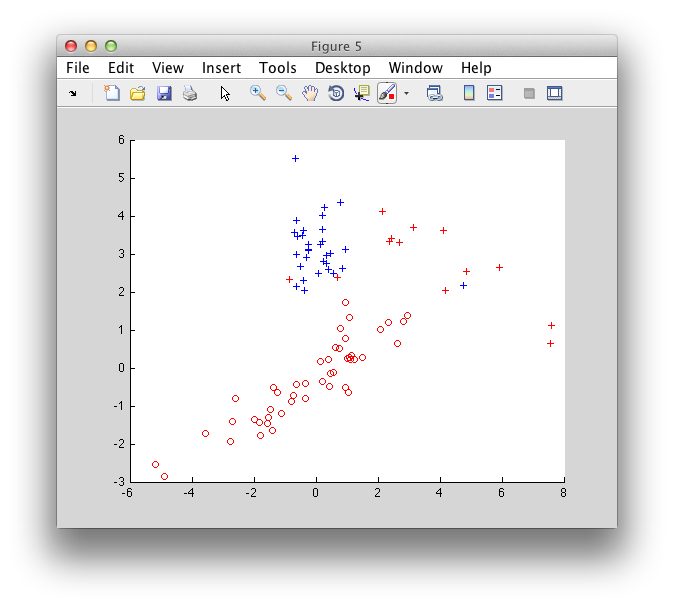
Estimar a Classe 1 utilizando o método da máxima verossimilhança e estimar a Classe 2 usando o algoritmo de Expectation Maximization supondo uma mistura de distribuições multivariadas.

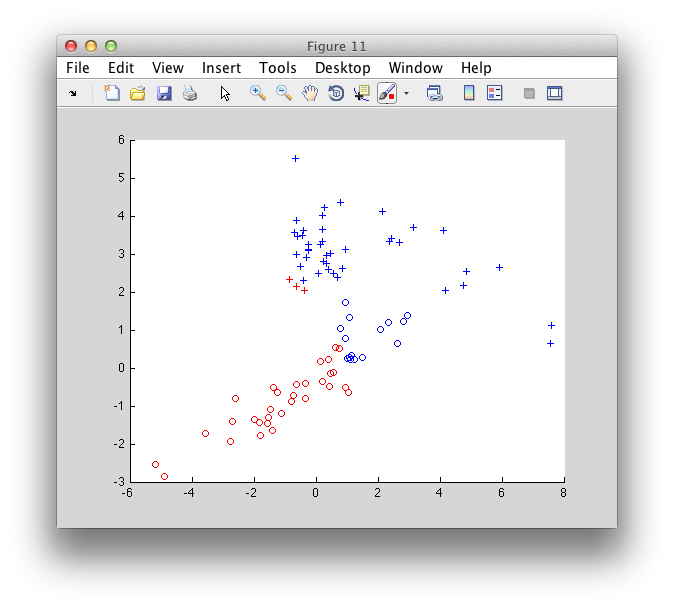


|  |  |
| --- | --- |
| Resultados Obtidos |  |
| Erro de Classificação Global | 0.0333 |
| Erro de Classificação na Classe 1 | 0.0217 |
| Erro de Classificação na Classe 2 | 0.0455 |

## Questão 2.b (Janela Parzen)

Classificar utilizando a Janela Parzen variando o parâmetro de suavização h

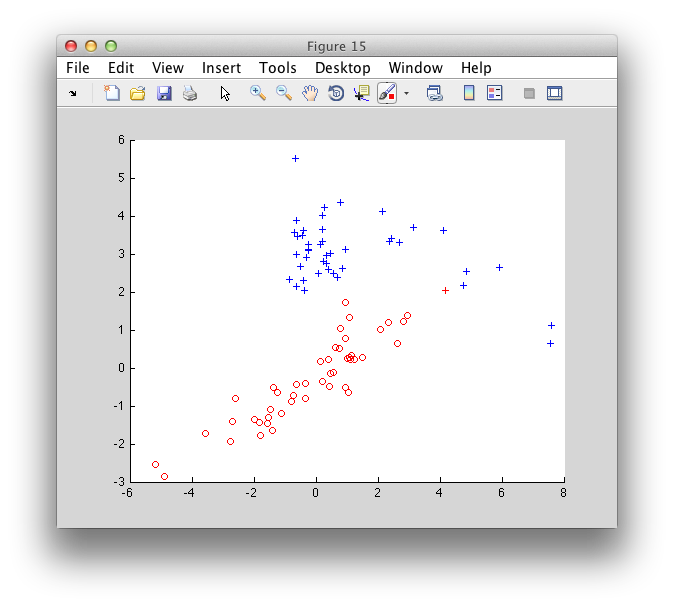
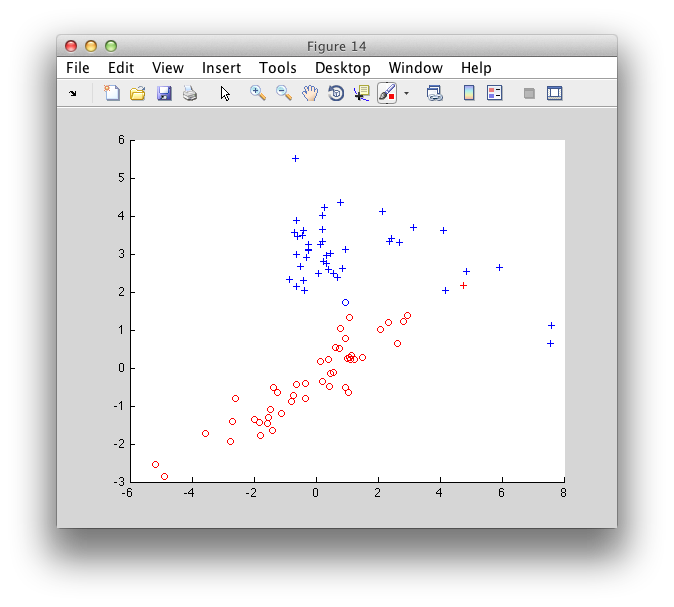
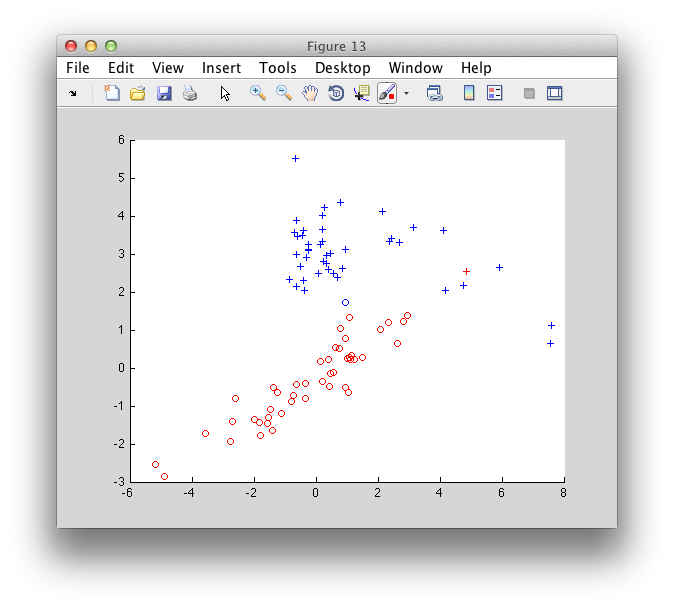
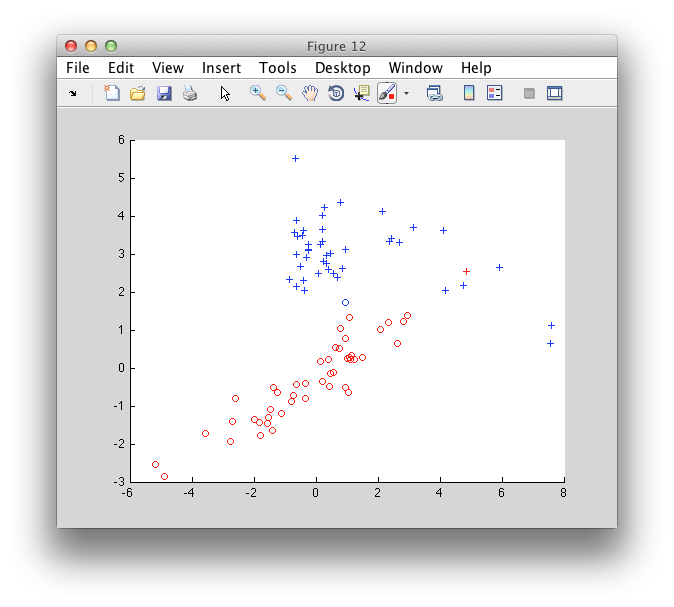


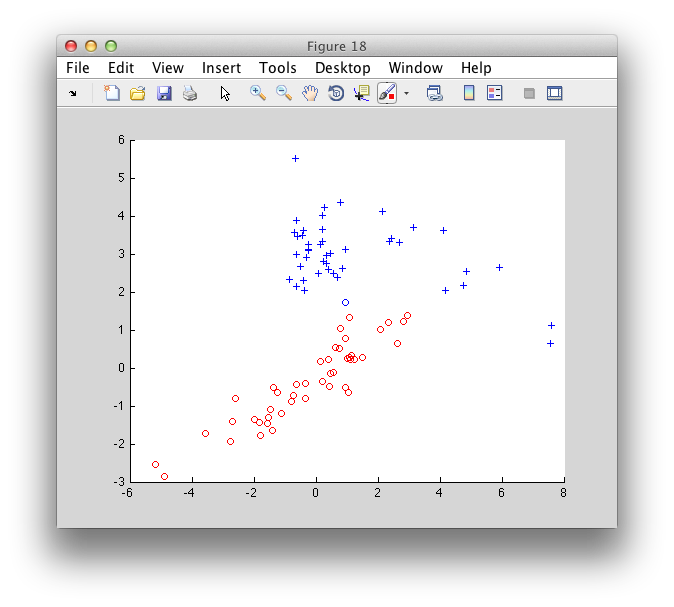
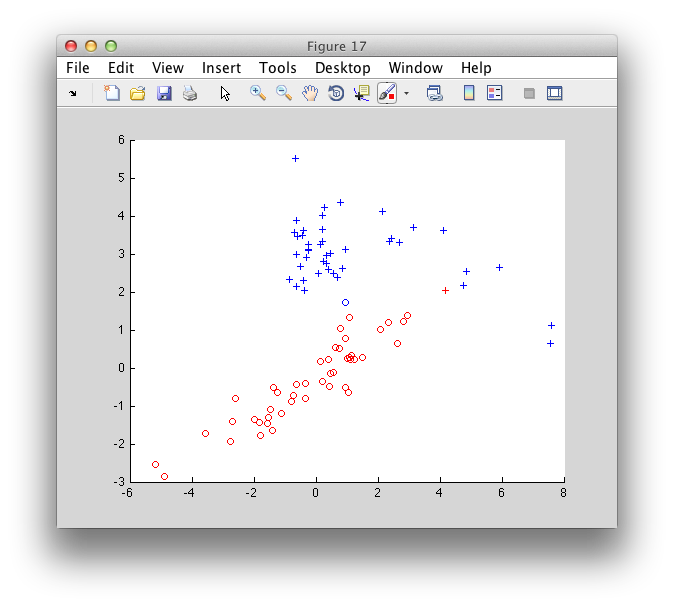
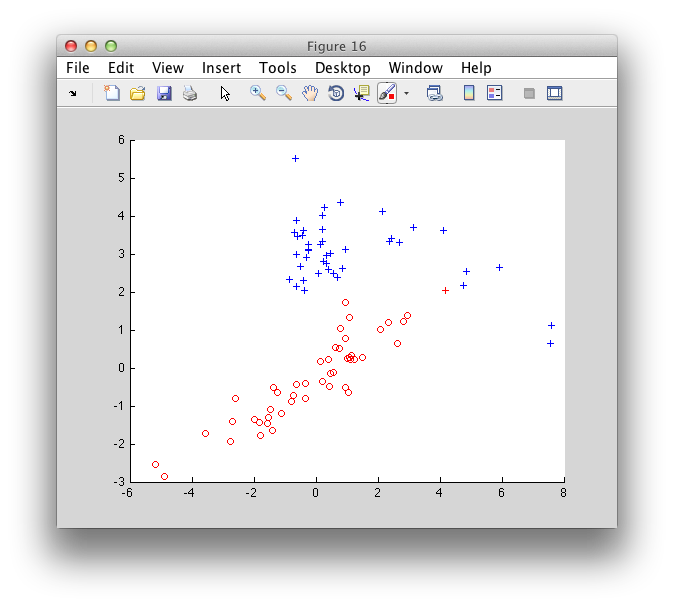


|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Resultados (h) | 0.1 | 0.4 | 1 | **5** | **10** | 20 | 50 |
| Erro Global | 0.1444 | 0.0889 | 0.0667 | **0.0111** | **0.0111** | 0.0556 | 0.2000 |
| Erro Classe 1 | 0.0000 | 0.0000 | 0.0217 | **0.0217** | **0.0217** | 0.0870 | 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.





|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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 Pazen 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

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