

Projeto de Aprendizagem de Máquina

Dayvid Victor Rodrigues de Oliveira e Denys Lins Farias

13 de maio de 2012

Resumo

ABSTRACT

Sumário

1	Introdução	1
1.1	Conceito	1
1.2	Objetivo	1
1.3	Estrutura do Trabalho	1
2	Técnicas Avaliadas	2
2.1	Contexto e Histórico	2
2.1.1	Sistemas de Reconhecimento de Padrões	2
2.2	Agrupamento	2
2.2.1	K-Means	3
2.3	Classificadores	3
2.3.1	K-Nearest Neighbor	3
2.3.2	Janela de Parzen	4
2.3.3	Estimação da Máxima Verossimilhança	5
2.3.4	Algoritmo Expectation-Maximization	5
2.4	Combinação de Classificadores	6
2.4.1	Regra da Soma	6
2.4.2	Regra do Produto	7
2.4.3	Regra do Máximo	7
2.4.4	Regra do Mínimo	7
2.4.5	Regra da Mediana	7
3	Experimentos	8
3.1	Geração de Dados	8
3.2	Avaliação	9
3.2.1	K-Means	10
3.2.2	Máxima Verossimilhança e Algoritmo EM	11
3.2.3	Janela de Parzen	12
3.2.4	KNN	13
3.2.5	Combinação de Classificadores	13
3.3	Comparação	14
4	Conclusão	15

Lista de Tabelas

3.1	Dados para geração do conjunto de padrões	8
3.2	Tabela da taxa de erro do K-Means	11
3.3	Tabela da taxa de erro do MLE e EM combinados	11
3.4	Tabela de erro da Janela de Parzen	13
3.5	Tabela de erro do KNN	13
3.6	Tabela de erro dos Classificadores Combinados	13

Lista de Figuras

3.1	Dados gerados para os experimentos	9
3.2	Clusters gerados pelo K-Means	10
3.3	Gráfico de erro global por larguras de janela	12

Capítulo 1

Introdução

1.1 Conceito

1.2 Objetivo

1.3 Estrutura do Trabalho

Capítulo 2

Técnicas Avaliadas

2.1 Contexto e Histórico

Esta seção abordará sistemas de reconhecimento de padrões e classificadores com aprendizagem baseadas em instâncias.

2.1.1 Sistemas de Reconhecimento de Padrões

No final dos anos 50, surgiram os primeiros trabalhos de aprendizagem de máquina. De uma forma geral, elas consistiam em dar ao computador a habilidade de reconhecer formas. A partir daí, surgiram diversos problemas onde a aprendizagem de máquina atuava.

Existem três problemas gerais que a aprendizagem de máquina tenta resolver. Um deles é o problema do agrupamento, que consiste em agrupar dados de acordo com suas características, de forma que seja possível extrair informação útil destes agrupamentos. Um outro problema é a discriminação, que basicamente é achar uma forma de reconhecer um conceito, dado um conjunto de conceitos exemplos. O terceiro e último problema é o da generalização, que é o problema de como reduzir uma regra de classificação, tornando-a mais abrangente e menos custosa.

Reconhecimento de padrões ataca principalmente o problema da discriminação, tendo por objetivo classificar padrões, discriminando-os entre duas ou mais classes. A classificação pode ser feita com padrões pertencentes a qualquer domínio, como reconhecimento de digitais, gestos, escrita, fala, entre outros.

2.2 Agrupamento

Agrupamento, também conhecido como Clusterização, é a classificação não-supervisionada de dados que forma grupos, ou clusters. Assim, a análise de clusters envolve a organização de um conjunto de padrões em agrupamentos, de acordo com algumas regras de similaridade.

De forma simples, pode-se dizer que a clusterização é o agrupamento de instâncias baseadas nas suas características, ou seja, instâncias similares pertencem ao mesmo grupo. Estes agrupamentos são formados por padrões não rotulados, que não se sabe a classe de cada padrão, assim, rótulos podem ser definidos após a realização da clusterização.

Uma vantagem da clusterização é que atualmente existem muitos dados, mas poucos destes dados são classificados. Algoritmos de clusterização possibilitam a rotulação de forma mais rápida e factível por um especialista.

2.2.1 K-Means

Um dos algoritmos de clusterização mais famosos é o K-means. Este método divide T padrões em K clusters onde cada padrão x pertence ao cluster de média mais próxima de x . O algoritmo é repetido até que poucas alterações sejam feitas em cada iteração. O resultado deste algoritmo é uma partição do espaço em Células de Voronoi.

Algorithm 1 K-Means

Require: K : um número de clusters desejados

Require: T : um conjunto de padrões

```
1:  $M =$  Obtenha  $K$  padrões aleatoriamente
2:  $G =$  Lista de grupos de instâncias  $\in$  ao cluster do representante em  $M$ 
3: while  $M$  não convergiu do
4:   for all  $t_i \in T$  do
5:      $j =$  index de  $KNN(t_i, M)$ 
6:     adicione  $t_i$  ao grupo  $G_j$ 
7:   end for
8:   atualize  $M$  com as médias de  $G$ 
9: end while
10: return os clusters e as médias  $G, M$ 
```

2.3 Classificadores

Todo sistema de reconhecimento de padrões utiliza um classificador para discriminar os padrões de teste. A eficiência de um classificador é medida pela taxa de acerto média, pela variância, e pelo seu custo computacional. Um classificador de aprendizagem baseada em instâncias muito utilizado é o *K-Nearest Neighbor*, KNN [Patrick and II (1969)].

2.3.1 K-Nearest Neighbor

O KNN é muito usado por ser um método de aprendizagem supervisionado simples, e por possuir uma taxa de acerto relativamente alta. O conceito básico consiste em: dado um padrão x a ser classificado e um conjunto de padrões conhecidos T ,

obter as classes dos K elementos de T mais próximos de x . A classe que obtiver maior ocorrência, ou peso, será a classe de x . Pode-se dizer que o KNN utiliza uma abordagem "*Diga-me com quem andas, e direi quem és.*". O algoritmo está descrito em Algorithm 2.

Algorithm 2 KNN

Require: K : um número

Require: T : conjunto de treinamento

Require: x : elemento para ser classificado

Require: L : uma lista

```

1: for all  $t_i \in T$  do
2:    $d_i = \text{distance}(t_i, x)$ 
3:   adicione  $(d_i, \text{Classe}(t_i))$  em  $L$ 
4: end for
5:  $\text{Ordene}(L)$  de acordo com as distâncias
6: obtenha os  $K$  primeiros elementos de  $L$ 
7: return a classe de maior ocorrência, ou peso, entre os  $K$ 

```

Pode-se dizer que o K – *NearestNeighbor* atua considerando a densidade das classes na região do padrão que se deseja classificar. Ou seja, o mesmo calcula diretamente a probabilidade a posteriori.

O KNN é uma técnica de estimativa de densidade não paramétrica, onde a densidade é calculada a partir da equação 2.1, onde K_n é um valor fixo que é função do número de exemplos n , V_n é uma variável aleatória que, no caso do KNN é a distância da amostra x para o k -ésimo vizinho mais próximo.

$$p(x) = \frac{k_n}{V_n} \quad (2.1)$$

A partir da estimativa de probabilidade a priori (proporção de classes no todo) e estimativa da densidade condicional, encontra-se a probabilidade a posteriori, intuitivamente representada pela regra de decisão do KNN:

$$P(\omega_i|x) = \frac{k_i}{K} \quad (2.2)$$

Onde k_i é a quantidade de instâncias da classe ω_i .

2.3.2 Janela de Parzen

Janela de Parzen é uma técnica de estimação de densidade não-paramétrica que se baseia na interpolação de dados. O estimador tenta interporlar os dados a partir de usos de janelas, como em um histograma, por exemplo, afim de estimar a densidade a partir da frequência das amostras encontradas em cada janela.

Esta estimativa é feita assumindo-se uma função $K(x)$ que determina a janela centrada em x com largura h , esta função é chamada de função de kernel.

De maneira simples, para estimar a densidade fixa-se uma região R e um volume fixo V , e k correspondente é determinado a partir dos dados de aprendizagem. Assumindo que a região R é um hipercubo de d dimensões, o seu volume é dado por h^d .

Para se obter a função de densidade de probabilidade de uma amostra x utilizando-se a Janela de Parzen, simplesmente centra-se a região R em x e conta-se o número de instâncias dentro desta região (valor de k) na equação 2.1.

Determinar o h ideal é uma tarefa minuciosa, pois, se o valor de h for muito pequeno, haverá muita especialização e estará mais sujeita a erros ocasionados por ruídos, já se h for muito grande, ocorrerá uma super-generalização. A escolha da função $K(x)$ apropriada também é um problema para esta técnica.

2.3.3 Estimação da Máxima Verossimilhança

Estimativa por máxima verossimilhança, *Maximum – Likelihood*, é um método para estimar os parâmetros de um modelo estatístico. A partir de um conjunto de dados e um dado modelo estatístico (i.e. Normal), o *MLE* estima valores adequados para os parâmetros do modelo.

De forma geral, dado um conjunto de instâncias de treinamento e um modelo estatístico, o *MLE* estima os valores dos parâmetros do modelo estatístico de forma que a probabilidade dos dados observados sejam maximizadas, conforme a equação 2.3.

$$\hat{\theta} = \arg \max_{\theta} \ln P(D|\theta) \quad (2.3)$$

Onde $\hat{\theta}$ é o valor de θ mais compatível com os exemplos observados, D é o conjunto de exemplo, e $P(D|\theta)$ é a verossimilhança de θ em relação ao conjunto de exemplos.

Para o caso de uma distribuição normal, o modelo é definido pelo vetor de médias e pela matrix de covariância. Neste caso, pode-se mostrar que as estimativas de máxima verossimilhança para os parâmetros são dados pelas equações 2.4 e 2.5.

$$\hat{\mu} = \frac{1}{n} \sum_{k=1}^n \mathbf{x}_k \quad (2.4)$$

$$\hat{\Sigma} = \frac{1}{n} \sum_{k=1}^n (\mathbf{x}_k - \hat{\mu})(\mathbf{x}_k - \hat{\mu})^t \quad (2.5)$$

O algoritmo *MLE* só é capaz de tratar distribuições unimodais, para distribuições multimodais, utiliza-se o algoritmo *EM*, abordado a seguir.

2.3.4 Algoritmo Expectation-Maximization

Expectation-Maximization é um algoritmo iterativo para se encontrar a máxima verossimilhança, os parâmetros de um modelo, onde o modelo é desconhecido e

depende de variáveis não observadas.

A iteração do *EM* alterna as etapas *E* (Expectation) e *M* (Maximization). A etapa *E* computa o valor esperado da log-verossimilhança avaliada, utilizando a estimativa atual para os parâmetros. Já a etapa *M* computa os parâmetros, maximizando a esperada log-verossimilhança encontrada na etapa *E*.

A diferença do *EM* para o *MLE* é que o *EM* é capaz de tratar distribuições multimodais, além de ser capaz de estimar os parâmetros mesmo com *missing values*.

2.4 Combinação de Classificadores

Combinar classificadores é a ação de utilizar dois ou mais classificadores para classificar um dado padrão. No caso, a ideia é obter o melhor de cada classificador. A combinação de classificadores deve ser feita utilizando classificadores que diferem, que classificam um mesmo padrão de forma divergente.

Por combinar "*o melhor*" dos classificadores, esta técnica é interessante para tratar de regiões de indecisão, driblando limitações dos classificadores utilizados, pois cada um fornece uma informação complementar na tarefa de classificar um padrão.

Existem diversas formas de combinar classificadores, algumas delas serão abordadas logo abaixo.

2.4.1 Regra da Soma

A regra da soma faz a combinação de classificadores somando as probabilidades a posteriori de cada classificador. Dado um padrão x , ele será classificado como define a equação 2.6.

$$(1 - L)p(\omega_j) \sum_{i=1}^L p(\omega_j | \mathbf{x}_i) > (1 - L)p(\omega_k) \sum_{i=1}^L p(\omega_k | \mathbf{x}_i), \quad (2.6)$$

$$k = 1, \dots, C; k \neq j$$

Sendo C o número de classes e L a quantidade de classificadores.

Para probabilidades a priori iguais, o padrão x será classificado como sendo da classe que obtiver maior soma de probabilidades a posteriori.

Esta regra é adequada para classificadores que possuem o mesmo sensor, pois o mesmo não é muito sensível a erros na estimação de densidade. Para se obter melhores resultados, pode-se atribuir diferentes pesos para as probabilidades a posteriori de cada classificador.

Além da regra da soma, existem outras possíveis regras abordadas brevemente nas subseções abaixo.

2.4.2 Regra do Produto

Uma forma de fazer a combinação de classificadores é utilizando a regra do produto. Esta regra, como o próprio nome sugere, nada mais é do que fazer a multiplicação das probabilidades *a posteriori* de cada classificador.

Dado um padrão x , ele será classificado como sendo da classe que obtiver maior valor da multiplicação entre as probabilidades *a posteriori* de um classificador.

Esta técnica é adequada quando as entradas dos classificadores são providas por sensores diferentes.

2.4.3 Regra do Máximo

A regra do máximo é uma aproximação da regra da soma. Nesta aproximação, a probabilidade de uma determinada classe é a maior probabilidade *a posteriori* dentre as saídas dos classificadores para aquela classe.

2.4.4 Regra do Mínimo

A regra do mínimo é uma aproximação da regra do produto. Nesta aproximação, a probabilidade de uma determinada classe é a menor das probabilidades *a posteriori* dentre as saídas dos classificadores para aquela classe.

2.4.5 Regra da Mediana

Esta regra é um caso particular onde é obtida as medianas das probabilidades *a posteriori* de cada classe. A classe que obtiver maior mediana é utilizada para classificar um padrão x de entrada.

Capítulo 3

Experimentos

3.1 Geração de Dados

Os dados gerados para serem analisados nos experimentos foram especificados como um conjunto de 300 padrões formando 3 conjuntos, um conjunto com 150 padrões, outro com 100 e a última com 50 padrões. Esses padrões são descritos por duas variáveis quantitativas, e todos gerados a partir de distribuições bi-variadas, conforme a tabela 3.1.

Conjunto	Classe	Instâncias	μ_1	μ_2	σ_1	σ_2	σ_{12}	ρ_{12}
Conjunto 1	1	150	0.00	0.00	2	1	1.7	0.85
Conjunto 2	2	100	0.00	3	0.5	0.5	0.00	0.00
Conjunto 3	2	50	4	3	2	1	-1.7	-0.85

Tabela 3.1: Dados para geração do conjunto de padrões

Para criar os conjuntos com os dados acima, foi utilizada a função do matlab $mvnrnd(\mu, \sigma, q)$, sendo q a quantidade de instâncias a serem geradas.

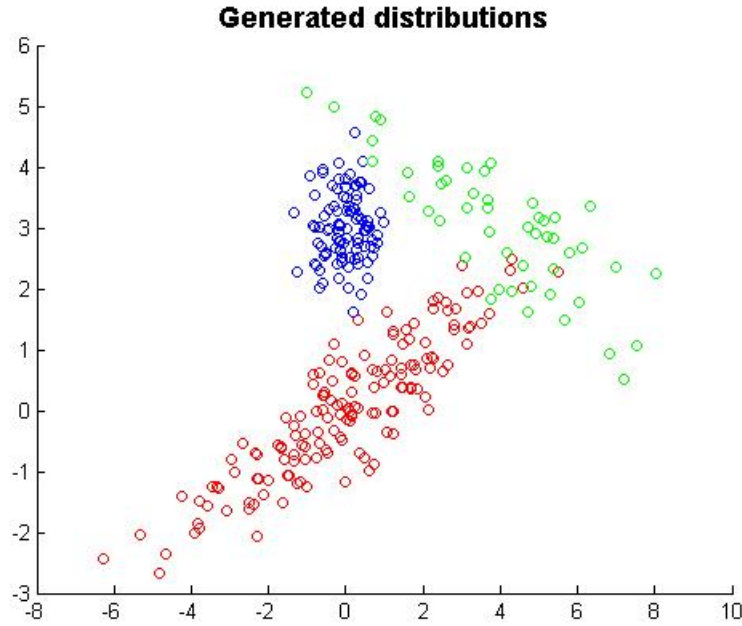


Figura 3.1: Dados gerados para os experimentos

A figura 3.1 mostra o resultado final da base de dados gerada. Os conjuntos 1, 2 e 3 estão representados pelas cores vermelha, azul e verde, respectivamente. No caso, a Classe 1 está representada pela cor vermelha, e a Classe 2, pelas demais cores. Estes dados serão o referencial para todos os experimentos que seguem.

3.2 Avaliação

Neste capítulo, serão descritos os experimentos realizados e os resultados obtidos serão avaliados.

Para avaliar o desempenho de cada classificador, foi calculada a taxa de erro de classificação global e por classes. Para o K-means, também foi calculado o Índice de Rand Corrigido.

Para avaliar os classificadores, utilizou-se *Holdout Cross-Validation* estratificado e repetido. A base dados foi dividida em treino e teste, utilizando a proporção 70% e 30%, respectivamente.

O experimento foi executado 50 vezes, para se obter métricas confiáveis de avaliação das técnicas, e para, posteriormente, utilizar o teste de hipótese.

$$P(\omega_i|x_k, \theta_i) = \frac{p(x_k|\omega_i, \theta_i) \times P(\omega_i)}{\sum_{j=1}^c p(x_k|\omega_j, \theta_j) \times P(\omega_j)} \quad (3.1)$$

$$j = \arg \max_i P(\omega_i|x_k, \theta_i) \quad (3.2)$$

Os classificadores se baseiam na estimativa da probabilidade a posteriori das classes, especificada na equação 3.1. Sendo a regra de decisão classificar um padrão como sendo da classe de maior probabilidade a posteriori, conforme equação 3.2.

Como as classes são igualmente proporcionais, a probabilidade a priori de cada classe é a mesma, 50%. Reduzindo o problema de classificação, em alguns classificadores.

3.2.1 K-Means

O K-Means, já abordado na seção 2.2.1, foi utilizado neste experimento com $K = 2$, o que significa que ele gera 2 clusters. O algoritmo foi executado 100 vezes e foi escolhido o resultado de maior adequação entre os clusters e seus representantes.

A figura 3.2 mostra graficamente os clusters formados pelo K-Means.

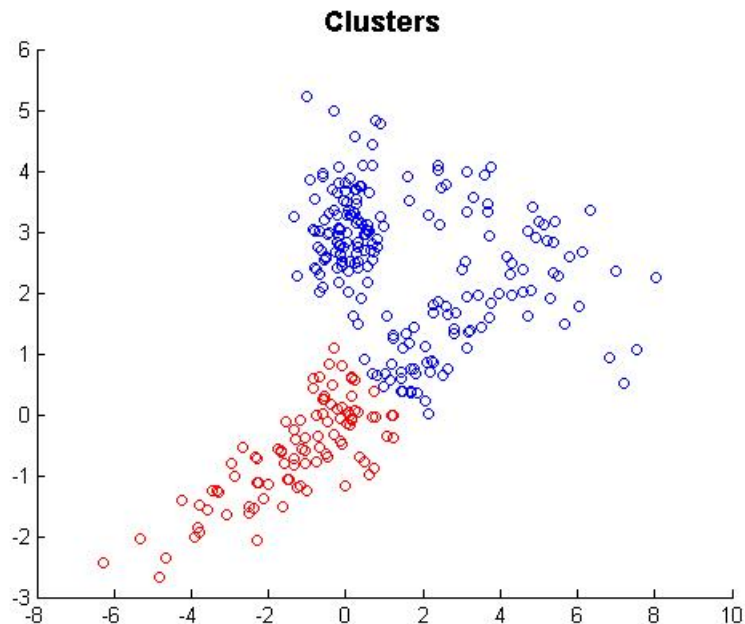


Figura 3.2: Clusters gerados pelo K-Means

Após a clusterização, é necessário associar cada cluster a uma classe. Para fazer isso, foi obtido o centróide de cada cluster e a média das amostras de cada classe. Com os centróides e as médias, o grupo com centróide mais próximo da média da Classe 1, foi associada a esta classe. O grupo com centróide mais distante da média da Classe 1, foi associado à Classe 2.

Os resultados obtidos estão representados na tabela 3.2.

Erro Global	Erro da Classe 1	Erro da Classe 2	Rand Corrigido
0.18	0.36	0.00	0.4079

Tabela 3.2: Tabela da taxa de erro do K-Means

Como existe sobreposição entre as distribuições, e a Classe 2 é multi-modal, o K-means encontra uma certa dificuldade em separar o conjunto em apenas 2 grupos. Pode-se observar também que, apesar de a taxa de erro global estar abaixo de 20%, o Index de Rand Corrigido esteve mais próximo de uma categorização randomica do que de uma categorização perfeita, estando na região intermediária, próxima de 50%.

3.2.2 Máxima Verossimilhança e Algoritmo EM

Nesta subseção, as estimativas de $P(\omega_i|x_k, \theta_i)$ foram feitas de formas diferentes para as duas classes. Para Classe 1, foi utilizado o Método da Máxima Verossimilhança, supondo uma normal multivariada. Já para a Classe 2 foi utilizado o algoritmo EM, supondo uma mistura de duas distribuições normais.

Para a Classe 1, obteve-se o vetor de média amostral e a matrix de covariância amostra, conforme as equações 2.4 e 2.5, respectivamente. Com estes parâmetros, foram calculados os valores das densidades para cada padrão de teste.

Já para a Classe 2, por se tratar de uma distribuição multimodal, foram calculados vetor de média amostral de cada compona e a matriz de covariância de cada componente. Isto foi feito por meio do *Expectation-Maximization*. A partir destes parâmetros, foi calculada os valores da função de densidade para os padrões de teste de cada componente.

Uma vez com as funções de densidade de cada componente, a função densidade resultante foi calculada como a soma das funções de densidade de probabilidade das componentes ponderada pelas respectivas probabilidades a priori. Estas probabilidades a priori foram encontradas observando-se as proporções dos padrões de teste encontradas em cada distribuição da mistura.

Por fim, visto que a probabilidade a priori das classes são iguais, a regra de classificação se reduz a atribuir um padrão à classe de maior densidade de probabilidade seja maior para o padrão em questão.

Os resultados estão descritos na tabela abaixo:

Erro Global	Erro da Classe 1	Erro da Classe 2
0.031 ± 0.015	0.051 ± 0.026	0.010 ± 0.013

Tabela 3.3: Tabela da taxa de erro do MLE e EM combinados

3.2.3 Janela de Parzen

Para estimar a função de densidade de probabilidade dos padrões de teste, foi considerada a Função de Kernel Gaussiana centrada em cada padrão de teste, com variância igual ao tamanho da janela h .

A forma utilizada para estimar a função de densidade, está expressa na equação 3.3.

$$\hat{p}(x) = \frac{1}{n} \frac{1}{h_1 \dots h_p} \sum_{i=1}^n \prod_{j=1}^p K_j \left(\frac{x_j - x_{ij}}{h_j} \right) \quad (3.3)$$

onde Kernel Gaussiano utilizado foi

$$K_j(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}, t = \left(\frac{x - x_i}{h_n} \right) \quad (3.4)$$

Conforme mencionado na seção 2.3.2 é muito importante escolher a largura da janela adequada. Foram considerados valores independentes para largura da janela em cada dimensão. Para tal, o algoritmo foi executado 50 vezes para cada combinação de larguras de janelas pertencentes ao intervalo [0.1, 0.2, 0.3, ..., 9.8, 9.9, 10] e compondo o gráfico de superfície da figura 3.3. Nesta figura o eixo vertical representa o erro global médio, e os demais eixos representam combinações de valores da Janela.

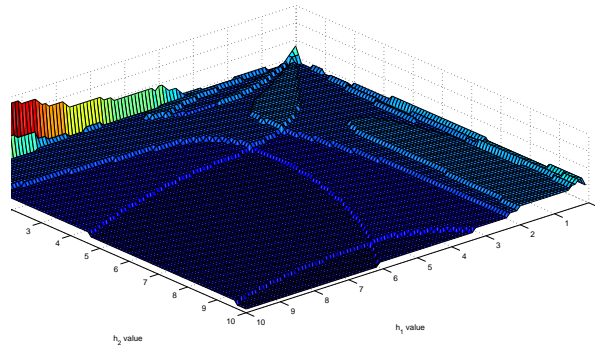


Figura 3.3: Gráfico de erro global por larguras de janela

Assim, escolheu-se um dos valores que estão na região de menor erro global, e os valores escolhidos foram 10 e 5 para o primeiro e o segundo atributo, respectivamente.

Novamente, como a probabilidade a priori é a mesma para ambas as classes, a decisão se resume a classificar o padrão como sendo da classe de maior densidade de probabilidade. tabela 3.4 mostra os erros obtidos por este classificador.

Erro Global	Erro da Classe 1	Erro da Classe 2
0.024 ± 0.013	0.032 ± 0.021	0.017 ± 0.020

Tabela 3.4: Tabela de erro da Janela de Parzen

3.2.4 KNN

Como visto na seção 2.3.1 a estimativa de probabilidade a posteriori se dá pela proporção de amostras de cada classe, K_i , dividido pelo número de vizinhos considerados, K , de acordo com a equação 2.2.

Assim como para o classificador baseado na Janela de Parzen, é preciso determinar o parâmetro K adequado. Após testes exaustivos, concluiu-se que o valor de K pouca varia a partir do $K = 3$, para partição de testes utilizada. Portanto, o valor de K escolhido para avaliação e comparação, foi $K = 3$.

O algoritmo foi executado 50 vezes, e segue a tabela 3.5 mostra os resultados obtidos.

Erro Global	Erro da Classe 1	Erro da Classe 2
0.030 ± 0.012	0.037 ± 0.024	0.024 ± 0.023

Tabela 3.5: Tabela de erro do KNN

3.2.5 Combinação de Classificadores

Para o classificador baseado em combinação, utilizou-se a regra da soma para estimativa da probabilidade a posteriori das classes. Dada a igualdade das probabilidades a priori das classes, a regra de decisão se reduz a atribuir um padrão de testes a classe ω_j se reduz a equação 3.5 (que é um caso particular da equação 2.6).

$$\sum_{i=1}^L p(\omega_j | \mathbf{x}_i) > \sum_{i=1}^L p(\omega_k | \mathbf{x}_i), \quad (3.5)$$

$$k = 1, \dots, C; k \neq j$$

Sendo C o número de classes e L a quantidade de classificadores.

Os classificadores combinados foram 3: o *MLE* e *EM*, o de Janela de Parzen e o KNN.

Assim, os resultados obtidos estão contidos na tabela 3.6.

Erro Global	Erro da Classe 1	Erro da Classe 2
0.024 ± 0.014	0.041 ± 0.025	0.007 ± 0.010

Tabela 3.6: Tabela de erro dos Classificadores Combinados

3.3 Comparação

Capítulo 4

Conclusão

ADICIONAR CONCLUSÃO.

Referências Bibliográficas

Patrick, E. A. and II, F. P. F. (1969) A generalization of the k-nearest neighbor rule.
In *IJCAI*, pp. 63–64.