Universidade Federal de Pernambuco
Centro de Informática
Programa de Pós-Graduação em Ciências da Computação

Aprendizagem de Máquina 2018.1

Relatório do Projeto

Prof. Dr. Francisco de A. T. de Carvalho

Alunos
Jefferson Lima
João Antônio
Évora Leite

Recife - PE
2018

Sumário

1. Introdução

a. KCM-K-GH

Através de uma função objetivo, o algoritmo KCM-K-H, do qual o KCM-K-GH é uma variante, disponibiliza iterativamente em três passos uma partição de clusters, hiperparâmetros para cada uma das variáveis e uma matriz de protótipos dos clusters.

O KCM-K-GH recebe o número de clusters *c* como parâmetro, a partir disso é gerado aleatoriamente um centróide para representar cada cluster. Para a inicialização do vetor de hiperparâmetros s^2, são calculadas as distâncias euclidianas entre os atributos da base de dados, esse vetor de distâncias é ordenado, e por fim, é tirado a média entre os 0.1 e 0.9 elementos. O parâmetro γ é calculado com a equação abaixo, onde *p* é igual ao número de características da base de dados.

$$\gamma = \prod_{j=1}^{p} \frac{1}{s_j^2}$$

Depois da inicialização dos parâmetros as amostras da base de dados são alocadas em cada cluster $P_i (1 \leq i \leq c)$ se:

$$2(1 - K^{(s)}(\mathbf{x_k}, \mathbf{g_i})) = min_{h=1}^{c} 2(1 - K^{(s)}(\mathbf{x_k}, \mathbf{g_h})),$$

$$K^{(\mathbf{s})}(\mathbf{x_i}, \mathbf{x_k}) = exp\left\{ -\frac{1}{2} \sum_{j=1}^{p} \frac{1}{s_j^2} (x_{ij} - x_{kj})^2 \right\}$$

A partir desse ponto o algoritmo entra em loop, atualizando os centróides de acordo com os elementos de cada grupo, recalculando o vetor de hiperparâmetros s^2 e recalculando em qual grupo vai ficar cada amostra. Os centróides são recalculados a partir da equação abaixo.

$$\mathbf{g_i} = \frac{\sum_{e_k \in P_i} K^{(\mathbf{s})}(\mathbf{x_k}, \boldsymbol{g_i})\mathbf{x_k}}{\sum_{e_k \in P_i} K^{(\mathbf{s})}(\mathbf{x_k}, \boldsymbol{g_i})}, 1 \leqslant i \leq c$$

Os hiperparâmetros são atualizados a partir da equação.

$$\frac{1}{s_j^2} = \frac{\gamma^{\frac{1}{p}} \left\{ \prod_{h=1}^{p} \left[\sum_{i=1}^{c} \sum_{\boldsymbol{e_k} \in P_i} K^{(\mathbf{s})}(\mathbf{x_k}, \mathbf{g_i})(x_{kh} - g_{ih})^2 \right] \right\}^{\frac{1}{p}}}{\sum_{i=1}^{c} \sum_{\boldsymbol{e_k} \in P_i} K^{(\mathbf{s})}(\mathbf{x_k}, \mathbf{g_i})(x_{kj} - g_{ij})^2}$$

As amostras vão ser alocadas para os clusters utilizando a equação que já tínhamos apresentado:

$$2(1 - K^{(s)}(\mathbf{x_k}, \mathbf{g_i})) = min_{h=1}^{c} 2(1 - K^{(s)}(\mathbf{x_k}, \mathbf{g_h}))$$

e o processo vai ser repetido até que as amostras não se movam mais de um cluster para outro. Por fim, é escolhida a configuração de centróides, hiperparâmetros e as partições que resultarem no menor valor da função objetivo *JKCM-K-GH*.

$$JKCM{-}K{-}GH = \sum_{i=1}^{p} \sum_{e_k \in P_i} 2(1{-}K^{(\mathbf{s})}(\mathbf{x_k}, \mathbf{g_i}))$$

Como saída do algoritmo temos os centróides de cada grupo, o número de objetos por grupo e o vetor de hiperparâmetros.

b. Naive Bayes Gaussiano

É um modelo de classificação que tem como base o Teorema de Bayes. Esse modelo se utiliza das probabilidades condicionais entre eventos relacionados com um conhecimento a priori.

A regra de decisão bayesiana é dada pela fórmula:

$$P(w_i|\mathbf{x}_k) = max_{i=1}^{7} P(w_i|\mathbf{x}_k),$$

onde

$$P(w_i|\mathbf{x}_k) = \frac{p(\mathbf{x}_k|w_i)P(w_i)}{\sum_{r=1}^{c} p(\mathbf{x}_k|w_r)P(w_r)}$$

$$posteriori = \frac{verossimilhana \times priori}{evidencia}$$

Para cada classe, é feito a estimação da verossimilhança pelo método da Máxima Verossimilhança supondo uma distribuição Normal Multivariada com parâmetro $\boldsymbol{\theta}$ **:**

$$p(\mathbf{x}_k|w_i) = p(\mathbf{x}_k|w_i, \theta_i)$$

onde,

$$\theta_i = \begin{pmatrix} \mu_i \\ \boldsymbol{\Sigma} \end{pmatrix}$$

Na realização dos experimentos utilizando o Naive Bayes Gaussiano, foi suposto independência entre as variáveis da base de dados. Dessa forma, somente as variâncias foram utilizadas, as covariâncias foram desconsideradas. Logo, temos que

$$\boldsymbol{\Sigma} = \boldsymbol{\sigma}^2 \times \boldsymbol{I}$$

Além disso, como algumas variâncias eram nulas, para encontrar a inversa da matriz de covariância, foi necessário somar uma constante a sua diagonal principal.

c. Janela de Parzen

É um método não paramétrico que tem como objetivo estimar a função de densidade de probabilidade de um conjunto de dados *p(**x**)*, ou dada uma classe *p(**x** | ω_i)* (Verossimilhança). A ideia base dessa abordagem é contar quantas amostras estão dentro de uma região específica ***R**n* ou “janela”, no caso. A probabilidade de uma região cair nessa região é: *p(**x**) = total de amostras em **R** / total de amostras*.

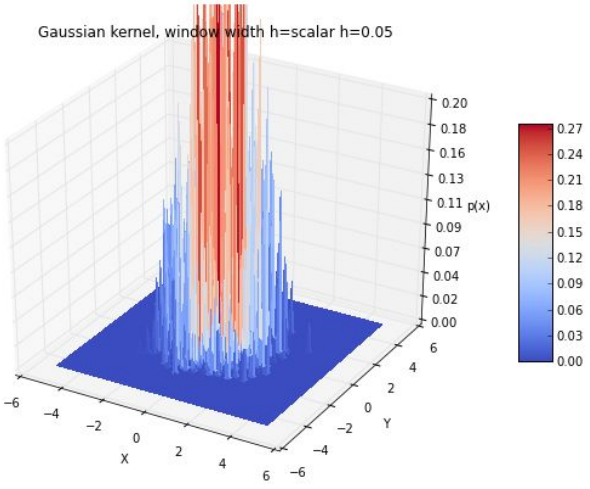
Fixamos o tamanho da região **R** para estimar a densidade, fixamos o volume V e determinamos o correspondente k a partir dos dados de aprendizagem e

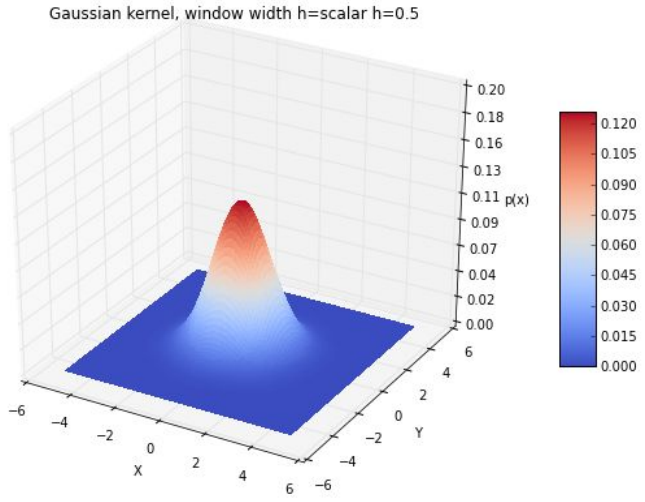
assumimos que a região **R** é um hipercubo de tamanho *h* cujo volume é h^d. Considerando que essa região tem a forma gaussiana, e que **x** é uma amostra *p*-dimensional, temos que a ***p(x)*** é dado por:

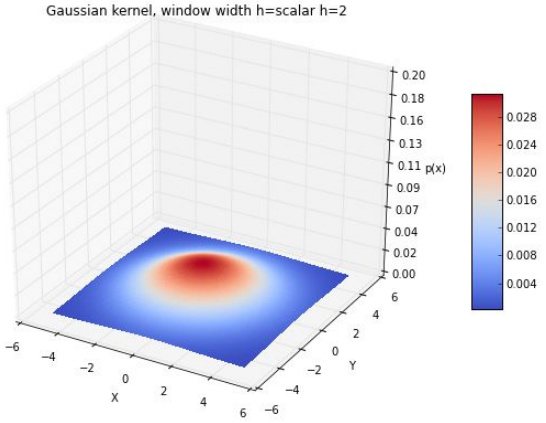
$$p(\mathbf{x}) = \frac{1}{n}\frac{1}{h_1 \ldots h_p}\sum_{i=1}^{n}\prod_{j=1}^{p} K_i$$

$$K_i = \frac{1}{\sqrt{2\pi}}\exp^{-\frac{x_j - X_{ij}}{2h}}$$

Uma boa escolha do valor de *h* influencia bastante no nível suavidade e de generalização do modelo. No caso do gaussiano, vemos a diferença da escolha de valores para *h* no kernel gaussiano.







d. Combinação de classificadores

Combinação de múltiplos sistemas com o objetivo de obter uma melhor performance do que seria obtida por um classificador simples. A combinação de classificadores pode ser para minimizar as limitações de algumas técnicas, cada uma podendo fornecer informações complementares na tarefa de classificação. Porém, não há garantias que a combinação produza resultados adequados. Da mesma forma, classificadores combinados podem ser mais difíceis de analisar.

No presente trabalho, a combinação dos classificadores (Bayes e Parzen) é feita através do método da soma, que consiste basicamente na seguinte regra de decisão:

$$max_{r=1}^{7}((1-L)P(w_r)) + P_1(w_i|\mathbf{x}_k) + P_2(w_i|\mathbf{x}_k) + ... + P_n(w_i|\mathbf{x}_k)$$

2. Metodologia

Os algoritmos utilizados neste projeto foram todos implementados na linguagem python, utilizando algumas das bibliotecas mais utilizadas para tarefas de aprendizagem de máquina e análise de dados da atualidade, como, numpy, scipy, sklearn entre algumas outras.

A base de dados utilizada neste projeto foi a *Image Segmentation* do site da *uci machine learning repository* que pode ser encontrada na url (http://archive.ics.uci.edu/ml/machine-learning-databases/image). Esta base de dados é disponibilizada em dois arquivos, o *segmentation.data* e o *segmentatio.test*, porém neste trabalho foi utilizado apenas o arquivo *segmentation.test* que contém 2100 amostras igualmente divididas entre 7 classes distintas: *brickface, sky, foliage, cement, window, path* e *grass*. Cada amostra é descrita por 19 atributos: r*egion-centroid-col, region-centroid-row, region-pixel-count, short-line-density-5, short-line-density-2, vedge-mean, vegde-sd, hedge-mean, hedge-sd, intensity-mean, rawred-mean, rawblue-mean, rawgreen-mean, exred-mean, exblue-mean, exgreen-mean, value-mean, saturatoin-mean* e *hue-mean*. Ainda, podemos dividir as amostras em 3 outros grupos, o *shape view,* o *rgb view* e o *complete view*. O *shape view* é o grupo das amostras descritos pelas 9 primeiras características: *region-centroid-col, region-centroid-row, region-pixel-count, short-line-density-5, short-line-density-2, vedge-mean, vegde-sd, hedge-mean e hedge-sd*. O *rgb view* é o grupo das amostras descrita pelos 10 últimos atributos: i*ntensity-mean, rawred-mean, rawblue-mean, rawgreen-mean, exred-mean, exblue-mean, exgreen-mean, value-mean, saturation-mean* e *hue-mean*. E por fim, o *complete view* é o grupo descrito por todas as características.

Analisando os atributos da base de dados, percebemos que o terceiro atributo, o *region-pixel-count*, não tem variância nenhuma na base de dados, desta forma, o atributo foi removido para não prejudicar os algoritmos. Outra abordagem para tentar melhorar o desempenho dos algoritmos foi a padronização da base de dados, que transformou cada um dos atributos para uma escala de 0 a 1. A padronização geralmente é feita para diminuir os efeitos trazidos pela diferença de

escala dos atributos, que pode acabar fazendo com que os algoritmos deem um peso maior para determinados atributos que tenham uma escala maior. A padronização dos atributos na base de dados foi dada pela equação abaixo, onde $X_{i,j}$ é o valor da base de dados para o j-ésimo atributo da i-ésima amostra.

$$\mathbf{X_{ij}} = \frac{\mathbf{X_{ij}} - min(\mathbf{X_j})}{max(\mathbf{X_j}) - min(\mathbf{X_j})}$$

3. Experimentos e Resultados

Para avaliação dos algoritmos utilizados neste trabalho foram feitos experimentos com validação cruzada dos dados. Para o algoritmo não supervisionado *KCM-K-GH*, foi utilizado um experimento de 100 *holdouts validation* onde treinamos o algoritmo 100 vezes e salvamos a configuração que nos deu o melhor resultado de acordo com a função objetivo.

Para avaliação do Naive Bayes e da Janela de Parzen, foi utilizado o 10 *fold validation* repetido 30 vezes para pegar a média de acerto dos 10 *folds* em cada repetição. Uma observação importante sobre a avaliação da Janela de Parzen consiste na utilização de um conjunto de validação para estimação do melhor valor de *h*, variando o valor de *h* em: {1, 2, 3, 4, 5, 6, 7, 8, 9 e 10} quando era considerada a base de dados original, sem nenhuma normalização; E {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 e 1} quando era considerada a base normalizada, na qual as características estavam situadas em de 0 à 1. Como métrica foi utilizado a taxa de acerto média, e o desvio padrão médio. Além disso, foi calculado um intervalo com 95% de confiança (IC) para a taxa de acerto média. Cada tabela contém o desempenho individual dos três classificadores nas três views (*Complete, RGB* e *Shape*), considerando a base de dados original e normalizada.

KCM-K-GH

Nas tabelas abaixo, temos as informações de quantas amostras ficaram em cada grupo e do ARI (os demais dados estão nos arquivos em anexo). Foram realizados experimentos com a base de dados padronizada e sem a utilização da padronização.

Na base de dados Complete View temos que o melhor índice rand corrigido (ARI) foi de 0.31, que foi obtido utilizando a base de dados padronizada. Por outro lado, sem utilizar a padronização o melhor ARI obtido para essa base de dados foi de 0.11.

Base de dados: Complete View

Partição	Número de elementos	
	Padronizado	**Não Padronizado**
1	16	509
2	561	2
3	299	4
4	66	6
5	6	71
6	74	3
7	1138	1505
ARI	**0.3116992969**	**0.1171395422**

Para base de dados Shape View o resultado da clusterização foi um pouco inferior ao resultado obtido na base de dados Complete View. Aqui temos que o melhor ARI conseguido foi de 0.14 utilizando a base de dados padronizada e 0.05 utilizando a base de dados sem padronização.

Base de dados: Shape View

Partição	Número de elementos	
	Padronizado	**Não Padronizado**
1	10	10
2	1203	1
3	3	666
4	2	1
5	10	1
6	620	2
7	243	1410
ARI	**0.147981892**	**0.0599238581**

A base de dados RGB View foi a que deu um resultado mais homogêneo, onde as amostras foram melhor distribuídas entre as partições. Também foi nessa base de dados que conseguimos o melhor resultado de ARI, 0.52 para a base de dados normalizada e 0.51 para a base de dados não normalizada.

Base de dados: RGB View

Partição	**Número de elementos**	
	Padronizado	**Não Padronizado**
1	259	399
2	297	176
3	300	404
4	368	284
5	196	299
6	383	241
7	297	297
ARI	**0.522503057**	**0.5134576609**

Naive Bayes Gaussiano

Os resultados da tabela abaixo consistem na taxa de acerto médio nos 10 *folds*, para cada view em cada repetição.

Repetição	Complete	RGB	Shape
1	0,792	0,818	0,482
2	0,794	0,814	0,498
3	0,782	0,792	0,508
4	0,780	0,812	0,471
5	0,796	0,796	0,500
6	0,776	0,788	0,494
7	0,786	0,771	0,465
8	0,802	0,796	0,457
9	0,804	0,780	0,471
10	0,796	0,788	0,447
11	0,796	0,816	0,484
12	0,782	0,792	0,459
13	0,769	0,806	0,465
14	0,794	0,788	0,469
15	0,776	0,800	0,510
16	0,802	0,794	0,467
17	0,794	0,782	0,457
18	0,790	0,784	0,482
19	0,794	0,788	0,498
20	0,786	0,784	0,490
21	0,800	0,776	0,510
22	0,812	0,812	0,484
23	0,810	0,792	0,480
24	0,784	0,790	0,465
25	0,808	0,769	0,508
26	0,804	0,771	0,486
27	0,794	0,784	0,482
28	0,786	0,798	0,492
29	0,800	0,794	0,459
30	0,792	0,786	0,480
Média	**0,794**	**0,791**	**0,482**

Std Dev	**0,011**	**0,013**	**0,018**
IC (0.05)	**0,790. 0,798**	**0,786;0,796**	**0,475;0,488**

Naive Bayes Gaussiano (Base Normalizada [0;1])

Os resultados da tabela abaixo consistem na taxa de acerto médio nos 10 *folds*, para cada view em cada repetição.

Repetição	Complete	RGB	Shape
1	0,782	0,820	0,631
2	0,788	0,816	0,641
3	0,794	0,812	0,604
4	0,812	0,808	0,629
5	0,792	0,798	0,639
6	0,796	0,802	0,629
7	0,780	0,804	0,622
8	0,790	0,814	0,653
9	0,812	0,806	0,665
10	0,796	0,806	0,635
11	0,800	0,796	0,639
12	0,798	0,788	0,631
13	0,782	0,800	0,629
14	0,786	0,816	0,637
15	0,792	0,820	0,620
16	0,810	0,800	0,649
17	0,798	0,810	0,663
18	0,814	0,812	0,653
19	0,808	0,798	0,629
20	0,786	0,816	0,645
21	0,802	0,822	0,641
22	0,784	0,778	0,657
23	0,806	0,810	0,637
24	0,790	0,792	0,649
25	0,804	0,816	0,645
26	0,786	0,822	0,639
27	0,818	0,806	0,651
28	0,771	0,804	0,633
29	0,802	0,810	0,647

30	0,812	0,782	0,643
Média	**0,796**	**0,807**	**0,639**
Std Dev	**0,012**	**0,012**	**0,013**
IC (0.05)	**0,792;0,800**	**0,803;0,811**	**0,634;0,643**

Janela de Parzen

Os resultados da tabela abaixo consistem na taxa de acerto médio nos 10 *folds*, para cada view em cada repetição.

Repetição	Complete	RGB	Shape
1	0,947	0,896	0,645
2	0,955	0,902	0,618
3	0,953	0,904	0,653
4	0,941	0,914	0,600
5	0,941	0,898	0,622
6	0,941	0,892	0,633
7	0,945	0,902	0,620
8	0,945	0,900	0,633
9	0,937	0,904	0,635
10	0,943	0,898	0,647
11	0,939	0,898	0,649
12	0,951	0,890	0,639
13	0,945	0,914	0,647
14	0,943	0,914	0,659
15	0,935	0,904	0,614
16	0,943	0,896	0,647
17	0,955	0,888	0,608
18	0,937	0,900	0,618
19	0,949	0,892	0,622
20	0,947	0,910	0,641
21	0,943	0,896	0,616
22	0,949	0,892	0,627
23	0,943	0,900	0,624
24	0,947	0,918	0,641
25	0,949	0,902	0,629
26	0,949	0,890	0,647
27	0,935	0,904	0,627

28	0,941	0,890	0,618
29	0,937	0,916	0,637
30	0,953	0,904	0,633
Média	**0,944**	**0,900**	**0,633**
Std Dev	**0,006**	**0,009**	**0,014**
IC (0.05)	**0,94;0.946**	**0,897; 0,903**	**0,628;0,638**

Janela de Parzen (Base Normalizada [0;1])

Os resultados da tabela abaixo consistem na taxa de acerto médio nos 10 *folds*, para cada view em cada repetição.

Repetição	Complete	RGB	Shape
1	0,808	0,694	0,645
2	0,810	0,694	0,647
3	0,808	0,690	0,643
4	0,812	0,694	0,655
5	0,810	0,694	0,639
6	0,812	0,692	0,645
7	0,810	0,694	0,643
8	0,822	0,694	0,645
9	0,814	0,692	0,645
10	0,822	0,692	0,647
11	0,818	0,694	0,649
12	0,804	0,694	0,647
13	0,806	0,694	0,645
14	0,798	0,692	0,635
15	0,820	0,692	0,651
16	0,812	0,694	0,645
17	0,816	0,694	0,639
18	0,812	0,696	0,645
19	0,812	0,692	0,647
20	0,810	0,690	0,647
21	0,808	0,692	0,651
22	0,804	0,692	0,643
23	0,806	0,694	0,637
24	0,814	0,694	0,645
25	0,818	0,692	0,643

26	0,810	0,692	0,641
27	0,808	0,694	0,641
28	0,802	0,690	0,647
29	0,818	0,694	0,647
30	0,812	0,692	0,641
Média	**0,811**	**0,694**	**0,645**
Std Dev	**0,006**	**0,001**	**0,004**
IC (0.05)	**0,809;0,813**	**0,693;0,694**	**0,643;0,646**

Método da Soma

Os resultados da tabela abaixo consistem na taxa de acerto médio nos 10 *folds*, para cada view em cada repetição.

Repetição	Complete	RGB	Shape	Todas as Views
1	0,900	0,869	0,559	0,951
2	0,918	0,900	0,606	0,957
3	0,910	0,902	0,594	0,965
4	0,914	0,914	0,549	0,955
5	0,914	0,890	0,600	0,953
6	0,902	0,894	0,578	0,961
7	0,900	0,886	0,561	0,961
8	0,941	0,902	0,557	0,937
9	0,904	0,892	0,584	0,959
10	0,927	0,902	0,549	0,959
11	0,937	0,890	0,588	0,949
12	0,918	0,886	0,559	0,965
13	0,900	0,910	0,573	0,953
14	0,929	0,892	0,573	0,963
15	0,896	0,902	0,602	0,957
16	0,931	0,884	0,580	0,961
17	0,931	0,867	0,565	0,955
18	0,904	0,855	0,573	0,953
19	0,916	0,878	0,569	0,957
20	0,908	0,898	0,588	0,953
21	0,900	0,878	0,592	0,963
22	0,943	0,900	0,563	0,963
23	0,892	0,910	0,580	0,961

24	0,931	0,900	0,571	0,959
25	0,939	0,906	0,578	0,969
26	0,914	0,896	0,571	0,963
27	0,908	0,880	0,582	0,941
28	0,918	0,898	0,592	0,939
29	0,941	0,892	0,588	0,951
30	0,910	0,902	0,594	0,965
Média	**0,914**	**0,895**	**0,578**	**0,958**
Std Dev	**0,015**	**0,014**	**0,015**	**0,008**
IC (0.05)	**0,909;0,920**	**0,890;0,9**	**0,572;0,583**	**0,955;0,961**

Método da Soma (Base Normalizada [0;1])

Os resultados da tabela abaixo consistem na taxa de acerto médio nos 10 *folds*, para cada view em cada repetição.

Repetição	Complete	RGB	Shape	Todas as Views
1	0,782	0,820	0,631	0,812
2	0,788	0,818	0,643	0,831
3	0,794	0,812	0,606	0,822
4	0,812	0,806	0,629	0,849
5	0,792	0,798	0,639	0,835
6	0,800	0,806	0,629	0,829
7	0,780	0,804	0,622	0,827
8	0,790	0,816	0,653	0,831
9	0,812	0,806	0,665	0,869
10	0,800	0,808	0,635	0,831
11	0,800	0,794	0,641	0,831
12	0,798	0,790	0,631	0,818
13	0,782	0,800	0,629	0,829
14	0,786	0,824	0,637	0,837
15	0,792	0,820	0,622	0,824
16	0,810	0,800	0,649	0,839
17	0,796	0,814	0,663	0,833
18	0,814	0,812	0,653	0,839
19	0,808	0,798	0,631	0,845
20	0,786	0,816	0,645	0,816
21	0,802	0,822	0,641	0,849
22	0,784	0,786	0,657	0,845

23	0,806	0,808	0,637	0,829
24	0,790	0,790	0,653	0,822
25	0,804	0,818	0,645	0,824
26	0,788	0,822	0,639	0,822
27	0,820	0,808	0,651	0,851
28	0,771	0,804	0,633	0,820
29	0,802	0,814	0,649	0,839
30	0,794	0,812	0,606	0,822
Média	0,795	0,808	0,639	0,831
Std Dev	0,012	0,011	0,014	0,012
IC (0.05)	0,791;0,799	0,804;0,812	0,634;0,644	0,826;0,835

Sumarização dos Resultados

Primeiramente, consideramos a base de dados original (houve apenas a remoção da coluna 2). Vemos que considerando a *Complete View*, o classificador baseado na janela de Parzen, com escolha automática do *h* através de validação, obteve o melhor desempenho, seguido da combinação através do método da soma e pelo Naive Bayes Gaussiano (*GNB*). A mesma ordem de desempenho foi observada nas views RGB e Shape. É interessante observar que mesmo o classificador da soma tendo as informações das posterioris do GNB e do Parzen, o resultado não foi superior. Exceto quando houve uma combinação das posterioris do *GNB* e do Parzen nas três views, pois houve um aumento na acurácia média considerando o melhor desempenho obtido até então pelo Parzen no *Complet View*.

Abaixo seguem os resultados do intervalo de 95% de confiança para a média da taxa de acerto de cada um dos experimentos em cada *view* na base de dados original.

Base Original				
	Complete View	**RGB View**	**Shape View**	**Todas as Views**
GNB	0,790 ; 0,798	0,786 ; 0,796	0,475 ; 0,488	
Parzen	**0,94 ; 0,946**	**0,897 ; 0,903**	**0,628 ; 0,638**	
Soma	0,909 ; 0,920	0,890 ; 0,9	0,572 ; 0,583	**0,955 ; 0,961**

Os resultados são diferentes quando aplicamos uma normalização na base de dados, limitando o *range* das características no intervalo [0;1]. Podemos observar que o classificador baseado na janela de Parzen obteve um melhor desempenho na *Complet View*, porém os classificadores através da Soma e o GNB obtiveram uma melhor acurácia que o Parzen na *RGB View*. Já na *Shape View* os resultados são praticamente os mesmos. Podemos ver o poder da combinação de classificadores através da soma, observando o ganho de acurácia na Soma considerando as posterioris do GNB e do Parzen nas três *views*.

Abaixo seguem os resultados do intervalo de 95% de confiança para a média da taxa de acerto de cada um dos experimentos em cada *view* na base de dados normalizada.

Base Normalizada [0;1])				
	Complete View	**RGB View**	**Shape View**	**Todas as Views**
GNB	0,792 ; 0,800	0,803 ; 0,811	0,634 ; 0,643	
Parzen	**0,809 ; 0,813**	0,693 ; 0,694	**0,643 ; 0,646**	
Soma	0,791 ; 0,799	**0,804 ; 0,812**	0,634 ; 0,644	**0,826 ; 0,835**

Análise Estatística

A análise estatística realizada neste trabalho, visa observar se há uma diferença na acurácia média de cada classificador supervisionado aqui estudado: Naive Bayes Gaussiano, Parzen e Soma. Para isso, foi feito o uso do teste de Friedman, que dado três ou mais conjuntos de médias o teste informa com um nível de confiança de (1-alpha)% se todas as médias são iguais, ou se existe pelo menos uma média diferente. Isso ocorre através do teste das seguintes hipóteses:

- H0: Todas as médias são iguais;
- Ha: Existe pelo menos uma média diferente;

Após a aplicação do teste de Friedman, foi aplicado o teste de Nemenyi para saber se, considerando que H0 foi rejeitada, qual é a diferença que existe entre essas médias e quão diferem entre si.

No nosso caso, o teste de Friedman foi aplicado considerando um nível de significância de 5%. A entrada dos testes de Friedman e Nemenyi foram três para cada *View*: GNB, Parzen e Soma (com a combinação de GNB e Parzen na respectiva *view*). Foram consideradas as seguintes hipóteses:

- H0: As médias do GNB, Parzen e Soma são iguais;
- Ha: Pelo menos uma média é diferente;

A saída do teste de Nemenyi consiste nos ranks dos postos de cada classificador correspondente aos conjuntos de médias como entradas. Dessa forma, foi possível observar o quão cada média difere das demais.

Podemos observar que somente em um caso Ha foi rejeitada, ou seja, não houve uma diferença significativa considerando um nível de significância de 5%: O caso onde foi considerado as médias de acerto dos classificadores na *Shape View* normalizada no range [0;1].

Complete View		
	Original	**Normalizada**
Teste de Friedman alpha = 0.05	Rejeita-se H0 Pelo menos uma diferente	Rejeita-se H0 Pelo menos uma diferente
Teste de Nemenyi Rank médio	**[GNB Parzen Soma]** [1. 2.96 2.03]	**[GNB, Parzen, Soma]** [1.53 2.766 1.7]
RGB View		
	Original	**Normalizada**
Teste de Friedman alpha = 0.05	Rejeita-se H0 Pelo menos uma diferente	Rejeita-se H0 Pelo menos uma diferente
Teste de Nemenyi Rank médio	**[GNB, Parzen, Soma]** [1. 2.733 2.2666]	**[GNB, Parzen, Soma]** [2.33 1. 2.667]
Shape View		
	Original	**Normalizada**
Teste de Friedman alpha = 0.05	Rejeita-se H0 Pelo menos uma diferente	Rejeita-se Ha Médias iguais
Teste de Nemenyi Rank médio	**[GNB, Parzen, Soma]** [1. 3. 2.]	**[GNB, Parzen, Soma]** [1.7 2.25 2.017]

4. Conclusão

Durante a execução dos experimentos foi constatado o algoritmo KCM-K-GH é muito sensível a inicialização dos seus centróides, e, como neste trabalho iniciamos os centróides de forma aleatória, é normal que haja alguma diferença entre o valores do ARI calculado nos experimentos aqui realizados e os valores contidos no artigo do qual este trabalho foi baseado. Ainda, a partir da variância percebida nas execuções dos experimentos aqui realizados, acreditamos que repetir o processo de convergência do algoritmo apenas 100 vezes não é suficiente para se chegar a um resultado sólido quanto ao poder do algoritmo. Sendo assim, se houvesse mais tempo, ao invés de 100 repetições poderíamos fazer 1000 ou mais por exemplo, assim talvez se chegasse a um resultado com menos variância entre uma execução e outra.

Com relação à etapa supervisionada do trabalho, pode-se observar algumas coisas. Primeiramente, o tratamento da base de dados é fundamental para um bom desempenho do classificador, sendo importante a realização de remoção de variáveis correlacionadas ou constantes. Também foi visto a importância do valor de *h* na janela de Parzen, reforçando a preocupação de selecionar o melhor *h* possível e a possibilidade de utilização de validação cruzada (com conjunto de validação) para essa tarefa. Pode-se observar que a janela de Parzen é uma boa forma de estimar a verossimilhança. Sobre o método da soma, pode-se concluir que realmente pode melhorar o desempenho geral considerando a saída de *n* classificadores. Porém, não é garantia que essa combinação sempre melhore o resultado, pode ser que um classificador diminua a eficácia de outro utilizado na combinação.