

Projeto Aprendizagem de Máquina

Estela de Andrade Joffily (eaj2)

Gabriel Galdino Gadelha (ggg)

Gabriel Miranda de Souza (gms5)

João Pedro Henrique Santos Duarte (jphsd)

Sergio de Vasconcelos Filho (svf)

Descrição dos *datasets*

- Composto por *features* de numerais manuscritos (0 a 9) extraídos de uma coleção de mapas holandeses.
- 10 classes, cada uma com 200 exemplos em diferentes representações.
- Para este projeto, foram utilizados três representações distintas: FAC, FOU e ZER.
- FAC: 216 correlações de perfil;
- FOU: 76 Coeficientes de Fourier das formas dos caracteres;
- ZER: 47 momentos Zernike;

Questão 01

Implementação do KFCM-K-W.2

KFCM-K-W.2

- Gaussian kernel fuzzy c-means with width parameter computation and regularization.
- Diferente de outros algoritmos kernel fuzzy c-means (KFCM), o proposto possui um parâmetro de largura particular para cada cluster e cada atributo.
- Tais parâmetros são automaticamente calculados através da regularização da entropia total.

Passo a passo

- Inicialização

- Vetores locais dos parâmetros de largura $\mathbf{s}_1, \dots, \mathbf{s}_c$, fazendo $\frac{1}{s_{ij}^2} \leftarrow 1, (1 \leq i \leq c; 1 \leq j \leq p)$.
- Seleção aleatória dos protótipos de cada cluster \mathbf{g}_i .
- Cálculo do grau de pertencimento usando:

$$u_{ki} = \left[\sum_{h=1}^c \left(\frac{2 - 2 \mathcal{K}^{(\mathbf{s}_i)}(\mathbf{x}_k, \mathbf{g}_i)}{2 - 2 \mathcal{K}^{(\mathbf{s}_i)}(\mathbf{x}_k, \mathbf{g}_h)} \right)^{\frac{1}{m-1}} \right]^{-1} \quad \mathcal{K}^{(\mathbf{s}_i)}(\mathbf{x}_l, \mathbf{x}_k) = \exp \left\{ -\frac{1}{2} \sum_{j=1}^p \frac{1}{s_{ij}^2} (x_{lj} - x_{kj})^2 \right\}$$

- Cálculo da função objetivo J usando:

$$J_{KFCM-K-W.2} = \sum_{i=1}^c \sum_{k=1}^n (u_{ki})^m (2 - 2 \mathcal{K}^{(\mathbf{s}_i)}(\mathbf{x}_k, \mathbf{g}_i))$$

Passo a passo

- Repetição

- Atualização dos parâmetros de largura:

$$\frac{1}{s_{ij}^2} = \frac{\left\{ \prod_{h=1}^p \left[\sum_{k=1}^n (u_{ki})^m \mathcal{K}^{(s_i)}(\mathbf{x}_k, \mathbf{g}_i) (x_{kh} - g_{ih})^2 \right] \right\}^{\frac{1}{p}}}{\sum_{k=1}^n (u_{ki})^m \mathcal{K}^{(s_i)}(\mathbf{x}_k, \mathbf{g}_i) (x_{kj} - g_{ij})^2}$$

- Atualização dos protótipos:

$$\mathbf{g}_i = \frac{\sum_{k=1}^n (u_{ki})^m \mathcal{K}^{(s_i)}(\mathbf{x}_k, \mathbf{g}_i) \mathbf{x}_k}{\sum_{k=1}^n (u_{ki})^m \mathcal{K}^{(s_i)}(\mathbf{x}_k, \mathbf{g}_i)}$$

- Cálculo do grau de pertencimento.
- Cálculo da função objetivo J.

Implementação em Python

- Utilização da biblioteca Numpy.
- Uso do “broadcasting”: looping dos cálculos numéricos dentro da linguagem C ao invés do Python.
- Substituição do valor 0 para números pequenos (da ordem de $1e-20$) para o cálculo dos parâmetros de largura **s** e o grau de pertencimento **u**.
- Transformação de produtório em somatório através do uso de logaritmo para o cálculo dos parâmetros de largura **s**.

Resultados

Tabela 1. *Tabela com valores das métricas referentes à base multiple features fac*

Hiperparâmetro "m"	Métricas		
	Acurácia	Índice Rand Corrigido	MPC
1.3	0.6724 ± 0.0636	0.4845 ± 0.0780	0.1020 ± 0.0140
1.4	0.6520 ± 0.0659	0.4595 ± 0.0776	0.0505 ± 0.0087
1.5	0.6371 ± 0.0684	0.4530 ± 0.0763	0.0282 ± 0.0054
1.6	0.6190 ± 0.0686	0.4398 ± 0.0769	0.0169 ± 0.0031
1.7	0.6077 ± 0.0680	0.4302 ± 0.0757	0.0109 ± 0.0020
1.8	0.5930 ± 0.0644	0.4148 ± 0.0709	0.0074 ± 0.0014
1.9	0.5695 ± 0.0560	0.3915 ± 0.0629	0.0054 ± 0.0009

Resultados

Tabela 7. Matriz de confusão para $m = 1.3$ referente à base multiple features fac para a Questão 1

Classes	Clusters									
	1	2	3	4	5	6	7	8	9	10
0	0	0	6	0	0	0	1	106	1	86
1	10	3	4	1	1	166	10	0	5	0
2	0	4	0	2	3	0	190	0	1	0
3	0	9	0	5	9	2	5	0	170	0
4	156	0	0	0	0	25	0	6	13	0
5	4	45	47	0	1	0	22	4	77	0
6	189	0	8	0	0	0	0	1	2	0
7	0	19	0	84	95	0	2	0	0	0
8	2	0	177	0	0	0	10	4	6	1
9	0	183	1	0	1	0	10	0	5	0

Resultados

Tabela 2. *Tabela com valores das métricas referentes à base multiple features fou*

Hiperparâmetro "m"	Métricas		
	Acurácia	Índice Rand Corrigido	MPC
1.3	0.3547 ± 0.0061	0.1800 ± 0.0038	0.0743 ± 0.0040
1.4	0.2861 ± 0.0340	0.1471 ± 0.0134	0.0265 ± 0.0063
1.5	0.1985 ± 0.0000	0.1114 ± 0.0000	0.0000 ± 0.0000
1.6	0.1985 ± 0.0000	0.1114 ± 0.0000	0.0000 ± 0.0000
1.7	0.1985 ± 0.0000	0.1114 ± 0.0000	0.0000 ± 0.0000
1.8	0.1980 ± 0.0000	0.1112 ± 0.0000	0.0000 ± 0.0000
1.9	0.1980 ± 0.0000	0.1112 ± 0.0000	0.0000 ± 0.0000

Resultados

Tabela 8. *Matriz de confusão para $m = 1.3$ referente à base multiple features fou para a Questão 1*

Classes	Clusters									
	1	2	3	4	5	6	7	8	9	10
0	0	18	0	2	0	0	0	0	180	0
1	0	13	0	148	0	0	39	0	0	0
2	0	14	0	156	0	0	30	0	0	0
3	0	1	0	72	0	0	127	0	0	0
4	0	1	0	26	0	0	173	0	0	0
5	0	0	0	70	0	0	130	0	0	0
6	0	4	0	28	0	0	168	0	0	0
7	0	0	0	21	0	0	179	0	0	0
8	0	196	0	0	0	0	4	0	0	0
9	0	6	0	27	0	0	167	0	0	0

Resultados

Tabela 3. *Tabela com valores das métricas referentes à base multiple features zer*

Hiperparâmetro "m"	Métricas		
	Acurácia	Índice Rand Corrigido	MPC
1.3	0.4891 ± 0.0092	0.3066 ± 0.0075	0.3487 ± 0.0072
1.4	0.4506 ± 0.0046	0.2972 ± 0.0029	0.1677 ± 0.0026
1.5	0.4196 ± 0.0158	0.2769 ± 0.0229	0.0870 ± 0.0071
1.6	0.3582 ± 0.0273	0.2165 ± 0.0249	0.0388 ± 0.0016
1.7	0.2879 ± 0.0176	0.1701 ± 0.0112	0.0210 ± 0.0002
1.8	0.2841 ± 0.0132	0.1613 ± 0.0092	0.0138 ± 0.0000
1.9	0.3028 ± 0.0190	0.1612 ± 0.0145	0.0093 ± 0.0000

Resultados

Tabela 9. *Matriz de confusão para $m = 1.3$ referente à base multiple features zer para a Questão 1*

Classes	Clusters									
	1	2	3	4	5	6	7	8	9	10
0	1	0	95	0	2	0	0	98	1	3
1	0	80	2	0	26	84	0	5	1	2
2	6	0	7	1	7	3	67	2	3	104
3	8	0	2	7	28	7	106	0	10	32
4	5	49	0	7	79	56	0	1	3	0
5	7	2	4	1	15	8	35	28	37	63
6	107	0	5	4	11	10	5	3	50	5
7	1	0	0	161	21	0	1	0	16	0
8	9	0	105	0	2	0	4	64	8	8
9	117	1	3	1	14	8	4	1	46	5

Questão 02

Aplicação de classificadores pré-existentes aos datasets

Descrição

- Realizar o treinamento de cada dataset em quatro classificadores diferentes:
 - O Bayesiano Gaussiano, o K-Vizinhos, a Janela de Parzen e a Regressão Logística.
 - Utilizamos as implementações disponíveis no SciKit Learn ajustando quando necessário.
- Construir um classificador composto utilizando a regra do voto majoritário
- Realizar uma validação cruzada estratificada 30 x 10-folds.
- Calcular métricas a respeito do desempenho dos classificadores e realizar testes de hipótese

Processo de Tuning de Hiper-parâmetros

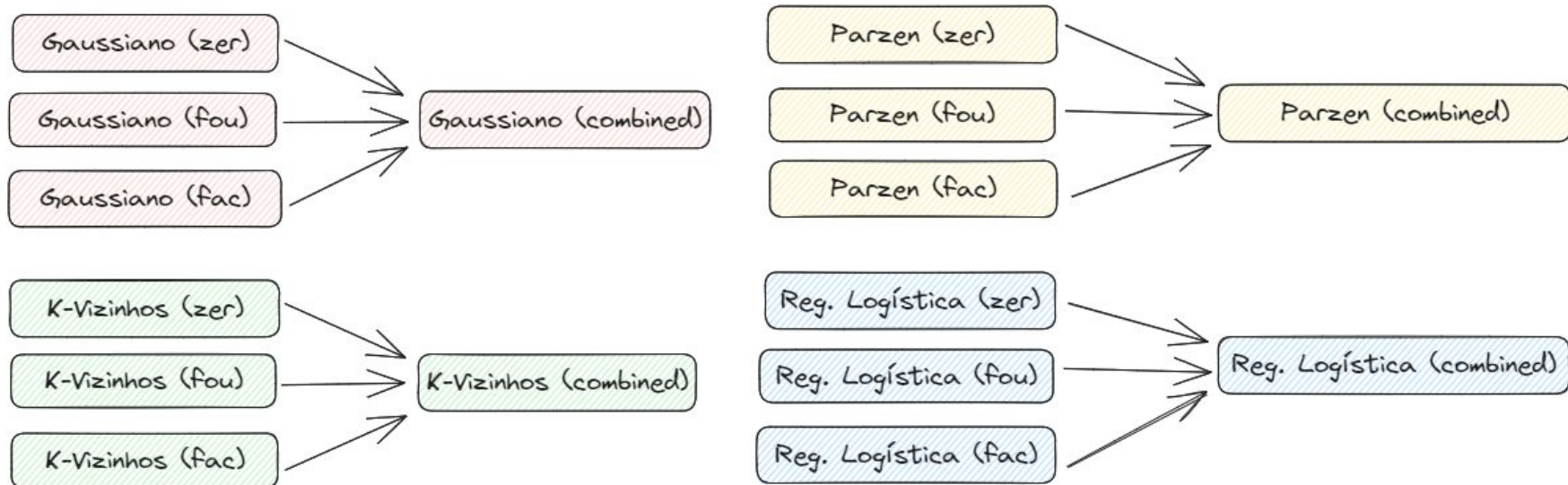
1. Para cada uma das 30 iterações do treinamento, escolhemos uma das permutações de *folds*.
2. Selecionamos aleatoriamente uma amostra estratificada de 20% do conjunto de treinamento.
3. Executamos uma busca de valores para cada hiper-parâmetro. A combinação de valores que maximizou a acurácia foi salva.
4. Por fim, os valores dos hiper-parâmetros foram definidos utilizando critérios como a moda e média aritmética.

Classificador	Hiperparâmetro	Valor em FAC	Valor em FOU	Valor em ZER
Bayesiano Gaussiano	Fator de regularização	1	0.001	1
K-Vizinhos	k	4	5	5
Parzen	Largura de banda	26.060	0.035	20.833
Regressão Logística	Função de Penalidade	Lasso	Ridge	Ridge
	Fator de regularização	0.1	0.1	1000

Composição dos Classificadores

- Para cada classificador base, construímos um classificador composto.
- Recebe o exemplo com os atributos de cada *dataset* e repassa à cada classificador que o compõe apenas os atributos relevantes para que seja realizada a predição da classe ao qual o exemplo pertence.
- As predições são avaliadas utilizando a regra do voto majoritário. Em caso de empate, o classificador escolhe aleatoriamente uma das classes.

Composição dos Classificadores



Estimativa pontual e intervalo de confiança a 95%

Classificador Combinado	Taxa de Erro	Precisão	Cobertura	Score F1
Regressão Logística	0.062 ± 0.002	0.939 ± 0.002	0.938 ± 0.002	0.938 ± 0.002
K-Vizinhos	0.100 ± 0.002	0.902 ± 0.002	0.900 ± 0.002	0.900 ± 0.002
Bayesiano Gaussiano	0.138 ± 0.002	0.867 ± 0.002	0.862 ± 0.002	0.861 ± 0.002
Parzen	0.249 ± 0.003	0.759 ± 0.003	0.751 ± 0.003	0.751 ± 0.003

Testes de Significância Estatística

- Objetivo: comparar os classificadores para verificar a presença de diferenças significativas entre eles.
- Testes realizados:
 - Teste não paramétrico de Friedman;
 - Pós-testes de Nemenyi
- Significância: **5%**
- Os grupos de análise escolhidos foram:

Group 1	Group 2	Group 3	Group 4	Combined group
bayesian gaussian (fac)	bayesian knn (fac)	bayesian parzen (fac)	logistic regression (fac)	combined bayesian gaussian
bayesian gaussian (fou)	bayesian knn (fou)	bayesian parzen (fou)	logistic regression (fou)	combined bayesian knn
bayesian gaussian (zer)	bayesian knn (zer)	bayesian parzen (zer)	logistic regression (zer)	combined bayesian parzen
combined bayesian gaussian	combined bayesian knn	combined bayesian parzen	combined logistic regression	combined logistic regression

Resultados dos testes de significância Estatística

- Teste de Friedman:
 - Todos os 5 grupos **rejeitaram a hipótese** nula com resultados com ***p value*** $<< 0.00001$ para todas as métricas;
- Pós-teste de Nemenyi:
 - Apenas os 2 classificadores em azul tiveram ***p value*** > 0.05 , todos os demais rejeitaram a hipótese nula com ***p value*** $<< 0.05$;

Group 1	Group 2	Group 3	Group 4	Combined group
bayesian gaussian (fac)	bayesian knn (fac)	bayesian parzen (fac)	logistic regression (fac)	combined bayesian gaussian
bayesian gaussian (fou)	bayesian knn (fou)	bayesian parzen (fou)	logistic regression (fou)	combined bayesian knn
bayesian gaussian (zer)	bayesian knn (zer)	bayesian parzen (zer)	logistic regression (zer)	combined bayesian parzen
combined bayesian gaussian	combined bayesian knn	combined bayesian parzen	combined logistic regression	combined logistic regression

Conclusões

- Questão 1 - Estratégia não supervisionada:
 - A melhor configuração foi 'm' = 1.3 na base fac, com acurácia 0.6724 ± 0.0636
- Questão 2 - Estratégia supervisionada:
 - As melhores métricas foram do modelo de regressão logística, com $f\text{-measure} = 0.938 \pm 0.002$
- A melhor estratégia para o problema analisado foi a estratégia supervisionada;