Projeto AM 2020-1

Francisco de A. T. de Carvalho[1]

1 Centro de Informatica-CIn/UFPE
Av. Prof. Luiz Freire, s/n -Cidade Universitaria, CEP 50740-540, Recife-PE, Brasil,
fatc@cin.ufpe.br

1) Considere os dados "multiple features" do site uci machine learning repository

(http://archive.ics.uci.edu/ml/).

- Normalize os dados e compute 3 matrizes de dissimilaridade (uma para cada tabela de dados mfeat-fac (VIEW1), mfeat-fou (VIEW2), mfeat-kar (VIEW3)) usando a distancia Euclidiana.
- Execute o algoritmo "Partitioning fuzzy K-medoids clustering algorithms with relevance weight for each dissimilarity matrix estimated locally" 100 vezes para obter uma partição fuzzy em 10 grupos e selecione o melhor resultado segundo a função objetivo, considerando os seguintes casos: i) simultaneamente nessas 3 matrizes de dissimilaridade; ii) em cada uma delas individualmente.
- Para detalhes do algoritmo "Partitioning fuzzy K-medoids clustering algorithms with relevance weight for each dissimilarity matrix estimated locally" veja o artigo: "F.A.T. de Carvalho, Y. Lechevallier, F.M. Melo, Relational Partitioning Fuzzy Clustering Algorithms Based on Multiple Dissimilarity Matrices, Fuzzy Sets and Systems, 215 (2013), 1-28". Implemente a seguinte variante desse algoritmo:
  - Função objetivo: equação 6; Matching function: equação 7
  - Cálculo do prototipo: proposição 2.3
  - Cálculo dos pesos de relevância das visões nos grupos: equação 9
  - Cálculo do grau de pertinência de um objeto em um grupo: equação 11 com s=1
- Para cada partição fuzzy, calcule o Modified partition coefficient e o Partition entropy. Comente.
- Para cada partição fuzzy, produza uma partição crisp em 10 grupos e calcule o índice de Rand corrigido, a F-measure e erro de classificação (atribuição) em relação à partição à priori em 10 classes.
- Observações:
  - Parametros: $K = 10$; $m = 1.6$; $T = 150$; $\epsilon = 10^{-10}$;
  - Para o melhor resultado imprimir: i) o set-medoids (protótipo) ii) a partição crisp (para cada grupo, a lista de objetos), iii) o numero de objetos de cada grupo crisp, iv)o Modified partition coefficient e o Partition entropy v) 0 indice de Rand corrigido, a F-measure e erro de classificação (atribuição).

2) Considere novamente os dados "multiple features". **Os exemplos são rotulados segundo a partição crisp obtida com o algoritmo de agrupamento da questão 1) aplicado simultâneamente as 3 matrizes de dissimilaridade**.

a) Use validação cruzada estratificada “30 times ten fold” para avaliar e comparar os classificadores combinados descritos abaixo. Se necessário, retire do conjunto de aprendizagem, um conjunto de validação para fazer ajuste de parametros e depois treine o modelo novamente com os conjuntos aprendizagem + validação.

b) Obtenha uma estimativa pontual e um intervalo de confiança para a taxa de acerto de cada classificador;

c) Usar o Friedman test (teste não parametrico) para comparar os classificadores;

Considere os seguintes classificadores:

i) Classificador combinado pela regra da soma a partir do classificador bayesiano gaussiano. Classificador bayesiano gaussiano: considere a seguinte regra de decisão: afetar o exemplo $\mathbf{x}_k$ à classe ω_l se $P(\omega_l|\mathbf{x}_k) = \max_{i=1}^{10} P(\omega_i|\mathbf{x}_k)$ com $P(\omega_i|\mathbf{x}_k) = \frac{p(\mathbf{x}_k|\omega_i)P(\omega_i)}{\sum_{r=1}^{c} p(\mathbf{x}_k|\omega_r)P(\omega_r)}$ $(1 \leq l \leq 10)$

a) Use a **estimativa de maxima verossimilhança** para $P(\omega_i)$

b) Para cada classe ω_i $(i = 1, \ldots, 10)$ use a seguinte estimativa de máxima verossimilhança de $p(\mathbf{x}_k|\omega_i) = p(\mathbf{x}_k|\omega_i, \boldsymbol{\theta}_i)$, supondo uma normal multivariada:

$p(\mathbf{x}_k|\omega_i, \boldsymbol{\theta}_i) = (2\pi)^{-\frac{d}{2}} (|\boldsymbol{\Sigma}_i^{-1}|)^{\frac{1}{2}} \exp\left\{-\frac{1}{2}(\mathbf{x}_k - \boldsymbol{\mu}_i)^{tr}\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}_k - \boldsymbol{\mu}_i)\right\}$, onde

$\boldsymbol{\theta}_i = \begin{pmatrix} \boldsymbol{\mu}_i \\ \boldsymbol{\Sigma}_i \end{pmatrix}$, $\boldsymbol{\Sigma_i} = diag(\sigma_{i1}^2, \ldots, \sigma_{id}^2)$

$\boldsymbol{\mu}_i = \frac{1}{n}\sum_{k=1}^{n} \mathbf{x}_k$, $\sigma_{ij}^2 = \frac{1}{n}\sum_{k=1}^{n}(x_{kj} - \mu_{ij})^2$ $(1 \leq j \leq d)$

c) Classificador combinado pela regra da soma: afetar o exemplo $\mathbf{x}_k$ a classe ω_j se

$$(1 - L)P(\omega_j) + P_{GAUSS,VIEW1}(\omega_j|\mathbf{x}_k) + P_{GAUSS,VIEW2}(\omega_j|\mathbf{x}_k) + P_{GAUSS,VIEW3}(\omega_j|\mathbf{x}_k) = \max_{r=1}^{10}\left[(1 - L)P(\omega_r) + P_{GAUSS,VIEW1}(\omega_r|\mathbf{x}_k) + P_{GAUSS,VIEW2}(\omega_r|\mathbf{x}_k) + P_{GAUSS,VIEW3}(\omega_r|\mathbf{x}_k)\right]$$

ii) Usar um classificador combinado pela regra da soma a partir do classificador bayesiano baseado em k-vizinhos para fazer a classificação dos dados.

a) Treine três classificadores bayesianos baseados em k-vizinhos, um para cada view. Normalize os dados e use a distância Euclidiana para definir a vizinhança. Use conjunto de validação para fixar o o número de vizinhos k.

b) Regra da soma: afetar o exemplo $\mathbf{x}_k$ a classe ω_j se

$$(1-L)P(\omega_j) + P_{KVIZ,VIEW1}(\omega_j|\mathbf{x}_k) + P_{KVIZ,VIEW2}(\omega_j|\mathbf{x}_k) + P_{KVIZ,VIEW3}(\omega_j|\mathbf{x}_k) =$$

$$\max_{r=1}^{10}\left[(1-L)P(\omega_r) + P_{KVIZ,VIEW1}(\omega_r|\mathbf{x}_k) + P_{KVIZ,VIEW2}(\omega_r|\mathbf{x}_k) + P_{KVIZ,VIEW3}(\omega_r|\mathbf{x}_k)\right]$$

com $L = 3$ (três views: mfeat-fac (VIEW1), mfeat-fou (VIEW2), mfeat-kar (VIEW3))

iii) Usar um classificador combinado pela regra da soma a partir do classificador bayesiano baseado em janela de Parzen para fazer a classificação dos dados.

a) Treine três classificadores bayesianos baseados em janela de Parzen, um para cada view. Use a função de kernel multivariada produto com um mesmo h para todas as dimensões e a função de kernel Gaussiana unidimensional. Use conjunto de validação para fixar o parâmetro h.

b) Regra da soma: afetar o exemplo $\mathbf{x}_k$ a classe ω_j se

$$(1-L)P(\omega_j) + P_{PARZEN,VIEW1}(\omega_j|\mathbf{x}_k) + P_{PARZEN,VIEW2}(\omega_j|\mathbf{x}_k) + P_{PARZEN,VIEW3}(\omega_j|\mathbf{x}_k) =$$

$$\max_{r=1}^{10}\left[(1-L)P(\omega_r) + P_{PARZEN,VIEW1}(\omega_r|\mathbf{x}_k) + P_{PARZEN,VIEW2}(\omega_r|\mathbf{x}_k) + P_{PARZEN,VIEW3}(\omega_r|\mathbf{x}_k)\right]$$

Observações Finais

- No Relatório deve estar bem claro como foram organizados os experimentos de tal forma a realizar corretamente a avaliação dos modelos e a comparação entre os mesmos. Fornecer também uma descrição dos dados.
- Data de apresentação e entrega do projeto: TERÇA-FEIRA 08/09/2020
- Colocar no google classroom : o programa fonte, o executável (se houver), os dados e o relatório do projeto
- Tempo de apresentação: 15 minutos para cada equipe (rigoroso).
- Apresença de todos os membros de cada equipe é obrigatória durante a apresentação;