IA - Eduardo

Parte 1

1) Agrupamento de Dados

- Medida que calcula similaridade:
cos O = produto escalar (u,v) / produto vetorial (u,v)

Example 1. $\overline{a} = \{3; 4\}$ and $\overline{b} = \{4; 3\}$.

Produto escalar (u,v) = $\overline{a} \cdot \overline{b} = 3 \cdot 4 + 4 \cdot 3 = 12 + 12 = 24$

Produto vetorial (u,v): $|a| = \sqrt{3^2 + 4^2} = \sqrt{9 + 16} = \sqrt{25} = 5$ $\quad |a| * |b| = 5 * 5 = 25$

$|b| = \sqrt{4^2 + 3^2} = \sqrt{16 + 9} = \sqrt{25} = 5$

$$\cos \alpha = \frac{\overline{a} \cdot \overline{b}}{|\overline{a}| \cdot |\overline{b}|} = \frac{24}{5 \cdot 5} = \frac{24}{25} = 0.96$$ → angulôs com similaridade de 96%

- Data Mining: Agrupamento, pex: (distancia euclidiana)

- **K-means** -> parte do pressuposto que sabemos o numero de clusters (grupos)

1) escolher cruz aleatorio -> O(1) / calcular vetor medio
2) recalcular distancia para cada cruz → O(n 'pontos' * k 'dist'), classificar grupos
3) repetir até que as cruzes nao se deslocam mais

-> O(i * k * N * n)
i = iterações, k = clusters, N = objetos, n = atributos

Tempo: - computar distancia euclidiana custa caro
-> Acelerar: elevar ao quadrado distancia euclidiana
-> Paralelizar ou distribuir o processamento

2) Regressão

- Modelo: descrição aproximada da realidade (ex: uso de funções matematicas)

Ideia simples e bem estudada: induzir **modelos locais** (mais simples) que aproximem suficientemente bem o polinômio. (ex: função 1* grau)

- Treinar 4 modelos de regressão (um por grupo)
- Mover linhas/colunas para os grupos que minimizam o erro;
- MSE global é (garantidamente) minimizado nas iterações:

Exs:
- LASSO (Regressão Linear)
- Arvores de Regressão/ random forests
- KNN
- Redes Neurais, SVM

Parte 2 – Classificadores

1) 1R (função discriminadora)
-> O(n * m)
n = atributos, m = valor de atributos

2) Classificador Bayesiano
→ O(n * m)
n = linhas, m = colunas (dados)

3) KNN
→ O(n * m)
n = linhas, m = colunas (dados)

a) d(9,1) = 65
d(9,2) = 65
d(9,3) = 149
d(9,4) = 159
d(9,5) = 226
d(9,6) = 60
d(9,7) = 46
d(9,8) = 31

b)
k = 1 : exemplo 8 -> Classe B
k = 2 : exemplo 8,7 -> 1B e 1A -> Empate (Ponderação)
k = 3 : exemplo 8,7,6 -> 2B e 1A -> B
k = 4 : 1) exemplo 8,7,6,1 -> 2B e 2A -> Empate (Ponderação inverso da distância manh)
voto do 8(B) = 1/31
voto do 7(A) = 1/46
voto do 6(B) = 1/60
voto do 1(A) = 1/65
B: 1/31 + 1/60 → B
A: 1/46 + 1/65
2) exemplo 8,7,6,2 -> 3B e 1A -> B

k = 5 : exemplo 8,7,6,1,2 -> 3B e 2A -> B

c) Escolher por meio da validação cruzada livronauta para k = 1, 2,..,8 e escolher a o k com maior taxa de acerto

d) descreva um algoritmo para otimizar algoritmos os parâmetros do k-NN, k em {1,2 .. 8}

```
best_accuracy = 0
best_k = 0
for i in range k<=8
        obtain k_accuracy by cross validation(i)
        if k_accuracy > best_acuracy
                best_acurracy = k_acuracy
                best_k = i
```

$cross\text{-}validation:\ O(C(n,k))$, $\mathrm{n} = \mathrm{atributos}, \mathrm{k} = \mathrm{k}$