Prova de IC – UFPA – 21/dez/2020

Para resolver as questões, cada discente deve usar o “dataset” correspondente ao seu número de matrícula (exemplo: estudante_201504050090.txt), disponível juntamente com a prova. A nota será zero caso, mesmo por engano, seja usado o “dataset” errado.

1) (7 pontos, 1 por classificador caso consiga explicar todos os que constam na resolução) Para cada um dos 7 métodos de aprendizado abaixo, projete um classificador que tenha o menor erro possível no conjunto de treino (não preocupe com “overfitting” / sobreajuste). Daí: a) descreva em detalhes suficientes os passos de matemática usados para gerar o classificador (aqui você pode contar com um software para lhe ajudar a calcular valores, mas transponha os resultados para texto e explique cada passo, não precisando descrever passos repetidos), b) estude e indique o melhor pré-processamento dos dados (normalização, etc.) para cada caso, e busque economizar o custo computacional do pré-processamento (eventualmente até não fazendo pré-processamento, caso o mesmo não seja útil). c) Usando um software (Python, Octave, etc.) gere uma figura como a Figura 2 no documento das “soluções” dos exercícios com as regiões de decisão deste classificador e indique também a localização dos exemplos de treino, diferenciando suas classes. d) Quando pertinente, indique os parâmetros do modelo que foram mais importantes para minimizar o erro no conjunto de treino, incluindo os de qualquer pré-processamento dos dados (normalização, etc.). **IMPORTANTE**: *Só inclua a resolução dos classificadores que você entende e consegue explicar a matemática. Caso não entenda uma das resoluções, todos os (até 7) pontos da questão serão perdidos. Exemplo: se entende 5 dos 7, inclua a resolução apenas dos 5. Se incluir 6 e não souber 1, ficará com 0.*

a) Classificador “decision stump”
b) Classificador SVM com kernel linear
c) Classificador SVM com kernel Gaussiano
d) Classificador árvore de decisão (com critério GINI ou Entropia)
e) AdaBoost (versão original) tendo a “decision stump” como classificador fraco
f) Classificador Naive Bayes modelando as likelihoods como distribuições Gaussianas
g) Classificador Naive Bayes versão de Bernoulli (binarize adequadamente a entrada)

2) (1 ponto) Para o classificador “decision stump” projetado na questão 1, informe sua matriz de confusão e, usando o método descrito na questão 5.7 (vide “soluções), informe qual a probabilidade de erro P(e) assumindo não o conjunto de treino para estimativa da mesma, mas que as “likelihoods” condicionais “verdadeiras” são distribuições uniformes nas faixas U(m-10, m+1) e U(m-1, m+6), onde m é a média no conjunto de treino dos valores para a “feature” escolhida para a “decision stump” projetada por você. Por exemplo, se m=3, daí as duas classes têm “likelihoods” U(-7, 4) e U(2, 9) e as probabilidades a priori (“priors”) de cada classe são estimadas a partir do conjunto de treino.

3) (1 ponto) Converta a SVM linear da questão 1 em um perceptron e informe todos os parâmetros do perceptron. Avalie a eventual redução no custo computacional (em número de operações aritméticas) ao se utilizar um perceptron ao invés da SVM em si.

4) (1 ponto) Para a SVM com kernel Gaussiano da questão 1, mostre em sua figura das regiões de decisão (aprimore a Fig. 2 das “soluções”), quem são os exemplos que correspondem a vetores de suporte. E para a árvore de decisão, gere uma figura PNG no estilo da Fig. 1 das “soluções”, inclua na sua prova e explique a diminuição de “impureza” observada para a árvore projetada.