Projeto 1

Considere as amostras de uma base de dados, cada qual com um vetor de características e rótulo verdadeiro da classe. Dado um arquivo contendo uma base com N amostras, faça um programa para dividir esta base em três conjuntos; Z_1 - treinamento; Z_2 - avaliação, e Z_3 - teste; gravando eles em arquivos separados. O número de amostras em cada conjunto deve ser definido por parâmetros do programa. Use um percentual para cada conjunto (e.g., 25% para Z_1, 25% para Z_2, e 50% para Z_3), mas garanta que este percentual também é respeitado para cada classe.

Considere o classificador Knn (*K nearest neighbors* ou K vizinhos mais próximos em Z1).

Para classificar uma amostra s de avaliação (ou de teste) com este classificador, você deve buscar as K amostras mais próximas de s em Z_1 de acordo com a distância Euclideana entre os respectivos vetores de características. O rótulo mais frequente entre as K amostras mais próximas é o rótulo usado para classificar s. Se este não for o rótulo verdadeiro de s, então você conta um erro.

Seu objetivo é projetar um classificador usando os conjuntos Z_1 e Z_2 de forma a minimizar o número de erros em Z_2. Uma vez projetado, você estará assumindo que este classificador também obterá um erro baixo em Z_3. Esta metodologia é interessante por duas razões: (1) Ela demonstra a capacidade de aprendizado com os erros, usando as amostras de Z_2, e (2) também demonstra a robustez do classificador, no caso dos erros em Z_3 serem da mesma ordem de grandeza dos erros em Z_2.

Projeto do classificador.

Método 1:

Escreva um programa para ler os arquivos de Z_1 e Z_2, e descobrir qual valor de K (1,3,5,7,9,...) minimiza os erros em Z_2. Primeiramente encontre o valor ótimo de K. Após encontrar o melhor valor de K, classifique novamente as amostras de Z_2 identificando as amostras erroneamente classificadas. As amostras erroneamente classificadas em Z_2 são trocadas por amostras da mesma classe em Z_1, e o processo se repete por T iterações (T pode ser um parâmetro do programa junto com os nomes dos arquivos de Z_1 e Z_2). Ao final de T iterações, você deve identificar qual instância de Z_1 gerou o menor número de erros em Z_2. Este arquivo com a instancia melhor de Z_1 deverá ser usado depois para testar seu classificador em Z_3.

Método 2:

Note que você pode sugerir mudanças no projeto do seu classificador para termos um outro método. Uma possibilidade é selecionar o melhor K a cada nova instância de Z_1, de modo a garantir que o par (Z_1,K) será o que gera o erro mínimo em Z_2. Neste caso, além de gravar Z_1, você deve armazenar o número de erros e o valor ótimo de K para este Z_1. O método de escolha também pode ser um parâmetro do programa.

Testes com o classificador para comparar os métodos 1 e 2.

Para comparar os métodos 1 e 2 de projeto, você deve testar o classificador em Z_3 e verificar qual método levou ao menor número de erros. Faça um programa para ler o arquivo de Z_3, o melhor arquivo de Z_1 (que pode ser um do método 1 ou do método 2) e o K ótimo. Classifique as amostras de Z_3 e meça o número de erros. Para que sua medida seja confiável, repita o processo inteiro, desde a divisão da base em três conjuntos, e meça o erro em Z_3 várias vezes (30, por exemplo). Calcule a média e o desvio padrão desta medida. Um método pode ser dito melhor que o outro se não houver superposição entre seus intervalos de erro em torno das médias.

Para este projeto você pode selecionar no mínimo 5 bases de dados do site http://archive.ics.uci.edu/ml/

Ex.: Iris, Car Evaluation, Contraceptive Method Choice, Haberman's Survival, Pima Indians Diabetes, Dermatology, Ionosphere, Mammographic Masses, Abalone, Magic, Liver Disorders, Wine, Balance Scale,...