Universidade Tecnológica Federal do Paraná
Campus Campo Mourão
Departamento de Computação - DACOM
Prof. Dr. Diego Bertolini
Disciplina: BCC35-G - Inteligência Artificial

Conteúdo: SVM
Data de Entrega: 11/05/2023

---

1) Baixe e instale a libSVM no seu diretório. Disponível nos links abaixo: *http://www.csie.ntu.edu.tw/~cjlin/libsvm/*

2) Compile o código para gerar os executáveis (Verifique se o gnuplot está instalado!)
apt-get install gnuplot
make all

3) Em caso de dúvidas, leia o documento disponível no link abaixo:
http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

4) Baixe os conjuntos de treinamento e teste para este e outros experimentos (.SVM): download

5) Analise as bases de treinamento e teste e responda (ver os arquivos treino.SVM and teste.SVM):
a) Número de classes ; 10 classes
b) Número de Instâncias no Treinamento ; 1000 instâncias
c) Número de Instâncias no Teste ; 1000 instâncias
6) Execute o script python que acompanha a libSVM, chamado easy.py. Esse script faz a busca pelos parâmetros do kernel Gaussiano (g) e da variável de custo (C). O script gera alguns arquivos. Liste quais são esses arquivos gerados:

- teste.SVM.predict
- teste.SVM.scale
- treino.SVM.model
- treino.SVM.range
- treino.SVM.scale
- treino.SVM.scale.out
- treino.SVM.scale.png

6.1) Reporte a acurácia através do experimento acima. 96.2%
6.2) Reporte o número de vetores de suporte encontrados para cada classe e o total;
(NSV: 21 37 47 40 33 39 28 34 47 42)

6.3) Utilize o conjunto de treinamento para treinar um modelo e o mesmo para testar (>python3 easy.py train.svm train.svm). Descreva a taxa de acerto: 99.9%
6.4) Inverta, utilize o conjunto de teste como treinamento e o de treinamento como teste (>python3 easy.py test.svm train.svm). Descreva a taxa de acerto: 96.4%

Segunda Parte:

Usando o scikit-learn(dados)

1) Taxa de acerto usando o Código (configuração padrão usando RBF):
2) Taxa de acerto usando modelos com outros kernels:

0 -- linear: 96%
1 -- polynomial: 92%
2 -- radial basis function: 95%
3 -- sigmoid: 96%

4) A taxa de acerto usando estes dados no k-NN com k = 3: 90%

5) A taxa de acerto usando estes dados na Árvore de Decisão ; 85%

Terceira Parte:

Utilize o conjunto de dados extraído por vocês para o classificador SVM. Compare com os outros classificadores já vistos em aula.

Relatório

Introdução:
O objetivo deste relatório é descrever os experimentos e resultados obtidos a partir do conjunto de dados "treino_5x5.csv", que consiste em imagens de dígitos numéricos divididos em quadrantes de 5x5 pixels. O conjunto de dados possui 50 colunas, representando a contagem de pixels pretos e brancos em cada quadrante.

Experimento 1 - Validação Cruzada:
No primeiro experimento, utilizamos o algoritmo SVM (Support Vector Machine) para classificar os dígitos numéricos. Inicialmente, carregamos o conjunto de dados e separamos as variáveis preditoras (x) e a variável alvo (y). Em seguida, normalizamos os dados usando o MinMaxScaler para que todas as características estejam na mesma escala.

Criamos o modelo SVM utilizando a classe SVC do scikit-learn e definimos o número de folds para a validação cruzada usando a classe StratifiedKFold com 3 splits. Executamos a validação cruzada usando a função cross_val_score, que retorna o resultado da métrica de acurácia para cada fold. Imprimimos a média dos resultados obtidos para avaliar o desempenho geral do modelo.

A acurácia média obtida no experimento de validação cruzada foi de 0.8538808269347191, indicando um desempenho razoável do modelo SVM na classificação dos dígitos numéricos.

Experimento 2 - Grid Search:
No segundo experimento, realizamos uma busca em grid para encontrar os melhores hiperparâmetros para o modelo SVM. Definimos os valores a serem testados para os parâmetros C (constante de regularização), kernel (tipo de kernel), degree (grau do polinômio para o kernel polinomial) e gamma (coeficiente do kernel).

Utilizamos a classe GridSearchCV do scikit-learn, passando o modelo SVC, os valores do grid e o número de folds para a validação cruzada (cv=3). A busca em grid avalia todas as combinações possíveis de hiperparâmetros e retorna o melhor modelo encontrado.

Após a execução do Grid Search, obtemos os melhores valores para cada hiperparâmetro. O melhor valor para a constante de regularização (C) foi 2.0, o melhor kernel foi linear, o melhor grau do polinômio foi 2 e o melhor valor para gamma foi scale. O melhor score obtido pelo modelo com esses hiperparâmetros foi de 0.9339698980417545.

Conclusão:
Neste relatório, descrevemos os experimentos realizados com o conjunto de dados "treino_5x5.csv" utilizando o algoritmo SVM. No experimento de validação cruzada, obtivemos uma acurácia média de 0.8538808269347191, indicando um desempenho razoável do modelo. No experimento de busca em grid, encontramos os melhores hiperparâmetros para o modelo SVM, resultando em uma melhoria significativa no desempenho, com uma acurácia de 0.9339698980417545. Esses resultados sugerem que o modelo SVM é promissor para a classificação de dígitos numéricos com base nas características de pixels pretos