UNIVERSIDADE ESTADUAL DO OESTE DO PARANÁ
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
COLEGIADO DE CIÊNCIA DA COMPUTAÇÃO

Reconhecedor de Números Manuscritos
Aprendizagem de Máquina

Alunos: Gilberto Antunes Monteiro Junior & Henrique Tomé Damasio
Data: 27/06/2019

Descrição do Trabalho

O objetivo do trabalho consiste em propor duas abordagens de extração de características para reconhecer números unitários (0-9) manuscritos, estes da base de dados conhecida como *MNIST*, utilizando uma aprendizagem supervisionada em forma de classificadores, analisando qual é o melhor por meio de tunelamento de seus parâmetros, realizando uma análise individual de cada abordagem em quem foi aplicado o classificador e após uma conjunta, verificando se há diferença significativa entre as abordagens.

Descrição do Conjunto de Dados

O conjunto de dados utilizado durante o experimento consiste na base de dados *MNIST* composta por 42000 imagens de resolução 28x28 em tons de cinza de numerais manuscritos de zero a nove, das quais 4132 são do numeral zero, 4684 do numeral um, 4177 do numeral dois, 4351 do numeral três, 4072 do numeral quatro, 3795 do numeral cinco, 4137 do numeral seis, 4401 do numeral sete, 4063 do numeral oito, 4188 do numeral nove.

Descrição Passo-a-passo do Experimento

O primeiro passo consiste no carregamento do conjunto de dados. Para tanto, carrega-se todas as imagens da base de dados e as armazenamos em uma lista, de modo que cada elemento da lista representa uma imagem.

O segundo passo se deu por identificar e gerar características para que os algoritmos pudessem ser empregados. Para uma primeira abordagem foi realizada uma divisão de cada imagem em 16 seções, e em cada parte faz-se a proporção de pixels pintados pelo tamanho da seção, eliminamos então as seções 0, 3, 12 e 15 pois estas representavam os cantos mais extremos da imagem e portanto não possuíam pixels com informações relevantes, essas proporções foram então salvas em arquivo formato csv para uso posterior.

Para a segunda abordagem empregamos a mesma divisão de 16 seções à imagem, porém dessa vez realizamos um processamento em cada uma delas, antes de realizarmos a divisão. Em cada imagem é aplicado o algoritmo de esqueletização, dessa forma a imagem fica mais fina, e por consequência espera-se que a mesma fique mais propensa e ser

reconhecida pelos classificadores e também salvo em arquivo csv.

Com isso pronto foi então definido os 6 classificadores a serem testados, sendo estes: *Decision Tree*, *K Nearest Neighbor*, *Multi Layer Perceptron*, *Naive Bayes*, *Random Forest* e *Support Vector Machine*.

Feita a identificação e extração de característica, para cada base criada (1ª e 2ª abordagem) disparamos uma *thread*, esta dividindo a base de dados original em três subconjuntos mutuamente exclusivos: treino, teste e validação. A instância que for designada para um conjunto não aparece nos outros. O conjunto de treino possui 80% do tamanho do arquivo original. Já as bases de validação e teste, tem 10% da dimensão, seguindo a estratégia de avaliação Hold-out.

Após essa divisão encontramos os melhores parâmetros para cada classificador de forma que se obtivesse a melhor acurácia possível, os parâmetro foram encontrados por meio de força bruta, fazendo a variação dos mesmos dentro intervalos definidos por meio de testes isolados, para cada abordagem os parâmetros afinados e seus respectivos intervalos ou valores utilizados podem ser vistos na Tabela 1. Para os melhores resultados em cada execução, salva-se também o vetor de votos do classificador, ou seja, a taxa de certeza que o classificador tem quando sugere a qual classe pertence aquela instância. Este processo é então repetido 10 vezes por motivos estatísticos.

Tabela 1 - Parâmetros e intervalos ocilados dos classificadores

Classificador	Parâmetro	Intervalo	Passo do intervalo
Decision Tree	Profundidade Máxima	Sem poda, 1 - 25	2
K Nearest Neighbor	Número de vizinhos	3-19	1
K Nearest Neighbor	Métrica de distância	Uniforme, Inverso da distância	-
Multi Layer Perceptron	Número de iterações	200 - 300	20
Multi Layer Perceptron	Número de camadas escondidas	2 - 5	1
Multi Layer Perceptron	Programação da taxa de aprendizado para atualização dos pesos.	Constant, Invscaling, Adaptive	-
Multi Layer Perceptron	Taxa de aprendizado	1 - 0,001	1/10
Naive Bayes	-	-	-
Random Forest	Quantidade de Árvores	150 - 350	10

Random Forest	Função para medir a qualidade de uma divisão	Gini, Entropy	-
Random Forest	Profundidade Máxima	Sem poda, 1 - 45	2
Support Vector Machine	Penalidade	0,1 - 12,1	1
Support Vector Machine	Kernel	Poly, RBF	-

AVALIAÇÃO DAS ABORDAGENS

Para avaliar o experimento, foram realizadas 10 repetições para os 6 classificadores para cada uma das abordagens de extração de características implementadas, com estas concluídas realiza-se então os testes de *Kruskal-Wallys*, para identificar se existe alguma diferença significativa entre os resultados apresentados pelos classificadores, caso não possa ser identificada tal diferença optamos por utilizar os resultados gerados pelo classificador *Random Forest*, uma vez que nos teste executados préviamente este modelo se demonstrava mais eficiente. Mas se alguma diferença for identificada pelo teste, executa-se então o teste de *Mann-Whitney* para identificar quais classificadores que apresentaram melhores resultados, se apenas um classificador for identificado como melhor este então é selecionado para o cálculo de combinação, caso dois ou mais classificadores sejam apresentados sem diferença significativa, teremos um vetor com estes classificadores, e por não apresentarem diferença significativa acabamos por selecionar o primeiro classificador no vetor.

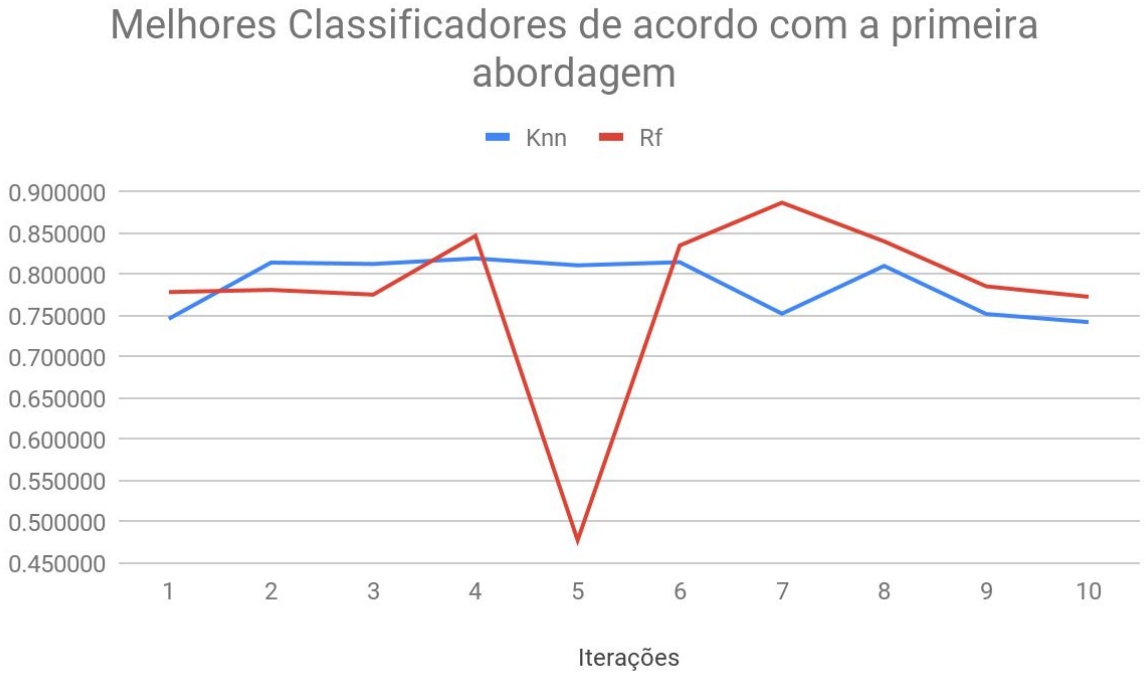
O processo descrito acima é realizado para ambas as implementações, ou seja, cada uma delas selecionará um classificador que tenha obtido a melhor acurácia dentre os 6 avaliados, vale ressaltar que nem sempre o mesmo classificador será selecionado por ambas as abordagens. Deste modo temos dois classificadores e se faz necessário combiná-los, este processo é realizado por meio de duas abordagens, a primeira é a estratégia de combinação Borda Count e a segunda abordagem é a estratégia de combinação regra da soma.

Para a implementação dos métodos de combinação, inicialmente lê-se os dados dos arquivos que contém os votos dos classificadores que foram selecionados (cada arquivo continha 4200 instâncias, 10% do conjunto total), para cada instância atribui-se uma nota de 4 à 1 do maior voto ao menor, não optamos por deixar uma nota de 10 à 1 pois muitas vezes o classificador atribuía nota a apenas 4 classes e as demais ficavam zeradas. Com as notas atribuídas soma-se a nota para a instância 1 de ambos os classificadores, resultando assim em uma nova classe que é avaliada a partir da classe original. Esse processo de atribuição de notas e combinação é repetido enquanto ainda houver instâncias. E o processo todo é repetido 10 vezes, 1 para cada arquivo de votos gerados pelos classificadores.

A forma como a regra da soma foi implementada se assemelha muito a borda count, a diferença se dá na forma como o voto da combinação dos classificadores é calculada. Nesta regra não é atribuída uma nota aos votos das instâncias, mas sim soma-se a porcentagem de certeza dos classificadores para cada classe, e então a classe que possuir a maior soma é tida como a nova classe de resposta, assim como no borda count essa resposta gerada pela regra da soma é avaliada pela classe original. E então a regra da soma é realizada para todas as 4200 instâncias do arquivo lido, e também é realizada para as 10 execuções que o

classificador realizou.

Abaixo pode ser encontrado os gráficos com a acurácia das execuções dos classificadores *K Nearest Neighbors* e *Random Forest* para a primeira abordagem, visto que ambos não apresentaram diferença significativa entre os resultados apresentados, *Random Forest* para a segunda abordagem, e as combinações realizadas.



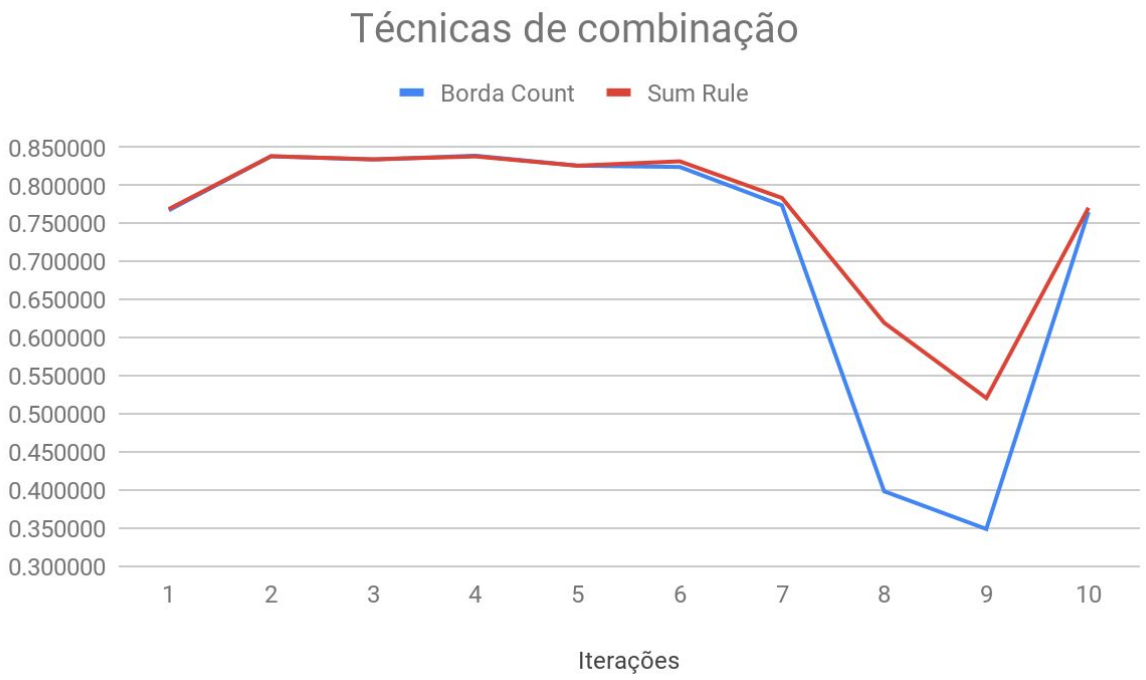
Acurácia média;
KNN = 0,787476
Random Forest = 0,7780

Desvio Padrão:
KNN = 0,03230
Random Forest = 0,106702

Acurácia média = 0,820095
Desvio Padrão = 0,029869

Para a segunda abordagem o classificador que mais se aproximou da *random forest* foi o *knn* com uma acurácia média de 0,787071 e um desvio padrão de 0,031454.

Combinação dos classificadores *K Nearest Neighbors* (primeira abordagem) e *Random Forest* (segunda abordagem)



Acurácia média:
Borda Count = 0,721571
Regra da Soma = 0,763238

Desvio padrão:
Borda Count = 0,176486
Regra da Soma = 0,102322