

UNIVERSIDADE FEDERAL DE PERNAMBUCO

CENTRO DE INFORMÁTICA

**Relatório da Segunda Lista 2**

Rodrigo Monteiro de Moraes de Arruda Falcão

RECIFE

Março de 2021

1. **INTRODUÇÃO**

O presente relatório tem por objetivo detalhar a execução e experimentos da segunda lista de exercícios da disciplina de Aprendizagem de Máquina, que objetiva o exercício de geração e seleção de protótipos. Foram feitas implementações do algoritmo de geração de protótipos **Learning Vector Quantization (LVQ)**, e mais duas variações, que são: **LVQ 2.1** e **LVQ 3.0**. Para este exercício foram utilizadas duas bases de dados do repositório *Promise*, [http://promise.site.uottawa.ca/SERepository/datasets/pc1.arff](http://promise.site.uottawa.ca/SERepository/datasets/pc1.arff%20) e <http://promise.site.uottawa.ca/SERepository/datasets/datatrieve.arff>.

1. **IMPLEMENTAÇÃO**

Para implementar os algoritmos, foi utilizado a linguagem *Python* com a ferramenta *Jupyter Notebook*, para facilitar no desenvolvimento e execução dos scripts. A distância Euclidiana foi utilizada para os cálculos das distâncias necessárias para a execução dos algoritmos, conforme solicitado na especificação do exercício.

* 1. BIBLIOTECAS UTILIZADAS

Foram utilizadas algumas bibliotecas para auxiliar na execução da atividade, são elas: Numpy, Pandas e Sklearn.

* **Numpy**: biblioteca python para manipulação de arrays e matrizes;
* **Pandas**: biblioteca python para manipulação e análise de dados, que oferece estrutura de dados e operações para manipular tabelas e datasets;
* **Sklearn**:biblioteca python de aprendizado de máquina, com diversas funções implementadas.
  1. ESTRUTURA

Após o carregamento dos dados, foi realizado a normalização de todos os dados numéricos de entrada, com o objetivo de evitar que algum atributo domine a medida de distância, como também foram criadas funções para cada variação do LVQ, cálculos das distâncias e cálculo da janela para o LVQ 2.1 e 3.0. Também foi criado uma função para seleção aleatória de protótipos, respeitando as proporções das classes, a partir de conjunto de treinamento. Constantes foram declaradas para os parâmetros dos algoritmos, tais como, o número de épocas, quantidade de folds na validação do *k-fold cross-validation* e quantidade de protótipos (que durante a execução variou).

1. **METODOLOGIA**

Para avaliação dos algoritmos, foi utilizado, a técnica k-fold cross-validation, com o k igual a 5, conforme é de costume na literatura da área. Com esses folds, para cada algoritmo, foi executado cinco iterações, uma para cada fold, conforme é estabelecido pela técnica. O número de épocas utilizado foi 10, e o número de protótipos utilizado para testar os algoritmos começou com 20, depois variou duas vezes, para 40 e 60. A taxa de aprendizado, inicialmente começou com 0.3, porém ao longo das épocas diminuía com o tempo.

Para cada algoritmo, a partir do conjunto de treinamento, dividido em 5 folds, conforme dito anteriormente, é selecionado uma quantidade fixa de protótipos, de forma aleatória, e com isso o algoritmo é executado com esses protótipos. Após a execução, o conjunto resultante do treinamento do LVQ corrente, é utilizado para execução do k-NN com valores de k igual a 1 e 3. Após a execução do k-NN, é calculado a taxa de acerto para cada fold utilizado como teste, e ao final é calculado a média entre as cinco taxas, resultando na taxa de acerto final para o algoritmo.

1. **RESULTADOS**

Para cada base de dados do repositório *Promise*, [http://promise.site.uottawa.ca/SERepository/datasets/pc1.arff](http://promise.site.uottawa.ca/SERepository/datasets/pc1.arff%20) (base de dados 1) e <http://promise.site.uottawa.ca/SERepository/datasets/datatrieve.arff> (base de dados 2), foram executados os algoritmos desenvolvidos, variando o número de protótipos e o valor de k para o k-NN. Foram obtidos os seguintes resultados, levando em consideração a variação do valor k entre os valores, {1,3} e variando o número de protótipos entre os valores {20, 40, 60}, assim como as acurácias do k-NN sem o uso de seleção e geração de protótipos com os mesmos valores de k:

* 1. BASE DE DADOS 1
     1. **LVQ** Para todos os protótipos e valores de K a taxa de acerto foi de 93%.
     2. **LVQ 2.1**Para todos os protótipos e valores de K a taxa de acerto foi de 93%.
     3. **LVQ 3.0**Para todos os protótipos e valores de K a taxa de acerto foi de 93%.
  2. BASE DE DADOS 2
     1. **LVQ**Para 60 protótipos e K=1 a taxa de acerto foi 90%, já para a mesma quantidade de protótipos e K=3 a taxa de acerto foi de 92%. Para 40 protótipos e K=1 a taxa de acerto foi 88%, já para a mesma quantidade de protótipos e K=3 a taxa de acerto foi de 91%. Para 20 protótipos e ambos os valores de K a taxa de acerto foi de 89%.
     2. **LVQ 2.1**Para 60 protótipos e K=1 a taxa de acerto foi 89%, já para a mesma quantidade de protótipos e K=3 a taxa de acerto foi de 90%. Para 40 protótipos e K=1 a taxa de acerto foi 88%, já para a mesma quantidade de protótipos e K=3 a taxa de acerto foi de 92%. Para 20 protótipos e ambos os valores de K a taxa de acerto foi de 92%.
     3. **LVQ 3.0**Para 60 protótipos e K=1 a taxa de acerto foi 89%, já para a mesma quantidade de protótipos e K=3 a taxa de acerto foi de 90%. Para 40 protótipos e K=1 a taxa de acerto foi 89%, já para a mesma quantidade de protótipos e K=3 a taxa de acerto foi de 88%. Para 20 protótipos e ambos os valores de K a taxa de acerto foi de 91%.
  3. k-NN sem o uso de seleção e geração de protótipos
     1. **Base de Dados 1**Para K=1 a taxa de acerto foi de 91% e para K=3 a taxa de acerto foi de 92%
     2. **Base de Dados 2**Para K=1 a taxa de acerto foi de 88% e para K=3 a taxa de acerto foi de 90%

1. **ANÁLISE DOS RESULTADOS**Para a primeira base de dados todas as taxas de acerto foram iguais a 93%, o que mostra a baixa quantidade de instâncias em regiões de indecisão, já que o LVQ tem o mesmo aproveitamento em relação à classificação correta dos protótipos, em relação aos outros algoritmos (por exemplo, a precisão foi maior do que o K-NN sem o LVQ). Já para a segunda base de dados as diferenças entre protótipos e valores de K existem e a taxa de acerto, no geral, foi menor do que para a primeira base de dados, o que mostra uma maior quantidade de instâncias em regiões de indecisão.
2. **CONCLUSÃO**O algoritmo de geração de instâncias, LVQ, demonstrou ser bastante poderoso, já que a acurácia foi maior com que ele do que sem ele, utilizando o k-NN como classificador. Com suas três variações, onde cada uma refina algum aspecto da outra, porém, quando a base de dados não possui muitas instâncias em regiões de indecisão, o LVQ é suficientemente bom. Um aspecto importante, é o tempo de processamento, que é alto, tanto para treinar o LVQ quando para executar o k-NN, pois muitas operações são realizadas.