UFJF-MLTK: Um *Framework* para Algoritmos de Aprendizado de Máquina

Mateus Coutinho Marim (mateus.marim@ice.ufjf.br)
Alessandreia Marta de Oliveira Julio (alessandreia.oliveira@ice.ufjf.br)
Saulo Moraes Villela (saulo.moraes@ufjf.edu.br)
Departamento de Ciência da Computação - Universidade Federal de Juiz de Fora

Introdução

Um dos grandes problemas ao se procurar a implementação de algoritmos de Aprendizado de Máquina (*Machine Learning* – ML) desenvolvidos por pesquisadores é a falta de padronização, tornando difícil a reprodução de experimentos e o uso dos mesmos. Algumas tentativas de abordar esse problema já foram feitas. Uma das primeiras foi com o MLC++ [1], que fornecia uma estrutura de classes integradas que facilitava o desenvolvimento de algoritmos de ML, mas o mesmo não é de fácil acesso.

Nesse trabalho é apresentado o UFJF-MLTK (*Machine Learning Toolkit* da Universidade Federal de Juiz de Fora), um *framework* com o objetivo de atacar esse problema, incialmente desenvolvido com intuito de criar um padrão para os algoritmos desenvolvidos dentro do Departamento de Ciência da Computação (DCC) da UFJF.

Algoritmos de Aprendizado de Máquina

Um algoritmo é dito ser de Aprendizado de Máquina quando é capaz de descobrir novos conhecimentos a partir de observações sobre dados fornecidos como entrada, e são usados para predizer informações de novos dados [2].

Existem vários tipos de algoritmos de ML, dentre eles é possível citar os de aprendizado supervisionado, em que as amostras do conjunto de treinamento possuem suas respectivas saídas, não supervisionado, no qual a saída dos dados de entrada não é conhecida, e semissupervisionado, onde apenas algumas amostras possuem saídas conhecidas. Existem outros tipos de aprendizado, mas esses são os principais tipos de que serão abordados pelo *framework*.

Aplicações de algoritmos de ML se estendem nas mais diversas áreas, desde a detecção de *spams*, reconhecimento de voz, reconhecimento de padrões em imagens, até aplicações médicas como a análise de laudos médicos para tentar descobrir a doença de um paciente [2].

UFJF-MLTK

O objetivo do projeto UFJF-MLTK é oferecer aos pesquisadores e desenvolvedores ferramentas básicas para a implementação e teste de algoritmos de aprendizado. Os utilizadores inicialmente terão a dificuldade de aprender a linguagem do *framework,* mas a longo e médio prazo os benefícios no desenvolvimento vão se tornar evidentes.

As vantagens de um *framework* para algoritmos de ML incluem a possibilidade de se padronizar os algoritmos desenvolvidos, facilitar o seu entendimento por futuros desenvolvedores, diminuir o custo de tempo de um projeto e, também, ser utilizado em aulas de ML para auxiliar o ensino e a aprendizagem do assunto.

Principais componentes

O *framework* desenvolvido segue as características de um projeto orientado a objetos com desenvolvimento orientado ao reuso, ou seja, todas as classes implementadas podem ser reutilizadas em outros projetos conforme a necessidade [3].

Para a manipulação das bases de dados, são usadas as classes “Data”, “Point” e “Statistics”, que permitem utilizar operações para extrair informações estatísticas, remover ou adicionar pontos etc. Para a visualização dos dados é utilizada a classe “Visualisation”.

Também foram desenvolvidos componentes para auxiliar na implementação de classificadores representados pela classe “Classifier”, podendo ser primais ou duais, e de algoritmos de seleção de características, representados pela classe “Feature Selection”. Para a validação dos modelos criados para os conjuntos de entrada, é usado o componente “Validation”, que permite verificar a acurácia dos mesmos.

A Figura 1 apresenta os componentes que fazem parte do núcleo do *framework* até o momento e a forma como se relacionam.

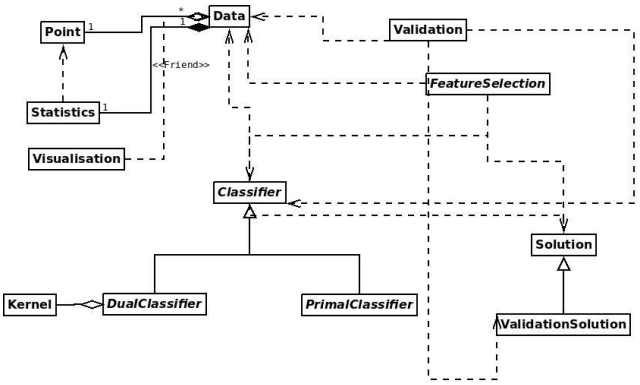

Figura 1: Diagrama de classes simplificado do *framework* proposto.

Considerações finais

Todos os componentes apresentados, considerados como o núcleo principal do *framework*, além de alguns algoritmos desenvolvidos no DCC, já se encontram implementados e prontos para o uso. O próximo passo é a inclusão de algoritmos considerados como estado da arte de ML e de novos componentes para permitir o desenvolvimento de uma variedade maior de algoritmos, como comitês de classificadores, algoritmos de regressão e de agrupamento.

Referências bibliográficas

[1] Kohavi, R; John, G.; Long, R.; Manley, D.; Pfleger, K. **MLC++: a machine learning library in C++**. Proceedings of Sixth ICTAI, 740-743, 1994.

[2] Mitchell, T. M. **Machine Learning.** McGraw-Hill Science/Engineering/Math, 1997.

[3] Pressman, R. S. **Software Engineering: A Practitioners’s Approach.** 7 ed. McGraw-Hill higher Education, 2010.