05

Visualizando as bordas de classificação

Transcrição

Então visualizaremos como nossos algoritmos de classificação estão funcionando.

Para isso, vamos à primeira tela mostrada quando abrimos o Weka, "*Weka GUI Chooser*", ou escolhedor de interfaces gráficas do Weka. Clicaremos então na opção "*Visualization*" na barra superior da janela e em "*BoundaryVisualizer*, o visualizador de bordas do Weka.

Da mesma forma que fizemos anteriormente, para começar a trabalhar precisaremos abrir um dado em "*Open file*". Não abriremos nenhum dado da base principal que estamos trabalhando, a base do banco, dessa vez usaremos um dado clássico, o "iris.2D.arff". Esse será um dado para a classificação de flores, baseado no cumprimento e largura de pétalas. Nesse tipo de dado, teremos três tipos de classes diferentes, diferentemente do nosso caso em que tínhamos apenas duas.

Escolheremos um classificador, assim como fazíamos na aba "*Classifier*". Clicando em "*Choose*" vamos optar pelo J48 que já usamos bastante. Na opção "*Plotting*" selecionaremos "*Plot training data*" e clicaremos em "*Start*". Ele demorará um pouco para rodar, e veremos que ele cria bordas, de forma que tudo que estiver na área de borda azul pertencerá à classe correspondente a essa cor, a do "Iris-virginics". Em vermelho, da "Iris-setosa". Então, as bordas foram criadas de forma quadrada, e é a forma de funcionamento desse algoritmo, pois ele divide por atributos de forma bem fechada.

Outro algoritmo que utilizamos foi o de Naive Bayes. Vamos rodá-lo também e ver que ele se comporta de maneira um pouco mais flexível na situação atual, gerando as bordas mais arredondadas, classificando alguns elementos que não tinham sido classificados, mas ele ainda errará em alguns pontos.

Usamos também o algoritmo IBk, e entre todos, ele aparentemente se comporta de modo mais flexível do que os demais para fazer a classificação de bordas, mas não funcionará tão bem para todo tipo de exemplo. Estamos vendo um exemplo 3d e temos que lembrar que nossos dados são multidimensionais e tem vários atributos, mas aqui estamos visualizando apenas dois. Vemos quais são os atributos mostrados em "*Visualization attributes*", onde veremos o cumprimento da pétala em x e a largura em y.

Enfim, esse tipo de algoritmo funciona bem para enxergarmos como os algoritmos operam em dados que conseguimos separar visualmente. Abrimos o dado "Iris.2d.arff" em especial porque para nosso caso, não conseguimos separar visualmente os dados do nosso banco. Abriremos o arquivo "banco.2atributos.arff" e poderemos comprovar que esse é um dado que não podemos separar visualmente. Mesmo criando uma área, não conseguiremos separar todos os pontos, pois estão muito misturados.

Testaremos o algoritmo J48 para esses dados, que é mais leve, para entender o que acontecerá. Ele vai demorar bem mais para rodar. Quando o procedimento acabar, veremos que foram criadas áreas diferentes. Temos duas classes e haverá uma área mais avermelhada e outra mais esverdeada, o que podemos entender como uma probabilidade de encontrarmos mais elementos de uma determinada classe numa determinada área, mas as áreas não funcionarão da mesma forma que no caso do dado "Iris.2D.arff", em que os dados realmente se dividiam e toda aquela área correspondia a um valor.

De qualquer forma podemos entender melhor como cada algoritmo funciona a partir desse tipo de ferramenta. Compreendemos como o resultado da classificação funcionará e qual opção é mais ou menos flexível, a depender do tipo de problema. O método que vimos, no entanto, será ideal quando conseguirmos separar visualmente as classes da nossa base de dados.