Relatório do Trabalho de Inteligência Artificial: Aprendizado de Máquina

Daniel Atkinson Oliveira[114054054]

Universidade Federal do Rio de Janeiro, Rio de Janeiro RJ, Brasil
danatkhoo@gmail.com

Objetivo

O trabalho descrito nesse relatório é uma das atividades da disciplina "Inteligência Artificial"da Universidade Federal do Rio de Janeiro. O objetivo desta atividade é prever as saídas de dois datasets: Iris Plants Database [1] e Discrimination in Salaries [2]. O modelo que foi implementado para a classificação é o perceptron e nenhuma biblioteca de aprendizado de máquina foi usada, uma das restrições impostas sobre o trabalho.

Metodologia

A linguagem escolhida para escrita do código [3] foi R. Nesta seção serão descritas todas as etapas que foram executadas para obter a implementação final do algoritmo de classificação de perceptron para ambos os datasets.

Análise dos Datasets

Antes de começar a treinar o classificador é importante compreender os datasets que serão usados.

O dataset Iris [1] contém três classes, cada uma representa uma espécie de planta. Essas são:

- setosa;
- versicolour;
- virginica.

Para cada uma dessas classes (atributo espécie chamado de "Species"no arquivo iris.dat) temos 50 observações, totalizando 150 observações.

Para cada uma das 3 classes temos 4 atributos numéricos, que são:

- comprimento da sépala ("Sepal.Length"no arquivo iris.dat)
- largura da sépala ("Sepal.Width"no arquivo iris.dat)
- comprimento da pétala ("Petal.Length"no arquivo iris.dat)
- largura da pétala ("Petal.Width"no arquivo iris.dat)

Tabela 1. Estatísticas do dataset.

Atributo	Min	Max	Esperança	DP	Correlação de classes
comprimento sépala	4.3	7.9	5.84	0.83	0.7826
largura sépala	2.0	4.4	3.05	0.43	-0.4194
comprimento pétala	1.0	6.9	3.76	1.76	0.9490
largura pétala	0.1	2.5	1.20	0.76	0.9565

Para cada uma das 150 observações os valores de todos esses atributos são conhecidos.

Estátisticas referentes ao dataset Iris [1] podem ser consultados na Tabela 2

Correlação baixa das classes nos indica que as observações desse atributo aparecem frequentemente em intervalos com bastante interseção entre classes. Podemos observar esse fenômeno na Figura 1

Conversamente, correlação alta das classes nos indica que as observações desse atributo geralmente possuem valores únicos à classe. Podemos observar esse fenômeno na Figura 2

O dataset Salary [2] contém duas classes, referente à variável "sexo"(sx no arquivo .dat):

- masculino (codificado como 0 no arquivo salary.dat)
- feminino (codificado como 1 no arquivo salary.dat)

Ao contrário do dataset Iris [1] não temos um número de observações iguais para cada classe.

Cada classe tem 5 atributos, que são:

- ranking (rk no arquivo salary.dat, com a seguinte codificação: 1 para professor assistente, 2 para professor associado e 3 para professor efetivado)
- número de anos no ranking atual (yr no arquivo salary.dat)
- maior grau atingido (dg no arquivo salary.dat, com a seguinte codificação: 0 para mestrado e 1 para doutorado)
- número de anos desde que atingiu grau mais alto (yd no arquivo salary.dat)
- salário acadêmico, em dólares (sl no arquivo salary.dat)

Como ambos os datasets tem quantidade de classes diferentes, diferentes estratégias serão usadas para a previsão de classe em cada um.

Validação Cruzada

Tanto para o dataset Iris [1] quanto para o dataset Salary [2], foi feita a validação cruzada 5-fold. Para isso ambos os datasets foram divididos, aleatóriamente, em 5 conjuntos com a mesma quantidade de observações. Uma rodada de validação cruzada 5-fold consiste em 4 dos 5 conjuntos serem utilizados para treino e 1 ser utilizado para teste. Foram feitas 5 rodadas de validação cruzada 5-fold, tal que o conjunto utilizado para teste muda a cada rodada, assim como o conjunto de treino. O processo está ilustrado na Figura 3.

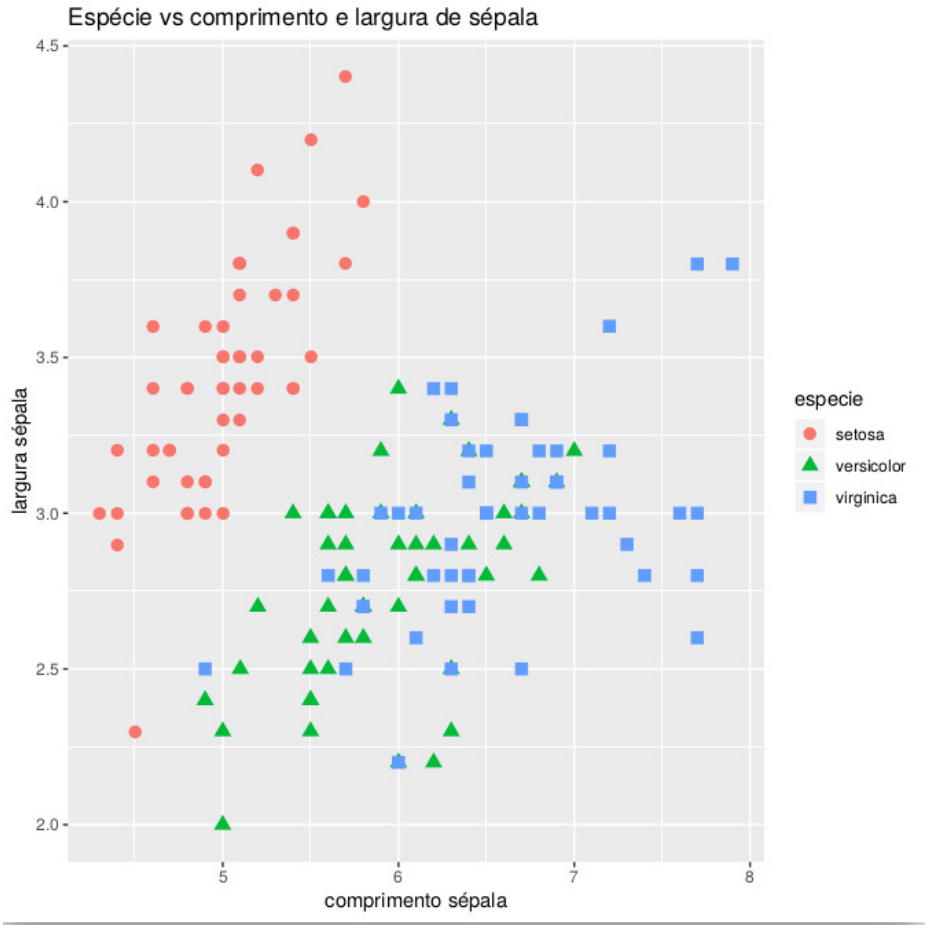

Figura 1. Podemos ver que, tanto no atributo comprimento sépala quanto em largura sépala, há bastante interseção entre os valores

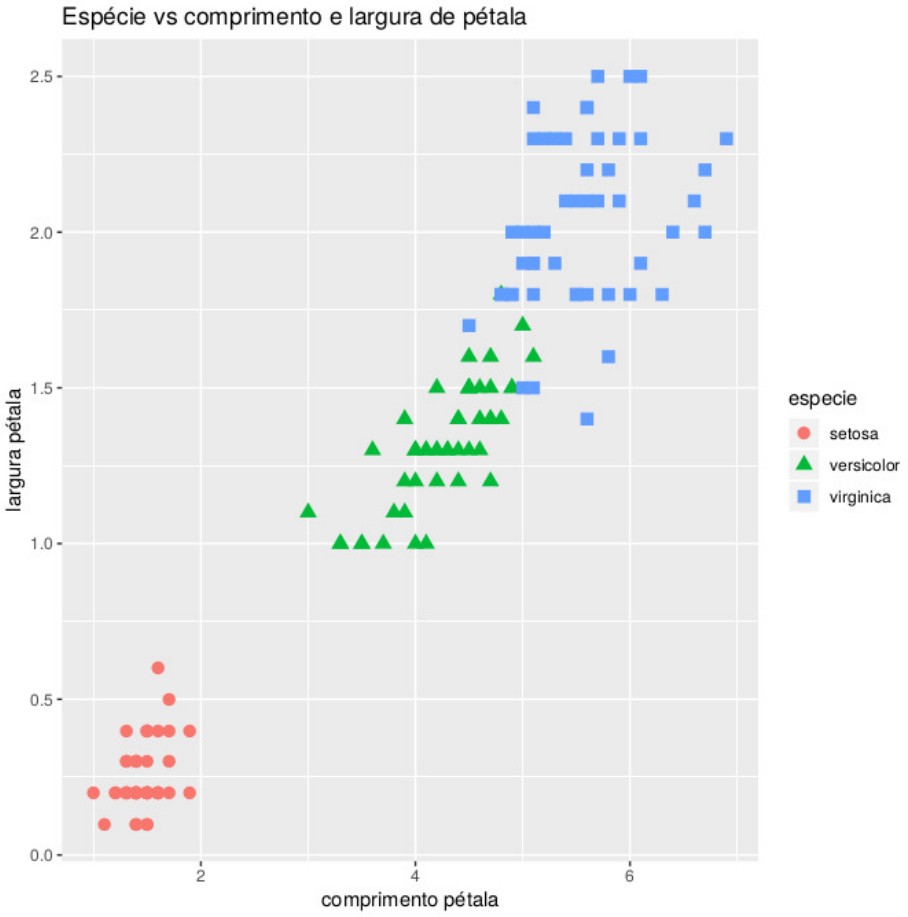

Figura 2. Podemos ver que, tanto no atributo comprimento pétala quanto em largura pétala, não há muita interseção entre os valores

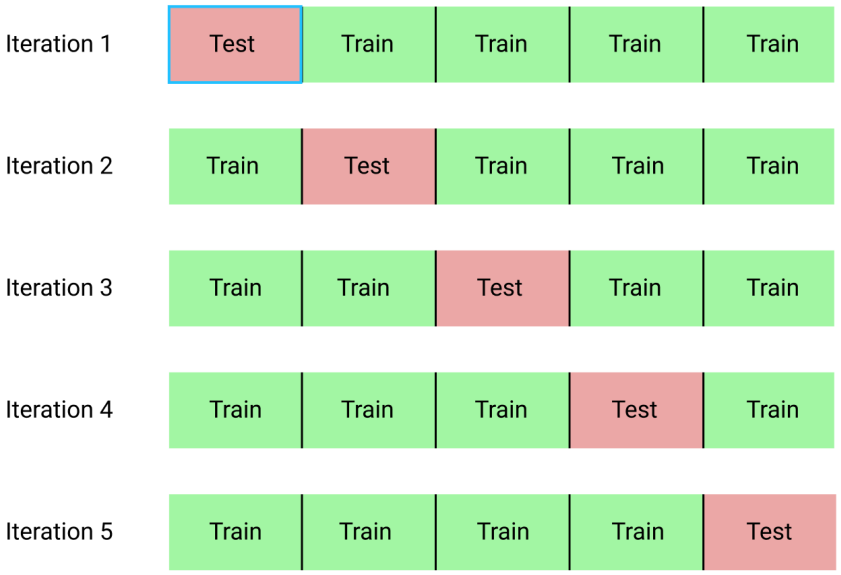

Figura 3. Processo de valicação cruzada 5-fold

Perceptron

Como dito anteriormente, por causa da diferença no número de classes em cada dataset duas técnicas diferentes serão usadas para implementar o algoritmo de perceptron, uma para cada dataset.

O perceptron funciona como um neurônio de uma rede neural. Ele recebe como entrada um conjunto de treino $D = (x_1, d_1, ..., (x_m, d_m)$, onde x_j é um vetor contendo observações de n atributos e d_j é o valor de label esperado como output para esses valores de x_j. Também é passado como entrada o número de épocas e, que é o número de vezes que se deseja iterar as etapas de treino sobre todas as m observações. O output do perceptron é a previsão dele para o label da classe para cada uma das m observações.

Antes de realizar os cálculos sobre os inputs, um vetor de pesos w e um vetor de erros err são inicializados com todas as suas entradas igual a zero. O vetor de pesos será instanciado com n entradas e o de erros com e entradas. Também é instanciado um vetor de previsões de saídas y com m entradas.

Os seguintes passos serão realizados e vezes:

– Para cada elemento j no nosso conjunto de treino D faça:

1. Use a função de ativação Heaviside para fazer previsão da label: $z = \sum_{n=2}^{m}(w_n \times x_{j,n-1}) + w_1$
   $y_j = \begin{cases} -1, & z < 0 \\ 1, & z \geq 0 \end{cases}$
2. Atualize os pesos: $w_i = (d_j - y_j) \times x_{j,i}$ para $i = 0, 1, ..., n$
3. Se $d_j \neq y_j$ atualize os erros: $err_k = err_k + 1$

Como o dataset Salary [2] possui duas classes (0 e 1), só precisamos fazer classificação binária. Logo, para cada uma das 5 rodadas de validação cruzada 5-fold, um conjunto de treino diferente e o número de épocas é dado como entrada.

Já o dataset Iris [1] possui três classes (setosa, versicolour e virginica), logo, é necessário fazer classificação binária três vezes, uma isolando cada uma das classes. Essa estratégia se chama "One vs All".

Resultados e Discussão

Ao executar o perceptron para o dataset Salary [2], 5 vezes na validação cruzada, por 50 épocas em cada rodada, conseguimos os seguintes pesos:

Tabela 2. Pesos dos atributos por rodada da validação cruzada 5-fold.

Rodada	rk	yr	dg	yd	sl
1	216	-1928	116	4418	12174
2	-44	-1218	-2	1960	15646
3	200	-2696	-100	5396	2500
4	100	-800	-200	5400	3200
5	-68	-3124	-64	2280	2764

O que podemos entender por um atributo com médio dos pesos correspondentes negativos é que, ao tentar dividir as classes no plano bidimensional, na maioria das vezes que uma classe que estava com o label negativo era dada a previsão de um label positivo a observação desse atributo tinha um valor alto. O raciocínio é análogo para atributos com média de pesos correspondentes positivos.

Pela Tabela 2 podemos ver que, quando um homem (inicializado com label -1) era classificado incorretamente, na maioria das vezes ele estava há muito tempo no seu ranking atual, o que nos indica que, em média, as mulheres estão há mais tempo no seu ranking atual do que os homens desse dataset, o que mostra que as mulheres estão há mais tempo sem ser promovidas.

Também podemos ver que, quando uma mulher (inicializado com label 1) era classificada incorretamente, na maioria das vezes ela estava com um salário alto e havia muito tempo que tinham atingido seu grau mais alto, o que nos indica que, em média, os homens no dataset recebem mais que as mulheres e atingiram o seu grau mais alto há mais tempo.

Pelo dataset podemos ver que ambas essas afirmações estão corretas.

Nessa mesma vez em que o perceptron foi executado, a média de acerto foi aproximadamente 61%. Como uma taxa de aprendizado não é necessária para o perceptron e a taxa de erro permaneceu relativamente estável durante as épocas, pode se concluir que os dados não são linearmente separáveis.

Não consegui terminar de implementar o algoritmo que executaria o perceptron para o dataset Iris [1], porém, intuitivamente, pela análise inicial visual feita imagino que os resultados seriam melhores. O dataset não só possui menos atributos como os dados parecem ser mais linearmente separáveis.

Conclusão

O perceptron é um modelo muito interessante porém parece ter algumas limitações. Uma é a necessidade de que o dataset seja linearmente independente. A forma mais intuitiva de resolver esse problema, conhecendo o perceptron, é aplicá-lo em camadas, chamando-o recursivamente e fazendo back propagation dos resultados, porém essa implementação é bem mais complexa.

Considerei um trabalho desafiador e tive que estudar bastante para realizá-lo. Após completar a estratégia "One vs All"para classificação de mais de duas classes pretendo implementar outras estratégias para aplicar nesses mesmos datasets.

Referências

1. Iris Plants Database
   `https://archive.ics.uci.edu/ml/machine-learning-databases/iris/`
2. Discrimination in Salaries
   `http://data.princeton.edu/wws509/datasets/#salary`
3. Código do Perceptron
   `https://github.com/danielatk/perceptron`