INTELIGÊNCIA COMPUTACIONAL II

Prof. Carlos Eduardo Pedreira
PESC/COPPE/UFRJ
Prof[a]. Carolina Marcelino
DCC/UFRJ

Trabalho Prático I

Data de Entrega: 15/05/2020

O Trabalho Prático I visa implementar alguns dos exercícios computacionais descritos no **Homework #1** e no **Homework #2** disponibilizados em:
https://work.caltech.edu/homework/hw1.pdf
https://work.caltech.edu/homework/hw2.pdf

Você deverá implementar os exercícios computacionais na linguagem de sua escolha (preferencialmente em Python ou Matlab). Um relatório prático sucinto deve ser escrito e entregue com o seguinte conteúdo:
- Você deve informar o seu nome no topo do relatório (primeira informação);
- Uma breve introdução sobre o assunto;
- O código fonte da implementação devidamente comentado;
- Resultados alcançados e sua interpretação dos mesmos.

* O código deverá ser apresentado e executado em tempo oportuno, para fins de avaliação, com possível arguição do mesmo. A ser agendado.

Descrição das tarefas

*Uma tradução livre dos enunciados presentes em ambos **Homeworks** foi gerada com algumas alterações. Caso ocorram dúvidas na interpretação recorram aos textos originais em inglês.

O Algoritmo de Aprendizagem Perceptron

Neste problema, você vai criar a sua função *target* (alvo) f e o conjunto de dados D para verificar como o Algoritmo de Aprendizagem Perceptron trabalha. Utilize $d = 2$ para você visualizar o problema, e assuma $\mathcal{X}$ = [-1,1] $\times$ [-1,1] com probabilidade uniforme de seleção $\mathbf{x} \in \mathcal{X}$.
Em cada execução, escolha uma reta aleatória no plano como sua função *target* f (faça isso selecionando dois pontos aleatórios, uniformemente distribuídos em [-1,1] $\times$ [-1,1] e gerando a reta que passa entre eles) na qual de um lado a reta mapeia +1 e do outro -1. Escolha as entradas $\mathbf{x_n}$ de uma base de dados de pontos aleatórios (uniformemente em $\mathcal{X}$), e avalie a função *target* em cada $\mathbf{x_n}$ e obtenha a correspondente saída y_n.
Para cada execução use o Algoritmo de Aprendizagem Perceptron (PLA) para encontrar g. Inicie o PLA como o vetor de pesos $\mathbf{w}$ zerado (todos os pesos iguais a zero) e em cada iteração o algoritmo deverá escolher um ponto aleatório a partir do conjunto de pontos classificados incorretamente. Nós estamos interessados

em dois valores: número de iterações que o PLA precisa para convergir a g, e no erro fora da amostra, ou seja, a divergência entre f e g na qual $\mathbb{P}[f(\mathbf{x}) \neq g(\mathbf{x})]$ (a probabilidade que f e g vão divergir na classificação de um ponto aleatório). Você pode calcular exatamente esta probabilidade ou você pode gerar uma aproximação gerando uma grande quantidade de conjuntos separados de pontos para estimar o erro fora da amostra.

* Para avaliar o erro fora da amostra E_{out} de g gere 1000 novos pontos aleatórios uniformes em $\mathcal{X}$ de acordo com a função *target* utilizada;
A fim de obter uma estimativa confiável para estes dois valores, você deve repetir o experimento por 1000 execuções (cada uma como foi especificado acima) e tomar a média sobre estas execuções.

1) Para N = 10. Em média quantas iterações são necessárias para que o PLA convirja para N = 10 pontos treinados? Apresente o valor aproximado de seu resultado (resultado próximo a media | sua resposta - opção | é próxima de 0).
a) 1 b) 15 c) 300 d) 5000 e) 10000

2) Qual a opção mais se aproxima de $\mathbb{P}[f(\mathbf{x}) \neq g(\mathbf{x})]$ para N = 10;
a) 0.001 b) 0.01 c) 0.1 d) 0.5 e)0.8

3) Agora, teste N = 100. Em média quantas iterações são necessárias para que o PLA convirja para N = 100 pontos de treinamento? Informe o valor mais próximo ao seu resultado.
a) 50 b) 100 c) 500 d) 1000 e)5000

4) Qual a opção mais se aproxima de $\mathbb{P}[f(\mathbf{x}) \neq g(\mathbf{x})]$ para N = 100;
a) 0.001 b) 0.01 c) 0.1 d) 0.5 e)0.8

5) Acrescente ao seu relatório o gráfico do tipo *scatter* com os pontos utilizados para calcular o E_{out} e as retas correspondentes a função target e à hipótese g encontrada;

Regressão Linear

Nestes problemas nós vamos explorar como a Regressão Linear para classificação trabalha. Da mesma maneira com uso do Algoritmo de Aprendizagem Perceptron no **Homework #1**, você vai criar a sua própria função *target* (alvo) f e o conjunto de dados D. Utilize $d = 2$ para que você possa visualizar o problema, e assuma $\mathcal{X}$ = [-1,1] × [-1,1] com probabilidade uniforme de selecionar cada $\mathbf{x} \in \mathcal{X}$. Em cada execução, escolha uma reta aleatória no plano como sua função *target* f (faça isso selecionando dois pontos aleatórios, uniformemente distribuídos em [-1,1] × [-1,1] e gerando a reta que passa entre eles), de forma que a reta mapeie +1 por um lado e -1 pelo outro. Escolha as entradas $\mathbf{x_n}$ do conjunto de dados de pontos aleatórios (uniformemente em $\mathcal{X}$), e avalie a função *target* em cada $\mathbf{x_n}$ para encontrar a saída correspondente y_n.

6) Utilize N = 100. Use Regressão Linear para encontrar g e avaliar E_{in}, a fração de pontos dentro da amostra que foram classificados incorretamente. Repita

o experimento 1000 vezes e use o valor médio (guarde as g's que serão usadas novamente no Problema 7). Qual é o valor médio aproximado de E_{in}? (aproximado é a opção que faz a expressão | sua resposta - dada opção| próxima a 0. Use esta definição aqui e sempre).

a) 0 b) 0.001 c) 0.01 d) 0.1 e) 0.5

7) Agora, gere 1000 novos pontos e os use para estimar o erro fora da amostra E_{out} de g que você fez no Problema 6 (número de pontos classificados incorretamente/ número total de pontos fora da amostra). Novamente, execute o experimento 1000 vezes e guarde a média. Qual é o valor médio aproximado de E_{out}?

a) 0 b) 0.001 c) 0.01 d) 0.1 e) 0.5

8) Agora, utilize N = 10. Posteriormente, procurando os pesos usando Regressão Linear, os use como um vetor de pesos inicial para o Algoritmo de Aprendizagem Perceptron. Execute PLA até que convirja para o vetor final de pesos que separe completamente todos os pontos dentro da amostra. Entre as opções abaixo, qual é o valor mais próximo do número médio de iterações (mais de 1000 execuções) que o PLA leva para convergir? (Quando estiver implementando o PLA, escolha um ponto aleatório para o conjunto classificado incorretamente para cada iteração).

a) 1 b) 15 c) 300 d) 5000 e) 10000

9) Neste exercício iremos avaliar o desempenho da versão *pocket* do PLA em um conjunto de dados que não é linearmente separável. Para criar este conjunto, gere um ruído simulado selecionando aleatoriamente 10% do conjunto de treinando (de tamanho N_1) e inverta os rótulos dos pontos selecionados. Em seguida, implemente a versão *pocket* do PLA e avalie o E_{out} médio (1000 execuções) após i iteraçoes em um conjunto gerado aleatoriamente em $\mathcal{X}$ de tamanho N_2 de acordo com as seguintes configurações: (Não esqueça de gerar gráficos comparando a hipótese g encontrada com a função *target* utilizada)

a) Inicializando os pesos com 0, $i = 10$, $N_1 = 100$ e $N_2 = 1000$
b) Inicializando os pesos com 0, $i = 50$, $N_1 = 100$ e $N_2 = 1000$
c) Inicializando os pesos utilizando Regressão Linear, $i = 10$, $N_1 = 100$ e $N_2 = 1000$
d) Inicializando os pesos utilizando Regressão Linear, $i = 50$, $N_1 = 100$ e $N_2 = 1000$

Regressão Não-Linear

Nestes problemas, nós vamos novamente aplicar Regressão Linear para classificação. Considere a função *target*:

$f(x_1, x_2) = sign(x_1^2 + x_2^2 - 0.6)$.

Gere um conjunto de treinamento de N = 1000 pontos em $\mathcal{X}$ = [-1,1]×[-1,1] com probabilidade uniforme escolhendo cada $\mathbf{x} \in \mathcal{X}$. Gere um ruído simulado selecionando aleatoriamente 10% do conjunto de treinamento gerado e trocando o sinal de saída.

10) Execute a Regressão Linear sem transformação usando o vetor de atributos:

$(1, x_1, x_2)$,

para encontrar o peso $\mathbf{w}$. Qual é o valor aproximado de classificação do erro E_{in} dentro da amostra? (Execute o experimento 100 vezes e use o valor médio de E_{in} para reduzir a variação nos seus resultados.)

a) 0 b) 0.1 c) 0.3 d) 0.5 e) 0.8

11) Agora, transforme os N = 1000 dados de treinamento seguindo o vetor de atributos não-linear:

$(1, x_1, x_2, x_1x_2, x_1^2, x_2^2)$.

Encontre o vetor $\tilde{\mathbf{w}}$ que corresponde a solução da regressão linear. Quais das hipóteses a seguir é a mais próxima que você encontrou? Neste caso, próximo significa o valor que mais entra em acordo com sua hipótese (existe uma alta probabilidade de estar acordando com um ponto aleatoriamente selecionado). Em média algumas execuções serão necessárias para assegurar uma resposta estável.

a) $g(x_1, x_2) = sign(-1 - 0.05x_1 + 0.08x_2 + 0.13x_1x_2 + 1.5x_1^2 + 1.5x_2^2)$
b) $g(x_1, x_2) = sign(-1 - 0.05x_1 + 0.08x_2 + 0.13x_1x_2 + 1.5x_1^2 + 15x_2^2)$
c) $g(x_1, x_2) = sign(-1 - 0.05x_1 + 0.08x_2 + 0.13x_1x_2 + 15x_1^2 + 1.5x_2^2)$
d) $g(x_1, x_2) = sign(-1 - 1.5x_1 + 0.08x_2 + 0.13x_1x_2 + 0.05x_1^2 + 0.05x_2^2)$
e) $g(x_1, x_2) = sign(-1 - 0.05x_1 + 0.08x_2 + 1.5x_1x_2 + 0.15x_1^2 + 0.15x_2^2)$

12) Qual o valor mais próximo do erro de classificação fora da amostra E_{out} de sua hipótese no Problema 11? (Estime isso gerando um novo conjunto de 1000 pontos e adicione ruído, como antes. Em média 1000 execuções reduzem a variação em seus resultados).
a) 0 b) 0.1 c) 0.3 d) 0.5 e) 0.8