Universidade Federal de São Carlos
Aprendizado de Máquina
Prof. Tiago A. Almeida `<talmeida@ufscar.br>`

Exercício Prático 2: Regressão Linear

Introdução

Neste exercício você implementará o método de regressão linear univariada e verá como ele utiliza os dados de treinamento para ajustar o modelo de regressão a fim de fazer predições para amostras não vistas. Antes de começar este exercício, é recomendável que você revise os conceitos apresentados em aula.

Arquivos incluídos neste exercício

`ex02.m` – Script geral do exercício
`ex02Dados.txt` – Base de dados de treinamento
[$\star$] `plotarDados.m` – Função para plotar e visualizar os dados da base de treinamento
[$\star$] `computarCusto.m` – Função para calcular a função de custo (J) para um dado θ
[$\star$] `gradienteDescente.m` – Função para computar o gradiente descendente e ajustar/otimizar os parâmetros θ_j

$\star$ indica os arquivos que você precisará completar.

O arquivo `ex02.m` conduzirá todo o processo desse exercício.

O Problema

Neste exercício, você irá implementar o método de regressão linear com uma variável para prever o resultado orçamentário mensal de uma cidade. Suponha que Governo Federal deseja estimar o valor (em reais) que sobrará no final de cada mês no caixa de uma prefeitura levando-se em conta o número de habitantes residentes da cidade. Para isso, funcionários do governo coletaram vários resultados médios obtidos para diversas cidades e armazenaram na base de dados `ex02Dados.txt`.

Você foi contratado para desenvolver um método que ofereça uma boa previsão do resultado orçamentário mensal de uma cidade qualquer baseado apenas no tamanho da sua população (número de habitantes).

O arquivo `ex02Dados.txt` contém o conjunto de dados que deverá ser utilizado para o problema da regressão linear. A primeira coluna contém o tamanho da população da cidade (x 10.000 habitantes) e segunda coluna corresponde ao resultado orçamentário mensal médio da cidade (x R$ 100.000,00). Um valor negativo indica que os recursos provenientes pelo Governo Federal foram insuficientes para cobrir todos os gastos.

O *script* `ex02.m` já está configurado para carregar esses dados.

Visualização dos Dados

Antes de iniciar qualquer tarefa, muitas vezes é útil visualizar os dados para melhor compreendê-los. Para o conjunto de dados oferecido, você pode usar um gráfico de dispersão 2D para visualizá-los, já que a base é composta por amostras com apenas dois atributos.

Em `ex02.m`, o conjunto de dados é carregado a partir do arquivo de dados para as variáveis X e y.

A seguir, a rotina chama a função `plotarDados` para criar um gráfico de dispersão dos dados. Seu trabalho é completar o arquivo `plotarDados.m` para gerar um gráfico similar ao apresentado na Figura 1.

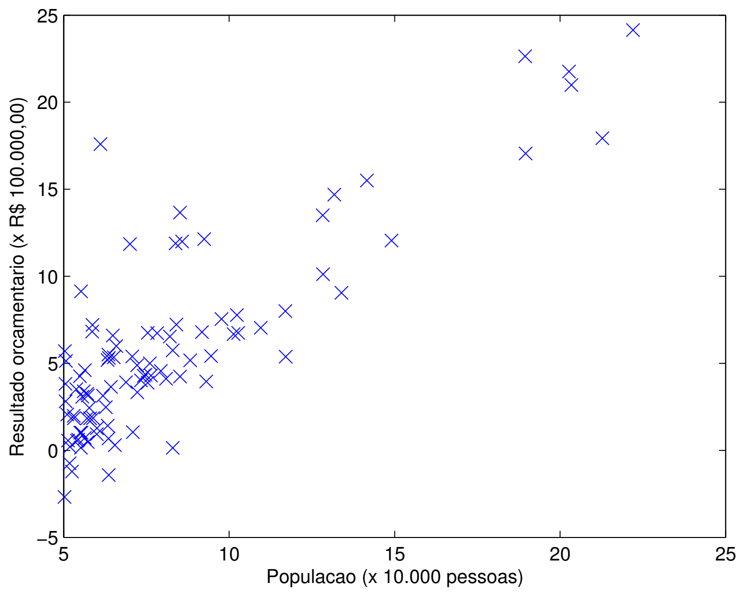

Figura 1: Visualização dos dados obtida a partir de `plotarDados.m`

Você precisará completar a função ***plotarDados.m****.*

Gradiente Descendente

Nesta parte, você usará o método do gradiente descendente para ajustar os parâmetros da regressão linear θ para conjunto de dados de treinamento.

Equações de ajuste

O objetivo da regressão linear é minimizar a função custo

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^i) - y^i)^2),$$

onde a hipótese $h_\theta(x)$ é determinada pelo modelo linear

$$h_\theta(x) = \theta_0 + \theta_1 x_1.$$

Os parâmetros do seu modelo são os valores θ_j que você precisará ajustar para minimizar o custo $J(\theta)$. Uma maneira de fazer isso é utilizando o algoritmo iterativo do gradiente descendente. Em tal método, cada iteração realiza o ajuste simultâneo de θ_j para todo j:

$$\theta_j = \theta_j - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^i) - y^i)x_j^i).$$

A cada passo (iteração) do gradiente descendente, os parâmetros θ_j se aproximam de valores ótimos para obter o menor custo $J(\theta)$.

É importante observar que, para levar em conta o parâmetro θ_0, foi inserida uma coluna adicional de 1's em X (primeira coluna) correspondendo a x_0. Isso permite tratar θ_0 como outro atributo.

Na rotina `ex02.m`, os dados para a regressão linear já estão previamente configurados. Foi acrescentada uma dimensão extra aos dados para acomodar o termo θ_0. Além disso, foram inicializados os parâmetros θ com um vetor de zeros e a taxa de aprendizagem $\alpha = 0.01$.

Função Custo $J(\theta)$

Nesta seção, você precisará implementar a função custo $J(\theta)$ para poder checar a convergência do método do gradiente descendente. No processo de validação do gradiente descendente, muitas vezes, é útil monitorar a convergência do método usando um gráfico de custo ($J(\theta)$) pelo número de iterações. O valor da função $J(\theta)$ não pode aumentar no decorrer das iterações. Se isso ocorrer, é provável que a taxa de aprendizado α esteja muito alta.

A sua próxima tarefa é completar o código do arquivo `computarCusto.m`. Para tal, lembre-se de que as variáveis X e y não são valores escalares, mas matrizes cujas linhas representam as amostras do conjunto de treinamento.

Você precisará completar a função ***computarCusto.m****.*

Se a sua implementação estiver correta, é esperado que seja exibido um valor de custo aproximadamente igual à **32.07**.

Gradiente Descendente

Em seguida, você precisará completar o método do gradiente descendente presente no arquivo `gradienteDescente.m`. A estrutura do *loop* já está pronta, e portanto, você precisará apenas escrever os comandos referentes às atualizações dos valores de θ.

Enquanto programa, certifique-se de que você entende o que está sendo otimizado e atualizado. Tenha em mente que o custo $J(\theta)$ é parametrizado pelo vetor θ, e não por X e y. Consequentemente, para minimizar o valor de $J(\theta)$ é necessário ajustar exclusivamente os valores dos parâmetros θ.

Uma boa maneira de verificar se o método do gradiente está funcionando corretamente é observar o valor de $J(\theta)$ e certificar-se de que ele diminui a cada passo. O código inicial presente no arquivo `gradienteDescente.m` chama a função `computarCusto` em cada iteração e armazena o valor da função custo. Se você tiver implementado o gradiente descendente e a função custo corretamente, o valor de $J(\theta)$ nunca deve aumentar, mas deve convergir para um valor até o final da execução algoritmo.

Você precisará completar a função ***gradienteDescente.m****.*

Se você completou os arquivos corretamente, é esperado que seja plotado um regressor similar ao apresentado na Figura 2 e o gráfico de variação da função custo similar ao da Figura 3.

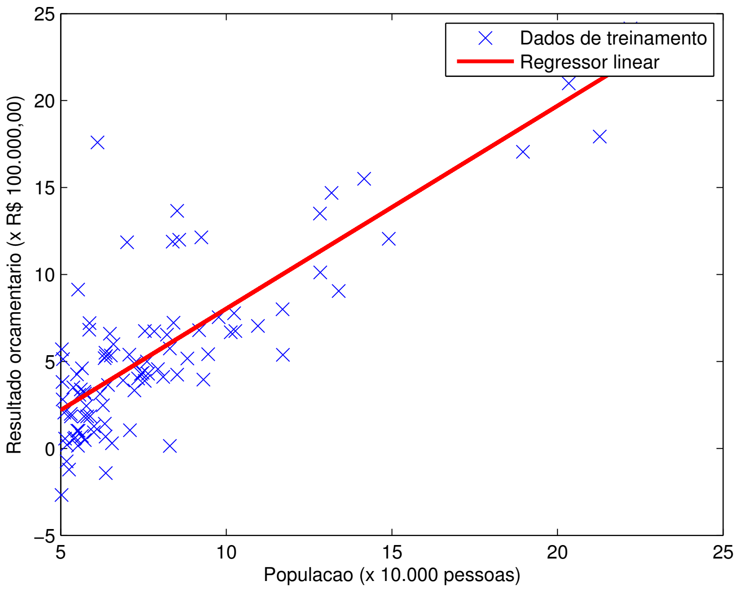

Figura 2: Visualização dos dados e do regressor enconcontrado

Os valores finais obtidos para θ são usados para fazer previsões sobre o resultado orçamentário mensal de prefeituras de cidades com 40.000 e 80.000 habitantes, respectivamente. Espera-se que os valores previstos pelo seu algoritmo seja aproximadamente igual à R$ 103.515,79 e R$ 570.060,73, para cada uma das prefeituras, respectivamente.

Visualizando $J(\theta)$

Para ajudar a entender melhor a função custo $J(\theta)$, é oferecido um gráfico 3D com o valor de $J(\theta)$ com relação aos parâmetros θ_0 e θ_1. Se o seu código estiver correto, é esperado

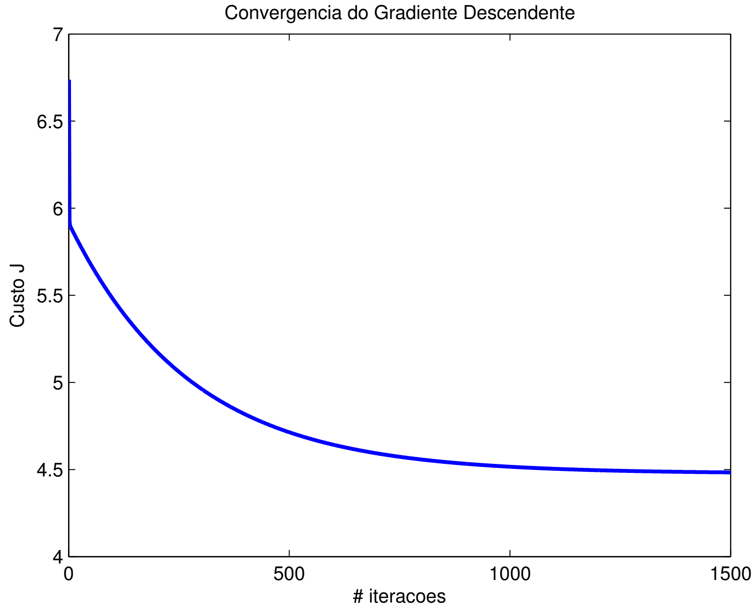

Figura 3: Convergência do método do gradiente descendente – variação de $J(\theta)$

que você visualize o gráfico apresentado na Figura 4.

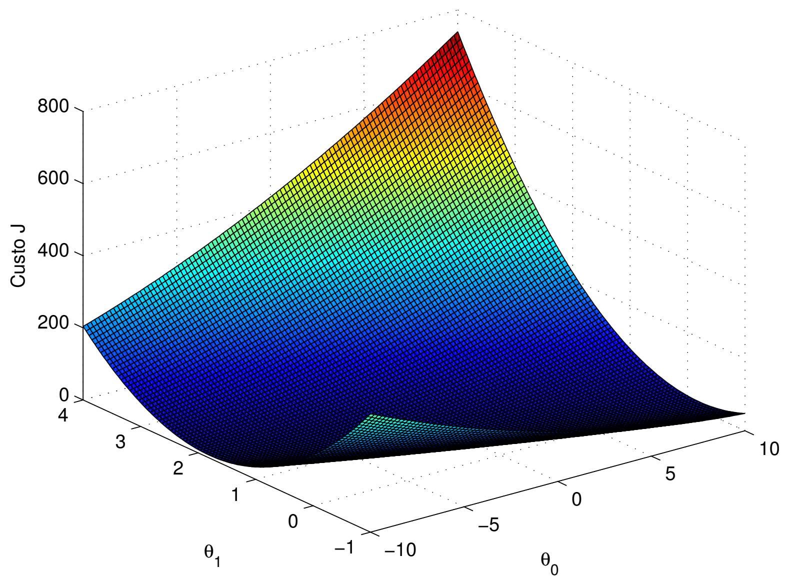

Figura 4: Valor de $J(\theta)$ em função de θ_0 e θ_1

A seguir, são apresentadas as linhas de contorno referentes a Figura 4 e o ponto ótimo (θ_0 e θ_1) encontrado pelo método do gradiente descendente, que representa o mínimo da função custo $J(\theta)$ (Figura 5).

Como $J(\theta)$ é uma função convexa, é possível garantir que os valores de θ encontrados pelo método do gradiente descendente representam o mínimo global para $J(\theta)$.

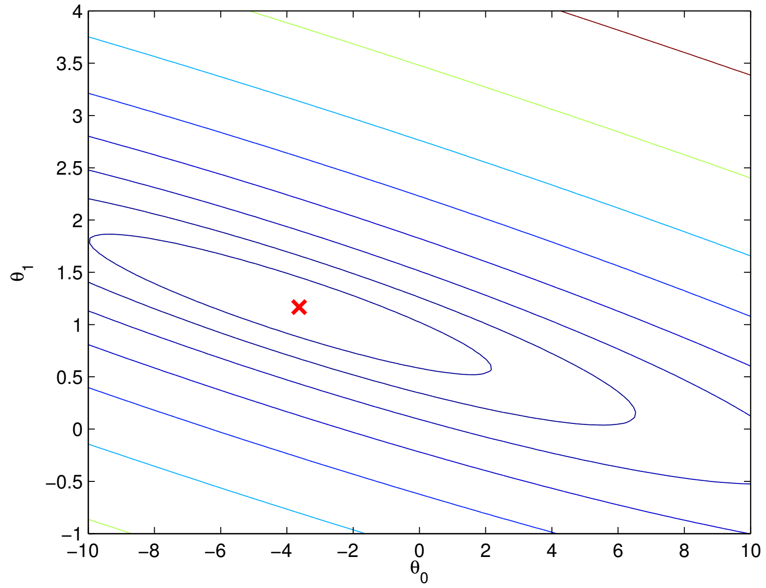

Figura 5: Linhas de contorno do valor de $J(\theta)$ em função de θ_0 e θ_1

Fazendo previsões

Na última etapa, o programa `ex02.m` entra em modo de previsões. É solicitado o tamanho da população da cidade ou `-1` para encerrar a execução. No caso, para uma cidade com 50.000 habitantes (entrada igual à 5) é esperado um resultado orçamentário mensal de aproximadamente R$ 220.152,03.