# Regressão Linear

Cesar Bueno Vilela Silveira
145715
barddes@gmail.com

Seong Eun Kim
177143
s177143@g.unicamp.br

## I. Introdução

Este trabalho teve como objetivo estudar a regressão linear como modelo para resolver problemas simples. Para isso, foi implementado um algoritmo que prevê o ano de lançamento de uma música a partir da média dos seus timbres e da covariância deles. As músicas são na maioria de comerciais ocidentais de 1922 a 2011, com o pico no ano de 2000. Para a implementação do algoritmo foi usado o software `Octave (v4.0.3)`

## II. Atividades

Nosso *dataset* de treino é formado de 463715 músicas, e o dataset de teste é formado por 36285 músicas. Estes dados são um subconjuto do `Million Song Dataset`. Cada linha do conjunto de dados representa uma música diferente, e contém o ano de lançamento e as 90 features (12 médias de timbres e 78 covariâncias de timbre).

Escrevemos um script para calcular a regressão linear com os dados de treino e obter uma boa função que relacione as features e estime o ano de lançamento (treinávamos os coeficientes $\theta$s). Depois, testamos a função obtida com o conjunto de dados de teste. Durante o projeto, realizamos algumas regressões de maior complexidade (que apresentavam diferentes números de features). Em algumas regressões, além das 90 features originais de cada dado, adicionamos outras 540 features adicionais, que foram obtidas multiplicando algumas das 90 features originais entre si.

Primeiramente, iniciamos importando o *dataset* de treino. Destes dados, obtivemos o vetor $y$ de anos e a matriz $x$ das features. Incluímos uma coluna adicional à matriz $x$ para armazenarmos $x_0 = 1$, relativo ao termo independente de nossa regressão. Fizemos então a regressão linear e posteriormente, multiplicamos algumas de nossas 90 features entre si para conseguir novas features. Começamos elevando todas as 90 features ao quadrado e ao cubo. Depois, apenas com as 12 medias de timbre, multiplicamos pares e trios de features entre si para obtermos outras features. Ao final disso, concatenamos diferentes combinações destas novas features a nossa matriz $x$.

Agora com nossa matriz $x$ de features completa, realizamos uma normalização destes dados. Normalizá-los foi bem importante para o projeto visto que as features estão em ordens de gradeza bem diferentes ($x_0$ não foi normalizado e continua com valor 1).

Com nossas features prontas, fizemos a descida de gradiente e resolvemos a equação normal. Para a Descida de Gradiente, inicializamos nossos $\theta$s com 0.

O *learning rate* $\alpha$ é um numero que representa o tamanho de nosso passo no sentido do gradiente. Um $\alpha$ pequeno demais faz com que os $\theta$s demorem a convergir para a melhor solução. Um $\alpha$ muito grande faz com que os $\theta$s passem da melhor solução, fazendo com que o Custo aumente exponencialmente. Afim de escolher um bom valor para $\alpha$, analisamos o comportamento do Custo para diferentes valores de $\alpha$:

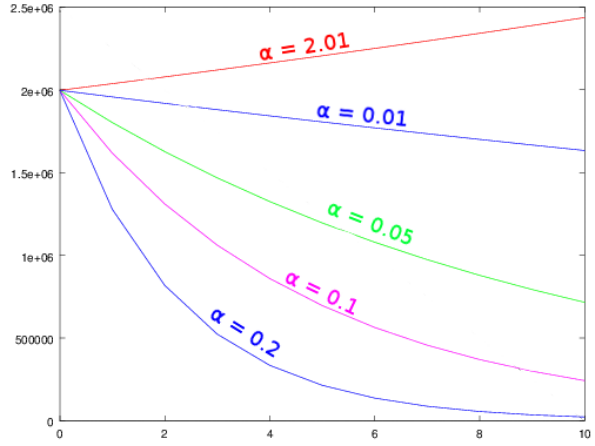


Figura 1. Gráfico do custo em relação ao número de iterações para *learning rates* diferentes.

Analisando estes resultados, verificamos que um bom valor para o *learning rate* é $\alpha = 0.2$.

Com as 90 features originais, nosso custo $J$ passou de quase 2000000 para 61.7 em 30 iterações. Após 2000 iterações, o Custo abaixou para 47.5.

Adicionamos as features ao quadrado em nossa regressão. Reduzimos o custo para para 61.7 em 30 iterações.

Adicionamos as combinações de dois a dois das features multiplicados ao modelo anterior. Reduzimos o custo para 60.4 em 30 iterações. Adicionamos as features ao cubo ao modelo anterior. Reduzimos o custo para 60.4 em 30 iterações.

Devido a limitações de memoria do computador, não pudemos adicionar a combinação três a três das features multiplicadas.

Para a resolução da equação normal, tivemos de resolver o sistema de equações:

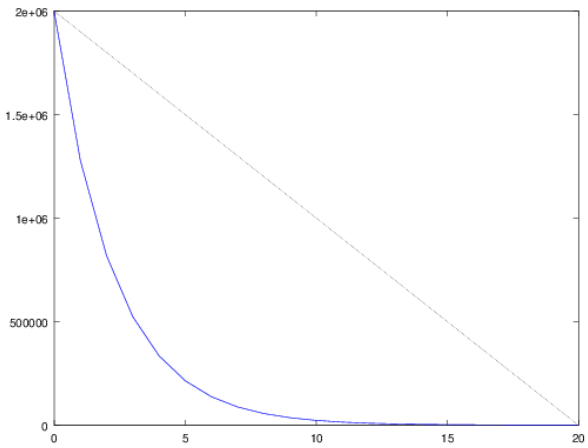


Figura 2. Gráfico do custo em relação ao número de iterações do modelo com a covariância dos timbres ao cubo e combinações dois a dois.

$$\theta = (X^T * X)^{-1} * X^T * Y$$

Essa equação nos deu o Custo mínimo possível para nossos dados.

Além do Custo, existem outras formas de mensurar a qualidade de nossa solução. Uma bem famosa é o Coeficiente de Determinação, tambem conhecido por $R^2$. Este coeficiente é um valor numerico que varia de $-\infty$ até 1, e é calculado pela formula:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

onde $SS_{res}$ é a somatória dos quadrados dos erros, e $SS_{tot}$ é proporcional à variância dos dados [1].

Seguindo esta formula, nosso score foi de $0.27$ para nosso melhor modelo.

## III. Experimentos e Discussões

Primeiramente, fizemos a regressão linear no conjunto de dados de treino. Porém, este apresentava grande variação nas suas ordens de grandeza, o que afetava os valores dos $\theta$s e fazia com que o *learning rate* $\alpha$ fosse muito pequeno (na ordem de $10^{-9}$). Consequentemente, a convergência era muito lenta. Por essa razão, normalizamos os valores, e obtivemos um custo $J$ da regressão linear de $J = 45.629$ no teste e $J = 45.628$ nos treinos quando o $\alpha = 0.2$.

Então, decidimos aumentar a complexidade na esperança de diminuir o valor do custo. Elevamos os dados ao quadrado, e o custo foi para $J = 44.139$ no treino e para $J = 44.610$ no teste. Depois, elevamos os dados ao cubo, e o custo foi para $J = 43.507$ no treino e $J = 44.226$ no teste. Em seguida, adicionamos combinações de dois a dois dos valores da média dos timbres, e obtivemos $J = 42.424$ no treino e $J = 43.257$ no teste. Ao adicionarmos combinações três a três, estoramos a memória do computador. Por último, acidentalmente removemos a potência cúbica das médias dos timbres (12 primeiros valores) e obtivemos $J = 42.511$ no treino e $J = 42.100$ no teste, que representa nosso melhor modelo. Nele, calculamos o coeficiente de determinação $R^2$, que foi igual a $R^2 = 0.27$. Comparando o último modelo ao penúltimo testado, julgamos que diminuímos o overfitting dos valores, pois houve uma diminuição do custo do teste.

Nos vários modelos feitos, o custo do teste e do treino foram similares, mostrando que não houve overfitting. Entretanto, ambos os custos tiveram valores relativamente altos, e o coeficiente de determinação no nosso melhor caso não foi próximo de 1, o que indica um possível underfitting.

Além disso, pudemos observar, para cada experimento feito, como o *learning rate* era de extrema importância para o alcance dos nossos objetivos. Tínhamos que ter cautela para não usar um valor muito alto, de forma que aumentasse o custo, ou muito baixo, de forma que demorasse muito para convergir.

## IV. Conclusões e Próximos passos

Nesse experimento, pudemos estudar a regressão linear e alternativas baseadas na regressão na solução de problemas - nesse caso em particular, predizer o ano de lançamento de uma música a partir da média de timbres e de suas covariâncias. refinar as features, selecionando quais sao uteis e quais nao sao aplicar logaritimos

## Referências

[1] Coefficient of determination. Wikipedia.