Predicting the number of shares of a social network with linear regression

Gustavo Ciotto Pinton, RA117136*

Abstract

Baseando-se em dados reais da rede social Mashable [1], este relatório utiliza técnicas de regressão linear para predizer o número de compartilhamentos que uma eventual publicação teria a partir dos seus atributos. São exploradas técnicas de regularização e seleção de variáveis, e discutidas as escolhas de alguns parâmetros, tal como a taxa de aprendizado e a complexidade dos modelos propostos. A regressão linear foi obtida por dois métodos diferentes, sendo eles gradient descent e mínimos quadrados, cujos resultados são comparados posteriormente.

1. Introdução

A regressão linear consiste em uma técnica simples cujo objetivo é relacionar um conjunto de dados de entrada com os de saída através do uso de uma equação linear $Y = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + \ldots + \theta_N X_N$, em que Y, X_i e θ_i representam, respectivamente, a saída gerada pelo modelo a partir de um conjunto de atributos e os coeficientes da equação. A equação anterior é utilizada apenas para uma amostra singular, porém, como visto em aula, o principal aspecto que garante a qualidade de resultados obtidos por métodos de aprendizado de máquina é o grande volume de dados. Tendo isso em vista, um conjunto de M amostras de entrada descritas por N atributos é representado pela matriz $\boldsymbol{x}_{N+1\times M}$. Da mesma maneira, os coeficientes da equação são representados pelo vetor $\boldsymbol{\theta}$ de $N+1$ elementos e a saída, pelo vetor $\boldsymbol{y}$ de M valores, de forma a obter a equação vetorial 1, logo abaixo.

$$\boldsymbol{y} = \boldsymbol{\theta}^T \boldsymbol{x} \tag{1}$$

A determinação das componentes de $\boldsymbol{\theta}$ é realizada a partir da minimização da função de custo $J(\boldsymbol{\theta}, \boldsymbol{x})$, descrita pela equação 2, em que $\boldsymbol{x}_i$ corresponde à i-ésima amostra e t_i ao *target* que deseja-se atingir.

$$J(\boldsymbol{\theta}, \boldsymbol{x}) = \frac{1}{2M} \sum_{i=1}^{M} \left(\boldsymbol{\theta}^T \boldsymbol{x}^{(i)} - \boldsymbol{t}^{(i)} \right)^2 \tag{2}$$

Os dados utilizados para a minimização $\boldsymbol{x}$ e $\boldsymbol{t}^{(i)}$ pertencem a um conjunto chamado **conjunto de treinamento**, enquanto que a validação dos parâmetros obtidos ocorre em um conjunto que recebe este mesmo nome. Um regra geral é separar 80% dos dados disponíveis para treinamento e o restante, 20%, para testes e validação.

A função de custo $J(\boldsymbol{\theta}, \boldsymbol{x})$ possui a importante propriedade de ser **convexa** [2]. Isso garante que ela não possui mínimos locais, exceto pelo global. Deste modo, dois métodos de minimização podem ser utilizados. O primeiro, chamado de **equação normal**, permite obter o $\boldsymbol{\theta}$ ótimo a partir de uma única expressão, representada em 3. Observa-se rapidamente que o ponto negativo desta abordagem é o cálculo da matriz inversa, cuja complexidade assimptótica de computação é $\Theta(n^3)$, conforme visto em aula. Para a matrix $\boldsymbol{x}^T\boldsymbol{x}$ de dimensão $N+1 \times N+1$, portanto, o número de atributos N pode representar um impedimento para o uso desta técnica.

$$\boldsymbol{\theta} = \left(\boldsymbol{x}^T \boldsymbol{x}\right)^{-1} \boldsymbol{x}^T \boldsymbol{t} \tag{3}$$

O segundo método, chamado de **gradient descent**, é iterativo e não requer o cálculo de nenhuma matriz inversa. Derivando-se parcialmente a equação 2 em relação a $\boldsymbol{\theta}_j$, obtém-se o coeficiente linear e portanto a direção que devemos tomar para nos aproximar do mínimo global. Repetindo o raciocínio para todos os parâmetros $\boldsymbol{\theta}_j$, obtém-se à operação 4 que deve ser aplicada a cada um deles a cada iteração. Ressalta-se que $\boldsymbol{x}_1 = 1$ para todo i.

$$\boldsymbol{\theta}_j = \boldsymbol{\theta}_j - \alpha \frac{\partial J}{\partial \boldsymbol{\theta}_j} = \boldsymbol{\theta}_j - \frac{\alpha}{M} \sum_{i=1}^{M} \left(\boldsymbol{\theta}^T \boldsymbol{x}^{(i)} - \boldsymbol{t}^{(i)} \right) \boldsymbol{x}_j^{(i)} \tag{4}$$

A equação 4 introduz o parâmetro α, chamado de *learning rate*. Tal parâmetro controla a velocidade de convergência ao mínimo global, isto é, quanto maior o seu valor mais rapidamente ele é atingido. Por outro lado, valores muito grandes produzirão um efeito denominado *overshooting*, isto é, a solução encontrada sempre ficará *balançando* ao redor do mínimo sem nunca atingí-lo. Nas próximas seções, o autor explica a estratégia para a definição desta grandeza para o problema deste relatório.

Tendo visto toda teoria por trás da regressão linear, podemos aplicá-la a dados reais. Neste relatório, usamos dados reais [1] da rede social *Mashable*, que visam relacionar uma série de atributos de uma determinada publicação com o número de *shares* que ela recebeu. Em outras palavras, busca-se encontrar uma relação linear entre o número de *shares* recebidos com os atributos de uma publicação. No total, 31715 dados foram usados para treinamento dos modelos e 7929 para teste e validação.

*Is with the Institute of Computing, University of Campinas (Unicamp).
Contact: gustavociotto@gmail.com

2. Atividades

As próximas subseções visam explicar as escolhas dos diversos parâmetros adotados pelo autor.

2.1. Ajuste do learning rate

Conforme mencionado na seção 1, o parâmetro *learning rate* controla a velocidade de convergência ao mínimo global da função de custo. Tendo em vista as questões levantadas nesta mesma seção, adota-se um valor α inicial igual a 0.5, modificando-o em apenas duas situações:

- Quando o erro da iteração k é superior aquele da iteração $k-1$. Neste caso, conclui-se que a solução encontrada foi pior e, portanto, devemos diminuir a velocidade de convergência.
- A variação do erro entre as iterações k e $k-1$, isto é, $\frac{R_{k-1}-R_k}{R_k}$, foi inferior a uma porcentagem $\Delta = 0.5\%$. Neste caso, deseja-se eliminar os efeitos de *overshooting* e aproximar o melhor possível do mínimo global. Denota-se de R_k, dado pela fórmula 5, o *root-mean-square error* obtido na iteração k:

$$R_k = \sqrt{\frac{\sum_{i=1}^{M}\left(\boldsymbol{\theta}^T\boldsymbol{x}^{(i)} - \boldsymbol{t}^{(i)}\right)^2}{M}} \tag{5}$$

Nestes dois casos, o valor de α é diminuído pela metade.

2.2. Ajuste das variáveis discretas

A descrição dos atributos de cada uma das entradas revelou a presença de algumas variáveis discretas. Dois grupos foram identificados: um com variáveis do tipo *a publicação ocorreu na segunda-feira?*, *a publicação ocorreu na terça-feira?* e assim por diante, e outro com atributos semelhantes a *o canal é de entretenimento?*. Antes de utilizá-las ou não nos modelos propostos a seguir, é preciso verificar se tais variáveis são **ortogonais** entre si, isto é, se a distância entre elas consideradas duas a duas é sempre a mesma. Levando-se em consideração que as variáveis de ambos os grupos nunca podem ser setadas simultaneamente (uma publicação não pode ocorrer na segunda e na terça e ela não pode pertencer a dois canais ao mesmo tempo, por exemplo), observa-se a propriedade de **ortogonalidade** e, portanto, podemos utilizá-las sem quaisquer modificações.

2.3. Normalização dos dados de entrada e de saída

A normalização dos dados antes do treinamento previne que os valores $\boldsymbol{\theta}_j$ sejam extremamente pequenos ou grandes de forma a causar instabilidades ou interferir na velocidade de convergência do modelo. Dentre as muitas maneiras de normalização que podem ser aplicadas, escolhemos para esta atividade aquela da equação 6. Neste caso, são calculados a média μ_j e o desvio padrão σ_j de cada *feature* $\boldsymbol{x}'_j$ sobre todas as amostras de treinamento e depois é aplicada a expressão 6.

$$\boldsymbol{x}'_j = \frac{\boldsymbol{x}_j - \mu_j}{\sigma_j} \tag{6}$$

Vale lembrar que os valores μ_j e σ_j devem ser amarzenados para serem utilizados posteriormente no conjunto de teste e em qualquer outra entrada a ser aplicada no modelo.

2.4. Regularização

A regularização é uma técnica que permite limitar os efeitos negativos de *overfitting* sem a eliminação de nenhum atributo do modelo a partir da penalização dos parâmetros $\boldsymbol{\theta}_j$. Neste caso, a equação 2 torna-se:

$$J(\boldsymbol{\theta}, \boldsymbol{x}) = \frac{1}{2M}\left[\sum_{i=1}^{M}\left(\boldsymbol{\theta}^T\boldsymbol{x}^{(i)} - \boldsymbol{t}^{(i)}\right)^2 + \lambda\sum_{j=2}^{N+1}\boldsymbol{\theta}_j^2\right] \tag{7}$$

Observa-se que se λ é muito grande, então o processo de minimização produzirá $\boldsymbol{\theta}_j$ pequenos, resultando em um modelo polarizado que não seria bem-sucedido em se adequar corretamente aos dados. Em oposição, λ pequenos não conseguiriam combater os efeitos de *overfitting* e a variação criada pelos dados de testes seria grande.

A equação 6, por sua vez, transforma-se na expressão 8, sendo $\boldsymbol{I}_{N\times N}$ a matriz identidade:

$$\boldsymbol{\theta} = \left(\boldsymbol{x}^T\boldsymbol{x} + \lambda\begin{bmatrix}0 & 0\\ 0 & \boldsymbol{I}_{N\times N}\end{bmatrix}\right)^{-1}\boldsymbol{x}^T\boldsymbol{t} \tag{8}$$

Destaca-se que $\boldsymbol{\theta}_1$ não é penalizado.

3. Soluções propostas

Essa seção é dedicada à discussão das soluções propostas. Foram realizados testes com modelos simples, regularizados e contendo variáveis de segunda e terceira ordens.

3.1. Modelo linear simples

Neste modelo, utiliza-se o método de regressão linear sem regularização e sem quaisquer termos de segunda ou terceira ordens. Para a técnica de *gradient descent*, itera-se a fórmula 4 100 vezes no total. A figura 1 representa os erros quadrados médios conforme equação 5 obtidos no treinamento e no teste a cada iteração. Observa-se que ambos os erros se estabilizam em torno de, respectivamente, 10700 e 14700, e que não há a presença dos efeitos negativos de *overfitting*.

A figura 2, por sua vez, representa os resultados encontrados a partir do uso da expressão 6. Dois gráficos são mostrados: no primeiro, compara-se os erros obtidos no treinamento e teste para diversos tamanhos de conjuntos de entrada indo de 10000 até 31715, enquanto que no segundo, compara-se os vetores $\boldsymbol{\theta}$ obtidos pelos dois métodos descritos nessa seção. Conforme esperado, o RMS encontrado para o método normal, em torno de 10580, foi inferior aquele encontrado por *gradient descent*. O erro de teste, por sua vez, também ficou em torno de 14570. Observa-se também que algumas diferenças consideráveis entre os vetores $\boldsymbol{\theta}$ sobretudo para os conjuntos de dados com tamanhos intermediários.

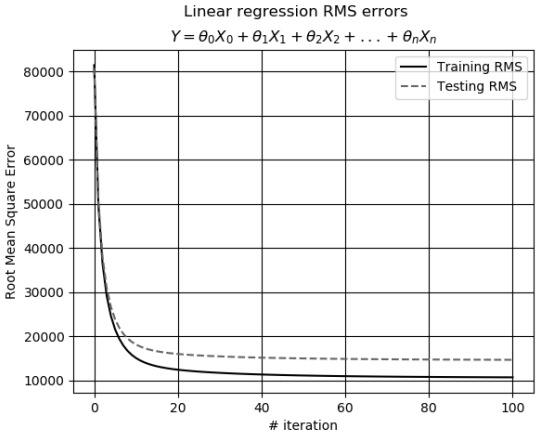

Figura 1. RMS obtidos no treinamento e teste a cada iteração.

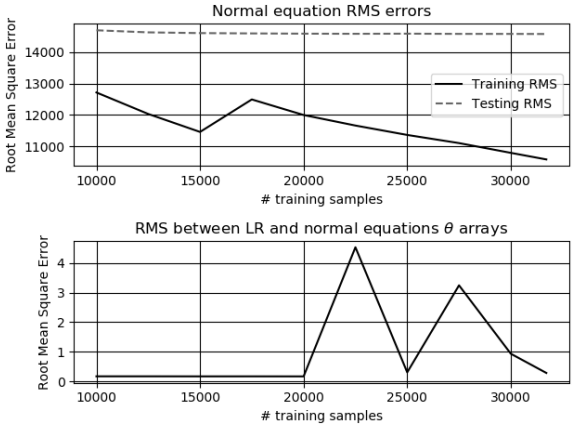

Figura 2. RMS obtidos no treinamento e teste para diversos números de amostras.

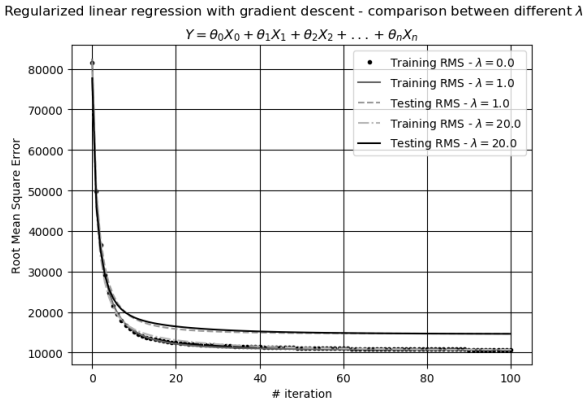

Figura 3. RMS obtidos nos modelos regularizados.

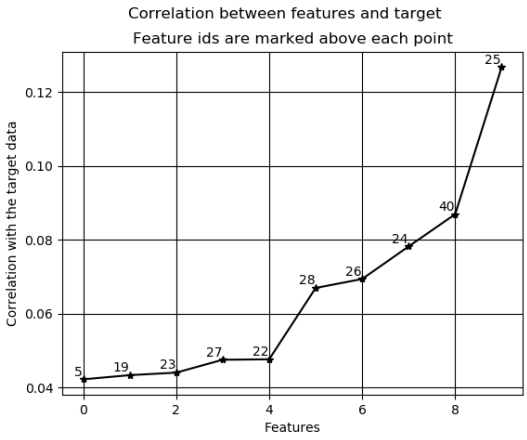

Figura 4. Correlação dos atributos de entrada com a saída.

3.2. Modelo linear simples regularizado

Apesar de não ter sido observado efeitos de *overfitting* nos resultados anteriores, propomos o uso da regularização, conforme explicado na subseção 2.4. Foram testados dois valores de λ, um igual a 1.0 e o outro, vinte vezes maior, 20.0. A figura 3 compara os resultados encontrados para a técnica de *gradient descent* entre os modelos regularizados e aquele da subseção anterior. Verifica-se que, para estes casos, a regularização não modificou substantivamente o erro encontrado, tanto para o conjunto de treinamento quanto o de teste. Isso indica que os valores sugeridos de λ são muito pequenos quando comparados a cada um dos fatores $\boldsymbol{\theta}_j^2$ individualmente. Estes mesmos modelos também foram submetidos às equações normais, porém, assim como no *gradient descent*, nenhuma grande alteração foi observada.

3.3. Redução de atributos do modelo

Uma outra tentativa realizada para minimizar o erro foi reduzir o número de atributos utilizados para o modelo. A seleção dos atributos foi feita de acordo com a correlação de cada um deles em relação à saída, uma vez que, como o modelo ainda é linear, espera-se que variáveis mais correlatas sigam o mesmo padrão que aquele observado na saída. A figura 4 representa apenas as 10 variáveis mais correlatas e que foram utilizadas para o novo modelo proposto. Vale dizer que a correlação mais alta, correspondente ao atributo 25 (*Avg. keyword (avg. shares)*), ainda é um valor bem baixo, isto é, 0.125, indicando que nenhuma vaiável influencia fortemente o número de compartilhamentos de uma publicação. A figura 5 compara os resultados encontrados para o modelo linear simples e para o modelo proposto nesta subseção. Observa-se um ligeiro aumento do erro de treinamento para o modelo reduzido a partir da iteração 10, indicando uma leve presença de *overfitting*. Além disso, verifica-se que ambos os erros de treinamento e teste do modelo reduzido convergem mais rapidamente do que aqueles do modelos simples aos valores onde permanecem praticamente constantes. Por fim, ressalta-se que as equações normais também foram utilizadas, porém seus resultados não são representados neste relatório por efeitos de simplificação.

3.4. Modelo linear com atributos de segunda ordem

Outra técnica que pode ser empregada para reduzir os erros de treinamento e teste é a adição de novos atributos baseados nos já existentes ao modelo simples. Nessa etapa, portanto, adiciona-se ao modelo linear simples todas as suas variáveis ao quadrado, de forma a obter a nova relação $Y_2 =$

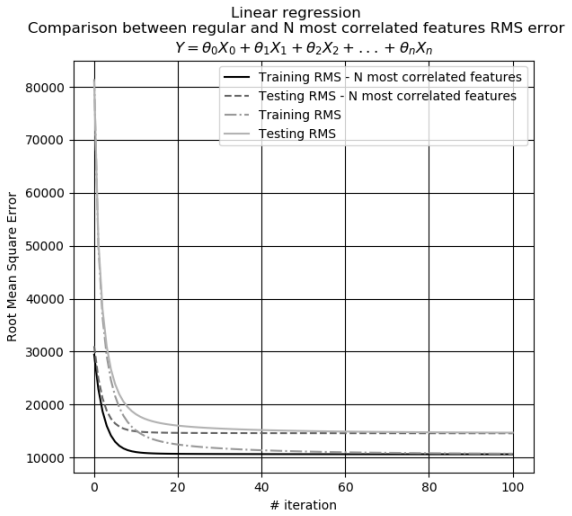

Figura 5. Comparação entre modelo linear simples e aquele com as 10 variáveis mais correlatas com a saída.

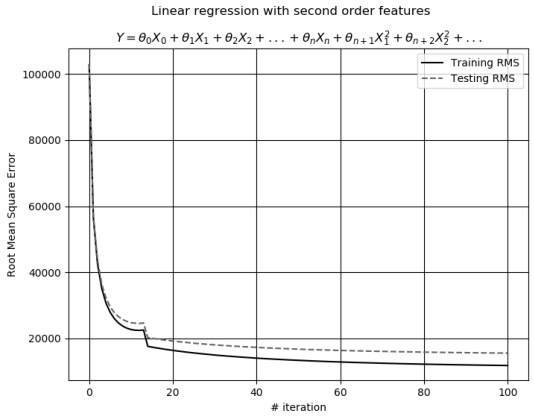

Figura 6. RMS obtidos no treinamento e teste do modelo com atributos de segunda ordem a cada iteração.

$Y + \theta_{N+1} X_1^2 + \theta_{N+2} X_2^2 + \ldots + \theta_{2N} X_N^2$. Espera-se com esse novo modelo uma melhor adequação aos dados de treinamento e aos de teste simultaneamente, isto é, sem a observação de *overfitting*. Vale lembrar que as equações 3 e 4 não sofrem quaisquer alterações, dado o fato que a equação de custo 2 continua linear em $\boldsymbol{\theta}$. A figura 6 possui o resultado da técnica de *gradient descent* para este modelo. Observa-se que na iteração 17, o valor de α é modificado, conforme descrito na seção 2.1, uma vez que o erro produzido foi superior àquele encontrado na iteração precedente. Os erros RMS de treinamento e de teste ao fim de 100 iterações encontrados foram, respectivamente, 11830 e 15570, superiores àqueles verificados para o modelo linear simples. O uso da equação 3, por sua vez, resultou em erros em torno de 10540 e 14580, muito parecidos àqueles obtidos pela mesma técnica aplicado ao modelo linear simples.

3.5. Modelo linear com atributos de terceira ordem

O último modelo testado neste relatório utilizou atributos de segunda e terceira ordens, obtendo a expressão $Y_3 = Y_2 + \theta_{2N+1} X_1^3 + \theta_{2N+2} X_2^3 + \ldots + \theta_{3N} X_N^3$. A figura 7 possui a evolução dos erros encontrados no decorrer das 100 iterações. Assim como ocorrido na figura 6, o valor do parâmetro α - *learning rate* - teve que ser diminuído, já que o erro de treinamento da iteração 3 foi superior àquele da iteração 2. Os menores erros RMS encontrados para treinamento e teste foram, respectivamente, 12743 e 16208, superiores àqueles obtidos no modelo linear simples. Por fim, o uso das equações normais resultou em erros iguais a 10458 e 15084. O primeiro foi ligeiramente inferior àquele do modelo simples, enquanto que o segundo, superior. Esse fato ilustra bem uma das dificuldades encontradas quando aumenta-se a complexidade do modelo, o *overfitting*. Neste caso, diminui-se o erro do conjunto de treinamento, porém aumenta-se o do conjunto de teste, indicando que o modelo se adequou de maneira exagerada aos dados de treinamento. Conforme já discutido, uma das possíveis soluções a este problema é a regularização.

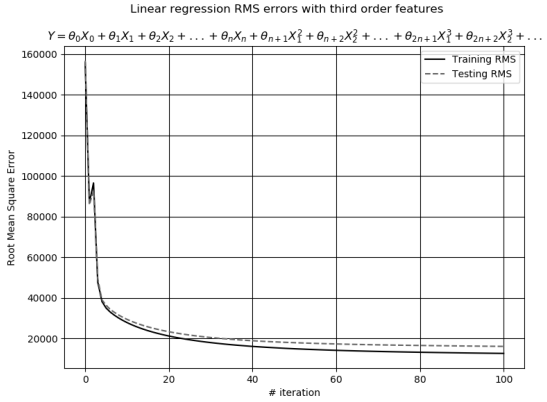

Figura 7. RMS obtidos no treinamento e teste do modelo com atributos de segunda e terceira ordens a cada iteração.

4. Conclusions

Cinco modelos lineares foram propostos para a predição do número de compartilhamentos de uma publicação da rede social Mashable. Em praticamente todos eles, os erros RMS de treinamento e teste ficaram em torno de 10500 e 14500, respectivamente, indicando que o modelo linear não é capaz de modelar com precisão o comportamento dos usuários desta rede social.

Referências

[1] Kelwin Fernandes, Pedro Vinagre, and Paulo Cortez. A proactive intelligent decision support system for predicting the popularity of online news. In Francisco Pereira, Penousal Machado, Ernesto Costa, and Amílcar Cardoso, editors, *Progress in Artificial Intelligence*, pages 535–546, Cham, 2015. Springer International Publishing. 1

[2] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. 1