Usando regressão linear para prever a popularidade de notícias online

Mateus Coelho *

Abstract

Com a popularização da Internet e de redes sociais, meios de comunicação como jornais e revistas têm adquirido importância considerável devido à rápida divulgação de notícias. Assim, saber se uma notícia publicada na Web será popular ou não tornou-se um tópico quente da área de Machine Learning. Nesse projeto, utilizando um dataset grande e de features diversificadas, esse problema foi abordado de maneira simples testando diversos modelos de regressão linear para estimar a quantidade de compartilhamentos de uma notícia. Após várias tentativas de modelos que forneciam medidas de erro absoluto médio entre 3000 e 3300, o melhor resultado foi obtido removendo notícias com mais de 20000 compartilhamentos do conjunto de treino e usando-o em um modelo de regressão linear simples com regularização. O erro absoluto médio foi de 2642 compartilhamentos, o qual não é satisfatório pois não é possível prever nada com segurança, já que 94% das notícias do conjunto de teste tem até 10000 mil compartilhamentos.

1. Introdução

A previsão de popularidade de uma notícia publicada na Internet é um tópico quente na área de machine learning tendo em vista que vários trabalhos já foram publicados abordando técnicas diferentes e *datasets* de diferentes meios de comunicação. Em geral, há duas maneiras de prever a popularidade: usando *features* conhecidas somente após a publicação ou usando features pré-definidas. O primeiro modo é mais fácil e, geralmente, fornece resultados melhores do que o segundo modo. Existe também uma divisão na forma como a popularidade é estimada, já que alguns trabalhos estimam a quantidade de compartilhamentos que uma notícia terá e outros apenas classificam ela como popular ou impopular.

Usando o método de prever a popularidade com *features* pré-definidas, alguns autores já conseguiram resultados interessantes. Petrovic et al. [1] com um *dataset* de 21 milhões de tweets tentou prever a quantidade de retweets usando *features* relacionadas ao texto do tweet e a dados do autor. Fazendo uma classificação binária entre tweet e retweet, um F-1 *score* de 47%. Bandari et al. [2], com um *dataset* correspondente a uma semana de notícias retiradas do Feedzilla, se propôs a classifica-las em 4 classes de popularidade tendo como valor de referência o número de tweets que mencionam um artigo. Fazendo uso de um conjunto diversicado de *features* os melhores resultados foram de 77% a 84% de acurácia com os métodos Bayes e Bagging, respectivamente. Por último, Fernandes et al. [3], que utilizou o mesmo *dataset* que esse projeto, testou cinco modelos modernos para prever a quantidade de compartilhamento de notícias. Obteve o melhor resultado com o método *Random Forests* e teve um *discrimination power* de 73%.

Tendo em vista estes bons resultados usando métodos que refletem o estado da arte em *machine learning*, esse projeto analisou se é possível com métodos simples prever a popularidade de uma notícia. Para tanto, foi utilizado o clássico modelo de regressão linear com dois métodos de resolução, que foram *gradient descent* e *normal equations*. Além disso, um *dataset* com 39000 notícias do popular site Mashable retiradas em 2015 também foi utilizado para treinar e testar os modelos propostos.

2. Soluções Propostas

2.1. Conjunto de dados

O conjunto de dados utilizado neste *paper*, foi criado e disponibilizado por Fernandes et al. [3], apresenta 39797 notícias retiradas do popular site de notícias Mashable no idioma inglês. O *dataset* como foi dispobilizado tem 61 *features*, sendo 58 atributos preditivos, 2 não preditivos (URL e tempo entre mineração e publicação) e 1 objetivo, que é a quantidade de compartilhamentos. Os dados foram coletados entre janeiro de 2013 e janeiro de 2015 e os artigos que não poderiam ser tratados automaticamente foram removidos, assim como os artigos mais recentes (até quatro semanas). Os atributos do conjunto de dados podem ser divididos em 6 categorias: palavras, *links*, mídia digital, data, palavras-chave e processamento de linguagem natural. O valor de todos os atributos são booleanos, números ([0,∞]) ou razões ([0-1]). Mais informações sobre o *dataset* podem ser obtidas na sua página no repositório UCI Machine Learning.

Antes de começar a testar os modelos foram efetuadas

*Contact: mateus_coelho@live.com

pequenas alterações no *dataset*. Os atributos não preditivos foram removidas, o atributo *É fim de semana?* foi removido pois tem informação repetida e um atributo com todos os valores iguais a 1 foi adicionado pois correspondente à *feature* que multiplica o coeficiente linear da função utilizada para predizer o valor de compartilhamentos. Além disso, os 7 atributos que indicam o dia semana em que a notícia foi publicada foram substituídas por 3 três atributos binários que juntos correspondem ao número do dia em base binária, por exemplo 010 corresponde a terça-feira.

2.2. Modelos

Primeiramente foi testado o modelo de regressão linear básico correspondente à equação 1 com todas as *features* disponíveis. Foram testados também modelos lineares com termos quadrados e cúbicos dos atributos. Em uma tentativa mais extrema de tratar os atributos discretos do *dataset*, a equação 1 foi aplicada com todas os atributos menos os discretos.

$$h_\theta = \theta_1 + x_1\theta_2 + x_2\theta_3 + \cdots + x_n\theta_n \qquad (1)$$

A partir da aplicação do método de *Random Forests* efetuado por Fernandes et al. [3] no mesmo *dataset*, foram obtidas as 10 *features* mais importantes, que são:

1. Palavra-chave média (média de compartilhamentos).
2. Palavra-chave média (máximo de compartilhamentos).
3. Proximidade do terceiro tópico mais relevante do LDA.
4. Categoria do artigo
5. Quantidade mínima de compartilhamento de links do Mashable
6. Melhor palavra-chave (média de compartilhamentos).
7. Quantidade média de compartilhamento de links do Mashable
8. Proximidade do segundo tópico mais relevante do LDA.
9. Pior palavra-chave (média de compartilhamentos).
10. Proximidade do quinto tópico mais relevante do LDA.

Assim, o modelo da equação 1 também foi aplicado para apenas esses 10 atributos.

Vale ressaltar que a regularização também foi utilizada em todos os modelos mesmo sem *overfitting*.

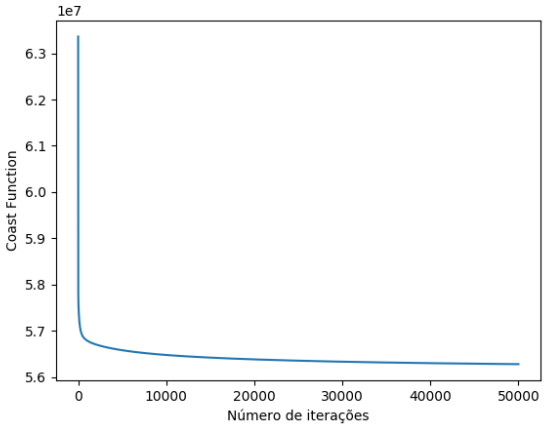

Figure 1. Convergência do gradient descent.

3. Experimentos e discussão

Para realizar os testes foram desenvolvidos os algoritmos *gradient descent* e *normal equations* em Python com bibliotecas auxiliares como *pandas* e *numpy*, sem entretanto usar bibliotecas que já tinham soluções prontas.

A respeito desse métodos de minimizar a *coast function* ($J(\theta)$) percebeu-se uma clara vantagem do segundo especialmente para casos em que a quantidade de *features* a serem analisadas era pequena (menor que 200). Além disso, foram obtidos melhores resultados usando *normal equations* porque para ter um valor parecido usando *gradient descent* era necessário fazer muitas iterações, o que demoraria muito para ser processado. Note que as condições de parada do método iterativo foram: diferença entre coast functions menor que 10 ou número máximo de iterações igual a 50000. Portanto, os resultados expostos neste *paper* são os conseguidos pelo *normal equations* apesar de ambos serem testados.

Ao aplicar o gradient descent foram plotados gráficos de coast function vs. número de iterações, mas todos têm obviamente o mesmo formato, mudando apenas o valor da coast function. Um exemplo desse gráfico pode ser visto na figura 1, que corresponde ao primeiro e mais básico modelo testado sem regularização. Para aumentar a rapidez da convergência foi empregado também a normalização da média em todas os atributos que não tinham valores entre -1 e 1.

Com relação ao uso do dataset para treinar e testar os modelos, foi respeitada uma repartição de 80% para o conjunto de treino e 20% para o conjunto de treino. Ademais, nos casos em que regularização foi aplicado, o método de cross-validation foi empregado para encontrar o valor ideal do parâmetro λ da regularização. A proporção foi de 80% do conjunto de treino para efetivamente treinar e 20% do conjunto de treino para encontrar o parâmetro.

Table 1. Resultados após remoção de casos do conjunto de treino

Modelo	MAE
Eq 1 com regularização e remoção de 33 casos	3110
Eq 1 com regularização e remoção de 226 casos	3082
Eq 1 com regularização e remoção de 1235 casos	2642

Table 2. Erros dos diversos modelos testados

Modelo	MAE
Eq 1	3181
Eq 1 com regularização	3210
Eq 1 com quadrados	3181
Eq 1 com quadrados e regularização	3152
Eq 1 com cubos	3181
Eq 1 com cubos e regularização	3146
Eq 1 sem atributos discretos	3186
Eq 1 sem atributos discretos com reg.	3186
Eq 1 com top 10 atributos	3184
Eq 1 com top 10 atributos e reg.	3184

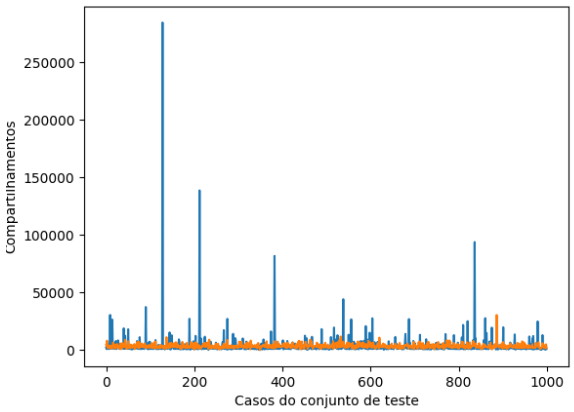

Figure 2. Comparação entre valores dados pelo modelo e os valores reais de compartilhamentos.

De maneira geral os resultados foram ruins já que os valores do erro médio absoluta (MAE) foram altos. A tabela 3 mostra resumidamente os erros para os modelos especificados na seção anterior. Já que os valores de compartilhamentos no conjunto de teste são quase todos abaixo de 10000, pode-se concluir que a função preditora tem uma precisão muito baixa.

O gráfico da função preditora e dos valores reais de compartilhamento para o modelo básico pode ser visualizado na figura 2, na qual o eixo x indica o número do caso do conjunto de teste. Com ele, pode-se perceber que existem alguns casos em que o valor de compartilhamento real é muito alto e a função não acompanha. Além disso, é patente que a maioria dos casos do *dataset* tem valores baixos do atributo objetivo.

A solução, então, que resultou no melhor erro foi remover alguns casos do conjunto de treino que tinham valores de compartilhamentos altos. Assim o modelo ficaria mais ajustado aos casos que são maioria no conjunto de teste. Esse processo foi realizado de forma gradual ao longo de três iterações e testando o modelo da equação 1 para verificar se o erro seria menor. Os erros dos testes nessas três iterações podem ser visualizados na tabela 1.

A remoção foi feita plotando o gráfico de cada *feature* pela quantidade de compartilhamentos e retirando pontos (casos) que estavam distantes da maior concentração de pontos. Foram removidos pouco mais de 1000 casos que tinham a maior quantidade de compartilhamentos do conjunto de treino.

4. Conclusões e direções futuras

O objetivo do projeto é verificar se é possível obter uma boa estimativa da quantidade de compartilhamentos que uma notícia terá usando modelos simples de regressão linear. Os resultados obtidos mostram que não é possível pois o erro é muito grande comparado com os valores de compartilhamentos reais. O melhor resultado teve um MAE de 2642, mas ainda não é satisfatório pois 94% das notícias do conjunto de teste tem até 10000 mil compartilhamentos.

Algumas opções que poderiam ser testadas em uma eventual continuação deste projeto são testar modelos mais complexos de regressão com multiplicação de *features* ou eleva-las a graus maiores, como 25 ou 50. Além disso pode-se verificar se remover mais casos do conjunto de treino ajuda a função preditora a ficar mais ajustada e reduzir o erro.

References

[1] Sasa Petrovic, Miles Osborne, and Victor Lavrenko. Rt to win! predicting message propagation in twitter. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, pages 586–589, Barcelona, Espanha, 2011. 1

[2] Roja Bandari, Sitaram Asur, and Bernardo A. Huberman. The pulse of news in social media: Forecasting popularity. In *Proceedings of the Sixth International AAAI Conference on Weblogs and Social Media*, pages 26–33, Anchorage, AK, USA, 2012. 1

[3] Kelwin Fernandes, Pedro Vinagre, and Paulo Cortez. A proactive intelligent decision support system for predicting the popularity of online news. In *Proceedings of the 17th EPIA 2015 - Portuguese Conference on Artificial Intelligence*, pages 535–546, Coimbra, Portugal, 2015. 1, 2