O Uso de um Classificador de Texto para um Sistema de Recomendação de Twitters

Leonardo Panta Leão
DCOMP - Universidade Federal de Sergipe
Av. Marechal Rondon s/n
São Critóvão-SE, Brasil
leonardopspl@gmail.com

Maria Augusta S. N. Nunes
DCOMP - Universidade Federal de Sergipe
Av. Marechal Rondon s/n
São Critóvão-SE, Brasil
gutanunes@dcomp.ufs.br

Christian Nunes Aranha
Cortex Intelligence
Rua da Assembléia, 10, salas 3711/3712
Rio de Janeiro-RJ, Brasil
cnaranha@uol.com.br

RESUMO

Sistemas de recomendação usam dados de histórico baseado em preferências de usuário ou dados estatísticos sobre usuários para predizer um novo item que um novo usuário possa gostar. Aplicações que utilizam esses métodos focam em recomendar itens para compra e para personalização da experiência de navegação na rede. Utilizar apenas os dados de histórico de navegaccão e preferências do usuário têm sido a ideia principal dos métodos de Filtragem Colaborativa. Esses métodos foram categorizados como métodos baseados em memória, enquanto que métodos que usam esses dados históricos para a construção de um modelo para essas predições são chamados de métodos baseados em modelo. Podemos utilizar uma abordagem baseada em modelo para o problema de recomendar um novo Twitter para novos usuários baseados em um modelo construído utilizando dados históricos de navegação de usuário. Neste paper apresentamos um classificador de texto baseado em classificadores lineares para a construção de um modelo que foca na predição de Twitters para usuários.

Palavras-chave

Sistema de recomendação, Twitter, categorização de texto, classificador linear

1. INTRODUÇÃO

Sistemas de recomendação usam dados históricos das preferências do usuário, compras e outros dados disponíveis sobre os usuários para recomendar itens que têm alta probabilidade de aceitação por parte do usuário [7] . Uma das primeiras técnicas de recomendação era baseada em filtragem colaborativa de vizinho mais próximo (*Nearest Neighbor*) ([8], [9], [2]) que usa apenas o histórico do usuário como entrada. Geralmente quando é citado filtragem colaborativa essas técnicas são utilizadas. As técnicas de vizinho mais próximo utilizam a noção da similaridade das preferências atuais do usuário com preferências já obtidas de usuários com perfis semelhantes do novo usuário. A escalabilidade é um problema a ser considerado no uso da técnica de vizinho mais próximo.

Os métodos de filtragem colaborativa que usam métodos baseados em modelo utilizam dados históricos para a construção de modelos que são utilizados para predizer novas preferências para os usuários. Existe uma abordagem baseada em modelo proposta por [2] que faz uso de redes bayesianas.

Neste paper exploraremos uma abordagem baseada em modelo para a construção de um sistema de recomendação de Twitters baseado em classificação de texto para a construção de um modelo que serve para as predições dos Twitters que possivelmente o usuário aceite. Classificadores lineares têm sucesso no domínio da classificação de texto ([6], [10], [11]). A ideia principal da abordagem que faz uso de modelos é que o maior custo computacional seja na etapa de construção do modelo, já a etapa de uso do classificador tem um custo muito mais baixo em relação a esta etapa anterior.

O trabalho descrito neste paper basicamente utiliza um classificador de textos para agrupar os Twitters em conjuntos de temas principais, em seguida utilizamos uma base de dados de naveção para a construção do modelo. Esses dados de navegação são importantes para darmos prioridade quanto à relevância do Twitter a ser recomendado, ou seja, para ranquearmos os elementos dos conjuntos.

Na seção 2 apresentamos uma breve explicação sobre a construção de modelos utilizando um classificador de texto (seções 2.1, 2.2 e 2.3), na seção 3 mostramos os experimentos e uma breve descrição do aplicativo que aplica a técnica e por fim na seção 4 damos a conclusão sobre os resultados alcançados e trabalhos futuros.

2. ABORDAGEM BASEADA EM MODELO

O problema de predizer quando um usuário (ou um consumidor) irá aceitar certas recomendações pode ser modelado para um problema de classificação binária. Porém, a probabilidade do usuário aceitar ou não a recomendação é um fator que deve ser levado em consideração também em sistemas de recomendação. Essa informação pode ser utilizada para ranquear os itens quanto à probabilidade e construirmos uma lista de recomendações para apresentarmos

ao usuário. Assim, é necessário que nosso classificador retorne uma lista (um *score*), onde os maiores pontuadores têm maior probabilidade de serem aceitos pelo usuário.

A principal função da construção de um modelo é gerar um classificador que seja eficiente e computacionalmente rápido no momento de seu uso, mas que possa levar o tempo que for necessário na etapa off- line de contrução. Na etapa de construção desse modelo, geralmente é utilizada uma quantidade massiva de dados de entrada e algoritmos com custo computacional muito elevado. Os dados são tratados pelo algoritmo e o resultado é um classificador ou um conjunto de dados agrupados, conforme o domínio e o objetivo da utilização do modelo, de forma que seu resultado seja utilizado de maneira eficiente, seja através de um algoritmo complexo, mas de custo computacional consideravelmente baixo em relação à contrução do modelo, ou utilizando os dados resultantes da construção do modelo de forma útil.

2.1 Classificação de texto

No domínio da classificação de texto, podemos utilizar um modelo para a construção de um classificador genérico de texto. O que fazemos é reduzir o problema para um problema de classificação binária, um para cada categoria, onde cada um deseja determinar um método para predizer quem é membro da classe e quem está fora dessa classe. O primeiro passo é lermos o documento, segmentarmos em seções (se existirem) onde a identidade separada de cada seção seja significante para a categorização, simbolizamos cada seção que contém texto no documento, convertemos todos os *tokens* (palavras e símbolos) para suas formas canônicas, isto é, realizar *stemming* (converter a palavra para sua palavra de origem, exemplo: converter, brasileiro, brasileira, brasileiros, brasileiras, para Brasil), esta etapa é opcional, em seguida deletamos palavras comuns que não são úteis para a classificação (exemplo: interjeições, preposições), esta etapa também é opcional, e por fim mostrar uma saída representada por *tokens* (símbolos) r do documento d de forma que a lista de *tokens* em cada seção possa ser determinada. Essa sequência de passos é o início da tarefa de classificação de texto, é a preparação do documento para a que ele possa ser classificado. Dessa forma, por consistência, usamos esse procedimento tanto para a inserção de documentos no treinamento, quanto na etapa de classificação do documento [3].

No término da primeira etapa cada documento é representado por uma lista das ocorrências das palavras. Podemos saltar a etapa posterior de seleção de características utilizando a própria lista de palavras como características. Porém, para atingirmos uma eficiência mais elevada é útil que levemos em consideração características que representam mais claramente a categoria pertencente aos textos. Basicamente nesta etapa: reunimos as representações em *tokens* dos documentos de treinamento, a partir desse conjunto selecionamos o conjunto de características relevantes para a pertinência do documento à categoria, e por fim listamos as características para esse conjunto de treinamento e categoria. Dessa forma a saída da etapa de seleção de características consiste na lista de ocorrência dessas palavras que representam melhor as categorias dos textos. Essa saída muitas vezes já é suficiente para o sistema computacional julgar a categoria à qual o texto pertence.

Após a etapa de seleção de características, cada documento é representado por um vetor de ocorrência das palavras para cada categoria onde cada componente do vetor corresponde a uma palavra característica selecionada para a categoria na etapa anterior. Na última etapa, lemos a representação em *tokens* do documento e a lista de características selecionadas para o conjunto de treinamento e para a categoria. Para esse documento criamos um vetor com a quantidade das ocorrências das características na lista. Opcionalmente transformamos esse vetor por normalização ou arredondando cada componente para um valor máximo. Aumentamos o vetor obtido adicionando um componente final ajustado para um valor não-nulo A, produzindo um vetor x. A saída é o vetor x que é uma representação numérica do documento.

2.2 Algoritmo utilizando classificador linear

Formalmente, um problema de categorização de duas classes é determinar um rótulo $y \in \{-1, 1\}$ associado a um vetor x de variáveis de entrada. Usamos então uma combinação linear de pesos das entradas rotuladas da seguinte maneira $(x_1, y_1), ..., (x_n, yn)$. Onde n é o número de itens no conjunto de dados de treinamento. Especificamente desejamos encontrar um vetor de pesos w e um *threshold* θ de forma que $w^T x < \theta$ se o rótulo tem $y = -1$ e $w^T x \geq \theta$ se o rótulo tem $y = 1$. Uma valor de $w^T x - \theta$ pode ser atribuído a cada ponto de dado como um substituto para a probabilidade de x ser da classe.

O problema descrito pode ser prontamente convertido para um considerando o *threshold* θ como sendo zero. É possível fazer isso convetendo um ponto de dado x no espaço original em $\tilde{x} = [x, 1]$ no espaço ampliado. Cada hiperplano w no espaço original com *threshold* θ pode então ser convertido em $[w, -\theta]$ que passa pela origem do espaço ampliado. Invés de procurarmos por ambos um vetor de pesos d-dimensional com um *threshold* θ, podemos procurar por um vetor de pesos $(d+1)$-dimensional com um *threshold* antecipado de zero. Então, assumimos que os vetores de variáveis de entrada foram adequadamente transformados de forma que podemos atribuir $\theta = 0$. Também assumimos que x e w são vetores d-dimensionais [11].

Para a construção do modelo utilizamos o método de classificação de texto utilizando o algoritmo de menores quadrados [10] que é baseado na seguinte formulação para computar um separador linear $\hat{w}$:

$$\hat{w} = \arg\min_{w} \frac{1}{n} \sum_{i=1}^{n} (w^T x_i - y_i)^2. \quad (1)$$

O método dos menores quadrados é extensivamente utilizado em engenharia e estatística. Embora o método tem principalmente sido associado com problemas de regressão, ele pode ser usado para classificação de texto [5].

A solução de (1) é dada por

$$\hat{w} = \left(\sum_{i=1}^{n} x_i x_i^T \right)^{-1} \left(\sum_{i=1}^{n} x_i y_i \right).$$

Um problema com essa formulação é que a matriz $\sum_{i=1}^{n} x_i x_i^T$ deve ser singular ou mal-condicionada [5]. Isso ocorre, por exemplo, quando n é menor que a dimensão de x. Note que nesse caso, para qualquer $\hat{w}$, existem uma infinidade de soluções para $\tilde{w}$ de $\tilde{w}^T x_i = \tilde{w}^T x_i$ para $i = 1, ..., n$. Isso implica que (1) tem uma infinidade de possíveis solucoes para $\hat{w}$.

Um remédio para esse problema é o uso de uma pseudo-inversão [10]. Porém, um problema com a abordagem utilizando pseudo-inversão é a complexidade computacional. Afim de manipular amplos sistemas dispersos precisamos usar algoritmos iterativos que não se baseiam em técnicas de fatoração de matrizes. Consequentemente, utilizamos um método padrão de regressão de pico [4] que adiciona um termo de regularização a (1):

$$\hat{w} = \arg\min_{w} \left[\frac{1}{n} \sum_{i=1}^{n} (w^T x_i y_i - 1)^2 + \lambda w^2 \right], \qquad (2)$$

Onde λ é um parâmetro de regularização apropriadamente escolhido. A solução é dada por

$$\hat{w} = \left(\sum_{i=1}^{n} x_i x_i^T + \lambda n I \right)^{-1} \left(\sum_{i=1}^{n} x_i y_i \right),$$

onde I denota a matriz identidade. Note que $\sum_{i=1}^{n} x_i x_i^T + \lambda n I$ irá sempre ser não singular, o que resolve o problema do mal-condicionamento.

2.3 Construção do Modelo

Para a construção do modelo utilizamos o classificador de texto descrito nas seções 2.1 e 2.2. Inicialmente fazemos o tratamento do texto conforme a seção 2.1 para simplificar e tornar mais eficiente a tarefa de atribuir os valores das características do texto. Com o texto já tratado procuramos no texto pelas palavras que caracterizam aquele texto como sendo pertencente a tal categoria. Exemplificando, para a construção da parte do modelo sobre Twitters do tema cinema, adicionamos ao vetor de características (o documento x_i) na posição c_j o incremento conforme a palavra representada na posição c_j no documento x_i é encontrada.

No término desta etapa temos os documentos com os vetores de características preenchidos. Nessa etapa utilizamos o método dos mínimos quadrados [10] descrito na seção 2.2. Assim, obtemos o vetor de pesos w com os pesos dos textos obtidos aplicando esse método. Finalmente percorrendo o vetor w de pesos tomamos uma decisão binária classificando os documentos x_i como pertencentes a tal classe conforme o *threshold* obtido apartir da taxa de acerto verificando o rótulo y_i. Dessa maneira, temos um classificador de texto construido. Utilizamos o classificador construído para cada tema principal para agruparmos os Twitters da base de dados.

Quando esta última etapa está concluída temos um modelo de dados que contêm Twitters classificados que serão utilizados para a recomendação conforme o perfil do usuário.

3. EXPERIMENTOS

Nos experimentos foi utilizado uma base de dados de Twitters fornecida pela Twitter Trending Topics [1]. Utilizamos uma instância da base de dados com cerca de 20 mil usuários, onde tínhamos a descrição, *id*, título e endereço do Twitter respectivo.

Os temas escolhidos para a carcterização do usuário foram: cinema, futebol, jornalismo, política e tecnologia. Inicialmente o usuário se cadastra, passando seus dados e suas preferências pessoais.

A base de dados foi construída utilizando o classificador de texto sendo agrupada conforme esses temas principais. O tempo para a construção do classificador para cada tema foi de aproximadamente 6 minutos. O tempo para a construção do modelo dos dados agrupados foi de cerca de 8 minutos. A base de dados utiliza a quantidades de seguidores (*followers*) para realizar o ranqueamento dos Twitters classificados, dessa forma o usuário receberá os Twitters em ordem dos que têm maior quantidade de seguidores.

A utilização do aplicativo que aplica a técnica foi feita de forma objetiva. O usuário faz o cadastro com seu nome, login e senha, para que ele seja identificado com suas preferências escolhidas também no momento do cadastro. Após isso o usuário realiza o login fazendo com que as opções escolhidas sejam disponíveis para a busca de Twitters que tem grande chance de ser aceito pelo usuário.

Quando o usuário solicita a recomendação de um Twitter o aplicativo retorna um Twitter de acordo com a preferência do usuário (Figura 1).

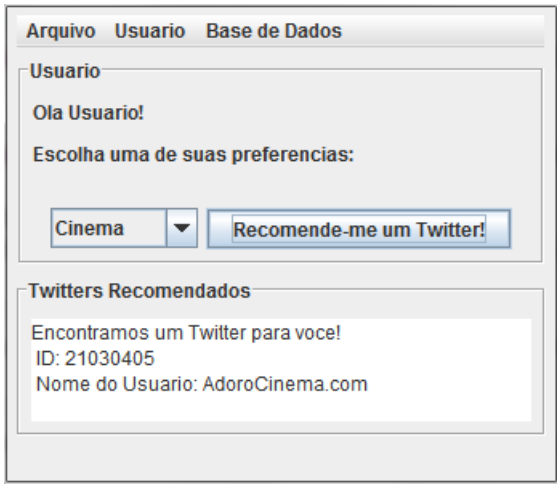

Figure 1: Aplicativo exemplificando utilização da técnica.

4. CONCLUSÕES

A técnica na prática seguiu as premissas de levar bastante tempo e ciclos computacionais durante a etapa de contrução do modelo. O tempo total da contrução do modelo é bastante elevado se comparado com a utilização do aplicativo pronto na hora de obter uma recomendação.

O classificador mostrou muitas limitações, uma taxa de falso

positivo (quando classifica errôneamente algo como pertecente à classe) de cerca de 42% das amostras. Esse fato se deve também à quantidade baixa de texto contida no campo de descrição do usuário do Twitter.

O aplicativo construído não é um sistema final pronto para ser aplicado num sistema de recomendação eficiente, mas sua construção mostra que a ideia de se usar um modelo para a o uso num sistema de recomendação de Twitters é válida.

5. REFERENCES

[1] T. T. Brasil. http://www.twittertrendingtopics.com/. *Acessado em 21 de Junho de 2010 às 22:56h.*

[2] J. Breese and D. Heckerman. Emperical analysis of predictive algorithms for collaborative filtering. *Proceedings of Fourteenth Conference on Uncertainty in Artificial Intelligence*, 1998.

[3] D. Heckerman and C. Kadie. Dependency networks for inference, collaborative filtering, and data visualization. *Journal of Machine Learning Research*, 1:49–75, 2000.

[4] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[5] V. Iyengar and T. Zhang. Empirical study of recommender systems using linear classifiers. *Proceedings of the Fifth Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 16–27, 2001.

[6] T. Joachims. Text categorization with support vector machines: learning with many relevant features. *European Conference on Machine Learing, ECML-98*, pages 137–142, 1998.

[7] M. A. S. N. Nunes. *Recommender Systems based on Personality Traits:Could human psychological aspects influence the computer decision-making process?* Berlin:VDM Verlag Dr. Müller, 2009.

[8] P. Resnick and P. Bergstrom. Grouplens: An open architecture for collaborative filtering of netnews. *Proceedings of CSCW*, 1994.

[9] U. Shardanand and P. Maes. Social information filtering: Algorithms for automating word of mouth. *Proceedings of CHI'95*, 1995.

[10] Y. Yang and C. G. Chute. An example-based mapping method for text categorization and retrieval. *ACM Transactions on Information Systems*, 12:252–227, 1994.

[11] T. Zhang and F. J. Oles. Text categorization based on regularized linear classification methods. *Information Retrieval*, 12:5–31, 2001.