Análise do Ruído Natural na Filtragem Colaborativa

Lívia de Azevedo
Universidade Federal Rural do Rio de Janeiro
Av.Governador Roberto da Silveira, Nova Iguaçu, S/N
Rio de Janeiro, Brasil
livia_de_azevedo@ufrrj.br

Gustavo Ebbo
Universidade Federal Rural do Rio de Janeiro
Av.Governador Roberto da Silveira, Nova Iguaçu, S/N
Rio de Janeiro, Brasil
gustavoebbo@ufrrj.br

Ivo Paiva
Universidade Federal Rural do Rio de Janeiro
Av.Governador Roberto da Silveira, Nova Iguaçu, S/N
Rio de Janeiro, Brasil
ivopaiva@ufrrj.br

ABSTRACT

Este relatório visa mostrar empiricamente como o ruído natural pode impactar nos modelos da Filtragem Colaborativa usados para realizar as recomendações e como alguns algoritmos de correção de ruído natural se desempenham para realizar tal correção. Assim, dois algoritmos famosos da Filtragem Colaborativa foram selecionados: *K-Nearest Neigborhood*(KNN) e *Regulared Singular Value Decomposition* (RSVD). Além disso, dois modelos de correção de ruído natural foram escolhidos para representar as táticas de correção. Experimentos na base de dados do *MovieLens* foram feitos para mostrar este impacto nos algoritmos de previsão e a eficácia dos modelos de correção. Por fim, foram apresentados os resultados dos testes realizados e as conclusões a respeito dos experimentos.

General Terms

Teoria, Experimental, Performace

Keywords

Sistemas de Recomendação, Ruído Natural, Filtragem Colaborativa

1. INTRODUÇÃO

Sistemas de Recomendação são sistemas providos de diversas técnicas e ferramentas de software que dão sugestões de itens para o consumo do usuário. Esses meios de avaliação consideram diversas características, como design, a interface gráfica mostrada ao usuário e ao tipo de técnica de recomendação considerada para gerar as melhores sugestões de itens para o usuário. A ideia e a motivação do uso de Sistemas de Recomendação vem da grande importância de auxiliar o usuário a escolher um determinado item dentro de um conjunto de escolhas numerosas que existem(e se permitem existir) em sistemas de grande porte de dados e variedades, como aplicações *e-commerce*. As recomendações evitam uma possível reação negativa do usuário perante a esta grande diversidade de escolhas [4]. Os Sistemas de Recomendação utilizam todas as possíveis informações fornecidas pelo usuário, diretamente ou indiretamente, para melhorar progressivamente, suas recomendações.

Esta análise é focada em sistemas de Filtragem Colaborativa, que realizam as recomendações baseadas em uma extensa base de dados de usuários ou itens. A filtragem colaborativa recomenda de acordo com a semelhança entre usuários ou itens, de forma que usuários semelhantes recebem recomendações semelhantes, assim como itens semelhantes recomendam itens semelhantes. Por recomendar baseado na avaliação de outros usuários, o sistema é passível de recomendar incorretamente por conta de ruídos na base, sejam naturais ou maliciosos. Os ruídos naturais são derivados da própria tendência humana de errar, adicionando o fato de suas avaliações dentro de um contexto terem uma alta probabilidade de serem incertas, ocasionando erros na base de dados. O ruído malicioso nas avaliações é derivado da inserção proposital de avaliações em busca de alcançar certos interesses, alterando as recomendações do sistema para atender a estes objetivos [3].

Devido a este problema do ruído poder limitar a eficácia da recomendação, algumas abordagens de detecção e correção de ruído foram proposta. Neste trabalho, a atenção foi dada aos algoritmos chamados aqui de O'Mahony [3] e R.Toledo [5].

O objetivo deste trabalho é promover uma análise da minimização dos efeitos do ruído natural utilizando as duas técnicas citadas anteriormente e também compará-las em questão de desempenho de suas acurácias. Com tudo isso, a acurácia é um fator relevante para definir o desempenho do modelo usado, levando a concluir que os impactos negativos neste fator causam uma depreciação do algoritmo.

Na próxima seção, será explicado com mais detalhes o ambiente da Filtragem Colaborativa. Na seção 3, será abordado a relação do ruído natural com os algoritmos da Filtragem Colaborativa e detalhado os algoritmos de correção e detecção. Na seção 4, será apresentado a metodologia e dos resultados dos experimentos. Por fim, é relatado a conclusão com relações aos resultados obtidos e possíveis trabalhos futuros.

2. FILTRAGEM COLABORATIVA

A recomendação é utilizada para auxiliar pessoas indecisas em relação à alguma coisa. Neste caso, é comum buscar referência sobre outras pessoas que obtenham tal objeto para saber se este será útil para o usuário que busca a recomendação. Para suprir esta dúvida, é preferível a acei-

tação de recomendações de especialistas e usuários comuns ao que busca a recomendação, pois estes passam uma maior personalização à recomendação. A Filtragem Colaborativa busca avaliar a recomendação de um usuário tomando em conta a recomendação de outros usuários similares a este, para simular o costume do usuário em buscar referência em outras pessoas de costumes parecidos. Na Filtragem Colaborativa existem duas abordagens diferentes, utilizando algoritmo baseado em memória e baseado em modelo. Neste artigo, dois algoritmos são utilizados para teste, o *K-Nearest Neighbours*(Baseado em memória) e *Regulared Singular Value Decomposition*(RSVD) (baseado em modelo). Mas detalhes sobre o algoritmo do KNN encontra-se em [1] e do RSVD em [2], identificado-o como o modelo básico de fatorização. A recomendação baseada em memória utiliza, diretamente, a recomendação dos usuários ou itens similares para realizar a recomendação. Ambas abordagens recomendam baseados em uma função de similaridade, que é escolhida de acordo com a base de dados. Quanto maior a similaridade, maior a probabilidade de recomendação. A dificuldade do algoritmo baseado em memória é administrar bem o uso da memória e a esparsidade da matriz de usuários ou de itens.

O algoritmo baseado em modelo, utiliza modelos de mineração de dados e de aprendizado de máquina para encontrar padrões de recomendações na base de dados e prever a outros usuários baseado nestes padrões. O algoritmo baseado em modelo depende de uma quantidade de dados elevada para prever com maior acurácia.

3. RUÍDO NATURAL NA FILTRAGEM COLABORATIVA

O ruído natural é inserido na base de dados quando uma avaliação imprevisível ocorre. É comum usuários terem o seu próprio padrão de notas, uma nota é alta para um usuário enquanto essa mesma nota é baixa para outro e cometer erros de entendimento do contexto na hora de dar uma nota. Neste trabalho, todas as notas foram normalizadas com o critério de normalização *Z-score* [1], tomando em conta a variância das notas dos usuários. O problema para corrigir o ruído natural é encontrar uma forma de identificar uma nota imprevisível de um usuário dentro de sua variância comum e esta é um dos desafios nas algoritmos que tratam o ruído. A Filtragem Colaborativa pode ser sensível ao ruído, pois esta diferença nas avaliações irá ser utilizada para recomendar para outros usuários, podendo resultar em recomendações inconsistentes, abaixando a confiança do usuário nas recomendações do sistema. As tentativas de minimizar os efeitos do ruído natural na recomendação foram utilizados dois algoritmos, O'Mahony [3] e R.Toledo [5].

3.1 Algoritmo O'Mahony

O algoritmo de detecção de ruído do O'Mahony [3], tenta detectar as notas ruidosas na base de dados para em seguida removê-las. A técnica detecta os ruídos da seguinte forma: Sendo R o conjunto de notas e $r_{u,i}$ a nota de um usuário u para um item i. A técnica utilizada para descobrir as notas ruidosas é checar se a consistência, critério adotado pelo autor para identificar um ruído, de uma nota $r_{u,i}$ em relação a nota $p_{u,i}$ prevista pelo algoritmo de predição é maior que um limiar th. As respectivas fórmulas estão exemplificadas abaixo.

$$c(G,T)u,v = \left| \frac{r_{u,v} - p_{u,v}}{r_{max} - r_{min}} \right| \quad (1)$$

Sendo $c(G,T)u,v$ a consistência de uma nota de um usuário u ao item v no algoritmo de predição G com o conjunto de treinamento T, a avaliação $r_{u,v}$, nota prevista pelo algoritmo $p_{u,v}$ e r_{max} e r_{min}, respectivamente, a nota máxima e mínima aceita pelo sistema de recomendação.

$$c(G,T)u,v > th \quad (2)$$

Sendo th o limite escolhido para notas ruidosas. Caso a consistência seja maior que o valor de th, a nota é considerada um possível um ruído.

3.2 Algoritmo R.Toledo

O algoritmo de detecção e correção de ruído de R.Toledo tem como base a ideia de que a personalidade do usuário(ou sua preferência) está relacionada com as avaliações que ele provê para os itens e, consequentemente, os itens teriam um nível de preferência global com base em todas as avaliações destes usuários. Além disso, o algoritmo é divido em duas partes: A parte de detecção de ruído e a parte de correção. Esta última parte é o diferencial do algoritmo do R.Toledo com o O'Mahony.

A detecção do ruído consiste na técnica que utiliza a motivação anteriormente apresentada para verificar se existe uma contradição entre as classificações do usuário, item e a nota da base de treinamento, para determinar se aquela nota pode ser um ruído natural ou não. Desta forma o algoritmo classifica os possíveis ruídos. Para classificar, existe para usuário, item e nota, dois limiares k e v para definir a classificação de cada um. Estes limiares dividem os níveis das notas em três: tendências de notas altas, notas médias e notas baixas, limitadas pelos limiares citados anteriormente. As classificações se sustentam em:

- Usuário:
  - *Benevolente*: $r(u,i) > v_u$;
  - *Médio*: $k_u > r(u,i) \geq v_u$;
  - *Crítico*: $r(u,i) \leq k_u$;
  - *Variável*: Não se encaixa em nenhuma das anteriores.
- Item:
  - *Preferência alta*: $r(u,i) > v_i$
  - *Preferência média*: $k_i > r(u,i) \geq v_i$
  - *Preferência baixa*: $r(u,i) \leq k_i$
  - *Variável*: Não se encaixa em nenhuma das anteriores.
- Nota:
  - *Baixa*: $r(u,i) > v$
  - *Média*: $k > r(u,i) \geq v$
  - *Alta*: $r(u,i) \leq k$
  - *Variável*: Não se encaixa em nenhuma das anteriores.

Como é possível perceber, As classes de cada usuário, item e nota com tendências idênticas são homologas, o que caracteriza como a nota em si deva ser, caso contrário, esta avaliação pode ser considerada como ruído. Por exemplo: suponha uma nota $r(u,i)$ tenha classificação alta; então, pela tendência global desta nota, o usuário deve ser benevolente e o item com preferência alta; caso algumas das classificações sejam diferentes, a nota pode ser um possível ruído. Para que cada classificação seja feita utilizando as informações classificatórias anteriores, cada usuário e item possuirão três conjuntos: W, A e S, representando as notas baixas, médias e altas, respectivamente. Assim, caso a nota analisada seja de um desses grupos, ela é adicionada no conjunto correspondente. Então, o usuário u ou o item i terão uma classificação relacionada ao conjunto que possui maior cardinalidade. Depois de classificados todos os usuários e itens presentes na base de dados considerada, há a verificação de contrariedade das classificações e, todas as notas que possam ser ruído, são armazenadas em um conjunto separado que será utilizado para o algoritmo da correção.

Tendo o conjunto de possíveis ruídos, o algoritmo de correção percorre iterativamente o conjunto destas possíveis notas ruidosas e, para cada uma, realiza uma nova previsão de nota para este usuário u e item i, utilizando o KNN *user-based* com Correlação de Pearson como similaridade e adotando um $k = 60$, considerando a base de treinamento original, ou seja, sem alterações das correções anteriores do algoritmo. Se $abs(novaNota(u,i) - r(u,i)) > \delta$,onde δ é um limiar de aceitação e $abs()$ o valor absoluto, a nota nova é substituída pela atual, consolidando a correção, caso contrário, a nota antiga permanece. No fim do algoritmo, todas as notas que satisfizeram a condição terão sido corrigidas.

Os fatores importantes do algoritmo de R.Toledo são os limiares anteriormente citados, que determinam o comportamento do algoritmo de modo geral. No artigo, há duas abordagens possíveis para tal definição: a perspectiva global e a perspectiva adaptativa. A perspectiva global utiliza as fórmulas 3 e 4 para tentar dividir o intervalo de valores das notas de forma mais igualitária possível:

$$k = k_u = k_i = min\ R + round\left(\frac{1}{3} * (max\ R - min\ R)\right) \tag{3}$$

$$v = v_u = v_i = max\ R + round\left(\frac{1}{3} * (max\ R - min\ R)\right) \tag{4}$$

onde R é o intervalo de avaliações possíveis.

O limiar δ é definido como sendo a menor diferença absoluta que existe entre valores consecutivos dentro do intervalo de valores possíveis das notas. Por exemplo, considerando o intervalo $[1,5]$, o δ seria 1, pois todos os resultados de diferença entre os valores consecutivos do intervalo são iguais($abs(1-2) = 1, abs(2-3) = 1$, etc).

A perspectiva adaptativa obtém a média X' das notas de todos os usuários e a média das notas de todos os itens, fazendo o mesmo para o desvio padrão p'. Assim, os limiares são definidos da seguinte forma:

$$k_u = X'_u - p'_u \tag{5}$$

$$v_u = X'_u + p'_u \tag{6}$$

$$k_i = X'_i - p'_i \tag{7}$$

$$v_i = X'_i + p'_i \tag{8}$$

O cálculo de k, v segue uma perspectiva do usuário($k = k_u$, $v = v_u$) ou do item($k = k_i$ e $v = v_i$). O limiar δ pode ser baseado no usuário($\delta = p'_u$) ou no item($\delta = p'_i$).

4. EXPERIMENTOS

Para a análise dos dados, utilizamos a base de dados do *MovieLens* de 100K de notas dadas por 943 usuários a 1682 filmes. O processo do experimento foi feito considerando uma divisão aleatória da base de dados em 90% para treinamento e 10% para teste, sendo criado 5 divisões diferentes com a mesma configuração, com o objetivo de ter uma variação dos testes realizados. Nas bases de treinamento de cada uma dessas divisões foram gerados, em uma porcentagem definida de notas, ruídos naturais nas notas selecionadas aleatoriamente, consistindo apenas em substituir a nota atual por outra diferente da existente, escolhida de forma aleatória. Esta forma de geração de ruído aqui usada representa o fenômeno imprevisível de como o ruído natural pode aparecer nos dados. Para obter a acurácia de cada aplicação, utilizou-se o *Mean Absolute Error*(MAE), uma métrica com frequente ocorrência na literatura que consiste na em:

$$MAE = \frac{\sum_{u,i \in test\_set} |r_{ui} - \hat{r}_{ui}|}{|test\_set|} \tag{9}$$

onde *test_set* é a base de teste considerada, $\hat{r}_{ui}$ é a nota prevista para o item i do usuário u e r_{ui} é a nota original na base.

Para determinar o MAE final, foi obtida a média aritmética dos 5 resultados das execuções.

O KNN foi executado com a quantidade de vizinhos k = 50, considerando a similaridade como a Correlação de Pearson e adotando o modelo *user-based*(similaridades entre os usuários serão calculadas). Os parâmetros foram decididos empiricamente através de testes realizados com outros parâmetros previamente, observando-os no fim como bons parâmetros. O RSVD possui os seguintes parâmetros: $\lambda = 0.02$, $\gamma = 0.001$ e um critério de parada como sendo 1. Os vetores correspondentes as variáveis latentes(p e q) têm seus valores gerados aleatoriamente e com quantidade de variáveis latentes igual a 100, garantindo que o algoritmo possa alcançar bons resultados para auxiliar nas análises dos experimentos.

No algoritmo de O'Mahony foi adotado o $th = 0.55$ e como algoritmo de previsão o KNN com $k = 35$ e com similaridade como Correlação de Pearson. O th foi definido com base em experimentos pelo próprio artigo de O'Mahony, observando-o com um parâmetro para o bom desempenho do algoritmo. O algoritmo de R.Toledo foi adotada o modelo da perspectiva global, por um motivo semelhante ao th no de O'Mahony: este foi o modelo que alcançou melhores resultados nos experimentos de R.Toledo.

Para analisar o impacto do ruído natural nos algoritmos de recomendação de Filtragem Colaborativa aqui selecionados, a base de treinamento sofreu gerações de ruído natural, como explicado anteriormente, em três níveis: 10%, 20% e 30% para que se possa analisar como a acurácia se comporta a medida que a quantidade de ruído natural aumenta no conjunto de dados. Em seguida, as acurácias destes níveis foram comparadas com as dos algoritmos KNN e RSVD na base de treinamento sem alterações(padrão). Como esperado e como um argumento adicional da hipótese, para todos os níveis os algoritmos de Filtragem Colaborativa sofreram uma piora considerável e os maiores valores de erro consistiram na base ruidosa de 30%, com 11.29% para o KNN e 24.64% para o RSVD. O KNN sofreu menos impacto do ruído que o RSVD, sendo permitido levantar a hipótese de que o RSVD é mais sensível ao ruído que o KNN. Uma explicação lógica para este fenômeno seria o fato do algoritmo baseado em modelo construir a heurística de previsão iterativamente utilizando o conjunto de dados, ocasionando em uma propagação de erro maior e, consequentemente, apresentar uma acurácia menor. Os resultados desta parte do experimento estão representados na Tabela 1.

Table 1: Valores do MAE para cada nível de ruído na base

	Padrão	10%	20%	30%
KNN	0.71702	0.74126	0.76765	0.79796
Diferença		3.38%	7.06%	11.29%
RSVD	0.78250	0.86153	0.92141	0.97531
Diferença		10.10%	17.75%	24.64%

Na comparação dos algoritmos de correção do ruído natural, duas formas experimentais foram adotadas: uma aplicando as correções do ruído na base original(sem alterações ruidosas), representada na Tabela 2, e outra aplicando na base ruidosa em 30%, escolhida por ser o caso de teste com maior quantidade de ruído que eu teoria deverá ser corrigido, mostrada na Tabela 3. As correções de ruído na base original mostraram um desempenho melhor do O'Mahony, para o KNN, e do R.Toledo, para o RSVD. Para o KNN os algoritmos de correção apenas pioraram a acurácia, enquanto para o RSVD ambos obtiveram uma melhora. Nas correções de ruído com o conjunto de dados ruidoso em 30%, para o KNN a diferença com relação a análise da Tabela 2 na piora do O'Mahony e na melhora do R.Toledo(chegando a alcançar uma acurácia melhor que a de referência), enquanto para o RSVD a melhora da acurácia aumentou para as duas abordagens.

Table 2: Valores do MAE para cada abordagem de correção de ruído

	Padrão	Mahony	Toledo
KNN	0.71702	0.72006	0.74059
Diferença		0.42%	3.29%
RSVD	0.78250	0.77719	0.75420
Diferença		-0.68%	-3.62%

5. CONCLUSÃO

A forma como o ruído impacta na previsão dos algoritmos aqui considerados é realmente um fator que merece destaque durante o processo de construção de um modelo de recomendação e na mineração dos dados da base. A medida que a incidência do ruído aumenta, a acurácia e, consequentemente, a representatividade do algoritmo de previsão não alcançam as expectativas esperadas. Além disso, algoritmos baseados em modelo, a exemplo aqui pelo RSVD, tendem a sofrer mais os efeitos negativos do ruído natural do que os baseados em memória, aqui representados pelo KNN.

Table 3: Valores do MAE para cada abordagem de correção de ruído(geração de ruído)

	30%	Mahony	Toledo
KNN	0.79796	0.80583	0.78203
Diferença		0.99%	-2.00%
RSVD	0.97531	0.94732	0.89711
Diferença		-2.87%	-8.02%

Com relação a comparação dos algoritmos de O'Mahony e R.Toledo para o tratamento de ruído e a eficácia de ambos, o algoritmo de R.Toledo obteve um desempenho melhor que o O'Mahony nas duas formas experimentais adotadas nos experimentos, melhorando os MAE's em 3 dos 4 testes, enquanto O'Mahony em 2 dos 4 testes, além dos casos de melhora possuírem uma taxa quase irrelevante. Uma explicação possível é pelo fato de O'Mahony apenas remover as notas ruidosas sem um critério mais rígido, o que ocasiona numa alta probabilidade de uma diminuição da quantidade de dados que poderiam ser usados como informações cruciais para melhorar a acurácia dos algoritmos. Como se pode ver nos resultados, o modelo de R.Toledo teve uma porcentagem maior de melhora das acurácias, levando a crê que algoritmos que além de detectar corrigem possuem um maior sucesso de desempenho de previsão. Há também os casos em que os algoritmos resultaram em uma piora na eficiência, mostrando que ambas as abordagens necessitam de refinamentos em suas metodologias para que diminuam a probabilidade destes casos ocorrerem.

Como trabalhos futuros, serão feitos outros testes variando o valor do limiar de consistência th, para o O'Mahony, e implementar outras abordagens possíveis, para o R.Toledo. Também será feita a avaliação dos mesmos algoritmos em outras bases de dados e uma análise profunda sobre as particularidades de cada algoritmo de minimização de ruído.

6. REFERENCES

[1] C. Desrosiers and G. Karypis. A comprehensive survey of neighborhood-based recommendation method. In *Recommender systems handbook*, pages 107–144. Springer, 2010.

[2] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer Society*, pages 42–49, August 2009.

[3] M. O'Mahony, N. Hurley, and G. Silvestre. Detecting noise in recommender system databases. *ACM International Conference on Intelligent Users Interfaces (IUI)*, pages 109–115, January 2006.

[4] B. Schwartz. *The paradox of choice: why more is less.* Ecco, 2004.

[5] R. Toledo, Y. Mota, and L. Martinez. Correcting noisy ratings in collaborative recommender systems. *Knowledge-Based Systems*, 76:109–115, December 2014.