Relatório do trabalho final da disciplina Recuperação da Informação

Alexandre Costard Soares

Universidade Federal do Rio de Janeiro
alexandrecs@dcc.ufrj.br

Resumo Neste trabalho detalhamos o caso mais abrangente de coleta de dados proposto por Biega *et al.* [1] para operacionalização do princípio de minimização de dados. Para uma amostra de 1000 usuários selecionada do dataset MovieLens 25M, vimos que ainda que 94% dos dados sejam omitidos do recomendador, é possível chegar a 5% do desempenho global de quando o recomendador tem acesso ao dado completo. Contudo, o desempenho para usuários individuais pode ser afetado de forma relevante. Discutimos também as implicações da aplicação desse modelo em sistemas reais. Finalmente, apontamos possíveis desdobramentos diretos para maior exploração.

1 Introdução

Sistemas de recomendação são um exemplo típico de sistemas de recuperação de informação que coletam grande volume de dados pessoais dos usuários, sob a justificativa de melhoria da qualidade do serviço. Contudo, trabalhos recentes mostram que é possível ter recomendações de qualidade sem que o recomendador tenha acesso direto aos dados dos usuários.

Nessa linha por exemplo, temos o trabalho de Singla *et al.* [5], no qual é descrito um esquema no qual usuários decidem quanto de seus dados querem compartilhar, e o de de Biega *et al.* [6], onde é detalhado um sistema intermediário que divide e agrupa as consultas, expondo ao recomendador apenas uma visão de perfis consolidados. Em ambos os trabalhos temos que, apesar das redução dos dados acessíveis ao recomendador, seu desempenho não é significativamente degradado.

Finalmente, em [1] são examinadas diversas maneiras de se amostrar os dados dos usuários para treinar sistemas de recomendação baseados tanto em fatoração de matrizes quanto na vizinhança em similaridade. Seus resultados sugerem que recomendadores baseados em fatoração de matrizes são mais robustos e que poucas estratégias de amostragem conseguem superar uma amostragem aleatória. Esse trabalho abre caminho para pesquisas que busquem responder à pergunta: *quanto dado pessoal é necessário para prover um serviço personalizado de qualidade?*

Essa pergunta é relevante e atual, pois vai ao encontro das legislações sobre de dados pessoais que surgiram na última década, como o Regulamento Geral

sobre a Proteção de Dados (RGPD, em português ou GDPR, em inglês) [2] e a Lei Geral de Proteção de Dados Pessoais [3].

Assim, devido a relevância do tema na nossa realidade, decidimos explorar com mais detalhes o desempenho de um sistema de recomendação baseado em fatoração de matrizes com amostragem aleatória conforme ele é treinado com mais dados, bem como descrever as bases teóricas desse tipo de recomendador.

2 Princípio de minimização de dados

Na última década tem entrado em vigor diversas regulamentações acerca da coleta e processamento de dados pessoais, em geral buscando mudar a propriedade do dado da entidade que faz a coleta para o usuário sobre o qual os dado se refere. Essa mudança visa proteger usuários para que seus dados não sejam usados de forma indesejada, aumentando sua privacidade e garantindo que ele possa revogar o direito de uso dos dados por terceiros. Como exemplos de regulamentações nesse sentido temos a RGPD, na Europa [2], a CCPA, na Califórnia [4] e a LGPD no Brasil [3].

O artigo 5º da RGPD trata dos princípios relativos ao tratamento de dados pessoais. Dentre os princípios destacados, são relevantes para nossos propósitos o princípio da limitação das finalidades, que diz que os dados devem ser recolhidos com fim determinado e explícito, e o princípio da minimização de dados, que diz que os dados devem ser adequados, pertinentes e limitados ao necessário para o fim para o qual ele é tratado.

No contexto brasileiro, temos que os incisos I, II e III do artigo 6º da LGPD determinam os princípios da finalidade, adequação e necessidade, que em conjunto afirmam que a coleta de dados pessoais deve ser limitada e adequada para o fim para o qual eles foram recolhidos, e esse fim deve ser específico e explícito.

Uma questão que pode ser levantada é se dados anonimizados relacionados a avaliações de itens podem ser considerados como dados pessoais. Ambas as regulamentações determinam que podem ser considerados pessoais aqueles dados que podem ser usados para identificação de um indivíduo, ainda que estejam anonimizados. Em [7] foi demonstrado que é possível identificar pessoas no dataset MovieLens 20M quando os dados são cruzados com outros datasets públicos e privados. Temos então que a simples informação de avaliação de filmes consiste sim em dados pessoais sujeitos ao rigor das regulamentações pertinentes.

Apesar de diversos princípios e garantias das regulamentações surgidas na última década terem sidos discutidos na literatura (como o direito de ser esquecido e o consentitimento informado), há poucos trabalhos sobre o princípio de minimização de dados, e em geral eles se encontram mais no âmbito da privacidade dos indivíduos, como em [8], onde é feita uma pesquisa qualitativa sobre como os desenvolvedores levam em consideração esse princípio no contexto de privacidade.

Outra questão pertinente é a explicitação do fim para o qual o dado pessoal é coletado. As regulamentações exigem que esse fim seja específico, e *melhorar a experiência do usuário* é um fim vago demais. Nesse contexto as contribuições

detalhadas em [1] trazem a ideia de uma *minimização baseada em desempenho* com diversas metodologias de coleta. No presente trabalho vamos detalhar o caso mais robusto e abrangente reportado pelos autores.

3 Montagem experimental

3.1 Dataset

Vamos analisar um subconjunto do dataset MovieLens 25M. Devido a limitações de recursos computacionais, amostramos 1000 usuários com pelo menos 200 avaliações cada, totalizando aproximadamento 490 mil avaliações sobre cerca de 20 mil filmes. Cada usuário tem uma mediana de 360 avaliações.

3.2 Protocolo experimental

Para cada usuário do nosso dataset dividimos as avaliações em dois grupos: teste e candidatos a minimização. No grupo de candidatos sorteamos n amostras por usuário para montar um conjunto de dados que será usado para treinar o recomendador. Em seguida comparamos as predições do recomendador com os dados de teste e coletamos métricas para analisar o desempenho.

3.3 Algorítmo de recomendação

Para nossa análise escolhemos usar uma recomendação baseada em fatoração de matrizes, o mais robusto ante minimização dentre os estudados em [1]. O método usado foi FunkSVD [9], que é um método baseado no SVD (decomposição em valores singulares). Esse método de fatoração de matrizes diz que toda matriz M pode ser fatorada como

$$M = U\Sigma V^T \tag{1}$$

onde U e V são matrizes unitárias e Σ é uma matriz diagonal com entradas não negativas. A equação 1 pode ser expandida como

$$M = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \sigma_3 u_3 v_3^T + ... \tag{2}$$

que é uma soma de matrizes de posto 1, onde $\sigma_i > \sigma_{i+1}$ para todo i natural. Uma particularidade dessa fatoração é que caso a série seja truncada em um valor k, teremos como resultado a matriz de posto k com menor erro quadrático em relação a M. Assim, se fizermos a suposição de que a matriz de avaliações tem posto baixo, podemos truncar essa série e obter as avaliações que não estão presentes.

Assim como o SVD tradicional, o algorítmo FunkSVD busca minimizar o erro quadrático das predições em relação as avaliações registradas para uma determinada quantidade de fatores latentes, que no nosso caso podem ser entendidos como gêneros de filmes que surgem a partir dos dados. Contudo, no FunkSVD

as matrizes não são obtidas explicitamente. Em vez disso é usado o método do gradiente para minimizar o erro quadrático. Essa alteração deixa o método mais maleável, possibilitando, por exemplo, garantir que as predições se encontram no intervalo no qual as notas estão, bem como ajustes que garantem as predições para usuário com poucas avaliações não vão distar muito da média.Neste trabalho usamos 30 fatores latentes.

3.4 Medidas coletadas

Para avaliar o desempenho do recomendador, usamos duas medidas: RMSE e nDCG@10. O RMSE, raiz do erro quadrático médio, é calculado como

$$RMSE = \sqrt{\frac{\sum_1^N (r_{ui} - \hat{r}_{ui})^2}{N}} \tag{3}$$

onde r_{ui} é a avaliação que o usuário u deu ao item i e $\hat{r}_{ui}$ é a predição feita pelo recomendador. O RMSE é uma medida útil para avaliar o desempenho geral do recomendador, já que ele leva em consideração todas as avaliações do grupo de teste. Contudo, em sistemas de recomendação o desempenho geral é menos importante do que o desempenho para as recomendações principais. Por isso optamos por também calcular o nDCG@10 para cada usuário. Assim, para cada usuário calculamos para cada usuário o DCG@10 como

$$DCG_{10} = \sum_{i=1}^{10} \frac{2^{rel_i} - 1}{log_2(i+1)} \tag{4}$$

onde rel_i é a relevância do i-ésimo resultado do recomendador e é calculada a partir da normalização da avaliação real que o usuário deu para esse item para intervalo entre zero e um. Optamos por essa versão do DCG com a exponencial no numerador por ela enfatizar a recomendação de items relevantes [10]. O nDCG é obtido pelo quociente do DCG pelo IDCG, como vimos durante o curso. Para a análise global, fizemos a média do nDCG de todos os usuários.

4 Resultados

4.1 Implementação

Devido à diversidade de bibliotecas para análise de dados e sistemas de recomendação, escolhemos implementar os experimentos usando Python. Para o recomendador usamos a biblioteca Surprise e a análise dos dados foi feita usando as bibliotecas Pandas e Numpy. Para acelerar a execução usamos o módulo de multiprocessamento da biblioteca padrão, o que nos permitiu rodar quatro experimentos em paralelo.

Todos os resultados são comparados com o caso base no qual todos os dados de teste são usados para treinar o recomendador.

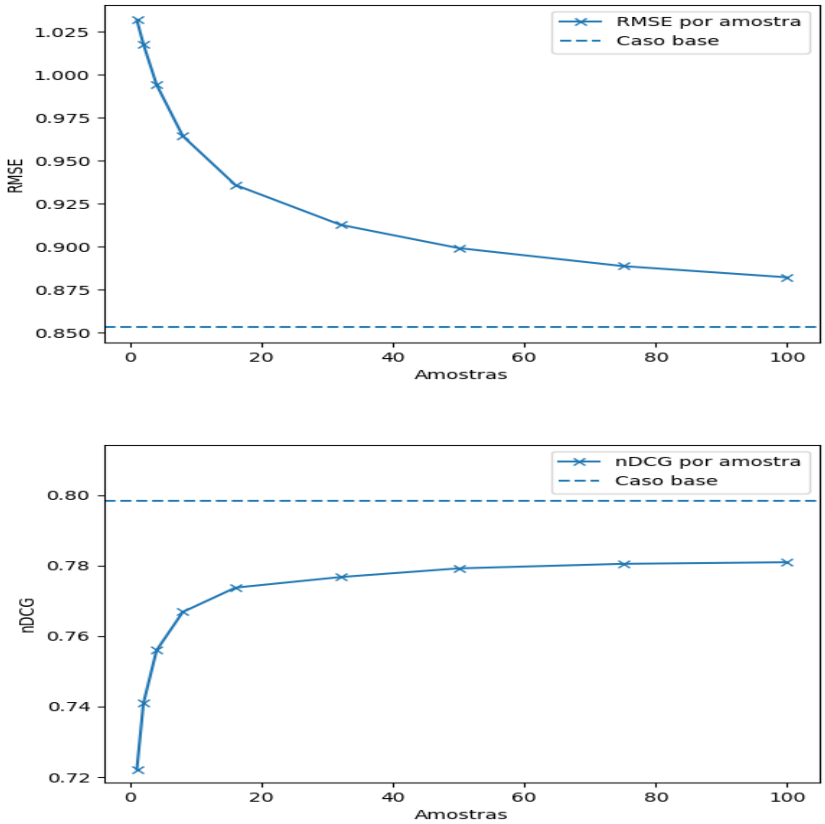

Figura 1. RMSE (acima) e nDCG (abaixo) para cada valor de amostra usado.

4.2 Resultados globais

A primeira análise que fizemos é relativa a quantos dados de cada usuário realmente é necessário para atingir um desempenho ótimo quando comparado com o melhor possível. A figura 1 mostra que a distância entre o desempenho com n amostras e o caso base diminui exponencialmente com a quantidade de amostras usadas para treinar o recomendador.

Vemos também que há uma distância entre o assíntota aparente dos dados e o caso base. Como o dataset utilizado tem um número de avaliações variável por usuário, o caso base treina o recomendador usando um número variável de amostras por usuário. Assim, para alguns são usados 100 amostras enquanto para outros são usadas 2500. Uma análise expedita para os dados acima de 100 amostras por usuário mostra que o comportamento da curva tende rapidamente ao caso base, com diferença inferior a 0.1% quando usamos até 1000 amostras por usuário para treino. Como mencionado anteriormente, o caso base usa até cerca de 2500 amostras por usuário.

4.3 Resultados por usuário

Para analisar o comportamento do recomendador por usuário, decidimos olhar os dados de duas maneiras similares. A primeira consistiu no ordenamento da medida coletada para cada usuário pelo valor da medida. Como antecipado na seção anterior, vemos na figura 2 que o resultado rapidamente se aproxima do caso base.

A segunda abordagem foi observar a métrica por usuário e comparar com o caso base. Diferente dos resultados da figura 2, os usuários nas figura 3 estão ordenados de acordo com o caso base, ou seja, o usuario 1 é o mesmo para todas as amostras. Vemos nesse caso que quanto mais amostras o recomendador se torna mais bem comportado. Mas ainda assim, há usuários para os quais o desempenho do recomendador desvia significativamente do comportamento base.

5 Conclusões

Neste trabalho detalhamos o caso de otimização mais robusto e abrangente apresentado por Biega *et al.*[1]: amostragem aleatória com um recomendador baseado em fatoração de matrizes.

Os resultados obtidos mostram que, apesar de sempre melhorar conforme a quatidade de dados coletada aumenta, o retorno obtido no desempenho do recomendador diminui rapidamente. Isso sugere que, para uma base de usuários grande o suficiente, é possível ter um recomendador com desempenho global extremamente satisfatório mesmo com pouquíssimos dados coletados para cada indivíduo. No nosso modelo, com apenas 15 amostras conseguimos esconder do recomendador 94% dos dados e obter um desempenho com nDCG@10 médio a menos de 5% do que se o recomendador tivesse acesso a todos os dados.

Assim, é possível imaginar um sistema de coleta de dados pessoais no qual a coleta cessa quando o aumento de desempenho por dado coletado está abaixo de

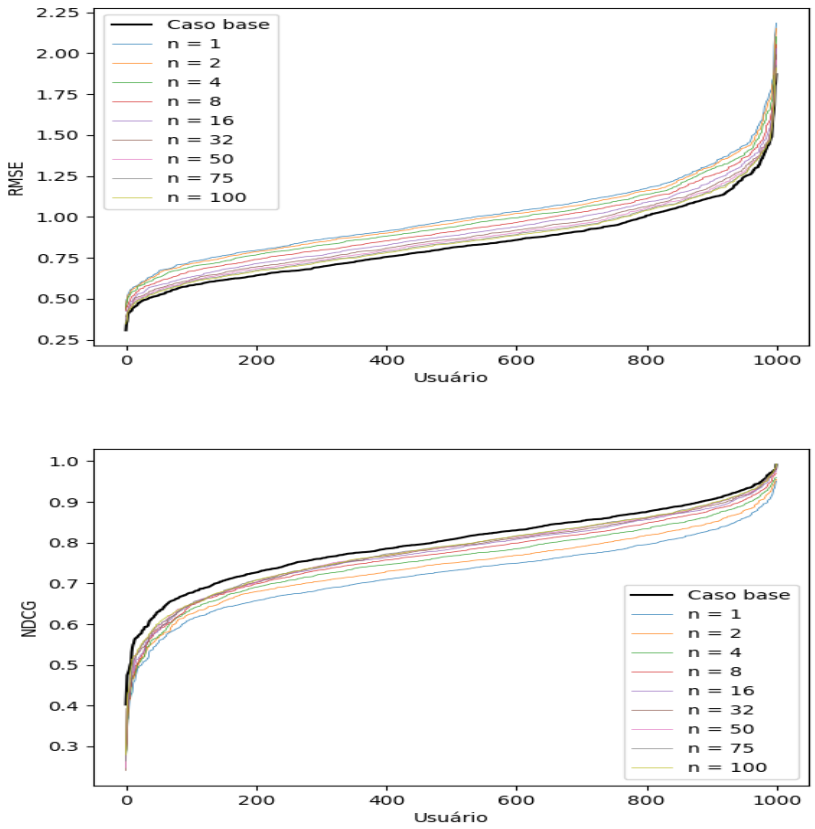

Figura 2. Valor de RMSE (acima) e nDCG (abaixo) por usuário, ordenados por valor.

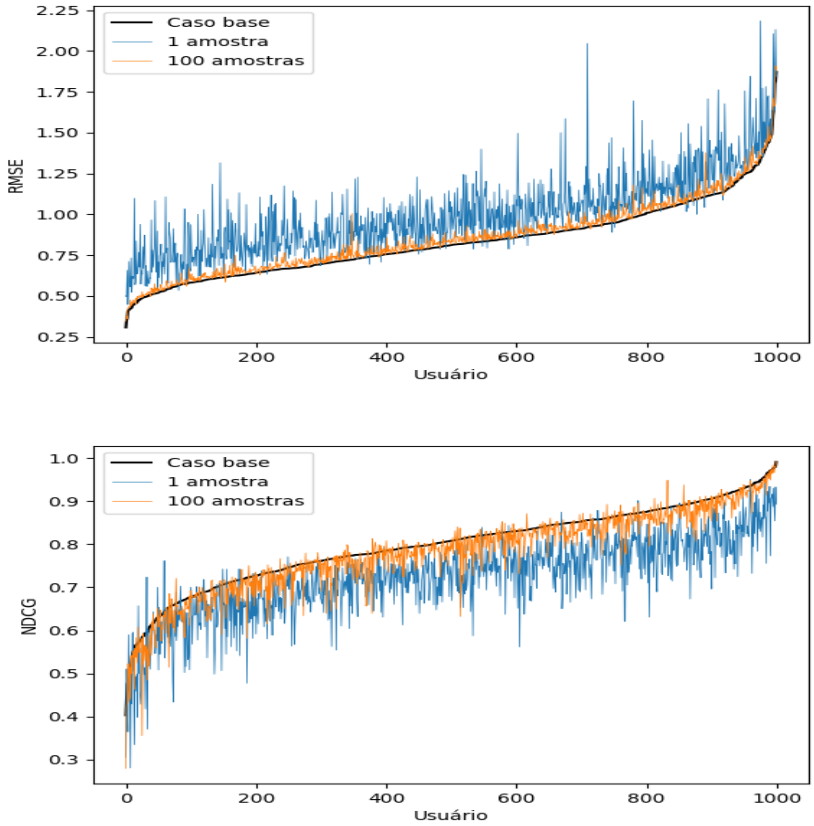

Figura 3. Valor de RMSE (acima) e nDCG (abaixo) por usuário, ordenados pelo valor do caso base. Note que esses dados são exatamente os mesmos da figura 2, apenas ordenados de forma diferente.

um patamar determinado. Como no nosso modelo as preferências de um usuário não variam com o tempo, esse sistema deve periodicamente coletar novos dados para validar seu desempenho. A investigação da evolução temporal desse modelo é um bom trabalho futuro que é possível com o dataset utilizado, já que temos acesso também à data na qual cada avaliação foi feita.

Os resultados obtidos, contudo, mostram que apesar de ter um desempenho global bastante satisfatório com uma quantidade reduzida de amostras, vemos que para alguns usuários o desempenho fica bastante aquém do esperado. Esse resultado sugere que usuários minoritários estariam em uma situação na qual ou o serviço prestado a eles tem qualidade inferior ou deve-se coletar mais dados sobre eles, colocando-os em maior risco em relação a seus dados.

Desdobramentos diretos desse trabalho incluem a investigação com outros datasets usados para recomendação de itens e outros algorítmos de recomendação. Pensamos ser particularmente interessantes algorítmos que maximizam o nDCG, como o descrito por Valizadegan *et al.* [11], mesmo que não seja esperados resultados muito diferentes [1].

Outro eixo de investigação interessante seria a variação da quantidade de usuários no dataset. Quando consideramos que temos n amostras por usuário, na verdade temos um recomendador com $n \times N_u$ amostras. Esse é o motivo do recomendador conseguir fazer predições úteis para um usuário que tem apenas uma avaliação. Esperamos que quanto maior a quantidade de usuários, menor será a quantidade de amostras necessárias para se aproximar do caso base.

Referências

1. Biega, A., Diaz F., Potash P., Finck M., Daumé H.: Operationalizing the Legal Principle of Data Minimization for Personalization https://arxiv.org/abs/2005.13718 Acessado em 25 de outubro de 2020
2. Parlamento Europeu: Regulamento Geral sobre a Proteção de Dados (2016) https://gdprinfo.eu/pt-pt/pt-pt-article-5. Acessado em 14 de novembro de 2020.
3. Brasil: Lei Geral de Proteção de Dados Pessoais (2018) http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/L13709.htm Acessado em 14 de novembro de 2020.
4. California: California Consumer Privacy Act (2018) https://www.oag.ca.gov/privacy/ccpa. Acessado em 14 de novembro de 2020.
5. Singla A., Horvitz E., Kamar E., White R.: Stochastic privacy (2014) https://arxiv.org/abs/1404.5454 Acessado em 4 de novembro de 2020.
6. Biega A., Roy R., Weikum G.: Privacy through solidarity: A utility-preserving framework to counter profiling (2017). Proceedings of the 40th International ACM SIGIR Conference on Research and Development inInformation Retrieval. ACM, 675–684.
7. Eslami S., Biega A., Roy R., and Gerhard Weikum: Privacy of hidden profiles: Utility-preserving profile removal in online forums (2017). Proceedings of the 2017 ACM on Conference on Information and KnowledgeManagement. ACM, 2063–206.
8. Senarath A., Arachchilage N.: Understanding Software Developers' Approach towards Implementing Data Minimization https://arxiv.org/abs/1808.01479 Acessado em 14 de novembro de 2020.

9. Funk S.: Netflix update: Try this at home (2006).
http://sifter.org/ simon/journal/20061211.html Acessado em 25 de outubro de 2020.
10. Burges C., Shaked T., Renshaw E., Lazier A., Deeds M., Hamilton N., Hullender G.: Learning to rank using gradient descent. (2005) In Proceedings of the 22nd international conference on Machine learning (ICML '05). ACM, New York, NY, USA, 89-96.
11. Valizadegan H, Jin R., Zhang R., Mao J.: Learning to Rank by Optimizing NDCG Measure (2009)
https://papers.nips.cc/paper/2009/file/3967a0e938dc2a6340e258630febd5a-Paper.pdf Acessado em 15 de novembro de 2020.