

6 Conclusão e Trabalho Futuro

Hoje em dia temos uma quantidade enorme de informação disponível na Internet. Os sistemas de busca nos ajudam razoavelmente a encontrar as informações que procuramos. No entanto, o mais eficaz seria ter algo nos sendo oferecido diretamente sem a necessidade de procurarmos o conteúdo em questão. Inclusive, desde uma simples lista com os links mais acessados nos sites de notícias até recomendações personalizadas que recebemos nos principais sites de comércio eletrônico, estamos bastante acostumados a receber recomendações na Web.

Recomendações podem ser geradas por uma grande diversidade de algoritmos. Enquanto modelos de filtragem colaborativa baseados em itens e usuários são simples e intuitivos, técnicas de fatoração matricial são geralmente mais eficientes pelo fato de conseguirem demonstrar os fatores latentes que não costumam estar explicitados nas relações entre os usuários e os itens.

Nesta dissertação apresentamos algumas abordagens de fatoração matricial para o problema de recomendação de vídeos. Para isso, utilizamos os registros de vídeos vistos nos catálogos da Globo.com como dados implícitos de um modelo incremental de fatoração matricial visando encontrar outros vídeos que pudessem ser oferecidos para usuários com gostos parecidos.

Diferente dos casos onde temos um retorno explícito sobre a preferência dos usuários, nosso algoritmo de fatoração latente deveria utilizar todos os pares usuário-vídeo como entrada, incluindo aqueles que não estão relacionados a nenhuma observação (indicado como um zero em nosso modelo). Isto é bastante crítico, dado que a forma como o algoritmo foi implementado faz com que as preferências dos usuários sigam uma tendência positiva com relação a todos os itens, não refletindo verdadeiramente seu perfil.

No entanto, usar todos os pares usuário-vídeo como entrada do sistema nos levaria a sérios problemas de escalabilidade. A quantidade de pares tende a ser muito maior que o conjunto de entrada trabalhado, dado que a quantidade de vídeos que um usuário típico assiste é apenas uma pequena fração do total de vídeos em questão. Endereçamos esta questão explorando a estrutura algébrica do

modelo, chegando a um algoritmo de implementação simples e com um tempo de execução que escala linearmente com o tamanho da entrada (dados de visualizações e quantidade de variáveis latentes).

Vimos que muitas das variações observadas na preferência dos usuários por alguns vídeos se deve a características dos próprios usuários ou vídeos independentemente de qualquer interação entre eles através da adição de um componente de *bias* ao modelo de aprendizado. Além disso, procuramos entender como a inclusão da evolução temporal da popularidade dos vídeos tende a melhorar significativamente o resultado das previsões.

Também mostramos que um dos principais pontos negativos do método utilizado é a quantidade de parâmetros que precisamos ajustar para chegar ao resultado almejado. Encontrar o conjunto de valores ideal para os parâmetros em questão é uma tarefa que nos consumiu grande parte do tempo do projeto. Os experimentos mostraram também que o *overfitting* causado pela escolha errada destes parâmetros tende a ser um sério problema. Valeria investir mais tempo analisando outras técnicas de regularização que pudessem minimizar o efeito deste problema.

Dado que o projeto de um sistema de filtragem colaborativa depende fortemente dos dados que alimentam o mesmo, como o tamanho do conjunto de dados, seus requisitos de complexidade, entre outros, enxergamos como uma possível melhoria na continuidade deste trabalho a inclusão de outras fontes de informação sobre o conteúdo e as características dos usuários em questão. Isso poderia endereçar o problema de *cold-start*, onde queremos prover recomendações para usuários sem um histórico prévio de visualizações de vídeo.

Uma outra proposta de continuidade deste trabalho é a paralelização do passo de aprendizado do algoritmo de fatoração. Dependendo do tamanho do conjunto de dados de entrada, a construção do modelo de recomendação pode tomar muito tempo. Alguns estudos [36, 37] relatam ganhos de performance expressivos a partir de algoritmos de aprendizagem paralelizáveis como o *Alternating Least Squares*.

Por fim, achamos que a adição de outras técnicas de predição ao sistema, em questão, como por exemplo algoritmos de vizinhança ou modelos gráficos

tende a melhorar a precisão das recomendações. A combinação de diferentes modelos de filtragem colaborativa se mostrou bastante eficiente durante a competição *Netflix Prize*.