

Um modelo baseado na evolução temporal de consumo e sua aplicação em domínios de recomendação

Camila Souza Araújo, Fernando H. Mourão, Wagner Meira Jr. ¹

¹Departamento de Ciência da Computação – Universidade Federal de Minas Gerais (UFMG)

{camilaaraujo, fhmourao, meira}@dcc.ufmg.br

Resumo. *Em domínios de recomendação o gosto dos usuários, bem como o próprio domínio, varia ao longo do tempo. Porém, essa evolução é pouco compreendida na literatura. Um maior entendimento deste processo permitiria melhorar as recomendações. Neste trabalho, modelamos a evolução temporal de forma a identificar itens potencialmente relevantes para recomendação. Utilizamos o conceito de transições evolutivas representativas, transições entre quaisquer par de itens ao longo do tempo, e extraímos tais transições através da modelagem proposta. Além disso, realizamos experimentos para validar nossa premissa de que transições evolutivas representativas existem e são: mensuráveis, relevantes, úteis e não óbvias.*

1. Introdução

Atender necessidades ou desejos específicos de cada usuário vem se tornando uma tarefa cada vez mais desafiadora em diversos domínios, dado o crescente número de opções existentes. Sistemas de Recomendação (SsR) são ferramentas utilizadas para orientar o usuário no processo de escolha e reduzir esse número excessivo de opções [Burke 2002]. SsR geram recomendações personalizadas a partir de informações, tais como o conjunto de itens consumidos pelo usuário, o conjunto de itens consumidos por usuários similares a ele ou características dos itens do próprio usuário no sistema (e.g., idade no sistema, frequência de consumo, dentre outras).

Uma vez que o gosto dos usuários varia ao longo do tempo, domínios de recomendação são ambientes dinâmicos. O próprio domínio evolui a todo momento, pois novos usuários e itens podem surgir, e os antigos podem desaparecer. Saber o que os usuários gostam a cada momento, a partir do que há disponível, é um dos grandes desafios em recomendação atualmente. Apesar da relevância dessa evolução temporal, não encontramos estudos que almejem entender mais a fundo seu impacto sob as recomendações. A maior parte dos estudos existentes se limitam a aceitar tal evolução como uma premissa e propor mudanças necessárias para que o sistema se adapte a distintos momentos no tempo. Contudo, um maior entendimento deste processo nos permitiria melhorar recomendações, aumentando a capacidade de SsR resgatarem itens potencialmente interessantes. Por exemplo, o resgate de itens já consumidos pelo usuário no passado, mas que podem vir a ser reconsumidos, cria uma nova possibilidade de recomendação. Em [Mourão et al. 2011], os autores abordam esse problema no cenário de recomendação e apresentam o conceito de *itens esquecidos*: qualquer item que tenha sido relevante para um determinado usuário no passado, mas que não o é no presente.

Neste trabalho, almejamos modelar essa evolução temporal de forma a identificar itens potencialmente úteis para recomendação, que não são explorados por SsR atuais.

Com essa modelagem buscamos encontrar o que denominamos de ‘transições evolutivas representativas’ no consumo dos usuários. Definimos transição evolutiva como um par ordenado de itens consumidos, pelo mesmo conjunto de usuários, em ordem cronológica ao longo do tempo. Para serem representativas, tais transições devem possuir quatro características: serem mensuráveis, relevantes, não óbvias e úteis. Nossa principal hipótese é que essas transições existem e podem ser utilizadas tanto para entender o comportamento do usuário ao longo do tempo quanto melhorar recomendações SsR.

Nossa proposta é definir e avaliar uma nova modelagem probabilística do consumo de itens pelos usuários em domínios de recomendação ao longo do tempo. O modelo é baseado na construção de uma matriz de transição temporal utilizando processos markovianos. Todas as implementações, execuções dos experimentos e avaliação dos resultados foram realizadas pela aluna Camila Araújo, sob a supervisão dos orientadores. A concepção do modelo proposto e dos algoritmos contou com grande colaboração do orientador Fernando Mourão.

2. Referencial Teórico

Nesta seção, apresentamos alguns dos principais conceitos para o entendimento do problema abordado e da solução adotada. Além disso, discutimos sucintamente alguns dos trabalhos relacionados mais relevantes, bem como diferenciamos nosso trabalho dos demais e explicitamos sua contribuição para a área de recomendação.

2.1. Sistemas de Recomendação

O crescente volume de informação em ambientes sociais virtuais torna cada vez mais difícil a sua organização. Em seu livro *The Paradox of Choice: Why More is Less* [Schwartz 2005] menciona que:

À medida que aumenta o número de opções, o esforço exigido para tomar uma decisão acertada também aumenta; esse é um dos motivos pelos quais a escolha pode deixar de ser uma vantagem para se transformar em um ônus. (pág. 68)

Segundo [Schwartz 2005], como consumidores, quando nos deparamos com uma grande variedade de opções, ficamos indecisos e paralisados. Esse foi o cenário ideal para o surgimento de SsR, que são considerados por [Burke 2002] como quaisquer sistemas que tenham por objetivo produzir como saída recomendações personalizadas, ou seja, produzir o efeito de orientar o usuário de forma individualizada diante de uma grande variedade de opções, aproximando o usuário do conteúdo realmente relevante.

2.2. Random Walk

Um Random Walk é uma cadeia de Markov finita e pode ser utilizado para determinar a probabilidade de, dado um grafo, se alcançar o nodo X a partir do nodo Y em N passos. Dado um ponto de partida em um grafo, selecione um nodo vizinho de forma aleatória e se mova para essa nova posição; a partir deste novo ponto, selecione outro vizinho de forma aleatória e repita o processo. Como podemos ver em [Lovász 1996], essa sequência de pontos selecionados aleatoriamente é um Random Walk em um grafo. Utilizaremos o Random Walk para estimar as probabilidades de alcance entre itens do sistema.

2.3. Trabalhos Relacionados

A crescente importância da área de SsR tem impulsionado várias pesquisas sobre o tema [Ricci et al. 2011]. [Adomavicius and Tuzhilin 2005] afirmam que, embora existam inúmeras propostas de SsR, as recomendações ainda requerem melhorias para serem mais eficazes e aplicáveis a uma ampla gama de cenários do mundo real, tais como recomendação de viagens, serviços financeiros, dentre outros. As dificuldades para fornecer boas recomendações estão relacionadas aos vários desafios inerentes à tarefa de recomendação, que vão desde a escolha de quais dados devem ser usados para representar o comportamento do usuário até a seleção da melhor técnica para fornecer as recomendações.

Um dos principais desafios inerente aos SsR, e abordado neste trabalho, é a evolução temporal. Basicamente, os trabalhos sobre a evolução temporal em SsR podem ser classificados em dois grupos. O primeiro inclui estudos que avaliam a qualidade das recomendações ao longo do tempo [Campos et al. 2011]. Por exemplo, em [Zhang and Hurley 2008] os autores avaliam como a diversidade nas recomendações é afetada ao longo do tempo. No segundo grupo, temos trabalhos que propõem novos modelos de recomendação, que levam em conta a evolução temporal [Cremonesi and Turrin 2010]. Os estudos existentes, em sua maioria, se limitam a aceitar tal evolução como uma premissa e propor mudanças necessárias para que o sistema se adapte a distintos momentos no tempo. Nosso trabalho, por sua vez, busca entender como essa evolução se manifesta e como o comportamento do usuário no passado, ou até mesmo um item antigo, pode ser útil para novas recomendações.

3. Metodologia

Nesse trabalho propomos uma nova modelagem do consumo de itens por usuários de SsR ao longo do tempo. Tal modelagem visa identificar, em distintos domínios, transições evolutivas de consumo, bem como identificar possibilidades de recomendação a partir dessas transições. O framework da figura 1 ilustra o fluxo dos dados e as etapas necessárias, partindo da análise do consumo até o método de recomendação, para a entrega das recomendações personalizadas para o usuário.

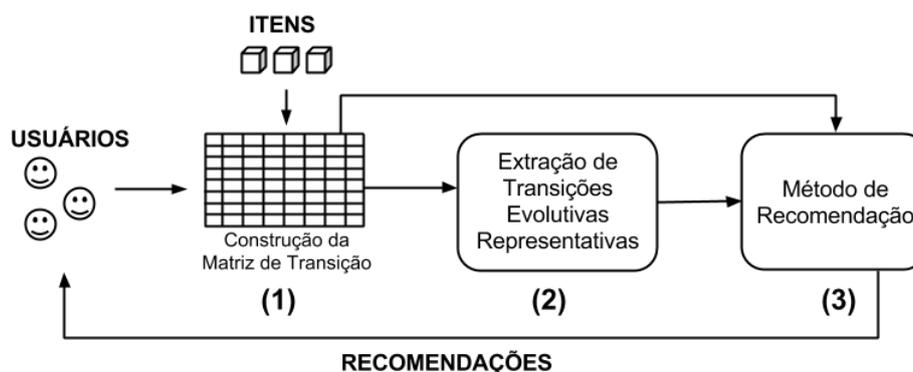


Figura 1. Framework.

Entendemos que esse processo como um todo é um problema complexo, por isso, restringimos nosso trabalho no desenvolvimento da primeira e da segunda etapa, que serão

detalhadas nas próximas seções. Deixaremos a etapa (3), de recomendação, para trabalhos futuros.

3.1. Construção da Matriz de Transição Temporal

Para agregar os dados de consumo dos usuários, modelamos o sistema como uma matriz de transição temporal. Primeiro, utilizamos a equação (1) para calcular a probabilidade de transição $T_t(x, y)$ entre pares de itens x e y em momentos adjacentes no tempo, em que t é o momento no tempo e $u_{t,x}$ é o conjunto de usuários que consumiram o item x no tempo t .

$$T_t(x, y) = \frac{|u_{t,x} \cap u_{t+1,y}|}{|u_{t,x}|} \quad (1)$$

A equação (2) representa a probabilidade de ocorrência de um item x no tempo t , ou seja, o número de usuários que consumiram o item x no tempo t dividido pelo número total de usuários ativos no tempo t . Por último, as duas últimas equações são utilizadas para a construção da matriz de transição temporal M (3). A matriz M é normalizada, linhas somam 1, para que seja irredutível¹.

$$P_t(x) = \frac{|u_{t,x}|}{|total|} \quad (2)$$

$$M[x, y] = \sum_{t=0} P_t(x) * T_t(x, y) \quad (3)$$

É importante ressaltar que, como os usuários deixam de consumir um conjunto de itens para consumir outros, modelar seu comportamento a partir das transições entre itens $x \rightarrow y$ é inserir a evolução temporal do domínio ao nosso modelo. Na figura 2 representamos um sistema com três semanas (unidade de tempo escolhida) e três usuários em um cenário musical. Para esse sistema, temos a matriz M (4), assim como a matriz irredutível, gerada pela equação (3).

$$\begin{bmatrix} 0.666 & 0.666 & 0 & 0 & 0.666 \\ 0 & 0 & 0.333 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0.333 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0.3333 & 0.3333 & 0 & 0 & 0.3333 \\ 0 & 0 & 1 & 0 & 0 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\ 1 & 0 & 0 & 0 & 0 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \end{bmatrix} \quad (4)$$

Para exemplificar um modelo que não utiliza as informações temporais do sistema, vamos considerar uma matriz de transição gerada utilizando as probabilidades condicionais entre os pares de itens. Aqui, utilizamos apenas os pares de músicas que ocorrem em semanas subsequentes para calcular as probabilidades condicionais. Por exemplo, se

¹Não existem estados absorventes no grafo, em outras palavras, não existe nenhuma linha da matriz com todos os valores iguais a zero.

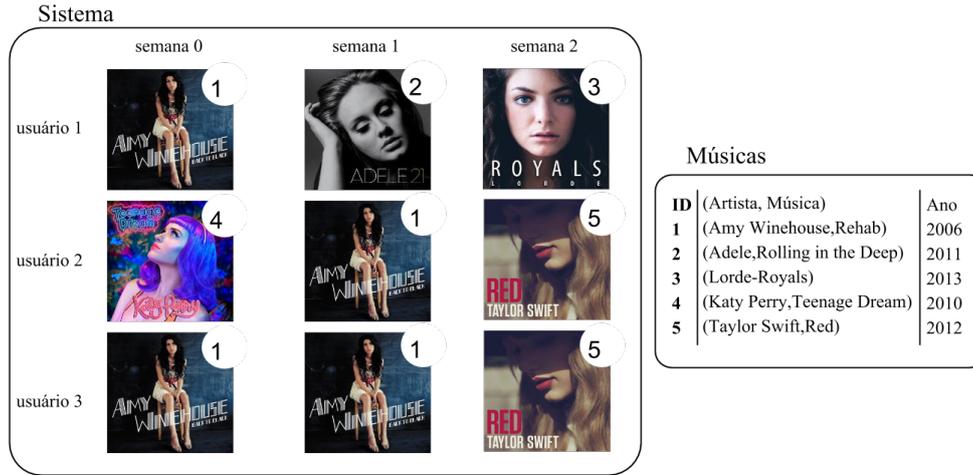


Figura 2. Sistema com 3 usuários.

a música ‘Red’ ($ID = 5$) está sendo consumida, a probabilidade de ‘Rehab’ ($ID = 1$) ter sido consumida é igual a 1. A matriz de transição e a matriz irredutível, gerada são:

$$\begin{bmatrix} 0.333 & 1 & 1 & 1 & 1 \\ 0.333 & 0 & 1 & 0 & 0 \\ 0.333 & 1 & 0 & 0 & 0 \\ 0.333 & 0 & 0 & 0 & 0.5 \\ 0.666 & 0 & 0 & 1 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0.0769 & 0.2308 & 0.2308 & 0.2308 & 0.2308 \\ 0.2498 & 0 & 0.7502 & 0 & 0 \\ 0.2498 & 0.7502 & 0 & 0 & 0 \\ 0.3998 & 0 & 0 & 0 & 0.6002 \\ 0.3998 & 0 & 0 & 0.6002 & 0 \end{bmatrix} \quad (5)$$

Considerando as matrizes geradas como grafos de transição, onde o número da linha representa o nodo de saída da aresta e a coluna o nodo de chegada, podemos perceber que nosso modelo mantém a informação de transitividade temporal entre os itens. Ao utilizar apenas o histórico do usuário, percebemos que as transições geradas dependem muito mais da frequência dos itens no sistema. A música ‘Rehab’, por exemplo, por ser frequente, possui arestas de entrada e saída para todas as outras músicas.

3.2. Extração das Transições Evolutivas Representativas

3.2.1. Random Walk

Para a construção do nosso modelo, o algoritmo Random Walk será utilizado para estimar a probabilidade de se partir de um item x qualquer e alcançar outro item y em n iterações, sendo que cada iteração representa uma unidade temporal no sistema. Visando otimizar nosso processo, modificamos o algoritmo para que a cada iteração as probabilidades menores que a média das probabilidades, ou seja, as arestas que possuem um valor menor que o esperado, sejam removidas. O resultado de cada iteração do Random Walk é dado pela matriz resultante R_i , onde i é o valor da iteração. Cada par $R_i[x, y] > 0$ representa uma transição evolutiva possível com i iterações no sistema.

Algoritmo 1: executeRandomWalk

```
matriz de transição;  
matriz resultante = matriz de transição;  
iteração = 0;  
while (valores não convergem) and (matriz resultante existe) do  
    iteração++;  
    matriz resultante = matriz resultante * matriz de transição;  
    imprime resultado da iteração corrente;  
    limite inferior = média dos valores diferentes de zero;  
    remove probabilidades menores que o limite inferior e redistribui esses valores  
    entre os outros valores da linha;  
    remove linha/colunas zeradas;  
end
```

Além do Random Walk, também calculamos o valor de PageRank, $PR(x)$, para cada item. O PageRank, assim como o Random Walk, utiliza o peso dos links de entrada e saída de cada nodo no grafo. Porém, como explicado em [Chung and Zhao], "instead of having to determine the number of steps a random walks is taking, PageRank uses a positive real value α , where $\alpha \in [0, 1)$ to control the 'diffusion' of a combination of random walks". Então, ao invés de retornar a probabilidade de alcance entre dois itens ele estima a importância, no nosso caso, a popularidade, de cada item para o sistema.

3.2.2. Transições Evolutivas Representativas

Como definimos anteriormente, uma transição evolutiva é um par de itens ordenado de acordo com o momento em que cada um foi consumido pelo mesmo conjunto de usuários. Além disso, como buscamos transições evolutivas representativas, elas devem ser: mensuráveis, para que seja possível medir sua relevância em relação as outras transições existentes; relevantes, com probabilidade de ocorrência significativamente mais alto que o esperado; não óbvias, consideramos óbvias transições que envolvam apenas itens populares, pois os mesmos são facilmente alcançáveis sem o auxílio de SsR; úteis para recomendação, que sejam aplicáveis à um conjunto maior de usuários.

O processo de identificação das transições evolutivas, com a utilização do Random Walk modificado, garante que elas são mensuráveis, uma vez que um valor é atribuído a cada transição possível. Ordenando todos os valores de $R_i[x, y]$, ou posteriormente $P_{TAN}(i, x, y)$, podemos identificar as transições mais relevantes, ou seja, com probabilidade maior que a esperada. Para identificar as transições não óbvias e úteis, vamos considerar uma transição do tipo $x \rightarrow y$, isto é, $R_i[x, y]$. Partindo de qualquer origem a probabilidade de se atingir uma musica popular é alta, por isso dizemos que se a popularidade de y no sistema for alta a transição é uma transição óbvia. Por outro lado, se a probabilidade de x for baixa a transição não é útil, pois a transição $x \rightarrow y$ não é aplicável à muitos usuários.

$$P_{TAN}(i, x, y) = \frac{PR(x) * R_i[x, y]}{PR(y)} \quad (6)$$

A equação (6) representa a probabilidade da transição absoluta normalizada pela

relevância global do destino no domínio. P_{TAN} nos permite extrair as transições evolutivas representativas do domínio de recomendação, ao atribuir um valor para cada transição em função do produto da probabilidade de alcance entre itens, $x \rightarrow y$, obtida pelo Random Walk, transições relevantes, e a popularidade do item x , transições úteis, obtida pelo PageRank. Esse valor é normalizado pela popularidade de y , transição não óbvia.

3.3. Recomendação

Nossa motivação é que a existência de transições evolutivas possa ser útil na resolução de alguns problemas inerentes em SsR, como: (1) o problema do Cold Start, que ocorre quando o sistema possui usuários com pouca informação, ou itens poucos consumidos. A acurácia do recomendador é comprometida, uma vez que ele não possui um histórico extenso sobre esses usuários, ou itens, e não existe informação o suficiente para boas recomendações; (2) o problema do reconsumo, ou seja, identificar o subconjunto de itens que o usuário já consumiu e que eventualmente gostaria de reconsumir e (3) o fenômeno de Cauda Longa, um comportamento típico em SsR no qual o consumo fica concentrado em poucos itens populares, enquanto os demais itens são pouco consumidos e recomendados. Nesse sentido, podemos utilizar as transições evolutivas relevantes encontradas da seguinte forma:

1. Ao identificar que existe uma transição evolutiva relevante do tipo usuários que consumiram um item A, também consumiram um item B após um certo tempo, podemos utilizá-la para tentar recomendar o item B para os usuários que gostam de A e não consumiram B. Essa é uma solução para o problema do Cold Start, pois mesmo sem um histórico de consumo extenso é possível fazer recomendações para novos usuários.
2. Caso essa transição permaneça relevante ao longo do tempo, podemos fazer essa mesma recomendação mesmo para os usuários que já consumiram B, pois existe uma chance que essa seja uma boa oportunidade de reconsumo.
3. Partindo da nossa premissa - existem transições evolutivas relevantes e que elas são não óbvias - também podemos atenuar o problema da Cauda Longa ao resgatar itens da cauda.

4. Avaliação Experimental

O domínio de recomendação utilizado para o trabalho foi um domínio musical. A coleção de dados utilizada foi uma base extraída do sistema Last.fm. A Last.fm é uma comunidade virtual focada em música; os usuários possuem um perfil e podem se conectar aos seus amigos, receber recomendações de músicas, eventos e álbuns. Nossa amostra contém informações sobre a execução das músicas ao longo das semanas pelos usuários. A base de dados contém informações sobre 85.075 usuários e 12.585.511 músicas em um período de análise de 220 semanas distintas.

O único parâmetro de execução que precisamos definir manualmente é o número de iterações utilizado pelo Random Walk e pela equação P_{TAN} , o valor i . Como o sistema utilizado nos nossos experimentos é dividido temporalmente por semanas, contextualmente, i representa o número de semanas. Ou seja, quando temos a transição $R_i[x, y]$ dada pelo Random Walk, no nosso cenário, podemos dizer que temos a probabilidade de alcance entre a música x e y em i semanas. Nos nossos experimentos, utilizamos $i = 1$

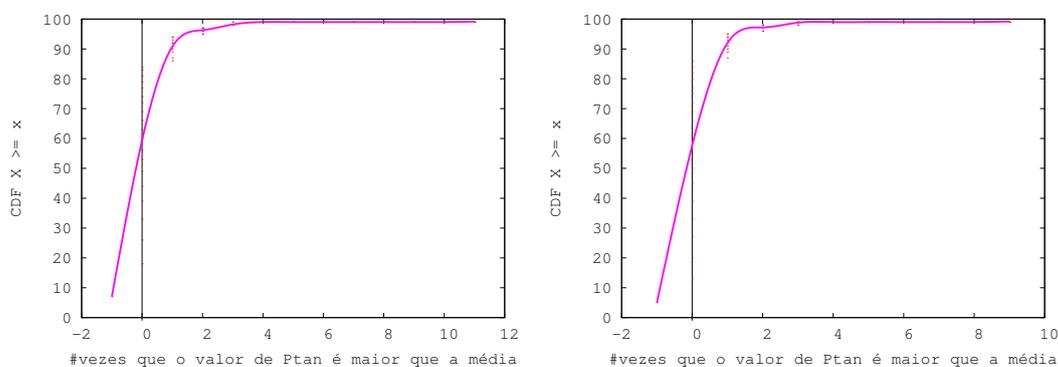
e $i = 24$. Escolhemos esses dois valores, 1 semana e 6 meses, para podermos avaliar como os resultados obtidos com nossa metodologia se mantêm estáveis, mesmo quando aumentamos a distância temporal utilizada.

4.1. Resultados

O objetivo dos nossos experimentos é verificar a existência das transições evolutivas e avaliá-las para validar nossa hipótese de que essas transições evolutivas representativas são: mensuráveis, relevantes, úteis e não óbvias. Após gerarmos todas as transições evolutivas a partir da fórmula P_{TAN} , com $i = 1$ e $i = 24$, analisamos algumas informações gerais sobre elas com a finalidade de mostrar que realmente existem transições relevantes.

	# Regras	Média	Desvio	# Regras acima da média
$P_{TAN}(1, x, y)$	3.233.296	0,00044	0,00046	1.228.875 (38%)
$P_{TAN}(24, x, y)$	3.286.960	0,00042	0,00039	1.283.590 (39%)

Observando a média dos valores encontrados, para $i = 1$ e $i = 24$, percebemos que existe um número significativo de transições com probabilidades maiores que o valor esperado (38% e 39%, respectivamente). Além disso, plotamos uma curva de distribuição acumulada (CDF) desses valores utilizando como parâmetro de comparação o valor médio encontrado. Assim, o valor plotado no eixo x representa, na verdade, quantas vezes a probabilidade de uma transição está acima da média (60%). Os gráficos da figura 3 também demonstram que nossa hipótese - a existência de transições evolutivas relevantes - está correta. Uma vez que conseguimos observar que uma porcentagem significativa de transições tem probabilidade maior que a média.



(b) CDF dos valores de P_{TAN} para 1 semana. (c) CDF dos valores de P_{TAN} para 24 semanas.

Figura 3. Distribuição acumulada das probabilidades.

Com o objetivo de verificar que existem transições úteis e não óbvias, realizamos a segunda parte da nossa avaliação focados nos valores de $PR(x)$ e $PR(y)$. Vimos que uma transição é óbvia quando envolve apenas itens populares. Por isso, buscamos por transições $x \rightarrow y$ onde o valor de $PR(y)$ é mais baixo. Por outro lado, o valor de $PR(x)$ não pode ser muito baixo, caso contrário a transição não é abrangente. Para a comparação dos valores encontrados, plotamos uma CDF dos valores de PageRank de todas as músicas do sistema (figura 4). Na tabela abaixo, podemos verificar o valor médio de $PR(x)$ e $PR(y)$, além da correlação² desses valores com o valor obtido por $P_{TAN}(i, x, y)$.

²Coefficiente de correlação de Pearson.

	Correlações		Médias	
	PR(x)	PR(y)	PR(x)	PR(y)
$P_{TAN}(1, x, y)$	0,5215	0,1146	0,0005448	0,0002376
$P_{TAN}(24, x, y)$	0,6127	0,2781	0,0005453	0,0002388

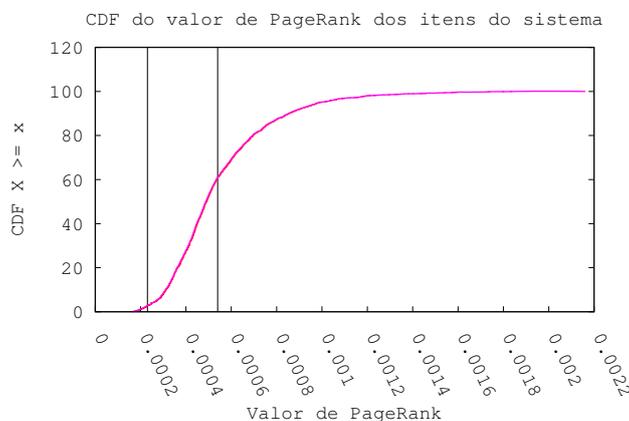


Figura 4. CDF dos valores de PageRank.

Verificamos que nenhum dos valores está altamente correlacionado e $PR(x)$ apresenta valores maiores que $PR(y)$ em ambas as métricas. Verificamos também que, na média, $PR(y)$ não é muito alto nas transições encontradas, uma vez que menos de 10% dos itens possuem o valor de PageRank menor que 0,00023. Por outro lado, pelo menos 60% dos itens possuem PageRank maior que 0,00054. Além disso, plotamos um gráfico de dispersão das duas variáveis para as transições mais relevantes encontradas. Neste gráfico, percebemos que os valores de $PR(x)$ tendem a ser maiores que $PR(y)$.

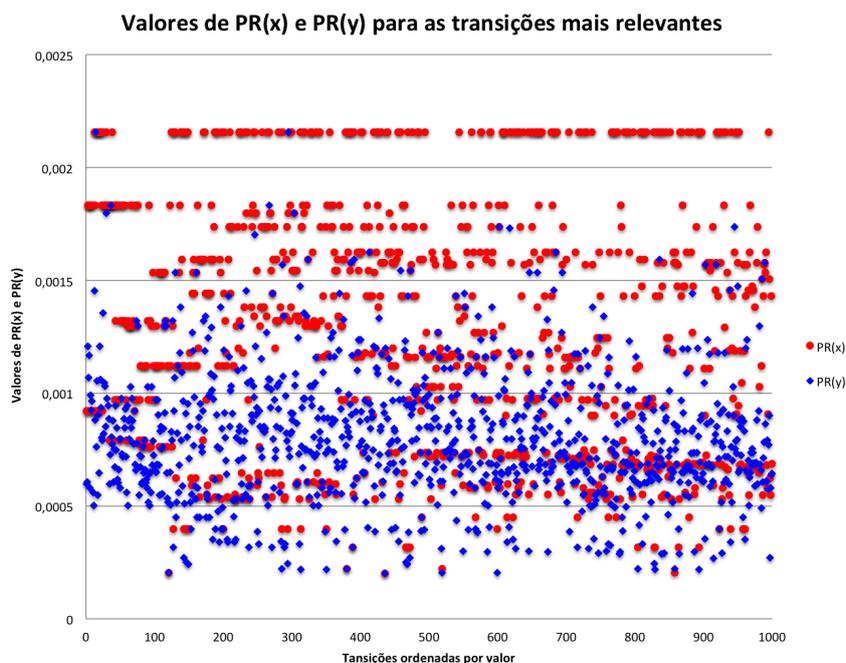


Figura 5. Gráfico de dispersão dos valores PR(x) e PR(y) para as transições mais relevantes.

5. Conclusão e trabalhos futuros

Neste trabalho, apresentamos uma nova modelagem probabilística capaz de agregar informações sobre a evolução temporal do sistema. A partir da modelagem proposta, extraímos o que denominamos de transições evolutivas, um par ordenado de itens consumidos, pelo mesmo conjunto de usuários, em ordem cronológica ao longo do tempo. Nossos experimentos demonstraram que essas transições existem e são relevantes, ou seja, são mensuráveis, relevantes, úteis e não óbvias. Acreditamos que tais transições serão úteis para o aprimoramento do processo de recomendação. Por isso, como trabalhos futuros, pretendemos explorar as transições encontradas para aumentar a capacidade de SsR de resgatarem itens potencialmente interessantes.

Referências

- Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749.
- Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370.
- Campos, P. G., Diez, F., and Sanchez-Montanes, M. A. (2011). Towards a more realistic evaluation: testing the ability to predict future tastes of matrix factorization-based recommenders. In Mobasher, B., Burke, R. D., Jannach, D., and Adomavicius, G., editors, *RecSys*, pages 309–312. ACM.
- Chung, F. and Zhao, W. Pagerank and random walks on graphs.
- Cremonesi, P. and Turrin, R. (2010). Controlling consistency in top-n recommender systems. In Fan, W., Hsu, W., Webb, G. I., 0001, B. L., Zhang, C., Gunopulos, D., and Wu, X., editors, *ICDM Workshops*, pages 919–926. IEEE Computer Society.
- Lovász, L. (1996). Random walks on graphs: A survey. In Miklós, D., Sós, V. T., and Szőnyi, T., editors, *Combinatorics, Paul Erdős is Eighty*, volume 2, pages 353–398. János Bolyai Mathematical Society, Budapest.
- Mourão, F., Fonseca, C., Araújo, C., and Meira Jr, W. (2011). The oblivion problem: Exploiting forgotten items to improve recommendation diversity. In *Proc. of the workshop DiveRS*, Chicago, IL, USA.
- Ricci, F., Rokach, L., and Shapira, B. (2011). Introduction to recommender systems handbook. *Recommender Systems Handbook*, pages 1–35.
- Schwartz, B. (2005). *The Paradox of Choice: Why More Is Less*. Harper Perennial.
- Zhang, M. and Hurley, N. (2008). Avoiding monotony: Improving the diversity of recommendation lists. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys '08, pages 123–130, New York, NY, USA. ACM.