

Aplicando Nuvem de Partículas para Atenuar o Dilema entre Diversidade e Acurácia em Domínios de Recomendação

Tiago Trotta¹, Diego Carvalho¹, Nícollas Silva², Leonardo Rocha¹

¹ DCOMP/UFSJ - São João del-Rei, MG , Brasil

² DCC/UFMG - Belo Horizonte, MG , Brasil

{trotta,lcrocha}@ufsj.edu.br, {ncsilvaa}@dcc.ufmg.br

Abstract. *Advances in Recommender Systems (RSs) have been focused on presenting the most relevant items to users, in order to improve the sales and system's accuracy. However, accuracy alone is not enough to assess the practical effectiveness of recommendations. Metrics like novelty, diversity, and unexpectedness are key dimensions of recommendation utility in real scenarios. Specifically, there is an apparent accuracy-diversity dilemma that remains open and can boost sales by offering consumers both their mainstream and specific tastes. For this dilemma, we propose a post-processing approach, based on a Particle Swarm Optimization (PSO), that re-ranks the recommendation lists from traditional RSs, improving diversity up to 70% without accuracy losses.*

Resumo. *Sistemas de Recomendação (SsR) focam na apresentação dos itens mais relevantes aos usuários a fim de melhorar a acurácia do sistema. Entretanto, a acurácia não é suficiente para avaliar a efetividade prática das recomendações. Métricas como novidade, diversidade e imprevisibilidade são chave para se avaliar a utilidade de SsR em cenários reais. Especificamente, há um dilema em aberto de acurácia-diversidade capaz de impulsionar as vendas ao oferecer aos consumidores tanto itens convencionais quanto específicos. Para lidar com esse dilema, propomos uma abordagem de pós-processamento, baseada em Nuvem de Partículas, que reordena as listas recomendações de SsR tradicionais, melhorando a diversidade em até 70% sem perda de acurácia.*

1. Introdução

A enorme quantidade de informação disponível nas aplicações Web, como Amazon e Netflix, tem gerado um cenário desafiador: os usuários têm mais opções do que são capazes de lidar [Schwartz 2009]. Nesse sentido, ferramentas capazes de filtrar o conteúdo e prover uma personalização estão se tornando cada vez mais importantes. Essas aplicações, chamadas Sistemas de Recomendação (SsR), visam identificar itens potencialmente relevantes aos usuários. Basicamente, o sistema modela perfis dos usuários, por meio de suas informações pessoais (i.e. suas características e itens previamente consumidos), e dos itens, por meio de suas características (i.e. descrições técnicas e usuários que já os consumiram), e tenta estimar a relevância de cada item para os usuários.

Em grande parte dos avanços dos SsR o foco dos pesquisadores esteve sempre na acurácia do modelo. Entretanto, surgiu um crescente consenso de que somente acurácia não é suficiente para avaliar a eficiência prática e o valor das recomendações [Kunaver and Požrl 2017]. Particularmente, a diversidade tem sido identificada como uma dimensão

chave para medir a utilidade de uma recomendação [Kunaver and Požrl 2017]. Pesquisas recentes têm visado a maximização de dois objetivos distintos: acurácia e diversidade. Em cenários reais de comércio eletrônico, por exemplo, a combinação entre acurácia e diversidade pode impulsionar as vendas, por oferecer aos consumidores tanto produtos convencionais quanto produtos específicos [Kunaver and Požrl 2017]. No entanto, esse desafio mantém um aparente dilema, pois acurácia e diversidade representam conceitos opostos. Se, por um lado, itens que aumentam a acurácia são mais populares entre os usuários, por outro, itens que aumentam a diversidade são menos convencionais.

Nesse contexto, este trabalho visa aumentar a diversidade de SsR focados apenas na acurácia dos itens. Esperamos aumentar a diversidade das recomendações sem comprometer a acurácia obtida. Ao contrário das principais abordagens apresentadas na literatura, apresentamos um método de otimização a ser aplicado como uma etapa de pós-processamento que reordena as listas de recomendações de SsR tradicionais. Assim, nosso objetivo é similar ao clássico Problema de Diversidade Máxima (PDM), uma vez que é necessário escolher os n itens mais diversos dentre um conjunto de m itens distintos. Entretanto, também é preciso considerar a acurácia dos itens para mitigar o dilema de acurácia-diversidade. Assim, definimos uma função objetivo para balancear a combinação acurácia-diversidade dos itens e otimizá-la por meio de um método evolutivo. Optamos pela Nuvem de Partículas devido à sua eficácia e eficiência no contexto do PDM.

Para avaliar nossa proposta, selecionamos três SsR implementados na biblioteca *MyMediaLite* (i.e. ItemkNN, UserkNN e WRMF), considerando dois cenários de recomendação: entretenimento e comércio eletrônico. Os resultados mostram que nossa proposta obtém ganhos de 70% em diversidade, sem diferenças estatísticas na acurácia. Dessa forma, ressaltamos o balanceamento entre acurácia-diversidade como a principal contribuição do nosso modelo inspirado em um método evolutivo. É importante mencionar a escalabilidade e facilidade de implementação da nossa proposta, em situações reais. Além disso, é possível aplicar nossa proposta sobre a recomendação de qualquer SR tradicional para melhorar diferentes características, tais como a novidade dos itens.

Todas as implementações necessárias para execução desse trabalho, bem como as execuções dos experimentos foram realizadas pelo aluno Tiago Trotta, sob a orientação do professor Leonardo Rocha. A concepção do projeto e as análises de resultados foram feitas em conjunto, aluno e professor, com a colaboração do aluno recém formado, Diego Carvalho, e do aluno de doutorado do Programa de Pós-Graduação do DCC/UFMG, Nicollas Silva.

2. Conceitos Básicos

O Problema de Diversidade Máxima - PDM é um problema de otimização combinatorio que visa selecionar o conjunto de elementos mais diversos entre si. No domínio de SsR, a diversidade deve ser combinada com algum conceito de relevância (i.e., acurácia). Formalmente, deve-se selecionar um subconjunto de n elementos a partir de um conjunto de m elementos, tal que a soma das distâncias entre os elementos escolhidos seja maximizada [Kuo et al. 1993]. A distância d_{ij} entre dois elementos i e j tradicionalmente utilizada é a Distância Euclidiana. Essas distâncias são usadas para formular o PDM como um problema binário quadrático no qual a variável x_i assume o valor 1 se o elemento i for selecionado e 0 senão, $i = \{1, \dots, n\}$. O objetivo é maximizar

a diversidade dentre os elementos selecionados:

$$\text{Maximizar } \sum_{i=1}^{n-1} \sum_{j=i+1}^n d_{ij} x_i x_j, \text{ sujeito a } \sum_{i=1}^n x_i = m, x_i = \{0, 1\}, 1 \leq i \leq n$$

Os mais relevantes métodos aproximados para resolver esse problema são categorizados por metodologia: (1) Métodos baseados em GRASP; (2) Métodos baseados em pesquisa local; (3) Métodos baseados em população. No primeiro grupo, destacamos dois trabalhos. *Andrade et. al.* [de Andrade et al. 2005] propõem um método híbrido que combina metodologia GRASP com *Path-Relinking*. Já *Duarte e Martí* [Duarte and Martí 2007] propõem um método construtivo combinado com uma busca local first-improvement baseada em GRASP. No segundo, *Palubeckis* [Palubeckis 2007] propõe uma Busca Tabu iterativa que alterna entre busca tabu e procedimentos de perturbação. No terceiro grupo, *Bonotto et. al.* [Bonotto et al. 2017] propõem uma abordagem de nuvem de partículas combinada com busca tabu para lidar com o PDM sendo melhor que a maioria das linhas de base propostas anteriormente.

Otimização por Nuvem de Partículas (*Particle Swarm Optimization - PSO*) é um método evolucionar para otimizar um problema iterativamente, tentando melhorar um candidato a solução em relação a uma determinada medida de qualidade. Uma população de candidatos a solução, chamados *partículas*, são movimentados pelo espaço de busca de acordo com uma equação matemática simples que define a posição e a velocidade da partícula. O movimento de cada partícula é influenciado pela posição *ótima local* por ela conhecida, mas também é guiada em direção à *ótima global* conhecida em todo o espaço de busca, atualizada à medida em que melhores posições são descobertas por outras partículas. Assim, espera-se movimentar a nuvem em direção à melhor solução. No PSO, cada partícula inicia um movimento aleatório ou predeterminado. A cada iteração, o movimento é ajustado de acordo com a sua própria experiência, indo em direção à melhor posição por ela encontrada (*PBest* – um fator individual), e com a experiência de todas as outras partículas, movimentando em direção à melhor posição encontrada por qualquer partícula (*GBest* – um fator de sociabilidade). O *PBest* de cada partícula, em uma iteração específica, é sempre salvo em um histórico pessoal, comumente chamado de fator de individualidade. Por sua vez, cada partícula tem acesso irrestrito ao aspecto social, ou seja, ao ótimo global *GBest*. Assim, o movimento de cada partícula pode ser descrito pela Equação 1, na qual X_i é a posição da partícula i e V_i é a velocidade dessa partícula. Ambas estão na dimensão d do espaço multidimensional na k -ésima iteração.

$$X_{id}^{k+1} = X_{id}^k + V_{id}^{k+1} \quad (1)$$

No algoritmo convencional de PSO, no qual todas as partículas são consideradas vizinhas, a velocidade do movimento de cada partícula (i.e., o tamanho do salto no espaço de solução) é definida pela Equação 2. As constantes c_1 e c_2 são usadas para balancear as contribuições dos componentes cognitivos e sociais, respectivamente. As variáveis r_1 e r_2 são aleatoriamente e uniformemente distribuídas entre $[0, 1]$ e w é uma constante que representa o peso inercial da velocidade de uma partícula. A velocidade de uma partícula leva em consideração a influência de três fatores: (1) o fator inercial da partícula; (2) a posição ótima local; e (3) a posição ótima global. Esses fatores combinados são responsáveis por definir a velocidade V_{id}^{k+1} e a próxima posição X_{id}^{k+1} .

$$V_{id}^{k+1} = w \cdot V_{id}^k + c_1 \cdot r_1 \cdot (PBest_{id}^k - X_{id}^k) + c_2 \cdot r_2 \cdot (GBest_{id}^k - X_{id}^k) \quad (2)$$

3. Maximizando a Diversidade de Sistemas de Recomendação (MDSR)

Nesse trabalho propomos uma solução para o aparente dilema da diversidade-acurácia que visa aumentar a diversidade de SsR tradicionais, sem comprometer a acurácia obtida. Em um processo tradicional de recomendação o modelo de predição aprende as preferências de um usuário baseando-se em informações anteriores e seleciona um conjunto de m itens relevantes. A partir de m o SR então recomenda os n mais relevantes, tal que $n \ll m$. Nossa proposta consiste em uma etapa de pós-processamento da saída de um SR qualquer, visando encontrar a melhor combinação de n que maximize a diversidade, sem comprometer a acurácia. Esse processo de otimização é definido na próxima seção.

3.1. Lidando com o dilema da diversidade-acurácia

O dilema da acurácia-diversidade já vem sendo estudado há algum tempo em cenários de Recuperação de Informação, seguindo o conceito de *Maximal Marginal Relevance* [Carbonell and Goldstein 1998]. Nesses cenários, trata-se de uma combinação linear ponderada de relevância e diversidade (Equação 3), ou seja, considera-se a diversidade na reordenação de documentos para evitar a apresentação de informações duplicadas. De forma similar, no cenário de recomendação, também se considera como função objetivo a Equação 3, na qual o conjunto S representa um candidato a solução contendo n itens selecionados da lista de recomendações R com m elementos. O parâmetro λ é usado para balancear as duas métricas no intervalo $[0, 1]$.

$$f(S) = (1 - \lambda) \cdot rel(S) + \lambda \cdot div(S) \quad (3)$$

A função $rel(S)$ é usada para medir a relevância de S para um usuário específico, similarmente ao conceito de acurácia. Visto que nossa proposta é um método de pós-processamento, não temos acessos às informações usuários-itens. Portanto, propomos medir a relevância de um item com base na sua posição no ranking gerado previamente pelo SR. Essa proposição está relacionada à premissa básica de uma lista de recomendação, na qual os primeiros itens são mais relevantes que os últimos. A Equação 4 representa esse método, sendo $pos(i, R)$ uma função que retorna a posição do item i na lista R . Nesta equação, agregamos a relevância de cada item i para medir a relevância do conjunto S . A função $div(S)$ representa a diversidade alcançada por S que é calculada, em nossa proposta, com base em suas características, como o gênero de um filme. A Equação 5 mede a diversidade de S com base em *cobertura de gêneros* [Puthiya Parambath et al. 2016]. Basicamente, computamos o número de gêneros distintos relacionados aos itens de S e normalizamos o valor com base no total de gêneros distintos dentre todos os itens I em determinado cenário. Valores elevados de cobertura de gêneros indicam grande diversidade.

$$rel(S) = \sum_{i \in S} \frac{n - pos(i, R)}{n} \quad (4) \quad div(S) = \frac{|\bigcup_{i \in S} \text{gêneros}(i)|}{|\text{gêneros}(I)|} \quad (5)$$

3.2. Adaptando a Otimização por Nuvem de Partículas

Conforme mencionado, o objetivo é, a partir de um conjunto R com m elementos, selecionar o conjunto S com n elementos, tal que $n \ll m$, que maximiza a função $f(S)$. Trata-se de um problema NP-Completo com $n!/(m-n)!$ possíveis soluções [Kuo et al. 1993]. Para se aproximar da solução ótima em um tempo computacional aceitável, adaptamos o método evolutivo de Nuvem de Partículas (PSO), sendo cada partícula referente a

uma potencial solução S , a posição de uma partícula em qualquer iteração é denotada por seus itens e mudar a posição de uma partícula significa mudar seus itens. A diferença para a PSO tradicional está no movimento das partículas. Tradicionalmente, a PSO é aplicada a variáveis contínuas num espaço d -dimensional. Contudo, em cenários de recomendação as variáveis (itens, neste caso) são discretas. Assim, representamos o movimento das partículas com base numa estratégia tradicional no campo de RI: *team draft interleaving (TDI)* [Radlinski et al. 2008]. Esta estratégia foi proposta para misturar rankings distintos de documentos e obter um único com os melhores. Basicamente, a cada iteração o algoritmo TDI seleciona aleatoriamente um documento de um dos rankings. Para o nosso problema, consideramos como rankings os três fatores que influenciam as velocidades das partículas numa PSO tradicional. O primeiro ranking é a partícula atual (P_0). O segundo é a partícula com a melhor posição ótima local (P_{best}), i.e., a que obteve o maior valor de $f(S)$ após variações locais. O terceiro é a partícula com a posição ótima global (G_{best}), i.e., a que obteve o maior valor de $f(S)$ dentre todas as partículas. A cada iteração, o Algoritmo 1 escolhe um ranking de maneira aleatória-ponderada e um item aleatório dentro desse ranking. A fim de guiar as partículas em direção à posição ótima global, balanceia-se a escolha de ranking com pesos: rankings atual e local têm 30% de chance de serem escolhidos e o global tem 40%, valores foram escolhidos com base em testes experimentais.

Algoritmo 1 – Team Draft Interleaving (TDI)

Entrada: Rankings P_0, P_{best}, G_{best} e um valor n .

Saída: Um ranking S_P com itens misturados.

```

1:  $L \leftarrow \{P_0, P_{Best}, G_{Best}\}$ 
2:  $w \leftarrow \{0.3, 0.3, 0.4\}$ 
3: while  $itr < n$  do
4:    $l \leftarrow \text{roleta}(L, w)$   $\triangleright$  Selecionando a lista  $l$  de  $L$ 
5:    $i \leftarrow \text{aleatório}(l)$   $\triangleright$  Selecionando o item  $i$  da lista  $l$ 
6:    $\text{adiciona}(S, i)$   $\triangleright$  Adicionando  $i$  no conjunto  $S$ 
7:    $\text{remove}(L, i)$   $\triangleright$  Removendo  $i$  de todas as listas
8:    $itr \leftarrow itr + 1$ 
9: end while
10: return  $S$ 

```

Algoritmo 2 – Maximizando Diversidade em SRs (MDSR)

Entrada: Uma lista de recomendações R , quantidade partículas q , número de iterações $itrMax$

Saída: A melhor partícula encontrada.

```

— INICIALIZANDO A NUVEM DE PARTÍCULAS —
1:  $Nuvem[P_0, \dots, P_{int(q*0.4)}] \leftarrow \text{topItens}(R)$ 
2:  $Nuvem[P_{int(q*0.4)}, P_q] \leftarrow \text{itensAleatorios}(R \setminus Nuvem)$ 
3: for all  $Nuvem$  do
4:    $P_{i-best} \leftarrow P_i$ 
5:   if  $(f(P_{i-best}) > f(G_{best}))$  then
6:      $G_{best} \leftarrow P_{i-best}$ 
7:   end if
8: end for

— ITERANDO ENTRE POTENCIAIS SOLUÇÕES —
9:  $itr \leftarrow 0$ 
10: while  $(itr < itrMax)$  do
11:   for all  $Nuvem$  do
12:      $P_i \leftarrow \text{TDI}(P_i, P_{i-best}, G_{best})$   $\triangleright$  Atualizando  $P_i$ 
13:     if  $(f(P_i) > f(P_{i-best}))$  then
14:        $P_{i-best} \leftarrow P_i$   $\triangleright$  Atualizando  $P_{i-best}$ 
15:       if  $(f(P_{i-best}) > f(G_{best}))$  then
16:          $G_{best} \leftarrow P_{i-best}$   $\triangleright$  Atualizando  $G_{best}$ 
17:       end if
18:     end if
19:   end for
20:    $itr \leftarrow itr + 1$ 
21: end while
22: return  $G_{best}$ 

```

O Algoritmo 2 ilustra o PSO. Primeiro, definimos uma nuvem de partículas para representar uma quantidade de soluções em potencial para se explorar. O número de partículas é definido antes da execução e deve ser maior ou igual a m/n . Essa restrição deve ser seguida para que haja a garantia de que todos os itens de R estejam presentes em pelo menos uma partícula S . 40% de todas as partículas são compostas de itens do topo da lista R (linha 1) e 60% de itens aleatórios (linha 2). Essa divisão foi adotada de acordo com [Bonotto et al. 2017]. Em seguida, o MDSR define os P_{best} e G_{best} iniciais para cada

partícula (linhas 3-8). No segundo passo, o MDSR mede os melhores resultados locais e globais das partículas por um número fixo de iterações. Basicamente, o MDSR perturba as partículas usando o algoritmo TDI, misturando a lista de partículas e as melhores soluções globais e locais (linha 12). O critério de parada adotado foi número de iterações (i.e. 100). Finalmente, o algoritmo retorna a partícula que se refere à solução ótima global.

4. Avaliação Experimental

Para avaliar a qualidade da nossa proposta, selecionamos dois cenários de recomendação: entretenimento e comércio eletrônico. Nesses cenários, há informações sobre itens e usuários que assistem ou compram filmes em aplicações Web. Além disso, selecionamos três Sistemas de Recomendação tradicionais para tentar otimizar a diversidade em suas listas de recomendações. Esses métodos estão implementados na biblioteca *MyMediaLite*, um repositório clássico de modelos de predição. A solução ótima para o Problema de Máxima Diversidade desses cenários não é conhecida, portanto, nossas comparações estão restritas aos valores de acurácia e diversidade medidas anteriormente pelos SsR.

4.1. Bases de dados

Selecionamos as bases de dados *MovieLens 1M*¹ e *CiaoDVD*² para simular cenários de entretenimento (avaliações de filmes) e de comércio virtual (compra de DVDs). Ambas apresentam notas entre 1 e 5 assinaladas por usuários. Outras características estão presentes na Tabela 1. Em nossos experimentos, separamos 70% do histórico de consumo dos usuários para treino e 30% para teste. Essa divisão é feita com base nas *timestamps* de cada interação usuário-item. A fim de evitar viés no comportamento do usuário, precisamos selecionar os itens consumidos mais recentemente para construir o conjunto de teste.

Tabela 1. Bases de dados - informações gerais.

Base	Usuários	Itens	Eaparsidade	Gêneros
<i>ML-1M</i>	6.040	3.952	95,82%	18
<i>CiaoDVD</i>	17.615	16.621	99,97%	17

4.2. Métricas de qualidade

Para se medir a acurácia das recomendações selecionamos a **Precisão**, que corresponde à fração de itens relevantes (N_{rs}) ou irrelevantes (N_s) selecionados existente no conjunto de teste T do usuário. Para diversidade, selecionamos a Cobertura de Gêneros, que corresponde à proporção de gêneros distintos de itens, relevante (R^+) ou irrelevante (R), recomendada a usuários. Ambas foram implementadas com e sem consciência de relevância. O conceito de relevância foi adicionado para assegurar a qualidade dos itens recomendados. Neste trabalho, um item i é relevante para um usuário u se u deu a i uma nota maior que sua média pessoal. A Tabela 2 mostra as variações dessas métricas.

Tabela 2. Especificação das métricas com e sem relevância.

Métrica	Sem Relevância	Com Relevância
<i>Precisão</i>	$\frac{N_s \cap T}{N}$	$\frac{N_{rs} \cap T}{N}$
<i>Cobertura de Gêneros</i>	$\frac{ \bigcup_{i \in R} g(i) }{ \bigcup_{i \in T} g(i) }$	$\frac{ \bigcup_{i \in R} g(i) \cap \bigcup_{i \in R^+} g(i) }{ \bigcup_{i \in R^+} g(i) }$

¹Disponível em: <http://www.grouplens.org/node/12>

²Disponível em: <https://www.librec.net/datasets.html>

4.3. Sistemas de Recomendação tradicionais

Como mencionado, o MDSR proposto é uma etapa de pós-processamento que visa a melhoria da acurácia e da diversidade em qualquer lista de recomendações. Portanto, selecionamos três modelos distintos implementados pela biblioteca *MyMedialite*, todos baseados em filtragem colaborativa e brevemente descritos abaixo, para avaliar nossa proposta:

- **User-kNN:** esse método visa correlacionar usuários para descobrir itens relevantes de usuários similares [Schafer et al. 2007]. O método assume que, se dois usuários A e B possuem a mesma opinião sobre algum assunto, há mais chances de que A e B tenham a mesma opinião em um outro assunto diferente que usuários escolhidos aleatoriamente. O *User-kNN*: (1) determina k usuários vizinhos ao usuário ativo u ; (2) implementa uma abordagem de agregação com a nota de um vizinho em itens não avaliados por u ; e (3) extrai as predições do passo 2 e então seleciona as $top - k$ recomendações [Bobadilla et al. 2013]. Em nossos experimentos, consideramos 80 vizinhos, conforme configuração padrão do *MyMedialite*.
- **Item-kNN:** esse método visa correlacionar itens e encontrar itens relevantes [Schafer et al. 2007]. Supõe-se que itens similares a itens consumidos por uma pessoa A tem grandes chances de satisfazer seus gostos. Neste caso, o *kNN*: (1) determina k itens similares aos itens consumidos por u ; (2) implementa uma abordagem de agregação com as notas da vizinhança em itens não avaliados por u ; e (3) extrai a predição do passo 2 e então seleciona as $top - k$ recomendações [Bobadilla et al. 2013]. Também consideramos 80 vizinhos, conforme configuração padrão do *MyMedialite*.
- **Weighted Regularized Matrix Factorization (WRMF):** esse método emprega as similaridades usuário-item em seu espaço latente para extrair os padrões de compra usuário-item através de *Singular Value Decomposition* (SVD). Especificamente, o método WRMF aplica uma matriz de ponderamento para diferenciar contribuições de atividades de compra/avaliação observadas e não observadas [Schafer et al. 2007]. Os $top - k$ itens são selecionados por meio de produto linear entre autovalores e autovetores. Consideramos 10 autovalores, conforme configuração padrão do *MyMedialite*.

4.4. Resultados Experimentais

Todas as estratégias selecionadas geram uma lista R com 100 itens ordenados por relevância dos quais precisamos extrair 10 itens que maximizem os resultados acurácia-diversidade, ou seja, o conjunto S . Como definido na Equação 3, nossa proposta tem um parâmetro λ , cujo valor considerado no primeiro conjunto de experimentos foi 0.5, para balancear igualmente as métricas de acurácia e diversidade de nossa função objetivo.

A Tabela 3 mostra os valores médios, máximos (Max.) e do desvio padrão (STD) para cada ranking antes e depois a aplicação do MDSR - os valores mínimos obtidos foram 0 para todas as métricas de todos os recomendadores em ambas as bases, portanto foram omitidos. Analisando a métrica de *Cobertura de Gêneros*, os resultados mostram uma grande melhora na diversidade dos itens. Enquanto os $top - 10$ itens do *User-kNN*, por exemplo, cobrem 52% dos gêneros dos filmes, os $top - 10$ itens da nossa abordagem MDSR cobrem 80%. Em uma simples comparação, agregando os ganhos de *Cobertura de Gêneros* em cada SR, a diversidade após o MDSR apresenta um ganho médio de 72% na base ML-1M e 70% na CiaoDVD. Além disso, os resultados mostram que a *Precisão* do modelo não sofre grandes mudanças em nenhuma das bases. O teste estatístico de

Chi – Quadrado confirma que as perdas de precisão são insignificantes. Em 95% dos casos, a acurácia após a aplicação do MDSR é similar à acurácia medida antes. Considerando relevância, os resultados dessas métricas são menores que os resultados sem relevância. Contudo, esses resultados reforçam as conclusões previamente observadas. Em suma, essas análises mostram que nossa abordagem é relevante para cenários reais, já que é capaz de diversificar os resultados sem perda em acurácia.

Tabela 3. Esta tabela mostra os resultados das métricas antes e depois do pós processamento com o MDSR, denotados por *< antes / depois >*. O símbolo ▲ denota significativos ganhos positivos, ● ganhos não significativos e ▼ e significativas perdas negativas. Esses ganhos são calculados com relação aos resultados dos SRs antes do MDSR, aplicando-se um teste de Chi-Quadrado com 95% de confiança. Os valores mínimos foram omitidos pois são equivalentes a 0 antes e depois do pós-processamento.

		ML-1M			CiaoDVD		
		Métrica	Média	Max.	STD	Média	Max.
ItemkNN	Prec.	0,118 / 0,117 ●	1 / 1	0,15 / 0,14	0,0019 / 0,0016 ●	0,3 / 0,2	0,015 / 0,013
	Rel-Prec.	0,072 / 0,071 ●	0,9 / 0,8	0,11 / 0,1	0,0012 / 0,0009 ●	0,3 / 0,1	0,011 / 0,009
	Cob. Gên.	0,34 / 0,7 ▲	0,83 / 1	0,139 / 0,136	0,3 / 0,5 ▲	0,55 / 0,55	0,077 / 0,062
	Rel-Cob. Gên.	0,35 / 0,57 ▲	0,83 / 1	0,13 / 0,16	0,064 / 0,082 ▲	0,44 / 0,5	0,048 / 0,062
UserkNN	Prec.	0,17 / 0,16 ●	1 / 1	0,19 / 0,17	0,0069 / 0,0063 ●	0,3 / 0,3	0,027 / 0,026
	Rel-Prec.	0,11 / 0,1 ●	0,9 / 0,9	0,15 / 0,13	0,0048 / 0,0046 ●	0,2 / 0,2	0,071 / 0,063
	Cob. Gên.	0,52 / 0,8 ▲	0,89 / 1	0,1 / 0,08	0,25 / 0,44 ▲	0,5 / 0,56	0,071 / 0,069
	Rel-Cob. Gên.	0,45 / 0,63 ▲	0,89 / 1	0,12 / 0,16	0,068 / 0,083 ▲	0,39 / 0,5	0,048 / 0,063
WRMF	Prec.	0,17 / 0,16 ●	1 / 1	0,19 / 0,17	0,0059 / 0,005 ●	0,3 / 0,3	0,027 / 0,024
	Rel-Prec.	0,114 / 0,11 ●	1 / 1	0,15 / 0,14	0,0045 / 0,0037 ●	0,3 / 0,3	0,023 / 0,02
	Cob. Gên.	0,5 / 0,79 ▲	0,89 / 1	0,12 / 0,08	0,23 / 0,39 ▲	0,44 / 0,56	0,073 / 0,072
	Rel-Cob. Gên.	0,44 / 0,62 ▲	0,89 / 1	0,13 / 0,16	0,056 / 0,073 ▲	0,39 / 0,56	0,051 / 0,062

Em nossa segundo conjunto de experimentos variamos o valor de λ no intervalo $[0, 1]$. Esta análise visa verificar se a adição de diversidade à função objetiva prejudicará ou não a acurácia do modelo. Neste caso, avaliamos o impacto da variação de λ na função objetiva e a eficácia da recomendação. Por um questão de espaço, limitamos nossa segunda análise ao algoritmo *User-kNN*. Como define nossa função objetiva, o MDSR visa maximizar o dilema da acurácia-diversidade ajustando o parâmetro λ . Maiores valores (ou seja, quando λ se aproxima de 1), indicam mais diversidade na função objetiva. Contrariamente, valores menores (ou seja, quando λ se aproxima de 0) indicam mais acurácia. Especialmente, com $\lambda = 0$, o MDSR mantém a ordem dos itens e retorna o mesmo ranking do recomendador usado antecipadamente. Quando $\lambda = 1$, o MDSR não segue o ranking da recomendação e busca apenas a maximização da diversidade.

A Figura 2 mostra o comportamento do MDSR aplicado ao *User-kNN* em duas bases de dados. As primeiras duas linhas são resultados da base ML-1M, enquanto as outras duas são da CiaoDVD. Cada coluna de gráficos representa um valor de λ no intervalo $[0, 1]$. Movendo para a direita e observando as linhas 1 e 3, podemos ver que os valores de acurácia³ após a aplicação do MDSR (as linhas pretas no gráfico) decrescendo contrariamente a λ . Entretanto, na mesma observação nas linhas 2 e 4, podemos ver os valores de diversidade após a aplicação do MDSR (novamente, as linhas pretas) aumentarem junto a λ . Assim, confirmamos que $\lambda = 0.5$ é o melhor valor, pois maximiza em 70% a diversidade do sistema de recomendação sem introduzir perdas de acurácia.

³O menor valor de acurácia para a CiaoDVD, mostrado na linha 3 da Figura 2, está relacionado ao cenário do comércio eletrônico no qual os usuários avaliam poucos itens.

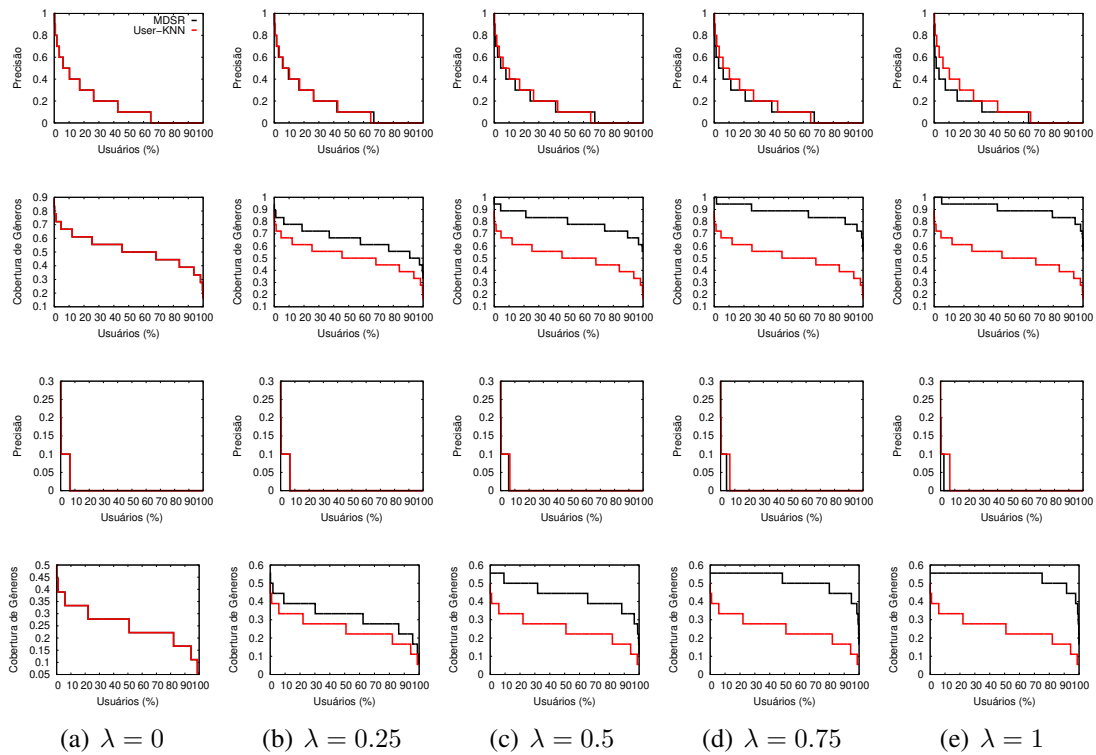


Figura 1. Análise das variações de λ com as métricas *Precisão* e *Cobertura de Gêneros* sem relevância para o *User-kNN*. As primeiras duas linhas são resultados da base *ML-1M*, enquanto as outras duas são da *CiaoDVD*. Com base na nossa função objetiva, os maiores valores de λ indicam que mais diversidade está sendo adicionada. Contrariamente, menores valores indicam mais acurácia na recomendação. Quando $\lambda = 0$, o *MDSR* retorna o mesmo ranking do recomendador e, portanto, as curvas das métricas são idênticas. Incrementando λ , aumentamos a diversidade do *SR*, mas, ao mesmo tempo, diminuímos a acurácia.

Em resumo, a abordagem *MDSR* é capaz de recomendar filmes relevantes de gêneros distintos que não haviam aparecido anteriormente. A Tabela 4 mostra os filmes recomendados antes e depois da aplicação do *MDSR*, ressaltando seus gêneros. Antes do *MDSR*, os *top-5* filmes eram muito relacionados a *Ação* e *Aventura*. Após a aplicação, foram incluídas mais opções para o usuário, selecionando filmes relacionados a *Mistério* e *Infantil*. Assim, nossa proposta, inspirada em um método evolutivo, é capaz de balancear a troca entre acurácia e diversidade, além de ser facilmente aplicada em qualquer *SR* tradicional.

Tabela 4. Gêneros dos filmes recomendados pelo *User-kNN* na base *ML-1M* antes e depois da aplicação do *MDSR*.

GÊNEROS DOS FILMES			
Antes do <i>MDSR</i>		Após o <i>MDSR</i>	
Filme	Gêneros	Filme	Gêneros
Star Wars V	Ação, Avent., Sci-Fi	Pulp Fiction	Crime, Drama
Star Wars VI	Ação, Avent., Guerra	Ghostbusters	Comédia, Terror
Indiana Jones	Ação, Avent.	Groundhog Day	Comédia, Romance
Schindler's List	Drama, Guerra	L.A. Confidential	Mistério, Suspense
Princess Bride	Ação, Romance	Beauty and Beast	Animação, Infantil

5. Conclusão

Existe um crescente consenso na comunidade de Sistemas de Recomendação de que tanto acurácia quanto diversidade devem ser consideradas para avaliar a eficácia prática de recomendações. Contudo, esses dois conceitos são conflitantes: itens que melhoram a acurácia são aqueles mais populares e itens que melhoram a diversidade são, geralmente, não convencionais. Esse desafio é conhecido como o aparente dilema acurácia-diversidade. Mapeando esse dilema no clássico Problema de Diversidade Máxima, apresentamos uma estratégia de pós-processamento que recebe como entrada a lista de recomendações gerada por um SR e reordena a lista a fim de maximizar a diversidade sem comprometer a acurácia. Para esse processo de otimização, utilizamos o método evolutivo Nuvem de Partículas. Avaliamos nossa proposta considerando dois cenários de recomendação, entretenimento e comércio eletrônico, e três SRs distintos. Os resultados demonstram que, além de ser escalável e de fácil implementação, nossa proposta obtém ganhos de até 70% em diversidade sem perder acurácia. Como trabalho futuro, nossa intenção é avaliar outros métodos evolutivos diferentes, considerando mais algoritmos de recomendação e mais bases de dados relacionadas a outros cenários.

Referências

- Bobadilla, J., Ortega, F., Hernando, A., and Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems*, 46:109–132.
- Bonotto, E. L. et al. (2017). Otimização por nuvem de partículas e busca tabu para problema da diversidade máxima.
- Carbonell, J. and Goldstein, J. (1998). The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proc. ACM SIGIR*, pages 335–336.
- de Andrade, M. R., de Andrade, P. M., Martins, S. L., and Plastino, A. (2005). Grasp with path-relinking for the maximum diversity problem. In *International Workshop on Experimental and Efficient Algorithms*, pages 558–569. Springer.
- Duarte, A. and Martí, R. (2007). Tabu search and grasp for the maximum diversity problem. *European Journal of Operational Research*, 178(1):71–84.
- Kunaver, M. and Požrl, T. (2017). Diversity in recommender systems—a survey. *Knowledge-Based Systems*, 123:154–162.
- Kuo, C.-C., Glover, F., and Dhir, K. S. (1993). Analyzing and modeling the maximum diversity problem by zero-one programming. *Decision Sciences*, 24(6):1171–1185.
- Palubeckis, G. (2007). Iterated tabu search for the maximum diversity problem. *Applied Mathematics and Computation*, 189(1):371–383.
- Puthiya Parambath, S. A., Usunier, N., and Grandvalet, Y. (2016). A coverage-based approach to recommendation diversity on similarity graph. In *RecSys*, pages 15–22.
- Radlinski, F., Kurup, M., and Joachims, T. (2008). How does clickthrough data reflect retrieval quality? In *Proc. of CIKM*, pages 43–52. ACM.
- Schafer, J. B., Frankowski, D., Herlocker, J., and Sen, S. (2007). Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer.
- Schwartz, B. (2009). *The Paradox of Choice: Why More Is Less*. HarperCollins.