FAiR: Framework de Avaliação e Caracterização em Sistemas de Recomendação*

Diego Carvalho[1] **, Nícollas Silva**[2] **, Leonardo Rocha**[1]

[1] Universidade Federal de São João del-Rei - UFSJ
[2] Universidade Federal de Minas Gerais - UFMG

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

Abstract. *Recommender systems (RSs) have become essential tools in e-commerce applications. Several strategies have been proposed to help users in decision-making. The evaluation process of these RSs is, nowadays, a major divergence point, since there is no consensus regarding which metrics are necessary to consolidate a new RSs. In the present work, we perform an extensive study of these metrics organizing them into three groups: Effectiveness-based, Complementary Dimensions of Quality and Domain Profiling. Further, we consolidate a framework named FAiR that is able to help researchers in evaluating their RSs using these metrics, besides identifying the characteristics of data collections that may be intrinsically influencing RSs performance.*

Resumo. *Sistemas de Recomendação (SsR) tornaram-se essenciais em e-commerces. Estratégias de SsR vêm sendo propostas para auxiliar usuários na tomada de decisão. O processo de avaliação de SsR é hoje um grande ponto de divergência, uma vez que não existe um consenso de quais métricas são necessárias para se consolidar um novo SR. Nesse trabalho, realizamos um amplo estudo dessas métricas organizando-as em três grupos: Effectiveness-based, Complementary Dimensions of Quality e Domain Profiling. Consolidamos um framework denominado FAiR, capaz de auxiliar pesquisadores na avaliação de SsR frente a essas métricas, além de identificar as características das coleções de dados que possam intrinsecamente influenciar no desempenho deles.*

1. Introdução

Atualmente, a grande quantidade de produtos disponíveis em uma variedade de aplicações Web gera um cenário desafiador: usuários dispõem de mais opções do que conseguem lidar efetivamente [Schwartz 2009]. Ao apresentar milhares ou mesmo milhões de produtos distintos, sistemas comerciais como *Amazon.com*, *Netflix* ou *Last.fm* involuntariamente criam obstáculos aos usuários que almejam apenas encontrar produtos de seu interesse. Dessa forma, Sistemas de Recomendação (SsR) têm se tornado uma das principais ferramentas utilizadas para auxiliar usuários na tomada de decisão. Esses SsR visam estimar a relevância dos itens por meio de diferentes fontes de informações (e.g., sociais, demográficas, histórico de consumo, etc.) [Modarresi 2016]. Em razão do retorno financeiro que um SR bem elaborado pode gerar para grandes corporações, muitos pesquisadores estão envolvidos em projetos relacionados a esse tema [Kouki et al. 2015].

*Esse trabalho foi parcialmente financiado por CNPq, CAPES, FINEP, Fapemig, e INWEB.

O ciclo básico de pesquisa para desenvolver um novo SR é composto de quatro etapas: (1) implementar um SsR utilizando uma nova abordagem; (2) selecionar e implementar as principais estratégias (i.e., *baselines*) relacionadas a nova abordagem proposta; (3) selecionar e implementar métricas de avaliação inerentes a tarefa de recomendação a ser realizada; e por fim (4) avaliar os resultados encontrados comparando a nova proposta com os *baselines*, em distintos cenários de recomendação. Apesar de intuitivo, este processo de pesquisa é flexível e nem sempre confiável por ser altamente dependente do pesquisador, que deve realizar todas as etapas. Para auxiliar na etapa (2), fornecendo mais confiabilidade no processo, bibliotecas como *MyMediaLite* e *LensKit* se destacam por implementar os principais recomendadores estado da arte. Entretanto, o processo de avaliação de um novo SsR (etapas 3 e 4) permanece com um dos grandes desafios ainda em aberto. Mesmo com os avanços da área, não existe um consenso sobre quais métricas de avaliação são necessárias para consolidar um novo SsR. Frequentemente, os pesquisadores se limitam a avaliar métricas inerentes às técnicas implementadas, não validando simultaneamente requisitos de qualidade e satisfação dos usuários [Bobadilla et al. 2013].

Nesse sentido, a primeira contribuição desse trabalho consiste em um amplo estudo das diversas métricas utilizadas na literatura para se avaliar SsR [Herlocker et al. 2004, Adomavicius and Tuzhilin 2005, Bobadilla et al. 2013]. Por critérios de organização, dividimos as métricas relacionadas em três grupos: (1) *Effectiveness-based*, que consiste em métricas de decisão clássicas, tais como acurácia, precisão, revocação, entre outras, selecionadas a fim de quantificar a utilidade de um SsR; (2) *Complementary Dimensions of Quality*, que consiste em métricas relacionadas a dimensões que extrapolam a análise de acertos nas recomendações. Dentre tais dimensões destacamos novidade e diversidade das recomendações geradas frente ao histórico de consumo dos usuários; e (3) *Domain Profiling*, que consiste em métricas que visam avaliar as principais características dos domínios avaliados, a fim de se relacionar as características de usuários e itens com o desempenho dos SsR avaliados.

Como nossa segunda contribuição, selecionamos, implementamos e testamos as mais populares métricas para, no fim, consolidarmos um *framework* denominado *FAiR* (**Framework for Analyses in Recommender Systems**). *FAiR* é capaz de caracterizar e distinguir estratégias de recomendação de maneira confiável e automatizada, avaliando qualquer SR frente aos principais requisitos fundamentais de qualidade. Além disso, *FAiR* se apresenta como um *framework* para caracterização de distintos domínios de SsR, como músicas, filmes ou *e-commerce*. Outra vantagem em utilizar o *FAiR* durante as pesquisas, é que o mesmo é compatível com o formato de saída dos resultados gerados pelas principais bibliotecas de SsR existentes (i.e., *MyMediaLite* e *LensKit*). Esperamos assim, auxiliar os diversos pesquisadores da área, por apresentar um *framework* capaz de facilitar a execução das fases (3) e (4) do processo de pesquisa. O *framework* proposto é não-fechado e extensível, uma vez que outras métricas podem ser incorporadas ao mesmo.

A concepção do framework, bem como todas as implementações, execuções de experimentos e análises de resultados foram realizadas pelo aluno Diego Carvalho, sob a orientação do professor Leonardo Rocha. O aluno de pós-graduação Nícollas Silva colaborou no estudo e validação das métricas de avaliação de SsR.

2. Trabalhos Relacionados & Bibliotecas

Recentemente, as etapas de pesquisa relacionadas à avaliação de recomendações é destacada por vários pesquisadores [Lu et al. 2015]. Em [Konstan and Riedl 1999], os autores sugerem que as abordagens existentes para avaliar SsR podem ser divididas em duas categorias: (1) *avaliação online*, onde o desempenho é avaliado nos usuários de um sistema de recomendação em execução; e (2) *avaliação offline*, onde o desempenho de um mecanismo de recomendação é avaliado em base de dados existentes. Em geral, avaliação online é problemática devido à necessidade de um sistema totalmente funcional e uma comunidade de usuários. Por exemplo, em [Ozok et al. 2010] os autores exploraram a usabilidade e as preferências dos usuários através de uma pesquisa de um grupo de estudantes universitários. Além disso, um teste online deve estar preocupado com a interface da aplicação, pois influencia a satisfação dos usuários e a intenção de fornecer feedback. Consequentemente, a maioria dos pesquisadores prefere a avaliação offline, devido a simplicidade e eficiência em executá-la. Neste caso, usa-se base de dados existentes coletadas durante a execução de um sistema real [Bobadilla et al. 2013].

Na avaliação offline, a recomendação pode ser vista como recuperação de informações, i.e. a seleção do subconjunto de ativos que são relevantes para o usuário. Nesta perspectiva, as métricas para avaliação são bem conhecidas. Por esse motivo, vários *framework* foram desenvolvidos com um objetivo similar em mente: fornecer uma maneira fácil de implementar recomendadores em ambientes de pesquisa e/ou produção. Entre muitas bibliotecas, destacamos *MyMediaLite* [Gantner et al. 2011], implementada em Common Language Runtime (CLR) por membros da Universidade de Hildesheim, e *LensKit* [Ekstrand et al. 2011], uma biblioteca de SsR criada pelo *GroupLens* na Universidade de Minnesota. *MyMediaLite* destaca-se por conter os principais recomendadores em um ambiente leve, fácil de instalar e volátil, permitindo que os usuários alterem as características e parâmetros de entrada dos SsR. No entanto, o processo de avaliação é limitado à análise de qualidade das recomendações, apresentando apenas métricas de decisão, e dificultando a análise do comportamento das recomendações. Por sua vez, *LensKit* é uma biblioteca implementada em *Java*, compatível com muitos sistemas operacionais, que visa facilitar aplicações de e-commerce na tarefa de recomendação. No entanto, *LensKit* possui uma documentação de difícil entendimento.

Outras bibliotecas relevantes são *RiVal*[1], *LibRec*[2] e *RankSys*[3]. *RiVal* não possui SsR implementados e avalia as recomendações por meio de métricas clássicas. As outras duas bibliotecas possuem algum algoritmo de recomendação clássico e algumas métricas de avaliação implementadas. *LibRec* está dividida em quatro módulos: entrada de dados, processo de recomendação, processo de avaliação e saída de dados. Embora tenha as principais métricas de avaliação, esta biblioteca não inclui métricas de novidade e serendipidade, que se tornaram fundamentais para SsR [Wu et al. 2012]. *RankSys* também é implementado em Java e além dos algoritmos e métricas de avaliação implementadas por *LibRec*, possui algumas métricas relacionadas a novidade e serendipidade. Nossa proposta difere das anteriores ao implementar um conjunto muito grande de métricas, ser capaz de caracterizar a base de dados e consolidar uma interface intuitiva e extensível.

[1]Available: https://github.com/recommenders/rival
[2]Available at: https://www.librec.net/
[3]Available at: https://github.com/RankSys

3. Métricas para avaliação de SsR

SsR são fundamentados por algoritmos bem estruturados e incrementais que se diferem em relação a seus pontos fortes e fracos [Ricci et al. 2011]. Em geral, os principais SsR existentes se preocupam em melhorar a acurácia sob a premissa de apresentar itens potencialmente úteis aos usuários [Bobadilla et al. 2013]. Para estes, as principais métricas de avaliação estão relacionadas aos conceitos de acurácia, precisão e revocação [Herlocker et al. 2004]. Precisão e revocação visam quantificar informações sobre listas de recomendações geradas [Adomavicius and Tuzhilin 2005]. Por sua vez, as métricas relacionadas a acurácia consistem em avaliar a predição realizada pelos SsR, como no caso das métricas de MAE e RMSE [Adomavicius and Tuzhilin 2005]. Todas essas métricas relacionadas à qualidade dos SsR foram consolidadas em um grupo denominado *Effectiveness-based*, sumarizadas na tabela 1.

Tabela 1. *Effectiveness-based*: Métricas Relacionadas à qualidade dos SsR.

1. *Precision*: representa a probabilidade que um item selecionado seja relevante, definida como a razão dos itens relevantes selecionados pelo número de itens da lista de recomendação [Herlocker et al. 2004].
2. *Recall*: representa a probabilidade de um item relevante ser selecionado, definida como a razão dos itens relevantes selecionados pelo número de itens relevantes existentes [Herlocker et al. 2004].
3. Distribuição de Probabilidade de *Precision* e *Recall*: medida estatística básica que nos permite verificar quantas vezes cada valor de *precision* e *recall* ocorre durante o processo de avaliação.
4. Distribuição Acumulada de *Precision* e *Recall*: medida estatística básica conhecida como CDF (*Cumulative Distribution Function*), que possibilita uma visão geral das distribuições.
5. *F-measure* (F1): representa a combinação desejada de *precision* e *recall*, por meio da média harmônica penalizada [Bobadilla et al. 2013].
6. *Average Precision*: representa a precisão média de um conjunto de k itens recomendados, a fim de mostrar a relevância do conjunto de k primeiros itens apresentados [Herlocker et al. 2004].
7. *Mean Average Precision*: consiste em calcular a média dos valores de *Average Precision*, considerando o conjunto de itens recomendados [Herlocker et al. 2004].
8. *Hit Rate* (Acurácia): representa a quantidade de itens recomendados corretamente, com base no conjunto de itens consumidos pelo usuário alvo (teste) [Bobadilla et al. 2013].
9. *Mean Absolute Error* (MAE): representa a distância entre o *rating* real, atribuído pelo usuário u ao item i, para o *rating* previsto pelo recomendador [Bobadilla et al. 2013].
10. *Root Mean Squared Error* (RMSE): representa a distância quadrática entre o *rating* real para o *rating* previsto pelo recomendador. A diferença dos valores quadrados é utilizada a fim de penalizar erros maiores [Bobadilla et al. 2013].

Recentemente, alguns trabalhos observaram que apenas as métricas de acurácia não são suficientes para avaliar a eficácia prática das recomendações [Wu et al. 2012]. Na maioria dos casos, o objetivo de recomendação é inerentemente ligado a uma noção de descoberta, uma vez que a recomendação traz maiores benefícios quando esta expõe o usuário a uma experiência relevante, que ele não teria encontrado sozinho [Adomavicius and Tuzhilin 2005]. Neste contexto, métricas que remetem surpresa são características altamente desejáveis, pois refletem a satisfação dos usuários. Estes conceitos de surpresa estão relacionados às métricas de novidade, diversidade e serendipidade [Ricci et al. 2011]. Apesar de existirem diversas implementações para as métricas de novidade e diversidade, grande destaque pode ser dado ao *framework* proposto em [Vargas and Castells 2011]. Por outro lado, o conceito de serendipidade relacionado à recomendação está bem consolidado em [Zhang et al. 2012]. Todas essas métricas relacionadas à satisfação dos usuários foram consolidadas no grupo denominado *Complementary Dimensions of Quality*, sumarizadas na tabela 2.

Tabela 2. *Complementary Dimensions of Quality*: Métricas relacionadas à satisfação dos usuários.

1. Novidade: se refere a quão diferente um item recomendado é com relação a todos os outros que foram previamente consumidos pelo usuário alvo. É mensurado por meio da distância entre os itens recomendados e os itens do perfil de cada usuário, como proposto no *framework* de [Vargas and Castells 2011].
2. Diversidade: se aplica a um conjunto de itens, e está relacionada com o quão diferente os itens são com relação uns aos outros [Ricci et al. 2011]. Assim como em [Vargas and Castells 2011], diversidade é mensurada como a distância média esperada de um item para uma lista de itens (ILD).
3. Serendipidade: é uma forma de mensurar o quão surpreso o usuário ficou com o sucesso das recomendações. Calculamos a serendipidade por meio do complemento da similaridade de cosseno dos itens presentes no histórico de um usuário e as recomendações geradas [Zhang et al. 2012].
4. Cobertura de Catálogo: é a média dos itens relevantes que são recomendados pelo menos uma vez. Valores mais altos de cobertura do catálogo indicam que o algoritmo contrabalança o viés de popularidade cobrindo uma grande parte do conjunto geral de itens [Puthiya Parambath et al. 2016].
5. Cobertura de Gênero: é proporção média de gêneros (i.e., características dos itens) relevantes recomendados ao usuário, a fim de medir os interesses do usuário [Puthiya Parambath et al. 2016].

De maneira complementar, surgiu a necessidade de analisar o conjunto de dados ao qual os SsR são aplicados, a fim de evidenciar o comportamento dos usuários em cada contexto. Trabalhos recentes apontam a existência de um viés no consumo dos usuários, que priorizam itens populares, prejudicando estratégias que são capazes de recomendar itens não populares [Lee and Lee 2015]. Assim, avaliar características como popularidade dos itens, histórico de consumo dos usuários, distribuições de *ratings* atribuídos, dentre outras, são relevantes para entender os resultados obtidos pelos SsR. Consolidamos essas métricas relacionadas à caracterização de domínio em um conjunto denominado *Domain Profiling*, sumarizadas na tabela 3.

Tabela 3. Características das coleções utilizadas no módulo de *Domain Profiling*.

1. Popularidade dos itens: é a quantidade de usuários distintos que consumiu cada item da coleção. Essa distribuição nos permite, por exemplo, analisar se os itens recomendados estão enviesados ou não.
2. Histórico de consumo dos usuários: é a quantidade de itens distintos consumidos por cada usuário, para, por exemplo, demonstrar se uma técnica está relacionado a assiduidade de cada usuário.
3. Nota média de cada usuário: calculamos a média dos *ratings* atribuídos por cada usuário, afim de verificar, por exemplo, se o desempenho das técnicas está relacionado as notas atribuídas pelos usuários.
4. Nota média de cada item: calculamos a média dos *ratings* recebidos por cada item, a fim de verificar, por exemplo, se o desempenho das técnicas estão relacionadas as notas recebidas pelos itens.
5. Variância média das notas do usuário: calculamos a variância dos *ratings* atribuídos por cada usuário, a fim de verificar, por exemplo, se o desempenho das técnicas está relacionado a esta variância.
6. Variância média das notas dos itens: calculamos a variância dos *ratings* atribuídos a cada item, a fim de verificar, por exemplo, se o desempenho das técnicas está relacionado a esta variância.
7. Probabilidade dos *ratings* atribuídos: calculamos o número de vezes que cada *rating* distinto ocorre, para, por exemplo, compreender o viés dos usuários no desempenho de cada técnica.

4. FAiR - Framework de Análises em SsR

Nesta seção apresentamos a nossa segunda contribuição que consiste na implementação, teste e consolidação das métricas apresentadas na seção anterior em um *framework*, o qual denominamos de *FAiR - Framework for Analyses in Recommender Systems*.

4.1. Arquitetura

De maneira geral, organizamos o *framework* em três módulos: (1) *Effectiveness-based*; (2) *Complementary Dimensions of Quality*; e (3) *Domain Profiling*. Basicamente, por meio dos módulos (1) e (2), pretendemos avaliar qualquer recomendador sob os três requisitos de qualidade fundamentais: utilidade, novidade e diversidade. Com o módulo (3), avaliamos as características comumente relevantes dos domínios de recomendação. Um uso prático dessa caracterização é a identificação de padrões relevantes e recorrentes que permitam entender o desempenho das estratégias de acordo com características de cada domínio. Além dos três módulos principais, nosso *framework* possui outros dois módulos: (4) *Input Configuration* e (5) *Output Evaluation*. O *Input Configuration* é responsável pela configuração e gerência dos dados de entrada. Por sua vez, o *Output Evaluation* é responsável pela apresentação dos resultados das análises solicitadas.

4.2. Interface do Sistema

Com base na arquitetura apresentada, consolidamos o *FAiR* em uma interface que preza pela legibilidade e completude, a fim de bem cooperar com os pesquisadores da área. Na aba "Input Configuration" os usuários são responsáveis por prover os arquivos que são necessários para o processo de avaliação realizado pelo *FAiR*. Além disso, os usuários também precisam repassar para o *framework* o arquivo contendo as recomendações a serem avaliadas. Nesta etapa, o usuário pode selecionar um arquivo de recomendação no formato especificado por nosso *framework* ou pode também informar qual biblioteca foi utilizada, para que o *FAiR* realize as conversões, quando necessário[4]. No intuito de facilitar a interação do usuário, nessa mesma aba há a opção "Help" que apresenta uma descrição bem completa de todos os formatos de arquivo. Além disso, as opções "Generate Feature File" e "Generate Train and Test File" podem ser usadas para que o *FAiR* gere automaticamente os arquivos de características, treino e teste. No último passo, o usuário deve informar a quantidade de itens presentes nas listas de recomendações. Na aba "Choice Metric", os usuários devem escolher quais métricas deseja que sejam avaliadas.

Com os parâmetros configurados, os usuários podem finalmente clicar no botão "Run". A princípio dois diretórios são criados a partir do caminho de saída especificado. "Domain Profiling" é o diretório com o resultado de todas as métricas de extração de características. O segundo diretório está relacionado à avaliação de qualidade e receberá o nome do recomendador utilizado. Caso sejam avaliados mais de um recomendador, será criado um diretório para cada um deles. Aninhado a este, outros dois diretórios são criados, de acordo com as métricas de avaliação (*Effectiveness-based* e *Complementary Dimensions of Quality*), de tal forma que os resultados sejam devidamente organizados. Finalmente, na aba "Output Evaluation", o *FAiR* apresenta duas maneiras de visualizar o resultados encontrados: utilizando a própria interface ou pelo diretório de saída previamente definido. Na interface há informações relacionadas a quantidade de itens e usuários nas coleções de dados informadas como entrada e duas opções para cada métrica definida na aba anterior, uma para visualizar os resultados obtidos por meio de gráficos e outra que possibilita o usuário inspecionar o arquivo gerado diretamente.

[4] *LensKit* segue todos os formatos utilizados pelo *FAiR*. No caso da *MyMediaLite*, existe uma opção que ao ser selecionada muda o formato das listas de recomendação para o formato da saída dessa biblioteca.

4.3. Configuração de Sistema

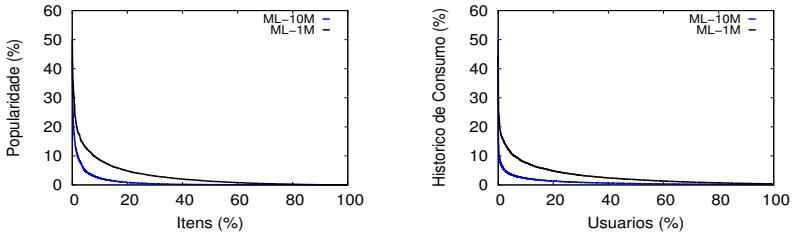
FAiR é uma aplicação *Desktop* para sistemas operacionais baseado em Linux, testado e homologado no sistema operacional Ubuntu 16.04. Toda sua interface gráfica foi implementada em Python utilizando GTK+3. Em termos de bibliotecas extras, foram utilizadas as bibliotecas (a) numpy; (b) matplotlib; e (3) seaborn. Todos os códigos estão publicamente disponíveis a partir do endereço *https://github.com/dcomp-labPi/FAiR*.

5. Validação e Aplicação Prática do FAiR

No intuito de validar nosso *framework*, apresentamos um estudo de caso que aponta as vantagens de utilizar o *FAiR* no processo de pesquisa. Em um processo de pesquisa convencional, onde um pesquisador pretende propor um novo recomendador, este se preocupa em estudar os principais *baselines* existentes, a fim de encontrar algum ponto fraco que possa ser superado. Entretanto, em domínios amplamente estudados, poucos pesquisadores se preocupam em avaliar o comportamento dos itens e usuários, negligenciando informações importantes. Além disso, no processo de avaliação, poucos são os pesquisadores que se preocupam em comparar sua abordagem com todas as categorias de SsR, por muitas vezes não terem em mãos as ferramentas necessárias para avaliar distintos recomendadores. Neste contexto, selecionamos duas coleções estudadas na literatura, disponibilizadas pelo *GroupLens*[5]: *MovieLens* 1M e *MovieLens* 10M. Para demonstrar a compatibilidade do *FAiR* com as bibliotecas da literatura, utilizamos os recomendadores implementados por *MyMediaLite* e *LensKit*. Selecionamos as estratégias *UserKNN* e *ItemKNN* [Sarwar et al. 2001], relacionadas à abordagem de filtragem colaborativa, e a estratégia de *MostPopular* [Herlocker et al. 2004], relacionada à abordagem não personalizada. De maneira complementar, selecionamos a estratégia de *PureSVD* [Cremonesi et al. 2010], referente à abordagem de fatoração de matrizes.

5.1. Características do Domínio

Por meio das métricas implementadas no módulo *"Domain Profiling"*, o *FAiR* consegue ressaltar o comportamento dos usuários e itens de qualquer domínio, bem como as relações de consumo altamente desejáveis para um recomendador. Avaliando o histórico de consumo dos usuários e a popularidade dos itens, na figura 1, observa-se um comportamento similar a *long-tail* [Anderson 2008]. Em outras palavras, existem poucos itens que são muito consumidos por poucos usuários do domínio. Essa característica indica, por exemplo, que o recomendador a ser proposto para este domínio deve se preocupar com o viés da popularidade de alguns itens.

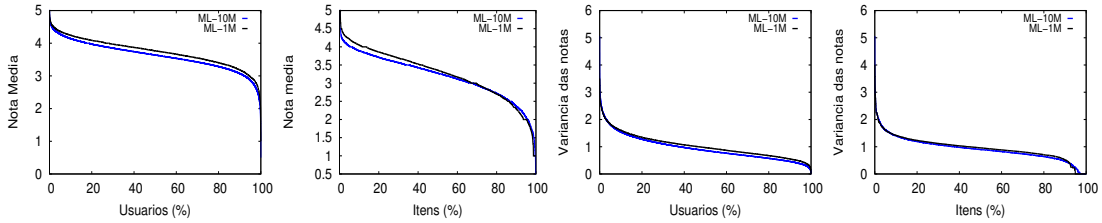


(a) Popularidade (b) Histórico de Consumo

Figura 1. Distribuição de popularidade e consumo dos itens de cada domínio.

[5]Disponíveis em: http://www.grouplens.org/node/12

FAiR pode ser usado para analisar o comportamento dos usuários e itens do domínio. Tais análises visam evidenciar para o pesquisador se abordagens *user-user* tendem a representar mais características do que abordagens *item-item*, ou vice-versa. Entender corretamente o domínio estudado permite modelar melhores estratégias para satisfazer os usuários com recomendações precisas. Neste intuito, a figura 2 mostra as distribuições de nota média dos usuários e itens, bem como a variância dessas notas em todo o domínio.



(a) Ratings usuários (b) Ratings itens (c) Variância ratings (d) Variância ratings

Figura 2. Distribuições médias e variância dos ratings de cada domínio.

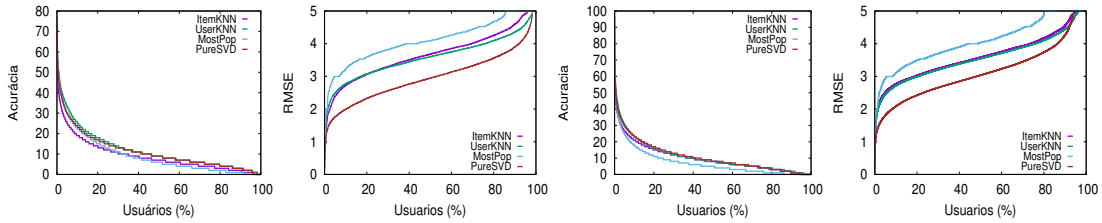
5.2. Análise de Qualidade

Como ponto principal do processo de pesquisa, o pesquisador deve analisar seu recomendador frente aos principais *baselines*. Para essa análise, a coleção de dados é dividida em subconjuntos de treino (70%) e teste (30%), preservando a ordenação temporal de consumo dos usuários. Em seguida, o pesquisador fornece o subconjunto de treino para alguma biblioteca com os principais SsR implementados, a fim de gerar uma lista de recomendação de n itens para cada usuário. Estes itens recomendados são avaliados frente a um conjunto de métricas, até então, implementadas pelo próprio pesquisador. Por meio do *FAiR*, todo este processo se torna mais fácil, prático e confiável. Desde a geração dos subconjuntos de treino e teste até a avaliação dos resultados, uma vez que o *FAiR* apresenta as principais métricas de avaliação por meio de uma interface intuitiva.

Para simular este processo de análise, cada recomendador escolhido gera uma lista com 100 itens. Para cada cenário, foram avaliadas todas as métricas do módulo *Effectiveness-based*. A figura 3 apresenta os resultados das métricas de acurácia e RMSE do *FAiR*. Com esses é possível notar que, para ambos cenários do ML-1M e ML-10M, *PureSVD* possui maiores valores de RMSE (3(b), (d)), sendo aquela cujo os *ratings* estimados mais se aproximam dos *ratings* reais atribuídos pelos usuários. Quando comparamos as estratégias por meio de acurácia (3(a), (c)) notamos um comportamento similar de *PureSVD* e *UserKNN*, ambas superando as demais estratégias. Outro aspecto importante é que apesar das estratégias *UserKNN* e *ItemKNN* terem abordagens similares, o comportamento delas é distinto na prática. *UserKNN* possui um melhor desempenho que *ItemKNN*. Além disso, pode-se notar que a estratégia de *MostPopular*, apesar de ser não-personalizada, possui um desempenho satisfatório.

5.3. Dimensões Complementares de Qualidade

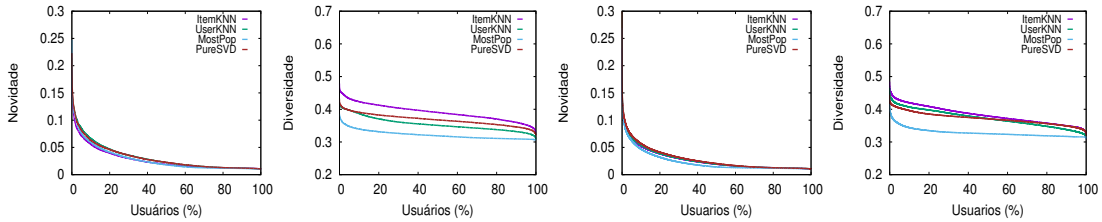
Além de apresentar avaliações de dimensões altamente desejáveis na análise de qualidade de um recomendador, *FAiR* também permite ao pesquisador analisar a recomendação sobre os aspectos complementares de qualidade, tais como diversidade e novidade, conforme apresentado na figura 4. A análise dos resultados gerados pelo *FAiR*, possibilita notar que todos os recomendadores apresentam níveis similares de novidade (figuras 4 (a) e (c)). Entretanto, nota-se também que *ItemKNN* possui maior diversidade que as demais



(a) ML-1M - Accuracy (b) ML-1M - RMSE (c) ML-10M - Accuracy (d) ML-10M - RMSE

Figura 3. Acurácia e RMSE de cada domínio e recomendador utilizado.

estratégias (figuras 4 (b) e (d)). Uma hipótese pertinente, neste caso, seria: abordagens *item-item* apresentam mais diversidade nas recomendações geradas? Além disso, destaca-se o baixo desempenho da estratégia *MostPopular* em apresentar diversidade de itens aos usuários. Neste caso, uma correlação com a distribuição de popularidade mostrada na figura 1, pode gerar a questão: abordagens focadas em itens populares não apresentam tanta diversidade, pois estes itens são consumidos pelo mesmo grupo de usuários? Novamente, *FAiR* não possui todas as respostas. Ao contrário, o seu uso permite ao pesquisador elaborar novas perguntas pertinentes ao processo de pesquisa.



(a) ML-1M Novidade (b) ML-1M Diversidade (c) ML-10M Novidade (d) ML-10M Diversidade

Figura 4. Novidade e diversidade de cada recomendador em cada domínio.

6. Conclusões & Trabalhos Futuros

Neste trabalho, consolidamos *FAiR*, um *framework* de métricas capazes de caracterizar e avaliar distintos recomendadores. Tal *framework* consiste em módulos que visam avaliar e distinguir cada recomendador frente aos requisitos de utilidade, novidade e diversidade. Além disso, esses módulos permitem identificar as principais características das coleções de dados utilizadas que possam estar intrinsecamente relacionadas ao desempenho dos recomendadores. O estudo de caso realizado sobre os dois principais conjuntos da *MovieLens*, nos permite vislumbrar a importância de se correlacionar as características do domínio para entender o comportamento dos recomendadores. Dessa forma, *FAiR* é um *framework* prático e efetivo para auxiliar o processo de pesquisa em SsR, uma vez que possui as principais métricas de avaliação. Futuramente, pretendemos adicionar outras métricas ao *FAiR*, bem como realizar correlações entre as métricas de avaliação e as características relacionadas aos domínios estudados.

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 transactions on knowledge and data engineering*, 17(6):734–749.

Anderson, C. (2008). Long tail, the, revised and updated edition: Why the future of business is selling less of more.

Bobadilla, J., Ortega, F., Hernando, A., and Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems*, 46:109–132.

Cremonesi, P., Koren, Y., and Turrin, R. (2010). Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of ACM RecSys*, pages 39–46.

Ekstrand, M. D., Ludwig, M., Kolb, J., and Riedl, J. T. (2011). Lenskit: a modular recommender framework. In *Proc. of ACM RecSys*, pages 349–350.

Gantner, Z., Rendle, S., Freudenthaler, C., and Schmidt-Thieme, L. (2011). MyMediaLite: A free recommender system library. In *Proc. of 5th ACM RecSys*.

Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM TOIS*, 22(1):5–53.

Konstan, J. A. and Riedl, J. (1999). Research resources for recommender systems. In *CHI99 Workshop Interacting with Recommender Systems*.

Kouki, P., Fakhraei, S., Foulds, J., Eirinaki, M., and Getoor, L. (2015). Hyper: A flexible and extensible probabilistic framework for hybrid recommender systems. In *Proc. of ACM RecSys*, pages 99–106.

Lee, K. and Lee, K. (2015). Escaping your comfort zone: A graph-based recommender system for finding novel recommendations among relevant items. *Expert Systems with Applications*, 42(10):4851–4858.

Lu, J., Wu, D., Mao, M., Wang, W., and Zhang, G. (2015). Recommender system application developments: a survey. *Decision Support Systems*, 74:12–32.

Modarresi, K. (2016). Recommendation system based on complete personalization. pages 2190 – 2204. ICCS 2016, California, USA.

Ozok, A. A., Fan, Q., and Norcio, A. F. (2010). Design guidelines for effective recommender system interfaces based on a usability criteria conceptual model: results from a college student population. *Behaviour & Information Technology*, 29(1):57–83.

Puthiya Parambath, S. A., Usunier, N., and Grandvalet, Y. (2016). A coverage-based approach to recommendation diversity on similarity graph. In *Proc. of ACM RecSys*, pages 15–22.

Ricci, F., Rokach, L., and Shapira, B. (2011). *Introduction to recommender systems handbook*. Springer.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proc. ACM World Wide Web*, pages 285–295.

Schwartz, B. (2009). *The Paradox of Choice: Why More Is Less, Revised Edition*. HarperCollins.

Vargas, S. and Castells, P. (2011). Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of ACM RecSys*, pages 109–116.

Wu, W., He, L., and Yang, J. (2012). Evaluating recommender systems. In *Proc. of IEEE ICDIM*, pages 56–61.

Zhang, Y. C., Séaghdha, D. Ó., Quercia, D., and Jambor, T. (2012). Auralist: introducing serendipity into music recommendation. In *Proc. of ACM WSDM*, pages 13–22.