# AppRecommender: um recomendador de aplicativos GNU/Linux

**Tássia Camões Araújo** [1]

[1]Instituto de Matemática e Estatística – Universidade de São Paulo (USP)
Rua do Matão, 1010 – Cidade Universitária – São Paulo, SP – Brazil

tassia@ime.usp.br

***Abstract.*** *The increasing availability of open source software on the World Wide Web exposes potential users to a wide range of choices. Given the individuals plurality of interests, mechanisms that get them close to what they are looking for would benefit themselves and the software developers as well.* AppRecommender *is a recommender system for GNU/Linux applications which performs a filtering on the set of available software and individually offers suggestions to users. This is achieved by analyzing profiles and discovering patterns of behavior of the studied population, in a way that only those applications considered most prone to acceptance are presented to users.*

***Resumo.*** *A crescente oferta de programas de código aberto na rede mundial de computadores expõe potenciais usuários a inúmeras possibilidades de escolha. Em face da pluralidade de interesses destes indivíduos, mecanismos eficientes que os aproximem daquilo que buscam trazem benefícios para eles próprios, assim como para os desenvolvedores dos programas. O* AppRecommender *é um recomendador de aplicativos GNU/Linux que realiza uma filtragem no conjunto de programas disponíveis e oferece sugestões individualizadas para os usuários. Tal feito é alcançado por meio da análise de perfis e descoberta de padrões de comportamento na população estudada, de sorte que apenas os aplicativos considerados mais suscetíveis a aceitação sejam oferecidos aos usuários.*

## 1. Introdução

O universo de programas livres e de código aberto oferece aos usuários uma grande amplitude e diversidade de opções no que diz respeito a aplicativos para complementar seus sistemas. No entanto, muitas dessas alternativas permanecem em relativa obscuridade, pois o caráter majoritariamente não comercial desses sistemas se reflete na ausência de propaganda e outras formas de divulgação ostensiva. Desta forma, a descoberta de programas úteis para um determinado usuário por vezes empaca no excesso de informações disponíveis e organização inadequada. É costume referir-se a esse fenômeno (p. ex., [Iyengar 2010]) como "mais é menos", no sentido de que o aumento da disponibilidade de escolhas pode confundir o usuário e diminuir sua satisfação.

Neste contexto de muitas possibilidades onde poucas são de fato atrativas, um sistema capaz de recomendar aplicativos que presumidamente são objeto de interesse de usuários exerceria um papel importante. Desenvolvedores se beneficiariam por meio de um consequente aumento na utilização de seus programas que, por serem experimentados

por mais usuários, certamente receberiam mais relatórios de erro (*bug reports*), sugestões e contribuições diversas. Para os usuários o benefício seria alcançado de forma mais direta, dado que poupariam tempo e recursos outrora dedicados a buscas e filtragens manuais para encontrar os aplicativos mais adequados a seu ambiente de trabalho.

Tais benefícios motivaram a concepção do *AppRecommender*, um recomendador de aplicativos GNU/Linux desenvolvido no âmbito de um trabalho de mestrado, cujo objetivo principal é a experimentação de diferentes estratégias para recomendação no contexto de componentes de software.

O presente trabalho está organizado da seguinte forma: as seções 2 e 3 trazem uma breve introdução sobre distribuições GNU/Linux e sistemas de recomendação. A seção 4 apresenta o AppRecommender como solução em desenvolvimento para o problema exposto. Em seguida, na seção 5 trabalhos correlatos são apresentados e, por fim, a seção 6 traz considerações finais sobre o trabalho até a atual etapa de execução.

## 2. Distribuições GNU/Linux

O surgimento das distribuições GNU/Linux foi desencadeado pelo esforço do projeto GNU[1] em desenvolver um sistema operacional livre alternativo ao UNIX[2], juntamente com o surgimento do kernel Linux que foi desenvolvido inicialmente como um trabalho de graduação e posteriormente licenciado sob a GNU GPL[3].

Entre as distribuições GNU/Linux mais populares estão o Debian, Fedora, Mandriva e Ubuntu, que oferecem diferentes "sabores" deste sistema operacional constituídos por milhares de aplicativos selecionados por seus desenvolvedores. O processo de desenvolvimento e manutenção destes projetos está diretamente ligado à sua constituição, podendo variar entre o modelo gerido totalmente por voluntários ao coordenado por uma corporação, com diferentes níveis de interferência externa. A seleção dos aplicativos básicos de uma distribuição é ponto crucial do seu desenvolvimento e frequentemente é motivo de polêmica entre os colaboradores, visto que este é um dos fatores de grande influência na escolha dos usuários por uma distribuição ou outra.

As distribuições reduzem a complexidade de instalação e atualização do sistema para usuários finais na medida em que atuam como intermediários entre os autores dos programas e os usuários, por meio do encapsulamento de componentes de software em abstrações denominadas *pacotes* [Cosmo et al. 2008]. Tal infraestrutura facilita o processo de instalação do sistema básico, mas ainda assim, a seleção dos aplicativos que atendam a suas demandas específicas é de responsabilidade do usuário do sistema. Com o desenvolvimento deste trabalho, pretende-se auxiliar o indivíduo nesta tarefa, especialmente quando ele não possuir experiência pessoal para realizar escolhas neste contexto.

## 3. Sistemas de Recomendação

Sistemas de recomendação emergiram como uma área de pesquisa independente na década de 90, apoiando-se em soluções nas áreas de ciência cognitiva, teoria da aproximação, recuperação da informação, inteligência artificial, teorias de predição,

[1] `http://www.gnu.org`

[2] `http://www.unix.org/`

[3] Acrônimo para *General Public License*, é um suporte legal para a distribuição livre de software.

administração e marketing [Adomavicius and Tuzhilin 2005]. O tema ganhou destaque com o crescimento do comércio eletrônico, onde apresentar o que o usuário tem interesse pode significar conquistar o cliente.

Os recomendadores fazem a associação entre objetos e pessoas neles interessadas, filtrando as informações de forma a apresentar somente aquilo que seja relevante para o usuário. Além da agilidade para encontrar o que se deseja, tais sistemas possibilitam a personalização de serviços e conteúdos, permitindo que sejam apresentados de maneira individualizada a partir da identificação de interesses pessoais.

O problema abordado é comumente formalizado através de uma estrutura de pontuação como representação computacional da utilidade dos itens para os usuários. A partir de avaliações feitas pelos próprios usuários, tenta-se estimar pontuações para os itens que ainda não foram avaliados. Uma vez que esta estimativa tenha sido feita, pode-se recomendar os itens com maior pontuação estimada. Tal representação computacional impõe inúmeros desafios ao desenvolvimento de sistemas recomendadores, entre os quais, a escalabilidade dos algoritmos, a acurácia das recomendações e a dificuldade em lidar com dados esparsos [Vozalis and Margaritis 2003].

Algumas técnicas utilizadas na composição de recomendações são herdadas do tratamento de problemas clássicos como classificação e recuperação de informação em documentos de texto. A recomendação pode ser vista como uma classificação, na qual os itens são categorizados entre duas classes: relevantes e irrelevantes. Algoritmos de aprendizado de máquina geralmente são utilizados para tal abordagem, como *K-NN* e *classificadores bayesianos*. Por outro lado, a identidade ou o comportamento do usuário pode representar uma consulta num sistema de busca implementado sobre o conjunto de aplicativos. *Tf-idf* e *Okapi BM25* figuram entre soluções populares para implementação de buscadores. Outras abordagens baseiam-se em técnicas de mineração de dados, como o *Apriori*, algoritmo para descoberta de regras de associação entre itens a partir da análise de uma base de dados de transações.

## 4. AppRecommender

Um recomendador de aplicativos GNU/Linux – *AppRecommender* – está sendo desenvolvido para viabilizar a experimentação de diferentes estratégias de recomendação. Neste cenário, os pacotes são modelados como itens e as instâncias de sistemas instalados como usuários do recomendador. O fluxo da recomendação é o seguinte: dada a lista de pacotes instalados num determinado sistema (como representação de identidade), deve-se retornar uma lista de pacotes sugeridos composta por aplicativos de potencial interesse para tal usuário.

Uma peculiaridade deste trabalho é que o usuário (sistema instalado) não corresponde a um indivíduo humano: uma máquina pode ter mais de um administrador, assim como uma pessoa pode administrar diversas máquinas. Ainda assim, acredita-se que perfis possam ser traçados satisfatoriamente. No entanto, apenas um ser humano (administrador do sistema) pode atestar a relevância dos itens recomendados, ao escolher entre instalar ou não os itens sugeridos.

A distribuição escolhida como base para o desenvolvimento deste trabalho foi o

Debian GNU/Linux[4], com base nos seguintes critérios: existência de um esquema consistente de distribuição de aplicativos; disponibilidade de dados estatísticos; possibilidade de integração dos resultados e popularidade da distribuição. No entanto, a codificação está sendo realizada com o maior grau possível de independência de plataforma, com o intuito de que os resultados sejam facilmente adaptáveis para outros contextos.

O código-fonte do *AppRecommender* está licenciado sob a GNU GPL versão 3 e hospedado no repositório de projetos GitHub[5]. A linguagem de programação escolhida foi *Python*[6], principalmente pela facilidade de integração com outras ferramentas do Debian também desenvolvidas nesta linguagem. Ademais, a vasta documentação e grande variedade de bibliotecas de utilidade para o trabalho, a exemplo da *Xapian*[7], são fatores que contribuíram para esta escolha.

As fontes de dados principais para a computação de recomendações são *Popcon*[8], *Debtags*[9] e *UDD*[10]. A utilização de dados demográficos também está sendo considerada. Por exemplo, a declaração explícita por parte do usuário de que não tem interesse por determinado nicho de aplicativos eliminaria de antemão uma série de pacotes que a princípio seriam considerados, evitando anomalias nos resultados.

### 4.1. Seleção de atributos

Um pacote que compõe a instalação padrão pode ter dois estados num sistema funcional: (a) já faz parte do sistema ou (b) foi propositalmente removido pelo usuário. Assume-se que em ambos os casos, o pacote não seria de interesse do usuário, portanto, é previamente desconsiderado pelo recomendador. Pacotes instalados automaticamente, em decorrência da instalação de outros pacotes dos quais são dependência, também não devem compor o perfil do sistema.

Os dados disponíveis atualmente permitem apenas a indicação de relevância binária – um item é relevante ou irrelevante – pela presença do aplicativo no sistema. No entanto, a atribuição de pesos diferentes para pacotes que são utilizados com muita frequência e os que após a instalação foram acessados poucas vezes é uma característica desejável. Formas alternativas de aquisição desta informação estão sendo investigadas, visto que os dados temporais coletados pelo *Popcon* não são seguramente corretos[11].

Existem ainda algumas questões relativas ao pré-processamento dos dados e seleção de atributos específicas para o contexto deste trabalho, entre elas: (a) em que medida as relações de dependência entre pacotes devem interferir nas recomendação e (b) a recomendação de um conjunto de pacotes assume que as necessidades dos usuários seriam supridas pela instalação dos mesmos, no entanto, necessidades talvez sejam melhor representadas por funcionalidades em detrimento de aplicativos específicos. Desta forma, a consideração de conceitos como Debtags e pacotes virtuais, por exemplo, devem trazer benefícios ao cálculo.

[4] `http://www.debian.org`
[5] `http://github.com/tassia/AppRecommender`
[6] `http://www.python.org/`
[7] `http://xapian.org/`
[8] `http://popcon.debian.org`
[9] `http://debtags.alioth.debian.org/`
[10] `http://udd.debian.org/`
[11] `http://popcon.debian.org/README`

### 4.2. Estratégias de recomendação

(a) **Baseada em conteúdo**
Esta abordagem parte do princípio de que os usuários tendem a se interessar por itens semelhantes àqueles pelos quais já se interessaram no passado [Herlocker 2000]. Os itens são caracterizados por atributos, a partir dos quais aplica-se técnicas de recuperação da informação ou classificação para encontrar itens semelhantes. Portanto, analisando a lista de pacotes já instalados em um sistema pode-se recomendar novos programas a serem instalados. A implementação atual utiliza Debtags e descrição dos pacotes como alternativas para a caracterização dos pacotes.

(b) **Colaborativa**
A estratégia colaborativa é fundamentada na troca de experiências entre indivíduos que possuem interesses em comum. A essência desta solução, baseada no algoritmo *K-NN*, está na representação de proximidade entre os usuários. A vizinhança de um determinado usuário é composta pelos $k$ usuários que estiverem mais próximos a ele. Uma recomendação é produzida a partir da análise dos pacotes instalados por seus vizinhos, sendo composta pelos programas que ocorrem com maior frequência nesta vizinhança. Esta estratégia está em fase de implementação.

(c) **Híbrida**
Sistemas de recomendação híbridos combinam duas ou mais estratégias, com o intuito de obter melhor performance do que a que as estratégias oferecem individualmente [Burke 2002]. Os resultados produzidos por recomendadores simples podem ser refinados através de implementações em *cascata*. Fontes de dados adicionais podem ser consideradas para reposicionamento dos itens sugeridos, entre elas: indicação de áreas de interesse pelo usuário, popularidade dos pacotes, relatórios de erros pendentes para cada pacote etc. Outra possibilidade é a elaboração de estratégias para *revezamento* entre os sistemas básicos, de acordo com a natureza dos dados de entrada. Ou ainda a *combinação* dos resultados de múltiplos recomendadores para compor a sugestão apresentada ao usuário.

### 4.3. Avaliação

Métricas de acurácia medem o quanto as estimativas previstas pelo sistema se aproximam da real. Por exemplo, a *precisão* mede a proporção de itens relevantes entre os recomendados e a *recuperação* se refere a proporção de itens recomendados entre todos os relevantes de fato [Herlocker et al. 2004].

A avaliação das diferentes estratégias de recomendação, diferentes abordagens de seleção de atributos, bem como o ajuste de parâmetros dos algoritmos está sendo realizada através de rodadas de validação cruzada. Dado que o conjunto de pacotes instalados em um sistema é sabidamente relevante para o mesmo, um subconjunto deste é selecionado aleatoriamente para fins de teste e o conjunto restante é utilizado para gerar a recomendação. As métricas são então aplicadas aos conjuntos de teste, e a comparação é realizada a partir dos resultados obtidos para uma série de experimentos.

## 5. Trabalhos correlatos

Grande parte das distribuições GNU/Linux têm investido no desenvolvimento de interfaces para facilitar o gerenciamento de aplicativos e a forma como se obtém informações

sobre os mesmos, e a recomendação de aplicativos já é pauta de discussões. Entre os dias 18 e 21 de janeiro 2011 aconteceu a primeira reunião sobre a temática com a presença de desenvolvedores de distribuições variadas (*Cross-distribution Meeting on Application Installer*[12]). O encontro teve como principais objetivos a definição de padrões a serem implementados com o intuito de facilitar a interoperabilidade entre as ferramentas e compartilhamento de informações entre os diferentes projetos.

Dois esforços anteriores de desenvolvimento de recomendadores de pacotes Debian foram identificados, porém ambos descontinuados. A primeira foi o *PopSuggest*[13], que oferecia recomendações a partir de dados do *Popcon* como uma ilustração das possibilidades de uso dos dados coletados. A outra foi o *Debommender*[14], desenvolvido como prova de conceito no âmbito de um trabalho de graduação, não sendo porém integrado aos serviços da distribuição.

## 6. Considerações Finais

O presente texto relata o desenvolvimento de um trabalho em progresso que pretende extrair conhecimento a partir de dados de sistemas reais previamente coletados. Acredita-se que ao final do estudo um recomendador de pacotes possa ser integrado à infraestrutura do projeto Debian, e que de fato os usuários sejam surpreendidos com recomendações úteis e não óbvias, auxiliando-os a selecionar aplicativos mais adequados ao seu ambiente.

## 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.

Burke, R. (2002). Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction*, 12:331–370.

Cosmo, R. D., Zacchiroli, S., and Trezentos, P. (2008). Package upgrades in FOSS distributions: details and challenges. In *Proceedings of the 1st International Workshop on Hot Topics in Software Upgrades*, pages 7:1–7:5. ACM.

Herlocker, J. L. (2000). *Understanding and improving automated collaborative filtering systems*. PhD thesis.

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

Iyengar, S. (2010). *The Art of Choosing*. Twelve.

Vozalis, E. and Margaritis, K. G. (2003). Analysis of Recommender Systems' Algorithms. In *Proceedings of the 6th Hellenic European Conference on Computer Mathematics and its Applications*.


[12] `http://distributions.freedesktop.org/wiki/Meetings/AppInstaller2011`
[13] `http://www.enricozini.org/2007/debtags/popcon-play/`
[14] `http://ostatic.com/debommender`