

COMBINANDO INFORMAÇÃO CONTEXTUAL EM SISTEMAS DE RECOMENDAÇÃO DE PONTOS DE INTERESSE

Abner Ferreira Suniga (PIBIC/CNPq), Marcos Aurelio Domingues (Orientador), e-mail: ra93996@uem.br

Universidade Estadual de Maringá / Centro de Tecnologia / Maringá, PR

Área do conhecimento: Ciência da Computação

Subárea do conhecimento: Metodologia e Técnicas da Computação

Especialidade: Sistemas de Informação

Palavras-chave: Recomendação de pontos de interesse, dados georeferenciados, sistemas de recomendação sensíveis ao contexto.

Resumo: Com a necessidade de algoritmos que combinem informações contextuais de local e tempo para prover recomendações mais precisas, esse projeto trata de uma nova implementação de um algoritmo de recomendação sensível ao contexto que utiliza como informação contextual local e tempo, sendo este implementado na *framework CARSKit*.

Introdução

Um sistema de recomendação é uma tecnologia de filtragem de informação que pode ser utilizada para prever avaliações de itens e/ou gerar um ranking de itens que podem ser do interesse do usuário (Ricci et al., 2011). Os algoritmos utilizados por tais sistemas podem ser classificados em 3 categorias:

- Filtragem colaborativa - os itens recomendados ao usuário são itens que outros usuários, com preferências similares às dele, já escolheram anteriormente;
- Filtragem baseada em conteúdo - os itens recomendados ao usuário são similares aos itens que o usuário preferiu no passado;
- Abordagens híbridas - esses métodos combinam os métodos colaborativo e baseado em conteúdo.

Os sistemas de recomendação sensíveis ao contexto são sistemas que fazem recomendações considerando também informações de contexto, como local, tempo, etc. A importância da informação contextual tem sido reconhecida por pesquisadores e profissionais em muitas disciplinas, como personalização de sites de comércio eletrônico, recuperação de informação e computação móvel (Adomavicius e Tuzhilin, 2011).

De acordo com Adomavicius e Tuzhilin (2011), os gostos e preferências dos usuários são geralmente expressos como avaliações e modelados em função dos itens, usuários e contextos.

Em Levandoski et al. (2012) é apresentado o LARS (*Location-Aware Recommender System*), um sistema de recomendação sensível ao contexto

que usa avaliações baseadas na localização de usuários e itens para gerar as recomendações. Os autores descrevem três modos em que o LARS pode ser usado, sendo que para o nosso trabalho, escolhemos a geração de recomendações usando avaliações não espaciais para itens espaciais, que se baseia na ideia de usar uma penalidade conforme a distância do item para o usuário.

Basicamente o algoritmo implementado calcula para cada conjunto de usuário, item e contexto um *recScore* que é formado por uma predição $P(u, i)$ gerada por um método padrão de recomendação, nesse projeto usamos *ItemKNN* e o *BPR*, menos a penalidade $TravelPenalty(u, i)$ que é a distância entre o usuário e o item normalizada para os valores da escala da predição. Além disso, o algoritmo foi estendido para incorporar a informação contextual de tempo, funcionando de modo similar com uma forma de penalidade para quanto maior a diferença entre as horas.

Materiais e métodos

A implementação do algoritmo foi realizada utilizando a *framework CARSKit* (Zheng et al., 2015), que é uma ferramenta de recomendação de código aberto baseada em Java, que pode ser usada, modificada e distribuída sob a Licença Pública Geral GNU. A *framework CARSKit* é projetada especificamente para recomendações sensível ao contexto, e a versão contendo as implementações realizadas neste projeto está disponível online¹.

Resultados e Discussão

No projeto foi utilizado duas bases de dados, a do Yelp² e do Foursquare³. A base de dados do Yelp, que é um aplicativo que recomenda locais próximos a usuários por meio de avaliações de outros usuários, os dados consistem nas avaliações feitas de um usuário a um estabelecimento, logo cada avaliação apresenta ID do usuário, ID do item, rating, latitude e longitude. Esse contexto de latitude e longitude (dados-georeferenciados) representa tanto a localização do usuário quanto a do item no nosso modelo dependendo da fase de treino ou fase de teste. O Foursquare é uma empresa que fornece informações sobre atrações e restaurantes de acordo com a localização dada pelo usuário em um aplicativo. Quando um usuário deseja compartilhar em suas redes sociais que está em um estabelecimento, ele realiza um *check-in* no aplicativo do Foursquare. Nessas informações de *check-in*, além dos atributos ID do usuário, ID do item, rating, latitude e longitude, há também a hora que o *check-in* ocorreu.

Para avaliar os resultados das recomendações do algoritmo desenvolvido foram usadas as métricas *Precision*, *Recall* e *F-measure*,

1 <https://github.com/abnersuniga/CARSKit-pibic>

2 <https://www.yelp.com/dataset/challenge>

3 <http://www.public.asu.edu/~hgao16/dataset.html>

disponíveis na *framework CARSKit*, onde *Precision* está relacionado ao número de itens recomendados que são relevantes e o *Recall* ao número de itens relevantes que são recomendados. A *F-measure* é definida como uma média harmônica entre o *Precision* e o *Recall*.

Nas Tabelas 1 e 2, comparamos os resultados obtidos pelo nosso LARS (usando dados geo-referenciados, isto é, latitude e longitude, e também a informação de hora do acesso como contexto) com os resultados apresentados em Santana (2017), que usa algoritmos de *clustering* para transformar latitude e longitude em regiões geográficas. A comparação foi realizada usando 2 métodos de recomendação, ItemKNN e BPR.

Tabela 1 - Comparando resultados usando o método ItemKNN

Algoritmo de cluster	Resultados no Foursquare				Resultados no Yelp			
	Num. clusters	Precision	Recall	F-measure	Num. clusters	Precision	Recall	F-measure
Baseline	-	0,000873	0,006676	0,001518	-	0,003598	0,013619	0,005446
Aglomerativo	2	0,000873	0,006676	0,001518	2	0,003288	0,023944	0,005565
Aglomerativo	4	0,000884	0,006734	0,001538	4	0,003288	0,023944	0,005565
Aglomerativo	8	0,000867	0,006724	0,001512	8	0,003265	0,023937	0,005534
Aglomerativo	12	0,000867	0,006724	0,001512	12	-	-	-
Aglomerativo	16	0,000867	0,006724	0,001512	16	0,003265	0,023937	0,005534
DBSCAN	7	0,000872	0,006661	0,001519	8	0,003110	0,023861	0,005333
DBSCAN	12	0,000884	0,006734	0,001538	13	0,003110	0,023861	0,005333
DBSCAN	16	0,000884	0,006734	0,001538	-	-	-	-
DBSCAN	19	0,000884	0,006734	0,001538	23	0,003261	0,023933	0,005530
DBSCAN	26	0,000884	0,006734	0,001538	53	0,003261	0,023933	0,005530
KMeans	3	0,000884	0,006734	0,001538	2	0,003288	0,023944	0,005565
KMeans	5	0,000867	0,006724	0,001512	4	0,003288	0,023944	0,005565
KMeans	10	0,000857	0,007316	0,001517	8	0,003265	0,023937	0,005534
LARS 1.0	-	-	-	-	-	0,052002	0,519891	0,094545
LARS 1.1	-	0,088005	0,879802	0,160002	-	0,042277	0,422640	0,076863
LARS 1.2	-	0,022495	0,224947	0,194794	-	-	-	-

Tabela 2 - Comparando resultados usando o método BPR

Algoritmo de cluster	Resultados no Foursquare				Resultados no Yelp			
	Num. clusters	Precision	Recall	F-measure	Num. clusters	Precision	Recall	F-measure
Baseline	-	0,002319	0,016333	0,003874	-	0,002875	0,020781	0,004855
Aglomerativo	2	0,005544	0,044737	0,009704	2	0,003065	0,022937	0,005221
Aglomerativo	4	0,006157	0,049825	0,010772	4	0,003054	0,022941	0,005201
Aglomerativo	8	0,005946	0,046994	0,010376	8	0,002941	0,022375	0,005045
Aglomerativo	12	0,006079	0,047969	0,010601	12	-	-	-
Aglomerativo	16	0,005589	0,043755	0,009742	16	0,002749	0,020455	0,004705
DBSCAN	7	0,005394	0,042350	0,009408	8	0,003199	0,024231	0,005484
DBSCAN	12	0,005852	0,048167	0,010281	13	0,003361	0,026207	0,005795
DBSCAN	16	0,006382	0,052618	0,011222	23	0,002869	0,022062	0,004919
DBSCAN	19	0,005243	0,042015	0,009167	53	0,003042	0,022817	0,005194
DBSCAN	26	0,005622	0,045710	0,009859	-	-	-	-
KMeans	3	0,005907	0,046457	0,010282	2	0,002900	0,021289	0,004943
KMeans	5	0,005970	0,048917	0,010479	4	0,003080	0,022841	0,005239
KMeans	10	0,005643	0,046180	0,009912	8	0,003390	0,025482	0,005785
LARS 1.1	-	0,040148	0,401302	0,072991	-	0,002570	0,025700	0,004673
LARS 1.2	-	0,040881	0,408808	0,074329	-	-	-	-

O algoritmo do LARS desenvolvido sofreu modificações durante o projeto. Assim, os classificamos em 3 versões: LARS 1.0, utilizando como método para calcular a penalidade a distância euclidiana; LARS 1.1, que usa a fórmula de *Haversine* para gerar um cálculo mais preciso entre dois pontos de uma esfera através de suas latitudes e longitudes; e LARS 1.2, que é uma tentativa de melhoria do algoritmo original usando a informação contextual de tempo como penalidade.

Foram encontrados resultados inconsistentes entre as bases de dados e entre os métodos (ItemKNN e BPR). Porém o LARS se mostrou mais estável usando BPR, obtendo em média um aumento no *Precision* de 70% em relação aos outros e 80% no *Recall*. Notamos que houve também uma pequena melhora nos resultados do LARS 1.2 em comparação com LARS 1.1.

Conclusões

Além da contribuição com o desenvolvimento da versão modificada do LARS para a *framework CARSKit*, foi observado nos resultados uma melhora, apesar de pequena, do algoritmo com a extensão que usa a informação contextual de tempo. Como dificuldades encontradas ao implementar o algoritmo, pode-se mencionar o fato de que a ferramenta não possui uma documentação detalhada explicando o seu funcionamento.

Agradecimentos

Agradeço ao programa PIBIC/CNPq/FA/UEM pelo apoio financeiro e ao professor Marcos Aurélio Domingues pela orientação.

Referências

RICCI, F.; ROKACH, L.; SHAPIRA, B.; KANTOR, P. B. **Recommender Systems Handbook: A Complete Guide for Research Scientists and Practitioners**, 2011.

ADOMAVICIUS, G.; TUZHILIN, A. **Recommender Systems Handbook**. Chapter Context-Aware Recommender Systems, 2011.

LEVANDOSKI, J.J.; SARWAT, M.; ELDAWY, A.; MOKBEL, M. F. **LARS: A Location-Aware Recommender System**. IEEE ICDE Conference, 2012.

ZHENG, Y.; MOBASHER, B.; BURKE, R. **CARSKit: A Java-Based Context-aware Recommendation Engine**. IEEE ICDM Workshops, 2015.

SANTANA, I. A. P. **Aplicando o Processo de Mineração de Dados para Aquisição de Informação Contextual para Sistemas de Recomendação Sensíveis ao Contexto**. Monografia de graduação, 53 páginas, 2017.