Mineração de Preferências em *Data Streams*

Programa de Pós-Graduação em Ciência da Computação
da Universidade Federal de Uberlândia

Aluna:
Jaqueline Aparecida Jorge Papini
jaque@comp.ufu.br

Orientadora:
Profª Drª Sandra Aparecida de Amo
deamo@ufu.br

Nível: Mestrado

Ingresso no Programa: 2012/01 **Previsão da Defesa:** fevereiro de 2014

Etapas Concluídas:

- Aprovação do Plano de Trabalho 03/2013
- Proposta de Estratégias para Minerar Preferências em *Data Streams* 04/2013
- Proposta do Algoritmo FPSMining para Minerar Preferências em *Data Streams* 05/2013
- Submissão de Artigo para o KDMiLe 2013 – Aceito 06/2013
- Proposta do Algoritmo IncFPSMining para Minerar Preferências em *Data Streams* 06/2013
- Submissão de Artigo para o SBBD 2013 07/2013
- Submissão de Trabalho para o WTDBD do SBBD 2013 – Aceito 07/2013

Etapas Futuras:

- Implementação de um Gerador de *Stream* de Dados Sintéticos para Preferências 08/2013
- Submissão de Artigo para o ACM SAC 2014 09/2013
- Proposta de um Algoritmo Heurístico para Minerar Preferências em *Data Streams* 10/2013
- Submissão de Artigo para o FLAIRS 2014 11/2013
- Defesa da Dissertação 02/2014

Abstract. *The traditional preference mining setting has been widely studied in the literature in recent years. However, the problem of mining preferences increasingly requires solutions that quickly adapt to change. The main reason for this is that frequently user's preferences are not static and can evolve over time. In this work, we address the problem of mining contextual preferences in a data stream setting. Contextual Preferences have been recently treated in the literature and some methods for mining this special kind of preferences have been proposed in the traditional setting. As main contribution of this work, we formalize the contextual preference mining problem in the stream setting and propose appropriate algorithms for solving this problem.*

Keywords: bayesian networks, concept drift, context-awareness, data mining, data streams, preference mining

1. Introdução

O gigantesco aumento do volume de dados digitais presenciado nos últimos anos foi parcialmente ocasionado por uma nova classe de aplicações emergentes, em que os dados são gerados a taxas muito elevadas, na forma de *data streams*. Um *data stream* pode ser visto como uma sequência de tuplas relacionais que chegam continuamente em tempo variável. Alguns dos típicos domínios de aplicação de *streams* são: mercado financeiro, aplicações web, dados de sensores. Abordagens tradicionais não conseguem processar com êxito os *streams*, principalmente devido ao seu volume de dados potencialmente infinito e a sua evolução sobre o tempo. Com isso, várias técnicas de mineração de *streams* surgiram para lidar apropriadamente com este novo formato dos dados [Domingos and Hulten 2000, Bifet and Kirkby 2009, Gama 2010].

Apesar disso, a maioria das pesquisas de mineração de preferências têm se concentrado em cenários em que o algoritmo de mineração tem a sua disposição um conjunto de informações estáticas sobre as preferências do usuário [Jiang et al. 2008, de Amo et al. 2013]. As questões mais importantes que tornam o processo de mineração de *streams* muito mais desafiador do que no cenário tradicional, conhecido como *batch*, são: (1) Os dados não são armazenados e não estão disponíveis sempre; cada tupla deve ser aceita conforme ela chega e uma vez inspecionada ou ignorada, deve ser descartada; (2) o processo de mineração precisa lidar com limitações de memória e tempo; (3) o algoritmo de mineração deve ser capaz de produzir o melhor modelo em qualquer instante que é solicitado, usando apenas os dados de treinamento que foram observados até o momento. Mais detalhes são encontrados em [Rajaraman and Ullman 2011].

O aprendizado de preferências pode ser dividido em dois problemas distintos [Fürnkranz and Hüllermeier 2011]: *label ranking* e *object ranking*. No problema de *object ranking* as preferências de um usuário são obtidas a partir das características dos objetos, enquanto que no problema de *label ranking* as preferências de um conjunto de usuários são obtidas a partir das características dos usuários. Este trabalho atua sobre o problema de *object ranking*, ou seja, dado um usuário, estamos interessados em descobrir suas preferências a partir de suas escolhas.

Este trabalho foca em um tipo particular de preferências, as *preferências contextuais*. Modelos de preferências podem ser especificados sobre um framework *quantitativo* ou *qualitativo*. Na formulação quantitativa, preferências sobre filmes (por exemplo) podem ser elicitadas por pedir ao usuário para avaliar cada filme em uma base de dados. Na formulação qualitativa, o modelo de preferências consiste em um conjunto de regras especificadas em um dado formalismo matemático, capazes de expressar as preferências do usuário. Neste trabalho são consideradas as *regras de preferência contextuais* (*cp-rules*) introduzidas em [Wilson 2004]. Uma *cp-rules* permite informar preferências sobre os valores de um atributo dependendo dos valores de outros atributos. Por exemplo, um usuário pode preferir comédias a dramas *se o diretor for Woody Allen*.

1.1. Exemplo de Motivação

Considere que um site de notícias *online* deseja aprender as preferências dos usuários sobre notícias e, com base nisso, efetuar recomendações. Um cenário típico é aquele em que o usuário se loga no site e então se depara com várias manchetes de notícias. Para obter as preferências do usuário sobre notícias sem ser inoportuno, o sistema captura de

forma automática e *online* informações implícitas de preferências geradas através de ações do usuário. Uma das formas do sistema fazer isso é através da obtenção do *stream* de cliques e consultas do usuário. Normalmente, o usuário indica as notícias que tem maior interesse por efetuar cliques sobre a manchete das mesmas ou por buscar explicitamente por uma notícia. O sistema então converte este *stream* de ações do usuário em um *stream de preferências*, onde cada elemento do *stream* indica que uma notícia é preferível a outra em um dado instante de tempo. Através do acesso ao *stream de preferências* do usuário, o sistema consegue realizar a mineração de suas preferências ao longo do tempo.

1.2. Objetivos

O objetivo principal deste trabalho é formalizar o problema de mineração de preferências contextuais no cenário de *streams* e propor algoritmos eficientes para solucionar este problema. Para alcançar esse objetivo, os objetivos específicos propostos são: (1) Introduzir o conceito de *stream de preferências* como uma forma de representação das preferências do usuário ao longo do tempo; (2) Propor um método qualitativo para a elicitação *online* das preferências do usuário sob a forma de um *stream de preferências*; (3) Introduzir o conceito de *concept drift* para o ambiente de preferências; (4) Propor 4 algoritmos para solucionar o problema abordado, cada um com um enfoque diferente; (5) Avaliar os algoritmos propostos segundo critérios de qualidade a serem adaptados para *streams*; (6) Implementar um gerador de *stream* sintético de preferências para a realização de testes nos algoritmos propostos; (7) Avaliar os algoritmos propostos sobre dados reais.

1.3. Principais Contribuições

As principais contribuições desta pesquisa para a área de mineração de preferências são: (1) Identificação e formalização sistemática de um novo problema de mineração de preferências: "A partir do *stream de preferências* do usuário, é possível realizar o aprendizado sobre as suas preferências contextuais de forma eficiente em termos de tempo e memória?"; (2) Proposta de 4 algoritmos de mineração de preferências do usuário em *data streams*; (3) Introdução do conceito de *concept drift* para o cenário de preferências; (4) Desenvolvimento de um gerador de *stream* de dados sintéticos para preferências.

2. Problema de Pesquisa

O objetivo de um método de mineração de preferências é prover uma relação de preferência sobre certo conjunto de dados. Uma relação de preferência sobre um conjunto finito de objetos $A = \{a_1, a_2, ..., a_n\}$ é uma ordem parcial estrita sobre A, que é uma relação binária $R \subseteq A \times A$ satisfazendo as propriedades irreflexiva e transitiva.

Os dados de entrada do método de mineração de preferências são representados por um *stream de preferências*, que é potencialmente infinito. Um *stream* de preferências é formado por elementos do tipo (u, v, t), que chamaremos de bitupla temporal, representando o fato de que o usuário prefere a tupla u a tupla v no instante de tempo t. Fig. 1(b) ilustra parte de um *stream* de preferências sobre o esquema relacional $R(A, B, C, D)$, representando uma amostra de suas preferências em relação às tuplas de I (Fig. 1(a)) até o instante de tempo t_7, coletada do *stream* de ações do usuário em um site.

Esse problema consiste em extrair um modelo de preferência de um *stream* de preferências em um dado instante de tempo. O modelo de preferência será especificado por uma *Rede Bayesiana de Preferência* (RBP).

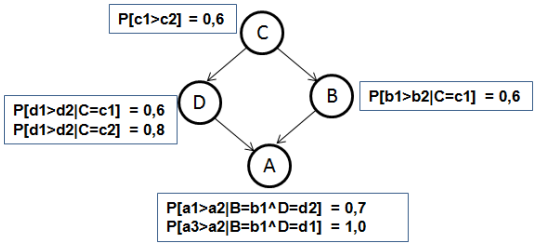
Uma RBP sobre um esquema relacional $R(A_1, ..., A_n)$ em um dado instante de tempo é um par (G, θ), onde: (1) G é um grafo direcionado acíclico, cujos nós são atributos em $\{A_1, ..., A_n\}$ e as arestas representam a dependência entre os atributos; (2) θ é um mapeamento que associa para cada nó de G um conjunto finito de regras de probabilidades condicionais. Fig. 1(c) ilustra uma $RBP\ PNet$ sobre R. Note que as preferências sobre os valores do atributo B dependem dos valores do contexto C: se $C = c1$, a probabilidade do valor $b1$ ser preferido ao valor $b2$ para o atributo B é 60%.

Id	A	B	C	D
u_1	a_1	b_1	c_1	d_1
u_2	a_1	b_1	c_1	d_2
u_3	a_2	b_1	c_1	d_2
u_4	a_1	b_2	c_1	d_2
u_5	a_2	b_1	c_2	d_1
u_6	a_3	b_1	c_1	d_1

(a)

(u_1,u_2, t_1)
(u_1,u_3, t_2)
(u_4,u_5, t_3)
(u_4,u_2, t_4)
(u_5,u_6, t_5)
(u_3,u_5, t_6)
(u_4,u_1, t_7)

(b)



(c)

Figura 1. (a) Uma instância I, (b) Um *stream* de preferências S, (c) Rede de Preferências PNet no instante t

Em nosso cenário, uma RBP é usada para comparar pares de tuplas. A qualidade de uma RBP como uma ferramenta de ordenação é mensurada por meio de sua precisão e *recall*.

Dessa forma, o problema de Mineração de Preferências Contextuais do Usuário em *Data Streams* é dado por:
Entrada: um esquema relacional R e um *stream* de preferências sobre R.
Saída: sempre que requisitado, retorne uma RBP sobre R no instante de tempo da solicitação, tendo boa precisão e *recall*.

3. Andamento do Trabalho

3.1. Estado Atual

No início deste trabalho foi identificado e formalizado o problema de mineração de preferências contextuais em *data streams*. Em seguida, alguns métodos para resolver esse problema foram propostos. Todos os métodos de mineração de preferências propostos trabalham apenas com um conjunto reduzido de estatísticas suficientes coletadas a partir do *stream* de preferências do usuário recebido como entrada. Estas estatísticas contêm o mínimo de informações necessárias para representar o *stream* de preferências, e proporcionam tanto uma maior velocidade na mineração quanto uma economia de memória.

Primeiramente, foram propostas duas estratégias para resolver este problema: a estratégia Gulosa e a Heurística. A principal ideia da estratégia Gulosa é criar uma relação de preferências a partir das mais promissoras dependências entre os atributos do *stream* de preferências. Já a estratégia Heurística é baseada em um Algoritmo Genético Incremental sobre um *stream* de preferências para executar a fase de aprendizado da estrutura de grafo da rede de preferências. Apenas a estratégia gulosa foi implementada, e testes preliminares sobre a mesma foram realizados, mostrando que ela é factível.

A estratégia Gulosa foi melhor trabalhada e deu origem ao algoritmo FPSMining. Mais detalhes sobre este algoritmo são apresentados nas próximas subseções.

Após isso, foi proposto o algoritmo IncFPSMining. Esse algoritmo constrói o modelo de preferências incrementalmente. A cada k bituplas temporais (parâmetro do algoritmo) que chegam no *stream* de preferências, o modelo corrente é atualizado. Essa atualização consiste em incrementar o modelo atual agregando novas arestas ao seu grafo. Utiliza-se a teoria do limite de Hoeffding para garantir que arestas inseridas anteriormente não percam sua validade ao longo do tempo. Foram realizados testes experimentais neste algoritmo, comprovando sua eficiência sobre dados reais.

3.1.1. O Algoritmo FPSMining

A principal ideia do FPSMining é criar uma relação de preferências a partir das mais promissoras dependências entre os atributos de um *stream* de preferências. Para medir a dependência de um par de atributos é usado o conceito de *grau de dependência*. O grau de dependência de um par de atributos (X, Y) com relação a um *snapshot* Q das estatísticas do *stream* de preferências S no instante t é um número real que estima como as preferências dos valores do atributo Y são influenciadas por valores do atributo X.

Dado isso, este algoritmo constrói uma RBP a partir das estatísticas como se segue: (1) Tire um snapshot Q das estatísticas de S no instante t; (2) Calcule o grau de dependência dd entre cada possível par de atributos de R de acordo com Q. Seja Ω o conjunto resultante desses cálculos, com elementos do tipo (A_i, A_j, dd), onde A_i e A_j são atributos de R; (3) Elimine de Ω todos os elementos cujo $dd < 0.5$ (indica fraca dependência); (4) Ordene os elementos (A_i, A_j, dd) em Ω de forma decrescente de acordo com o seu dd; (5) Considere que o grafo G da RBP começa com um nó para cada atributo de R. Dentro de um laço, pegue cada $(A_i, A_j, dd) \in \Omega$ e insira a aresta (A_i, A_j) no grafo G apenas se a inserção não formar ciclo em G. Assim, o grafo G da RBP é criado; (6) Uma vez que se tem G, estime as tabelas de probabilidade condicional θ da RBP usando o Princípio da Probabilidade Máxima (veja [de Amo et al. 2013] para detalhes) sobre Q.

3.1.2. Resultados Experimentais do FPSMining

Os dados reais usados nos testes do FPSMining contêm preferências relacionadas a filmes coletadas pelo *GroupLens Research* [1] a partir do *MovieLens web site* [2]. Foi simulado um *stream* de preferências a partir destes dados, como se segue: foi estipulado um intervalo de tempo λ (medido em dias e horas), e cada tupla no conjunto de dados foi comparada a todas as outras em um raio λ referente ao seu *timestamp*, gerando assim as bituplas temporais. O *stream* de preferências resultante possui cinco atributos (diretor, gênero, idioma, ator e ano), e seus elementos correspondem a preferências sobre filmes determinadas por um dado usuário. Como técnica de amostragem foi utilizado o *holdout* para *data stream*.

Fig. 2(a),(b) apresentam os resultados para a variação dos parâmetros *usuário* e λ, respectivamente. Os valores de *recall* e precisão apresentados são os valores médios de todas as medições efetuadas no *holdout*. Fig. 2(a) mostra que quanto mais filmes o usuário avaliou (tuplas), melhor é o *recall* e a precisão do FPSMining com relação a

[1] Disponível em `http://www.grouplens.org/taxonomy/term/14`
[2] Disponível em `http://movielens.umn.edu`

suas preferências. Fig. 2(b) mostra que até λ=1 dia os valores de qualidade aumentam conforme o número de elementos no *stream* de preferências aumenta. Na Fig. 2(a) foi utilizado λ=1 dia e na Fig. 2(b) foi utilizado *usuário*=$U1$.

Nos testes, os valores de *recall* e precisão foram satisfatórios, mostrando que o FPSMining é eficiente para minerar as preferências do usuário em *streams*. O tempo gasto na geração do modelo foi de 16ms e o tempo para completar um ciclo do *holdout* foi de 31ms, mostrando que o algoritmo atende as restrições de tempo exigidas em *streams*.

Usuário	Tuplas	Bituplas	*Recall*	Precisão
$U1$	7359	1580710	0.871	0.874
$U2$	6047	1658450	0.794	0.801
$U3$	4483	563419	0.778	0.784
$U4$	4449	198618	0.769	0.785

(a)

Stream	λ	Bituplas	*Recall*	Precisão
$S1$	1h	227100	0.800	0.814
$S2$	12h	645319	0.846	0.853
$S3$	1d	1048228	0.879	0.882
$S4$	7d	1580710	0.871	0.874
$S5$	15d	1759753	0.875	0.876

(b)

Figura 2. Conjunto de Testes Experimentais

3.2. Desenvolvimento Necessário para a Conclusão

De imediato, planeja-se elaborar uma comparação experimental entre os algoritmos FPSMining e IncFPSMining. Sobre as propostas de algoritmos, pretende-se ainda: 1) Implementar a estratégia Heurística, pois é esperado que ela apresente resultados bastante promissores, devido a sua capacidade potencial de poder atingir qualquer ponto do espaço de busca; 2) Implementar um algoritmo de *ensemble*, usando como algoritmo de aprendizado base os algoritmos propostos. Pretende-se também implementar um gerador de *stream* sintético de preferências (com introdução de *concept drift*), para possibilitar testes com uma quantidade de dados tão grande quanto se desejar. Finalmente, planeja-se avaliar o comportamento dos algoritmos propostos frente a *concept drifts*.

4. Trabalhos Relacionados

Com o intuito de prover uma análise comparativa, a Tabela 1 apresenta uma síntese sobre artigos importantes para a área de mineração de preferências. Segue o significado das siglas utilizadas na Tabela 1: LR/OR (*Label Ranking*/*Object Ranking*) – informa o problema de mineração de preferências; QT/QL (Quantitativo/Qualitativo) – informa se a elicitação de preferências é formalizada sobre um *framework* quantitativo ou qualitativo; U/A (Usuário/Automática) – informa se durante a elicitação de preferências foi solicitado ao usuário informações sobre suas preferências, ou se isso foi extraído de maneira automática; B/I (*Batch*/Incremental) – informa se o artigo aborda a mineração *batch* ou incremental.

Propostas de algoritmos adequados para resolver o problema de mineração de preferências do usuário em *streams* têm sido pouco explorados na literatura. Por exemplo, [Jembere et al. 2007] apresenta uma abordagem para minerar as preferências do usuário em um ambiente com vários serviços cientes de contexto, mas usa aprendizagem incremental somente para o contexto, e não para as preferências do usuário. Enquanto isso, [Somefun and La Poutré 2007] apresenta um método *online* que visa utilizar o conhecimento agregado sobre as preferências de muitos clientes para fazer recomendações a clientes individuais. Nenhum deles aborda especificamente o problema que o nosso trabalho aborda.

Tabela 1. Síntese de Artigos de Mineração de Preferências

Artigo	LR/ OR	QT/ QL	Modelo de Preferência	U/ A	Entrada	Saída	B/ I
[de Amo et al. 2013]	OR	QL	Contextual	U	BD de Preferências	RBP	B
[Jiang et al. 2008]	OR	QL	Pareto	U	$\{Superiores\}$, $\{Inferiores\}$	SPS	B
[Beretta et al. 2011]	OR	QL	Contextual	A	Log de Consultas	$\{\alpha - preferences\}$	I
[Jembere et al. 2007]	LR	QT	Contextual	A	Log de Dados	Sistema de Recomendação	I
[Somefun and La Poutré 2007]	LR	QL	Pareto	A	Ofertas sobre Pacotes	Mecanismo Aconselha.	I
Nossa Proposta	OR	QL	Contextual	A	*Stream* de Preferências	RBP_t	I

Referências

Beretta, D., Quintarelli, E., and Rabosio, E. (2011). Mining context-aware preferences on relational and sensor data. *International Workshop on Database and Expert Systems Applications*, pages 116–120.

Bifet, A. and Kirkby, R. (2009). Data stream mining: a practical approach. Technical report, The University of Waikato.

de Amo, S., Bueno, M. L. P., Alves, G., and da Silva, N. F. F. (2013). Mining user contextual preferences. *Journal of Information and Data Management - JIDM*, 4(1):37–46.

Domingos, P. and Hulten, G. (2000). Mining high-speed data streams. In *International Conference on Knowledge Discovery and Data Mining*, pages 71–80.

Fürnkranz, J. and Hüllermeier, E. (2011). Preference learning. Springer.

Gama, J. (2010). *Knowledge Discovery from Data Streams*. Chapman & Hall/CRC.

Jembere, E., Adigun, M. O., and Xulu, S. S. (2007). Mining context-based user preferences for m-services applications. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pages 757–763.

Jiang, B., Pei, J., Lin, X., Cheung, D. W., and Han, J. (2008). Mining preferences from superior and inferior examples. In *KDD*, pages 390–398. ACM.

Rajaraman, A. and Ullman, J. D. (2011). *Mining of Massive Datasets*. Cambridge University Press.

Somefun, D. J. A. and La Poutré, J. A. (2007). A fast method for learning non-linear preferences online using anonymous negotiation data. In *Proceedings of the AAMAS workshop and TADA/AMEC conference*, pages 118–131.

Wilson, N. (2004). Extending cp-nets with stronger conditional preference statements. In *Proceedings of the 19th national conference on Artifical intelligence*, pages 735–741.