Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Ciência da Computação

Recomendação Multi-Contextual de Eventos em Redes Sociais de Eventos

Augusto Queiroz de Macedo

Dissertação submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande - Campus I como parte dos requisitos necessários para obtenção do grau de Mestre em Ciência da Computação.

Área de Concentração: Ciência da Computação
Linha de Pesquisa: Mineração de Dados

Leandro Balby Marinho
(Orientador)

Campina Grande, Paraíba, Brasil

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

M141r Macedo, Augusto Queiroz de.
Recomendação multi-contextual de eventos em redes sociais de eventos / Augusto Queiroz de Macedo. – Campina Grande, 2015.
90 f. : il. color.

Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2015.

"Orientação: Prof. Dr. Leandro Balby Marinho".
Referências.

1. Redes Sociais. 2. Mineração de Dados. 3. Sistemas de Recomendação Multi-Contextuais. 4. Recomendação de Eventos. I. Marinho, Leandro Balby. II. Título.

CDU 004.771(043)

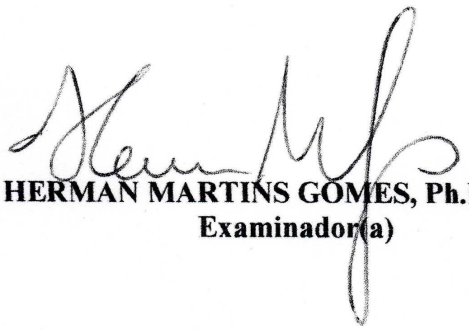
'RECOMENDAÇÃO MULTI-CONTEXTUAL DE EVENTOS EM REDES SOCIAIS DE EVENTOS"

AUGUSTO QUEIROZ DE MACEDO

DISSERTAÇÃO APROVADA EM 27/03/2015

LEANDRO BALBY MARINHO, Dr., UFCG
Orientador(a)

NAZARENO FERREIRA DE ANDRADE, D.Sc, UFCG
Examinador(a)



HERMAN MARTINS GOMES, Ph.D, UFCG
Examinador(a)

WAGNER MEIRA JUNIOR, Dr., UFMG
Examinador(a)

CAMPINA GRANDE - PB

Resumo

A Web tem crescido tornando-se um dos mais importantes canais para comunicar eventos sociais hoje em dia. As pessoas planejam, compartilham e comentam sobre os eventos na Web. As redes sociais baseadas em eventos (RSBEs) foram criadas para ajudar as pessoas a encontrar e conhecerem-se uns aos outros de uma forma mais simples e rápida. No entanto, o grande volume de eventos disponíveis, muitas vezes prejudica a capacidade dos usuários de escolher os eventos que melhor se adequam à suas preferências pessoais. Sistemas de recomendação aparecem como uma solução natural para este problema.

No entanto, diferentemente dos cenários de recomendação clássica (e.g. recomendação de filmes, livros, restaurantes), o problema de recomendação de eventos é intrinsecamente *cold-start* (início frio) quando inexistem informações sobre as interações de usuários e itens no momento da recomendação. A princípio, esta interação só acontece após a ocorrência do evento. E mesmo usando informações de RSVPs (i.e. intenção declarada do usuário em comparecer à um evento futuro), o recomendador continua enfrentando alta esparsidade de dados, que é agravada pela tendência que os usuários possuem em enviar RSVPs próximos à ocorrência dos eventos.

Para superar essas limitações, propomos um modelo de recomendação híbrido que otimiza o ranking personalizado dos eventos baseado no potencial de vários sinais contextuais disponíveis nas RSBEs. Além de sinais sociais derivados dos RSVPs e das associações dos usuários em grupos online, exploramos também os sinais de conteúdo das descrições dos eventos, sinais de localização baseados nas coordenadas geográficas da casa dos usuários e dos eventos e sinais temporais derivados das preferências de horário e dias da semana do usuário em relação aos seus eventos passados.

Por meio de experimentos utilizando uma grande coleta do Meetup.com melhoramos em mais de 60% a métrica de ranking personalizado avaliada com a nossa abordagem híbrida de aprendizagem multi-contextual em comparação com um recomendador de eventos do estado-da-arte da literatura.

Palavras-chave: Mineração de Dados, Sistemas de Recomendação Multi-Contextuais, Redes Sociais, Recomendação de Eventos

Abstract

The Web has grown into one of the most important channels to communicate social events nowadays. People plan, share and comment meetings through the Web. The event-based social networks (EBSNs) have been created to help people meet and know each others in a simpler and faster way. However, the sheer volume of events available often undermines the users' ability to choose the events that best fit their personal preferences. Recommender systems appear as a natural solution for this problem.

However, differently from the classic recommendation scenarios (e.g. movies, books, restaurants recommendations), the event recommendation problem is intrinsically cold-start, there is no information about the users and items interactions in recommendation time. At first, this interaction only happens after the ocurrence of the event. And even using the RSVPs informations (i.e. declared user intention to attend or not a future event), the recommender will still have to face its high sparsity worsened by the trend that users have to send RSVPs near the ocurrence of the events.

To overcome this limitation, we propose a hybrid recommendation model that optimizes a personalized ranking of events based on the several contextual signals available in EBSNs. Besides social signals derived from RSVPs and user's associations in online groups, we exploit the content signals from events' description, location signals based on the users' home and events geographic coordinates and temporal signals derived from the temporal and weekday users' preferences related to their past events.

Thorough experiments using a large crawl of Meetup.com we improved in more than 60% the evaluated personalized ranking metric with our multi-contextual learning approach when compared to a state-of-the-art event recommender from the literature.

Keywords: Data Mining, Multi-Contextual Recommender Systems, Social Networks, Event Recommendation

Agradecimentos

Primeiramente, eu agradeço a Deus, nosso Pai, que a tudo criou e para o qual busco caminhar diariamente e Jesus o seu filho, guia de toda a nossa humanidade.

Agradeço também a minha família, no nome de minha mãe, Miriam Queiroz de Macedo, que soube sozinha vencer a batalha da vida. Vitoriosa na missão de educar a mim e a meus irmãos, Alexande e Jaqueline, mesmo diante da viuvez prematura, sabendo reunir e fortalecer amizades mil que não a deixam esmorecer. Muito obrigado mãe, por ter nos ensinado a sorrir sem reservas, a abraçar a pobreza desfavorecida sem voz nem vez e a amar os estudos, "única forma de vencer na vida".

Lado a lado com ela não posso deixar de me referir às demais mães que tivemos, que souberam trabalhar dignamente em nosso lar mantendo um vínculo de amizade e amor com a nossa família de várias décadas, nomeio aqui algumas delas, talvez as principais para mim, Marluce dos Santos, Dailza de Lima e Adalgiza de Araújo.

Agradeço fortemente a Leandro Balby Marinho, que durante esses anos soube compreender minhas dificuldades, aproveitar e valorizar minhas capacidades e me forçar a trabalhar mais e melhor. Lembro também dos demais professores da graduação em Ciência da Computação na UFCG, que juntos foram como pontes de amor, verdadeiros intérpretes dos feitos computacionais humanos para a mente de um leigo.

Agradeço profundamente à minha futura esposa que exercitou o amor mais puro, o amor renúncia, sabendo silenciar as suas necessidades enquanto andava ao meu lado, na busca do meu sucesso nos estudos.

E por fim, aos amigos (e inimigos) encarnados e desencarnados, em especial aos verdadeiros amigos de ideal espirita. Citando dois destes que são e serão sempre verdadeiros pais para mim, que me educam até hoje na ciência do bem viver, com sábias lições de pessoas já experimentadas pela vida. Muito obrigado Sr. Ediberto e Dona Edith Paes Barreto!

Muito Obrigado!

Conteúdo

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

1.1 Motivação

O rápido crescimento e a variedade de informações disponíveis na Web levou ao desenvolvimento dos estudos em Recuperação de Informação que vem emprestando suas técnicas, previamente estudadas no contexto de Sistemas de Informação, para a busca automática de documentos na Web. Com o crescimento da Web despontaram os sistemas de *e-commerce* que se utilizam das facilidades da rede para aumentar a quantidade de opções de compra do usuário e alavancar suas vendas, como também variados tipos de mídia social. A Figura 1.1 chamada do prisma da conversação foi desenvolvida para auxiliar as empresas que desejam expandir seus negócios na mídia social da Web devido a grande quantidade de serviços existentes. Cada serviço possui objetivos específicos como vendas, promoção de marcas, promoção do conhecimento, sendo a maioria mantida pela própria comunidade de usuários, o que formam as conhecidas redes sociais. O ambiente Web dessas redes sociais tem se mostrado muito fecundo na produção de conteúdo pelos usuários o que tem causado um problema conhecido como sobrecarga de informações, isto é, os usuários perdem a capacidade de encontrar conteúdos de interesse (RICCI et al., 2011).

Em meados da década de 90, os sistemas de recomendação (SR) emergem como uma alternativa aos estudos clássicos em sistemas de informação, tendo como principal objetivo a diminuição do espaço de busca do usuário, auxiliando-o assim na tomada de decisão. Desde lá, as pesquisas na área tem buscado desenvolver ferramentas de software e técnicas com o objetivo de sugerir, automaticamente, itens relevantes para o usuário de forma a maximizar

Figura 1.1: Prisma da Conversação (SOLLIS; JESS3, 2013)

a chance do mesmo consumi-lo (RICCI; ROKACH; SHAPIRA, 2011).

Os SRs tem ganhado tal espaço na Web que tanto os provedores de serviço online quanto os usuários tem suas razões para desejarem implementar/receber recomendações. Os principais motivos para a implementação de SR pelos provedores de serviço são: aumentar o número ou a diversidade dos itens vendidos, aumentar a satisfação, fidelidade, e melhor entender os anseios do usuário. Sob o ponto de vista do usuário existem vários motivos que podem levar à implementação de SRs, como por exemplo: facilitar o encontro com novos itens similares aos já consumidos, gerar sequências de itens bem relacionados (e.g. sequências de músicas ou de fotos), auxiliar o usuário a definir suas preferências expondo a recomenda-

ções bem diversificadas. Devido ao grande sucesso alcançado em sistemas de *e-commerce*, a demanda por sistemas de recomendação tem aumentado e adentrado diversos outros domínios de aplicação, tais como rádios *online* (e.g. recomendação de músicas), agências de turismo (e.g. recomendação de pontos turísticos), softwares embarcados em GPSs (e.g. recomendação de pontos de interesse e rotas turísticas) e as redes sociais baseadas em eventos (RSBE) (LIU et al., 2012), domínio de aplicação deste trabalho.

Atualmente, vários sítios provedores de serviço tem aparecido com o objetivo de promover eventos sociais, tendo como principais funcionalidades a reunião das pessoas em grupos online nos quais elas podem criar, divulgar e comentar eventos, como também podem seguir listas de eventos de outros (e.g. de celebridades, cantores, políticos). São alguns exemplos o Meetup[1], Last.fm[2], Facebook[3], Plancast[4] e Eventful[5]. Com essa estrutura eles promovem ao mesmo tempo interações virtuais e reais entre seus usuários, as quais podem variar desde sessões de cinema com a família até concertos e shows com milhares de pessoas (e.g. o Last.fm promove eventos musicais e o Meetup permite qualquer tipo de evento que possa ser criado por um grupo de pessoas online).

Nesse domínio, o problema dos usuários recai sobre a seleção dos próximos eventos a participar, já que as RSBE possuem milhares de eventos disponíveis a todo instante. Sistemas de recomendação tem resolvido esse tipo de problema com sucesso em diversos outros domínios (CHENG et al., 2012; MARINHO et al., 2012; RICCI et al., 2011) e portanto é a abordagem escolhida no presente trabalho.

A pesquisa em sistemas de recomendação de eventos em RSBEs ainda é recente, os primeiros trabalhos apareceram nos últimos 3 anos, iniciando-se com Liu et al. (2012). Um dos principais problemas técnicos enfrentados pela academia, além da seleção de eventos relevantes, tem sido o curto tempo de vida dos eventos. O tempo entre a criação e a execução dos eventos é um forte limitador para as recomendações, já que antes ou depois desse intervalo o evento não existe ou não faz sentido ser recomendado. Assim, os eventos sofrem fortemente os efeitos do *cold-start* (início frio), ou seja, durante uma parte considerável do período entre a criação e a ocorrência do evento, não há dados sobre quais usuários estão interessados em

[1] <www.meetup.com>
[2] <www.last.fm/events>
[3] <www.facebook.com>
[4] <www.plancast.com>
[5] <www.eventful.com>

participar desse evento. Esse tipo de dado é essencial para o funcionamento eficaz da maioria dos algoritmos de recomendação estado-da-arte existentes (RENDLE et al., 2009; NING; KARYPIS, 2011). Esta característica é o que mais diferencia a recomendação de eventos de outros domínios de recomendação como filmes, livros, hotéis, restaurantes que tem um longo tempo de vida e podem ser recomendados a qualquer hora, até repetidas vezes.

Por outro lado, as RSBEs estão repletas de dados com informações contextuais relacionadas aos eventos e aos usuários (descrição textual, dia e horário de ocorrência dos eventos, localização geográfica de usuários e eventos, histórico de co-participação de usuários em eventos e co-filiação nos grupos online) que podem ser utilizados para mitigar a falta de informação do *cold-start* do evento e também do usuário, quando o usuário não manifestou interesse por evento algum (cenário comum à maioria dos sistemas de recomendação), e ainda gerar recomendações acuradas.

Propomos então uma abordagem multi-contextual (ADOMAVICIUS; TUZHILIN, 2011) para a recomendação de eventos, que explora os múltiplos sinais contextuais disponíveis em RSBEs. Além do sinal presente nas interações sociais em grupos online e nos eventos através das informações de RSVP[6], o conteúdo textual presente na descrição dos eventos, o histórico de localizações, datas e horas dos eventos comparecidos pelos usuários, como também a localização de sua casa. Em particular, nossa hipótese é que cada um desses sinais tem uma influência positiva na decisão do usuário em participar de um evento. Por exemplo, um usuário pode decidir ir a um show musical não somente porque ele gosta da banda, adequando-se às suas preferências pessoais, mas porque ele vai reencontrar amigos e o evento ocorrerá no seu horário de folga do trabalho.

Para explorar os contextos mencionados anteriormente, analisamos o comportamento de modelos de recomendação especializados para cada contexto. Combinando então, suas capacidades preditivas em um recomendador multi-contextual híbrido que aprende a ranquear os eventos de forma personalizada em uma RSBE.

[6]RSVP é uma expressão francesa “répondez s’il vous plaît”, que significa “por favor responda”.

1.2 Definição do Problema

Os sistemas de recomendação tradicionais tipicamente modelam interações entre dois tipos de entidade, usuários e itens. Recomendadores multi-contextuais, por outro lado, exploram sinais adicionais que ajudam a moderar essa interação, incluindo localização, tempo e o ambiente social no qual a interação acontece (ADOMAVICIUS; TUZHILIN, 2011). Quando os itens a recomendar são eventos futuros em fase de divulgação, a interação usuário-evento inexiste, sendo consolidada apenas após a sua ocorrência. Mesmo quando simulamos a interação real usuário-evento por meio dos CapítuloPs positivos – como fazemos depois no Capítulo 3 – essa informação é escassa para a maioria dos eventos e dos usuários. Para solucionar esse problema utilizamos a rica fonte de informações contextuais disponíveis nas RSBEs.

O problema que buscamos solucionar nesta dissertação pode ser especificado como segue: dado um usuário alvo e um conjunto de sinais contextuais, quais eventos candidatos tem maior probabilidade do usuário participar (i.e. receber um RSVP positivo)? Formalizando, além do conjunto de usuários U e do conjunto de eventos E, consideramos os sinais contextuais do conjunto de grupos G que os usuários podem se afiliar, das preferências temporais T dos usuários e do conjunto de pares de localização $C \subseteq L \times L$, onde o primeiro elemento representa a localização da casa do usuário e o segundo a localização do evento candidato, ambos codificados como as coordenadas geográficas (no caso, latitude e longitude). Neste trabalho consideramos apenas dados implícitos, i.e., o conjunto $S \subseteq U \times E \times G \times C \times T$ das relações entre usuários, eventos, grupos, tempo, e localizações. A tarefa do recomendador de eventos é então encontrar uma função

$$\hat{s} : U \times E \times G \times C \times T \to \mathbb{R} \tag{1.1}$$

que atribua um valor de preferência para os eventos candidatos. Assim, dado um usuário alvo $u \in U$ e os sinais contextuais $g \in G$, $c \in C$, e $t \in T$, as top-n recomendações podem ser computadas por

$$\textit{top-}n(u,g,c,t) := \underset{e \in E_c \setminus E_u}{\overset{n}{\operatorname{argmax}}} \hat{s}(u,e,g,c,t) \tag{1.2}$$

onde n denota o número de eventos a serem recomendados, E_c os eventos candidatos e E_u os eventos que o usuário u enviou RSVPs positivos no passado. Em cenários de recomendação top-n, como esse, o objetivo é minimizar uma função de perda de ranking da forma

$$\ell : \mathcal{P}(E) \times \mathbb{R}^E \rightarrow \mathbb{R}$$

que quantifica a diferença entre a lista de recomendação (gerada por $\hat{s} \in \mathbb{R}^E$) e a lista real (um sub-conjunto de $\mathcal{P}(E)$) nos usuários de teste cujos eventos reais (i.e. gabarito) são desconhecidos durante o treino. A função de perda específica utilizada neste trabalho é descrita no Capítulo 3.

1.3 Contribuições

As principais contribuições deste trabalho estão sumarizadas a seguir:

- **Análise contextual das RSBEs**: Realizamos uma análise meticulosa e abrangente da RSBE como também dos contextos para compreensão do seu potencial de predição na tarefa de recomendação de eventos;
- **Recomendadores contextuais especializados**: Propomos recomendadores personalizados especializados nos contextos social, geográfico, temporal e de conteúdo textual dos eventos em RSBEs por meio de novos algoritmos de recomendação ou aplicando algoritmos já propostos pela literatura;
- **Novo recomendador contextual híbrido**: Propomos um novo recomendador híbrido que aprende a ranquear os eventos candidatos agregando os múltiplos sinais contextuais, solucionando assim o problema do *cold-start* e adequando-se às variações temporais e culturais das RSBEs;
- **Avaliação dos recomendadores propostos**: Avaliamos os recomendadores propostos com dados do Meetup.com em cenários temporalmente particionados, demonstrando que o modelo híbrido melhora o ranking dos eventos recomendados em mais de 60% em relação aos modelos estado-da-arte.

Uma parte dessas contribuições foi publicada no artigo **Event Recommendation in Event-based Social Networks** apresentado no Workshop Social Personalization (*International Workshop on Social Personalisation*) do ano de 2014.

1.4 Estrutura da Dissertação

Os demais capítulos desta dissertação estão organizados como segue. O capítulo 2 lista os trabalhos da literatura mais relevantes relacionados com o problema de recomendação de eventos em RSBEs. Descrevemos a metodologia de coleta dos dados do Meetup.com, a metodologia experimental e de avaliação dos modelos incluindo a forma de particionamento dos dados, e uma análise aprofundada dos RSVPs no Capítulo 3. Os próximos três capítulos relatam em detalhes os contextos Social (Capítulo 4), de Conteúdo textual dos eventos (Capítulo 5), Geográfico e Temporal (Capítulo 6). Cada contexto é analisado em busca de *insights* e com os resultados propomos modelos de recomendação de eventos especializados. No Capítulo 7 estes modelos são combinados em um modelo Híbrido que aprende a ranquear os eventos personalizadamente, o qual é comparado com um método do estado-da-arte da literatura. E o Capítulo 8 conclui o presente trabalho.

Capítulo 2

Trabalhos Relacionados

Neste capítulo descrevemos os trabalhos relacionados com a recomendação de eventos abrangendo vários pontos de vista e abordagens para o problema. Os artigos aqui listados foram encontrados nas principais conferências da área de sistemas de recomendação (e.g. AAAI, KDD, SIGIR, RecSys) e com buscas por citação e referências, selecionando outros artigos correlatos que citaram ou foram citados por aqueles artigos. Em um segundo momento realizamos uma busca no *Google Scholar* pela string "event recommendation" desde o ano de 2011. Como resultado obtivemos mais de 440 artigos advindos dos mais variados repositórios todos com alguma referência a esse termo. Fizemos uma filtragem pelo título que resultou em 23 artigos não encontrados anteriormente, dentre os quais 8 abordam exatamente o tema deste trabalho, 11 possuem temas correlatos como recomendação de grupos em RSBEs (não listados aqui) e os 5 demais não obtivemos acesso para leitura.

Liu et al. (2012) definem e analisam as RSBEs. Realizam uma coleta de dados reais da RSBE Meetup e listam as propriedades desse tipo de rede, como por exemplo: a existência de grandes eventos (i.e. com muitos participantes) em quantidade considerável, a forte dependência geográfica das interações sociais, ou seja, usuários tendem a ir a eventos próximos às suas casas, e a possibilidade de interações *online* e *offline*, ou seja, interações pela Web e presenciais, respectivamente. Essas características diferenciam as RSBEs das Redes Sociais Baseadas em Localização (RSBLs), que se baseiam em dados de *checkin* dos usuários, como o Foursquare[1] e a Jiepang[2], as quais tendem a ter poucos locais com muitos *checkins*, não

[1]<https://pt.foursquare.com/>
[2]<http://jiepang.com/>

se restringindo a posições geográficas próximas a casa do usuário, acontecendo em quantidade considerável em cidades e até países diferentes, e principalmente, por estes locais não possuírem um prazo de validade como os eventos.

No contexto de recomendação, os autores propuseram também um modelo de difusão para recomendação de usuários em RSBE baseado em comunidades. Uma comunidade é formada a partir da co-participação em eventos e/ou grupos. O modelo agrupa os usuários em comunidades e dá maior probabilidade de difusão para usuários da mesma comunidade. Na avaliação, a base de dados é dividida em um único momento no tempo e o modelo proposto tem um *recall* ligeiramente superior à filtragem colaborativa baseada nos k-vizinhos mais próximos do usuário e ao modelo de difusão do caminho aleatório.

Dando prosseguimento às pesquisas de Liu et al. (2012), Qiao et al. (2014b) propõem um método de recomendação de eventos. Esse método utiliza interações sociais, tanto *online* quanto *offline*, como uma forma de regularização social em um modelo de fatoração de matrizes estado-da-arte para a aprendizagem de ranqueamentos personalizados de itens.

Os mesmos autores propõem também um segundo modelo (QIAO et al., 2014a) sob a forma de uma combinação linear ponderada de dois componentes: um componente social baseado no modelo apresentado em (QIAO et al., 2014b) e um componente geográfico que relaciona os usuários às regiões que englobam os eventos que eles compareceram no passado. Essas regiões foram definidas de forma não supervisionada pelo agrupamento das coordenadas geográficas dos eventos utilizando o algoritmo clássico de agrupamento KMeans. A mesma metodologia de avaliação é compartilhada por ambos os artigos (QIAO et al., 2014b, 2014a), i.e. mesma base de dados, mesmas cidades sobre as quais treinam e testam os modelos, e mesma métrica de avaliação. Ao final, o segundo modelo, teoricamente mais promissor, tem resultados inferiores ao primeiro em todas as cidades.

Sendo o *cold-start* do evento o principal problema do domínio de RSBEs, os trabalhos da literatura são unânimes ao buscarem solucioná-lo adicionando informações contextuais como dados geográficos, textuais e até com o enriquecimento via bases externas de informação, como DBPedia[3].

Pascoal et al. (2014) soluciona o problema do *cold-start* do usuário e do evento para recomendação de eventos no Facebook. O trabalho propõe uma nova função de similaridade

[3]<http://dbpedia.org/>

baseada em um método social evolucionário para encontrar os vizinhos mais próximos no método da filtragem colaborativa baseada no usuário. Para selecionar os k vizinhos de cada usuário-alvo eles treinam um algoritmo genético que prioriza os amigos que mais participaram em eventos a partir de atributos sociais (i.e. idade, sexo, nível de escolaridade e estado civil). Ao término, a nova função de similaridade é avaliada em um experimento com dados de dois usuários reais com relação a outras heurísticas já estabelecidas.

Um trabalho muito promissor foi desenvolvido em Yin et al. (2013) e Yin et al. (2014). Ao observarem uma forte similaridade dos domínios de RSBLs e RSBEs propuseram um modelo unificado capaz de recomendar tanto locais como eventos. O modelo LCA-LDA é uma evolução do LDA (BLEI; NG; JORDAN, 2003), modelo probabilístico gerador muito aplicado para a modelagem de dados textuais por meio de tópicos latentes. O novo modelo aprende os tópicos latentes das relações entre os usuários, as localizações e os conteúdos textuais dos eventos já frequentados. Dessa maneira os autores propõem uma solução para o *cold-start* do evento, que depende diretamente do histórico do usuário. O modelo é avaliado com dados do DoubanEvent[4] uma das maiores RSBEs da China, conhecida pela maioria dos usuários serem estudantes, e também com dados do Foursquare. O modelo proposto mostra-se superior a alguns modelos clássicos da literatura, como a filtragem colaborativa baseada na vizinhança de usuário e itens, em dois cenários aparentemente distintos para cada rede social: quando o usuário recebe recomendações de eventos na cidade onde mora, e quando este viaja para uma nova cidade. A diferença dos cenários está no fato que os modelos são treinados por cidade, no entanto, não existindo essa restrição os cenários se confundem.

Du et al. (2014) também avaliaram um novo modelo de recomendação na RSBE DoubanEvent. Considerando o problema como uma classificação binária, os autores propõem um classificador baseado em múltiplos fatores, no conteúdo textual dos eventos passados, na distância entre estes eventos, nas preferências temporais do usuário (i.e. dia da semana e hora do dia) e na relação do usuário-alvo com o organizador do evento (e.g. se é um seguidor ou não). Apesar dos múltiplos contextos todos se baseiam fortemente na descrição dos eventos, sendo este o principal fator dentre os demais. Um novo modelo baseado na fatoração de matrizes integra os múltiplos fatores. O modelo é avaliado com usuários que tenham mais de três eventos no passado e o mesmo alcança resultados marginalmente melhores do que a

[4]<http://www.douban.com/>

definição de pesos para os fatores por meio de uma árvore de decisão.

Daly e Geyer (2011) comparam o potencial da informação geográfica e da informação colaborativa na recomendação de eventos promovidos internamente pela IBM. Os eventos são inicialmente agrupados quanto ao seu nível de localidade, se ocorrerão via Web ou em um local físico, e de colaboração, se existe pouca ou muita sobreposição de eventos no histórico dos usuários participantes, resultando em quatro grupos de eventos. Para cada grupo aplicou-se o algoritmo de filtragem colaborativa baseado na vizinhança do item. Em uma primeira avaliação foram removidos os eventos com menos de cinco participantes e usuários com menos de cinco eventos no passado, uma taxa de revocação de apriximadamente 35% mostra que ainda há espaço para melhorias.

Pessemier et al. (2011) propõem o enriquecimento de dados de eventos culturais por meio de bases de dados públicas e bem definidas na Web como uma forma de possibilitar/melhorar os resultados da recomendação de eventos baseada em conteúdo. Khrouf e Troncy (2013) desenvolveram um algoritmo utilizando-se dessa ideia para melhorar as suas recomendações. Inicialmente eles propõem uma combinação linear da filtragem baseada em conteúdo com a filtragem colaborativa, no entanto, melhoram os resultados do primeiro algoritmo com o enriquecimento de palavras-chave usando dados de Linked Data[5] (BIZER et al., 2008). Logo após enfatizam a diversidade dos usuários mediante a definição de pesos para suas palavras-chave mais relevantes. O modelo foi avaliado com dados do Last.fm (uma rádio social online) considerando eventos musicais e após selecionarem os usuários com no mínimo 15 e no máximo 50 eventos (excluindo artificialmente a possibilidade de *cold-start* do usuário explicitamente) o modelo foi avaliado com uma partição temporal, treinado com 70% dos dados do passado e testados com o restante. Os resultados no gráfico de precisão e revocação mostram mostraram que o enriquecimento de dados trouxe pequenas melhorias quando comparado com a hibridização desse modelo de conteúdo enriquecido com um modelo baseado na filtragem colaborativa clássica.

Outros trabalhos propõem-se a recomendar eventos em contextos específicos, como conferências ou festivais. Apesar da similaridade do problema tratado nesta dissertação, a maioria dos eventos são de um mesma categoria geral (e.g. festival de música, conferência acadêmica), são pré-agendados por alguma empresa organizadora, e a maioria não tem o su-

[5] <http://linkeddata.org/>

porte de um sítio online onde os usuários podem se reunir previamente e posteriormente aos eventos (e.g. os grupos do Meetup, os planos do Plancast). Chin et al. (2012) cunharam o termo redes sociais *offline* efêmeras para denominar a rede social formada nesses conjuntos de eventos.

Liao et al. (2013) estudam uma rede social efêmera formada por duas fontes heterogêneas de agregação de pessoas, uma pelo encontro de pessoas nos eventos previamente agendados e outra nos chamados eventos espontâneos, quando as pessoas se aproximam e começam a conversar espontaneamente por determinado tempo. Essas redes dependem da utilização de dados em aparelhos móveis como a Identificação por Rádio Frequência (ou RFID) ou dispositos com Bluetooth.

No trabalho em questão, Liao et al. (2013) utilizaram a base de dados de "Attendee Meta-Data"(AMD), um projeto que busca explorar o potencial da utilização da tecnologia de RFID em conferências. Para a recomendação de eventos os autores definem três redes sociais latentes a partir das fontes heterogêneas citadas anteriormente: usuários com preferências similares, usuários que co-participaram em eventos (com uma restrição mínima de tempo) e usuários que se encontraram espontaneamente (com restrição de tempo e distância mínima). Essas redes são combinadas em um modelo unificado o qual é utilizado para recomendação de eventos. Na avaliação, os dados são particionados em passado e futuro, e o modelo mostra-se capaz de recomendar eventos que ainda não ocorreram na conferência (i.e. *cold-start* do evento), não sendo, no entanto, capaz de recomendar eventos para usuários com menos de 3 eventos no passado.

Forsblom et al. (2012) abordam um problema similar ao recomendarem eventos durante um festival de música em Helsinki. Os autores diferenciam-se dos demais trabalhos pois otimizam a métrica serendipidade. Métrica que pode ser traduzida como um acontecimento inesperado e com resultados benéficos, no contexto de recomendação de eventos é uma recomendação relativamente distinta do perfil de eventos passados do usuário e que o surpreenda beneficamente. Para tal, recomendam eventos baseando-se em dois algoritmos clássicos, a recomendação aleatória e a recomendação dos eventos mais próximos ao usuário. Para avaliação realizam um estudo de caso com dois grupos de usuários reais que afirmaram ser fiéis frequentadores do festival, e ao final os algoritmos alcançaram resultados satisfatórios para os grupos não havendo diferença estatística entre seus resultados.

Outros dois trabalhos avaliados sob a forma de estudos de caso com usuários reais foram desenvolvidos por Dooms, Pessemier e Martens (2011) e Minkov et al. (2010) desta vez com aplicação em RSBEs. Dooms, Pessemier e Martens (2011) compararam cinco algoritmos de recomendação para um sítio belga de eventos culturais. Os algoritmos foram estes: a recomendação aleatória, recomendação baseada em redução da dimensionalidade com SVD, filtragem baseada em conteúdo, filtragem colaborativa baseada no usuário e uma combinação da filtragem colaborativa com a baseada conteúdo. Cada usuário foi aleatoriamente atribuído a um dentre cinco grupos, e cada grupo recebeu recomendações de um dos algoritmos. Ao término, cada usuário respondeu um questionário dando notas para diferentes aspectos qualitativos das recomendações. A combinação do algoritmo baseado em conteúdo e a filtragem colaborativa baseada no usuário obteve os melhores resultados em quase todas as métricas com diferença estatística no teste de Wilcoxon com nível de confiança de 95%.

Minkov et al. (2010) buscaram solucionar o problema da recomendação de seminários científicos em universidades. Eles aplicaram o modelo RankSVM (JOACHIMS, 2006) que otimiza o ranking dos eventos para cada usuário a partir dos dados de conteúdo textual. Propuseram alternativamente uma modelo colaborativo que além de ser treinado uma única vez, aprende um modelo com dimensionalidade reduzida a partir do conteúdo textual dos eventos participados pelos usuários. Os dados de treino foram elicitados durante 15 semanas pelo *feedback* explícito de 90 usuários reais que responderam a formulários indicando quais seminários gostariam de participar. Os resultados do artigo mostram que o modelo colaborativo proposto foi superior ao RankSVM, e que a representação do conteúdo dos eventos foi definidora do resultado final, com o TFIDF (SALTON; WONG; YANG, 1975) superando a modelagem em tópicos com o LDA (BLEI; NG; JORDAN, 2003).

Outros trabalhos buscam ainda definir frameworks para implementação de sistemas de recomendação de eventos, considerando aspectos multi-contextuais principalmente para dispositivos móveis (BEER et al., 2013; PESSEMIER et al., 2013). Apesar de sua importância para a aplicação prática dos algoritmos desenvolvidos na academia, esse não é o foco desta dissertação.

A Tabela 2.1 e a Tabela 2.2 sumarizam os trabalhos estudados por meio de características do escopo do problema tratado, da solução proposta e do formato de avaliação. Percebemos que a maioria dos trabalhos soluciona o problema da recomendação de eventos, mas parte

deles trata do RSBEs e parte de redes efêmeras ou apenas de sites de eventos. Todos os trabalhos propõem modelos utilizando dados auxiliares com modelagens híbridas ou com modificações de algoritmos clássicos. Apesar disso nenhum trabalho soluciona o *cold-start* do evento e do usuário simultaneamente. Em termos de avaliação a maioria avalia a acurácia em experimentos *offline* (i.e. com dados históricos), poucos avaliam o *cold-start* explicitamente variando os níveis de esparsidade, poucos avaliam o impacto das mudanças culturais na preferência do usuário de forma explícita e nenhum deles avalia as mudanças temporais.

Apoiando-nos, portanto, sobre os "ombros" dessas pesquisas anteriores, reunimos informações valiosas que foram experimentadas no contexto de RSBEs. Somando a essas ideias nossas proposições, nos diferenciamos dos demais trabalhos por apresentamos uma solução híbrida de aprendizagem multi-contextual que lida diretamente com o *cold-start* do usuário e do evento, recomendando eventos sem restrição de categoria e avaliando-o para uma RSBE real com diversas condições culturais. Para tal utilizamos sinais contextuais inerentes ao domínio e presentes na própria base de dados da RSBE, sem a necessidade de enriquecimento dos dados por meio de fontes externas. Nesse modelo exploramos tanto algoritmos clássicos quanto do estado-da-arte, como a fatoração de matrizes multi-relacional que captura os fatores sociais latentes ao contexto social. Por fim, desenvolvemos e aplicamos uma metodologia de avaliação que mede a eficácia dos modelos em diversos momentos do tempo, capturando assim as mudanças temporais de preferência do usuário.

Características		Liu et al. (2012)	Qiao et al. (2014b)	Qiao et al. (2014a)	Pascoal et al. (2014)	Yin et al. (2013) e Yin et al. (2014)	Du et al. (2014)
Tipo de Recomendação	Eventos		X	X	X	X	X
	Usuários	X					
Tipo de RSBE	Persistente	Meetup	Meetup	Meetup	Facebook Events	Douban Event	Douban Event
	Efêmera ou Inexistente						
Tipos de Evento	Similares						
	Quaisquer	X	X	X	X	X	X
Tipo de Modelo	Apenas Contexto Alvo						
	Contextos Auxiliares	Difusão	Ranking	Híbrido	Clássico	Híbrido	Híbrido
Contextos Auxiliares	Social	X	X	X		X	X
	Conteúdo do Evento					X	X
	Geográfico			X		X	X
	Temporal						X
	Palavras-Chave						
	Atributos do Usuário				X		
Solução para o *Cold-Start*	Evento					X	X
	Usuário	X	X	X	X		
Avaliação com dados históricos	Acurácia	X	X	X	X	X	X
	Mudança Temporal						
	Mudança Cultural		Várias Cidades	Várias Cidades			
	Níveis de Esparsidade	X					
Avaliação por estudo de usuário							
Avaliação em sistema real						X	

Tabela 2.1: Sumário dos Trabalhos Relacionados (parte 1)

Características		Daly e Geyer (2011)	Khrouf e Troncy (2013)	Liao et al. (2013)	Forsblom et al. (2012)	Dooms, Pessemier e Martens (2011)	Minkov et al. (2010)
Tipo de Recomendação	Eventos	X	X	X	X	X	X
	Usuários						
Tipo de RSBE	Persistente						
	Efêmera ou Inexistente	Funcionários da IBM	Last.fm	Conferência Acadêmica	Festival	Sítio Belga de Eventos	Seminários Acadêmicos
Tipos de Evento	Similares	X	X	X	X		X
	Quaisquer					X	
Tipo de Modelo	Apenas Contexto Alvo						
	Contextos Auxiliares	Clássico	Híbrido	Híbrido	Clássico	Híbrido	Híbrido
Contextos Auxiliares	Social	X	X	X		X	X
	Conteúdo do Evento		X			X	X
	Geográfico	X			X		
	Temporal						
	Palavras-Chave		X				
	Atributos do Usuário						
Solução para o *Cold-Start*	Evento		X	X	X	X	
	Usuário						
Avaliação com dados históricos	Acurácia	X	X	X			
	Mudança Temporal						
	Mudança Cultural						
	Níveis de Esparsidade	X					
Avaliação por estudo de usuário					X	X	X
Avaliação em sistema real							

Tabela 2.2: Sumário dos Trabalhos Relacionados (parte 2)

Capítulo 3

Metodologia

Neste capítulo descrevemos o processo de coleta dos dados de uma RSBE real. Estudamos detalhadamente a relação entre usuários e eventos via RSVPs. Também especificamos um método mais realista e detalhado para avaliação dos modelos investigados que inclui particionamento temporal dos dados, avaliação estratificada por níveis de esparsidade. E finalizamos com a definição da métrica de ranking para comparação dos modelos de recomendação investigados nesta dissertação.

3.1 Coleta e Descrição dos Dados

O Meetup[1] é uma RSBE que promove o encontro de pessoas por meio de eventos. Ao se cadastrarem os usuários são motivados a filiarem-se e/ou criarem seus próprios grupos. Os grupos possuem interesses bem definidos (e.g. descritos por palavras-chave) e tem como objetivo principal permitir que os membros tenham um espaço *online* onde possam planejar, criar, divulgar e comentar eventos. O Meetup é uma das maiores RSBEs atualmente tendo registrado somente na primeira semana de Janeiro/2015 cerca 3.000 novos grupos e 330.000 usuários participando de eventos[2]. Os eventos do Meetup variam desde festas de aniversário com a família até congressos com milhares de pessoas.

Desenvolvemos um framework de coleta de dados que nos permitiu recuperar dados históricos da RSBE acessando diretamente a API REST do Meetup[3] para cidades específicas.

[1] <http://www.meetup.com>
[2] <http://blog.meetup.com/chart>
[3] <http://www.meetup.com/meetup_api/>

Selecionamos então três cidades dos Estados Unidos da América (EUA), Chicago, Phoenix e San Jose, pelos seguintes motivos: (i) os EUA é o país de origem do Meetup, provavelmente com a maior quantidade de grupos e eventos do site (ii) estas cidades estão entre as 10 mais populosas do país podendo indicar uma grande atividade de eventos sociais e (iii) por se localizarem em estados distintos, espera-se que representem uma certa diversidade cultural.

Para cada cidade foram coletados todos os seus grupos e, a partir daí, recuperamos as demais entidades. A Figura 3.1 apresenta o modelo entidade relacional do banco de dados montado a partir dos dados coletados. Esse modelo ajuda a entender a dinâmica da RSBE. Com o objetivo principal de reunir usuários em eventos, usuários do Meetup criam e publicam eventos através de seus grupos, e os demais podem expressar seus interesses pelos RSVPs com uma resposta positiva ou negativa, "sim" ou "não". Ao enviar um RSVP para um evento o usuário é instantaneamente considerado como membro do grupo, podendo então ficar melhor informado sobre os eventos futuros publicados pelo grupo em questão. No entanto, é importante ressaltar que os usuários tem acesso a todos os eventos no Meetup, independentemente dos grupos que os criaram ou suas localizações geográficas, e podem enviar RSVPs para estes eventos a qualquer momento. Cada evento possui uma localização cadastrada pelo criador e tanto os usuários como os grupos definem seus interesses por meio de *tags*, i.e., palavras-chave representando o tipo dos eventos.

Para fins de análise e avaliação experimental foram coletados dados de Janeiro, 2010 a Abril, 2014, dando margem a um estudo histórico abrangente da RSBE. A Tabela 3.1 sumariza as características da base de dados e enfatiza a extrema esparsidade da matriz de RSVPs formada pelas entidades usuário e evento (ver Figura 3.1).

Cidade	**# Grupos**	**# Usuários**	**# Eventos**	**# RSVPs**	**Esparsidade dos RSVPs**
Chicago	2.321	207.649	190.927	1.375.154	99,996%
Phoenix	1.661	117.458	222.632	1.209.324	99,995%
San Jose	2.589	242.143	206.682	1.607.985	99,996%

Tabela 3.1: Estatísticas dos Dados

A esparsidade de uma matriz é calculada como sendo a quantidade de células nulas sobre o total de células da matriz. Transformando a relação RSVP em uma matriz de usuários e eventos, formalizamos a esparsidade dos RSVPs como segue:

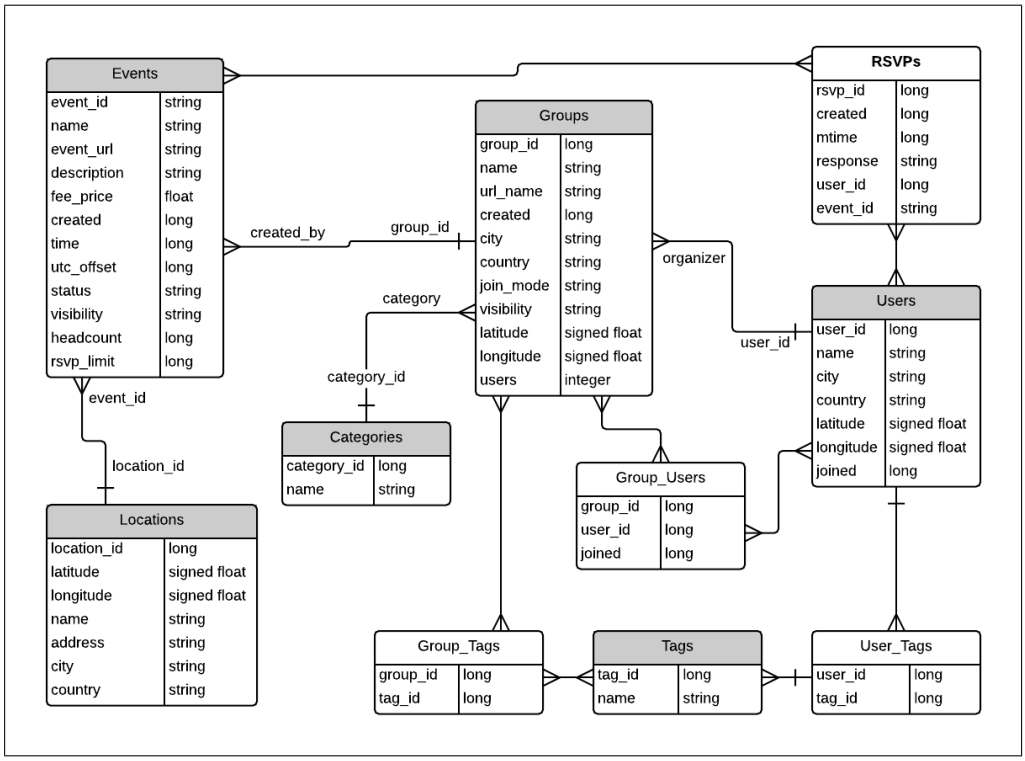


Figura 3.1: Modelo Entidade Relacional dos Dados Coletados do Meetup

$$\text{Esparsidade} := 1 - \left(\frac{\#\text{RSVPs}}{\#\text{Usuários} \times \#\text{Eventos}}\right) \tag{3.1}$$

3.2 Análise dos RSVPs

A meta dos modelos de recomendação de eventos é prever os próximos eventos que o usuário-alvo estará presente. Como a informação sobre a presença real dos usuários nos eventos não é disponibilizada pelo Meetup em sua API REST, assumimos os RSVPs positivos (ou RSVPs apenas, de agora em diante) como um *proxy* dessa informação. Assim, os modelos devem prever quando o usuário-alvo irá enviar uma resposta "sim" para o evento. Para compreendermos melhor o problema e obtermos *insights* para a solução do problema, realizamos nas próximas subseções uma análise exploratória nos dados coletados.

3.2.1 Quantidade de RSVPs por Evento

A Figura 3.2 mostra a distribuição acumulada da quantidade de RSVPs positivos por evento para todas as cidades. Os números mostram que mais de 45% dos eventos possui apenas 1 RSVP e aproximadamente 90% tem no máximo 10 RSVPs em todas as cidades. A escala logarítmica do eixo-x explicita o forte viés para a direita da distribuição, levando-nos a crer que a maioria dos eventos, nas cidades investigadas, possuem uma baixa taxa de participação.

Esses números representam um problema em potencial para os algoritmos de recomendação baseados em filtragem colaborativa, já que os mesmos são conhecidos por se deteriorarem diante de altas taxas de esparsidade.

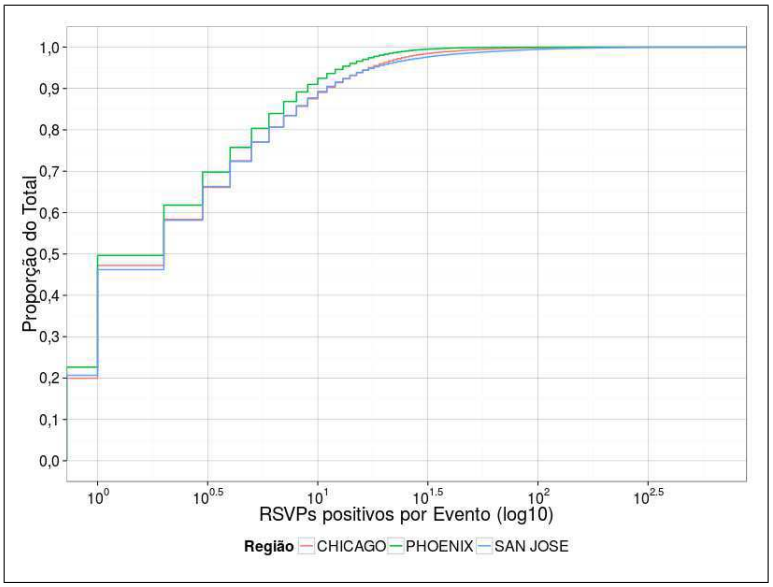


Figura 3.2: Distribuição Acumulada do # RSVPs positivos por Evento

3.2.2 Tempo de Vida dos Eventos

O intervalo de tempo entre a criação do evento até a sua ocorrência define o tempo de vida do mesmo. Compreender a longevidade dos eventos permite-nos afirmar quão voláteis são os eventos na RSBE. Na Figura 3.3 podemos ver que a maioria dos eventos variam de uma semana até 100 dias. Enquanto uma pequena parcela dos eventos existe por apenas 1 dia, o menor viés da curva demonstra que a maioria permanece ativo por um tempo razoável aumentando assim a probabilidade de serem descobertos pelos usuários.

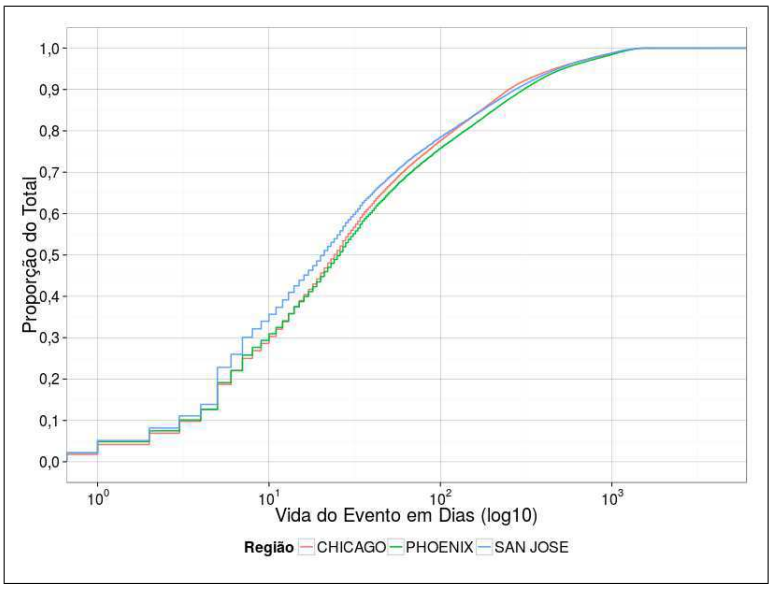


Figura 3.3: Distribuição Acumulada do Tempo de Vida do Evento

3.2.3 Quando os RSVPs ocorrem?

Durante o tempo de vida do evento os RSVPs podem acontecer a qualquer momento, no entanto, qual o momento que os RSVPs tendem a ocorrer? A Figura 3.4 mostra o percentual da vida do evento em que os primeiros 21 RSVPs positivos[4] ocorreram para todas as cidades. Para cada k-ésimo RSVP agregamos os percentuais para todos os eventos que tiveram ao menos k eventos, por exemplo, se um evento X da cidade de Chicago teve 10 RSVPs positivos, ordenamo-os temporalmente e calculamos os percentuais que aparecerão nos 10 primeiros diagramas de caixa vermelhos do eixo-x. O eixo-y representa a proporção de vida do evento, sendo 0 a criação e 1 a ocorrência.

Podemos perceber que quanto mais RSVPs tem um evento, mais próximo da ocorrência eles serão enviados. Por mais óbvio que aparente ser, essa aproximação se define desde os primeiros RSVPs, crescendo rapidamente. Apesar das cidades apresentarem pequenas variações, sendo Phoenix a cidade em que essa observação é mais acentuada, todas possuem um padrão similar, i.e. tendência de receberem RSVPs próximos à ocorrência do evento. No contexto de sistemas de recomendação uma observação como essa tem importantes consequências, por exemplo, na criação do evento quando a informação colaborativa é escassa algoritmos baseados em conteúdo ou nos demais contextos seriam de grande utilidade. E

[4] 95% dos eventos da base de dados tem no máximo 21 RSVPs positivos.

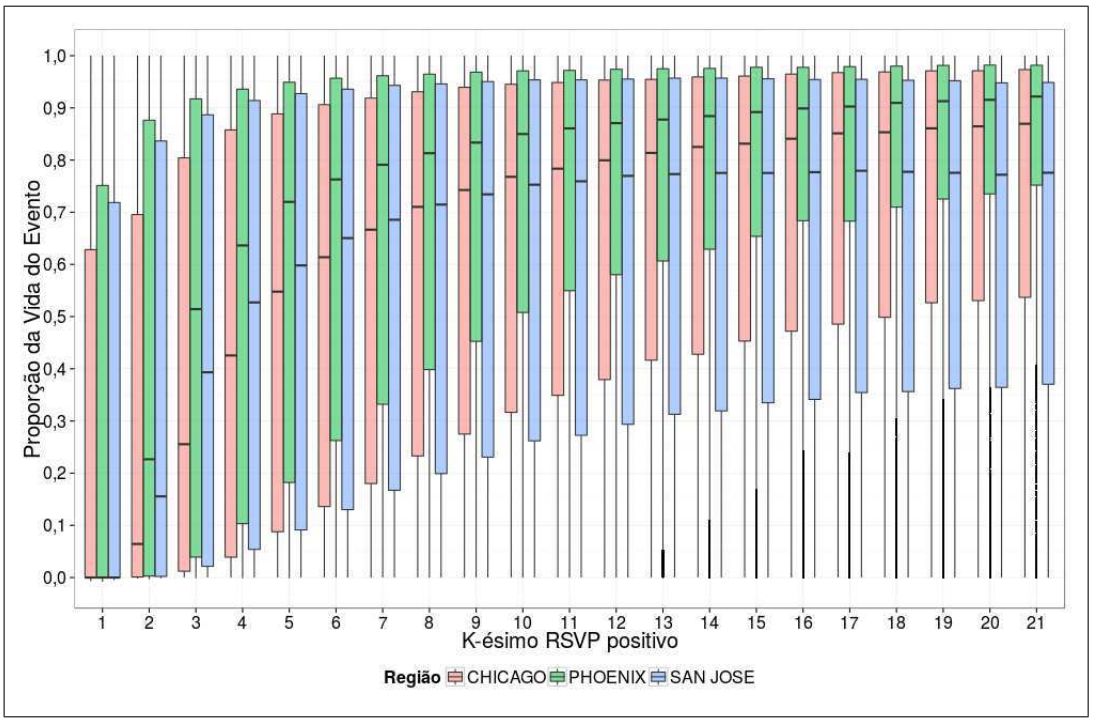


Figura 3.4: Distribuição Acumulada do Tempo do K-ésimo RSVP relativo à vida do Evento

com a aproximação do evento mais RSVPs seriam recebidos favorecendo os métodos de filtragem colaborativa.

3.3 Método de Avaliação

3.3.1 Particionamento dos Dados

Os dados coletados foram utilizados para criar um ambiente de avaliação o mais similar possível com a realidade da RSBE. A base de dados de cada cidade foi temporalmente particionada em 12 momentos[5] igualmente espaçados no tempo $\tau_p \in \mathcal{T}$, como mostra a Tabela 3.2.

Para cada partição p o treino foi formado pelos eventos criados e ocorridos nos 6 meses anteriores a τ_p (passado), e o teste pelos eventos criados nesse intervalo de 6 meses mas que ocorrerão depois de τ_p (futuro). A Figura 3.5 descreve com mais detalhes esse particionamento temporal. É importante ressaltar alguns pontos: apenas os RSVPs positivos são usados (marcados com um "S" de "sim" na Figura 3.5), os eventos criados antes dos 6 me-

[5]Número escolhido de forma empírica para gerar uma boa divisão das partições nos conjuntos de validação e avaliação, com 4 e 8 respectivamente, e para que os experimentos fossem executados em tempo hábil.

Conjunto	**Partição** (p)	**Dia e Hora** (τ_p)
Validação	1	02/05/2010 01:33:28
	2	31/08/2010 03:06:56
	3	30/12/2010 04:40:24
	4	30/04/2011 06:13:52
Avaliação	5	29/08/2011 07:47:19
	6	28/12/2011 09:20:47
	7	27/04/2012 10:54:15
	8	26/08/2012 12:27:43
	9	25/12/2012 14:01:10
	10	25/04/2013 15:34:38
	11	24/08/2013 17:08:06
	12	23/12/2013 18:41:34

Tabela 3.2: Partições Temporais

ses ou depois de um τ_p são removidos e que não existe interseção entre os dados de duas partições consecutivas.

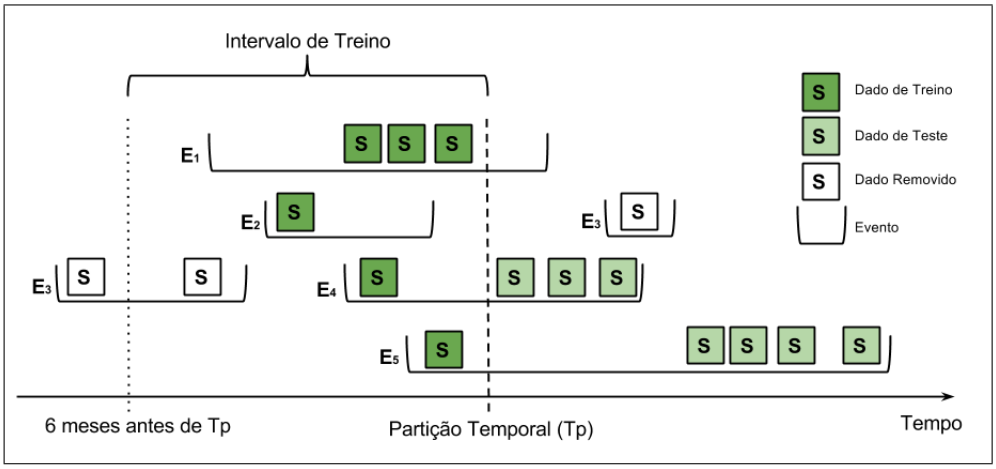


Figura 3.5: Particionamento Temporal em Treino e Teste

Agrupamos as partições em dois conjuntos a fim de facilitar a compreensão e as futuras referências às partições dos experimentos realizados. O primeiro conjunto, chamado de validação, foi formado pelas 4 partições iniciais (cada uma com treino e teste) as quais foram utilizadas para seleção de atributos, hiper-parâmetros e ajustes do modelo híbrido. O segundo conjunto, chamado de avaliação, contém as demais partições, sobre as quais foi executado o experimento final que comparou os nossos melhores modelos com os métodos propostos pela literatura. Por exemplo, propomos um algoritmo $\mathcal{A}$ de recomendação de eventos, o qual precisa definir o valor de um hiper-parâmetro α. Para cada valor de α considerado, reali-

zamos 4 ciclos de treino, teste e avaliação das recomendações de $\mathcal{A}$ correspondentes às 4 partições do conjunto de validação. Ao final, selecionamos o α que resultou no melhor modelo de $\mathcal{A}$ comparando as medianas. Este modelo será então comparado com os algoritmos da literatura, para tal seguimos processo análogo de treino, teste e avaliação, desta vez sobre as partições do conjunto de avaliação usando o melhor α encontrado na validação.

O diferencial desse particionamento com relação aos métodos de particionamento clássicos com seleções aleatórias dos conjuntos de treino e teste, está na manutenção das características temporais dos dados (LATHIA, 2010). Ao preservar essa informação os modelos investigados serão avaliados considerando a variação temporal dos perfis dos usuários como variações na própria RSBE como o aumento do número de usuários, grupos e eventos.

3.3.2 Avaliação por Níveis de Esparsidade

Nesta seção definimos um formato de avaliação que permite-nos comparar os algoritmos de recomendação de eventos por níveis de esparsidade.

Em uma dada partição, para recomendar eventos corretamente o algoritmo deve ser capaz de predizer os RSVPs que serão enviados pelos usuários do teste $u \in U^{teste}$ para os eventos candidatos $e \in E^{teste}$. Cada usuário u possui $|E_u^{treino}|$ eventos no treino, da mesma forma cada evento e possui $|U_e^{treino}|$ usuários no treino, calculados diretamente sobre os RSVPS passados. Estratificamos então o conjunto de usuários U^{teste} e eventos E^{teste} com relação aos tamanhos de seus históricos de RSVPs no treino e chamamos essa divisão de níveis de esparsidade S como segue

$$S := \{0, 1, 2, 3, 4, 5, \text{6-10}, \text{11-20}, > 20\} \tag{3.2}$$

Esses níveis são baseados no forte viés das distribuições (ver Seção 3.2), criamos um grupo para cada valor inicial (i.e. de 0 a 5) e agregamos os demais da cauda da distribuição (i.e. > 5) em três grupos (i.e. 6-10, 11-20 e > 20). Assim, criamos cenários de análise cada um com usuários e eventos similares em termos do tamanho do histórico de RSVPs. Com essa ferramenta de análise podemos entender o comportamento dos modelos propostos desde o usuário no *cold-start* até usuários com muitos RSVPs, como também desde o evento no *cold-start* até grandes eventos.

Como primeiro resultado dessa estratificação contamos a quantidade de usuários do teste $u \in U^{teste}$ por nível de esparsidade para cada uma das 12 partições. A Figura 3.6 apresenta-nos no eixo-y a quantidade de usuários, com as cores diferenciando os níveis de esparsidade, e o diagrama de caixa agregando estas quantidades para as partições. Vemos que a maioria dos usuários tem poucos ou nenhum evento no passado, e que menor tende a ser esta quantidade quanto mais RSVPs tenha o usuário. Mais ainda, esse padrão se mantém em todas as cidades investigadas.

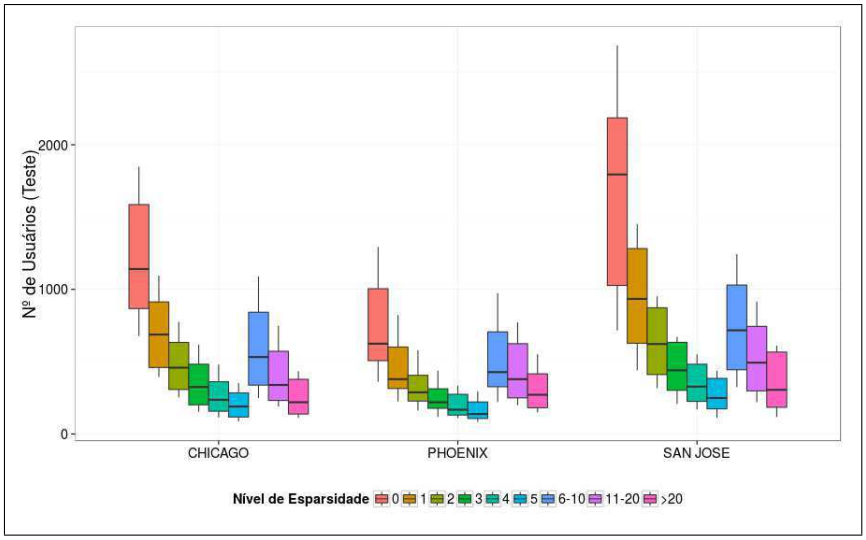


Figura 3.6: Esparsidade do Usuário

Similarmente, os eventos do teste $e \in E^{teste}$ foram analisados por nível de esparsidade na Figura 3.7. Vemos que a distribuição é ainda mais enviesada, com a grande maioria dos eventos não tendo recebido RSVP algum. Esse viés varia um pouco com o tempo, como podemos ver no comprimento das caixas para o nível de esparsidade 0, mas em geral se mantém.

Podemos concluir que não importa o momento que particionemos os dados, sempre haverá um grande número de eventos e usuários sem RSVPs, consolidando a hipótese que o *cold-start* é um problema inerente ao domínio.

3.3.3 Métricas de Avaliação

A recomendação de eventos é uma instância do problema de recomendação dos top-n itens na área de sistemas de recomendação. Este problema é frequentemente tratado como de ran-

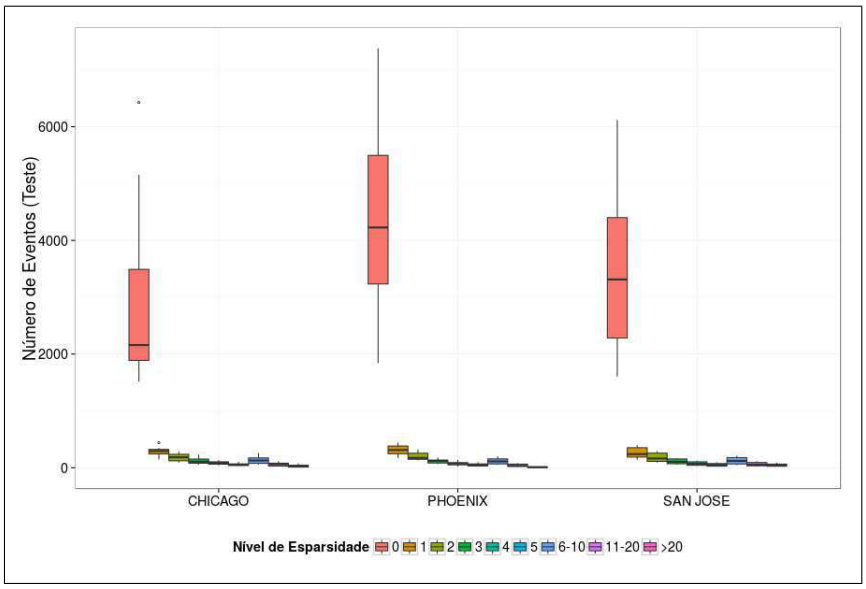


Figura 3.7: Esparsidade do Evento

king personalizado, ou seja, a geração de uma lista ranqueada de eventos ordenados decrescentemente pela probabilidade de receberem um RSVP positivo do usuário. Para mensurar o sucesso nessa tarefa utilizamos a métrica *Normalized Discounted Cumulative Gain* (NDCG) comumente aplicada para avaliar algoritmos em problemas de ranking (VALIZADEGAN et al., 2009; CAO et al., 2007; XU; LI, 2007). Calculamos a NDCG truncada para as 10 primeiras recomendações, i.e. $NDCG@10$ (tamanho de lista muito utilizado na literatura), para cada usuário $u \in U^{teste}$, a qual pode ser formulada como segue.

$$DCG@10 := \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{log_2(i+1)} \tag{3.3}$$

$$NDCG@10 := \frac{DCG@10(u)}{IDCG@10(u)} \tag{3.4}$$

onde rel_i assume o valor de 1 ou 0 se o evento na posição i é relavante ou não respectivamente, e a função $IDCG_p(u)$ retorna o valor de ranking perfeito, agindo como um termo de normalização.

Nos próximos três capítulos, Cap. 4 ao Cap. 6, apresentamos os modelos contextuais propostos nesta dissertação selecionados com os dados de validação. Em seguida, no Capítulo 7, definimos e avaliamos o modelo Híbrido de aprendizagem multi-contextual formado pela composição dos modelos contextuais.

Capítulo 4

Contexto Social

A facilidade de comunicação desde o início tem sido o principal motor da Web como um todo. Comunicar e relacionar-se é característico do ser humano, gregário, social por natureza, não é por acaso que as redes sociais tem alcançado tão ampla aceitação. As RSBEs da mesma maneira buscaram reunir as pessoas com gostos similares sobre a estrutura de duas redes sociais, uma *online* que reúne usuários por meio da internet, e outra rede *offline* formada pela interação presencial. A Figura 4.1 apresenta as entidades que dão origem as duas redes sociais no Meetup. Quando um usuário se afilia a um grupo, ele se liga indiretamente aos seus membros (rede social *online*), da mesma forma ao participar de um evento (i.e. enviar um RSVP positivo), ele se relaciona indiretamente aos demais usuários participantes do mesmo evento (rede social *offline*). A Figura 4.1 também mostra uma terceira relação de criação dos eventos pelos grupos (linhas pontilhadas), a qual não define uma rede social mas que foi de crucial importância para o sucesso do modelo multi-relacional proposto na Seção 4.3.

Neste capítulo apresentamos algoritmos que se utilizam da informação contida nas relações entre usuários e eventos $R_{UE} \subseteq U \times E$, usuários e grupos $R_{UG} \subseteq U \times G$ e grupos e eventos $R_{GE} \subseteq G \times E$ para recomendar eventos personalizadamente.

4.1 Algoritmos de Filtragem Colaborativa

A relação R_{UE} foi explorada por meio dos conhecidos algoritmos de filtragem colaborativa. Esses algoritmos usam um banco de dados de preferência dos usuários por itens extraídos

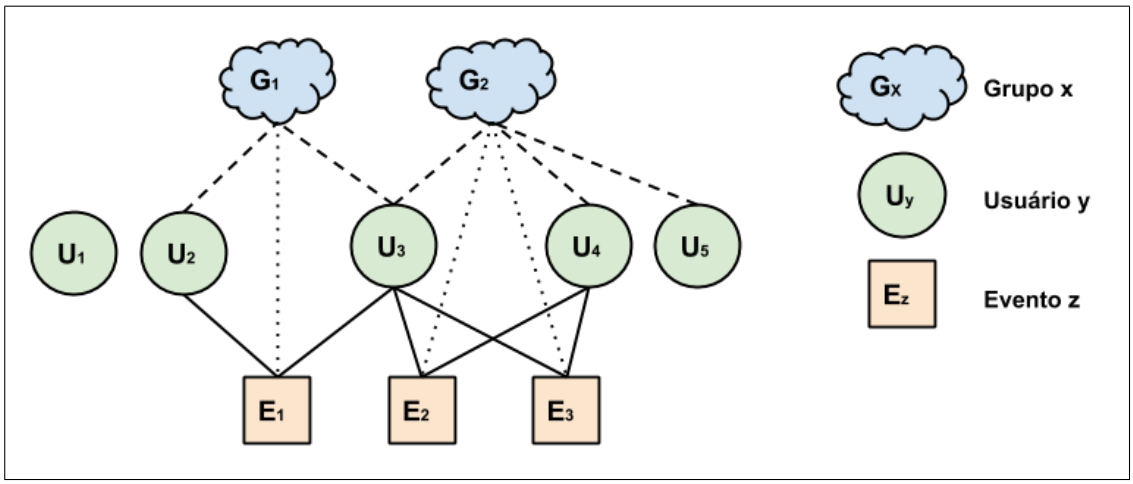


Figura 4.1: Exemplo da RSBE Meetup com relacionamentos *online* (grupos) e *offline* (eventos)

de forma explícita com notas que variam em escalas, e.g. 0 a 5 estrelas como o Netflix[1], ou implicitamente, e.g. pelos cliques de usuários em páginas web.

Os algoritmos colaborativos baseiam-se na hipótese de que se um usuário u_1 gosta apenas do item i e outro usuário u_2 também gosta desse item então o usuário u_2 é mais similar a u_1 do que um terceiro usuário u_3 que não goste de i. Assim, quanto maior for a correspondência de gostos entre dois usuários maior será a confiança que o algoritmo terá de recomendar itens de u_1 para u_2 e vice-versa. Quando deseja-se recomendar usuários para itens, a mesma hipótese se mantém trocando-se apenas as entidades.

4.1.1 Algoritmo de Vizinhança do Item

O conceito de vizinhança do item pode ser entendido por meio da abstração de um grafo não dirigido cujos nós representam os itens e as arestas interligam itens entre si, formando sua vizinhança. Os pesos dessas arestas definem a proximidade entre os vizinhos calculada por meio de uma função de similaridade específica. Por exemplo, na Figura 4.1 observando apenas a rede social *offline* (via eventos), o evento e_2 é vizinho de e_1 e de e_3, pois possuem ao menos um usuário em comum. No entanto, e_2 é mais próximo de e_3 do que de e_1, pois dois usuários, u_3 e u_4, co-participaram de ambos os eventos, enquanto que apenas u_3 co-participou de e_2 e e_1.

[1]<http://www.netflix.com>

Com o objetivo de recomendar eventos para usuários, utilizamos o algoritmo clássico de filtragem colaborativa baseado nos k-vizinhos mais próximos em sua versão que calcula a similaridade de itens, o ITEM-KNN (DESHPANDE; KARYPIS, 2004). A hipótese que nos motivou é de que usuários tendem a ir a eventos similares no futuro àqueles que foram no passado. O perfil dos usuários são representados na forma de uma matriz esparsa $\mathbf{M} \in \mathbb{R}^{|U|\times|E|} \cup \{.\}$, com $\{.\}$ denotando os valores ausentes. A matriz pode ser decomposta em vetores coluna

$$\mathbf{M} := [\vec{m}_1,...,\vec{m}_{|E|}] \quad \text{com} \quad \vec{m}_r := [m_{1,r},...,m_{|U|,r}]^T, \quad \text{para} \quad r := 1,\ldots,|E|, \tag{4.1}$$

onde cada vetor coluna $\vec{m}_e$ corresponde aos usuários que deram RSVPs positivos para o evento e. Assim cada vetor linha $\vec{m}_r$ da matriz é formada por valores 0 e 1. A similaridade entre pares de eventos é calculada pela similaridade do cosseno, i.e.,

$$\text{cosseno}(e,e') := \frac{\langle \vec{m}_e,\vec{m}_{e'} \rangle}{\|\vec{m}_e\|\|\vec{m}_{e'}\|}. \tag{4.2}$$

Para cada evento candidato $e \in E_{teste}$ calcula-se a similaridade com todos os demais e seleciona-se os top-k eventos mais similares, i.e. os vizinhos. No momento da recomendação de e para um usuário u, o valor de e será proporcional à soma das similaridade dos eventos vizinhos que u participou. Mais formalmente temos:

$$\hat{s}_{\text{ITEM-KNN}}(u,e) := \frac{\sum_{i=1}^{k} cosseno(e,e_i) \times i(u,e_i)}{\sum_{i=1}^{k} cosseno(e,e_i)}, \tag{4.3}$$

onde $i(u,e)$ é uma função indicadora que retorna 1 se o usuário u participou do evento e no passado, e 0 caso contrário.

4.1.2 Algoritmo de Fatoração de Matrizes

A fatoração de matrizes tem sido amplamente utilizada como uma alternativa aos métodos clássicos de filtragem colaborativa atingindo resultados similares ou melhores em diversos domínios além de ser mais eficiente computacionalmente (KOREN; BELL; VOLINSKY, 2009). Em vários problemas de listas ranqueadas como a recomendação de eventos, o algoritmo *Bayesian Personalized Ranking Matrix Factorization* (BPR-MF) (RENDLE et al.,

2009) tem sido utilizado com sucesso.

Formalizando o problema da recomendação de listas ranqueadas, para uma dada entidade x de uma relação R nós queremos ranquear objetos o de acordo com a chance deles serem observados em R. Definimos então o conjunto de instâncias que são relacionadas à entidade x via relação R como:

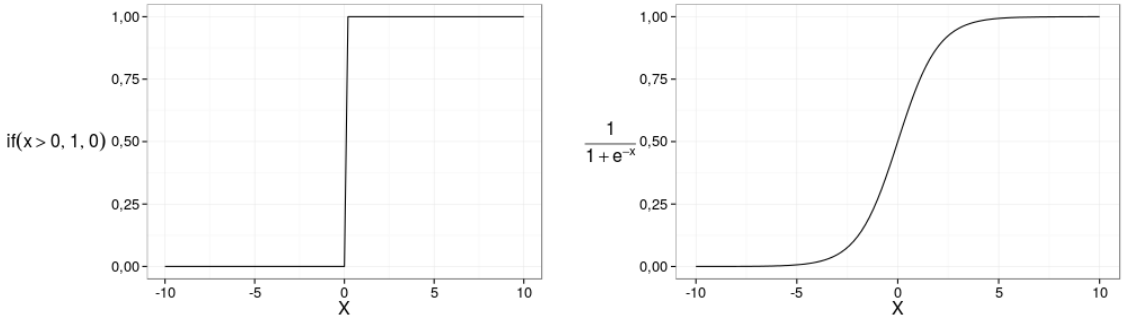
$$O_x^R := \{o | o \in O^R \wedge (x,o) \in R\}. \tag{4.4}$$

Por exemplo, $E_u^{R_{UE}}$ denota o conjunto de eventos que o usuário u participou na relação usuário-evento R_{UE} e $G_u^{R_{UG}}$ denota o conjunto de grupos que o usuário u é afiliado na relação usuário-grupo R_{UG}. Para facilitar a leitura, iremos especificar a relação R apenas em casos de ambiguidade (i.e. as mesmas entidades em duas ou mais relações). Dessa forma, a notação $E_u^{R_{UE}}$ é equivalente à notação E_u.

O BPR-MF aborda o problema de ranking otimizando o critério $BPROpt$ análogo à métrica AUC (Eq. 4.5), que em termos de ranking conta a quantidade de pares de objetos corretamente ranqueados. A intuição da métrica é que se um objeto o_1 tem melhor ranking que outro o_2 para uma dada entidade x então os valores retornados pelo algoritmo devem ser também maiores para o par (x,o_1) do que para o par (x,o_2).

$$AUC(x,\hat{s}) := \frac{1}{|O_x^R||O^R \setminus O_x^R|} \sum_{o \in O_x^R, o' \in O^R \setminus O_x^R} \delta(\hat{s}(x,o) - \hat{s}(x,o')) \tag{4.5}$$

onde $\delta(v)$ é uma função indicadora resultando em 1 se v é positivo, e 0 caso contrário. Como $\delta(v)$ é uma função descontínua do tipo escada (Fig. 4.2a) aproxima-se a mesma pela função logística (Fig.4.2b).



(a) Função Escada (b) Função Logística

Figura 4.2: Comparação de Funções Indicadoras

Encontrando-se assim um erro (logarítmico) diferenciável

$$\ell(\hat{s}(x,o), \hat{s}(x,o')) := \ln \sigma(\hat{s}(x,o) - \hat{s}(x,o')). \tag{4.6}$$

O $BPROpt$ pode ser definido como

$$\text{BPROpt}(R, \hat{s}) := \sum_{o \in O_s^R, o' \in O^R \setminus O_s^R} \ln \sigma(\hat{s}(x,o) - \hat{s}(x,o')). \tag{4.7}$$

Esse critério reduz o problema de ranking para uma classificação pareada, i.e., verifica se um dado par é corretamente ranqueado ou não. Para ilustrar, damos um exemplo com a nossa relação alvo, os RSVPs. Para cada usuário no treino $u \in U_{treino}$, amostra-se um evento que ele participou $e \in E_u$, e outro evento e' que ele não participou $e' \in E_{treino} \setminus E_u$. O critério então verifica se e tem ranking maior do que e' para u. Ao aprender uma função $\hat{s}_{\text{BPRMF}}$ otimizada para o $BPROpt$ pode-se criar uma ordenação linear para cada usuário u sobre os eventos por

$$e <_u e' \quad \Leftrightarrow \quad \hat{s}_{\text{BPRMF}}(u, e) > \hat{s}_{\text{BPRMF}}(u, e'). \tag{4.8}$$

4.2 Heurística de Frequência nos Grupos

Os grupos são uma das principais funcionalidades das RSBEs, sendo responsáveis principalmente por manter e fortalecer as relações sociais. Neles os usuários planejam, marcam, divulgam e comentam os eventos. Essa característica tem motivado a literatura para detecção de comunidades e recomendação de eventos (LIU et al., 2012; QIAO et al., 2014b, 2014a) a explorar a sinergia das redes sociais *online* e *offline* com o objetivo de detectar comunidades de usuários para recomendação de grupos e eventos. Baseando-nos no mesmo pressuposto, propomos aqui um novo modelo que explora essa sinergia das relações R_{UE} e R_{UG} heuristicamente, sem materializar as redes sociais. Apesar de sua simplicidade, tal modelo alcança resultados muito promissores (ver Seção 4.4).

Um bom medidor do potencial agregador de um grupo, mais do que a quantidade de membros, é a frequência desses membros nos eventos. Partindo dessa ideia definimos o recomendador *GRUPO-FREQUENTE* baseado no gosto do usuário pelos grupos. A intuição

desse algoritmo é que a probabilidade de um usuário $u \in U$ participar de um evento $e \in E$ criado por um grupo $g \in G$ depende da frequência de participação em eventos anteriores $e \in E_u$ criados por g. Em outras palavras, quanto mais eventos um usuário participa em um grupo, maiores são as chances de ir a outro evento criado pelo mesmo grupo. Formalmente temos

$$\hat{s}_{\text{GRUPO-FREQ}}(u,e,g) := \frac{|E_{u,g}|}{|E_u|} \quad \text{com} \quad \text{e} \in E_g \quad e \quad |E_u| > 0, \tag{4.9}$$

onde $E_{u,g}$ denota o conjunto de eventos criados pelo grupo $g \in G$ que o usuário u participou. Usuários no *cold-start* (i.e. $|E_u| = 0$) não recebem recomendação.

4.3 Fatoração Multi-Relacional de Matrizes

Um dos maiores problemas enfrentados pelos algoritmos de filtragem colaborativa é o *cold-start*, como é o caso dos dados de RSBEs (ver Cap. 3). Técnicas de aprendizagem multi-relacional, que exploram as relações auxiliares ou contextuais para otimizar o aprendizado da relação alvo, tem sido aplicadas para os casos de ausência de informação. A fatoração multi-relacional de matrizes tem vários exemplos de sucesso na literatura (BOUCHARD; YIN; GUO, 2013; SINGH; GORDON, 2008; LIPPERT et al., 2008). Essa abordagem representa as relações auxiliares como matrizes binárias que são fatoradas conjuntamente com a relação alvo. A ideia é que atributos latentes são compartilhados entre as relações que envolvem as mesmas entidades.

Para a recomendação de eventos no Meetup, propomos o aprendizado da relação alvo R_{UE} auxiliada pelas relações R_{UG} e R_{GE} adicionando ineditamente o sinal de criação de eventos pelo grupos inerente à RSBE. Assim, os fatores latentes das entidades U, E e G são compartilhados por duas relações cada um, como mostra a Figura 4.3. A direção de cada relação é importante pois indica a ordem em que as entidades são otimizadas internamente, por exemplo na relação R_{UE} o modelo tentará otimizar o ranking de eventos para usuários e não o contrário.

Este modelo partiu da seguinte intuição. Retomando a Figura 4.1 vemos que os usuários u_2, u_3 e u_4 já participaram de eventos e, portanto, terão seus fatores latentes otimizados mesmo que utilizássemos apenas a relação R_{UE}. Apesar disso, o usuário u_5 não receberia

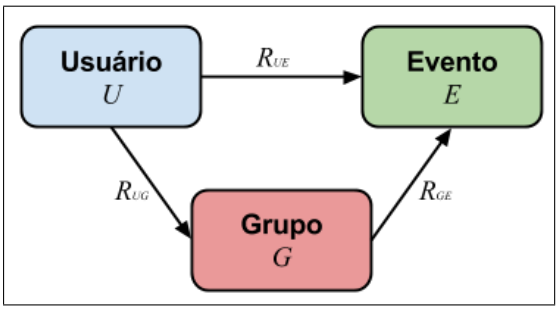


Figura 4.3: Entidades e suas Relações

recomendações. Assim, adicionamos a relação R_{UG} permitindo que os fatores dos usuários aprendam as interações com eventos e com grupos. Ainda assim, a relação entre o grupo g_2 e seus eventos e_2 e e_3 seria realizada apenas indiretamente nos fatores latentes do usuário (e.g. u_3 e u_4 tenderiam a ter fatores similares). Então, concluímos com a adição da terceira relação R_{GE} que concretizou este aprendizado nos fatores dos grupos, definindo assim um caminho alternativo entre as entidade U e E.

Como método de aprendizado propomos a utilização do *Multi-Relational Factorization with Bayesian Personalized Ranking* (MR-BPR) (KROHN-GRIMBERGHE et al., 2012) um algoritmo que soluciona o problema da predição de listas ranqueadas otimizando uma extensão da função $BPROpt$ (Eq. 4.7) para múltiplas relações binárias. Essa técnica foi aplicada no domínio da recomendação de vídeos do Youtube[2], de fotos no Flickr[3] e de blogs no BlogCatalog[4]. Em todos os domínios utilizou a rede social como relação auxiliar e ao avaliá-lo unicamente com usuários no *cold-start* obteve resultados superiores a outras técnicas do estado-da-arte desenvolvidas para a solução do *cold-start* (TANG; LIU, 2009a, 2009b). Sendo a recomendação de eventos também um problema de ranking com dados de feedback implícito e positivos (RSVPs positivos) aplicamos o MR-BPR em nossa investigação.

O MR-BPR codifica as entidades em matrizes cujas linhas representam as entidades em questão e as k colunas os valores de fatores latentes calculados para aquela entidade. Por exemplo, o conjunto U de usuários é associado com a matriz $\mathbf{U} \in \mathbb{R}^{|U| \times k}$. A linha $\mathbf{u}_i \in \mathbb{R}^k$ representa o vetor de atributos do i-ésimo usuário.

A tarefa é então encontrar uma função $\hat{s}_{UE}$ que reconstrua a relação alvo R_{UE} (i.e.

[2]<https://www.youtube.com>
[3]<https://www.flickr.com/>
[4]<http://www.blogcatalog.com/>

RSVPs positivos). O MR-BPR define essa função como o produto vetorial entre os vetores latentes das entidades relacionadas. Por exemplo, o valor previsto para uma relação R_{UE} entre o usuário $u \in U$ e o evento $e \in E$ é:

$$\hat{s}^{UE}_{\text{MR-BPR}}(u,e) = \sum_{f=1}^{k} \mathbf{u}_f \mathbf{e}_f, \tag{4.10}$$

com k sendo a dimensão do vetor de atributos latentes.

Para solucionar o problema de ranking de eventos o MR-BPR otimiza uma extensão do critério $BPROpt$ da Equação 4.7 para o caso multi-relacional gerando um novo critério de otimização

$$\text{MR-BPROpt}(\mathcal{R}, \Theta) = \gamma \sum_{R \in \mathcal{R}} \alpha_R \text{BPROpt}(R, \hat{s}_R) + \sum_{\theta \in \Theta} \lambda_\theta \|\theta\|^2, \tag{4.11}$$

onde $\mathcal{R}$ é o conjunto de relações R, γ é a taxa de aprendizado global, Θ são os parâmetros do modelo, i.e., as matrizes de fator latente, $\| \cdot \|$ denota a norma Frobenius e α_R é peso da relação R, sendo os pesos normalizados de forma que:

$$\sum_{R \in \mathcal{R}} \alpha_R = 1. \tag{4.12}$$

Em nosso caso existem três tipos de entidades, $\mathcal{E} = \{U, E, G\}$, sendo os parâmetros do modelo suas respectivas matrizes de atributos latentes, dado por $\Theta := \{\mathbf{U}, \mathbf{E}, \mathbf{G}\}$. E três relações, os RSVPs R_{UE}, a relação auxiliar R_{UG} e a relação de criação dos eventos pelos grupos R_{GE}. Definimos ainda $0{,}01$ como o peso mínimo da relação alvo $\alpha_{R_{UE}}$ para garantir que a relação alvo seja sempre otimizada.

Consequentemente, o MR-BPR otimiza o MR-BPROpt na seguinte configuração relacional:

$$\begin{aligned} \underset{(\mathbf{U},\mathbf{E},\mathbf{G})}{\operatorname{argmin}} \; \gamma\big(&\alpha_{R_{UE}} \text{BPROpt}(R_{UE}, \hat{s}_{UE}) + \alpha_{R_{UG}} \text{BPROpt}(R_{UG}, \hat{s}_{UG}) \\ &+ \alpha_{R_{GE}} \text{BPROpt}(R_{GE}, \hat{s}_{GE})\big) + \lambda_U \|\mathbf{U}\|^2 + \lambda_E \|\mathbf{E}\|^2 + \lambda_G \|\mathbf{G}\|^2. \end{aligned}$$

Os fatores latentes são então aprendidos via gradiente descendente estócastico (SGD) que

tem provado escalabilidade e rápida taxa de convergência (RENDLE et al., 2009; RENDLE, 2012; KROHN-GRIMBERGHE et al., 2012).

4.4 Seleção dos Modelos Sociais

Avaliamos aqui os algoritmos propostos sobre o conjunto de validação. Inicialmente selecionamos a combinação de pesos que maximiza a $NDCG@10$ para o MR-BPR fixando os hiper-parâmetros em 200 fatores latentes (k), $0{,}1$ de taxa de aprendizado (γ) e 1000 iterações do SGD baseando-nos em nossa experiência com o algoritmo. A Figura 4.4 apresenta essa comparação para a cidade de San Jose. No eixo-x estão os pesos da relação alvo R_{UE} seguida das relações auxiliares R_{UG} e R_{GE}, respectivamente e no eixo-y a $NDCG@10$. Essas combinações foram definidas seguindo o projeto de uma mistura do tipo *simplex-lattice* (CHASALOW; BRAND, 1995), muito utilizada para a definição de compostos em experimentos químicos. Esse tipo de mistura gera combinações de pesos com restrições como a que foi definida na Equação 4.12 maximizando a cobertura das possibilidades.

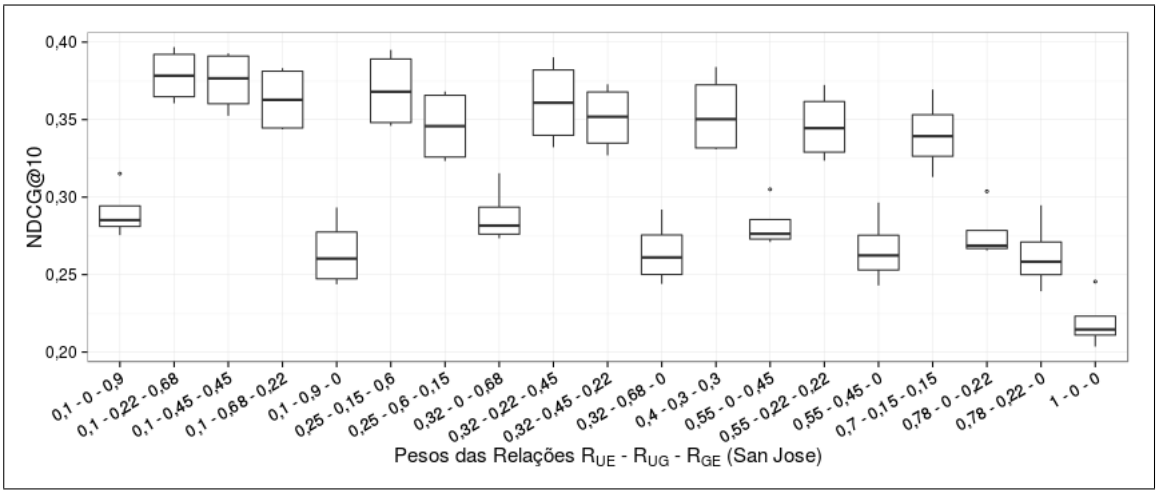


Figura 4.4: MR-BPR com diversas combinações de pesos de suas relações

Vemos que existe um padrão de três níveis de resultados, quando as duas relações auxiliares tem peso 0 (último gráfico de caixa) tem-se a pior $NDCG@10$, quando apenas uma das duas é zerada o resultado melhora, mas ainda não alcança os resultados de quando todas tem peso diferente de zero. A importância da relação de criação dos eventos pelos grupos R_{GE} fica clara ao observarmos que quando ela recebe peso nulo o ranking é pior do que quando

R_{UG} é igual a 0, e também dentre as melhores combinações a $NDCG@10$ é maximizada quando o seu peso é máximo. Esses padrões repetem-se nas demais cidades com pequenas variações na escala dos valores. Assim, a combinação mais promissora em todas as cidades foi que atribuiu peso de $0{,}1$ à relação alvo, $0{,}22$ à relação R_{UG} e $0{,}68$ a R_{GE}. Esse resultado enfatiza que mais importante do que a sinergia entre as relações R_{UE} e R_{UG}, é a terceira relação R_{GE} que associa os grupos aos eventos, fechando o ciclo entre as entidades U, E e G (ver Fig. 4.3)

Com os pesos das relações selecionados, variamos os hiper-parâmetros do MR-BPR (anteriormente fixados) e percebemos que com $k = 300$, $\gamma = 0{,}1$ e 1500 iterações a $NDCG@10$ melhorou e se estabilizou. Selecionamos também os hiper-parâmetros do BPR-MF e do ITEM-KNN com implementações de código livre disponível na biblioteca MyMediaLite (GANTNER et al., 2011)[5]. Para o BPR-MF a combinação de $k = 100$, $\gamma = 0{,}1$ e 3000 iterações obteve os melhores resultados no conjunto de validação. Já o k ótimo para o ITEM-KNN foi de 100 vizinhos nas cidades de Chicago e Phoenix e 25 em San Jose.

A Figura 4.5 apresenta a comparação desses algoritmos para as cidades investigadas nas partições de validação. Vemos que o BPR-MF, que otimiza uma função de ranking, não conseguiu superar o clássico ITEM-KNN, mesmo com 3000 iterações. Ainda assim, o GRUPO-FREQUENTE teve melhores resultados que ambos, mostrando que, apesar da simplicidade, a hipótese da frequência nos grupos é realmente um importante fator de decisão na seleção dos eventos pelos usuários. E, dentre todos o MRBPR aparenta ter aprendido efetivamente o potencial multi-relacional implícito na RSBE obtendo os melhores valores de $NDCG@10$ do contexto social.

Na seleção dos modelos do componente social decidimos manter os dois melhores modelos, GRUPO-FREQUENTE e MR-BPR. Apesar de fazerem parte do mesmo contexto, os sinais não são iguais, como o MR-BPR trata as relações como matrizes binárias, a quantidade de eventos que um usuário foi em dado grupo não é considerada, e o GRUPO-FREQUENTE captura exatamente essa informação.

[5]Mais detalhes ver o Apêndice A.

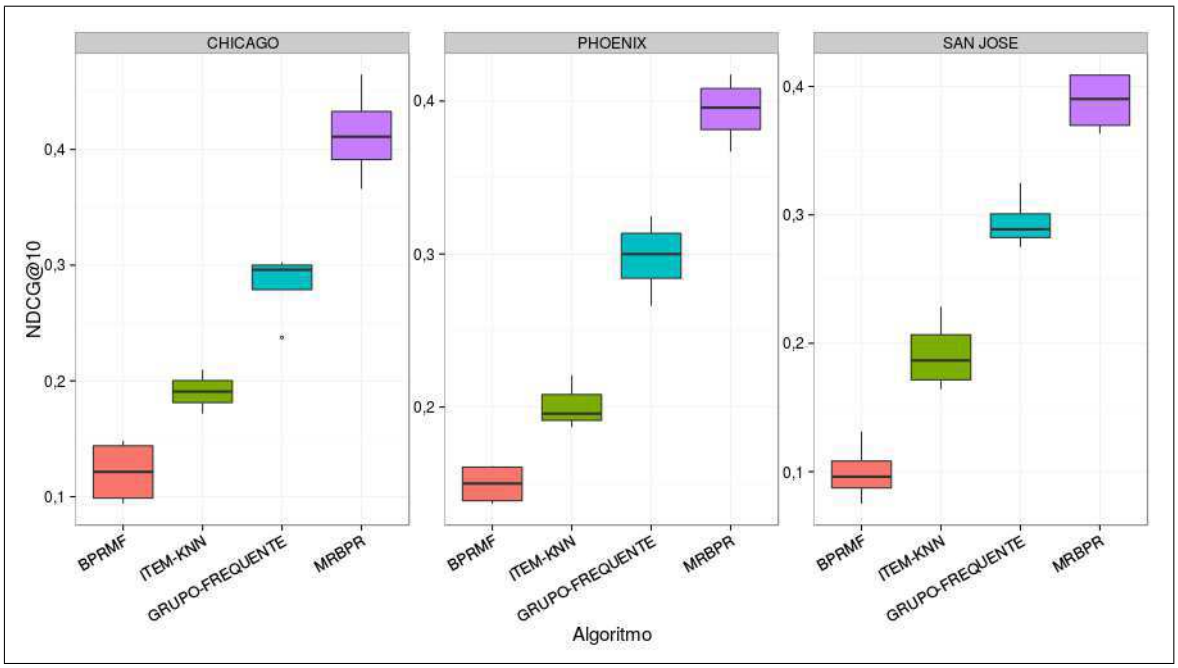


Figura 4.5: Comparação de Algoritmos do Contexto Social

4.5 Análise da Esparsidade

Analisamos ainda as recomendações geradas no experimento anterior separando-as por nível de esparsidade, desta vez apenas para os modelos selecionados. A Figura 4.6 apresenta dois gráficos da esparsidade do evento e do usuário com os respectivos níveis no eixo-x, e no eixo-y o valor da $NDCG@10$ para a cidade de Chicago.

Essa análise permite-nos extrair conclusões interessantes:

- Ambos os algoritmos são capazes de recomendar eventos com certa eficácia mesmo estes não tendo histórico algum de RSVP, i.e. *cold-start*. E no caso do MR-BPR, quanto mais RSVPs mais acurada é a recomendação;

- O MR-BPR vai mais além e recomenda eventos para usuários no *cold-start*. Esse comportamento só foi possível porque o modelo aprendeu a relacionar usuários e eventos pelo caminho alternativo, dos usuários U para os grupos G via R_{UG} e dos grupos G para os eventos E via R_{GE}. Acreditamos ser esta a principal causa do bom resultado em todas as cidades (Fig. 4.5);

- Apesar do GRUPO-FREQUENTE não recomendar para usuários no *cold-start* basta

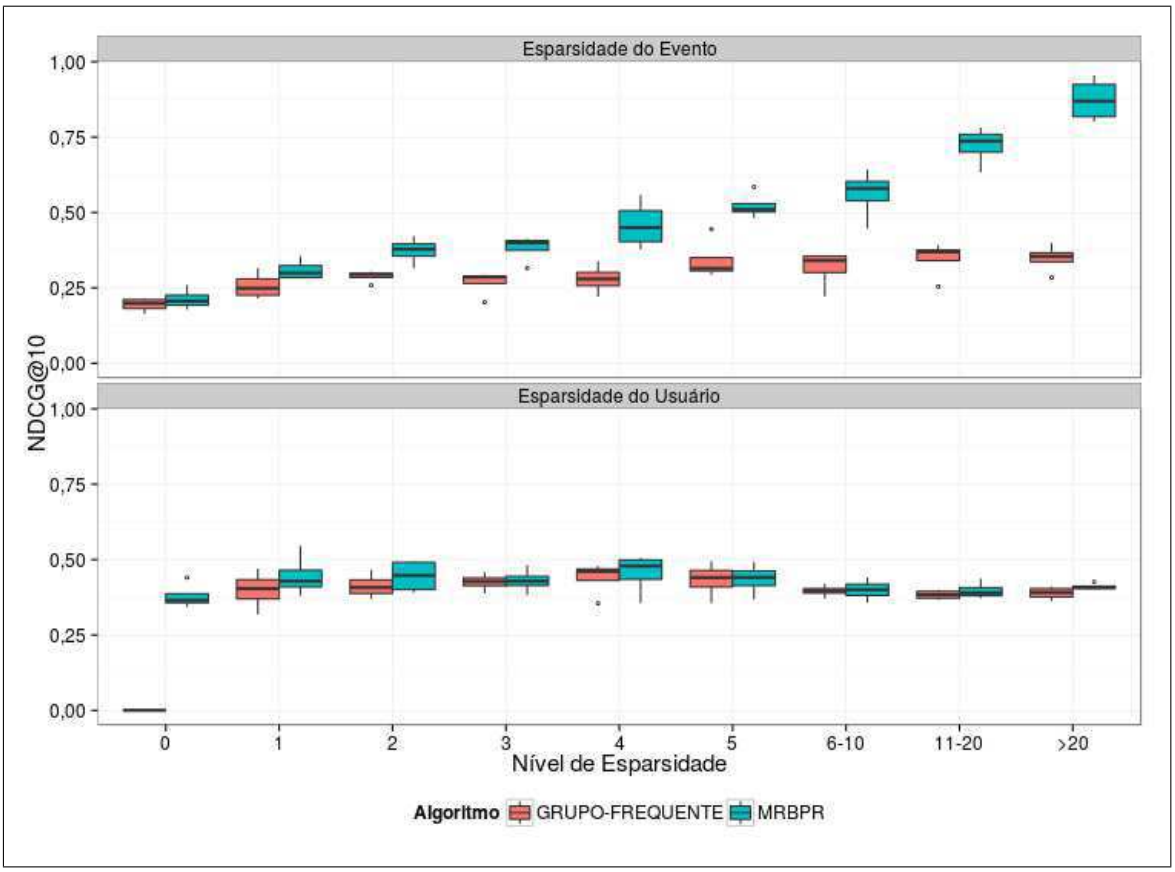


Figura 4.6: Esparsidade do Evento e do Usuário para os Modelos do Contexto Social (Chicago)

que o usuário envie um RSVP para qualquer evento que as recomendações já serão ranqueadas com razoável eficácia.

Capítulo 5

Análise do Conteúdo Textual dos Eventos

A descrição dos eventos é uma importante fonte de informação tipicamente utilizada pelos usuários para conhecer um pouco mais sobre o evento. Apesar de ser um texto escrito em linguagem natural, após pré-processado pode ser muito útil para identificação de padrões de interesse dos eventos, sendo uma alternativa a meta-dados como palavras-chave. O texto é também uma forma de detecção de eventos recorrentes quando a recorrência não está disponível na base de dados (como no caso do Meetup).

Partindo do pressuposto de que os usuários tendem a participar de eventos contendo conteúdo textual similar àqueles participados no passado, definimos um modelo de recomendação especializado no conteúdo textual dos eventos. O modelo aprende as preferências textuais do usuário no treino e recomenda os eventos candidatos do teste com conteúdo mais similar.

Para especificar o modelo organizamos o capítulo da seguinte maneira: inicialmente apresentamos o pré-processamento textual na Seção 5.1, listamos as abordagens de representação do conteúdo na Seção 5.2, em seguida na Seção 5.3 apresentamos os perfis de usuário avaliados, selecionamos a melhor combinação de representação e perfil do usuário na Seção 5.4 e analisamo-la diante dos vários níveis de esparsidade na Seção 5.5.

5.1 Pré-processamento Textual

No Meetup cada evento tem um nome (obrigatório), com poucas palavras, e uma descrição (opcional) sem limites no número de palavras (ver exemplo de Meetup na Figura 5.1).

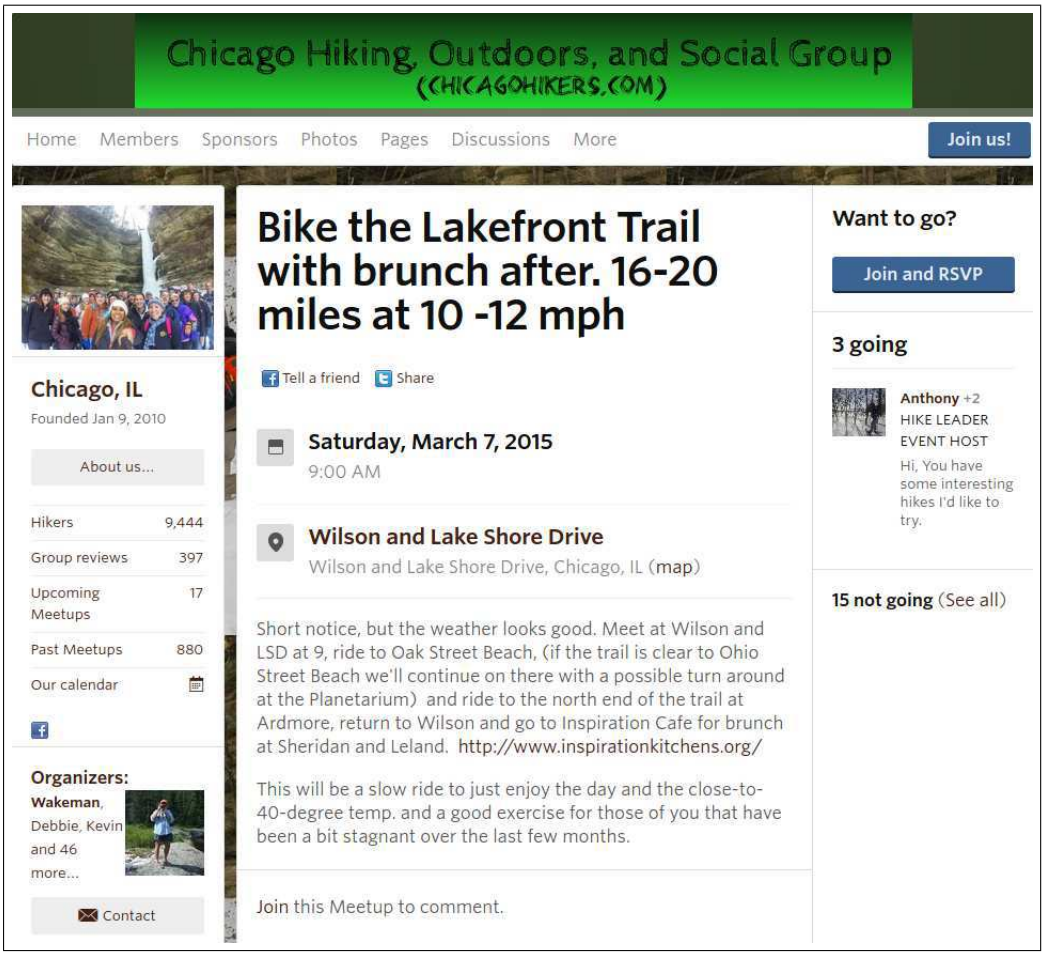


Figura 5.1: Exemplo nome e descrição de um evento na cidade de Chicago

Neste trabalho esse conteúdo foi pré-processado passando pelas seguintes fases sequenciadamente:

1. extração de conteúdo das tags HTML;
2. remoção de acentos (e.g. termos em francês);
3. tradução para ASCII;
4. remoção de pontuação;
5. remoção de *stopwords* do dicionário inglês;
6. redução das palavras aos seus radicais, *stemming* (apenas para os modelos de tópicos na Seção 5.2.2).

Com os conteúdos dos eventos pré-processados, montamos um *corpus* textual dos eventos contidos na partição (treino e teste), e removemos as palavras com menos de 6 ocorrências. Ao final cada evento foi representado como um vetor de termos (aka modelo *bag-of-words*), onde vários esquemas de ponderação dos termos foram usados (ver as seções abaixo).

5.2 Representação do Conteúdo

Para gerarmos a representação textual de cada evento aplicamos as transformações descritas nas seções abaixo. Nesta fase do processamento utilizamos a biblioteca de código livre Gensim (ŘEHŮŘEK; SOJKA, 2010)[1].

5.2.1 Modelo de Vetor de Termos: TFIDF

A representação no formato TFIDF (*Term Frequency - Inverse Document Frequency*), é uma abordagem bastante popular na Recuperação de Informação para representar documentos textuais. Considerando D como o *corpus* de documentos e T como o conjunto dos termos existentes em D. A ideia é representam um documento $d \in D$ (i.e. conteúdo do evento) em um espaço vetorial (SALTON; WONG; YANG, 1975) cujas dimensões especificam os termos $t \in T$ do *corpus* (i.e. no nosso caso os termos extraídos dos nomes e descrições dos eventos). Formalmente temos

$$\vec{d} := (t_1, t_2, \ldots, t_{|T|})$$

Cada palavra tem seu peso $w(t,d,D)$ dado por uma função local ao documento $tf(d,w)$, no caso a sua frequência, multiplicada por outra função global $idf(D,w)$, o inverso da frequência no *corpus*, i.e.

$$w(t,d,D) := tf(d,t) \times idf(t,D) = f(t,d) \times \log_2 \frac{|D|}{f(t,d)} \tag{5.1}$$

onde $f(t,d)$ é a frequência dos termos $t \in T$ no documento $d \in D$.

Por exemplo, considere uma base de dados de eventos como mostrado abaixo, onde cada linha representa o título de um evento específico:

[1] Mais detalhes ver o Apêndice A.

1. São João em Campina Grande, Paraíba;
2. Vaquejada em pleno São João de Campina Grande;
3. Algodão Colorido em pleno Semi-Árido da Paraíba;
4. Concerto de Natal no Teatro Municipal Severino Cabral;
5. Musicas bonitas e muita dança com a escola Passo a Passo no Teatro da cidade.

Após a remoção das pontuações teríamos um modelo estruturado como mostra a Tabela 5.1. Na coluna termos está a lista completa de todos os termos da base, nas demais colunas estão os índices dos eventos e para cada evento o contador de termos e seu valor definido pelo TFIDF.

Podemos ver que os termos que aparecem em um único evento tendem a ter valores maiores nesses eventos, como "algodão" no evento 3 e "municipal" no evento 5. E quando estes termos acontecem mais de uma vez o seu peso tende a ser ainda maior, como no caso do termo "passo" no evento 5. Em contra-partida, os termos que aparecem em vários eventos tem peso relativamente menor, como os termos "em" nos eventos 1, 2, e 3 e do termo "no" nos eventos 4 e 5.

5.2.2 Modelos de Tópicos: LSI e LDA

Avaliamos também as representações de tópicos providas pelo LSI (DEERWESTER et al., 1990), ou *Latent Semantic Indexing*, e LDA (BLEI; NG; JORDAN, 2003), ou *Latent Dirichlet Allocation*

O LSI baseia-se na hipótese de que termos usados nos mesmos contextos tendem a ter significados similares. A técnica SVD, ou *Singular Value Decomposition* da álgebra linear, é aplicada para fatorar a matriz esparsa de termos e documentos $T \times D$ em matrizes densas menores, $T \times k$ e $D \times k$. As k colunas representam os tópicos mais relevantes compartilhados pelos termos $t \in T$ e pelos documentos $d \in D$, alcançando alta compressão dos dados além de capturar fatores inerentes a linguagem natural como a sinonímia e a polissemia.

Já o LDA é uma evolução do modelo pLSI (HOFMANN, 2001) (*Probabilistic Latent Semantic Indexing*) o qual é uma classe de modelo probabilístico gerador para realizar decomposições de misturas probabilísticas. O pLSI explica a existência de conjuntos de ob-

Termos	**Evento 1**		**Evento 2**		**Evento 3**		**Evento 4**		**Evento 5**	
campina	1	0,4339	1	0,3262						
em	1	0,2419	1	0,1819	1	0,1456				
grande	1	0,4339	1	0,3262						
joao	1	0,4339	1	0,3262						
paraiba	1	0,4339			1	0,2613				
sao	1	0,4339	1	0,3262						
de			1	0,3262			1	0,233		
pleno			1	0,3262	1	0,2613				
vaquejada			1	0,573						
algodao					1	0,4589				
colorido					1	0,4589				
da					1	0,4589				
semi-arido					1	0,4589				
cabral							1	0,4092		
concerto							1	0,4092		
municipal							1	0,4092		
natal							1	0,4092		
no							1	0,233	1	0,1745
severino							1	0,4092		
teatro							1	0,233	1	0,1745
bonitas									1	0,3065
com									1	0,3065
danca									1	0,3065
escola									1	0,3065
muita									1	0,3065
musicas									1	0,3065
passo									2	0,6129

Tabela 5.1: Exemplo de Eventos e seus valores de TFIDF

servações em termos de tópicos não observados que, por sua vez especificam o por quê determinados conjuntos de dados são similares. Por exemplo, no contexto de termos e documentos, ele especifica que um documento é uma mistura de tópicos e que a criação de cada palavra é atribuível a um tópico. O LDA evolui o pLSI assumindo que os tópicos são amostrados da distribuição *Dirichlet* sobre os documentos.

Durante a avaliação dessas representações o LSI obteve melhores resultados quando processou os documentos representados no formato TFIDF, já o LDA foi melhor usando o *bag-of-words* original.

Repetimos o exemplo da sub-seção anterior para um modelo LSI com três tópicos latentes. A Tabela 5.2 mostra os tópicos identificados com seus pesos, e para cada um as dez

Tópico 1 (1,291)		**Tópico 2 (1,040)**		**Tópico 3 (0,976)**	
campina	0,4	passo	0,411	semi-arido	-0,426
sao	0,4	severino	0,277	algodao	-0,426
grande	0,4	concerto	0,277	colorido	-0,426
joao	0,4	municipal	0,277	da	-0,426
vaquejada	0,301	natal	0,277	passo	-0,186
paraiba	0,283	cabral	0,277	paraiba	-0,186
em	0,253	no	0,275	pleno	-0,172
pleno	0,225	teatro	0,275	sao	0,128
de	0,186	bonitas	0,206	joao	0,128
semi-arido	0,095	com	0,206	grande	0,128

Tabela 5.2: Tópicos aprendidos pelo LSI no exemplo

	Evento 1	**Evento 2**	**Evento 3**	**Evento 4**	**Evento 5**
Tópico 1	0,2137	**2,566**	1,1391	0,3411	0,0538
Tópico 2	-0,1734	0,0309	-0,2787	**2,1003**	**2,607**
Tópico 3	**0,2604**	0,509	**-2,1265**	0,4465	-0,9543

Tabela 5.3: Relevância de Tópicos para os Eventos

palavras mais importantes juntamente com suas relevâncias. Vemos que o modelo soube separar bem os tópicos latentes que explicam os eventos, permitindo dessa forma a seguinte interpretação: o primeiro diz respeito à festas campinenses, o segundo à eventos musicais e artísticos, e o terceiro mistura-se um pouco com o primeiro mas dá maior ênfase ao algodão. O peso dos tópicos indica a sua coesão, e de fato, o primeiro é o mais bem definido e o terceiro é o menos coeso.

A Tabela 5.3 apresenta as relevâncias de cada tópico para os eventos. Analisando os maiores valores de cada evento (em negrito) vemos que os eventos 4 e 5 foram corretamente atribuídos para o tópico 2 e o evento 2 para o tópico de festas campinenses. Já os eventos 1 e 3 aparentam uma certa confusão, mesmo assim vemos que o modelo começou a identificar um relacionamento global gerado pelo termo “paraiba” que possui uma carga polissêmica relativa à localização. A inversão dos seus pesos mostra que eles estão em dimensões opostas do mesmo fator latente, além disso, o evento 1 tem um peso relativamente alto no tópico 1.

5.3 Perfil Textual do Usuário

Cada usuário $u \in U$ tem um histórico de eventos os quais participou (denotado por $e \in E_u$), cada evento $e \in E$ tem sua descrição textual $D_e \subseteq D$ representado em um dos modelos supracitados (e.g. para o modelo TFIDF cada evento é representado pelo conjunto de termos e seus valores de TFIDF). Especificamos assim a preferência textual do usuário pelos conteúdos de seus eventos do passado. Considerando a função $\lambda_m(D_e)$ como a representação textual gerada pelo modelo m sobre o documento D_e, formalizamos o perfil textual do usuário da seguinte forma:

$$PerfilTextual(u) := \sum_{e \in E_u} PesoEvento(u,e) \times \lambda_m(D_e) \tag{5.2}$$

onde $PesoEvento(u,e)$ é uma função que quantifica a preferência do usuário u sobre o evento e.

Propomos três perfis alternativos variando a função $PesoEvento$. O primeiro, $PerfilConstante$, atribui peso constante a todos os eventos, resultando em um perfil textual que soma as representações textuais dos eventos do passado.

$$PerfilConstante(u) := \sum_{e \in E_u} 1 \times \lambda_m(D_e) \tag{5.3}$$

A segunda função, $PerfilTemporal$, dá maior valor aos eventos mais próximos ao dia da recomendação (momento da partição treino e teste). Ela atribui um decaimento temporal baseado na teoria do valor temporal do dinheiro (GALLAGHER; ANDREW, 1968). Uma abordagem similar foi utilizada em (SANDHOLM; UNG, 2011), mas no domínio de recomendação de conteúdo Web. A sua fórmula com o decaimento é dada abaixo

$$PerfilTemporal(u) := \sum_{e \in E_u} \frac{1}{(1+\alpha)^{\tau(e)}} \times \lambda_m(D_e) \tag{5.4}$$

onde α é a taxa de decaimento e $\tau(e)$ retorna o número de dias desde o envio do RSVP para o evento e até o momento da recomendação. A terceira função, $PesoPopularidade$, por outro lado, penaliza eventos muito populares (i.e. com muitos RSVPs no passado) partindo da hipótese de que quanto menor o evento, maior a relação interpessoal entre seus usuários (XU;

	Evento 1	**Evento 2**	**Evento 3**
$\tau(e)$	200	60	15
$\|U_e\|$	1	5	20

Tabela 5.4: Características dos Eventos

CHIN; COSLEY, 2013). A qual é formalizada a seguir

$$PerfilPopularidade(u) := \sum_{e \in E_u} \frac{1}{\log_2 |U_e|} \times \lambda_m(D_e) \tag{5.5}$$

onde U_e é o conjunto de usuários que enviaram um RSVP positivo para e.

Assim, o treino do modelo de recomendação baseado em conteúdo acontece da seguinte forma: representamos todos os eventos do passado no devido formato (e.g. LSI) e depois agregamos os eventos do passado de cada usuário de acordo com o perfil textual (e.g. $PerfilTemporal$).

Para gerar a lista de recomendação calculamos a similaridade do cosseno (Eq. 4.2) entre o perfil do usuário com todos os eventos candidatos e selecionamos os Top-n.

Ilustramos a definição dos perfis textuais de um usuário fictício u. Considerando que u participou dos eventos 1, 2, e 3 expostos no exemplo da seção anterior, e que estes eventos tem suas características definidas na Tabela 5.4, sendo o evento 3 o mais antigo e também o maior dentre eles. Representamos os eventos com o TFIDF e geramos os perfis textuais de u como mostra a Tabela 5.5. As primeiras colunas correspondem aos contadores dos termos por evento já expostas na Tabela 5.1 e as colunas finais os pesos de cada termo para cada perfil textual. O $PerfilConstante$ teve maior peso para os termos mais frequentes, o $PerfilTemporal$ atribuiu maior peso para o evento 3 que é o mais recente dentre os três, e o $PerfilPopularidade$ considerou os termos do evento 1 mais importantes pois este foi o com menor número de participantes.

5.4 Seleção do Modelo de Conteúdo

O modelo de conteúdo foi selecionado dentre as variações de representação textual e perfil textual de usuário mencionadas anteriormente com as partições do conjunto de validação.

Realizamos uma análise da sensibilidade das representações textuais para com a taxa de

Termos	**Ev. 1**	**Ev. 2**	**Ev. 3**	*Constante*	*Temporal*	*Popularidade*
campina	1	1		**0,3042**	0,2152	**0,3721**
em	1	1	1	0,2544	0,2323	0,2618
grande	1	1		**0,3042**	0,2152	**0,3721**
joao	1	1		**0,3042**	0,2152	**0,3721**
paraiba	1		1	**0,3042**	0,288	0,3147
sao	1	1		**0,3042**	0,2152	**0,3721**
de		1		0,1521	0,1288	0,1548
pleno		1	1	**0,3042**	0,3302	0,2522
vaquejada		1		0,2672	0,2262	0,2719
algodao			1	0,2672	**0,3539**	0,1712
colorido			1	0,2672	**0,3539**	0,1712
da			1	0,2672	**0,3539**	0,1712
semi-arido			1	0,2672	**0,3539**	0,1712

Tabela 5.5: Perfis Textuais do usuário u

decaimento temporal α do $PerfilTemporal$ de $0{,}005$ até $0{,}05$. Enquanto que um decaimento de $0{,}005$ leva mais de 500 dias para alcançar o 0 absoluto, um α de $0{,}01$ o faz em 300 dias e de $0{,}05$ em pouco mais de 50 dias. A Figura 5.2 mostra essa variação para duas das cidades investigadas. Percebemos que o padrão é similar na maioria dos casos, o aumento da taxa de decaimento leva a um resultado inferior. No entanto, em Phoenix o α de $0{,}01$ tem resultados marginalmente melhores. Em San Jose, o padrão é o mesmo de Chicago e o α é definido em $0{,}005$.

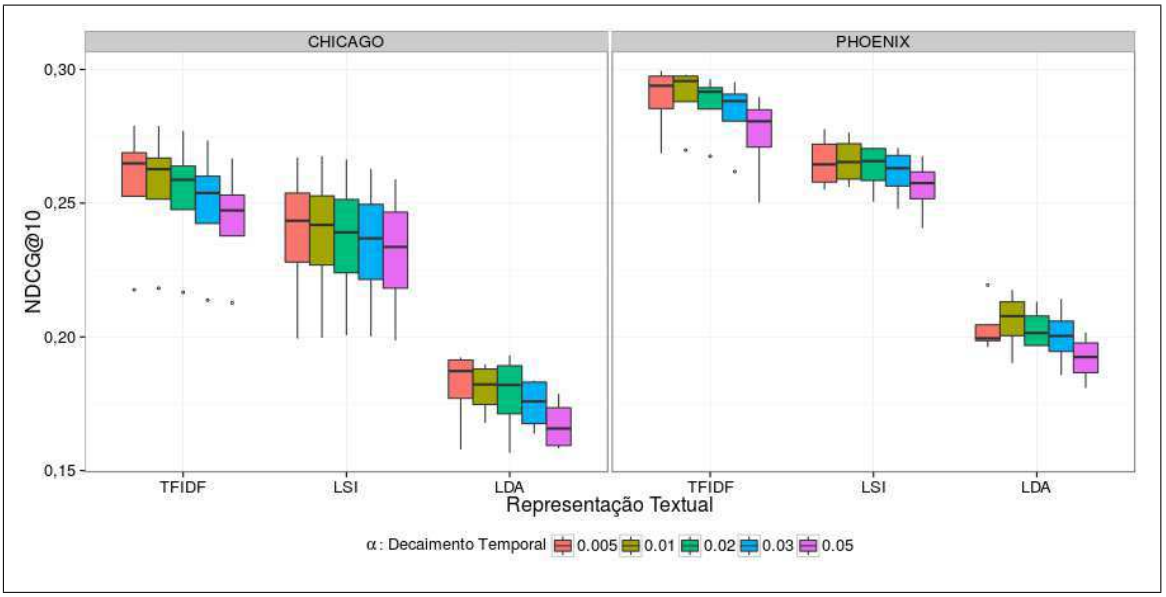


Figura 5.2: Comparação da Taxa de Decaimento Temporal (α)

Realizamos também experimentos sobre as partições de validação para selecionar os hiper-parâmetros das representações textuais LSI e LDA. Ambos os modelos obtiveram os melhores resultados representando os eventos em 250 tópicos latentes, contra as representações de 50 e 100 fatores, o que demonstra a grande variedade de eventos do Meetup. Para o LDA selecionamos ainda a quantidade de passagens sobre o *corpus* textual como 10 (em comparação com 1, 5 e 20 passagens) e o número de iterações internas como 250 (em comparação com 50, 100 e 500 iterações).

Com essas representações comparamos nove variações de modelos de recomendação de evento baseado no conteúdo pela combinação das três representações com os três perfis de usuário. Na Figura 5.3 vemos que o LSI e o LDA foram consistentemente inferiores ao TFIDF. E para o TFIDF, o $PerfilTemporal$ obteve resultados marginalmente melhores do que os demais, sendo esta a combinação escolhida para modelar o conteúdo textual dos eventos.

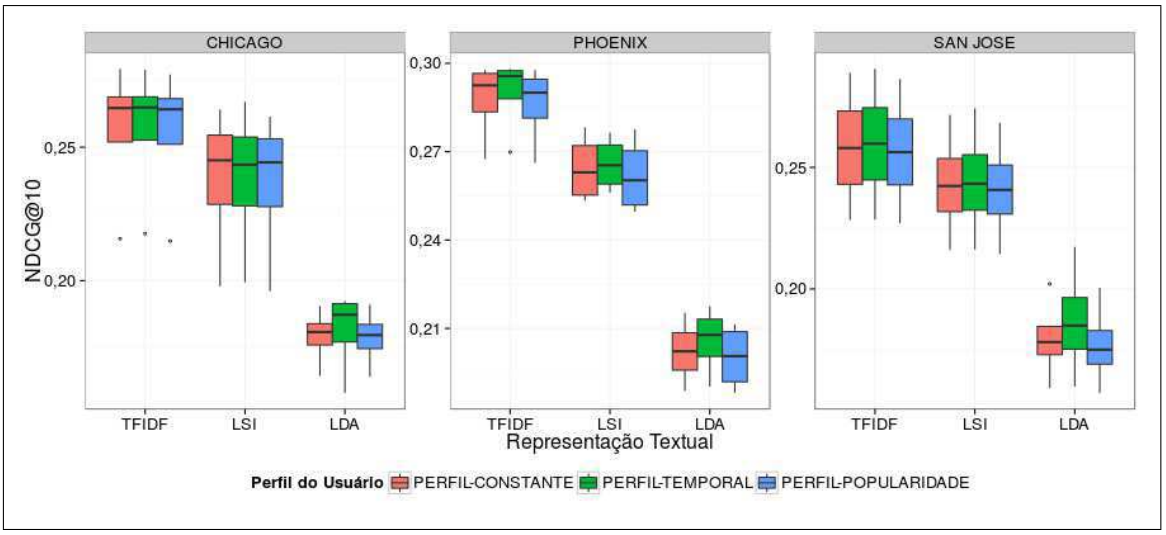


Figura 5.3: Comparação dos Algoritmos de Conteúdo com todas as Representações Textuais e Perfis de Usuário

5.5 Análise da Esparsidade

O modelo selecionado (i.e. representação com TFIDF e $PerfilTemporal$ dos usuários) foi também analisado em termos da esparsidade do usuário e do evento na Figura 5.4. A principal característica do modelo é sua capacidade de recomendar mesmo no *cold-start*

do evento, já que depende primordialmente da descrição textual dos eventos, informação existente desde a sua criação. No entanto, por depender do histórico de RSVPs do usuário não é capaz de recomendar eventos para usuários no *cold-start*.

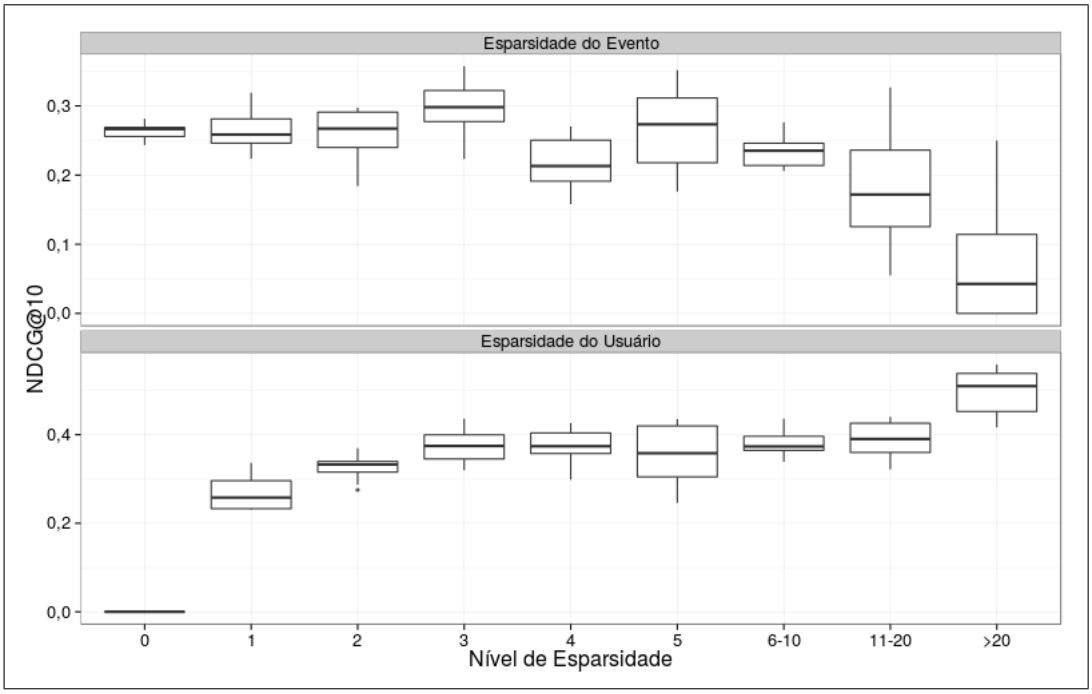


Figura 5.4: Análise da Esparsidade do Evento e do Usuário

Outro fato digno de observação do modelo proposto é sua eficácia inferior para eventos grandes (> 20 usuários), provavelmente a descrição textual dos eventos maiores tem menor valor do que outros contextos como o social e o geográfico. Essa análise valoriza a necessidade da hibridização dos contextos para que as desvantagens de uns sejam balanceadas com as vantagens dos demais.

Capítulo 6

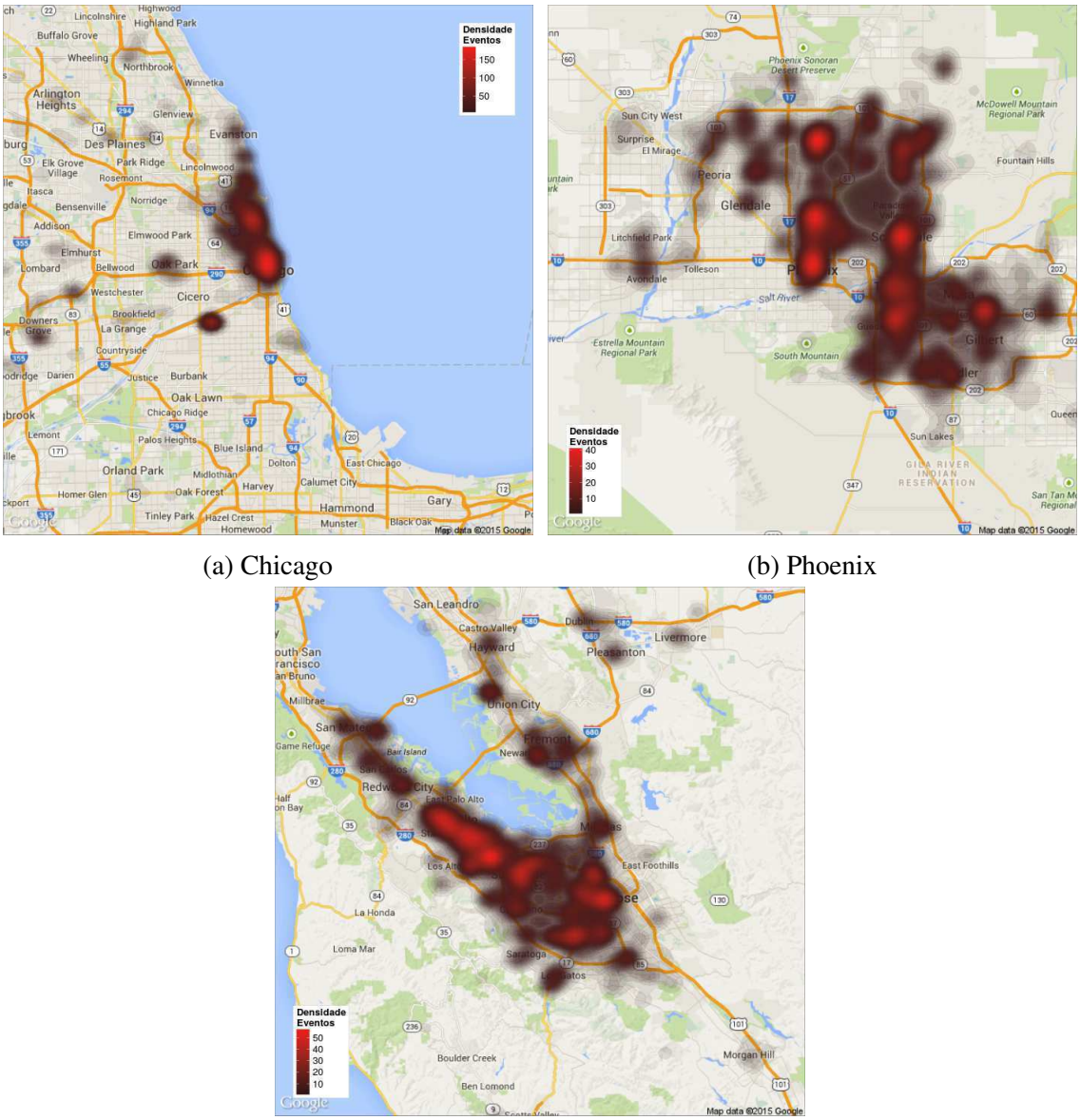
Análise das Preferências Geográficas e Temporais dos Usuários

Nossas escolhas e decisões na vida são carregadas de um significado interior que nos caracteriza, deixando nossas digitais nas nossas ações. Em RSBEs podemos dizer que a cada evento que alguém participa ele deixa para trás a sua pegada no tempo e no espaço. Partindo desse pressuposto definimos neste capítulo três modelos que capturam as preferências geográficas (Seção 6.1) e um modelo temporal do usuário em relação aos eventos do passado (Seção 6.2).

6.1 Contexto Geográfico

Observando a geografia das diversas cidades de um país com ampla extensão territorial como o EUA, observamos uma enorme diversidade de relevos, vegetações, áreas urbanas e áreas rurais. Cada cidade oferece condições geográficas diferenciadas para a criação de eventos. Apresentamos o mapa das cidades investigadas sob a mesma escala geográfica (i.e. mesma distância de zoom) na Figura 6.1 com a densidade de eventos em cada região.

De início observamos que as cidades estão localizadas em regiões completamente distintas do EUA. Chicago (Fig. 6.1a) localiza-se no estado de Illinois norte do país às margens do Lago Michigan, Phoenix (Fig. 6.1b) por outro lado está localizada em pleno deserto de Sonora uma das regiões mais secas do estado do Arizona, e San Jose (Fig. 6.1c) localiza-se no extremo oeste do país próximo da cidade de San Francisco capital do estado da Califór-



(a) Chicago

(b) Phoenix

(c) San Jose

Figura 6.1: Mapa das Cidades com a Densidade Geográfica dos Eventos

nia. Vemos também que a distribuição geográfica dos eventos em San Jose e Phoenix é mais espaçada abrangendo cidades menores da região metropolitana, enquanto que na cidade de Chicago a maioria dos eventos estão centralizadas na própria cidade, com regiões contendo mais de 150 eventos. Essas características refletem-se diretamente da distância que o usuário costuma percorrer para participar dos eventos.

Figura 6.2 apresenta a distribuição acumulada das distâncias entre a casa dos usuários e os eventos que eles participaram para as três cidades. Vemos que os usuários da cidade de Chicago (em vermelho), de fato, tendem a percorrer distâncias menores do que nas demais

cidades.

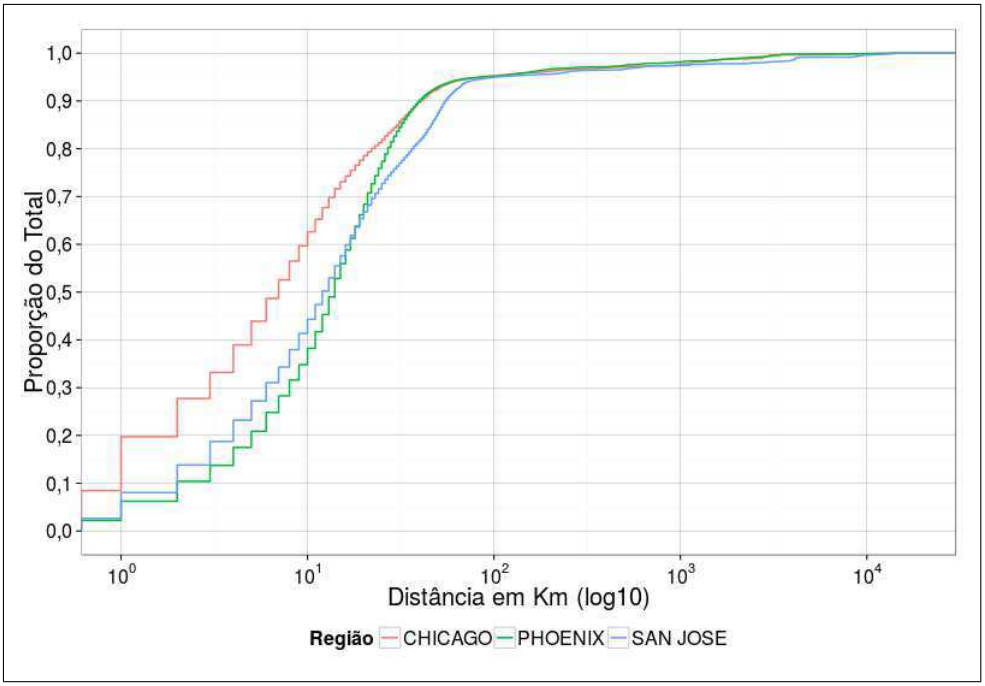


Figura 6.2: Função de Distribuição Acumulada da Distância Usuário-Evento

Analisar essas distribuições geográficas e de distância dos eventos por cidade permite-nos conhecer o padrão da população, por exemplo, em cidades mais urbanizadas provavelmente haverá mais eventos nos bairros centrais, enquanto que em cidades mais rurais e com acesso à natureza haverão mais eventos ao ar livre. Mesmo assim, sabendo o padrão da cidade, ao estudarmos as preferências de distância e geográficas dos usuários veremos que essas distribuições variam fortemente.

Com o objetivo de modelar os diferentes padrões geográficos dos usuários definimos três modelos de recomendação de eventos na sub-seção 6.1.1, e na sub-seção 6.1.2 analisamos e selecionamos àquele com maior potencial de ranking nas partições do conjunto de validação.

6.1.1 Modelos Geográficos

Modelo de Proximidade

O modelo de proximidade, referenciado posteriormente como MAIS-PRÓXIMO, assume a hipótese de que os eventos mais próximos da casa do usuário são mais apelativos do que aqueles mais distantes. Apesar da simplicidade dessa hipótese, ela é interessante e não de-

pende do histórico de RSVPs do usuário podendo portanto ser usada em cenários de *cold-start*.

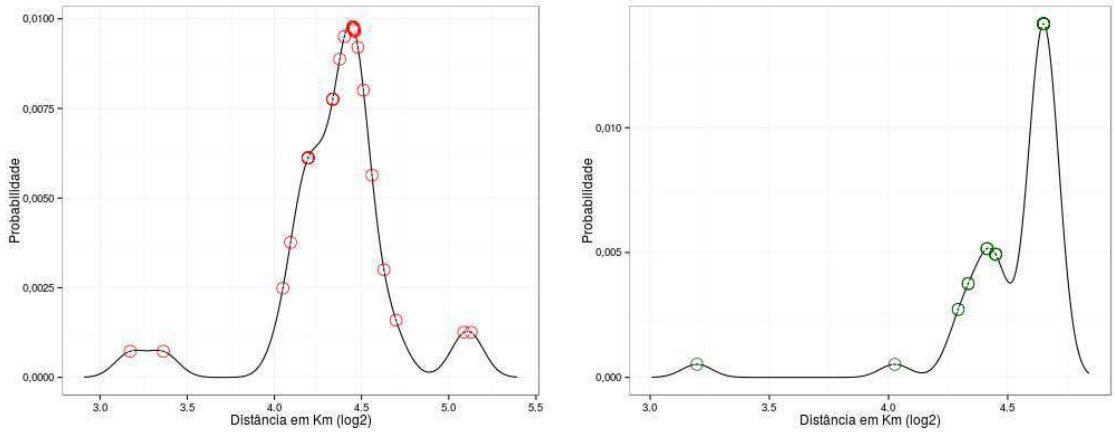
Formalizamos então a função de ranking desse modelo como segue:

$$\hat{s}_{\text{MAIS-PRÓXIMO}}(u, e) := \frac{1}{log_2(\text{dist}(\vec{l_u},\vec{l_e}) + 2))}, \tag{6.1}$$

onde a função dist$(\vec{x},\vec{y})$ retorna a distância geodésica entre os dois pontos no mapa $\vec{x}$ e $\vec{y}$ representados como vetores bidimensionais (i.e. latitude e longitude) e $\vec{l_u}$ e $\vec{l_e}$ representam os vetores com a localização da casa do usuário u e do local que o evento e aconteceu, respectivamente. O modelo aplica ainda transformação logarítmica na distância para reduzir a sua escala e diminuir o viés da distribuição (ver Fig. 6.2).

Modelo de Distância Personalizada

Para modelar os diferentes padrões de mobilidade dos usuários ao participarem de eventos, utilizamos um estimador de densidade por núcleo (*kernel density estimator* em inglês) para personalizar a influência geográfica como uma distribuição das distâncias. A Figura 6.3 ilustra essa diferenciação pela distribuição da distância usuário-evento de dois usuários, onde o eixo-x representa a distância da casa do usuário para os eventos que ele participou e o eixo-y é a distribuição de probabilidade (estimada com um núcleo gaussiano, ver abaixo) onde cada ponto representa um evento.



(a) Usuário 1 (b) Usuário 2

Figura 6.3: Distribuição da Distância Usuário-Evento

O estimador de densidade por núcleo modela as preferências geográficas do usuário baseando-se na hipótese de que os usuários tendem a ir a eventos no futuro com distâncias similares às distâncias dos eventos que participou no passado. Uma abordagem similar foi usada em (ZHANG; CHOW, 2013) mas para o domínio de redes sociais baseadas em localização (LSBNs). Mais formalmente, definimos D_u como uma amostra do logaritmo das distâncias do usuário u para os seus eventos E_u do passado, i.e.

$$D_u := \bigcup_{e \in E_u} log_2(\text{dist}(\vec{l_u}, \vec{l_e}) + 2), \tag{6.2}$$

no qual o logaritmo transforma a distribuição de distâncias aproximando-o de uma gaussiana, distribuição do modelo proposto e também utilizada por Zhang e Chow (2013).

Assumindo que D_u vem de uma distribuição desconhecida f, definimos a probabilidade de uma distância d qualquer por meio de uma função de densidade com núcleo gaussiano $\hat{f}$ sobre D_u:

$$\hat{f}_h(d) := \frac{1}{|D_u|} \sum_{d' \in D_u} K_h(d - d'), \tag{6.3}$$

onde h é um parâmetro de suavização, chamado de tamanho de banda, calculado pela *Silverman's rule of thumb* (SILVERMAN, 1986) para estimação de tamanho de banda sobre a variância de D. Sendo $K_h(.)$ a função de núcleo gaussiano univariada:

$$K_h(x) := \frac{1}{h\sqrt{2\pi}} \epsilon^{-\frac{x^2}{2h}}. \tag{6.4}$$

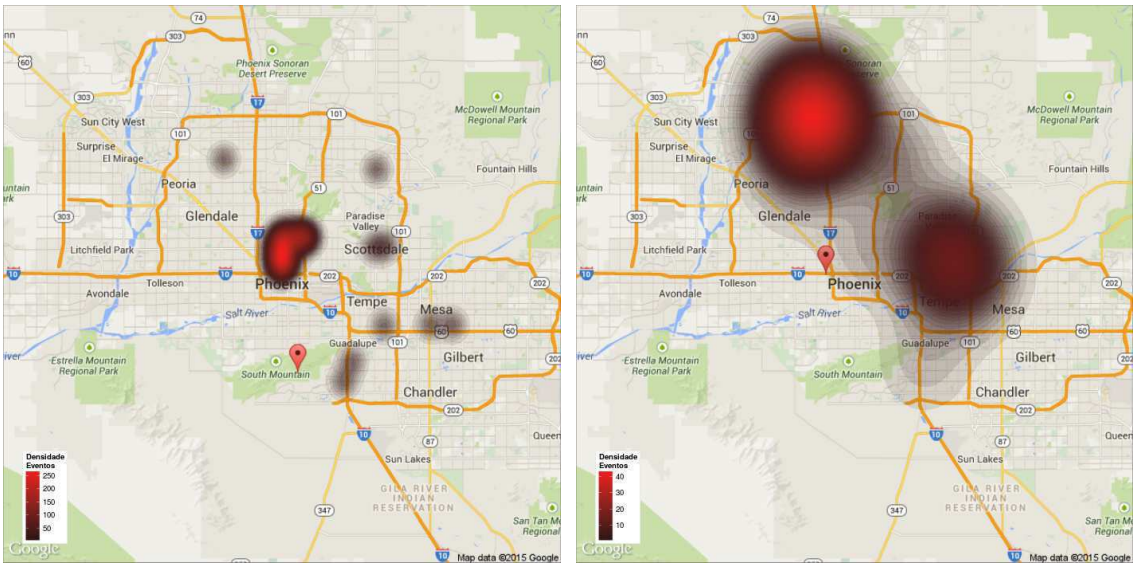
Por meio da distribuição estimada, o modelo DIST-KERNEL ranqueia os eventos candidatos $e \in E_{teste}$ de acordo com a probabilidade de suas distâncias para a casa do usuário u como segue

$$\hat{s}_{\text{DIST-KERNEL}}(u,e) := \hat{f}_h(log_2(\text{dist}(\vec{l_u},\vec{l_e}) + 2)). \tag{6.5}$$

Como esse modelo depende dos RSVPs passados do usuário para gerar recomendações, quando esses dados não existe para um dado usuário u ranqueamos os eventos com o modelo MAIS-PRÓXIMO.

Modelo de Coordenadas Personalizadas

Definimos por último um modelo mais sofisticado que captura não apenas os padrões de distância do usuário para seus eventos do passado, mas aprende a distribuição geográfica desses eventos. Vemos na Figura 6.4 a densidade geográfica dos eventos estimadas para os mesmos usuários da Figura 6.3. A casa dos usuários está apontada com o marcador em vermelho. Enquanto que o modelo DIST-KERNEL irá ranquear todos os eventos a um mesmo raio de distância igualmente, valorando igualmente eventos próximos aos eventos do passado (distribuição em vermelho na Fig. 6.4) e eventos próximos as regiões montanhosas do canto inferior esquerdo do mapa. Já o modelo GEO-KERNEL será muito mais seletivo, filtrando apenas os eventos em regiões do mapa próximas a dos eventos do passado.



(a) Usuário 1 (b) Usuário 2

Figura 6.4: Densidade Geográfica dos Eventos do Usuário

Para gerar as recomendações dos eventos utilizamos novamente um estimador de densidade por núcleo para modelar as preferências geográficas, seguindo a hipótese de que usuários tendem a ir a eventos no futuro em regiões próximas às regiões dos eventos que ele participou no passado. Formalizando, temos L_u como a amostra de todas as coordenadas geográficas dos eventos E_u que o usuário $u \in U$ participou no passado, i.e.

$$L_u := \bigcup_{e \in E_u} \mathbf{l_e}. \tag{6.6}$$

Assumindo que L_u advém de uma distribuição desconhecida g, definimos a probabilidade de uma localização l qualquer por meio de uma função de densidade com núcleo gaussiano bivariado $\hat{g}$ sobre L_u:

$$\hat{g}_{\mathbf{H}}(\mathbf{l}) := \frac{1}{|L_u|} \sum_{\mathbf{l}' \in L_u} K_{\mathbf{H}}(\mathbf{l} - \mathbf{l}') \tag{6.7}$$

onde $\mathbf{H}$ é uma matriz 2×2 simétrica e positiva que representa o tamanho de banda definida como $\mathbf{H} = (h_1,h_2) \times \mathbb{I}$, e $K_H(.)$ é função de núcleo gaussiano bivariado:

$$K_{\mathbf{H}}(\mathbf{x}) := \frac{1}{\sqrt{2\pi|\mathbf{H}|}} \epsilon^{-\frac{\mathbf{x}\mathbf{x}^\top}{2\sqrt{\mathbf{H}}}}. \tag{6.8}$$

Esse modelo gaussiano captura a intuição de que eventos futuros próximos aos locais dos eventos do passado devem receber maior peso do que os demais. A preferência geográfica de um usuário u é então formada pela soma das distribuições gaussianas com centro em l_e, para todo $e \in E_u$.

Com a função estimada para o usuário u, o modelo GEO-KERNEL ranqueia os eventos candidatos $e \in E_{teste}$ de acordo com a probabilidade de suas localizações fazerem parte das preferências geográficas do usuário u, como especificado abaixo

$$\hat{s}_{\text{GEO-KERNEL}}(u,e) := \hat{g}(\vec{l_e}). \tag{6.9}$$

Para aqueles usuários $u \in U$ que não possuem eventos no passado (i.e. $E_u = \emptyset$) recomendamos os eventos mais próximos da casa de u (i.e. modelo MAIS-PRÓXIMO).

6.1.2 Seleção do Modelo Geográfico

O modelo GEO-KERNEL possui então dois hiper-parâmetros h_1 e h_2, que especificam o tamanho de banda em cada uma das dimensões da função gaussiana, quanto maior mais larga é a distribuição naquela dimensão. Decidimos que ambos teriam o mesmo valor $h_1 = h_2 = h$ assim a distribuição seria simétrica, portanto, quanto maior o h mais esparsa será a distribuição estimada.

Definimos o valor de h experimentalmente sobre o conjunto de validação. Inicialmente imaginamos que trata-se de uma variável sem unidade, mas a sua escala está diretamente

relacionada com a escala das coordenadas geográficas. Além disso, é de se esperar que seus valores variem de acordo com a geografia dos eventos de cada cidade. A Figura 6.5 mostra os resultados para cada uma das cidades, e podemos ver que a $NDCG@10$ das cidades de Phoenix e San Jose estabilizam com valores maiores do que da cidade de Chicago, mas como selecionamos o h que alcançou a melhor $NDCG@10$, definimos $h = 0{,}001$ para Chicago e $h = 0{,}00075$ para as demais cidades.

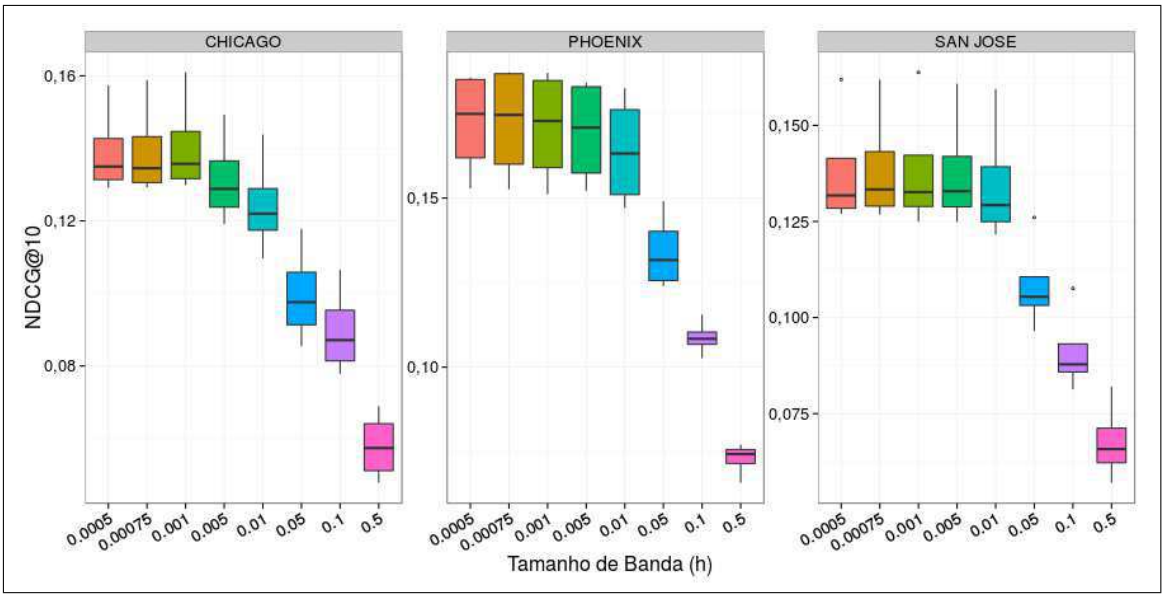


Figura 6.5: Seleção do Tamanho de Banda (h) para o modelo GEO-KERNEL

A melhor configuração do modelo GEO-KERNEL foi comparada com os demais modelos geográficos, o DIST-KERNEL, que aprende as preferências de distância do usuário, e o modelo MAIS-PRÓXIMO que recomenda sempre os eventos mais próximos à casa do usuário. Vemos na Figura 6.6 que modelar as preferências geográficas é, de fato, mais promissor do que simplesmente as distâncias percorridas.

Comparamos ainda os modelos propostos sob o ponto de vista da esparsidade do evento e do usuário. A Figura 6.7 compara-os nas partições de validação para a cidade de San Jose, da qual extraímos os seguintes análises:

- Os modelos conseguem recomendar eventos em quaisquer situações, tanto no *cold-start* do evento quanto do usuário. Quando o evento não tem RSVPs no passado, a informação geográfica dos eventos do usuário permitem que o DIST-KERNEL e GEO-KERNEL recomendem com eficácia considerável. Já quando o usuário não participou

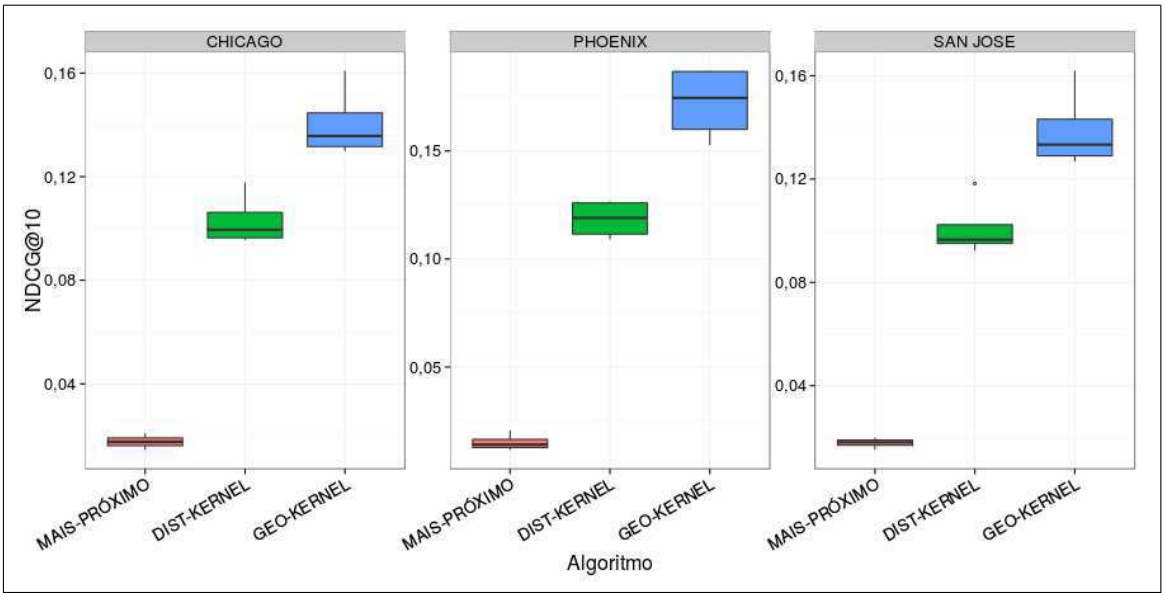


Figura 6.6: Comparação dos Modelos Geográficos

de evento algum (i.e. não tem histórico de RSVPs positivos) todos os modelos imitam o comportamento do MAIS-PRÓXIMO, recomendando os eventos mais próximos da casa do usuário;

- O aumento do número de RSVPs do GEO-KERNEL não traz tantas melhorias, até aparenta diminuir quando os eventos tem entre 5 e 20 RSVPs. Uma possível causa para esse comportamento é o fato de que com poucos RSVPs o modelo já consegue identificar a "assinatura" geográfica do usuário;

- Já o aumento do histórico de RSVPs do usuário é muito benéfico para o GEO-KERNEL, pois o perfil geográfico do usuário torna-se mais abrangente. Enquanto isso, o DIST-KERNEL tem uma queda na $NDCG@10$ resultado do possível alargamento da distribuição de distâncias do usuário, o que diminui seu potencial de seleção dos eventos.

6.2 Contexto Temporal

Nós, seres humanos, na realização de nossas atribuições costumeiras de trabalho, de alimentação, como também nos lazeres mais diversos, costumamos definir rotinas, ou seja,

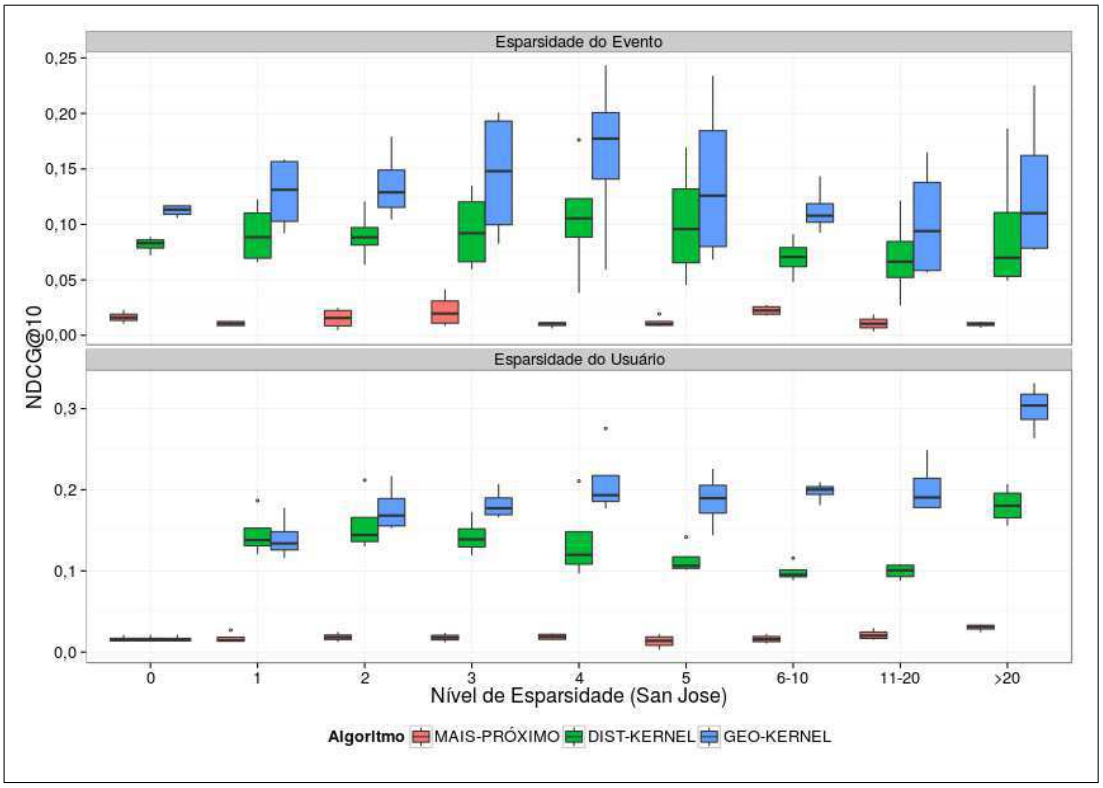
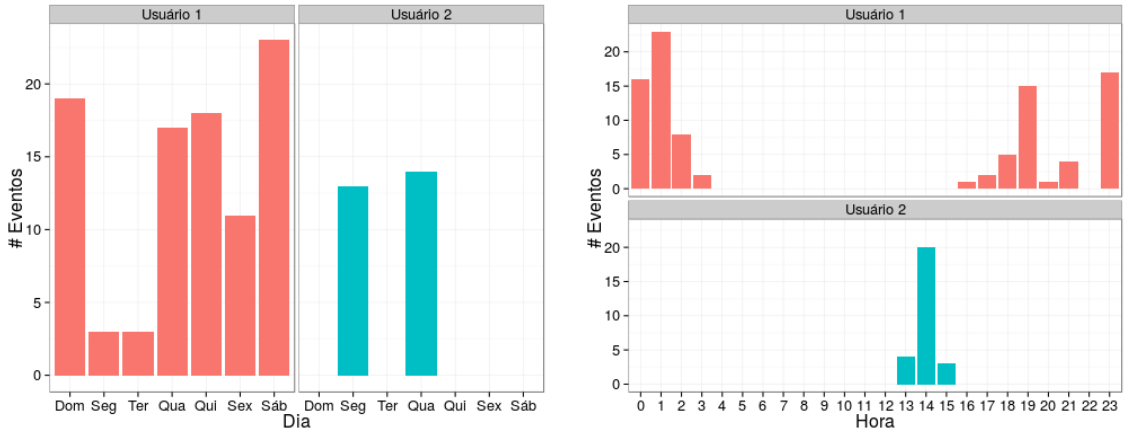


Figura 6.7: Análise da Esparsidade do Evento e do Usuário (San Jose)

repetições, padrões temporais de comportamento. Esses padrões são tão claros que muitas vezes nossos familiares são capazes de prever onde estaremos a qualquer momento do dia.

Da mesma forma que criamos rotinas de trabalho, também definimos perfis temporais para os eventos que frequentamos. Alguns usuários preferem participar de eventos nas noites de Sexta-Feira, outros nas manhãs de Domingo. A Figura 6.8 mostra dois casos reais interessantes. Enquanto o Usuário 1 participa de eventos todos os dias, especificamente a noite, o Usuário 2 participa de eventos durante a semana, no período da tarde apenas.

Para capturar essa intuição temporal utilizamos o algoritmo clássico de filtragem colaborativa baseado nos k-vizinhos mais próximos do usuário-alvo (HERLOCKER et al., 1999). Cada usuário é representado por um vetor tal que cada componente representa a quantidade de eventos que o usuário participou em cada dia (da semana) e hora. Trata-se de uma representação similar à utilizada por Du et al. (2014) também no contexto de recomendação de eventos para a RSBE *DoubanEvents*. No caso do Usuário 1 da Figura 6.8a seria representado



(a) Usuário 1 (b) Usuário 2

Figura 6.8: Perfil Temporal dos Usuário

pela concatenação do vetor de dias da semana

$$T_{u_1}^{Dias} := (19, 3, 3, 17, 18, 11, 24)$$

com o vetor T^{Horas} das horas do dia que prefere ir a eventos

$$T_{u_1}^{Horas} := (16, 23, 8, 2, 0, 0, ..., 0, 1, 2, 5, 15, 1, 4, 0, 17)$$

Esse algoritmo supõe que os usuários que participaram de eventos nos mesmos dias e horários do passado tendem a participar de eventos similares no futuro. O algoritmo TEMPO-KNN modela então essa relação de forma colaborativa. Para cada usuário-alvo $u \in U$ são selecionados seus k vizinhos mais próximos de acordo com a similaridade do cosseno (ver Equação 4.2) entre os vetores de atributos temporais que definem os seus perfis. No momento da recomendação os eventos dos k vizinhos são ponderados de acordo com o peso desses vizinhos. Mais formalmente temos

$$\hat{s}_{\text{TEMPO-KNN}}(u,e) := \sum_{i=1}^{k} cosseno(T_u, T_{u_k}) \times i(u_k,e), \tag{6.10}$$

onde u_k é o k-ésimo vizinho do usuário u e $i(u_k,e)$ é uma função indicadora que retorna 1 se $e \in E_{u_k}$, e 0 caso contrário.

Em casos de empate nos valores de dois ou mais eventos ou quando os vizinhos não pos-

suem eventos dentre os candidatos ranqueamos os eventos em ordem crescente de tempo para sua ocorrência, ou seja, os eventos mais próximos de ocorrer são recomendados primeiro.

6.2.1 Seleção da Vizinhança e Análise da Esparsidade

O número de vizinhos k é um fator definidor da cobertura de recomendações de eventos do TEMPO-KNN. Definimos cobertura de recomendações para um usuário qualquer como sendo a capacidade de um modelo recomendar ao menos um número n itens. Dessa maneira, diferentemente dos modelos MRBPR e TFIDF que ranqueiam sempre todos os eventos candidatos, o TEMPO-KNN depende fortemente do número de vizinhos para tal. Quanto mais vizinhos maior a chance de existirem eventos candidatos a serem ranqueados, no entanto, muitos vizinhos podem adicionar ruído e a ordem dos eventos ser danificada.

Com o objetivo de maximizar a $NDCG@10$, o hiper-parâmetro k foi selecionado no conjunto de validação sendo variado entre 25 e 150 vizinhos, como mostra a Figura 6.9. Vemos que essas duas situações parecem ter ocorrido nas cidades investigadas. Em Chicago a melhor $NDCG@10$ foi obtida com 65 vizinhos, deteriorando com o aumento desse número. A deterioração foi ainda mais forte em Phoenix, que com 50 vizinhos tem sua $NDCG@10$ máxima. Já em San Jose o modelo parece ter alcançado um "plateau" após 50 vizinhos, sendo o número de 100 o selecionado por ter os resultados mais promissores.

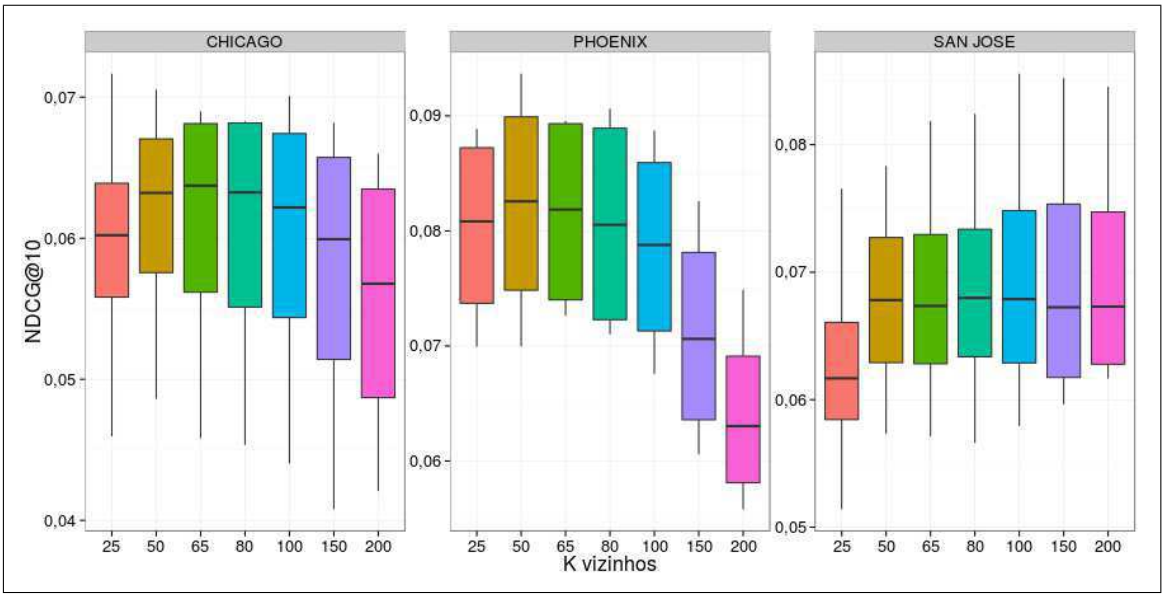


Figura 6.9: Seleção dos Número de Vizinhos

Por fim, buscamos entender o potencial desse modelo diante da esparsidade do usuário e do evento. Por tratar-se de um algoritmo de filtragem colaborativa ele mostrou-se altamente dependente da quantidade de RSVPs como mostrado na Figura 6.10. Nos casos de *cold-start* do evento, sua chance de ser recomendado pela proximidade da ocorrência foi tão baixa que a $NDCG@10$ foi praticamente zero. Com o aumento do número de usuários participantes, as chances dos eventos serem considerados na vizinhança aumentam e consequentemente os resultados melhoram. Já para o usuário, o aumento do número de eventos no passado tem um efeito benéfico de início mas a partir de 6 eventos o perfil temporal aparenta ficar mais ruidoso levando uma piora na $NDCG@10$. E, novamente, a heurística de proximidade de ocorrência de eventos não teve sucesso para os usuários do *cold-start*.

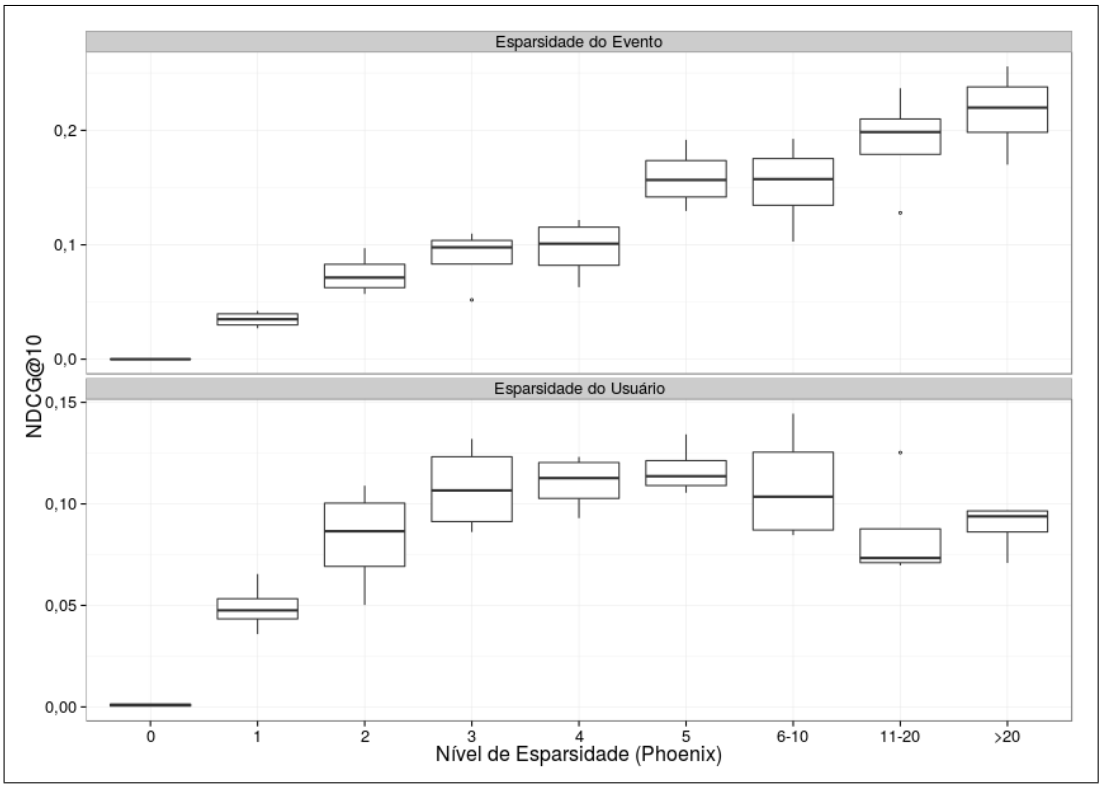


Figura 6.10: Análise da Esparsidade do Evento e do Usuário (Phoenix)

Capítulo 7

Modelo de Ranking Híbrido e Avaliação Experimental

A decisão de participar ou não de um evento é gerada após a análise de várias informações dentre as quais estão os contextos social, conteúdo textual, geográfico e temporal do evento. Além disso, a união desses contextos pode levar a recomendações mais acuradas.

Neste capítulo, definimos um novo modelo híbrido que ranqueia eventos considerando simultaneamente os contextos detalhados anteriormente. Na Seção 7.1 definimos os métodos utilizados para combinar os contextos, na Seção 7.2 apresentamos como os contextos selecionados foram combinados, e selecionamos o híbrido sobre o conjunto de validação na Seção 7.3.

Avaliamos ainda o modelo proposto com relação aos modelos contextuais especializados mostrando os efeitos positivos da hibridização na Seção 7.4 e certificamos a excelência de nossa proposta em relação a uma abordagem do estado-da-arte da literatura de recomendação de eventos em RSBEs na Seção 7.5.

7.1 Métodos de *Learning to Rank*

Learning to Rank é o termo dado para a construção de modelos ou funções para o ranking de itens. Esses modelos tem sido efetivamente aplicados à área de recuperação de informação desde a última década (LIU, 2009), onde deseja-se ranquear um conjunto de documentos de acordo com sua relevância para uma dada consulta. A semelhança entre o problema da área

de recuperação de informação com a de sistemas de recomendação permite-nos aplicar estes métodos analogamente. Considerando que temos uma base de dados de itens e desejamos selecionar os top-n a serem recomendados a determinado usuário, podemos treinar um modelo que aprende a ranquear os itens (ou documentos) de acordo com as características dos usuários (ou consultas).

Desde o seu desenvolvimento, vários tem sido os métodos de *Learning to Rank* desenvolvidos, a maioria pode ser classificada em uma das três seguintes categorias: métodos baseados em instâncias, métodos baseados em pares e métodos baseados em listas.

Métodos baseados em Instância tratam o problema como uma tarefa de regressão ou classificação comum. Esses métodos assumem que cada instância usuário-item possui (a) um valor numérico ou ordinal associado a ele, um ranking por exemplo ou (b) um rótulo de relevância dentre duas ou mais classes possíveis. No primeiro caso, o objetivo é prever o valor corretamente e no segundo caso, reduz-se o problema à um problema de classificação que pode ser resolvido com métodos clássicos de classificação como SVMs ou Regressões Logísticas (NALLAPATI, 2004).

Métodos baseados em Pares como SVMRank (JOACHIMS, 2006), RankNet (BURGES et al., 2005), FRank (TSAI et al., 2007) tentam aprender a preferência dos pares de itens candidatos ao invés do ranking absoluto. Nesses métodos, dada uma lista de itens ranqueada constrói-se um conjunto de pares de itens, de forma que para cada par é dado um rótulo binário $\{+1, -1\}$ representando se o primeiro item é mais relevante que o segundo ou não, respectivamente. Essa construção transforma o problema de ranking em um problema de classificação binária. O objetivo é então treinar um classificador binário que minimize a quantidade de erros de classificação de pares, ou seja, que seja capaz de inferir uma ordem total aos itens candidatos a partir de preferências entre pares.

Métodos baseados em Listas operam na lista completa de itens candidatos. Diferentemente dos anteriores que minimizam um erro baseado no ranking de um item específico ou de pares de itens, uma métrica adequada ao domínio do problema (e.g. $NDCG$) é otimizada sendo calculada sobre a lista corretamente ranqueada vs. a lista estimada. Esses métodos são mais recentes que os dois anteriores e obtiveram resultados superiores em vários problemas de recuperação da informação. Alguns exemplos deles são o ListNet (CAO et al., 2007), Coordinate Ascent (METZLER; CROFT, 2007) e AdaRank (XU; LI, 2007).

Para o problema da geração de listas ranqueadas de eventos para usuários em RSBEs selecionamos um método de cada categoria sendo o principal motivo a simplicidade, facilidade de implementação e utilização:

- A Regressão Logística é um método de classificação binária que retorna uma probabilidade para cada evento indicando a sua relevância. Seu aprendizado é realizado minimizando o erro logístico entre a classe real e classe estimada, no caso, se o evento realmente receberá um RSVP do usuário ou não. Durante o teste, cada evento candidato recebe uma probabilidade que é utilizada para ranqueá-los.

- O SVMRank (JOACHIMS, 2006) é um método de ranking pareado que transforma o problema de regressão ordinal em classificação binária, foi desenvolvido inicialmente para o ranking de páginas Web mas já obteve sucesso em diversas outras aplicações. As instâncias de treino da classificação são geradas da seguinte forma: dado que temos uma lista de itens corretamente ranqueados para um usuário no treino, calcula-se a diferença entre os vetores de atributos de cada par de item e define-se o rótulo como $+1$ ou -1, de acordo com a ordem dos itens. Com as instâncias definidas, treina-se um modelo SVM para aprender a superfície linear de separação dos pares de itens.

- O Coordinate-Ascent proposto por Metzler e Croft (2007) é um modelo linear de lista para a recuperação da informação que usa a clássica otimização do *Coordinate Ascent* para otimizar os parâmetros do modelo. Seu treino é realizado ciclicamente, de forma que em cada ciclo otimiza uma dimensão enquanto fixa as demais. Seu diferencial com relação aos outros algoritmos listados é sua capacidade de otimizar diretamente uma métrica de ranking durante o treino, no caso a $NDCG@10$.

7.2 Seleção de Atributos

A maioria dos algoritmos de *Learning to Rank* codifica os dados em vetores de atributos como no aprendizado de máquina clássico. Propomos então a definição dos atributos a partir da execução dos modelos contextuais especializados com o adicional do contador dos RSVPs recebidos pelo evento durante o treino. Listamos abaixo esses atributos calculados para cada instância usuário-evento, i.e. par (u,e):

1. $\hat{s}_{\text{MR-BPR}}(u,e)$ da recomendação do evento e para o usuário u pelo modelo MR-BPR (componente social);

2. $\hat{s}_{\text{GRUPO-FREQ}}(u,e)$ da recomendação do evento e para o usuário u pelo modelo GRUPO-FREQUENTE (componente social);

3. $\hat{s}_{\text{TFIDF}}(u,e)$ da recomendação do evento e para o usuário u pelo modelo TFIDF (componente de conteúdo);

4. $\hat{s}_{\text{GEO-KERNEL}}(u,e)$ da recomendação do evento e para o usuário u pelo modelo GEO-KERNEL (componente geográfica);

5. $\hat{s}_{\text{TEMPO-KNN}}(u,e)$ da recomendação do evento e para o usuário u pelo modelo TEMPO-KNN (componente temporal);

6. $|U_e|$ quantidade do usuários que já enviaram RSVPs para o evento e no treino.

Os atributos de treino para algoritmos de *Learning to Rank* podem ser categorizados em dinâmicos, estáticos ou de consulta. Já adequando ao contexto da recomendação de eventos, os atributos dinâmicos especificam as relações do usuário com o evento, os estáticos dizem respeito apenas aos eventos e os de consulta ao usuário. Sendo assim, os atributos de 1-5 são dinâmicos, pois advém de recomendações personalizadas de eventos e o atributo 6 é estático sendo relativo ao evento. Não utilizamos atributos do usuário, pois este não teria valor discriminativo algum já que seria igual para todos os eventos. É importante notar que todos os atributos foram normalizados centralizando os dados para média zero e desvio padrão unitário para evitar o viés da diferença nas escalas dos atributos.

Os estágios da recomendação de eventos do modelo de ranking híbrido são expostos na Figura 7.1. Consideramos que todos os modelos parciais foram previamente treinados. Dado um usuário de teste $u \in U_{teste}$ cada modelo especializado recebe e atribui seu valor de relevância para cada evento candidato $e \in E_{teste}$ personalizadamente. Monta-se então o vetor de atributos como listado anteriormente e repassa-o para o algoritmo de *Learning to Rank* que define o valor de relevância final do evento. Ao término, a lista de recomendação é gerada com os top-n eventos candidatos com maior relevância final, os quais são retornados ordenadamente para o usuário.

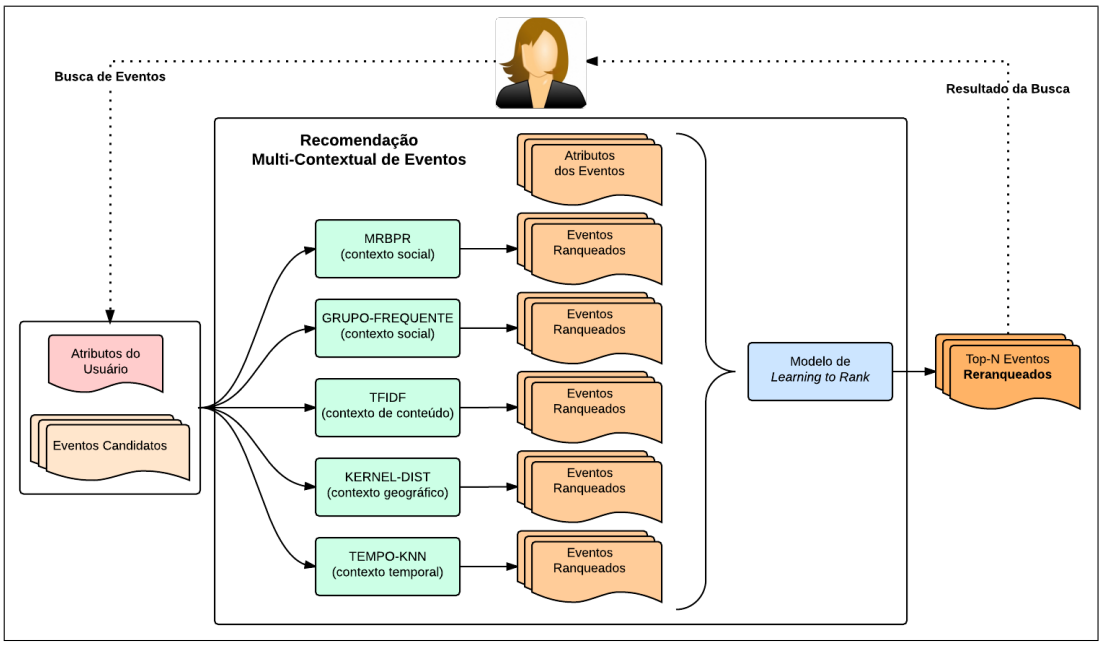


Figura 7.1: Estágios da Recomendação de Eventos pelo Modelo de Ranking Híbrido

7.3 Seleção do Método de *Learning to Rank*

Realizamos uma modificação na metodologia de treino e teste, pois necessitamos de uma fase de treino inicial para os modelos contextuais e outra adicional para os modelos de *Learning to Rank*. Assim, seguimos a metodologia abaixo para avaliar um modelo de *Learning to Rank* em uma partição p_i, considerando que existe uma partição p_{i-1} anterior temporalmente a p_i:

1. Treinamos os modelos contextuais nos dados de treino de p_{i-1} (primeira fase de treino);

2. Cada evento candidato $e \in E_{teste}$ de p_{i-1} recebe um valor de ranking por cada modelo contextual e tem seu vetor de atributos gerado da forma definida anteriormente;

3. Todos os vetores de atributos são reunidos e montam a base de treino dos modelos de *Learning to Rank* (segunda fase de treino);

4. Por fim, avaliamos estes modelos com os dados de teste da partição p_i.

É importante ressaltar que utilizamos os dados de treino e teste de p_{i-1} e os dados de teste de p_i, mas não utilizamos os dados de treino p_i. A ideia que nos motivou a definir essa

metodologia de treino e teste foi a de um ambiente de utilização real no qual este modelo fosse aplicado. Vejamos o exemplo a seguir:

Uma empresa W criou um sistema de recomendação de eventos utilizando a modelagem Híbrida proposta neste trabalho. Em uma primeira versão do sistema treinou cada um dos modelos contextuais e também o modelo de *Learning to Rank* com dados históricos coletados a partir de modelos não personalizados (e.g. recomendação de eventos mais populares, aleatórios). Essa versão funciona como exposto na Figura 7.1, cada modelo contextual ranqueia os eventos utilizando seus critérios particulares, essas listas de eventos alimentam o modelo de *Learning to Rank*, e esse, por sua vez, re-ranqueia os eventos antes de retornar para o usuário.

No entanto, a empresa W desejou atualizar o modelo periodicamente, para isso definiu o seguinte processo. Toda madrugada de sábado (i.e. p_{i-1}) os modelos contextuais são treinados com dados de históricos (e.g. uma semana de dados). Após o treino, são selecionadas cinco amostras aleatórias e representativas dos usuários do sistema, e cada modelo contextual realiza recomendações diretamente para elas, sem o re-ranqueamento do modelo de *Learning to Rank*, durante todo o fim de semana.

Na madrugada da segunda-feira, avaliam-se os erros e acertos dos modelos contextuais durante o final de semana e define-se uma base de treino para um novo modelo de *Learning to Rank*. Logo após, novos modelos contextuais são treinados com dados históricos que não incluem os usuários das amostras anteriores para evitar um possível viés, e também um novo modelo de *Learning to Rank* com a base previamente definida.

Ao final, o sistema em produção é atualizado com os novos modelos contextuais e de *Learning to Rank* (sem *downtime*, preferencialmente), e toda segunda-feira às 8 horas em média (i.e. p_i) os usuários recebem recomendações partindo de modelos atualizados.

Com esta metodologia, nossos experimentos para definição de hiper-parâmetros e seleção do método de *Learning to Rank* foram executados sobre os seguintes pares treino e teste advindos da conjunto de validação: treino em p_1 e teste em p_2, treino em p_2 e teste em p_3 e treino em p_3 e teste em p_4.

Os hiper-parâmetros do Coordinate-Ascent foram selecionados experimentalmente sobre estas partições, tendo os melhores resultados de $NDCG@10$ sem regularização, com 1 reinício apenas (variando entre 1 e 5 reinícios) em no máximo 25 iterações para Chicago e

Phoenix, e 10 iterações para San Jose (variando entre 1, 5, 10 e 25 iterações). E no SVMRank utilizamos os parâmetros padrão da implementação[1].

A Figura 7.2 compara os modelos acima em termos de $NDCG@10$ para as três partições de teste. Vemos que todos obtiveram resultados similares, a Regressão Logística mesmo sendo um modelo de classificação soube dosar bem os pesos dos atributos, já o SVMRank não teve tanto sucesso na cidade de Phoenix. Ao final, o Coordinate-Ascent mostrou-se o mais promissor dentre os demais, sendo selecionado como o método de *Learning to Rank* do nosso modelo Híbrido.

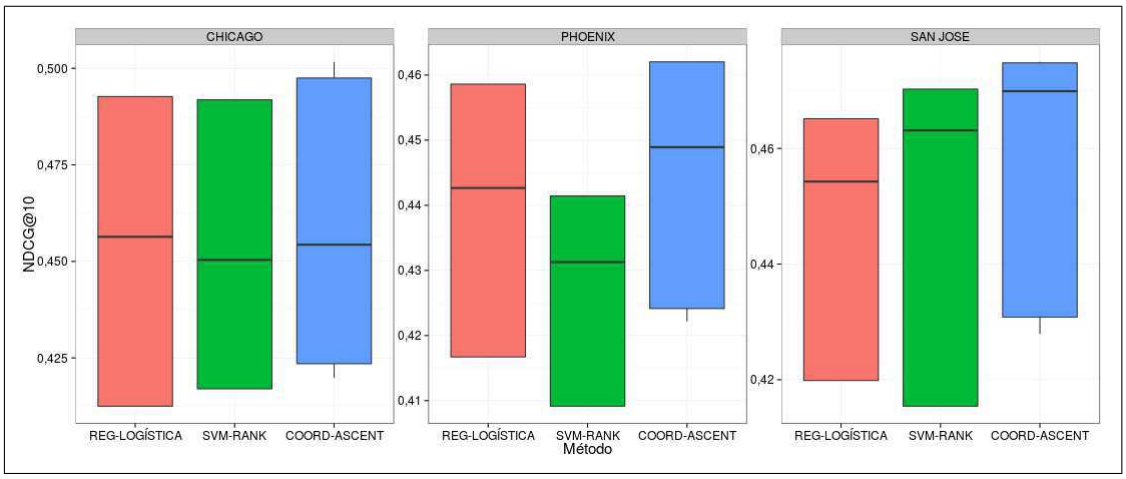


Figura 7.2: Comparação dos Métodos de *Learning to Rank*

7.4 Análise do Modelo Híbrido

Essa análise tem como objetivo atestar a capacidade do modelo híbrido em gerar listas de eventos ainda melhores em termos de $NDCG@10$ do que os modelos especializados. A Figura 7.3 compara os modelos por meio de diagramas de caixa, de forma que em cada caixa estão os valores da $NDCG@10$ para as oito partições do conjunto de avaliação (i.e. partições numeradas de 5 a 12). No eixo-x está o modelo MAIS-POPULAR, normalmente utilizando na literatura como base de comparação, o qual ranqueia os eventos em ordem decrescente de quantidade de RSVPs, logo em seguida estão os modelos especializados e o híbrido. A ordem dos modelos foi definida de acordo com as capacidades de recomendação

[1]Disponível online em: <http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html>

para facilitar a visualização.

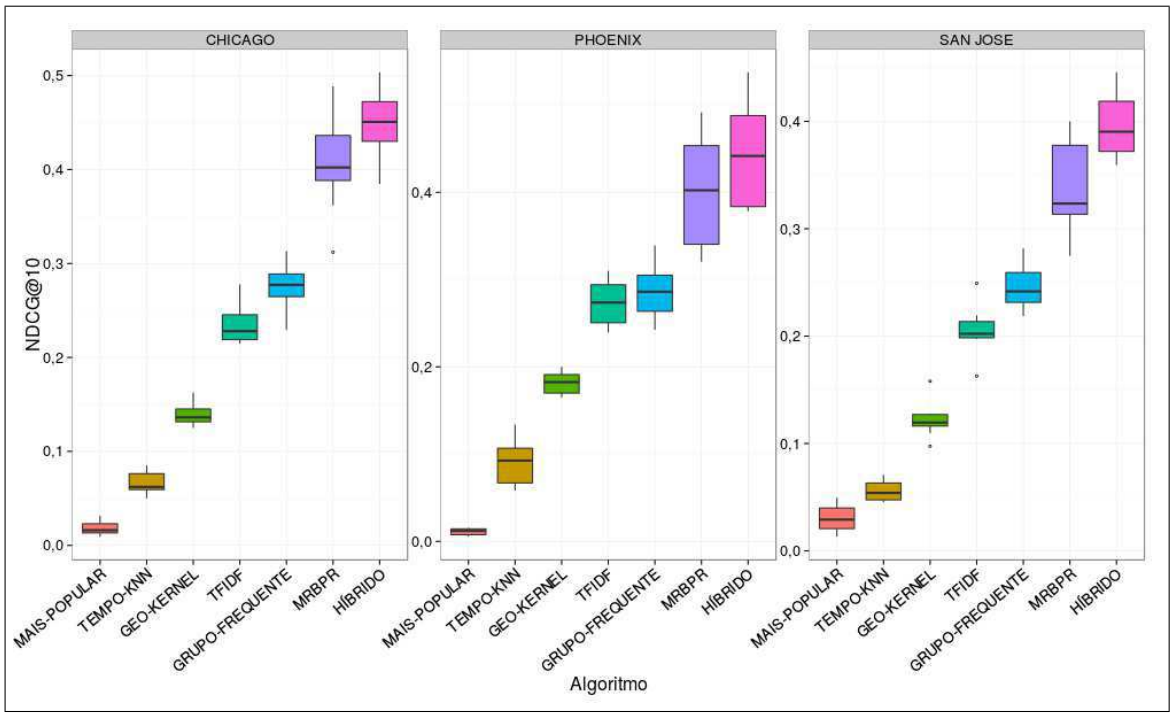


Figura 7.3: Comparação dos Modelos especializados com o modelo Híbrido

O contexto social desponta com os melhores valores de $NDCG@10$ com a heurística de frequência nos grupos (i.e. GRUPO-FREQUENTE), e, principalmente, com o MR-BPR que novamente captura a estrutura social multi-relacional latente aos dados. Vemos também que a recomendação baseada no perfil textual dos eventos é bastante promissora. E os contextos geográfico e temporal mesmo obtendo resultados inferiores aos demais conseguem capturar informações inerentes ao domínio obtendo resultados superiores ao algoritmo MAIS-POPULAR.

Os contextos isoladamente aparentam ter bons resultados, no entanto, ao unirmos o potencial ranqueador especializado em um Híbrido de aprendizagem multi-contextual alcançamos listas de recomendação ainda melhores. Esse argumento parece não ter solidez ao vermos que as caixas do MR-BPR e do Híbrido se sobrepõem na Fig. 7.3. Realizamos então uma análise estatística para diagnosticar ou não a diferença entre as distribuições da $NDCG@10$ do MR-BPR e do Híbrido em cada partição do conjunto de validação por região.

Inicialmente checamos a normalidade da distribuição da diferença das $NDCG@10$ com o teste Shapiro-Wilk, depois comparamo-las pareadamente por região com o teste T, bus-

cando refutar a hipótese de que ambas as distribuições são iguais. A Tabela 7.1 mostra os resultados dos testes por região. Vemos que o teste Shapiro-Wilk não refutou a hipótese de normalidade para nenhuma região, ou seja, as distribuições da diferença podem ser consideradas normais. Na próxima coluna vemos que o Teste-T refutou fortemente a hipótese de igualdade das distribuições, com p-valor menor que $0{,}001$ para todas as regiões. Concluímos assim que o modelo Híbrido é um modelo estatisticamente diferente do modelo contextual MR-BPR.

Região	**Teste Shapiro-Wilk (p-valor)**	**Teste T (p-valor)**
Chicago	0,6330	0,00004916
Phoenix	0,9646	0,00000225
San Jose	0,2169	0,00039945

Tabela 7.1: Resultados dos Testes Estatísticos da diferença entre os modelo MR-BPR e Híbrido

Visualizamos também essa diferença ao analisarmos a $NDCG@10$ por partição de cada modelo[2] como exposto na Figura 7.4. Vemos como a hibridização é de fato superior a todos os demais contextos. Existe uma distância entre os valores de $NDCG@10$ do MR-BPR com relação ao Híbrido resultado da influência dos demais contextos. Apesar da maioria dos contextos ter valores de $NDCG@10$ similares com o passar do tempo, os modelos sociais e, consequentemente, o modelo Híbrido demonstram variações bruscas em seus resultados em algumas partições. Esse fato aponta para uma possível causa digna de futuras análises, os modelos sociais são fortemente influenciados pelo tempo. Por exemplo, durante o natal as pessoas tendem a se reunir com mais pessoas e não apenas àquelas dos eventos e grupos habituais.

Analisamos ainda a importância de cada contexto na definição do Híbrido. O algoritmo de *Learning to Rank* Coordinate-Ascent é um modelo linear que estima um peso para cada atributo de entrada (ver atributos na Seção 7.2) maximizando a métrica de ranking, no caso a $NDCG@10$. Como os atributos foram normalizados previamente, podemos utilizar essas estimativas para compará-los de acordo com a sua importância para o modelo final.

A Figura 7.5 tem no eixo-y os pesos estimados para cada atributo (em cores) por partição (eixo-x). Podemos extrair alguns *insights* do comportamento do modelo Híbrido a partir da

[2]O Híbrido não é avaliado na partição 5, pois ele é treinado em uma partição $p-1$ e testado na partição seguinte p (ver mais detalhes na Seção 7.3)

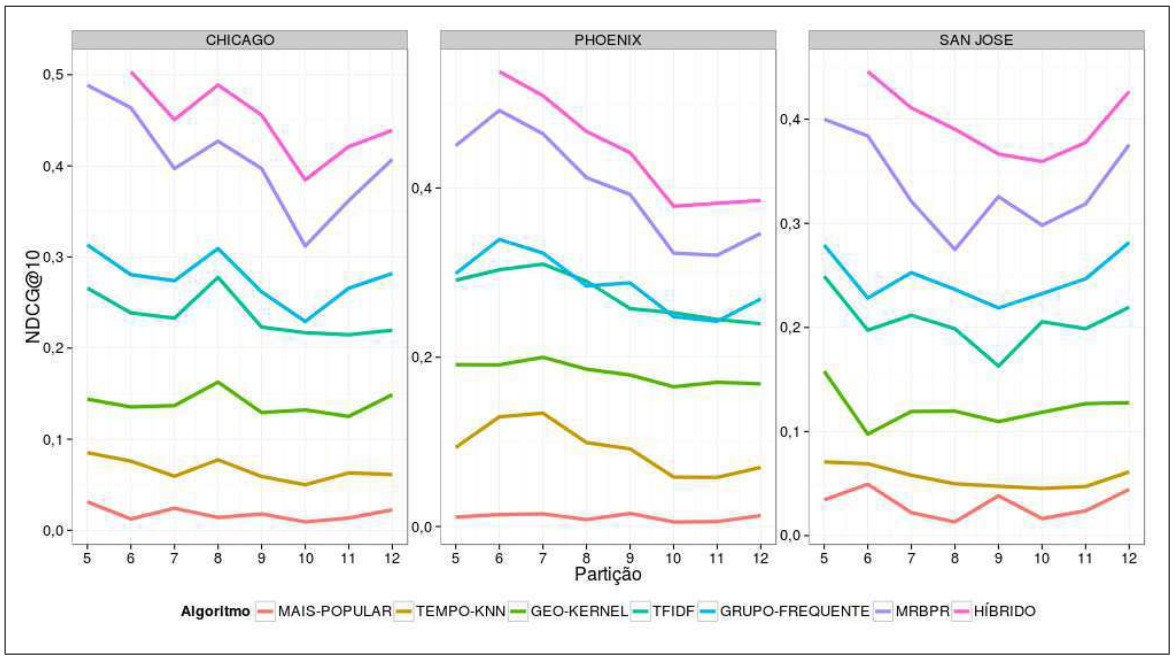


Figura 7.4: Comparação dos Modelos por Partição

análise detalhada dessas curvas.

- O MR-BPR tem os maiores pesos, o que era de se esperar dado que obteve os melhores resultados de $NDCG@10$. Enquanto isso, os demais atributos tem pesos de $0{,}1$ à $0{,}2$, exceto o TEMPO-KNN que contribui muito pouco para a recomendação, mas ainda positivamente.

- O #RSVPs do Evento tem pesos negativos na maioria dos casos, principalmente na cidade de San Jose, levando a um comportamento que minimiza o valor do evento quanto mais RSVPs ele houver recebido no passado, portanto, eventos menores tendem a ser melhor ranqueados. Por outro lado, a esparsidade do usuário é capturada individualmente pelos modelos contextuais. Dessa forma, o modelo Híbrido adapta-se às variações na esparsidade tanto do evento quanto do usuário inerentes à distribuição de RSVPs (ver Cap. 3).

- Na penúltima partição (relativa ao mês de Agosto de 2013) em todas as cidades (com menor intensidade em San Jose) percebemos que o MR-BPR perde relevância e os modelos GRUPO-FREQUENTE e TFIDF o substituem. Comportamentos como este atestam a excelência dos modelos contextuais e, principalmente, do modelo Híbrido

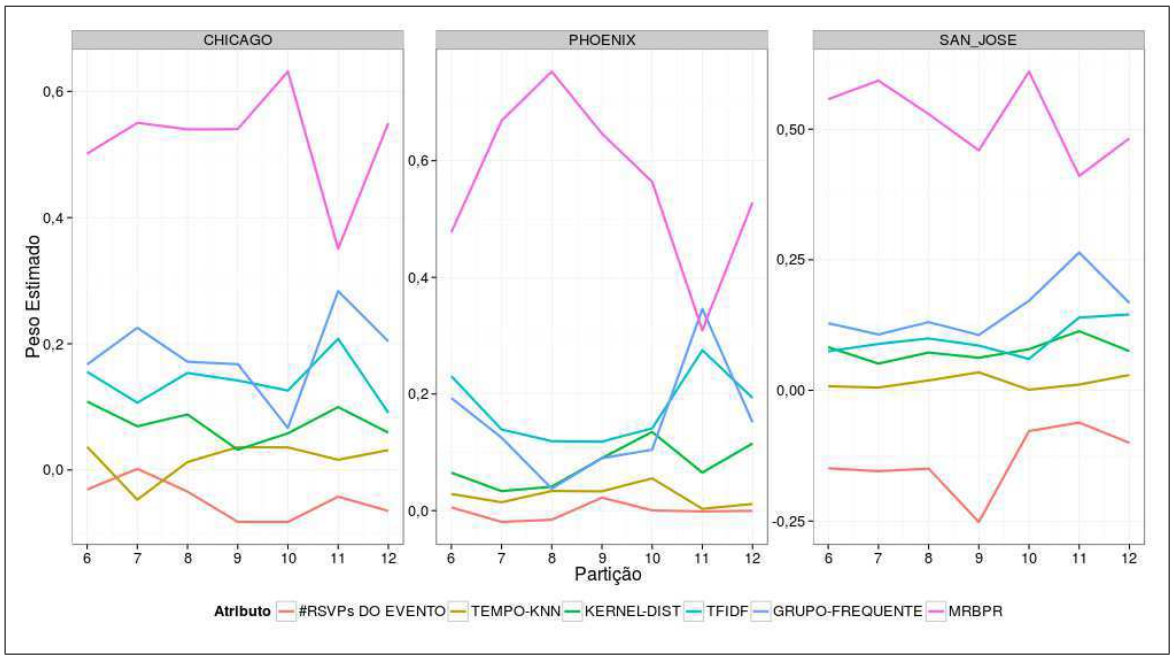


Figura 7.5: Importância dos Atributos para o modelo Híbrido

que consegue adaptar-se às mais variadas situações enquanto maximiza a função de ranking dos eventos recomendados.

7.5 Abordagem Comparativa

Qiao et al. (2014b) desenvolveram um modelo de recomendação de eventos em RSBEs e, similarmente ao nosso trabalho, o avaliaram com dados coletados do Meetup.com como descrito no Capítulo 2. Sendo este o trabalho mais recente e ao mesmo tempo mais similar ao nosso, selecionamo-lo como abordagem comparativa.

Os autores propõem uma variação do método BPRMF descrito e avaliado no Capítulo 4. A variação consistiu na modificação do termo de regularização, chamada de regularização social heterogênea no artigo original, durante o treino do modelo.

Ma et al. (2011) cunharam o termo regularização social no contexto de sistemas de recomendação baseados em fatoração de matrizes para denotar um fator de regularização que é baseado nas relações sociais (e.g. amizade) entre os usuários. A hipótese principal é que as preferências por itens de um usuário são similares à média ponderada das preferências de seus "amigos". Desde a publicação do artigo em 2011 ele já alcançou mais de 290 citações

no Google Scholar[3] tendo aplicações em diversos domínios.

Qiao et al. (2014b), por sua vez, chamam de regularização social heterogênea no contexto de RSBEs a aplicação de dois termos de regularização separados, um baseado nas relações de "amizade" da rede social *online* e outro na rede social *offline*. Para tal, definem a função de similaridade entre dois usuários $u_i \in U$ e $u_j \in U$ como a similaridade de Jaccard, que cresce proporcionalmente com o aumento do número de co-ocorrência nos grupos e nos eventos. Assim, o peso das interações sociais na rede *online* é formalizado como segue:

$$w_{ij}^{\text{on}} := \frac{|G_{u_i} \cap G_{u_j}|}{|G_{u_i} \cup G_{u_j}|}, \tag{7.1}$$

e na rede social *offline*, como:

$$w_{ij}^{\text{off}} := \frac{|E_{u_i} \cap E_{u_j}|}{|E_{u_i} \cup E_{u_j}|}. \tag{7.2}$$

Durante o treino do modelo, os fatores latentes da entidade U, i.e. dado por $\mathbf{U}$, serão regularizados proporcionalmente à média ponderada das similaridades dos vizinhos do usuário via grupos e via eventos. Por exemplo, para um usuário u, temos G_u como os grupos que u se afiliou e E_u os eventos que ele participou. Considerando $V_{G,u}$ como sendo o conjunto de vizinhos de u co-filiados a G_u e $V_{E,u}$ os vizinhos que co-participaram dos mesmos eventos E_u. Então, a chance de u participar de um evento candidato $e^{'}$ será ponderada de acordo com as preferências de seus vizinhos, i.e. $V_{G,u}$ e $V_{E,u}$, tendo maior peso os vizinhos que tiverem maior co-ocorrência nos grupos G_u e nos eventos E_u com u (i.e. similaridade de Jaccard).

Por não termos tido acesso ao código dos autores, mesmo depois de contactá-los por e-mail, e por se tratar de um artigo curto onde muitos detalhes para a correta implementação não estão disponíveis, decidimos simulá-lo por meio do framework de fatoração multi-relacional de matrizes, MR-BPR. Definimos então um modelo chamado de MRBPR-SOCIAL, baseado nas relações especificadas a seguir e expostas na Figura 7.6:

- R_{UE} a relação alvo de RSVPs positivos padrão;
- R_{UU^G} a relação binária baseada na rede social *online* gerada pela co-ocorrência de usuários em um mesmo grupo. Assim para cada par de usuários u_i e u_j afiliados ao mesmo grupo g existe uma aresta bi-direcional na matriz $U \times U$;

[3]<https://scholar.google.com.br/scholar?cites=12313386522558753870>

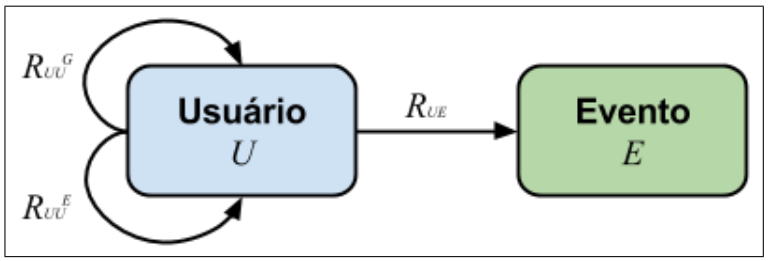


Figura 7.6: Entidades e Relações do Modelo MRBPR-SOCIAL

- R_{UU^E} a relação binária análoga a anterior baseada na rede social *offline* gerada pela co-participação de usuários u_i e u_j em um mesmo evento e.

Com essas relações induzimos o modelo a aprender as interações entre usuários por meio das redes sociais *online* e *offline* como proposto em (QIAO et al., 2014b), com o diferencial que os fatores latentes irão aprender seu potencial preditivo e não utilizá-las apenas como termos de regularização.

O principal obstáculo encontrado durante a implementação do MRBPR-SOCIAL foi a necessidade de materializarmos a rede social *online*, formada pelos grupos. Para solucionar decidimos realizar uma amostragem aleatória dos membros dos grupos. Analisamos a distribuição do tamanho dos grupos nas três cidades na Figura 7.7 para as partições do conjunto de avaliação e observamos que mais de 85% dos grupos tem 100 ou menos membros em todas as cidades. Assim, para os grupos com mais de 100 usuários selecionamos aleatoriamente essa quantidade durante a materialização da relação R_{UU_G}.

Os pesos das relações e os hiper-parâmetros do modelo foram selecionados em um processo similar ao descrito no Capítulo 4. Ao término a relação alvo R_{UE} teve peso de $0{,}25$ a relação social online R_{UU_G} de $0{,}6$ e a *offline* R_{UU_E} de $0{,}15$, demonstrando que novamente os grupos são determinantes na seleção dos eventos do Meetup. Os hiper-parâmetros foram definidos experimentalmente em $k = 200$, $\gamma = 0{,}1$ e 600 iterações do SGD.

O modelo foi treinado e testado com as partições de avaliação para compararmos os resultados com nossa solução Híbrida. A Figura 7.8 sumariza a $NDCG@10$ e mostra como a abordagem Híbrida foi superior ao método da literatura na recomendação de eventos com uma melhoria de aproximadamente 64% em Chicago, 71% em Phoenix e 78% em Phoenix comparando as medianas das distribuições.

Comparamos também a capacidade dos métodos diante de vários níveis de esparsidade.

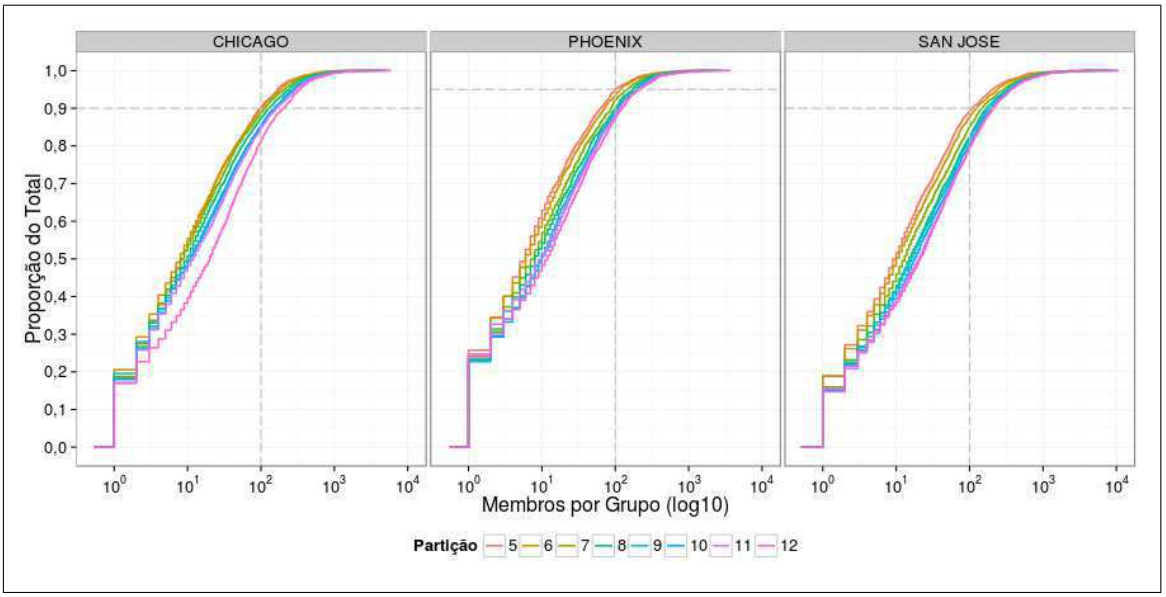


Figura 7.7: Distribuição Acumulada do Tamanho dos Grupos

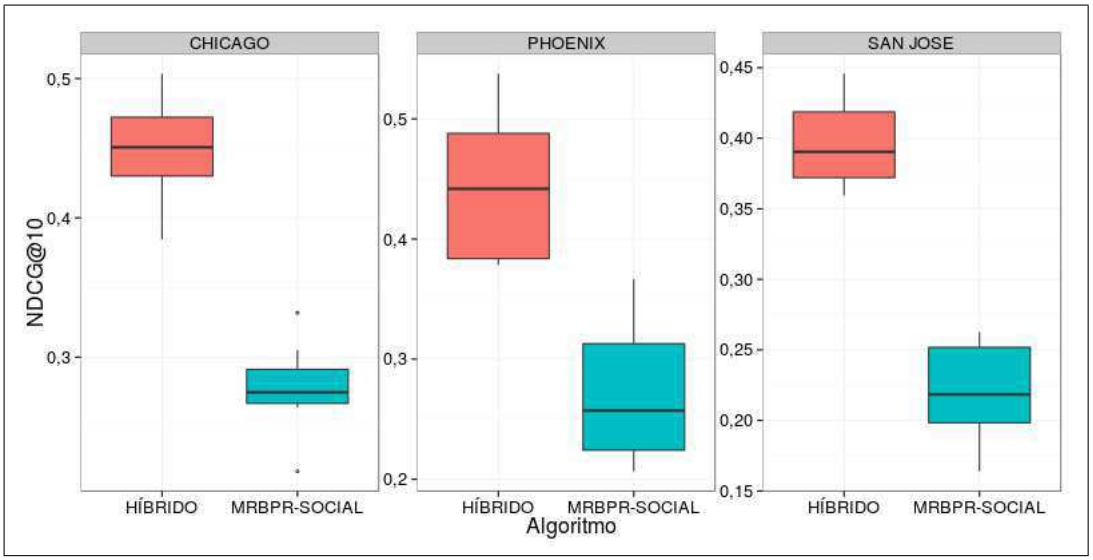


Figura 7.8: Comparação do modelo Híbrido e do MRBPR-SOCIAL

A Figura 7.9 avalia os modelos estratificando por nível de esparsidade do usuário. Vemos que o MRBPR-SOCIAL ensaia uma melhoria na $NDCG@10$ com o acréscimo de dados, no entanto, deteriora com o acréscimo do número de RSVPs do usuário. Por outro lado, o Híbrido já consegue ranquear eventos para usuários no *cold-start* com alta eficácia relativa aos demais níveis de esparsidade e tende a melhorar a $NDCG@10$ com o acréscimo do número de RSVPs.

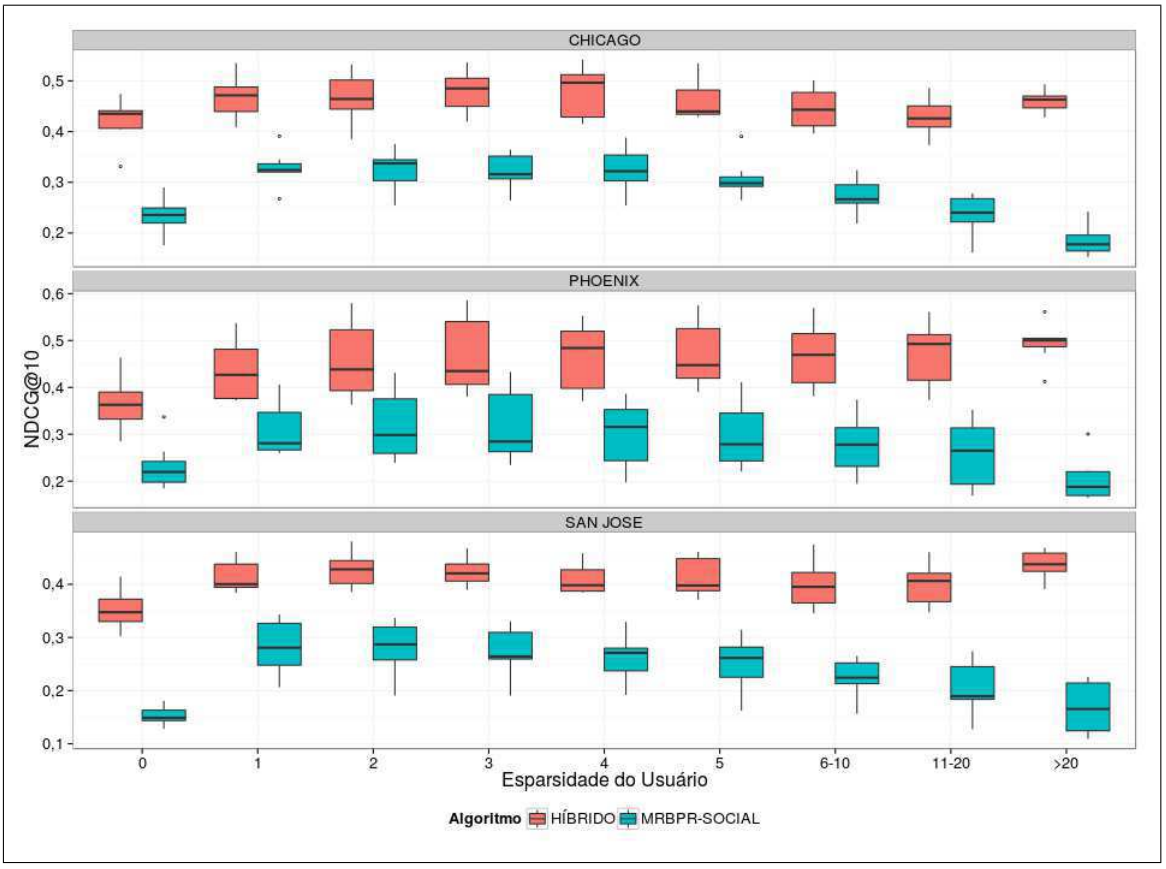


Figura 7.9: Esparsidade do Usuário entre o Híbrido e o MRBPR-SOCIAL

Já a análise da esparsidade do evento na Figura 7.10 apresenta-nos uma característica diferente. Os diagramas de caixa dos modelos estão distanciados para os níveis de esparsidade menores e se aproximam com o aumento do nível. O que indica que ambos os modelos são igualmente bons em recomendar eventos quando estes já possuem uma grande quantidade de informação colaborativa (i.e. RSVPs positivos), no entanto, quanto menos dados piores são as recomendações do método da literatura.

Uma característica marcante que diferencia o modelo Híbrido desse método da literatura é o fato dele ser capaz de recomendar eventos para usuários mesmo quando estes não possuem nenhuma informação colaborativa, *cold-start*. A componente geográfica recomendará o evento sem RSVPs aos usuários bastando que ele esteja em um local com alta probabilidade na distribuição geográfica do usuário. A componente temporal também recomendará os eventos mais próximos. A componente de conteúdo irá capturar a similaridade da descrição do evento com o perfil textual do usuário. E, principalmente, a componente social com o

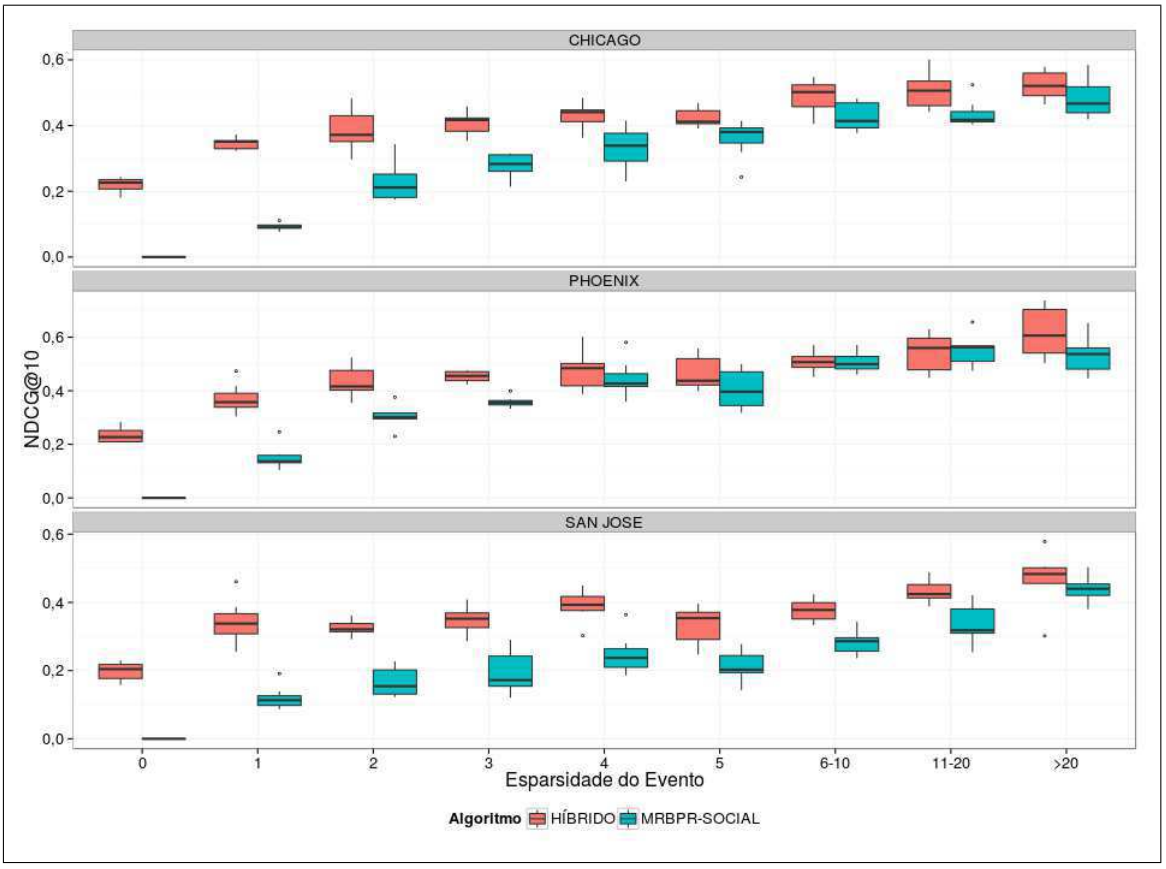


Figura 7.10: Esparsidade do Evento entre o Híbrido e o MRBPR-SOCIAL

MR-BPR que aprende nos fatores latentes um caminho alternativo para a relação R_{UE} por meio das relações R_{UG} e R_{GE} (ver Cap. 4). Já a abordagem da literatura não relaciona os novos eventos aos seus grupos por considerar as redes sociais separadamente em seu modelo. Até mesmo com a nossa simulação pelo MR-BPR, se um evento nunca recebeu um RSVP de um usuário ele não será otimizado no treino, não recebendo recomendações no teste.

Capítulo 8

Conclusão

Neste trabalho, propomos um modelo de recomendação multi-contextual de eventos em Redes Sociais Baseadas em Eventos. Diferentemente da maioria das soluções da literatura a adição de múltiplos contextos em um único modelo aumenta sua eficácia de recomendação, principalmente por lidar com casos extremos como *cold-start*, muito comum no domínio de RSBEs.

O modelo Híbrido foi desenvolvido com foco na RSBE Meetup.com utilizando os seguintes sinais contextuais inerentes ao domínio: informações sociais que relacionam usuários via grupos *online*, a descrição textual dos eventos que possibilita sua recomendação mesmo quando não possui RSVPs, as preferências geográficas e temporais do usuário, fatores primordiais na tomada de decisão dos usuários. Cada sinal contextual é capturado por modelos especializados que se unem por meio de um modelo Híbrido de aprendizagem que otimiza o ranking do eventos.

Avaliamos a eficácia de ranking do modelo proposto em níveis crescentes de esparsidade do histórico de RSVPs do usuário e do evento, em diversos momentos no tempo como também em diferentes condições culturais, e o mesmo obteve resultado superiores à uma abordagem do estado-da-arte da literatura (QIAO et al., 2014b) desenvolvida para a mesma RSBE, i.e. Meetup.com.

A utilização dos múltiplos contextos gera também resultados positivos para o usuário em si, além da maior eficácia em predizer as preferências do usuário. Em situações reais, o modelo Híbrido recomendará eventos com maior conhecimento dos contextos que envolvem as decisões do usuário, gerando muitas vezes listas mais diversificadas, ou com maior

abrangência da população de eventos e ainda com alta eficácia.

Contribuímos ainda com os resultados de uma análise extensiva dos contextos, tanto em termos de suas características como em termos de seu poder preditivo na recomendação de eventos. Além disso, realizamos uma coleta massiva de dados do Meetup.com com enfoque em três populosas cidades dos EUA a qual garantiu a representatividade estatística dos resultados aqui expostos para amostras similares.

As contribuições deste trabalho também tem aplicação prática na indústria. Uma aplicação direta seria no próprio Meetup.com. No final de 2013 eles anunciaram em uma conferência de tecnologia que recomendam eventos utilizando o algoritmo de Regressão Logística com atributos relativos à palavras-chave do usuário, amizades no Facebook, idade, sexo e dados de localização[1]. Apesar desse modelo utilizar dados diferentes dos que tivemos acesso, nosso modelo aprende as preferências dos usuários personalizadamente e poderia facilmente agregar novos modelos contextuais baseados em palavras-chave, por exemplo.

O modelo proposto é extensível para outras RSBEs apenas com a adequação dos sinais contextuais disponíveis. Outra contribuição prática para RSBEs reais está na extração de ideias de como implementar ou evoluir suas soluções e processos de recomendação de eventos, como por exemplo, o método de treino e teste dos modelos contextuais e do modelo Híbrido, o processo de hibridização da recomendação ou a implementação de um ou mais modelos contextuais específicos.

Em termos de escalabilidade os modelos contextuais podem ser treinados em paralelo. Caso seja necessário, todos os modelos podem também ser implementados com padrões *map-reduce*, exceto o MR-BPR e o Coordinate-Ascent os quais poderiam ser implementados utilizando frameworks como Spark[2] que implementam processos iterativos e abstração de memória única sobre *clusters* de computadores.

8.1 Limitações

Diversas foram as contribuições para o domínio de RSBEs desenvolvidas neste trabalho, no entanto, é importante ressaltarmos também as suas limitações.

[1] Os slides da apresentação estão disponíveis em <http://www.slideshare.net/eestola/ml-data-at-meetup>
[2] <https://spark.apache.org/>

Em termos experimentais a amostra de dados captada de três cidades populosas dos EUA pode não ser representativa de cidades em outros países, ou até mesmo em cidades com menos habitantes dos EUA.

Em termos da solução proposta, sendo o contexto social o mais importante, e, mais especificamente, as relações entre usuários com grupos (i.e. R_{UG}) e grupos com eventos (i.e. R_{GE}) as de maior valor para o modelo Híbrido, podemos concluir que a eficácia do modelo está diretamente relacionada a existência dos grupos. Em outras RSBEs que não disponibilizem essa informação o modelo Híbrido terá uma configuração diferente de pesos para os sinais contextos podendo ser relativamente menos eficaz. No entanto, essa hipótese pode ser refutada caso os demais sinais contextuais (e.g. conteúdo do evento, preferências geográficas do usuário) sejam de igual ou maior relevância para a recomendação que os grupos.

8.2 Trabalhos Futuros

Como trabalhos futuros, antevemos os seguintes direcionamentos:

- Definição de um novo modelo contextual temporal que considere características personalizadas (e.g. dias e horas preferidas do usuário), características não personalizadas do evento (e.g. fim de ano) e do momento da recomendação;
- Proposição de novos modelos contextuais para recomendação de eventos utilizando outros sinais contextuais inerentes ao domínio, como por exemplo:
  - a relação entre os usuários e suas preferências por palavras-chave específicas;
  - atributos pessoais do usuário (e.g. idade, gênero);
  - dados de RSVPs Negativos[3].
- Reexecução do experimento com um maior número de partições para aumentar a representatividade dos resultados e diminuir a variância;
- Avaliação dos modelos propostos com outras cidades do Meetup.com, em especial cidades brasileiras, já que a RSBE tem se expandido no território brasileiro no último ano;

[3] Acreditamos que essa seria uma contribuição inédita para a literatura

- Aplicação de nossa abordagem híbrida de aprendizagem multi-contextual em outras RSBEs (e.g. *Facebook Events*);
- Contactar o Meetup.com para que apliquemos o modelo proposto em uma base de dados real.

Bibliografia

ADOMAVICIUS, G.; TUZHILIN, A. Context-aware recommender systems. In: *Recommender Systems Handbook*. [S.l.: s.n.], 2011. p. 217–253.

BEER, W. et al. General framework for context-aware recommendation of social events. In: *INTELLI 2013, The Second International Conference on Intelligent Systems and Applications*. [S.l.: s.n.], 2013. p. 141–146.

BIZER, C. et al. Linked data on the web. In: *Proceedings of the 17th international conference on World Wide Web*. New York, NY, USA: ACM, 2008. (WWW '08), p. 1265–1266. ISBN 978-1-60558-085-2.

BLEI, D. M.; NG, A. Y.; JORDAN, M. I. Latent dirichlet allocation. *J. Mach. Learn. Res.*, JMLR.org, v. 3, p. 993–1022, mar. 2003. ISSN 1532-4435.

BOUCHARD, G.; YIN, D.; GUO, S. Convex collective matrix factorization. In: *AISTATS*. [S.l.]: JMLR.org, 2013. (JMLR Proceedings, v. 31), p. 144–152.

BURGES, C. et al. Learning to rank using gradient descent. In: *Proceedings of the 22Nd International Conference on Machine Learning*. New York, NY, USA: ACM, 2005. (ICML '05), p. 89–96. ISBN 1-59593-180-5.

CAO, Z. et al. Learning to rank: From pairwise approach to listwise approach. In: *Proceedings of the 24th International Conference on Machine Learning*. New York, NY, USA: ACM, 2007. (ICML '07), p. 129–136. ISBN 978-1-59593-793-3.

CHASALOW, S. D.; BRAND, R. J. Algorithm AS 299: Generation of simplex lattice points. *Applied Statistics*, v. 44, n. 4, 1995. ISSN 0035-9254.

CHENG, C. et al. Fused matrix factorization with geographical and social influence in location-based social networks. In: *AAAI*. [S.l.: s.n.], 2012.

CHIN, A. et al. Using proximity and homophily to connect conference attendees in a mobile social network. In: *Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on*. [S.l.: s.n.], 2012. p. 79–87. ISSN 1545-0678.

DALY, E. M.; GEYER, W. Effective event discovery: Using location and social information for scoping event recommendations. In: *Proceedings of the Fifth ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2011. (RecSys '11), p. 277–280. ISBN 978-1-4503-0683-6.

DEERWESTER, S. et al. Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, v. 41, n. 6, p. 391–407, 1990.

DESHPANDE, M.; KARYPIS, G. Item-based top-n recommendation algorithms. *ACM Trans. Inf. Syst.*, ACM, New York, NY, USA, v. 22, n. 1, p. 143–177, jan. 2004. ISSN 1046-8188.

DOOMS, S.; PESSEMIER, T. D.; MARTENS, L. A user-centric evaluation of recommender algorithms for an event recommendation system. In: FELFERNIG, A. et al. (Ed.). *Proceedings of the RecSys 2011 : Workshop on Human Decision Making in Recommender Systems (Decisions@RecSys'11) and User-Centric Evaluation of Recommender Systems and Their Interfaces - 2 (UCERSTI 2) affiliated with the 5th ACM Conference on Recommender Systems (RecSys 2011)*. [S.l.]: Ghent University, Department of Information technology, 2011. p. 67–73.

DU, R. et al. Predicting activity attendance in event-based social networks: Content, context and social influence. In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. New York, NY, USA: ACM, 2014. (UbiComp '14), p. 425–434. ISBN 978-1-4503-2968-2.

FORSBLOM, A. et al. Out of the bubble: Serendipitous event recommendations at an urban music festival. In: *Proceedings of the 2012 ACM International Conference on Intelligent User Interfaces*. New York, NY, USA: ACM, 2012. (IUI '12), p. 253–256. ISBN 978-1-4503-1048-2.

GALLAGHER, T. J.; ANDREW, J. D. *Financial Management; Principles and Practice*. [S.l.]: Freeload Press, Inc., 1968. ISBN 9781930789029.

GANTNER, Z. et al. MyMediaLite: A free recommender system library. In: *Proceedings of the 5th ACM Conference on Recommender Systems (RecSys 2011)*. [S.l.: s.n.], 2011.

HERLOCKER, J. L. et al. An algorithmic framework for performing collaborative filtering. In: *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: ACM, 1999. (SIGIR '99), p. 230–237. ISBN 1-58113-096-1.

HOFMANN, T. Unsupervised learning by probabilistic latent semantic analysis. *Mach. Learn.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 42, n. 1-2, p. 177–196, jan. 2001. ISSN 0885-6125.

JOACHIMS, T. Training linear svms in linear time. In: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2006. (KDD '06), p. 217–226. ISBN 1-59593-339-5.

KHROUF, H.; TRONCY, R. Hybrid event recommendation using linked data and user diversity. In: *Proceedings of the 7th ACM conference on Recommender systems*. New York, NY, USA: ACM, 2013. (RecSys '13), p. 185–192. ISBN 978-1-4503-2409-0.

KOREN, Y.; BELL, R.; VOLINSKY, C. Matrix factorization techniques for recommender systems. *Computer*, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 42, n. 8, p. 30–37, ago. 2009. ISSN 0018-9162.

KROHN-GRIMBERGHE, A. et al. Multi-relational matrix factorization using bayesian personalized ranking for social network data. In: *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*. New York, NY, USA: ACM, 2012. (WSDM '12), p. 173–182. ISBN 978-1-4503-0747-5.

LATHIA, N. *Evaluating Collaborative Filtering Over Time*. Tese (Doutorado) — University of London, Department of Computer Science, University College London, 2010.

LIAO, G. et al. An effective latent networks fusion based model for event recommendation in offline ephemeral social networks. In: *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*. New York, NY, USA: ACM, 2013. (CIKM '13), p. 1655–1660. ISBN 978-1-4503-2263-8.

LIPPERT, C. et al. Relation Prediction in Multi-Relational Domains using Matrix Factorization. In: *NIPS 2008 Workshop on Structured Input Structure Output*. [S.l.: s.n.], 2008.

LIU, T.-Y. Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, Now Publishers Inc., Hanover, MA, USA, v. 3, n. 3, p. 225–331, mar. 2009. ISSN 1554-0669.

LIU, X. et al. Event-based social networks: linking the online and offline social worlds. In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2012. (KDD '12), p. 1032–1040. ISBN 978-1-4503-1462-6.

MA, H. et al. Recommender systems with social regularization. In: *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*. New York, NY, USA: ACM, 2011. (WSDM '11), p. 287–296. ISBN 978-1-4503-0493-1.

MARINHO, L. B. et al. *Recommender Systems for Social Tagging Systems*. [S.l.]: Springer, 2012. ISBN 978-1-4614-1893-1.

METZLER, D.; CROFT, W. B. Linear feature-based models for information retrieval. *Inf. Retr.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 10, n. 3, p. 257–274, jun. 2007. ISSN 1386-4564.

MINKOV, E. et al. Collaborative future event recommendation. In: *Proceedings of the 19th ACM international conference on Information and knowledge management*. New York, NY, USA: ACM, 2010. (CIKM '10), p. 819–828. ISBN 978-1-4503-0099-5.

NALLAPATI, R. Discriminative models for information retrieval. In: *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: ACM, 2004. (SIGIR '04), p. 64–71. ISBN 1-58113-881-4.

NING, X.; KARYPIS, G. Slim: Sparse linear methods for top-n recommender systems. In: *Data Mining (ICDM), 2011 IEEE 11th International Conference on*. [S.l.: s.n.], 2011. p. 497–506. ISSN 1550-4786.

PASCOAL, L. M. L. et al. A social-evolutionary approach to compose a similarity function used on event recommendation. In: *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, Beijing, China, July 6-11, 2014*. [S.l.]: IEEE, 2014. p. 1512–1519.

PESSEMIER, T. D. et al. An event distribution platform for recommending cultural activities. In: *WEBIST*. [S.l.: s.n.], 2011. p. 231–236.

PESSEMIER, T. D. et al. Social recommendations for events. In: *CEUR workshop proceedings*. [S.l.: s.n.], 2013. v. 1066, p. 4. ISSN 1613-0073.

QIAO, Z. et al. *Combining Heterogenous Social and Geographical Information for Event Recommendation*. 2014.

QIAO, Z. et al. *Event Recommendation in Event-Based Social Networks*. 2014.

ŘEHŮŘEK, R.; SOJKA, P. Software Framework for Topic Modelling with Large Corpora. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, 2010. p. 45–50.

RENDLE, S. Factorization machines with libfm. *ACM Trans. Intell. Syst. Technol.*, ACM, New York, NY, USA, v. 3, n. 3, p. 57:1–57:22, maio 2012. ISSN 2157-6904.

RENDLE, S. et al. Bpr: Bayesian personalized ranking from implicit feedback. In: *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. Arlington, Virginia, United States: AUAI Press, 2009. (UAI '09), p. 452–461. ISBN 978-0-9749039-5-8.

RICCI, F.; ROKACH, L.; SHAPIRA, B. Introduction to recommender systems handbook. In: *Recommender Systems Handbook*. [S.l.: s.n.], 2011. p. 1–35.

RICCI, F. et al. (Ed.). *Recommender Systems Handbook*. [S.l.]: Springer, 2011. ISBN 978-0-387-85819-7.

SALTON, G.; WONG, A.; YANG, C. S. A vector space model for automatic indexing. *Commun. ACM*, ACM, New York, NY, USA, v. 18, n. 11, p. 613–620, nov. 1975. ISSN 0001-0782.

SANDHOLM, T.; UNG, H. Real-time, location-aware collaborative filtering of web content. In: *Proceedings of the 2011 Workshop on Context-awareness in Retrieval and Recommendation*. New York, NY, USA: ACM, 2011. (CaRR '11), p. 14–18. ISBN 978-1-4503-0625-6.

SILVERMAN, B. W. *Density estimation for statistics and data analysis*. London: Chapman and Hall, 1986.

SINGH, A. P.; GORDON, G. J. Relational learning via collective matrix factorization. In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2008. (KDD '08), p. 650–658. ISBN 978-1-60558-193-4.

SOLLIS, B.; JESS3. *The Conversation Prism*. 2013. [Online; Acessado em 9-Fevereiro-2015]. Disponível em: <http://thenextweb.com/insider/2013/07/01/reorganizing-the-social-media-landscape-with-the-updated-conversation-prism/>.

TANG, L.; LIU, H. Relational learning via latent social dimensions. In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2009. (KDD '09), p. 817–826. ISBN 978-1-60558-495-9.

TANG, L.; LIU, H. Scalable learning of collective behavior based on sparse social dimensions. In: *Proceedings of the 18th ACM conference on Information and knowledge management*. New York, NY, USA: ACM, 2009. (CIKM '09), p. 1107–1116. ISBN 978-1-60558-512-3.

TSAI, M.-F. et al. Frank: A ranking method with fidelity loss. In: *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: ACM, 2007. (SIGIR '07), p. 383–390. ISBN 978-1-59593-597-7.

VALIZADEGAN, H. et al. Learning to rank by optimizing ndcg measure. In: BENGIO, Y. et al. (Ed.). *NIPS*. [S.l.]: Curran Associates, Inc., 2009. p. 1883–1891. ISBN 9781615679119.

XU, B.; CHIN, A.; COSLEY, D. On how event size and interactivity affect social networks. In: *CHI '13 Extended Abstracts on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2013. (CHI EA '13), p. 865–870. ISBN 978-1-4503-1952-2.

XU, J.; LI, H. Adarank: A boosting algorithm for information retrieval. In: *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: ACM, 2007. (SIGIR '07), p. 391–398. ISBN 978-1-59593-597-7.

YIN, H. et al. LCARS: A spatial item recommender system. *ACM Trans. Inf. Syst.*, v. 32, n. 3, p. 11, 2014.

YIN, H. et al. Lcars: A location-content-aware recommender system. In: *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2013. (KDD '13), p. 221–229. ISBN 978-1-4503-2174-7.

ZHANG, J.-D.; CHOW, C.-Y. igslr: Personalized geo-social location recommendation: A kernel density estimation approach. In: *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. New York, NY, USA: ACM, 2013. (SIGSPATIAL'13), p. 334–343. ISBN 978-1-4503-2521-9.

Apêndice A

Detalhes de Implementação

Os experimentos e análises expostos nesta dissertação foram organizados em vários módulos e pequenos projetos com objetivos bem definidos.

A coleta de dados da API REST do Meetup foi totalmente implementada em Java[1] em um framework que recebe como entrada a lista de cidades das quais seriam realizadas as coletas, como também os identificadores e palavras-passe para realizar as requisições do serviço. Os dados foram persistidos em um banco de dados PostgreSQL[2].

Todas as análises dos dados foram realizadas com a linguagem R[3] sendo auxiliada por uma série de bibliotecas de manipulação de dados (i.e. plyr[4]) e bibliotecas gráficas (e.g. ggplot2[5] e ggmaps[6]). Com scripts R extraímos os dados do banco de dados e realizamos análises, geramos gráficos e preparamos arquivos CSV com os dados de treino e teste para cada partição dos experimentos.

Desenvolvemos um sistema de execução, avaliação e análise de experimentos em Python[7], C++[8], C#[9], Shell Script[10] e R. Esse sistema totalmente integrado agilizou enormemente a implementação de novos algoritmos, como também a modificação e reexecução dos experimentos.

[1] <www.java.com>
[2] <www.postgresql.org>
[3] <cran.r-project.org>
[4] <http://cran.r-project.org/web/packages/plyr/index.html>
[5] <http://cran.r-project.org/web/packages/ggplot2/index.html>
[6] <http://cran.r-project.org/web/packages/ggmap/index.html>
[7] <https://www.python.org/>
[8] <http://www.cplusplus.com/>
[9] <http://www.csharp-station.com/>
[10] <http://en.wikipedia.org/wiki/Shell_script>

Os modelos do contexto social foram implementados em várias linguagens. Para os algoritmos de Vizinhança do Item (i.e. ITEM-KNN) e de Fatoração de Matrizes (i.e. BPR-MF) utilizamos as implementações presentes na biblioteca de código livre MyMediaLite (GANTNER et al., 2011) escrita em C#. O modelo MR-BPR e seu derivado MRBPR-SOCIAL foram implementados em C++, sua alta eficiência temporal e de memória viabilizou a experimentação de uma enorme quantidade de variações de hiper-parâmetros, o que não foi o caso dos algoritmos do MyMediaLite, em especial do algoritmo de Vizinhança do Item. Por fim, o GRUPO-FREQUENTE foi implementado em poucas linhas de código R.

Todos os modelos de conteúdo foram implementados em Python, com a parte do aprendizado das representações textuais provido pela biblioteca de código livre Gensim (ŘEHŮŘEK; SOJKA, 2010) também escrita em Python. Os modelos dos contextos geográfico e temporal foram completamente implementados em R, a maioria com poucas linhas de código, alguns com uma maior quantidade. Por exemplo, para o modelo TEMPO-KNN a quantidade de usuários obrigou-nos a implementar a computação da matriz de similaridade do algoritmo de vizinhança USER-KNN com dados mapeados para o disco, escalando para quantidades muito maiores do que o necessário, com pouco compromisso do tempo de execução.

Após a execução dos modelos as listas de recomendação eram avaliadas em R e várias métricas eram calculadas ao mesmo tempo, não só a $NDCG$, também a Precisão, Revocação, F1 dentre várias outras. Os resultados das métricas eram persistidos em arquivos e consumidos posteriormente para geração de gráficos de análise (e.g. os gráficos de caixa).

Um dos principais aprendizados pessoais deste trabalho foi que quando buscamos implementar os modelos de recomendação do início ao fim, minimizando a utilização de bibliotecas de terceiros, aumentamos as chances de ter um modelo mais adequado ao problema. Um exemplo claro desse diferencial são os casos de *cold-start* em todos os modelos que implementamos esse caso é devidamente tratado, ou pelo algoritmo principal, ou com alternativas, como o algoritmo MAIS-PRÓXIMO aplicado para usuários no *cold-start* após a recomendação do GEO-KERNEL. O mesmo vale para a implementação das métricas, pois além de aprendermos sobre a métrica em si, temos maior controle sobre os resultados, permitindo que evoluamos algoritmos e heurísticas de acordo.

Todas as análises foram realizadas em desktops ou notebooks comuns sendo o mais res-

tritivo um com 4GB de memória RAM e dois núcleos de processamento. Ainda assim a maioria dos processamentos trabalhou com quantidades de dados que couberam na memória RAM, e nos poucos casos que isso não foi possível, ou quando o tempo de execução foi inviável realizamos amostragens dos dados.

Já os experimentos foram executados em uma máquina virtual cedida pelo Programa de Pós-Graduação em Ciência da Computação, com oito núcleos de processamento e 8GB de memória RAM. A execução da maioria dos experimentos em paralelo foi um dos principais móveis para o rápido desenvolvimento do trabalho.