Willian Jean Fuks

***Desenvolvimento de Preditores para Recomendação Automática de
Produtos***

São Paulo

2013

Willian Jean Fuks

***Desenvolvimento de Preditores para
Recomendação Automática de Produtos***

Dissertação apresentada à Escola
Politécnica da Universidade de São
Paulo para obtenção do título de
Mestre em Engenharia.

São Paulo

2013

Willian Jean Fuks

***Desenvolvimento de Preditores para
Recomendação Automática de Produtos***

Dissertação apresentada à Escola
Politécnica da Universidade de São
Paulo para obtenção do título de
Mestre em Engenharia.

Área de Concentração:

Engenharia Mecânica

Orientador:

Fabio Gagliardi Cozman

São Paulo

2013

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, de junho de 2013.

Assinatura do autor _____

Assinatura do orientador _____

FICHA CATALOGRÁFICA

Fuks, Willian Jean

**Desenvolvimento de preditores para recomendação automática de produtos / W.J. Fuks. -- versão corr. -- São Paulo, 2013.
106 p.**

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.

**1. Inteligência artificial 2. Preditores (Desenvolvimento)
3. Produtos I. Universidade de São Paulo. Escola Politécnica.
Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos II.t.**

Agradecimentos

Agradeço aos familiares, amigos, professores e orientadores que me auxiliaram ao longo do desenvolvimento deste projeto, incluindo até mesmo os que estão em outros continentes, cuja iniciativa (Coursera, video-aulas online e fóruns de discussão) me possibilitou superar diversos desafios nesta pesquisa.

Por fim, agradeço à empresa Buscapé por fornecer suporte e ajuda no desenvolvimento deste trabalho o qual não seria possível sem a sua participação.

Resumo

Com o avanço da internet, novos tipos de negócios surgiram. Por exemplo o sistema de anúncios online: produtores de sites e diversos outros conteúdos podem dedicar em uma parte qualquer de sua página um espaço para a impressão de anúncios de diversas lojas em troca de um valor oferecido pelo anunciante. É neste contexto que este trabalho se insere. O objetivo principal é o desenvolvimento de algoritmos que prevêm a probabilidade que um dado usuário tem de se interessar e clicar em um anúncio a que está sendo exposto. Este problema é conhecido como predição de CTR (do inglês, “Click-Through Rate”) ou taxa de conversão. Utiliza-se para isto uma abordagem baseada em regressão logística integrada a técnicas de fatoração de matriz que prevém, através da obtenção de fatores latentes do problema, a probabilidade de conversão para um anúncio impresso em dado site. Além disto, testes considerando uma estratégia dinâmica em função do tempo são apresentados indicando que o desempenho previamente obtido pode melhorar ainda mais. De acordo com o conhecimento do autor, esta é a primeira vez que este modelo com a inclusão da componente dinâmica é relatada na literatura.

Abstract

With the popularization of the internet, new types of business are emerging. An example is the online marketing system: publishers can dedicate in any given space of their websites a place to the printing of banners from different stores in exchange for a fee paid by the advertiser. It's in this context that this work takes place. Its main goal will be the development of algorithms that forecasts the probability that a given user will get interested in the ad he or she is seeing and click it. This problem is also known as CTR Prediction Task. To do so, a logistic regression approach is used combined with matrix factorization techniques that predict, through latent factor models, the probability that the click will occur. On top of that, several tests are conducted utilizing a dynamic approach (varying in function of time) revealing that the performance can increase even higher. According to the author's knowledge, this is the first time this test is conducted on the literature of CTR prediction.

Lista de Figuras

Figura 1: Exemplo dos anúncios em uma busca patrocinada.....	14
Figura 2: Exemplo de anúncios impressos no site da Exame.com.....	14
Figura 3: Gastos em anúncios online ao redor do mundo em comparação com gastos totais.....	15
Figura 4: Projeção de investimentos em anúncios online nos Estados Unidos até 2016 ³	16
Figura 5: Processo de decisão simplificado da empresa Lomadee que escolhe anúncios com o maior CTR para sites requisitantes.....	18
Figura 7: : Idéia geral de um processo de Data Mining aplicado na construção de preditores. [31].	23
Figura 7: Exemplo de amostragem por validação cruzada k-Fold.	24
Figura 11: Exemplo de uma matriz A qualquer representada pela multiplicação de 3 matrizes [31].....	26
Figura 12: Transformações resultantes das operações de multiplicação das matrizes ortogonais V e U..	27
Figura 13: Técnicas de Sistema de Recomendação.	30
Figura 11: Custo da função objetivo em função da quantidade de iterações calculadas.	33
Figura 12: Exemplo da distribuição Dirichlet em função de diferentes valores de α	37
Figura 16: As principais abordagens desenvolvidas na literatura para desenvolvimento de preditor de CTR.	40
Figura 14: Principais abordagens que constituem o modelo a ser desenvolvido.....	41
Figura 15: Matriz utilizada pelo preditor de CTR.	42
Figura 16: Vetores α associados aos sites e vetores β aos anúncios.....	42
Figura 20: Composição das células em eventos positivos ou negativos.....	43
Figura 21: Iteração entre Publicador, Anunciante e Usuário.....	48
Figura 19: Exemplo de Hierarquia construída na estrutura de Sites.	49
Figura 20: Exemplo de Hierarquia construída na estrutura de Anúncios.....	50
Figura 24: Estrutura Hierárquica.....	51
Figura 22: Como Ad 2 possui mais informações, as atualizações de Ad 1 e Campanha 1 são feitas de tal forma a se aproximarem de Ad 2.	52
Figura 23: Pares formados na etapa de agregação dos dados.	54
Figura 27: Exemplo de um Publisher que agrega dados de seus sites.	55
Figura 28: Exemplo das atualizações realizadas. Primeiro atualiza-se nós folhas, depois nós superiores e por fim novamente nós folhas.....	56
Figura 26: Exemplos dos nós sendo utilizados na etapa de predição.....	57
Figura 27: Composição do modelo híbrido.....	58
Figura 31: Estrutura Hierárquica com melhoria de Performance.....	64
Figura 29: Divisão empregada no modelo dinâmico 1. Em cada divisão do dia há um registro de como sites e publishers interagiram com os anúncios.	66
Figura 30: Modelo semelhante ao MDTH1 porém a divisão do dia é feita com maior precisão.	68
Figura 31: Apenas nós folhas possuem estrutura dinâmica temporal	69
Figura 32: O nó superior registra uma mistura entre as tendências temporais de cada site que possui. .	70
Figura 33: Exemplo dos dados fornecidos pela empresa Lomadee.....	71
Figura 34 : Erro log likelihood em função da dimensão f dos vetores.....	74
Figura 38 : Erro loglikelihood em função do total de iterações na etapa de aprendizagem.....	74
Figura 39: Desempenho preditor Baseline em função do parâmetro α	75
Figura 37: Resultados em todas abordagens implementadas.....	76
Figura 38: Melhores resultados obtidos em cada estratégia dinâmica.....	79

Lista de Símbolos e Abreviaturas

- **CPM:** Custo por Mil Impressões.
- **CPC:** Custo Por Click.
- **CPT:** Custo Por Transação.
- **CPA:** Custo Por Ação.
- **CTR:** Do inglês, Click Through Rate, ou Taxa de Conversão de Clicks.
- **SVD:** Singular Value Decomposition
- **$P(\Theta)$:** Probabilidade do evento Θ ocorrer.
- **PCA:** Principal Analysis Component.
- **U:** Valores singulares da esquerda.
- **V^* :** Valores singulares da direita.
- **Σ :** Matriz de auto-valores.
- **$\tilde{\Sigma}$:** Matriz reduzida de auto-valores.
- **RSME:** Root Square Mean Estimation.
- **$\mathcal{L}$:** Verossimilhança.
- **$M_i(\text{Dirichlet})$:** Quantidade de observações do i-ésimo atributo dos dados.
- **$M(\text{Dirichlet})$:** Total de observações de todos atributos dos dados.
- **$\alpha(\text{Dirichlet})$:** $\alpha \in \mathbb{R}$. Parâmetro da distribuição à priori utilizado no preditor Bayesiano.
- **CTR_{ij}^{MLE} :** Estimação de máxima verossimilhança de CTR do anúncio j impresso no site i.
- **V_{ij} :** Total de impressões que o anúncio j teve no site i.
- **C_{ij} :** Quantidade total de Clicks que o anúncio j obteve quando impresso V_{ij} vezes no site i.
- **$\widehat{CTR}$:** Valor estimado de CTR.
- **$\widehat{CTR}(\Theta)$:** Valor estimado de CTR em função de Θ .
- **X_{ij} :** Valor binário. Vale 1 se ocorreu um Click quando o anúncio j foi impresso no site i.
- **$\mathcal{O}$:** Representa todos os pontos utilizados na fase de treinamento dos preditores.
- **$\alpha(\text{SVD})$:** Vetor derivado de técnicas de SVD associados aos sites e publicadores. $\alpha \in \mathbb{R}^f$.

- **β (SVD)**: Vetor derivado de técnicas de SVD associados aos anúncios, campanhas e anunciantes. $\beta \in \mathbb{R}^f$.
- **f** : Dimensão dos vetores SVD. $f \in \mathbb{R}$.
- **Ω** : Função de Regularização.
- **J** : Função Custo ou Função Objetivo.
- **λ** : Valor de regularização utilizado no gradiente estocástico. $\lambda \in \mathbb{R}$.
- $\frac{\partial J}{\partial \alpha_{i_k}}$: Derivada da função custo em relação ao vetor α do site i no ponto k , onde $k \in [0, f]$.
- **ξ** : Taxa de aprendizagem utilizada no gradiente estocástico.
- **e** : Número de Euler.
- **CV** : Cross-Validation, ou Validação Cruzada.
- **w** : Vetor de pesos utilizados no modelo Side-Information.
- **x_{ij}** : Atributos intrínsecos de um dado elemento da estrutura de páginas i e anúncios j .
- **$\mathcal{N}$** : Distribuição Normal.
- **$\mathcal{H}^P$** : Estrutura Hierárquica de Páginas.
- **$\mathcal{H}^A$** : Estrutura Hierárquica de Anúncios.
- **$\text{Par}(i)$** : Nós pais de i .
- **$\|\alpha\|$** : Norma do vetor α .
- **G** : Estrutura hierárquica que combina as estruturas de páginas e anúncios.
- **C^{Agg}** : Quantidade agregada de Clicks.
- **$MHDT1$** : Modelo Híbrido Dinâmico Temporal 1.
- **ϵ** : Fator utilizado na inicialização de vetores. $\epsilon \in \mathbb{R}$.

Sumário

1	Introdução.....	13
1.1	Objetivos	17
1.2	Organização do Texto	19
2	Revisão Bibliográfica	20
2.1	Trabalhos Relacionados	20
2.2	Processamento dos Dados	22
2.3	Fatoração de Matriz	25
2.4	Sistemas de Recomendação	28
2.4.1	Filtragem Colaborativa.....	31
2.5	Gradiente Estocástico Descendente	32
2.6	Avaliação dos Preditores.....	34
3	Algoritmo Implementado.....	38
3.1	Desenvolvimento do Preditor de CTR.....	38
3.1.1	Fatoração Baseada em Pesos de Confiança.....	41
3.1.2	Fatoração Baseada em Confiança com Side Information	46
3.1.3	Modelo Hierárquico	48
4	Resultados.....	71
4.1	Experimentos	71
4.2	Resultados Obtidos	73
5	Conclusão	81
5.1	Sumário	81
5.2	Trabalhos Futuros	82
6	Referências Bibliográficas	83
7	Apêndice	88

1 Introdução

Com o advento da internet, a sua presença está cada vez maior no cotidiano das pessoas. Como consequência, lojas e fornecedores de serviços obtiveram uma nova forma de publicar seus produtos, feita através da impressão de anúncios em diferentes sites.

Trata-se da interação de três participantes: os publicadores, que são no geral aqueles que possuem sites visitados por um grupo de indivíduos, os anunciantes, que são aqueles que criam anúncios de seus produtos ou serviços para serem publicados e, por fim, o usuário final, que será exposto aos anúncios impressos nos sites que visita. Cada participante tem seu próprio interesse:

- O anunciante quer aumentar seu faturamento.
- O usuário quer ser exposto a anúncios úteis que lhe despertem interesse.
- O publicador quer aumentar seus ganhos através dos clicks dos usuários.

A forma como os anúncios são expostos pode variar. Os mesmos podem aparecer logo após uma consulta em uma ferramenta de busca como Google ou Bing conforme ilustra a Figura 1. Uma outra possibilidade é quando aparecem impressos em uma área de um dado site qualquer, como ilustra a Figura 2. No caso, a Figura 2 ilustra o site portal Exame¹ que dedica parte de sua página no canto inferior direito para a impressão de anúncios. A distinção entre a busca patrocinada (anúncios impressos após uma pesquisa, Figura 1) e sites imprimindo anúncios online em seus domínios (Figura 2) é importante pois para cada cenário utiliza-se técnicas específicas para selecionar os anúncios que serão impressos. Na busca patrocinada, pode-se combinar as palavras que o usuário digitou na ferramenta de busca com as palavras que o anunciante escolheu para descrever seu anúncio ou características intrínsecas dos anúncios como a sua área, animação, localização e assim por diante [1]. Outras variáveis também são consideradas como o quanto o publicador (que é o dono do site) irá ganhar caso o usuário clique na propaganda exibida.

¹ <http://exame.abril.com.br/>

Google whakatane accommodation

About 444,000 results (0.14 seconds)

Everything
Images
Videos
News
Places
More

Whakatane
Change location

The web
Pages from New Zealand

More search tools

Travel dates:
Check in: 31/07
Check out: 01/08

Whakatane Motels | aatravel.co.nz
www.aastravel.co.nz
Book accommodation online & get instant confirmation now!

Whakatane Hotels | jasons.co.nz
www.jasons.co.nz/Whakatane-hotels
Wide range of Whakatane hotels Book online, by phone or email now!
Hutner Springs Accommodation - Wanaka Accommodation

Whakatane Motels ? | Whakatane info
www.whakatane.info
All Your Whakatane Motels. Right Here!

Accommodation | Whakatane New Zealand
www.whakatane.com/accommodation - Cached
Whakatane has world-class accommodation from beachside B&B's to coastal camping grounds, from luxury motels with finest facilities to great backpackers.

Whakatane Accommodation
www.aastravel.co.nz > Accommodation - Cached
New Zealand Whakatane Accommodation. If you are looking for comprehensive travel information for Whakatane Accommodation including Hotels, Motels, ...

Whakatane Accommodation - Jasons
www.jasons.co.nz > New Zealand > North Island > Bay of Plenty - Cached
Results 1 - 20 of 22 - Visit Jasons for Whakatane Accommodation. Book and save on a wide range of Whakatane accommodation options including Whakatane hotels, ...

Whakatane Accommodation - Pacific Coast Motor Lodge
www.pacificcoastwhakatane.co.nz/ - Cached
14 luxury designer, spacious, ground-floor 4-star motel accommodation. Great for relaxing holiday, business, conference, family holiday or honeymoon.
41 Landing Road, Whakatane
(07) 308 0100
travelpod.com (349) · tripadvisor.com (5) · aastravel.co.nz (4) · tourism.net.nz (1)

Hotels in Whakatane?
www.amber-court.co.nz/whakatane
Looking to stay in Whakatane?
Get Quiet & Secluded Accommodation.

Hotels in Whakatane
www.booking.com/Whakatane-hotels
Book your Hotel in Whakatane online
No reservation costs. Great rates

Beachpoint Resort
Apartments
www.agoda.com/New_Zealand
Enjoy Huge Discounts.
More Hotels Available Now!

Luxury Bay of Plenty B&B
www.alfrescobedandbreakfast.co.nz
Friendly hosts and beautiful rooms
Whakatane Bed & Breakfast

Hotels Whakatane
nz.hotelium.com/Whakatane
We have complete offer of hotels.
Book online. Best Price Guaranteed!

Travel Me Property
www.travelsme.co.nz/realestate
Houses for sale & rent. Thousands
of listings. Find places near you!

361 reviews
Place page

Figura 1: Exemplo dos anúncios em uma busca patrocinada.

Deputados | 07:17
Câmara elege nesta segunda seu novo presidente

Oportunidades | 06:00
29 empresas buscam trainees

Mercados | 23:28
Veja o que importa na agenda do investidor hoje

Ações | 07:00
Os melhores dividendos

Fusão em 2014 | 11:06
Fiat e Chrysler se unem

Indicações
As melhores ações de consumo do ano

Preços dos seguros de carros em SP

Mercado
30 profissões em alta em 2013

Como se dar bem na dinâmica de grupo

Internet
Hackers atacam Washington Post

7 apps para iPhone, iPad e Android

Cachorro-queixe
Geneal custa a partir de R\$ 104 mil

10 franquias em alta neste ano

Top News | Últimas | Brasil | Empresas | PME

Cotações Bovespa EXAME.com

Focus | D Mercado mantém previsão do PIB, mas sobe inflação | 08:48

Parceria | Bradesco e GP Investimentos ficam sócios | 08:42

Mudanças | Abilio troca dois membros do conselho do Pão de Açúcar | 08:13

Trânsito | Lei seca mais rígida faz crescer número de prisões | 07:52

México | Total de mortos em explosão da Pemex vai a 36 | 07:30

Lider | PMDB elege Eduardo Cunha para liderar bancada na Câmara | 07:12

Mostrar 12 38

Todas as notícias

Vídeos & Infográficos

As mais em EXAME.com

60 minutos 24 horas 7 dias 30 dias

20 ideias de negócios promissoras

30 profissões que estarão em alta em 2013

Rotina de executivos inclui muito tempo perdido

Os melhores comerciais do Super Bowl 2013

29 empresas que lecionam estagiarismo e trainees

Só em EXAME

As + do Site

Nas bancas | 06 de fevereiro de 2013

EXAME ONDE INVESTIR 2013

Crise Coca-Cola assume: engorda mesmo. E daí?

Sumário da edição

Edições anteriores

ASSINE EXAME

Nas bancas | Janeiro de 2013

PLA Quem deve e não paga Há clientes que agem como se a culpa de não terem o dinheiro fosse minha

Sumário da edição

Edições anteriores

ASSINE EXAME PME

Figura 2: Exemplo de anúncios impressos no site da Exame.com.

Há ainda outras formas de se exibir anúncios na internet, como por exemplo, os que são apresentados durante a exposição de um vídeo (como no site de vídeos You Tube²) ou apresentações. Esta pesquisa focará em anúncios impressos nos próprios sites, ou seja, quando o site dedica uma parte de seu espaço para a impressão de anúncios virtuais.

O marketing virtual é uma área com grande potencial de crescimento para os próximos anos. A Figura 3 ilustra os gastos mundiais em anúncios online :

Figura 3: Gastos em anúncios online ao redor do mundo em comparação com gastos totais.³

A projeção de investimentos em anúncios online nos Estados Unidos para os próximos anos também é considerável:

² www.youtube.com

³ <http://www.go-gulf.com/blog/online-ad-spending/>

Figura 4: Projeção de investimentos em anúncios online nos Estados Unidos até 2016³

Outros meios de comunicação estão perdendo espaço para a publicidade online, revelando que a tendência para os próximos anos são expressivos investimentos nesta área⁴. Empresas e fornecedores de serviços também apresentam faturamentos consideráveis através do marketing virtual [2]. Com isto, estão cada vez mais investindo recursos para a pesquisa que visa o desenvolvimento de preditores para avaliar a probabilidade que seus anúncios têm de serem clicados pelos usuários. Quanto melhor a acurácia, melhor para a publicação do anúncio que consegue atrair mais clientes para sua loja e melhor para o publicador que ganha mais dinheiro com os clicks em seu site. Por fim, o cliente final se beneficia ao ser exposto a itens que são de seu interesse.

Os anunciantes pagam três tipos de transações, que podem ser negociadas com o publicador:

- CPM: Custo por mil impressões. É normalmente utilizado em banners gráficos. É possível que no acordo entre publicador e anunciante seja combinado uma entrega garantida de impressões, ou seja, o publicador deve imprimir uma quantidade pré-determinada de um dado banner independente da sua probabilidade de ser clicado. Este fator deve ser levado em conta na hora de otimizar os ganhos de todos envolvidos.
- CPC: Custo por click, ou seja, o dinheiro que o anunciante paga ao publicador por cada usuário que clica no seu banner.

⁴ <http://www.go-gulf.com/blog/online-ad-spending/>

- CPT ou CPA: Custo por Transação ou Custo por Ação, refere-se ao quanto o anunciante paga caso o usuário entre na loja e realize uma compra.

O desafio que surge então é encontrar o melhor anúncio que pode ser impresso para um dado usuário aumentando o máximo possível a probabilidade de que o mesmo goste do que está vendo e clique no anúncio. Este parâmetro é conhecido como **CTR** ou “Click-Through Rate”. O conjunto de técnicas que estuda como lidar com tal problema é conhecido como “*Computational Advertising*” [3] abrangendo conceitos como:

- Recuperação de Informação (Information Retrieval) [4]
- Big Data
- Modelamento Estatístico
- Aprendizagem de Máquinas
- Microeconomia
- Teoria dos Jogos
- Classificação
- Otimização
- Sistemas de Recomendação

1.1 Objetivos

O objetivo deste trabalho é construir um preditor de CTR. Dados reais da empresa de gerenciamento de anúncios virtuais Lomadee⁵ serão utilizados na pesquisa. De forma simplificada, a empresa Lomadee possui um conjunto de anúncios que pode escolher para serem impressos em diversos sites e a estimação do valor CTR é importante para que este processo de decisão possa ser feito [5].

⁵ <http://br.lomadee.com/>

CTR: 5%

SOCIEDADE DE NEGÓCIOS
O amigo com quem você pode contar.

Bradesco

[Clique e saiba mais](#)

CTR: 2.1%

Life companion
GALAXY S4

CTR: 0.8%

Nortasuna.
KELER

São Paulo | Brasil | Internacional | Saúde | Ciência | Educação | Planeta | Cultura | Paladar | Ásia | Blogs | Colunistas | Vídeos | Fotos | Infográficos | Tópicos | Horóscopo

Saint Paul Eleita novamente uma das melhores do mundo. **FT** Executive Education Ranking 2013 [CLIQUE AQUI](#)

ÚLTIMA HORA **Brasil** ganha até 100 bilhões com redução do ICMS

MP dos Portos: Senado tem até a meia-noite para aprovar proposta
Após 41 horas, Câmara aprovou medida; texto cria novas regras para terminais públicos e privados
+ Texto ficou próximo do desejo do governo
+ FOTOS: galinhada na votação da MP dos Portos
+ Governo já prepara vetos à MP dos Portos
+ Návios de emergência confundem até especialistas
+ ESPECIAL: entenda a MP dos Portos

Una mano Amarilla

Imprensa repercute erros de juiz
Corinthians x Boca ganha destaque internacional

TV ESTADÃO

SOCIEDADE DE NEGÓCIOS
O amigo com quem você pode contar.

Bradesco

[Clique e saiba mais](#)

Figura 5: Processo de decisão simplificado da empresa Lomadee que escolhe anúncios com o maior CTR para sites requisitantes.

Dados reais entre dezembro de 2011 a janeiro de 2012 foram utilizados para o desenvolvimento dos algoritmos e validação dos conceitos que serão discutidos nesta dissertação.

Diversas abordagens já foram discutidas na literatura sobre o problema de predição de CTR. Dentre elas, há modelos que consideram apenas a localização do anúncio no site que o imprime enquanto que outros empregam abordagens estatísticas para realizar a predição. O modelo empregado neste trabalho é baseado na pesquisa de MENON [6], que até 2012 conseguiu os resultados com melhor acurácia de predição e cuja estratégia é baseada em técnicas de fatoração de matriz integradas a diversos modelos estatísticos que serão melhores discutidos a seguir.

Um preditor baseline baseado em estimação bayesiana com perda quadrática modelada com probabilidades à priori como função de Dirichlet foi implementado para se comparar os diversos resultados obtidos, o que auxilia na avaliação do desempenho do algoritmo.

Por fim, foi implementada uma estratégia dinâmica em função do tempo comprovando-se que os preditores foram capazes de reduzir ainda mais o erro verossimilhança na base de teste, sendo estes portanto os melhores resultados obtidos até agora na literatura.

1.2 Organização do Texto

No Capítulo 2, uma revisão bibliográfica dos principais conceitos utilizados neste trabalho é feita para facilitar o entendimento do leitor sobre o assunto a ser desenvolvido. O Capítulo 3 discute o passo a passo da implementação do algoritmo de predição de CTR proposto, os equacionamentos necessários e a forma como os dados são manipulados. O Capítulo 4 comenta experimentos e os resultados obtidos, tanto dos preditores estáticos quanto dos dinâmicos, sendo que comparações e conclusões são discutidas ao longo do capítulo. Conclui-se o trabalho com o Capítulo 5 onde um breve sumário é apresentado e discussões para trabalhos futuros é feita. O Apêndice mostra algumas abordagens previamente testadas porém não consideradas na formulação do algoritmo final implementado.

2 Revisão Bibliográfica

A seguir, descreve-se os conceitos e técnicas utilizadas para implementação do algoritmo para predição de taxa de conversão de clicks, ou CTR. Começa-se com uma breve descrição do que foi pesquisado na literatura sobre o tema de construção de preditores, abordando as principais estratégias que já foram pesquisadas e testadas. Segue-se então com uma discussão sobre técnicas relacionadas ao processamento dos dados que servem como fonte de informação para a construção dos preditores, fornecendo uma visão geral do que é necessário fazer para preparar os dados para que os mesmos sejam utilizados com melhor eficiência pelos algoritmos.

Discute-se então idéias sobre fatoração de matriz, principalmente a técnica SVD (“Singular Value Decomposition”) a qual é extensamente utilizada nos preditores implementados.

Uma discussão sobre sistemas de recomendação também é feita dado que alguns conceitos desta área serão utilizados pelos preditores.

2.1 Trabalhos Relacionados

A literatura sobre preditores de CTR é extensa e diversas estratégias já foram testadas e discutidas [7]. WANG discute sobre alguns dos desafios para se construir estes preditores, apontando para as duas principais características do problema [8]. O primeiro é o fato de que um anúncio pode ser impresso apenas algumas vezes em um site e receber uma quantidade considerável de cliques. Como exemplo, uma propaganda pode ser impressa apenas uma vez e receber um clique neste momento. O CTR observado seria de 100% o que é muito alto para o que normalmente se observa na prática, revelando que observações tendenciosas podem acontecer ao longo do tempo.

Outro desafio é que na grande maioria das vezes têm-se milhares de impressões de um anúncio mas apenas alguns cliques, revelando que o CTR normalmente está entre 0 e 0.1%. WANG sugere, para lidar com este problema, um modelo com duas estratégias, as quais são

combinadas para formar uma abordagem só. Na primeira estratégia, um modelo que utiliza a estrutura hierárquica dos dados é implementado a fim de se obter de um site popular o que provavelmente acontecerá nos outros sites do mesmo publicador (esta estrutura será melhor discutida na próxima seção). A outra abordagem envolve a implementação de um modelo com decaimento exponencial em função do tempo entre as diferentes conversões observadas para tornar as previsões menos tendenciosas.

Há outras formas de se lidar com o problema. CHEN descreve um modelo comportamental dos usuários usando modelos de regressão linear de Poisson [9]. GLUHOVSKY também usa modelos de regressão linear porém considerando uma regressão monotônica tripla a qual usa não apenas a posição do anúncio, mas também o preço pago pelo mesmo ao ser clicado e as posições em que o anúncio é clicado [10].

O sistema de busca Bing, desenvolvido pela Microsoft, trabalha com um sistema de predição Bayesiano online. Mais precisamente, um modelo Bayesiano de regressão Probit [11] o qual obteve os melhores resultados na área até 2010.

ROSALES implementou um modelo para calcular [12]:

$$P(Conversion|Click, Context)$$

para analisar não somente o que acontece antes do click mas também como o usuário se comporta ao comprar o produto desejado e assim extrair informações sobre gostos e tendências.

Para análises temporais dos dados, XU desenvolveu um modelo para rastrear os cliques como sequências de eventos correlacionados, mostrando que tal abordagem pode melhorar a acurácia dos preditores empregados [13].

2.2 Processamento dos Dados

Quando se trabalha com o desenvolvimento de preditores, há basicamente dois tipos de dados que podem ser utilizados como fonte de informação [14], [15], [16]:

- *Feedback Explícito*: são dados nos quais o usuário revela o quanto gostou de determinado item, como no caso da competição Netflix [17] no qual usuários votavam em filmes entre 1 e 5 estrelas.
- *Feedback Implícito*: neste caso é possível apenas extrair padrões de comportamento, como sites visitados, produtos visualizados, clicks efetivados e outras fontes de dados que um usuário pode indicar [18], [18].

A distinção é importante pois as estratégias desenvolvidas variam de acordo com o tipo de dados alimentando a etapa de aprendizagem dos preditores. Para o problema de predição de CTR, os dados são implícitos e portanto não se sabe exatamente se os usuários gostam ou não dos anúncios que clicam ou deixam de clicar.

Os dados da empresa Lomadee contém informações dos publicadores dos sites, anunciantes e quantos cliques ocorreram dentre um total de impressões realizadas, fornecendo portanto apenas o feedback implícito dos anúncios que despertam interesse nos usuários.

De posse dos dados, as principais técnicas de Data Mining são aplicadas [19], o qual consiste de três etapas que serão seguidas na aplicação do trabalho: *pré-processamento*, *análise* e *interpretação*. A Figura 7 a seguir ilustra melhor todos os conceitos normalmente empregados na área:

Figura 6 : Idéia geral de um processo de Data Mining aplicado na construção de preditores. [31].

A etapa de pré-processamento consiste basicamente em organizar os dados. Nesta etapa, podem ocorrer diversos processos de amostragem o que normalmente ajuda a reduzir ou facilitar as futuras etapas que utilizarão estes dados. Há diversas formas de se amostrar e é importante levar em consideração o tipo de amostragem a ser feita, dado que até mesmo sua correta manipulação pode ser a diferença entre um algoritmo eficiente e outro que não funciona bem [20], [21]. Dentre as possibilidades, temos:

- Amostragem Aleatória (com e sem reposição).
- Amostragem Estratificada.

A amostragem estratificada consiste em selecionar os elementos do universo de estudo dividindo-os em estratos conhecidos.

- Validação Cruzada k-Fold.

Este processo de teste consiste em dividir os dados em k sub-grupos. Destas partições, $k-1$ são utilizadas para a etapa de aprendizagem dos algoritmos e k -ésimo grupo é utilizado para a etapa de validação, na qual os algoritmos treinados são testados em função de alguma métrica

de desempenho. Realiza-se a operação nos k sub-grupos e o resultado é combinado em um valor final para determinar a eficiência do algoritmo, o que normalmente é feito através da média dos desempenhos em cada partição. A Figura 7 ilustra melhor o conceito:

Figura 7: Exemplo de amostragem por validação cruzada k-Fold.

Nos dados de treinamento, diversos parâmetros são configurados como a taxa de aprendizagem e taxas de regularização, parâmetros utilizados no processo de aprendizagem do algoritmo. Um teste então é feito na validação cruzada para se encontrar o melhor preditor da base de treinamento. Uma parcela dos dados é separada para o teste final, o que se assemelha aos testes de predição que os preditores estariam sujeitos quando postos para funcionar na prática.

- Super Amostragem.

Técnica de amostragem utilizada em dados desbalanceados (presença de um grupo consideravelmente maior que outro) na qual amostra-se dados repetindo-se pontos do grupo com menor participação total nos dados [22].

- Sub Amostragem.

Semelhante à idéia de super amostragem porém alguns pontos do grupo de maior presença nos dados são removidos, restabelecendo-se assim um maior equilíbrio entre as classes dos dados.

Após esta etapa, diferentes técnicas podem ser aplicadas para se modelar os acontecimentos observados, sendo as mais populares mostradas na Figura 7. Os resultados então são testados em conjuntos de testes para a interpretação e avaliação final.

2.3 Fatoração de Matriz

Técnicas de fatoração de matriz ganharam popularidade recentemente para a construção de preditores devido aos desempenhos superiores em relação a outras técnicas já conhecidas [17], [23], [24]. Há duas grandes abordagens neste tema. Uma delas é a técnica conhecida por PCA (“Principal Components Analysis”). Trata-se de uma técnica que extrai os auto-vetores e auto-valores de uma matriz e então elimina alguns pares de auto-valor e auto-vetor (elimina-se uma parcela dos menores auto-valores encontrados). Com os auto-vetores resultantes, é possível montar uma aproximação para a matriz original. Melhores referências podem ser encontradas em [26] e [25]. Outra técnica é a “Singular Value Decomposition” ou SVD e devido à sua importância para este trabalho, uma explicação mais aprofundada é realizada a seguir.

Singular Value Decomposition (SVD)

SVD é uma técnica de fatoração de matriz (real ou complexa) cujo resultado pode ser aplicado em diversas áreas, desde processamento de sinais, como simplificar imagens para facilitar seu processamento ou em áreas como estatística [26]. Esta técnica ganhou popularidade por apresentar, em problemas de predição, resultados superiores em relação a outras ferramentas estatísticas [24].

Formalmente, a decomposição de uma matriz A $m \times n$ qualquer é dada por:

$$A = U\Sigma V^*.$$

Equação 2.1

Ilustrando-se a decomposição:

Figura 8: Exemplo de uma matriz A qualquer representada pela multiplicação de 3 matrizes [31].

U é uma matriz unitária $n \times r$, Σ é uma matriz retangular diagonal com números não negativos e V^* (transposta de V) é uma matriz $m \times r$ unitária também. Os valores em Σ são conhecidos por valores singulares de A . As colunas de V são chamadas de vetores singulares da direita e as colunas de U os vetores singulares da esquerda de A , tal que:

- Os vetores em U são os auto-vetores da matriz AA^* .
- Os vetores de V são os auto-vetores da matriz A^*A .
- Os valores singulares da matriz Σ são a raiz quadrada dos auto-valores de ambos AA^* e A^*A .

Há diversas aplicações do processo:

- Rotação e Redimensionamento

A transformação pode ser vista como a mudança de base do espaço vetorial que transforma uma circunferência em uma elipse deslocada, conforme ilustra a Figura 12 a seguir:

Figura 9: Transformações resultantes das operações de multiplicação das matrizes ortogonais V e U.

- Cálculo de Matriz Pseudo-Inversa.

Considerando-se a Equação (2.1), a pseudo-inversa de A é dada por:

$$A^+ = U\Sigma^+V^*.$$

Equação 2.2

Onde Σ^+ representa a pseudo-inversa de Σ , obtida através da transposta dos elementos recíprocos dos elementos não nulos na diagonal de Σ .

- Resolução de Equações Lineares homogêneas.
- Aproximação de Matriz.

Esta é uma das aplicações mais importantes da técnica SVD para a construção de preditores. A idéia é obter uma representação truncada de A que melhor se aproxima da mesma, cuja minimização é feita pela norma Frobenius entre a diferença da matriz aproximada com a original. Formalmente [23]:

$$\tilde{A} = U\tilde{\Sigma}V^*.$$

Equação 2.3

Onde $\tilde{\Sigma}$ é equivalente a Σ porém com alguns auto-valores removidos da matriz, o que equivale a reduzir a dimensão da matriz.

Alguns problemas podem surgir para a decomposição de A , porém diversas técnicas foram criadas, testadas e analisadas para se contornar situações tais como a alta esparsialidade da matriz ou dimensões grandes o suficientes para não serem computáveis com o hardware disponível [27]. Mais especificamente, o trabalho de KOREN [24] descreve métodos para se obter os vetores U e V através da aplicação de técnicas de gradiente estocástico, sendo esta mesma técnica a que obteve melhor desempenho na competição NetFlix de locação de filmes [17]. A partir de uma matriz contendo a relação entre itens e usuários, decompõe-se em uma dimensão reduzida em função de “conceitos” e o quanto cada elemento de um dado vetor se relaciona a este conceito, também considerando a “força” do elemento. A técnica é utilizada também para se obter variáveis latentes em alguns processos, representados pelos conceitos extraídos dos dados [30].

2.4 Sistemas de Recomendação

Sistemas de recomendação são sistemas implementados para rastrear as tendências e gostos de usuários para recomendar itens de um grupo extenso de possibilidades [28].

Para o desenvolvimento destes sistemas, um conjunto diferente de técnicas e classificadores podem ser utilizados [29], [30]. Basicamente, um sistema de recomendação é desenvolvido com as seguintes propostas [28], [31]:

1. Ensejar um aumento no número de itens vendidos. A idéia é que o sistema de recomendação deve filtrar de uma grande quantidade disponível de produtos aqueles que mais irão despertar o interesse dos usuários. Caso este objetivo seja cumprido consequentemente aumenta-se o número de itens vendidos [32].
2. Diversificar vendas. Apesar de se tratar de um desafio para sistemas de recomendação, um de seus maiores objetivos é conseguir aumentar a diversidade dos itens vendidos. Trata-se de um desafio pois os preditores normalmente são treinados em função de

uma base de dados de treinamento, porém tal base possui tendências sobressalentes, como por exemplo itens mais populares têm mais registros que os não populares, portanto é de se esperar que estes mesmos produtos sejam recomendados mais vezes. Deve-se tratar com cuidado as informações de produtos pouco vendidos ou clicados para que se possa aumentar suas vendas no futuro.

3. Melhorar serviço oferecido aos clientes. Como dito anteriormente, sistemas de recomendação foram desenvolvidos para auxiliar usuários no processo de compra, melhorando o serviço oferecido.
4. Aumentar a fidelidade dos clientes.
5. Melhor entendimento do que os clientes querem. Através dos preditores é possível de se observar padrões nos dados antes invisíveis com análises menos aprofundadas, como tendências em função do tempo ou características intrínsecas dos usuários.

Alguns domínios importantes para sua aplicação envolvem:

- Entretenimento: recomendação de filmes, música, textos e etc.
- Conteúdo: fornecer notícias personalizadas de acordo com o gosto do usuário.
- E-commerce: recomendações de compras de diversos produtos variando desde livros até eletrônicos e diversos outros itens.
- Serviços: recomendação de serviços como hotelaria, consultoria, aluguel de casas e tudo o mais.

Recentemente, não só a acurácia destes sistemas têm sido levada em consideração mas também a forma como as recomendações são apresentadas, o que aumenta a confiança dos usuários no que lhes está sendo sugerido, [33], [14], [34], [35], [36].

A figura a seguir resume todos os tipos de sistema de recomendação citados na literatura [37]:

Figura 10: Técnicas de Sistema de Recomendação.

Sistemas baseado em conteúdo [38] já foram bastante populares; visavam principalmente rastrear características dos itens (como extrair informações sobre um filme) e combinar estas informações com os gostos dos usuários [39], [40], [15].

Sistemas demográficos rastreavam perfis dos usuários (casado, solteiro, homem, mulher) para melhor recomendar itens [41]. Os baseados em conhecimento partem de alguma informação oferecida pelo cliente, como algum item que está procurando ou precisando [42], [21]. Sistemas mais recentes surgiram com a popularização de redes sociais. Trata-se dos sistemas baseados em comunidade, na qual informações obtidas principalmente de redes sociais são utilizadas para a construção de preditores [43], [44], [45]. Sistemas híbridos funcionam misturando todas as abordagens existentes [46], tal que o ponto fraco de um tipo de sistema de recomendação é compensado pelo ponto forte de outro [47], [48], [49], [50]. De especial interesse para este trabalho é o sistema baseado em filtragem colaborativa, dado que o mesmo será empregado na solução final do algoritmo proposto. Uma discussão mais aprofundada é feita a seguir.

2.4.1 Filtragem Colaborativa

É uma das principais vertentes em relação a sistemas de recomendação [21], [51], [33]. São basicamente métodos que rastreiam padrões no comportamento dos usuários sem precisar necessariamente reconhecer quem eles são ou qualquer outra informação de identificação [52], sendo esta uma das principais vantagens do método [28], [53], [54].

O registro histórico do comportamento dos usuários normalmente contém informações sobre as preferências ou ações dos mesmos interagindo com os itens de uma loja ou algo relacionado [55], [56]. Estes itens podem ser produtos, serviços, informações, músicas e diversas outras categorias [53]. Os dados são utilizados para treinar algoritmos preditores e os resultados são testados em uma base com informações ainda desconhecidas pelo algoritmo para assim ser possível avaliar o quão bem o mesmo prevê ações no futuro [20], [57], [58], [59].

Os melhores modelos registrados nesta área são os que se baseiam em técnicas de fatoração de matriz como a SVD [28]. Como exemplo, em 2007, a empresa de alocação de DVDs online Netflix⁶ ofereceu 1 milhão de dólares para quem conseguisse melhorar seus preditores que previam quantas estrelas os usuários dariam aos filmes que assistiam [60]. A técnica vencedora baseia-se em SVD com inclusão de dinâmica temporal, cujas predições são modeladas por:

$$\hat{r}_{ui}(t_{ui}) = \mu + b_i(t_{ui}) + b_u(t_{ui}) + q_i^T \left(p_u(t_{ui}) + |R(u)|^{-\frac{1}{2}} \sum_{j \in R(u)} y_j \right) \quad \text{Equação 2.4}$$

r_{ui} é a predição do quanto o usuário u dará em estrelas ao filme i . μ representa a média geral dos votos (este valor vale na base Netflix 3.7 estrelas), b_i e b_u são tendências de votos para o item i e para o usuário u (se o usuário “ u ” normalmente dá três estrelas aos filmes que assiste então b_u vale três). Os valores de b_* são variáveis em função do tempo. q_i representa um vetor SVD de dimensão f que modela fatores latentes do filme i . Por fim, os parâmetros entre

⁶ <https://signup.netflix.com/>

parênteses da Equação 2.4 modelam os vetores latentes associados aos usuários. p_u é um vetor que modela a relação que cada usuário tem com os conceitos do vetor q_i e que é variável em função do tempo, contabilizando o fato de que os gostos dos usuários mudam no longo prazo. Os vetores y_j são vetores associados aos filmes que cada usuário assistiu, contabilizando para o fato de que cada filme assistido revela informações latentes sobre o gosto dos clientes. Cada parâmetro é determinado através da aplicação de técnicas de gradiente estocástico descendente [25], cujo conceito será melhor explicado na sessão seguinte. O desempenho dos algoritmos é determinado pela métrica RSME (“Root Square Mean Error”, ou raiz do erro médio quadrático) cuja fórmula é mostrada na sub-sessão sobre avaliação dos preditores a seguir.

2.5 Gradiente Estocástico Descendente

É uma técnica utilizada para minimizar funções conhecidas por “função custo” ou “função objetivo” que é escrita como uma somatória de funções que são diferenciáveis, sendo escritas da seguinte forma:

$$Q(w) = \sum_{i=1}^m Q_i(w). \quad \text{Equação 2.5}$$

w são parâmetros que sofrem constantes atualizações visando a redução do valor total da somatória da Equação 2.5. Q_i é uma função associada à i -ésima observação no conjunto de dados de treinamento.

Para se reduzir a função custo, os parâmetros w são atualizados como se segue:

$$w := w - \alpha \nabla Q(w) = w - \alpha \sum_{i=1}^m \nabla Q_i(w). \quad \text{Equação 2.6}$$

onde α é conhecido como “taxa de aprendizagem” e serve para regular o nível da adaptação a ser feita em w . O valor $\nabla Q(w)$ representa o gradiente da função Q em relação a w .

O processo varre a base inteira de treinamento atualizando os valores de w . Porém, o processo se repete para garantir melhor convergência a um ponto de mínimo na função custo. O seguinte algoritmo pode ser descrito [61]:

- 1- Iniciar um valor aleatório para w e determinar uma taxa de aprendizagem α .
- 2- Repetir até um mínimo ser atingido:
 - a. Obter uma base aleatória de treinamento.
 - b. Para $i = 1, 2, \dots, m$ fazer:
 - i. $w := w - \alpha \nabla Q_i(w)$.

A Figura 11 a seguir ilustra o que normalmente se observa quando o processo de gradiente estocástico sendo executado. No eixo horizontal está o número de iterações que já foram executadas e no eixo vertical está o valor total da função custo, o qual decresce no longo prazo indicando que os vetores estão melhor se adaptando aos dados sendo observados:

Figura 11: Custo da função objetivo em função da quantidade de iterações calculadas.

Como é possível observar, o custo total diminui e por vezes aumenta, voltando mais tarde a um ponto inferior. O importante é confirmar se no longo prazo a função está diminuindo em valor [25].

É possível ainda incluir fatores de regularização na função objetivo, técnica utilizada para diminuir o efeito de overfitting. Este problema ocorre quando os preditores se adaptam exageradamente aos dados de treinamento e perdem a habilidade de predição em dados futuros. Para incluir o efeito de regularização, basta somar à função objetivo o módulo do vetor w multiplicado pelo fator conhecido por “taxa de regularização” λ . O efeito desta inclusão é uma diminuição (em função de λ) na atualização de w , o que evita a adaptação excessiva nos dados de treinamento:

$$Q(w) = \sum_{i=1}^n Q_i(w) + \lambda ||w||^2 \quad \text{Equação 2.7}$$

2.6 Avaliação dos Preditores

Há diversas formas de se avaliar um preditor e a escolha das métricas é bastante dependente do tipo de dados que se trabalha. No geral, o que se prevê é quanto um usuário irá gostar de um item e portanto a avaliação do erro médio absoluto ou erro quadrático médio são métricas comumente utilizadas (RMSE). O RMSE é calculado pela seguinte expressão:

$$RSME = \sqrt{\sum_{Base\ de\ Teste} \frac{(Valor\ Real - Valor\ Previsto)^2}{|Base\ de\ Teste|}} \quad \text{Equação 2.8}$$

Outros cálculos podem se mostrar necessários, como exemplo, em situações na qual os dados são desbalanceados ou extremamente tendenciosos como na classificação entre pessoas que

têm uma doença ou não. Se 99% dos pacientes não tiverem a doença, um preditor que prevê apenas esta informação já terá um erro bastante pequeno. Para avaliar o quão bem um preditor consegue rastrear pessoas que têm a doença, pode-se utilizar de informações como taxa de verdadeiros positivos, negativo, falsos-positivos ou falso-negativos e relações entre os mesmos para avaliar melhor o quão bom é o preditor. No caso da predição de CTR (probabilidade de um anúncio ser clicado quando impresso em um site), utiliza-se a métrica de verossimilhança ou loglikelihood, que indica a probabilidade total de que os preditores prevêem o que acontece nos dados de teste. Neste trabalho, o cálculo é dado por:

$$\mathcal{L} = - \sum_{Base\ de\ Teste} \log(CTR^{Clicks} * (1 - CTR)^{Não\ Clicks}). \quad \text{Equação 2.9}$$

Para se poder comparar o desempenho dos algoritmos implementados, propõe-se a aplicação de um preditor baseline baseado na contagem da frequência relativa observada dos anúncios na base de treinamento para avaliar seu desempenho na base de teste. A mesma é feita através de inferência bayesiana que minimiza perdas quadráticas com função à priori dada por uma distribuição Dirichlet(α , α) [62].

Não faz parte do escopo deste relatório desenvolver toda teoria por trás desta abordagem, porém a seguir apresenta-se de forma sucinta os principais conceitos utilizados.

A distribuição Dirichlet atribui a uma distribuição multinomial de k valores a distribuição [63]:

$$P(\theta) = \frac{1}{Z} \prod_{i=1}^k \theta_i^{\alpha_i - 1}, \quad \text{Equação 2.10}$$

onde:

$$Z = \frac{\prod_{i=1}^k \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^k \alpha_i)}. \quad \text{Equação 2.11}$$

e a função gamma é dada por:

$$\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt. \quad \text{Equação 2.12}$$

Então:

$$P(\theta | D) \propto P(D | \theta) P(\theta). \quad \text{Equação 2.13}$$

Onde D representa os dados coletados. Temos que:

$$P(D | \theta) = \prod_{i=1}^k \theta_i^{M_i}. \quad \text{Equação 2.14}$$

e:

$$P(\theta) \propto \prod_{i=1}^k \theta_i^{\alpha_i - 1}. \quad \text{Equação 2.15}$$

Mas dado as equações anteriormente desenvolvidas, observa-se que Dirichlet é um conjugado a priori para a multinomial, cuja posteriori é dada por $\text{Dir}(\alpha_1 + M_1, \alpha_2 + M_2, \dots, \alpha_k + M_k)$ e M equivale à estatística suficiente de um dado valor da multinomial, ou seja, quantas vezes ocorreu. O resultado para a etapa da predição bayesiana é, para um conjunto de M observações e modelado por θ :

$$P(X[M+1] = x^i | \theta, x[1], \dots, x[M]) = \frac{\alpha_i + M_i}{\alpha + M}. \quad \text{Equação 2.16}$$

Onde:

$$\alpha = \alpha_1 + \dots + \alpha_k. \quad \text{Equação 2.17}$$

No presente trabalho $k = 2$, ou seja, θ_i modela a probabilidade de Click e Não Click. Um exemplo da distribuição para $k = 2$ segue na Figura 12 a seguir:

Figura 12: Exemplo da distribuição Dirichlet em função de diferentes valores de α .

A Figura 12 mostra que quanto maior os valores de α na distribuição de Dirichlet mais concentrada e acentuada ela fica em torno do ponto 50%. Ao contrário, quanto mais próximo de 1 α está mais semelhante a distribuição fica da uniforme. A escolha portanto deve ser adequada aos tipos de dados nos quais se está implementando o modelo de Dirichlet.

3 Algoritmo Implementado

O principal objetivo desta pesquisa é a implementação de um preditor de CTR utilizando-se como referência o trabalho de MENON [6] o qual obteve os melhores resultados na literatura até 2012. Os resultados do seu modelo foram comparados com as melhores abordagens conhecidas até então [64], [65], superando-as em performance de predição. Não obstante, dados reais da empresa Lomadee serão utilizados na construção e testes destes preditores para melhor inferir o desempenho do preditor na prática. Por fim, uma contribuição é feita acrescentando ao modelo a componente dinâmica temporal, algo ainda não discutido na literatura.

Conforme já proposto neste relatório, uma versão dinâmica temporal será implementada para avaliar se a mesma consegue melhorar ainda mais os resultados já encontrados.

Nesta seção, todas as etapas para a construção do algoritmo final proposto são discutidas, desde o modelamento matemático até a sua implementação.

3.1 Desenvolvimento do Preditor de CTR

Começa-se uma discussão a respeito das características dos dados que normalmente se trabalha para se prever CTR. Primeiramente, vale discutir a solução mais simples para o problema. Supondo-se que um conjunto de observações foram coletadas em relação à impressão e cliques observados dos anúncios, pode-se aplicar a solução direta:

$$CTR_{ij}^{MLE} := \begin{cases} \frac{C_{ij}}{V_{ij}} & \text{se } V_{ij} \neq 0. \\ ? & \text{caso contrário.} \end{cases} \quad \text{Equação 3.1}$$

“i” representa um dado site qualquer e j um anúncio qualquer. “C_{ij}” representa a quantidade de cliques observados quando o site i imprimiu o anúncio j. “V” representa as visualizações ou impressões dos anúncios. “?” indica que nenhuma impressão foi feita.

Apesar de simples, a solução apresenta problemas devido à falta de robustez. Pode acontecer, conforme previamente discutido previamente, de que um site imprima apenas uma vez um dado anúncio e um clique seja observado, resultando em um CTR estimado de 100%, o que é bastante enviesado considerando-se que foi feita apenas uma observação no total. Outro fato que deve ser considerado é que as informações são esparsas, ou seja, frequentemente acontece de um dado site imprimir por exemplo apenas 5 vezes um dado anúncio e obter nenhum clique no período de tempo que o mesmo ficou exposto. Isto pode acarretar a previsão de que o CTR é nulo, o que é a princípio um pouco distorcido devido ao pequeno número de observações que foram feitas.

A Equação 3.1 só vale se V_{ij} tender ao infinito, porém isto nunca acontece na prática invalidando esta abordagem.

Outro problema bem conhecido para construção de preditores de CTR é que o evento do “Click” é bastante raro o que aumenta o desafio na etapa de estimação. MENON [6] sugere um modelo que integra diversas abordagens, sendo uma delas o uso de informações hierárquicas sobre páginas e anúncios para se prever com maior qualidade a taxa de conversão. Tais hierarquias podem revelar correlações entre os CTRs de diferentes anúncios, como, por exemplo, a correlação existente entre anúncios de uma mesma campanha iniciada por dado anunciante. Este conceito será discutido mais adiante.

Do que foi discutido, pretende-se desenvolver um estimador $\widehat{CTR}$ que satisfaça três propriedades:

1. Consegue fornecer estimativas mesmo quando $V_{ij} = 0$.
2. Deve suavizar a estimativa quando V_{ij} é pequeno.
3. Deve fornecer probabilidades, ou seja, valores entre 0 e 1.

Há basicamente duas principais abordagens desenvolvidas na literatura ao problema de predição de CTR: em uma delas, os preditores rastreiam características explícitas dos anúncios e sites (tamanho, texto, posição) conhecida como “Side Information” e na outra, emprega-se modelos estatísticos.

- Side Information

- Tem Texto?
- Tem Ilustração?
- Qual a posição?
- Qual Tamanho?
- Tem Animações?
- Qual Categoria?

- Modelos Estatísticos

Figura 13: As principais abordagens desenvolvidas na literatura para desenvolvimento de preditor de CTR.

A solução adotada no algoritmo a ser implementado visa empregar ambas abordagens, ou seja, são considerados tanto o side-information quanto equações estatísticas para estimar a taxa de conversão. MENON afirma que tal abordagem foi capaz, quando trabalhando de forma híbrida com todas as modelagens que serão discutidas a seguir, de superar os melhores preditores apresentados até então na literatura.

O equacionamento que deve ser desenvolvido é basicamente igual a:

$$CTR_{ij} = P[Clicks | View, i, j]. \quad \text{Equação 3.2}$$

“i” representa um item relacionado à estrutura de páginas como um site por exemplo, enquanto que “j” representa um dado anúncio. CTR_{ij} é a probabilidade de se ter um número de cliques dado que se tem um total de visualizações “View” e que o anúncio j foi impresso no site i.

O modelo desenvolvido por MENON [6] e que será seguido neste texto possui basicamente três grandes abordagens: “pesos de confiança”, “side-information” e por fim, “estrutura hierárquica”. Cada modelo utiliza-se do anterior e acrescenta um novo conceito para melhorar a acurácia de predição dos CTRs observados.

A Figura 14 ilustra as partes que constituem o modelo:

Figura 14: Principais abordagens que constituem o modelo a ser desenvolvido.

A abordagem de pesos de confiança visa construir uma matriz com Cliques e Visualizações observados para cada site e anúncio e a partir desta matriz aplicar a fatoração SVD para associar a cada site e a cada anúncio um vetor próprio que os caracteriza. Estes vetores são utilizados em um modelo de regressão logística para rastrear os CTRs observados na matriz, conforme será explicado adiante.

O modelo Side Information utiliza a abordagem Pesos de Confiança e acrescenta na formulação do CTR vetores de pesos associados às características dos anúncios, sites e campanhas. Por fim, o modelo hierárquico utiliza as duas abordagens anteriores e ainda acrescenta na formulação a dependência que as campanhas, anunciantes e publicadores causam nos CTRs observados. A seguir, cada abordagem será explicada detalhadamente, dado que cada parte do algoritmo a ser discutida foi implementada e testada para validação final do preditor.

3.1.1 Fatoração Baseada em Pesos de Confiança

A primeira abordagem consiste em utilizar técnicas de filtragem colaborativa [17] combinadas com técnicas de regressão logística. Primeiramente, os dados são organizados em matrizes contendo os Cliques e Visualizações que cada anúncio obteve quando impresso em dado site no período de tempo das amostras. A Figura 15 ilustra o conceito a seguir:

		Anúncios

Sites		C: 2 V: 17	C: 0 V: 32	?
	C: 5 V: 12	C: 1 V: 68	C: 0 V: 10
	C: 1 V: 23	C: 3 V: 15	C: 7 V: 143
	C: 4 V: 86	?	?
	?	C: 0 V: 130	C: 1 V: 10

Figura 15: Matriz utilizada pelo preditor de CTR.

As técnicas de fatoração de matriz (SVD) são utilizadas conforme discutido em [24], [24] e [17] e o resultado são vetores associados aos sites e anúncios. A Figura 16 ilustra o conceito a seguir:

		β_{Apple}	$\beta_{Saraiva}$	β_{Coca}

α_1		C: 2 V: 17	C: 0 V: 32	?
α_2		C: 5 V: 12	C: 1 V: 68	C: 0 V: 10
α_3		C: 1 V: 23	C: 3 V: 15	C: 7 V: 143
α_4		C: 4 V: 86	?	?
α_5		?	C: 0 V: 130	C: 1 V: 10

Figura 16: Vetores α associados aos sites e vetores β aos anúncios.

A idéia do processo de fatoração de matriz neste ponto é encontrar vetores (no caso α e β) que designam fatores latentes dos sites e anúncios e como eles se relacionam entre si. Estes vetores são então utilizados para explicar o que esta sendo observado na matriz original.

Cada célula pode ser dividida como a impressão ocorrida e o seu resultado observado, ou seja, se houve Clique ou não. Como exemplo, a primeira célula da Figura 16 registrou dezessete eventos dos quais dois destes são positivos indicados pela ocorrência de Cliques. Cada célula pode, portanto, ser considerada como uma combinação de diversas matrizes que registram a ocorrência de eventos positivos ou negativos, conforme ilustra a Figura 20 a seguir:

Figura 17: Composição das células em eventos positivos ou negativos.

Para o rastreamento de Cliques, cada Clique é considerado um evento positivo. O resultado é que em cada célula da matriz há C eventos positivos e (V-C) eventos negativos. A idéia então é que o preditor de CTR rastreie a probabilidade de ocorrência de eventos positivos, ou seja, Cliques, para cada célula da matriz.

Nesta abordagem, cada site é associado a um vetor α (conforme ilustra a Figura 16) com parâmetros latentes e o mesmo vale para cada anúncio, associados a β . Estes vetores são obtidos pelas técnicas de fatoração de matriz, conforme explicado na seção 2.3.

O valor esperado de CTR é dado em função do produto interno entre esses vetores $\langle \alpha, \beta \rangle$. Os mesmos são utilizados em uma função de regressão logística pois a mesma é adequada para rastrear eventos que ocorrem ou não, que é basicamente o cenário que se está trabalhando neste modelo, ou seja, a ocorrência ou não do evento Clique [25]. MENON não deixa claro a função de regressão logística utilizada para expressar e estimativa do CTR [6] então para este trabalho considera-se que a mesma é dada por:

$$\widehat{CTR}(\theta) = \frac{1}{1 + e^{-\theta}} \quad \text{Equação 3.3}$$

e portanto:

$$\widehat{CTR} = \frac{1}{1 + e^{-(\alpha^T * \beta)}} \quad \text{Equação 3.4}$$

A princípio, o modelo de regressão logística para rastrear a probabilidade de cada evento Clique é dada pela seguinte redução da função custo:

$$\min_{\alpha, \beta} \frac{1}{|O|} \sum_{(i,j) \in O} \left(-X_{ij} \log \left(\widehat{CTR}_{ij}(\alpha, \beta) \right) - (1 - X_{ij}) \log \left(1 - \widehat{CTR}_{ij}(\alpha, \beta) \right) \right) + \Omega(\alpha, \beta). \quad \text{Equação 3.5}$$

α são vetores latentes de dimensão f associados ao site e β também são vetores porém associados a cada anúncio. X_{ij} é a observação feita no conjunto de treinamento “O” e vale zero ou um, sendo zero a indicação de que não houve Clique e um de que houve. A base de dados “O” representa o conjunto de pontos (i,j) que se tem amostras com V_{ij} maiores que zero na base amostrada e a função Ω é a parte de regularização e será melhor descrita adiante.

Porém, isto só contabiliza o custo pela estimativa de um evento que pode ser Clique ou Não Clique. Como em cada célula da matriz há C_{ij} eventos Cliques e $(V_{ij} - C_{ij})$ Não Cliques, o erro total deve contabilizar o total de eventos ocorridos, o que é feito através da construção da seguinte função custo:

$$J(\alpha, \beta, C, V) = \frac{1}{|O|} \sum_{(i,j) \in O} \left(-C_{ij} \log \left(\widehat{CTR}_{ij}(\alpha, \beta) \right) - (V_{ij} - C_{ij}) \log(1 - \widehat{CTR}_{ij}(\alpha, \beta)) \right) + \frac{\lambda_\alpha}{2} \|\alpha\|^2 + \frac{\lambda_\beta}{2} \|\beta\|^2. \quad \text{Equação 3.6}$$

onde os parâmetros λ são fatores de regularização para evitar overfitting na adaptação dos vetores α e β . J é a notação utilizada para representar a função custo. A Equação 3.6 é semelhante à Equação 3.5, o que muda são os pesos que contabilizam pelo erro total com a ocorrência de C_{ij} Cliques e $(V_{ij} - C_{ij})$ Não Cliques (conforme explicado no exemplo da matriz com Cliques e Não Cliques).

O problema então é encontrar os vetores α e β que minimizam a função custo expressa na Equação 3.6, o que pode ser feito utilizando-se técnicas de gradiente estocástico. Para isto, utiliza-se a função custo objetivo para preditores baseados em regressão logística com pesos C_{ij} e $(V_{ij} - C_{ij})$:

$$J = \left(-C_{ij} \log \left(\frac{1}{1 + e^{-(\alpha_i^T \cdot \beta_j)}} \right) - ((V_{ij} - C_{ij}) \log \left(1 - \frac{1}{1 + e^{-(\alpha_i^T \cdot \beta_j)}} \right) \right) + \lambda_\alpha \|\alpha_i\|^2 + \lambda_\beta \|\beta_i\|^2. \quad \text{Equação 3.7}$$

Conforme previamente discutido na seção 2.5, as equações de atualização para os vetores são dadas por:

$$\alpha_{i_k} := \alpha_{i_k} - \xi \frac{\partial J}{\partial \alpha_{i_k}}. \quad \text{Equação 3.8}$$

$$\beta_{i_k} := \beta_{i_k} - \xi \frac{\partial J}{\partial \beta_{j_k}}. \quad \text{Equação 3.9}$$

Resultando nas atualizações para os vetores α e β :

$$\alpha_{i_k} = \alpha_{i_k} (1 - \xi \lambda_\alpha) - \xi \left(\frac{1}{1 + e^{-(\alpha_i \cdot \beta_j)}} \beta_{j_k} \right) [-C_{ij} + (V_{ij} - C_{ij}) e^{(\alpha_i \cdot \beta_j)}]. \quad \text{Equação 3.10}$$

$$\beta_{i_k} = \beta_{i_k}(1 - \xi\lambda_\beta) - \xi \left(\frac{1}{1 + e^{-(\alpha_i \cdot \beta_j)}} \alpha_{i_k} \right) [-C_{ij} + (V_{ij} - C_{ij})e^{(\alpha_i \cdot \beta_j)}]. \quad \text{Equação 3.11}$$

onde ξ é a taxa de aprendizagem.

Para obtenção dos melhores hiperparâmetros ξ e λ , obtém-se o desempenho de cada parâmetro e compara-se o valor total de verossimilhança obtida na base de validação cruzada, expressa por:

$$\mathcal{L} = - \sum_{(i,j) \in CV} \log \left(\widehat{CTR}_{ij}^{c_{ij}^{CV}} (1 - \widehat{CTR}_{ij})^{(v_{ij}^{CV} - c_{ij}^{CV})} \right). \quad \text{Equação 3.12}$$

E o mesmo vale para a etapa de teste com a diferença de que os pontos (i,j) pertencem ao conjunto T ao invés de CV (“cross-validation”).

3.1.2 Fatoração Baseada em Confiança com Side Information

É possível utilizar uma forma simples e eficiente de integrar as características tanto de anúncios quanto de páginas no modelo já descrito. Considerando-se que a predição CTR_{ij} deve ainda estar entre 0 e 1, é possível integrar as informações *side-information* através do seguinte modelamento:

$$\widehat{CTR}_{ij} = \frac{1}{1 + e^{-(\alpha_i^T \beta_j + w^T x_{ij})}}. \quad \text{Equação 3.13}$$

“w” no caso representa um vetor de pesos em relação às características x_{ij} . Para os dados da empresa Lomadee, x é dividido da seguinte maneira:

- Categoria primária da campanha do anúncio. Pode assumir um de 36 valores distintos então 36 pontos no vetor são destinados para esta identificação. Por exemplo, categoria primária dois é representada pelo vetor $[0 \ 1 \ 0 \ 0 \ (\dots) \ 0] \in \mathbb{R}^{36}$.

- Tipo do anúncio, 3 possíveis valores: estático, dinâmico e código externo (valor utilizado internamente pela empresa que cedeu os dados).
- Creative type, também 3 valores: HTML, Image, Flash.
- Dimensão do anúncio (altura e largura), neste caso considera-se 24 possíveis valores, pois são as únicas que aparecem na base de dados de dezembro e janeiro.
- Categoria primária do site, também 36 valores.
- Site type: 4 possíveis valores: Blog, Mobile, Social Network, Site.

Portanto, x_{ij} possui dimensão $36 + 3 + 3 + 24 + 36 + 4 = 106$. Se um dado anúncio possui categoria 2 então $x_{ij} = [0 \ 1 \ 0 \ 0 \ \dots]$, ou seja, x_{ij} é inicialmente um vetor nulo e preenche-se com 1 os pontos correspondentes das respectivas features.

Com este modelamento, não só uma estimativa estatística é feita para o CTR através do produto interno entre α e β mas também de como as características dos anúncios e sites influenciam no processo de decisão dos usuários.

As equações de atualização ficam bastante semelhantes, dadas por:

$$\begin{aligned} \alpha_{i_k} &= \alpha_{i_k} (1 - \xi \lambda_\alpha) \\ &- \xi \left(\frac{1}{1 + e^{-(\alpha_i^T * \beta_j + w^T * x_{ij})}} \beta_{j_k} \right) \left[-C_{ij} + (V_{ij} \right. \\ &\left. - C_{ij}) e^{(\alpha_i^T * \beta_j + w^T * x_{ij})} \right]. \end{aligned} \quad \text{Equação 3.14}$$

$$\begin{aligned} \beta_{i_k} &= \beta_{i_k} (1 - \xi \lambda_\beta) \\ &- \xi \left(\frac{1}{1 + e^{-(\alpha_i^T * \beta_j + w^T * x_{ij})}} \alpha_{i_k} \right) \left[-C_{ij} + (V_{ij} \right. \\ &\left. - C_{ij}) e^{(\alpha_i^T * \beta_j + w^T * x_{ij})} \right]. \end{aligned} \quad \text{Equação 3.15}$$

$$w_k = -\xi \left(\frac{1}{1 + e^{-(\alpha_i^T \beta_j + w^T x_{ij})}} x_{ijk} \right) \left[-C_{ij} + (V_{ij} - C_{ij}) e^{(\alpha_i^T \beta_j + w^T x_{ij})} \right].$$

Equação 3.16

Vale observar que não é recomendável w_k possuir taxa de regularização [6].

3.1.3 Modelo Hierárquico

É possível melhorar ainda mais a habilidade de predição dos algoritmos previamente discutidos através da inclusão de informações hierárquicas nos dados. Isto ocorre pois é possível que se tenha bastante informação em um ponto da hierarquia e pouca em outro ponto.

Primeiramente, é interessante entender do que se trata as estruturas hierárquicas possíveis de se encontrar nos dados. A impressão virtual de anúncios acontece envolvendo a interação entre três participantes: o publicador, o anunciante e o usuário final, conforme ilustra a Figura 21:

Figura 18: Interação entre Publicador, Anunciante e Usuário.

O publicador é basicamente o criador dos sites. Por exemplo, o Yahoo! é um publicador e possui diversos sites, dentre eles alguns são relacionados à esportes, outro a finanças dentre diversos temas. O anunciante é quem inicializa campanhas. A Figura 21 mostra como exemplo o anunciante “Apple” que inicializa uma campanha com anúncios sobre iPhones e iMacs. Estas campanhas são entregues aos publicadores que imprimem os anúncios para o usuário final, que decide por fim se o que esta sendo exposto é interessante ou não, através do Clique.

A Figura 19 e a Figura 20 a seguir ilustram melhor o conceito:

Figura 19: Exemplo de Hierarquia construída na estrutura de Sites.

Figura 20: Exemplo de Hierarquia construída na estrutura de Anúncios.

Como é possível observar, cada publicador (publisher) possui um ou mais sites. Cada anunciante (advertiser) inicializa uma ou mais campanhas e estas, por sua vez, possuem um ou mais anúncios.

MENON [6] não revela qual estrutura hierárquica que foi utilizada, sendo portanto a retratada na figura a que foi considerada para o presente projeto e que se mostrou melhor para os dados do Lomadee. Nela, basicamente cada hierarquia começa com o nó raiz ou nó “root”. No caso da hierarquia de páginas, depois do nó root vem os nós “publishers” e cada publisher pode possuir um ou mais sites (sources, ou seja, as diferentes partes que um site pode ter, não foram considerados pois o modelamento poderia ficar exponencialmente mais complexo). Para a hierarquia dos anúncios, do nó raiz surgem os anunciantes, cada um pode ter uma ou mais campanhas e cada campanha pode ter um ou mais anúncios, sendo que um dado anúncio pode participar de diferentes campanhas (o que faz com que a hierarquia não seja uma árvore, o que dificulta consideravelmente os cálculos).

A estratégia então visa analisar como um ponto com mais observações pode ajudar a prever os outros no qual pouca coisa se conhece. Conforme previamente discutido, MENON [6] sugere

três diferentes abordagens que utilizam a estrutura hierárquica dos dados e que serão explicadas logo adiante, conforme ilustra a a seguir:

Figura 21: Estrutura Hierárquica

Como será melhor explicado adiante, considerar a estratégia hierárquica acrescenta diversas vantagens. Com esta estrutura, é possível obter informações espalhadas em toda a hierarquia e verificar como elas impactam cada nó. Se para um dado publisher tem-se bastante informação sobre como um dado site de seu domínio se comporta em termos de conversão de cliques, esta informação pode auxiliar na tarefa de calcular o CTR de um outro site do mesmo publisher mas que não se conhece muito a respeito. No caso, a informação conhecida sobre o primeiro site é utilizada na predição do outro site do mesmo publicador. Com esta estrutura, o que se conhece de um nó acaba influenciando os outros nós e isto auxilia na etapa de predição.

Feita a explicação sobre hierarquias nos dados, aborda-se agora as três estratégias para melhorar o rastreamento de CTR utilizando-se hierarquias.

1 - Regularização Hierárquica

Nesta abordagem, cada nó na estrutura hierárquica recebe um vetor latente e as atualizações destes vetores são tais que cada nó influencia outros nós na hierarquia. A Figura 22 ilustra o conceito:

Figura 22: Como Ad 2 possui mais informações, as atualizações de Ad 1 e Campanha 1 são feitas de tal forma a se aproximarem de Ad 2.

A Figura 22 ilustra algo que pode acontecer na estrutura hierárquica dos dados. Uma campanha pode anunciar dois produtos ou serviços, no caso representados por Ad 1 e Ad 2. Um deles pode obter mais clicks e ser impresso mais vezes, representado por um retângulo maior em Ad 2 indicando que o mesmo obteve mais informações que Ad 1. O processo de atualização dos vetores faz com que os vetores da Campanha 1 se aproximem de Ad 2 e faz com que Ad 1 se aproxime da Campanha 1. Mas como a Campanha 1 está se aproximando de Ad 2, indiretamente o Ad 1 está se aproximando do Ad 2. O conhecimento intrínseco ao Ad 2 é transferido ao Ad 1 fornecendo mais informações para a predição do CTR associado ao Ad 1. Trata-se de uma abordagem eficiente para tratar dados esparsos pois os nós que se conhece pouco acabam “emprestando” conhecimento de nós que se conhece bastante.

Para se obter este efeito, na parte de regularização dos parâmetros, cada vetor é projetado para seguir uma distribuição normal com a média tendendo ao vetor dos respectivos pais. Matematicamente:

$$\alpha_{Root} \sim \mathcal{N}(0, \lambda_{Root}^{-1} I). \quad \text{Equação 3.17}$$

α do nós raiz segue uma distribuição normal com desvio padrão em função do fator de regularização λ . Para os outros nós, têm-se:

$$\forall i \in \{\mathcal{H}^P - \{Root\}\}: \alpha_i \sim \mathcal{N}(\alpha_{Par(i)}, \lambda_a^{-1} I). \quad \text{Equação 3.18}$$

Cada nó na estrutura de páginas, exceto o nó raiz, segue uma distribuição normal cuja média é o valor do vetor do nó pai. A mesma idéia modela os vetores na estrutura dos anúncios:

$$\forall j \in \{\mathcal{H}^A - \{Root\}\}: \beta_j \sim \mathcal{N}(\beta_{Par(j)}, \lambda_b^{-1} I). \quad \text{Equação 3.19}$$

No caso, $\mathcal{H}^P$ representa a hierarquia de páginas e $\mathcal{H}^A$ a de anúncios. A mesma fórmula vale para cada vetor nos nós dos anúncios, portanto a parte reguladora resulta:

$$\begin{aligned} & \frac{\lambda_a}{2} \sum_{i \in \mathcal{H}^P} \|\alpha_i - \alpha_{Par(i)}\|^2 + \frac{\lambda_b}{2} \sum_{j \in \mathcal{H}^A} \|\beta_j - \beta_{Par(j)}\|^2 \\ & + \frac{\lambda_R}{2} (\|\alpha_{Root}\|^2 + \|\beta_{Root}\|^2). \end{aligned} \quad \text{Equação 3.20}$$

Subtrair o vetor dos pais (se houver mais de um pai basta pegar a média dos mesmos) equivale a puxar a média da distribuição normal para o valor do vetor dos pais.

Este modelo tem uma pequena desvantagem e esta se deve ao fato de que os vetores dos pais aparecem apenas na função de regularização. Se λ for nulo esta abordagem não funciona. Uma forma de se lidar com isto é utilizando-se dados agregados (ou aglomerados).

2 – Ajuste Dados Aglomerados

A idéia é que é possível tornar a presença dos pais nos nós da estrutura hierárquica diretamente dependente dos dados. Para isto, basta agregar em uma nova hierarquia o total de visualizações e clicks ocorridos. No caso, a nova hierarquia é montada a partir de:

$$G := \mathcal{H}^P \times \mathcal{H}^A .$$

Equação 3.21

Então vale:

$$C^{Agg}(u, v) = \sum_{(u', v') : (u, v) \in Par((u', v'))} C^{Agg}(u', v') .$$

Equação 3.22

E o mesmo vale para visualizações:

$$V^{Agg}(u, v) = \sum_{(u', v') : (u, v) \in Par((u', v'))} V^{Agg}(u', v') .$$

Equação 3.23

O significado da fórmula é que nós pais absorvem os clicks e visualizações que seus filhos têm.

Os pares formados são ilustrados a seguir:

Figura 23: Pares formados na etapa de agregação dos dados.

Figura 24: Exemplo de um Publisher que agrega dados de seus sites.

Nos algoritmos desenvolvidos, utilizou-se a codificação a seguir, na qual o primeiro valor é um ponto da estrutura de páginas e o segundo da estrutura de anúncios:

Page	Ad	Type
site	ad	1
site	camp	2
site	adver	3
pub	ad	4
pub	camp	5
pub	adver	6

Tabela 1: Estrutura de dados agregados implementada.

Por exemplo, a linha na base de dados tal como na Tabela 2:

Page	Ad	Type	V	C
26896	1205542	1	150	3

Tabela 2: Exemplo de uma linha dos dados

significa que page representa um site e ad é o próprio anúncio, sendo que houve 150 impressões e um total de 3 cliques. Com esta nova abordagem, os parentes na estrutura hierárquica também estão presentes na função custo diretamente sendo mais dependentes dos dados. Há um problema importante de se considerar: um dado par (site, anúncio) terá menos visualizações e cliques que um par (site, campanha) por exemplo. Portanto, os vetores latentes associados ao site terão um peso maior na equação quando pareados às campanhas, desprezando-se automaticamente o par menor (site, anúncio).

Uma forma de se lidar com isto é utilizando-se o seguinte procedimento para aplicação do gradiente estocástico [6]:

1. Faz-se os parâmetros dos nós superiores fixos e atualiza-se apenas os nós folhas.
2. Tornar os vetores dos nós folhas fixos e atualiza-se apenas nós superiores.
3. Novamente atualiza-se os nós folhas e o procedimento esta completo.

A Figura 28 ilustra o procedimento:

Figura 25: Exemplo das atualizações realizadas. Primeiro atualiza-se nós folhas, depois nós superiores e por fim novamente nós folhas.

3 – Ajuste Residual

Em todas as predições anteriores, a função de predição era dada por $\sigma(\alpha_i^t, \beta_j)$, ou seja, apenas os nós folha entravam no resultado final. O que se propõe nesta terceira abordagem é utilizar os nós pais como biases à priori para a predição final, resultando no modelo:

$$\sigma(\tilde{\alpha}_i^t, \tilde{\beta}_j).$$

onde:

$$\tilde{\alpha}_{ik} = \alpha_{ik} + \sum_{u \in Path(i) - \{i\}} \alpha_{uk}.$$

Equação 3.24

e para beta:

$$\tilde{\beta}_{jk} = \beta_{jk} + \sum_{u \in \text{Path}(j) - \{j\}} \beta_{uk}.$$

Equação 3.25

E o mesmo vale para vetores associados ao anúncio. A figura a seguir ilustra os vetores que são utilizados através do caminho em preto:

Figura 26: Exemplos dos nós sendo utilizados na etapa de predição.

Cada nó possui um vetor latente que são utilizados na somatória acima.

4 – Método Híbrido

Por fim, o método híbrido engloba todas as estratégias discutidas até então: pesos de confiança, side-information e estrutura hierárquica [6].

Figura 27: Composição do modelo híbrido.

Apresenta-se agora as fórmulas matemáticas do modelo híbrido. Integrando-se todas as abordagens discutidas até este ponto, obtemos:

$$\begin{aligned}
 J(\alpha, \beta, C^{Agg}, V^{Agg}) &= -C_{ij}^{Agg} \log\left(\frac{1}{1 + e^{-\theta}}\right) - (V_{ij}^{Agg} - C_{ij}^{Agg}) \log\left(\frac{e^{-\theta}}{1 + e^{-\theta}}\right) \\
 &+ \frac{\lambda_\alpha}{2} \sum_{i \in \mathcal{H}^P} \|\alpha_i - Par(i)\|^2 + \frac{\lambda_\beta}{2} \sum_{j \in \mathcal{H}^A} \|\beta_j - Par(j)\|^2 \\
 &+ \frac{\lambda_R}{2} \sum \|\alpha_{Root} - \beta_{Root}\|^2.
 \end{aligned}
 \tag{Equação 3.26}$$

Onde:

$$\begin{aligned}
 \theta = &\left[\left(\alpha_{ik} + \sum_{u \in Path(i) - \{i\}} \alpha_{uk} \right) \cdot \left(\beta_{jk} + \sum_{u \in Path(j) - \{j\}} \beta_{uk} \right) \right. \\
 &\left. + \sum_{type} (w_{type} X_{type}) \right].
 \end{aligned}
 \tag{Equação 3.27}$$

Na Equação 3.26 é possível observar que o modelo de regressão logística com pesos é mantida, com a diferença de que desta vez os pesos provém dos dados que foram agregados para

acomodar nós mais altos da hierarquia. A regularização força os vetores a se aproximarem dos pais e os pais caminham próximos dos nós filhos com mais visualizações. Por fim, Θ engloba a somatória de todos os nós no caminho do anúncio e do site bem como incorpora o side-information através do produto interno entre w e X .

As taxas de regularização para β e root serão consideradas iguais para melhorar desempenho do preditor.

Este modelamento fornece as seguintes atualizações:

$$\begin{aligned} \frac{\partial J(\alpha, \beta, C^{Agg}, V^{Agg})}{\partial \theta} &= C_{ij}^{Agg} \left(\frac{1}{1 + e^{-\theta}} \right) e^{-\theta} (-1) \frac{\partial \theta}{\partial \phi} \\ &+ (V_{ij}^{Agg} - C_{ij}^{Agg}) \left(\frac{1}{1 + e^{\theta}} \right) e^{\theta} \frac{\partial \theta}{\partial \phi} + \frac{\partial \Omega(\theta)}{\partial \phi}. \end{aligned} \quad \text{Equação 3.28}$$

No caso, ϕ representa um dos possíveis vetores a serem atualizados (ou α , ou β ou w) e Ω é a parte reguladora. Desenvolvendo mais a Equação 3.28:

$$\begin{aligned} \frac{\partial J(\alpha, \beta, C^{Agg}, V^{Agg})}{\partial \theta} &= \left(\frac{1}{1 + e^{\theta}} \right) \frac{\partial \theta}{\partial \phi} (-C_{ij}^{Agg} + (V_{ij}^{Agg} - C_{ij}^{Agg})e^{\theta}) + \frac{\partial \Omega(\theta)}{\partial \phi}. \end{aligned} \quad \text{Equação 3.29}$$

Sugere-se, para implementar o modelo híbrido, dividir o processo de atualização dos vetores em três situações distintas [6]. Na primeira, os vetores de nós não folhas serão mantidos fixos e não participam da atualização. Na segunda fase, os vetores de nós folhas é quem são mantidos fixos e apenas vetores não folhas são atualizados. Por fim, vetores folhas são atualizados mais uma vez, garantindo vetores que rastreiam tanto dados agregados quanto pares normais sem causar overfitting pela configuração da função custo.

Como alguns pontos da regularização possuem valor em relação aos seus respectivos pais, as equações a seguir serão apresentadas em grupos relacionados à cada parte do algoritmo. As equações ficam:

$$\begin{aligned} \frac{\partial J(\alpha, \beta, C^{Agg}, V^{Agg})}{\partial \alpha_p} &= \left(\beta_{jk} + \sum_{u \in Path(j) - \{j\}} \beta_{uk} \right)_p \left(\frac{1}{1 + e^\theta} \right) (-C_{ij}^{Agg} \\ &+ (V_{ij}^{Agg} - C_{ij}^{Agg})e^\theta) + \lambda_\alpha (\alpha_p - Par(i)_p). \end{aligned} \quad \text{Equação 3.30}$$

Onde “p” indica um dado ponto no vetor latente, indo de 1 até f e f é a dimensão do vetor.

Atualização para Beta:

$$\begin{aligned} \frac{\partial J(\alpha, \beta, C^{Agg}, V^{Agg})}{\partial \beta_p} &= \left(\alpha_{ik} + \sum_{u \in Path(i) - \{i\}} \alpha_{uk} \right)_p \left(\frac{1}{1 + e^\theta} \right) (-C_{ij}^{Agg} \\ &+ (V_{ij}^{Agg} - C_{ij}^{Agg})e^\theta) + \lambda_\beta (\beta_p - Par(j)_p). \end{aligned} \quad \text{Equação 3.31}$$

Atualização para w:

$$\frac{\partial J(\alpha, \beta, C^{Agg}, V^{Agg})}{\partial w} = \left(\frac{1}{1 + e^\theta} \right) X_{type_p} (-C_{ij}^{Agg} + (V_{ij}^{Agg} - C_{ij}^{Agg})e^\theta). \quad \text{Equação 3.32}$$

Para a fase dois, a atualização dos nós sites e anuncios continuam iguais; publishers são atualizados da seguinte maneira:

$$\begin{aligned}
& \frac{\partial J(\alpha, \beta, C^{Agg}, V^{Agg})}{\partial \alpha_p} \\
&= \left(\beta_{jk} + \sum_{u \in Path(j) - \{j\}} \beta_{uk} \right)_p \left(\frac{1}{1 + e^\theta} \right) (-C_{ij}^{Agg} \\
&+ (V_{ij}^{Agg} - C_{ij}^{Agg}) e^\theta) \\
&+ \lambda_\alpha \left((|C(publisher)| + 1) \alpha_{publisher_p} - \left(\sum_{t \in C(pub)} \alpha_{t_p} \right) \right. \\
&\left. - \alpha_{Root_p} \right).
\end{aligned}$$

Equação 3.33

Onde $C(publisher)$ indica os filhos do nó publisher (sites que se conectam ao respectivo publicador).

O nó alpha root resulta:

$$\begin{aligned}
& \frac{\partial J(\alpha, \beta, C^{Agg}, V^{Agg})}{\partial \alpha_p} \\
&= \left(\beta_{jk} + \sum_{u \in Path(j) - \{j\}} \beta_{uk} \right)_p \left(\frac{1}{1 + e^\theta} \right) (-C_{ij}^{Agg} \\
&+ (V_{ij}^{Agg} - C_{ij}^{Agg}) e^\theta) \\
&+ \lambda_\alpha \left(|C(Root)| \alpha_{Root_p} - \sum_{t \in C(Root)} \alpha_{t_p} \right) + \lambda_R \alpha_{Root_p}.
\end{aligned}$$

Equação 3.34

Para Campanhas:

$$\begin{aligned}
& \frac{\partial J(\alpha, \beta, C^{Agg}, V^{Agg})}{\partial \beta_p} \\
&= \left(\alpha_{ik} + \sum_{u \in Path(i) - \{i\}} \alpha_{uk} \right)_p \left(\frac{1}{1 + e^\theta} \right) (-C_{ij}^{Agg} \\
&+ (V_{ij}^{Agg} - C_{ij}^{Agg}) e^\theta) \\
&+ \lambda_\beta \left((|C(camp)| + 1) \beta_{camp_p} - \left(\sum_{t \in C(Camp)} \beta_{t_p} \right) \right. \\
&\quad \left. - \beta_{Advertiser_p} \right).
\end{aligned}$$

Equação 3.35

Para Advertisers:

$$\begin{aligned}
& \frac{\partial J(\alpha, \beta, C^{Agg}, V^{Agg})}{\partial \beta_p} \\
&= \left(\alpha_{ik} + \sum_{u \in Path(i) - \{i\}} \alpha_{uk} \right)_p \left(\frac{1}{1 + e^\theta} \right) (-C_{ij}^{Agg} \\
&+ (V_{ij}^{Agg} - C_{ij}^{Agg}) e^\theta) \\
&+ \lambda_\beta \left((|C(Advertiser)| + 1) \beta_{Advertiser_p} \right. \\
&\quad \left. - \left(\sum_{t \in C(Advertiser)} \beta_{t_p} \right) - \beta_{Root_p} \right).
\end{aligned}$$

Equação 3.36

E finalmente para Beta Root:

$$\begin{aligned}
& \frac{\partial J(\alpha, \beta, C^{Agg}, V^{Agg})}{\partial \beta_p} \\
&= \left(\alpha_{ik} + \sum_{u \in Path(i) - \{i\}} \alpha_{uk} \right) \left(\frac{1}{1 + e^\theta} \right) (-C_{ij}^{Agg} \\
&+ (V_{ij}^{Agg} - C_{ij}^{Agg}) e^\theta) \\
&+ \lambda_\beta \left(|C(Root)| \beta_{Root_p} - \sum_{t \in C(Root)} \beta_{t_p} \right) + \lambda_R \beta_{Root_p} .
\end{aligned}$$

Equação 3.37

Para construção dos vetores w_{type} e X_{type} , vale a seguinte construção:

- w_{site} : vetor X_{site} possui dimensão 40. Os últimos 36 parâmetros são indicadores de sua categoria primária e os primeiros quatro do $site_type$.
- w_{Ad} : vetor X_{ad} possui dimensão 30. Os 3 primeiros associados ao ad_type_id , depois os 3 seguintes são associados ao $creative_type$ e por fim os 24 últimos ao $ad_dimension$
- $w_{Campaign}$: vetor $X_{campaign}$ possui dimensão 36 relacionado à categoria primária da campanha.

4 – Método Híbrido com Melhoria de Performance

Após testes realizados com o método híbrido, observou-se que a parte reguladora da equação de custo total acrescentou um aumento no tempo de convergência bastante considerável. De fato, cada iteração completa do gradiente estocástico passou a demorar em torno de duas a três horas o que antes consumia de 10 a 15 minutos. A complexidade do algoritmo o torna bastante custoso em termos computacionais.

Porém, algumas alterações podem ser realizadas a fim de simplificar a equação e otimizar consideravelmente a implementação do gradiente estocástico, sendo que uma delas é não mais “emprestar” forças de um nó para outro através de restrições na regularização [6], [66]. O mesmo pode acontecer apenas pela estrutura agregada dos dados combinada com a

implementação residual dos vetores de cada nó. O modelo simplificado considera apenas as duas últimas abordagens discutidas na estrutura hierárquica:

Figura 28: Estrutura Hierárquica com melhoria de Performance

Portanto, obtemos a seguinte equação logística para o custo:

$$\begin{aligned}
 J(\alpha, \beta, C^{Agg}, V^{Agg}) &= -C_{ij}^{Agg} \log\left(\frac{1}{1 + e^{-\theta}}\right) - (V_{ij}^{Agg} - C_{ij}^{Agg}) \log\left(\frac{e^{-\theta}}{1 + e^{-\theta}}\right) \\
 &+ \frac{\lambda_{\alpha}}{2} \sum_{i \in \mathcal{H}^P} \|\alpha_i\|^2 + \frac{\lambda_{\beta}}{2} \sum_{j \in \mathcal{H}^A} \|\beta_j\|^2 \\
 &+ \frac{\lambda_R}{2} \sum (\|\alpha_{Root}\|^2 + \|\beta_{Root}\|^2).
 \end{aligned}
 \tag{Equação 3.38}$$

A parte de regularização agora possui apenas a penalização dos vetores, não há mais a presença dos seus pais no cálculo. Θ mantém o mesmo equacionamento da Equação 3.27.

As atualizações para nós da hierarquia de páginas resulta:

$$\begin{aligned}
& \frac{\partial J(\alpha, \beta, C^{Agg}, V^{Agg})}{\partial \alpha_p} \\
&= \left(\beta_{jk} + \sum_{u \in Path(j) - \{j\}} \beta_{uk} \right)_p \left(\frac{1}{1 + e^\theta} \right) (-C_{ij}^{Agg} \\
&+ (V_{ij}^{Agg} - C_{ij}^{Agg})e^\theta) + \lambda_\alpha \alpha_p.
\end{aligned}$$

Equação 3.39

E para Beta:

$$\begin{aligned}
& \frac{\partial J(\alpha, \beta, C^{Agg}, V^{Agg})}{\partial \beta_p} \\
&= \left(\alpha_{ik} + \sum_{u \in Path(i) - \{i\}} \alpha_{uk} \right)_p \left(\frac{1}{1 + e^\theta} \right) (-C_{ij}^{Agg} \\
&+ (V_{ij}^{Agg} - C_{ij}^{Agg})e^\theta) + \lambda_\beta (\beta_p - Par(j)_p).
\end{aligned}$$

Equação 3.40

E w mantém a expressão da Equação 3.32.

5 – Método Híbrido com Dinâmica Temporal

A idéia é bastante semelhante, porém cada nó da hierarquia é associado, além do antigo vetor de variáveis latentes, um outro vetor para cada elemento de tempo em um conjunto pré-definido. Estes são, até então, os primeiros preditores baseados em dinâmica temporal discutidos na literatura de preditores de CTR baseados em técnicas de filtragem colaborativa e side-information integrados.

Diversos modelos foram implementados. A seguir discute-se cada variante.

MHDT1

No caso, divide-se o conjunto que modela o tempo no conjunto T que é dado por {madrugada, manhã, tarde, noite} e a divisão é tal que:

- Madrugada: das 0h às 6h.
- Manhã: das 6h às 12h.
- Tarde: das 12h às 18h .
- Noite: das 18h às 0h.

A Figura 29 a seguir ilustra a divisão empregada:

Figura 29: Divisão empregada no modelo dinâmico 1. Em cada divisão do dia há um registro de como sites e publishers interagiram com os anúncios.

A idéia com este modelamento é buscar padrões no comportamento dos usuários que se alteram ao longo do dia. Como exemplo, espera-se que o público que interage com

determinado site na parte da manhã tenha um perfil e interesses diferentes em relação ao grupo que visita o mesmo site de noite.

O que resulta no seguinte modelamento:

$$\theta(t) = \left[\left((\alpha_{ik} + \alpha_{ikt}) + \sum_{u \in Path(i) - \{i\}} (\alpha_{uk} + \alpha_{ukt}) \right) \cdot \left((\beta_{jk} + \beta_{jkt}) + \sum_{u \in Path(j) - \{j\}} (\beta_{uk} + \beta_{ukt}) \right) + \sum_{type} (w_{type} X_{type}) \right]. \quad \text{Equação 3.41}$$

Cada vetor com o sub-índice “t” indica um vetor latente associado ao respectivo nó em função da configuração do dia, ou seja, $t \in T$. As atualizações seguem as mesmas equações discutidas anteriormente. Como se trata de uma somatória, basta acrescentar os termos adicionados na equação para a etapa de atualização do algoritmo.

MHDT2

Modelo semelhante, porém desta vez o conjunto T tem maior nível de granularidade, representado pelas horas, logo $T = [0, 1, 2, \dots, 23]$ e novamente $t \in T$.

Figura 30: Modelo semelhante ao MDT1 porém a divisão do dia é feita com maior precisão.

MHDT3

Desta vez, considerou-se quais efeitos seriam obtidos variando-se a variável de peso w do side-information. As equações então são dadas por:

$$\theta(t) = \left[\left((\alpha_{ik} + \alpha_{ikt}) + \sum_{u \in \text{Path}(i) - \{i\}} (\alpha_{uk} + \alpha_{ukt}) \right) \cdot \left((\beta_{jk} + \beta_{jkt}) + \sum_{u \in \text{Path}(j) - \{j\}} (\beta_{uk} + \beta_{ukt}) \right) + \sum_{type} (w(t)_{type} X_{type}) \right]. \quad \text{Equação 3.42}$$

Novamente o nível de granularidade são as horas. O modelo desta vez visa rastrear como que as características intrínsecas dos anúncios ou sites interferem no processo de decisão das pessoas ao longo do dia. Como exemplo, anúncios com animações podem despertar maior interesse em um dado site de manhã e não muito de noite.

MHDT4

Para se verificar os efeitos dos nós mais altos na hierarquia sob o desempenho do preditor, considerou-se apenas a estratégia dinâmica em sites e anúncios, ou seja, os outros nós mais altos da hierarquia foram mantidos estáticos:

Figura 31: Apenas nós folhas possuem estrutura dinâmica temporal

O modelo resulta em:

$$\theta(t) = \left[\left((\alpha_{ik} + \alpha_{ikt}) + \sum_{u \in Path(i) - \{i\}} \alpha_{uk} \right) \cdot \left((\beta_{jk} + \beta_{jkt}) + \sum_{u \in Path(j) - \{j\}} \beta_{uk} \right) + \sum_{type} (w_{type} X_{type}) \right].$$

Equação 3.43

A estratégia deste modelo é eliminar os nós superiores o que diminui a sua complexidade, auxiliando o algoritmo a não causar overfitting. Outra questão é o fato de que as tendências temporais de nós superiores são constituídos pela agregação dos dados nos nós inferiores, o

que pode acabar encobrendo padrões dinâmicos. Como exemplo, ao se combinar tudo o que foi observado ao longo do tempo em cada site e agregar estas informações no publicador correspondente, muitos padrões podem ser anulados devido à mistura. Ao deixar apenas nós folhas rastreando tendências temporais aumenta-se também a precisão do modelo.

Figura 32: O nó superior registra uma mistura entre as tendências temporais de cada site que possui.

Vale notar que, independente da estratégia temporal utilizada, os vetores temporais são iniciados em nulo para não haver distorções na etapa de teste caso surja um ponto cujo horário não houve treinamento. Ao se inicializar em um vetor nulo, estes pontos não treinados não causam qualquer interferência na somatória em $\Theta(t)$ em função do tempo.

4 Resultados

Nesta seção, os resultados obtidos dos preditores implementados são apresentados. Primeiramente, os experimentos conduzidos são explicados bem como maiores explicações sobre os dados utilizados. Por fim, os desempenhos de cada abordagem é apresentada em função da verossimilhança obtida na base de validação-cruzada e base de teste.

4.1 Experimentos

Os resultados são expressos pelo log-likelihood ou verossimilhança (quão bem os CTRs descrevem os dados observados) tanto na base de validação cruzada (CV) quanto na base de teste.

Como exemplo, a Figura 33 a seguir ilustra, de forma simplificada, alguns campos presentes nos dados fornecidos pela empresa Lomadee e alguns valores fictícios utilizados apenas para melhor entendimento dos dados:

Site_id	Publicador_id	Anunciante_id	Campanha_id	Horário	Visualizações	Clicks
1	15	5001	13	13:00	100	6
2	15	5001	13	13:00	104	17
8	9	5700	7	16:00	223	2
10	41	5700	7	16:00	207	21

Figura 33: Exemplo dos dados fornecidos pela empresa Lomadee

Basicamente, cada horário registra o que cada site imprimiu e a quantidade de cliques obtidos naquele intervalo de tempo.

Os preditores são treinados em uma base de treinamento contendo dados de Dezembro de 2011 a 15 de Janeiro de 2012, o que resultou em 38.163.584 de pontos observados. Para validação cruzada, utilizou-se os dias 16 a 23 de janeiro de 2012 resultando em 8.282.542 e para teste o restante dos dados (os dados de treinamento ficaram divididos no mês de dezembro até o dia 15 de janeiro) o que resultou em 8.589.278. Demais informações sobre os dados serão omitidas em função dos mesmos serem de uma empresa privada.

Diversos valores para a dimensão dos vetores SVD foram testados, porém seguindo orientações dos pesquisadores em [6] e testes realizados com os dados recebidos estipulou-se para testes finais que f será 100 e a quantidade de iterações igual a 50.

Os algoritmos Pesos de Confiança, Side Information e Hierárquico foram implementados em um computador com as seguintes especificações:

Processador	Intel(R) Core(TM) i7-2600 CPU @ 3.40 GHz
RAM	8 GB DDR3
Sistema	Windows 7 64 Bits
HD	2 TB 7200

Tabela 3: Configuração do computador utilizado nos três primeiros modelos.

O modelo com dinâmica temporal foi testado em um computador com as seguintes especificações:

Processador	Intel(R) Core(TM) i7 X990 6 Cores 12 Threads @ 3.9 GHz
RAM	24 GB DDR3
Sistema	Linux 64 Bits
HD	2 HDs de 1 TB cada

Tabela 4: Configuração do computador utilizado para testes com algoritmo de dinâmica temporal.

Os tempos necessários para iteração dos algoritmos ficou próximo do que é relatado na Tabela 5 a seguir:

Algoritmo	Minutos por Iteração
Pesos Confiança	≈ 2
Side Information	≈ 3
Híbrido	≈ 15
Dinâmico	≈ 50

Tabela 5: Tempo aproximado consumido por iteração de cada algoritmo.

Cada algoritmo pode ser obtido nos seguintes endereços:

Pesos Confiança	https://www.dropbox.com/s/8l74lnxxjh4ad9u/PesosConfianca.zip
Side Information	https://www.dropbox.com/s/aftkuxagsizhd7n/SideInformation.zip
Híbrido	https://www.dropbox.com/s/p4cj6t2ei2xjj0z/ModeloHibrido.zip
Dinâmico	https://www.dropbox.com/s/9x4msl2aa0tw867/ModeloHibridoDinamico.zip

4.2 Resultados Obtidos

Para testar as estratégias implementadas, testou-se primeiramente a estratégia híbrida correndo para 10 iterações para se observar qual comportamento seria obtido. A seguir, a [Figura 34](#) ilustra testes realizados com a estratégia híbrida correndo para 10 iterações em função da dimensão dos vetores latentes.

Figura 34 : Erro log likelihood em função da dimensão f dos vetores.

Conforme esperado, há uma tendência do erro total diminuir ao se aumentar a dimensão f dos vetores, indicando que o algoritmo está funcionando corretamente pois quando usa mais conceitos para descrever a variância dos dados melhor é a performance do algoritmo.

A seguir, verificou-se na mesma estratégia o erro em função da quantidade total de iterações, obtendo-se o seguinte resultado:

Figura 35 : Erro loglikelihood em função do total de iterações na etapa de aprendizagem.

Para fins de comparação, a estratégia baseline discutida na Capítulo 2 foi implementada considerando-se um estimador bayesiano com perda quadrática. A seguir, resultados levantados de diversas distribuições Dirichlet variando-se o fator α é apresentado:

Figura 36: Desempenho preditor Baseline em função do parâmetro α .

Escolheu-se a distribuição de Dirichlet(1,1) para representar o preditor baseline neste trabalho.

Por fim, todas as estratégias (exceto a estratégia dinâmica que será discutida mais tarde) foram implementadas e comparadas e o resultado é apresentado na Figura 37:

Figura 37: Resultados em todas abordagens implementadas.

Observa-se que o preditor baseline erra com maior frequência em relação as outras abordagens. A diferença entre o baseline e a estratégia com pesos de confiança na fatoração já revela que o algoritmo de regressão logística pode adicionar informação suficiente para rastrear melhor os CTRs observados nos dados. Também é interessante notar que a abordagem usando side-information reduz consideravelmente o erro log-likelihood nos dados de treinamento, o que também explica o porquê que muitos preditores implementados só utilizavam estas informações dos anúncios na etapa de aprendizagem. A posição do anúncio na qual este é impresso, seu formato, dimensão e todas as outras características intrínsecas causam um impacto significativo no processo de decisão dos usuários. E, para finalizar, a estratégia híbrida para acrescentar informação hierárquica consegue reduzir o erro ainda mais, não apenas pela eliminação de enviesamento mas também pelo fornecimento de informação através de nós conhecidos que se espalha ao longo da hierarquia.

Testou-se então a estratégia temporal que apresentou os seguintes resultados para o modelo MHDT1 :

ξ	λ_a	λ_b	log Likelihood CV	log Likelihood Teste
0.01	0	0	divergiu	-
0.01	0	1	1409466	-
0.01	1	0	896791	-
0.01	1	1	671078	-
0.005	0	0	divergiu	-
0.005	0	1	1533100	-
0.005	1	0	846102	-
0.005	1	1	658710	174973
0.001	1	1	623463	148487

Tabela 6: Desempenho algoritmo MHDT1 estratégia não foi eficiente para melhorar o desempenho previamente obtido.

ξ é a taxa de aprendizagem e λ são os parâmetros de regularização do gradiente estocástico. Pontos em branco na coluna de teste representam pontos nos quais o teste não foi realizado, pois o erro na validação cruzada foi consideravelmente maior nesta estratégia que nas abordagens anteriores não dinâmicas. Diversos valores de ξ e λ foram testados, porém a tabela resume apenas os pontos mais relevantes.

Estes resultados revelam a importância do fator de regularização, em especial o fator λ_a que quando presente garante melhor convergência do preditor.

A abordagem não foi o suficiente para superar a estratégia anterior. Uma possibilidade testada foi o fato de que o conjunto T possuía apenas 4 divisões do dia, o que pode causar desvios suficientes para aumentar o erro. Testou-se também a estratégia cuja granularidade é cada hora do dia. O resultado segue a seguir apenas para pontos λ nulos pois os resultados no geral foram bastante próximos do que é mostrado a seguir:

ξ	log Likelihood CV	log Likelihood Teste
10^{-3}	514139	135429
10^{-7}	549262	92116
10^{-7} (vetores inicializados com $\epsilon = 10^{-3}$)	526529	124192

Tabela 7: Desempenho do algoritmo MHDT2 com horas do dia como nível de granularidade.

Apesar da diminuição do erro, ainda não foi possível superar a estratégia híbrida. Outro teste conduzido foi considerando que os pesos de side-information eram variáveis. Os resultados do MDHT3 foram (taxa de aprendizagem igual a 0.01 pois testes mostraram que este era o melhor valor possível):

λ_a	λ_b	λ_t	log Likelihood CV	log Likelihood Teste
0	0	0	divergiu	-
0	0	10^{-4}	divergiu	-
0	10^{-4}	0	divergiu	-
0	10^{-4}	10^{-4}	divergiu	-
10^{-4}	0	0	359653	-
10^{-4}	0	10^{-4}	449299	-
10^{-4}	10^{-4}	0	354684	87765
10^{-4}	10^{-4}	10^{-4}	380205	-

Tabela 8: MHDT3 reduziu erro mas ainda não superou estratégia híbrida pura.

Apesar da melhora em relação aos modelos temporais anteriores, ainda não foi possível superar o modelo híbrido estático. Novamente, observando-se a evolução dos fatores de regularização é possível concluir que o processo de overfitting deve ser avaliado com cuidado nesta etapa, pois o mesmo pode causar distorções consideráveis na etapa de predição.

O teste com o algoritmo MDHT4 considera que apenas as dinâmicas temporais de sites e anúncios são importantes para o aprendizado do algoritmo, o que torna o modelo menos

complexo e evita o efeito de overfitting na base de treinamento. O resultado foi (taxa de aprendizagem 10^{-2}):

λ_a	λ_b	log Likelihood CV	log Likelihood Teste
0	0	342728	72024
0	10^{-4}	360098	-
10^{-4}	0	357314	-
10^{-4}	10^{-4}	361230	-

Tabela 9: Estratégia MHDT4 superou estratégia anterior.

Os melhores resultados de cada estratégia dinâmica é mostrada a seguir:

Figura 38: Melhores resultados obtidos em cada estratégia dinâmica

A estratégia MHDT4 conseguiu, com sucesso, reduzir ainda mais o erro total observado nas estratégias desenvolvidas por [6]. Vale observar que para a estratégia dinâmica temporal funcionar foi necessário que apenas os nós folhas rastreassem as tendências em função do tempo. Os nós mais altos na estrutura, quando empregados para rastrear mudanças ao longo do tempo, acrescentaram desvios e prejudicaram as previsões feitas. A intuição por trás disto é

que um dado publicador pode ter sites que recebe mais clicks de manhã, outros sites que recebem mais de noite, o que pode causar desvios nos vetores que tentam ao mesmo tempo rastrear as duas tendências de uma só vez. Ao se colocar os vetores temporais associados apenas aos nós folhas este problema é eliminado.

Todos os resultados de cada abordagem implementada são mostrados a seguir:

O erro foi reduzido em, aproximadamente, **9.74 vezes** em relação ao preditor baseline, mostrando que as estratégias funcionaram e foram capazes de melhorar em cada nova abordagem o erro total produzido.

5 Conclusão

Esta seção descreve uma análise dos resultados, resume os pontos discutidos e por fim discute o que pode ainda ser implementado em trabalhos futuros.

5.1 Sumário

Este trabalho apresentou o desenvolvimento de preditores de CTR baseados em abordagens envolvendo regressão logística funcionando junto com conceitos empregados em sistemas de recomendação, em específico, filtragem colaborativa. Os dados das impressões dos anúncios e quantidade de Clicks (no presente trabalho fornecidos pela empresa Lomadee) são considerados como observações de Clicks e Não Clicks (evento positivo e negativo). O modelo de regressão logística então é considerado uma função de custo com pesos que adicionam ao erro total os clicks e “visualizações – clicks” observados.

Adiciona-se então modelos que integram o side-information que são basicamente os atributos dos anúncios e sites, como categoria a que pertencem (esportes, lazer, música e tudo o mais), tamanho do banner, posição, animações dentre outras. O erro, conforme observado, reduz-se consideravelmente.

Um modelo com estruturas hierárquicas é também implementado e reduz ainda mais o erro na base de teste. Isto ocorre pois é possível em uma estrutura hierárquica obter informação de pontos bem conhecidos na rede e “transportar” ela para pontos nos quais poucas observações foram feitas.

De posse de todos estes modelos, um algoritmo híbrido é proposto juntando, em uma só abordagem, todas as abordagens discutidas até então, o que é possível dado que uma função exponencial decrescente modela o equacionamento do CTR, garantindo que seu valor variará entre zero e um.

Por fim, um modelo dinâmico foi proposto, no qual se considerou que vetores associados a divisões no dia adicionam tendências na taxa de conversão. Os resultados mostram que alguns

problemas surgem, como o excesso de over-fitting, o que prejudica a predição. No entanto, um modelo que considerou dinâmica temporal apenas nos nós folhas conseguiu reduzir o erro em relação ao modelo estático, revelando que é possível melhorar a acurácia dos preditores implementados.

5.2 Trabalhos Futuros

Observando-se os resultados obtidos na estratégia dinâmica é possível concluir que modelos temporais podem facilmente causar over-fitting na etapa de aprendizagem. Várias suposições e testes devem ser executados para se desenvolver algoritmos que tenham melhor acurácia. Pode-se, no entanto, buscar novos tipos de filtragens para melhorar a performance. Uma possibilidade seria considerar desvios temporais semelhantes aos considerados no modelo temporal de [17]. Por exemplo, dezembro é uma data marcada pela comemoração do Natal, então é de se esperar que haverá desvios na taxa de clicks em certos anúncios. O mesmo vale para janeiro, época na qual os pais preparam seus filhos para a escola. Tais desvios podem ser filtrados no modelo e eliminados na etapa de predição, potencialmente melhorando os resultados já obtidos.

Outras divisões também podem ser propostas, como analisar as diferenças entre as tendências em dias de semana e dias de final de semana, podendo esta abordagem fornecer novos filtros aos dados temporais.

Outras possibilidades poderiam envolver uma análise espectro-temporal dos dados através da decomposição de Fourier e utilizar as frequências com maiores amplitudes para auxiliar na predição futura dos dados, porém esta abordagem pode tornar o custo computacional consideravelmente maior, podendo até mesmo se tornar inviável de ser implementado [67].

6 Referências Bibliográficas

1. JANSEN, B. J.; BROWN, A.; RESNICK, M. **Factors relating to the decision to click on a sponsored link.** [S.l.]. 2007.
2. CHENG, H. **Personalized Click Prediction in Sponsored Search.** New York. 2010.
3. BRODER, A.; JOSIFOVSKI, V. Stanford University, 2010. Disponível em: <<http://www.stanford.edu/class/msande239/>>. Acesso em: 12 set. 2012.
4. MANNING, C. **Introduction to Information Retrieval.** [S.l.]: IEEE International Conference, 2008.
5. SCHWARTZ, B. **The Paradox of Choice.** New York. 2004.
6. MENON, K. A. et al. **Response Prediction Using Collaborative Filtering with Hierarchies and Side-Information.** San Diego. 2011.
7. MOBASHER, A. S. S. **Intelligent Techniques for Web Personalization.** [S.l.]. 2005.
8. WANG, X. et al. **Click-Through Rate Estimation for Rare Events in Online Advertising.** Silicon Valley. 2010.
9. CHEN, Y.; PAVLOV, D.; CANNY, J. F. **Large-Scale Behavioral Targeting.** Paris. 2009.
10. GLUHOVSKY, I. **Forecasting Click-Through Rates Based on Sponsored Search Advertiser Bids and Intermediate Variable Regression.** [S.l.]. 2010.
11. GRAEPEL, T. et al. **Web-Scale Bayesian Click-Through Rate Prediction for Sponsored Search Advertising in Microsoft's Bing Search Engine.** [S.l.]. 2009.
12. ROSALES, R.; CHENG, H.; MANAVOGLU, E. **Post-Click Conversion Modeling and Analysis for Non-Guaranteed Delivery Display Advertising.** Seattle. 2012.
13. XU, W.; MANAVOGLU, E.; PAZ, E. C. **Temporal Click Model for Sponsored Search.** [S.l.]. 2010.
14. HERLOCKER, J.; KONSTAN, J.; RIEDL, J. **Explaining Collaborative Filtering Recommendations.** [S.l.]. 2000.
15. PAZZANI, M. J. **A Framework for Collaborative, Content-based and Demographic Filtering.** [S.l.]. 1999.
16. TUZILIN, A. G. **Toward the Next Generation of Recommender Systems: A Survey of the State of the Art and Possible Extensions.** [S.l.]. 2005.
17. KOREN, Y. **The BellKor Solution to the Netflix Grand Prize.** [S.l.]. 2009.

18. CANNY, J. F. **Collaborative Filtering with Privacy**. [S.l.]. 2002.
19. WITTEN, I. H.; FRANK, E.; HALL, M. A. **Data Mining: Practical Machine Learning Tools and Techniques**. [S.l.]: Kauffmann, 2011.
20. PAPAGELIS, M. et al. **Incremental Collaborative Filtering for Highly-Scalable Recommendation Algorithms**. [S.l.]. 2005.
21. ANDERSON, M. et al. **A Rule-Appling Collaborative Filtering System**. [S.l.]. 2003.
22. BANERJEE, S.; RAMANATHAN, K. **Collaborative Filtering on Skewed Datasets**. [S.l.]. 2008.
23. KARYPIS, S. B. M. **Application of Dimensionality Reduction in Recommender Systems: a Case Study**. [S.l.]. 2000.
24. KOREN, Y.; BELL, R. M.; VOLINSY, C. **Matrix Factorization Techniques for Recommender Systems**. [S.l.]: IEEE. 0018-9162/09, 2009.
25. NG, A. Coursera. **Coursera**, 01 set. 2011. Disponivel em: <www.cousera.org>. Acesso em: 28 ago. 2012.
26. ZHANG, F. **Research on Recommendation List Diversity of Recommender Systems**. [S.l.]. 2008.
27. SARWAR, B. et al. **Incremental Singular Value Decomposition Algorithms for Highly Scalable Recommender Systems**. [S.l.]. 2002.
28. RICCI, F. et al. **Recommender Systems Handbook**. [S.l.]: Springer, 2010.
29. SARWAR, B. M.; KONSTAN, J. A.; RIEDL, J. **Distributed Recommender Systems for Internet Commerce**. [S.l.]. 2005.
30. LIDEN, G.; SMITH, B.; YORK, J. **Amazon.com Recommendations: Item-to-Item Collaborative Filtering**. [S.l.]. 2003.
31. SCHAFER, J. B.; KONSTAN, J. A.; RIEDL, J. **E-commerce Recommendation Applications**. *Data Mining and Knowledge Discovery* 5(1/2), 115-153. [S.l.]. 2001.
32. MAHMOOD, T.; RICCI, F. **Improving Recommender Systems with Adaptive Conversational Strategies**. [S.l.]: ACM pp 73-82, 2009.
33. CARENINI, G.; SMITH, J.; POOLE, D. **Towards More Conversational and Collaborative Recommender Systems**. New York. 2003.
34. MASSA, P.; AVESANI, P. **Trust-Aware Collaborative Filtering for Recommender Systems**. [S.l.]. 2004.
35. MCNEE, S. M.; RIEDL, J.; KONSTAN, J. A. **Being Accurate Is Not Enough: How Accuracy Metrics Have Hurt Recommender Systems**. New York. 2006.

36. SHOKRI, R. et al. **Preserving Privacy in Collaborative Filtering Through Distributed Aggregation of Offline Profiles**. New York. 2009.
37. RESNICK, P.; VARIAN, H. R. **Recommender Systems**. *Communications of the ACM* 40(3), pp 56-58. [S.l.]. 1997.
38. MCCARTHY, K.; SALAMÓ, M.; COYLE, L. **Group Recommender Systems: a Critique Based Approach**. Paris. 2006.
39. BRIDGE, D.; MCGINTY, M. **Case-Based Recommender Systems**. *The Knowledge Engineering Review*, 315-320. [S.l.]. 2006.
40. SHOHAM, B. M. **Content-Based, Collaborative Recommendation**. [S.l.]. 1997.
41. RICCI, F. **Travel Recommender Systems**. [S.l.]. 2002.
42. COHEN, W. **Fast Effective Rule Induction**. [S.l.]. 1995.
43. GROH, G.; EHMIG, C. **Recommendations in Tate Related Domains: Collaborative Filtering vs. Social Filtering**. New York. 2007.
44. BERKOVSKY, S. **Mediation of User Models: Enhanced Personalization in Recommender Systems**. [S.l.]. 2009.
45. SINHA, R. R.; SWEARINGEN, K. **Comparing Recommendations Made by Online Systems and Friends**. [S.l.]. 2001.
46. ZHOU, B.; HUI, S.; CHANG, K. **An Intelligent Recommender System Using Sequential Web Access Patterns**. [S.l.]. 2004.
47. BURKE, R. **Hybrid web recommender systems**. Berlim. 2007.
48. CHIRSTAKOU, C.; STAFYLOPATIS, A. **A Hybrid Movie Recommender System Based on Neural Networks**. [S.l.]. 2005.
49. TAGHIPOUR, N.; KARDAN, A. **A Hybrid Web Recommender System Based on Q-Learning**. Fortaleza. 2008.
50. ZIEGLER, C. N. et al. **Improving Recommendation Lists Through Topic Diversification**. New York. 2005.
51. BREESE, J.; HECKERMAN, D.; KADIE, C. **Empirical Analysis of Predictive Algorithms for Collaborative Filtering**. [S.l.]. 1998.
52. BRASSARD, A. E. et al. **A Privacy-Preserving Recommender System for Electronic Commerce**. [S.l.]. 2008.

53. CONNOR, M.; HERLOCKER, J. **Clustering Items for Collaborative Filtering**. [S.l.]. 2001.
54. GOLDBERG, D. et al. **Using Collaborative Filtering to Weave an Information Tapestry**. [S.l.]. 1992.
55. SCHAFER, J. B. et al. **Collaborative Filtering Recommender Systems**. Berlin. 2007.
56. VAN ROY, B.; YAN, X. **Manipulation-Resistant Collaborative Filtering Systems**. New York. 2009.
57. COSLEY, D. L.; ALBERT, I. **Is Seeing believing? How Recommender Systes Interfaces Affect User's Opinions**. New York. 2003.
58. GEORGE, T.; MERUGU, S. **A Scalable Collaborative Filtering Framework Based on Co-Clustering**. [S.l.]. 2005.
59. HERLOCKER, J. L. et al. **Evaluating Collaborative Filtering Recommender Systems**. [S.l.]. 2004.
60. PIOTTE, M.; CHABBERT, M. **The Pragmatic Theory Solution to the Netflix Grand Prize**. [S.l.]. 2009.
61. WIKIPEDIA. Wikipedia, 2013. Acesso em: 18 February 2013.
62. KOLLER, D.; FRIEDMAN, N. **Probabilistic Graphical Models: Principles and Techniques**. [S.l.]: The MIT Press; 1 edition, 2009. ISBN 0262013193.
63. FRIEDMAN, N.; GEIGER, D.; GOLDSZMIDT, M. **Bayesian Network Classifiers, Mach. Learn., 29(2-3): pp 131-163**. [S.l.]. 1997.
64. AGARWAL, D.; AGRAWAL, R. **Estimating Rates of Rare Events with Multiple Hierarchies through Scalable Log-Linear Models**. Wasington. 2010.
65. ACM. ACM Recommender Systems 2012, 01 jan. 2012. Disponivel em: <www.recsys.acm.org/2012>. Acesso em: 27 ago. 2012.
66. FREUND, Y. et al. **An Efficient Boosting Algorithm for Combining Preferences**. [S.l.]. 2003.
67. KUTZ, N. Coursera, 2013. Disponivel em: <<https://class.coursera.org/compmethods-2012-001/lecture/index>>. Acesso em: 18 February 2013.
68. TOSCHER, A.; JAHNER, M. **The BigChaos Solution to the Netflix Grand Prize**. [S.l.]. 2009.
69. MICHIE, D.; SPIEGELHALTER, D. J.; TAYLOR, C. C. **Machine Learning, Neural and Statistical Classification**. [S.l.]. 1994.
70. OSUNA, R. G. **Lecture 8: The K Nearest Neighbor Rule**. [S.l.].
71. MITCHELL, T. M. **Machine Learning**. [S.l.]: McGraw Hill, 1997.
72. GAL, B. **Bayesian Networks, Encyclopedia of Statistics in Quality & Reliability**. [S.l.]. 2007.

73. ROMERO. Buscapé. **Buscapé**, 1999. Disponível em: <www.buscapi.com.br>. Acesso em: 29 ago. 2012.
74. OVERSTOCK. OverStock. **OverStock**, 1999. Disponível em: <www.overstock.com>. Acesso em: 29 ago. 2012.
75. **Unsupported source type (DocumentFromInternetSite) for source Bak05.**
76. POLAT, H.; DU, W. **Privacy-Preserving Collaborative Filtering Using Randomized Perturbation Techniques**. Melbourne. 2003.
- Sarwar, B., et al., et al. *Incremental Singular Value Decomposition Algorithms for Highly Scalable***
77. **Recommender Systems**. s.l. : Proceedings of the 5th International Conference in Computers and Informatin Technology, 2002.
78. TAGHIPOUR, N; KARDAN, A. **Usage-Based Web Recommendations: A Reinforcement Learning Approach**. Minneapolis. 2007.
79. TSAO, Y. W.; YUAN, S. T. **A Recommendation Mechanism For Contextualized Mobile Advertising. Expert Systems With Applications**. [S.l.]. 2003.
80. VAN SETTEN, M.; MCNEE, S. M. **Beyond Personalization: the Next Stage of Recommender Systems Research**. [S.l.]. 2005.
81. WANG, J. et al. **Distributed Collaborative Filtering for Peer-to-Peer File Sharing Sytems**. [S.l.]. 2006.
82. WANG, Y.; KOBASA, A. **Performance Evaluation of a Privacy-Enhancing Framework for Personalized Websites**. [S.l.]. 2009.
- Hu, Yifan, Koren, Yehuda e Volinsky, Chirs. Collaborative Filtering for Implicit Feedback Datasets.**
83. **AT&T Research**. [Online] 2008. [Citado em: 20 de February de 2013.]
<http://www2.research.att.com/~yifanhu/PUB/cf.pdf>.

7 Apêndice

Neste apêndice, descreve-se algumas idéias que foram testadas com os dados Lomadee para implementação de preditores de CTR com uma estratégia baseada em SVD mas cuja função custo não é baseada em regressão logística. Três etapas são discutidas, na qual cada uma discute diferentes abordagens testadas e os testes realizados. Vale lembrar que estas soluções não estão presentes na implementação do algoritmo final apresentado neste documento.

Etapas 1

O projeto se propõe a desenvolver preditores que obtenham com a maior acurácia possível a taxa de conversão de clicks esperada quando uma página, doravante chamada de source, expõe um banner com produtos (destination) em um determinado horário do dia. Os atributos considerados são:

- Data: horário da ocorrência da impressão
- Source id: site hospedeiro que exibe o banner(publisher)
- Destination id: endereço de direcionamento ao produto ao se clicar em determinado banner
- Impressões: quantidade total que determinado banner teve em dado source id e destination id
- Clicks: quantos usuários clicaram nos mesmos.

Outros atributos podem ser utilizados no desenvolvimento de preditores, como dados sobre as categorias tanto dos sources quanto destinations, bem como dados mais gerais.

Há 3 diferentes etapas nesta vertente de pesquisa: na primeira, os mesmos preditores aplicados em [17] foram adaptados para rastrear taxas de conversão de clicks; na segunda, uma estratégia “neighborhood” foi implementada (a mesma será melhor explicada a seguir) e por fim, na terceira etapa, um paper selecionado da equipe do Yahoo! Labs foi implementado.

A seguir, discute-se o desenvolvimento teórico da etapa 1.

Preditores de Base

A primeira idéia a ser implementada é a obtenção de tendências nas taxas de conversão entre os itens, ou seja, investigar sites que produzem taxas de conversão acima/abaixo da média, bem como destinations que produzem o mesmo efeito.

Uma notação matemática se faz necessária. Sites (source_ids) serão representados por “s”, destinations por “d”. A taxa de conversão estática (sem incorporar efeitos temporais) entre um source_id e um destination_id é dada por $ad_{s,d}$. Caso o tempo seja incorporado, $ad_{s,d,t}$. Dado a amostragem realizada e mantendo a notação em [17], temos o conjunto $K = \{(s,d) \mid ad_{s,d} > 0, \text{ para algum instante de tempo } t \text{ tal que } t \text{ esteja no conjunto de treinamento}\}$. $R(s)$ indica todos os destinations que se associam ao source_id s e que estão presentes na base amostrada, com taxa de adesão superiores a 0. $N(s)$ é o conjunto de todos os destinations que se ligam ao source s, mesmo que a taxa de conversão seja nula. $R(d)$ e $N(d)$ têm a mesma definição porém tratando-se de destinations.

Propõe-se, inicialmente, a aplicação de preditores de base sem dinâmica temporal, equacionada por:

$$ad_{s,d} = \overline{ad}_{s,d} + b_s + b_d .$$

Onde $\overline{ad}_{s,d}$ se refere à taxa de conversão média encontrada na base amostrada. b_s indica os desvios observados de dado source_id. Por exemplo, suponhamos que a média seja de 0.2 (20% de taxa de conversão), mas o source s atrai usuários que não gostam tanto de clicar em banners, independente do banner, sendo $b_s = -0.1$. Porém suponhamos que o destination impresso costuma despertar a curiosidade nas pessoas que eventualmente clicam mais, fornecendo $b_d = 0.15$. A taxa de conversão esperada neste caso é de $0.2 - 0.1 + 0.15 = 0.25$. Há basicamente duas formas de se obter os parâmetros de tendência b_s e b_d ; via equacionamento:

$$b_d = \frac{\sum_{s \in R(d)} (ad_{s,d} - \overline{ad}_{s,d})}{\lambda_2 + |R(d)|}.$$

e

$$b_s = \frac{\sum_{d \in R(s)} (ad_{s,d} - \overline{ad}_{s,d} - b_d)}{\lambda_3 + |R(s)|}.$$

Apesar de mais diretos, tais equações se mostraram menos precisas que técnicas de redução de custo, portanto a seguinte equação é empregada neste projeto:

$$J(b_*, \lambda_1) = \sum_{(s,d) \in K} (\overline{ad}_{s,d} + b_s + b_d - ad_{s,d})^2 + \lambda_1 \left(\sum_{s \in K} b_s^2 + \sum_{d \in K} b_d^2 \right).$$

Na qual os parâmetros b_* são obtidos via gradiente estocástico.

$$b_{*i} := b_{*i} - \alpha * \frac{\partial J(b, \lambda)}{\partial b_{*i}}.$$

Substituindo as equações:

$$b_{*i} := b_{*i} * - \alpha \sum_{(s,d) \in K} (\overline{ad}_{s,d} + b_s + b_d - ad_{id}).$$

Apesar do método de gradiente estocástico poder ser realizado eficazmente, faz-se a importante observação de que os dados para serem processados devem obrigatoriamente serem transferidos à memória RAM, caso contrário o tempo estimado de processamento ficará em torno de dias.

Como em todas as abordagens, parte da base é utilizada para aprendizagem, outra para validação cruzada e por fim a última parte é destinada ao teste de fato.

Para a etapa de teste, o valor MAE é proposto:

$$MAE = \frac{\sum_{s,d \in T} (|\widehat{ad}_{s,d} - ad_{s,d}|) * I}{|T|}.$$

Fatoração de Matriz

[17] sugere que a extração das relações entre sources e destinations são melhor obtidas via técnicas de decomposição Singular Value Decomposition (**SVD**), obtendo-se para isto um espaço de variáveis latentes que explicam parte da variância dos dados originais. O resultado é um modelo que descreve através destas variáveis as relações entre source e destination pelo produto interno de dois vetores que podem automaticamente extrair relações importantes, como sites de tecnologia que ao exporem produtos da mesma categoria ou sobre livros, ciências, inovação causa um aumento na taxa de adesão. Constrói-se então o vetor $q_d \in \mathbf{R}^f$ (f é a dimensão resultante da decomposição) relacionado à categoria do banner impresso, bem como $p_s \in \mathbf{R}^f$. O vetor q_d mede a intensidade de correlação que determinada categoria possui com cada um dos fatores f . O mesmo vale para p_s , que extrai o quanto cada source se relaciona com cada fator. O resultado da taxa final esperada pode ser incorporada com os preditores de base (referentes às relações que não envolvem diretamente as relações entre os banners e os sites) com as relações melhores descritas pela decomposição dos dados, fornecendo:

$$\widehat{ad}_{s,d} = \overline{ad}_{s,d} + b_d + b_s + q_d^T \cdot p_s.$$

Onde $\widehat{ad}$ chapéu corresponde à predição da taxa de conversão. Esta estratégia sera chamada de **SVD**.

Os parâmetros podem ser aprendidos através de redução de função custo, desta vez dada por:

$$J(b_*, q_*, p_*) = \sum_{(s,d) \in K} (-ad_{s,d_{id}} + \overline{ad}_{s,d} + b_d + b_s + q_d^T \cdot p_s)^2 + \lambda_2 (b_d^2 + b_s^2 + \|q_d\|^2 + \|p_s\|^2).$$

Novamente a obtenção de mínimo pode ser obtida via procedimento de gradiente estocástico, com as seguintes atualizações:

$$\begin{aligned} cost(h_{\theta}(row_{id}) - ad_{s,d_{id}}) \\ = \frac{1}{2} (\overline{ad}_{s,d} + b_s + b_d + q_d^T \cdot p_s - ad_{s,d_{id}})^2 + \lambda_2^2 (b_s^2 + b_d^2 + |p_s^2| + |q_s^2|). \end{aligned}$$

Atualização:

$$\theta := \theta - \alpha \frac{\partial cost}{\partial \theta}.$$

$$b_{s_i} = b_{s_i}(1 - \alpha\lambda_2) - \alpha(\overline{ad}_{s,d} + b_s + b_d + q_d^T \cdot p_s - ad_{s,d_{id}}).$$

$$q_{d_i} = q_{d_i}(1 - \alpha\lambda_2) - \alpha(\overline{ad}_{s,d} + b_s + b_d + q_d^T \cdot p_s - ad_{s,d_{id}}).$$

$$p_{s_i} = p_{s_i}(1 - \alpha\lambda_2|p_s|) - \alpha(\overline{ad}_{s,d} + b_s + b_d + q_d^T \cdot p_s - ad_{s,d_{id}}) * q_{d_i}.$$

$$q_{s_i} = q_{s_i}(1 - \alpha\lambda_2|q_d|) - \alpha(\overline{ad}_{s,d} + b_s + b_d + q_d^T \cdot p_s - ad_{s,d_{id}}) * q_{d_i}.$$

$$r = \varepsilon * (random(0,1) * 2 - 1)$$

SVD++

É possível, teóricamente, melhorar os resultados obtidos previamente através da introdução do feedback implícito. No presente projeto, tal informação pode ser obtida através da análise de quais sources receberam clicks ao expor dado destination ou não. A idéia é que alguns sources ao exporem banners produziram implicitamente uma interação com o usuário, que pode escolher em clicar ou não; se um determinado tipo de produto atrai um determinado tipo de site, esta informação pode auxiliar na taxa de predição de clicks. A inclusão do feedback implícito é feita através da inclusão de fatores implícitos que se somam aos fatores do usuário p_s , resultando matematicamente:

$$\widehat{ad}_{s,d_i} = \overline{ad}_{s,d} + b_s + b_d + q_d^T \cdot \left(p_s + |R(s)|^{-0.5} \sum_{j \in R(s)} y_j \right).$$

Pode-se observar que os parâmetros implícitos são associados ao vetor de conceitos de cada source amostrado e a função custo pode ser reduzida por gradiente estocástico novamente, levando às seguintes equações:

$$e_{s,d} = \widehat{ad}_{s,d_i} - ad_{s,d_i}.$$

$$b_s = b_s \cdot (1 - \lambda_1 * \alpha) - \alpha(e_i).$$

$$b_d = b_d(1 - \lambda_1 * \alpha) - \alpha(e_i).$$

$$q_d = q_d(1 - \lambda_2 * \alpha) - \alpha \cdot e_i \left(p_s + |R(s)|^{-0.5} \sum_{j \in R(s)} y_j \right).$$

$$p_s = p_s(1 - \lambda_2 * \alpha) - \alpha * e_i * q_d.$$

$$y_j = y_j(1 - \lambda_2 * \alpha) - \alpha(e_i * |R(s)|^{-0.5} q_d).$$

timeSVD

Espera-se que a inclusão da parte dinâmica nos preditores de base aumente consideravelmente a acurácia na predição. Sugere-se a seguinte modelagem:

$$ad_{s,d} = \overline{ad}_{s,d} + b_s(t) + b_d(t).$$

Resultando:

$$\widehat{ad}_{s,d_i} = \overline{ad}_{s,d} + (b_s + b(t)_s) + b_d + q_d^T(p_s).$$

Na qual $b(t)_s$ possui um parâmetro para cada horário do dia. A função custo é dada por:

$$J_{b,p,q} = \sum (\widehat{ad}_{s,d} - ad_{s,d_i})^2 + \lambda_1 \sum (b^2 + b(t)^2 + |p_s|^2 + |q_d|^2).$$

Cuja redução é feito por gradiente estocástico.

Implementada esta estratégia, tem-se funcionando para predição de clicks aquela que é considerada o estado da arte dentre os conjuntos de técnicas aplicadas em Filtragem Colaborativa.

Algumas adaptações podem ser testadas a fim de se analisar com melhores métricas o desempenho dos preditores.

SpartaVD

A estratégia é semelhante ao timeSVD já implementado, porém mais preciso. A taxa de conversão é modelada como:

$$\widehat{ad}_{s,d} = q_d^T \cdot (p_s(t)).$$

Observa-se que já não se rastreia mais fatores de tendência intrínsecos. Esta abordagem foi proposta devido ao fato de a grande maioria das conversões serem nulas, anulando também qualquer bias. Modela-se então considerando-se apenas as relações entre o site e o destination, extraídas pela estratégia SVD. O fator $p_s(t)$ é modelado com 3 componentes: p_s , a parte estática das preferências observadas dos usuários que visitaram o site, p_{s, t_1} modela as variações de comportamento esperados em $t_1 \in \{\text{madrugada, manhã, tarde, noite}\}$. Por exemplo, em cada parte do dia supõe-se que um tipo de usuário irá interagir com os banners expostos; o fator t_1 então visa rastrear as tendências e gostos em função de como variam ao longo do dia. Madrugada é definida das 00:00 às 06:00, Manhã das 06:00 às 12:00 e assim por diante.

Por fim, há o fator p_{s, t_2} que visa obter pontos de influência em cada dia observado na base de treinamento. Por exemplo, se um anúncio no dia primeiro de outubro em dado site teve taxa de conversão nula porém no dia seguinte obteve uma taxa extremamente alta (eventualmente 100%) e nos outros dias taxas nulas novamente, o parâmetro modelado por t_2 visa rastrear estes ‘picos’ ou fortes variações diárias para filtrar melhor os dados (denoising). Não é possível aplicar este parâmetro em predições na base de novembro, afinal comportamentos atípicos acontecidos no dia 01 não irão necessariamente acontecer novamente no mês seguinte, portanto para a etapa de predição este parâmetro é setado em 0, reforçando seu caráter de filtragem de ruídos do sinal.

Um ponto importante de se observar é que desta vez o algoritmo proposto é implementado visando otimizar o parâmetro F_1 -Score, para melhorar a taxa de predição de pontos nos quais

ocorre click. Outra alteração é que não se tenta otimizar a taxa de adesão (não diretamente), mas sim a quantidade direta de clicks. Matematicamente a função custo modelada é dada por:

$$J = \sum (q_d^T \cdot [p_s + p_{s,t1} + p_{s,t2}] * I - \text{clicks}_i)^2 + \lambda \sum (|p_s|^2 + |p_{s,t1}|^2 + |p_{s,t2}|^2).$$

Onde I é o total de impressões por banner. Resulta as seguintes atualizações por gradiente estocástico:

$$q_{d_i} = q_{d_i} * (1 - \alpha\lambda) - \alpha(q_d^T \cdot [p_s + p_{s,t1} + p_{s,t2}] * I - \text{clicks}_i) * I * p_{s_i}.$$

$$p_{s_i} = p_{s_i} * (1 - \alpha\lambda) - \alpha(q_d^T \cdot [p_s + p_{s,t1} + p_{s,t2}] * I - \text{clicks}_i) * I * q_{d_i}.$$

$$p_{s,t1_i} = p_{s,t1_i} * (1 - \alpha\lambda) - \alpha(q_d^T \cdot [p_s + p_{s,t1} + p_{s,t2}] * I - \text{clicks}_i) * I * q_{d_i}.$$

$$p_{s,t2_i} = p_{s,t2_i} * (1 - \alpha\lambda) - \alpha(q_d^T \cdot [p_s + p_{s,t1} + p_{s,t2}] * I - \text{clicks}_i) * I * q_{d_i}.$$

Teoria utilizada na etapa 2:

Os seguintes dados serão utilizados:

- Dia
- DiaDaSemana
- Mes
- Hora
- AdCreativeTypeID
- AdDimenisonID
- AdPrimaryCategoryID
- PubType
- PublicadorID
- AdID
- AdTypeID
- PubPrimaryCategoryID
- Impressions

- Clicks
- ID

Por motivos de segurança não serão divulgadas informações sigilosas da empresa, porém vale notar que observando a distribuição dos dados, algumas dimensões estão bem distribuídas, outras nem tanto, porém os dados são suficientes para a construção de preditores eficientes.

Estratégias para Desbalanceamento dos Dados

Para melhor tratar a característica da base em relação ao desbalanceamento dos dados, algumas amostragens são propostas:

- Amostragem com e sem reposição.
- Pares Selecionados.
- Bootstrap.
- Bias Bootstrap.
- Tomek Link.

O que se propõe é que os preditores sejam modelados para cada tipo de amostragem e testes de desempenho sejam levantados na base de teste.

Modelo Neighborhood

Dado que a nova base de dados empregada na pesquisa enseja a utilização de mais atributos, propõe-se testes utilizando-se da estratégia neighborhood para verificar eventuais melhoras nos preditores. Vale notar que a estratégia baseada em extração de fatores latentes descrevem de forma mais rica a variância dos dados, porém tal fato pode não se valer frente à nova estrutura empregada. Propõe-se uma abordagem baseada nos modelamentos de [28], Capítulo 5, através de vizinhos baseados em anúncios; a mais popular em Filtragem Colaborativa nos forneceria o seguinte equacionamento:

$$\hat{y}_{Pub,Adi} = b_{Pub,Adi} + \frac{\sum_{j \in S^k(Adi, Pub)} S_{Adi,Adj} (y_{Pub,Adj} - b_{Pub,Adj})}{\sum_{j \in S^k(Adi, Pub)} S_{Adi,Adj}}.$$

Onde b^* representa o enviesamento observado em dado Publicador que expõe o banner i . S^k é o conjunto dos k banners que dado Publicador imprimiu e que mais se assemelham ao Adi , sendo $s_{adi, adj}$ esta medição; y é a taxa de conversão da quantidade de impressões ocorridas, variando de 0 a 1 para ensejar convergência.

Apesar de popular, simples e direta, tal equacionamento não é por completo interessante aos dados Lomadee, pelos seguintes motivos:

- Um publicador pode por diversas vezes mostrar o mesmo banner em tempos futuros, o que força que o modelo empregado seja temporal para contabilizar tais efeitos (a ser discutido a seguir)
- Por definição, os coeficientes de similaridade somam 1. Mas caso aconteça de um dado anúncio não ter muitos vizinhos próximos, ainda sim sua soma deve ser 1, o que distorce a realidade, forçando números que deveriam se aproximar de 0 a ficarem próximos de 1.
- É também preciso considerar o fato de que se a variância das taxas de conversão forem muito altas o método pode obter vizinhos randomicos.

Uma forma proposta para se lidar com estas desvantagens é modelar as interações entre publicador e banner através de pesos de interporlação conjunta θ :

$$\hat{y}_{Pub,Adi} = b_{Pub,Adi} + \sum_{j \in S^k(Adi, Pub)} \theta_{Adi, Adj}^{Pub} (y_{Pub, Adj} - b_{Pub, Adj}).$$

Vale notar que o modelo ainda não é temporal, portanto ainda não é adequado.

Uma melhoria interessante que se integra com abordagens previamente testadas visa a implementação de um equacionamento mais global. O problema é que os métodos descritos em neighborhood têm uma atuação mais localizada, na qual se considera, para dado publicador, os k anúncios mais correlacionados ao i -ésimo banner da análise, o que pode ser contornado (e desta adaptação é que a equação se torna de solução “global”) através do seguinte equacionamento:

$$\hat{y}_{Pub,Adi} = b_{Pub,Adi} + \sum_{j \in R(Pub)} (y_{Pub,Adj} - b_{Pub,j}) * w_{Adi,Adj}.$$

A somatória desta vez é realizada em todos os anúncios que dado publicador imprimiu, não apenas os k mais correlatos o que torna os pesos $w_{Adi,Adj}$ em soluções mais generalistas (globais). Antes, cada θ exprimia para um dado publicador as relações entre os anúncios. Agora cada w estabelece uma relação direta sem depender de dado publicador. Tal formulação permite uma maior flexibilidade para se integrar novos parâmetros, como fatores implícitos, levando à alternativa:

$$\hat{y}_{Pub,Adi} = b_{Pub,Adi} + \sum_{j \in R(Pub)} [(y_{Pub,Adj} - b_{Pub,Adj}) * w_{Adi,Adj} + c_{Adi,Adj}].$$

No qual os pesos $c_{Adi, Adj}$ contabilizam por adições dado que houve impressão de um Ad em um publicador. Nesta mesma linha, propõe-se o seguinte modelo:

$$\begin{aligned} \hat{y}_{Pub,Adi} = & b_{Pub,Adi} + |R(pub)^{-\alpha}| * \sum_{j \in R(Pub)} [(y_{Pub,Adj} - b_{Pub,Adj}) * w_{Adi,Adj} + c_{Adi,Adj} \\ & + \frac{e_{Adi,Adj}}{1 + |Area_{Adi} - Area_{Adj}|} + f_{Adi,Adj} \delta_{Adi}(visibility_j) \\ & + g_{Adi,Adj} \delta_{Adi}(adprimarycategory_j) + h_{Adi,Adj} \delta_{Adi}(creative_{type}_j) \\ & + m_{Adi,Adj} \delta_{Adj}(pubprimarycateg)] + ptf(pub). \end{aligned}$$

No qual:

$e_{Adi, Adj}$ é um peso dado à diferença das áreas entre os banners; quanto maior esta for, menos este fator agrega à predição dado que os anúncios são considerados menos semelhantes.

$\delta_{Adi}(\text{atributo}_j)$ é uma função tal que:

$$\begin{cases} \delta_{Adi}(\text{atributo}_j) = 1, \text{ se } \text{atributo}_{Adi} = \text{atributo}_j. \\ \delta_{Adi}(\text{atributo}_j) = 0, \text{ caso contrário.} \end{cases}$$

$m_{Adi, Adj}$ é um parâmetro que contabiliza a relação entre o publicador e o anúncio j publicado pelo mesmo através da similaridade entre suas categorias.

$|R(pub)|^{-\alpha}$ exprime a quantidade de anúncios impressos pelo publicador pub . Serve para normalizar a somatória que aparece em seguida na equação, evitando que publicadores que imprimiram muitos anúncios distorçam a convergência.

ptf (publicador_type_factor) é um parâmetro semelhante ao enviesamento, porém determina apenas uma possível influência que um dado publicador pode causar na taxa de conversão de um anúncio por ser de determinado tipo (blog, site, mobile ou social). Os parâmetros podem ser obtidos por redução de custo via gradiente estocástico.

É possível modelar esta equação valendo-se de vetores de dimensão f que rastreiam automaticamente variáveis latentes de anúncios e publicadores. Matematicamente (o parâmetro q_{Adi} já foi colocado em evidência):

$$\begin{aligned} \hat{y}_{Pub,Adi} = & b_{Pub,Adi} + p_{Pub} * (q_{Adi} + |R(pub)|^{-\alpha}) * \left(\sum_{j \in R(Pub)} [(y_{Pub,Adj} - b_{Pub,Adj}) * \beta_{Adi,Adj} \right. \\ & + \gamma_{Adi,Adj} + \frac{\varepsilon_{Adi,Adj}}{1 + |Area_{Adi} - Area_{Adj}|} + \zeta_{Adi,Adj} \delta_{Adi}(visibility_j) \\ & + \eta_{Adi,Adj} \delta_{Adi}(adprimarycategory_j) + \vartheta_{Adi,Adj} \delta_{Adi}(creative_{type_j}) \\ & \left. + \kappa_{Adi,Adj} \delta_{Adi}(pubprimarycateg)] \right) + ptf(pub). \end{aligned}$$

Faltando portanto a parte dinâmica para que o modelo seja utilizável ao cenário do presente projeto.

Apesar de parecer complexo, a parte dinâmica pode ser considerada apenas no parâmetro de enviesamento b^* , resultando na expressão integrada com os resultados previamente obtidos:

$$\begin{aligned}
\hat{y}_{Pub,Adi} = & \mu + b_{Adi} + b_{Adi}(t) + b_{Pub} + b_{Pub}(t) + p_{pub} \\
& * \left(q_{Adi} \right. \\
& + |R(pub)^{-\alpha}| \left(\sum_{j \in R(Pub)} [(y_{Pub,Adj} - b_{Pub,Adj}) * \beta_{Adi,Adj} + \gamma_{Adi,Adj} \right. \\
& + \frac{\varepsilon_{Adi,Adj}}{1 + |Area_{Adi} - Area_{Adj}|} + \zeta_{Adi,Adj} \delta_{Adi}(visibility_j) \\
& + \eta_{Adi,Adj} \delta_{Adi}(adprimarycategory_j) + \vartheta_{Adi,Adj} \delta_{Adi}(creative_{type_j}) \\
& \left. \left. + \kappa_{Adi,Adj} \delta_{Adj}(pubprimarycateg) \right] \right) + ptf(pub).
\end{aligned}$$

A obtenção dos parâmetros pode ser feita através da redução da função custo, por gradiente estocástico, definida a seguir:

$$\begin{aligned}
J = & \sum_{(Ad, Pub, Date \in K)} \left(\left(\mu + b_{Adi} + b_{Adi}(t) + b_{Pub} + b_{Pub}(t) + p_{pub} * (q_{Adi} \right. \right. \\
& + |R(pub)^{-\alpha}| \sum_{j \in R(Pub)} [(y_{Pub,Adj} - b_{Pub,Adj}) * \beta_{Adi,Adj} + \gamma_{Adi,Adj} \\
& + \frac{\varepsilon_{Adi,Adj}}{1 + |Area_{Adi} - Area_{Adj}|} + \zeta_{Adi,Adj} \delta_{Adi}(visibility_j) \\
& + \eta_{Adi,Adj} \delta_{Adi}(adprimarycategory_j) + \vartheta_{Adi,Adj} \delta_{Adi}(creative_{type_j}) \\
& \left. \left. + \kappa_{Adi,Adj} \delta_{Adj}(pubprimarycateg) \right] \right) + ptf(pub) \Big)^2 - y_{Ad, Pub, Date}^2 \\
& + \lambda \left(\sum ((b_{Adi})^2 + b_{Adi}(t)^2 + (b_{Pub})^2 + (b_{Pub}(t))^2 + |q_{Adi}|^2 + |\beta_{Adi,Adj}|^2 \right. \\
& + |\gamma_{Adi,Adj}|^2 + |\varepsilon_{Adi,Adj}|^2 + |\zeta_{Adi,Adj}|^2 + |\eta_{Adi,Adj}|^2 + |\vartheta_{Adi,Adj}|^2 \\
& \left. + |\kappa_{Adi,Adj}|^2 + ptf(pub)^2 \right).
\end{aligned}$$

Doravante o termo:

$$\begin{aligned}
& \left(\left(\mu + b_{Adi} + b_{Adi}(t) + b_{Pub} + b_{Pub}(t) + p_{pub} * (q_{Adi} + \right. \right. \\
& |R(pub)^{-\alpha} | \sum_{j \in R(Pub)} [(y_{Pub,Adj} - b_{Pub,Adj}) * \beta_{Adi,Adj} + \gamma_{Adi,Adj} + \frac{\varepsilon_{Adi,Adj}}{1+|Area_{Adi}-Area_{Adj}|} + \\
& \zeta_{Adi,Adj} \delta_{Adi}(visibility_j) + \eta_{Adi,Adj} \delta_{Adi}(adprimarycategory_j) + \\
& \vartheta_{Adi,Adj} \delta_{Adi}(creative_{type_j}) + \kappa_{Adi,Adj} \delta_{Adj}(pubprimarycateg)] + ptf(pub) \Big) - \\
& y_{Ad, Pub, Date})^2 + \lambda (\sum ((b_{Adi})^2 + b_{Adi}(t)^2 + (b_{Pub})^2 + (b_{Pub}(t))^2 + (b_{Pub}(t))^2 + |q_{Adi}|^2 + \\
& |\beta_{Adi,Adj}|^2 + |\gamma_{Adi,Adj}|^2 + |\varepsilon_{Adi,Adj}|^2 + |\zeta_{Adi,Adj}|^2 + |\eta_{Adi,Adj}|^2 + |\vartheta_{Adi,Adj}|^2 + |\kappa_{Adi,Adj}|^2 + \\
& ptf(pub)^2)).
\end{aligned}$$

será chamado de $cost_{Ad, Pub}$, fornecendo a seguinte equação custo para o grandiente estocástico (desconsiderando-se os parâmetros de regulação):

$$J = \sum_1^m \frac{cost(parametros, y)^2}{2}.$$

E a atualização de um dado parâmetro é dada por:

$$\theta := \theta - \xi \frac{\partial J}{\partial \theta}.$$

Resultando nas atualizações (já acrescentando os parâmetros de regularização λ):

$$b_{Adi} := b_{Adi}(1 - \xi\lambda) - \xi cost.$$

$$b_{Adi}(t) := b_{Adi}(t)(1 - \xi\lambda) - \xi cost.$$

$$b_{Pub} := b_{Pub}(1 - \xi\lambda) - \xi cost.$$

$$b_{Pub}(t) := b_{Pub}(t)(1 - \xi\lambda) - \xi cost.$$

$$ptf(pub) := ptf(pub)(1 - \xi\lambda) - \xi cost.$$

$$\begin{aligned}
p_{Pub} := & P_{Pub}(1 - \xi\lambda) - \xi cost \\
& * (q_{Adi} + |R(pub)^{-\alpha}| \sum_{j \in R(Pub)} [(y_{Pub,Adj} - b_{Pub,Adj}) * \beta_{Adi,Adj} + \gamma_{Adi,Adj} \\
& + \frac{\varepsilon_{Adi,Adj}}{1 + |Area_{Adi} - Area_{Adj}|} + \zeta_{Adi,Adj} \delta_{Adi}(visibility_j) \\
& + \eta_{Adi,Adj} \delta_{Adi}(adprimarycategory_j) + \vartheta_{Adi,Adj} \delta_{Adi}(creative_{type_j}) \\
& + \kappa_{Adi,Adj} \delta_{Adj}(pubprimarycateg)]_p.
\end{aligned}$$

$$\forall j \in R(pub):$$

$$\beta_{Adi,Adj_p} := \beta_{Adi,Adj_p}(1 - \xi\lambda) - \xi cost * |R(pub)^{-\alpha}| * q_{Adi_p} * (y_{Pub,Adj} - b_{Pub,Adj}).$$

$$\gamma_{Adi,Adj_p} := \gamma_{Adi,Adj_p}(1 - \xi\lambda) - \xi cost * |R(pub)^{-\alpha}| * q_{Adi_p}.$$

$$\varepsilon_{Adi,Adj_p} := \varepsilon_{Adi,Adj_p}(1 - \xi\lambda) - \xi cost * |R(pub)^{-\alpha}| * q_{Adi_p} * (1 + |Area_{Adi} - Area_{Adj}|)^{-1}.$$

$$\zeta_{Adi,Adj_p} := \zeta_{Adi,Adj_p}(1 - \xi\lambda) - \xi cost * |R(pub)^{-\alpha}| * q_{Adi_p} * \delta_{Adi}(visibility_j).$$

$$\eta_{Adi,Adj_p} := \eta_{Adi,Adj_p}(1 - \xi\lambda) - \xi cost * |R(pub)^{-\alpha}| * q_{Adi_p} * \delta_{Adi}(adprimarycategory_j).$$

$$\vartheta_{Adi,Adj_p} := \vartheta_{Adi,Adj_p}(1 - \xi\lambda) - \xi cost * |R(pub)^{-\alpha}| * q_{Adi_p} * \delta_{Adi}(creative_{type_j}).$$

$$\kappa_{Adi,Adj_p} := \kappa_{Adi,Adj_p}(1 - \xi\lambda) - \xi cost * |R(pub)^{-\alpha}| * q_{Adi_p} * \delta_{Adj}(pubprimarycateg).$$

Onde o sub-índice “p” indica a p-ésima posição do vetor.

Vale mencionar que após estudos de diversos papers da área foi observada a importância de se considerar o advertiser na análise. O mesmo será feito através de um fator de enviesamento semelhante ao termo b_{adi} . A dedução das fórmulas para utilização do gradiente estocástico não serão desenvolvidas, porém o equacionamento final fica:

$$\begin{aligned}
\hat{y}_{Pub,Adi} = & \mu + b_{Adi} + b_{Adi}(t) + b_{AdAdvertiser} + b_{Pub} + b_{Pub}(t) + p_{Pub} (+q_{Adi} * \\
& |R(pub)^{-\alpha}| (\sum_{j \in R(Pub)} [(y_{Pub,Adj} - b_{Pub,Adj}) * \beta_{Adi,Adj} + \gamma_{Adi,Adj} + \frac{\varepsilon_{Adi,Adj}}{1 + |Area_{Adi} - Area_{Adj}|} +
\end{aligned}$$

$$\zeta_{Adi,Adj}\delta_{Adi}(visibility_j) + \eta_{Adi,Adj}\delta_{Adi}(adprimarycategory_j) + \vartheta_{Adi,Adj}\delta_{Adi}(creative_{type_j}) + \kappa_{Adi,Adj}\delta_{Adi}(pubprimarycateg)])) + ptf(pub) .$$

Resultados Obtidos

A tabela a seguir ilustra os resultados obtidos em diversas bases recebidas do Lomadee. A diferença entre cada base esta em função de filtros aplicados pela própria empresa e da qual não há conhecimento preciso dos critérios de filtragem, porém sabe-se que para efeitos de comparação todas as bases são válidas. Nos resultados é possível se observar que o parâmetro utilizado para comparação de desempenho foi o “F2-Score”, dado pela expressão:

$$F_2Score = \frac{F_1Score}{MAE}.$$

A proposta é que os algoritmos busquem maximizar o parâmetro F2, pois isto é equivalente a maximizar o parâmetro F1 Score e minizar ao mesmo tempo o erro médio. Isto se deve pois a base de dados Lomadee é fortemente desbalanceada, ou seja, é comum banners apresentarem 0 clicks, então ao se maximizar o F2 Score restringe-se as abordagens a serem escolhidas por aquelas que melhor descobrem onde de fato houve click. Os resultados foram:

Experimento	Base	Estratégia	MAE	F2-Score
1	1	Baseline (No Rounding)	6.75	-
2	1	Baseline (Rounded)	2.00	-
3	2	Baseline (No Rounding)	0.70	0.07
4	2	SVD	0.61	0.08
5	2	SVD++ (20 Iterações)	0.66	-
6	2	SVD++ (30 Iterações)	0.58	0.08
7	2	timeSVD (Vetores(t) randômicos)	0.63	0.10
8	2	timeSVD (Vetores(t) nulos)	0.58	0.12

9	3	spartaVD (Erro em Clicks)	0.89	0.35
10	3	spartaVD(Erro em CTR)	0.10	1.30
11	4	spartaVD	0.02	0.35
12	5	spartaVD	0.04	1.24
13	6	spartaVD	0.03	1.75
14	7	Baseline($\text{eps} = 1$, $\text{iter} = 30$)	0.123	0.06
15	7	SVD($\text{eps} = 1$, $\text{iter} = 30$)	0.123	0.06
16	7	Baseline($\text{eps} = 0.05$, $\text{iter} = 50$)	0.015	0.60
17	7	SVD($\text{eps} = 0.05$, $\text{iter} = 50$)	0.015	0.58
18	7	Baseline($\text{eps} = 0.05$, $\text{iter} = 50$, $\lambda \in [0,20]$)	0.013	0.69
19	7	SVD($\text{eps} = 0.05$, $\text{iter} = 50$, $\lambda \in [0,20]$)	0.013	0.82
20	7	Baseline($\text{eps} = 0.05$, $\text{iter} = 50$, $\lambda \in [0,20]$)	0.013	0.52
21	7	SVD($\text{eps} = 0.05$, $\text{iter} = 50$, $\lambda \in [0,20]$)	0.013	0.54
22	7	SVD++($\text{eps} = 0.05$, $\text{iter} = 50$, $\lambda \in [0,20]$)	0.013	0.54
23	7	timeSVD($\text{eps} = 0.05$, $\text{iter} = 50$, $\lambda \in [0,20]$)	0.013	0.57
24	7	timeSVD($\text{eps} = 0.05$, $\text{iter} = 50$, $\lambda_{1,2} \in [0,20]$)	0.013	0.57
25	7	Neighborhood($\text{eps} = 0.2$, $\text{iter} = 30$, $\lambda \in [0,20]$)	0.029	0.33
26	7	Neighborhood($\text{eps} = 0.05$, $\text{iter} = 30$, $\lambda \in [0,0.5]$)	0.015	0.43

Tabela 10: Resultados obtidos com estratégias aplicadas nas Etapas 1 e 2.

O que se observa de imediato é que o erro médio é relativamente pequeno. Porém isto já acontece caso o algoritmo prevesse apenas “zeros” em todos os pontos, sendo o parâmetro F2 Score mais interessante para comparação. Os resultados estão de acordo com o que era esperado, indicando que as técnicas de filtragem colaborativa empregadas na competição

Netflix também rastreiam tendências em anúncios impressos em sites. Entre parentesis estão indicados as condições as quais foram feitas os testes. Iter se refere ao número de iterações, lambda indica o intervalo de valores utilizados na etapa de regulação, eps é o parâmetro utilizado na inicialização dos vetores aleatórios.

Conforme aumentou-se a complexidade do algoritmo implementado, houve uma relativa melhora no fator F2 Score. Há, porém, um fato importante de ser notado neste ponto: ainda sim o algoritmo esta prevendo em muitos pontos apenas “zero”, portanto não é interessante para serem utilizados como discriminadores de qual banner deve ser selecionado para ser futuramente impresso.