ESTUDO COMPARATIVO DE ALGORITMOS DE RECOMENDAÇÃO

Marcos Masanobu Takahashi
Orientador: Roberto Hirata Jr.

ESTUDO COMPARATIVO DE ALGORITMOS DE RECOMENDAÇÃO

Marcos Masanobu Takahashi
Orientador: Roberto Hirata Jr.

CONTEXTUALIZAÇÃO: E-COMMERCE

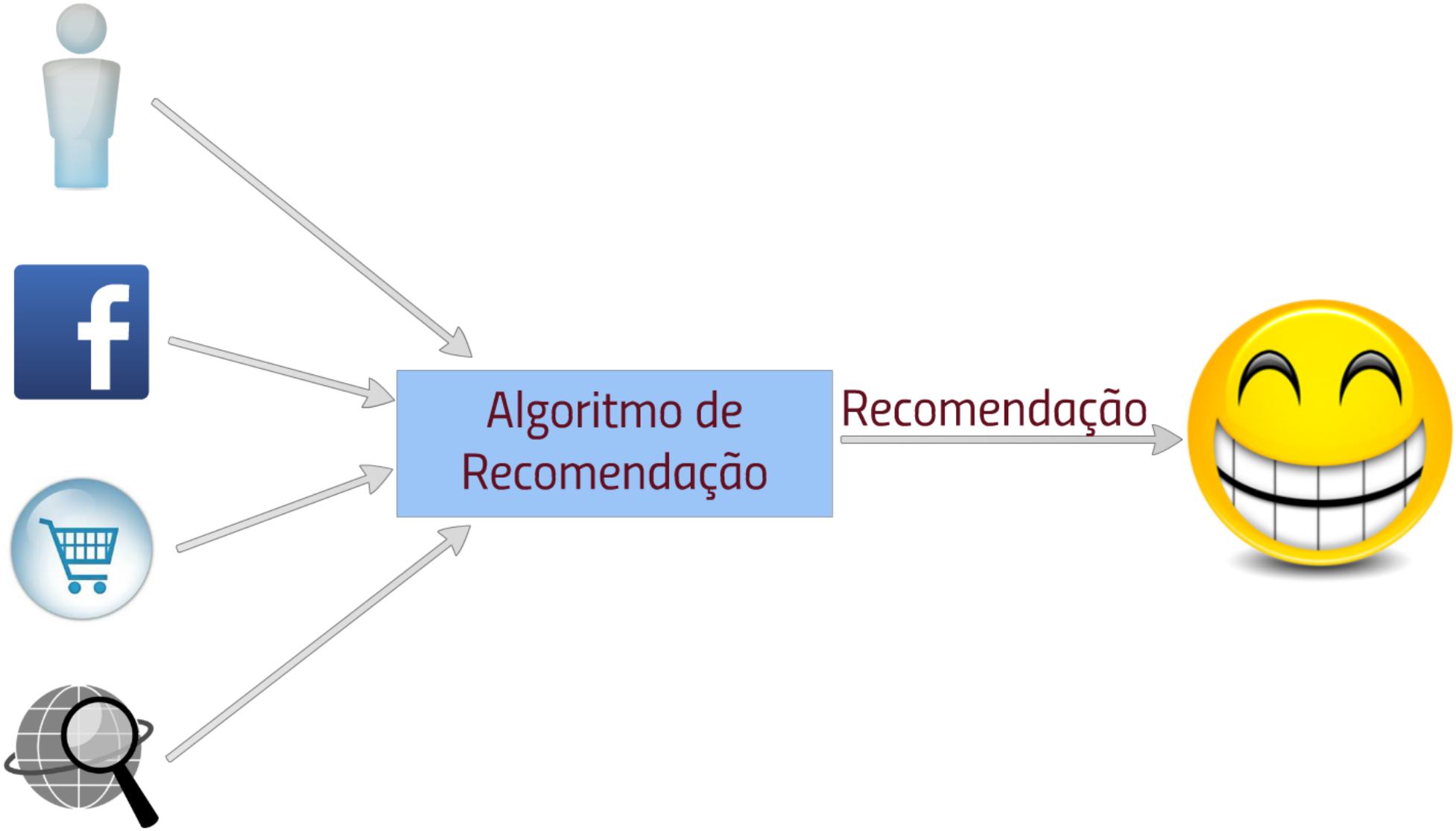
- Forte concorrência (muitos competidores)
- Baixa quantidade de visualização de páginas por visita (média de 4 a 5)
- Baixo tempo de navegação no site (3 a 4 minutos)

Comparativo dos maiores players

	Pageviews / Visita	Duração da Visita
Submarino	4,73	4:32
extra.com.br	5,36	4:42
magazineluiza vem ser feliz	3,79	3:57
americanas.com	4,61	4:30
Walmart	3,37	3:19
NETSHOES SEM LIMITES ENTRE VOCÊ E O ESPORTE	2,96	3:25

Fonte: Alexa

Recomendação



ESTUDO COMPARATIVO DE ALGORITMOS DE RECOMENDAÇÃO

Marcos Masanobu Takahashi
Orientador: Roberto Hirata Jr.

OBJETIVOS

- Implementação dos Algoritmos:
  - Large-scale Parallel Collaborative Filtering;
  - Fast Context-aware Recommendations with Factorization Machines;

- Compará-los em um contexto restrito do mundo real (e-commerce de Móveis e Utilidades Domésticas)

- Comparar o retorno financeiro de cada algoritmo.

SISTEMAS DE RECOMENDAÇÃO

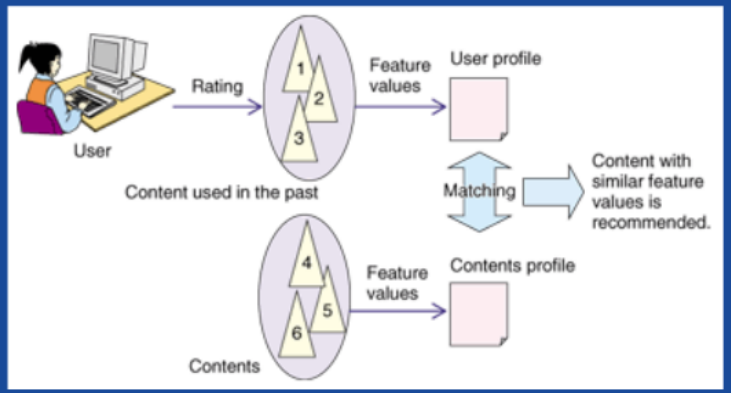
- Combinação de várias técnicas computacionais para selecionar itens personalizados com base nos interesses dos usuários e conforme o contexto no qual estão inseridos. (Wikipedia)

- São comumente divididos em 3 tipos de algoritmos:
  - Content-based Filtering
  - Collaborative Filtering
  - Hybrid Recommender Systems

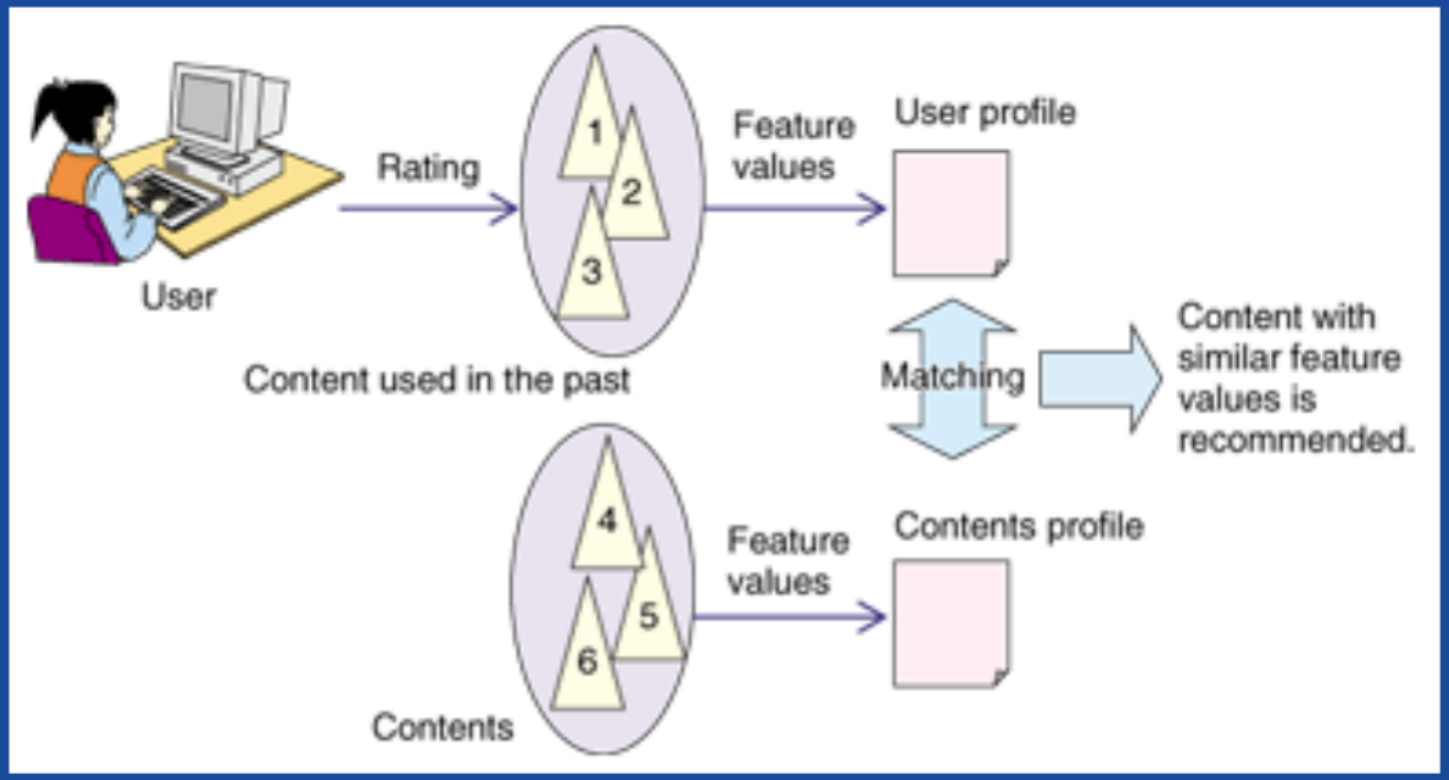
- Existe uma nova classe de sistemas de recomendação chamada Context-aware Recommender Systems

Content-based Filtering

- Recomenda itens semelhantes aos já comprados / interagidos pelo usuário;
- "Aprende" o perfil do usuário através dos itens anteriores;
- Procura novos itens com "match" no perfil do usuário;
- Recomenda os itens com a melhor taxa de "match".

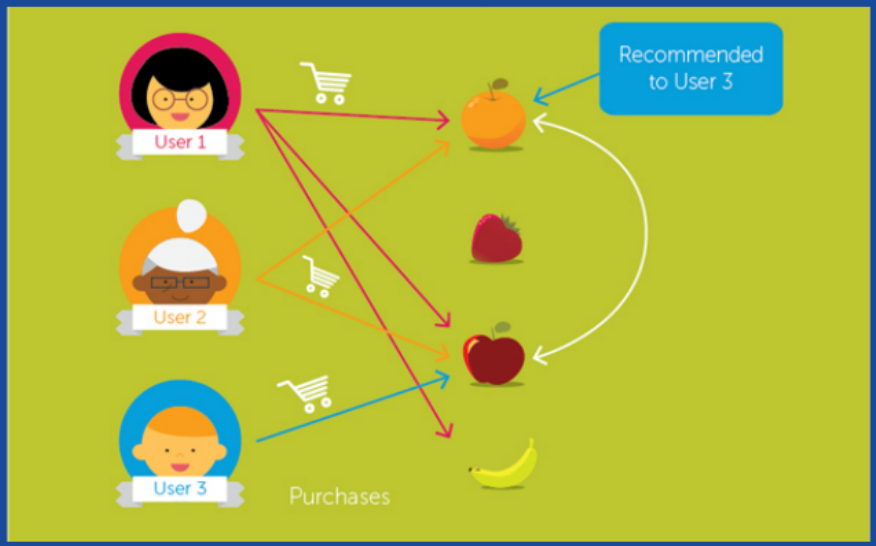


- Recomenda itens semelhantes aos já comprados / interagidos pelo usuário;
- "Aprende" o perfil do usuário através dos itens anteriores;
- Procura novos itens com "match" no perfil do usuário;
- Recomenda os itens com a melhor taxa de "match".

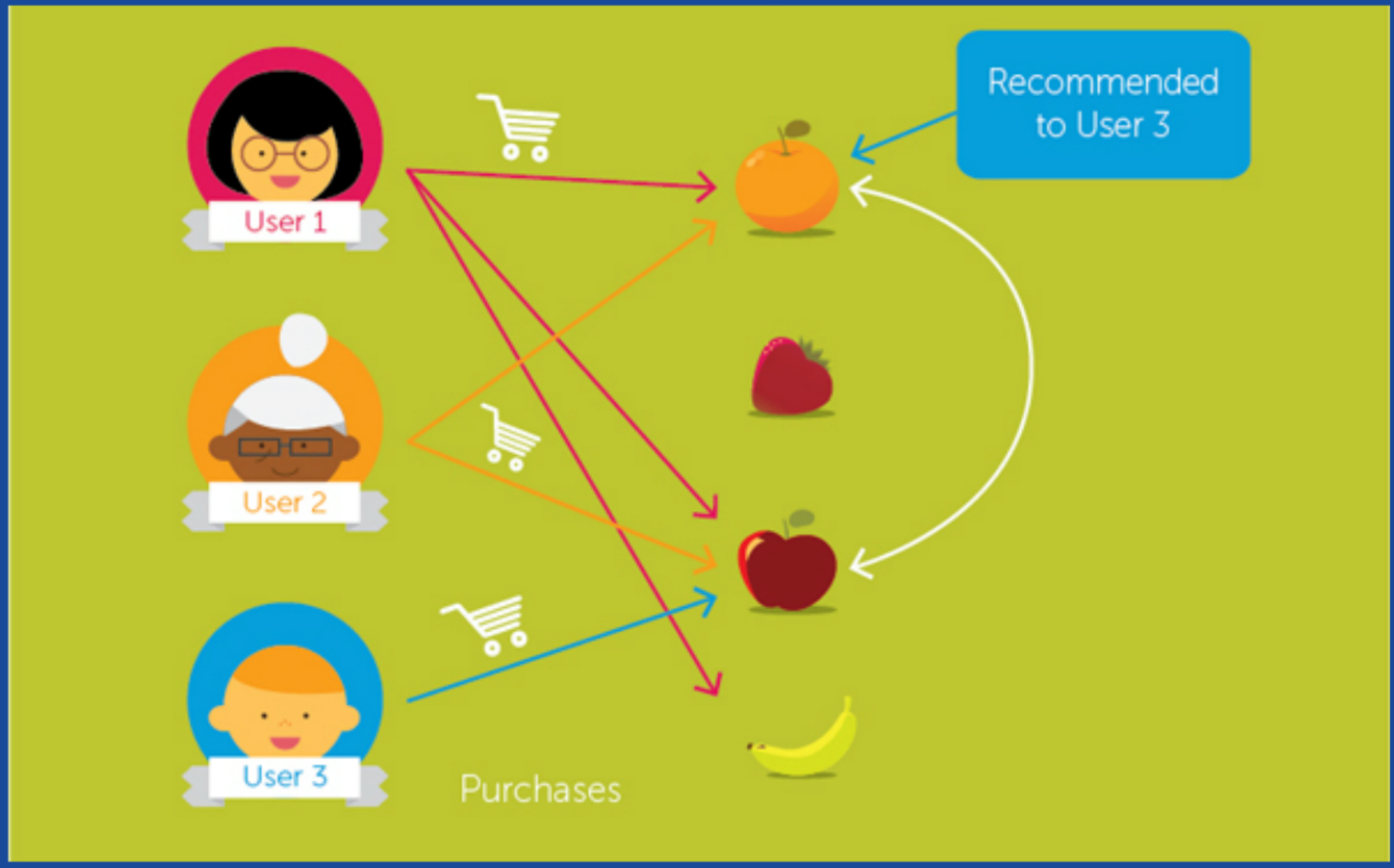

User
Rating
1
2
3
Content used in the past
Feature values
User profile
Matching
Content with similar feature values is recommended.
4
5
6
Contents
Feature values
Contents profile

Collaborative Filtering

- "Quem comprou X também comprou Y"
- Recomenda itens que usuários semelhantes já compraram / interagiram;
- Através dos produtos já comprados / interagidos, procura usuários semelhantes;
- Seleciona o produto com maior rating e que o usuário ainda não interagiu.

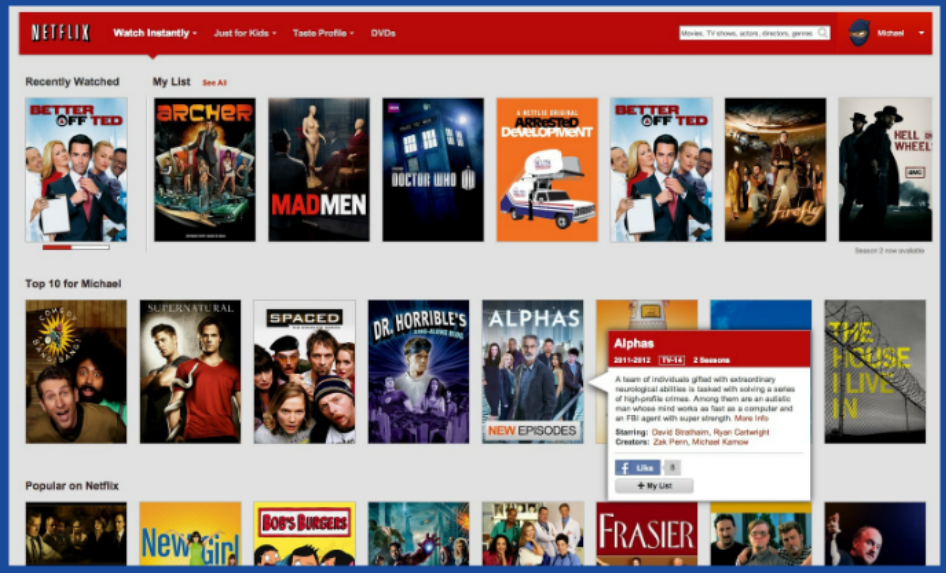


- "Quem comprou X também comprou Y"
- Recomenda itens que usuários semelhantes já compraram / interagiram;
- Através dos produtos já comprados / interagidos, procura usuários semelhantes;
- Seleciona o produto com maior rating e que o usuário ainda não interagiu.



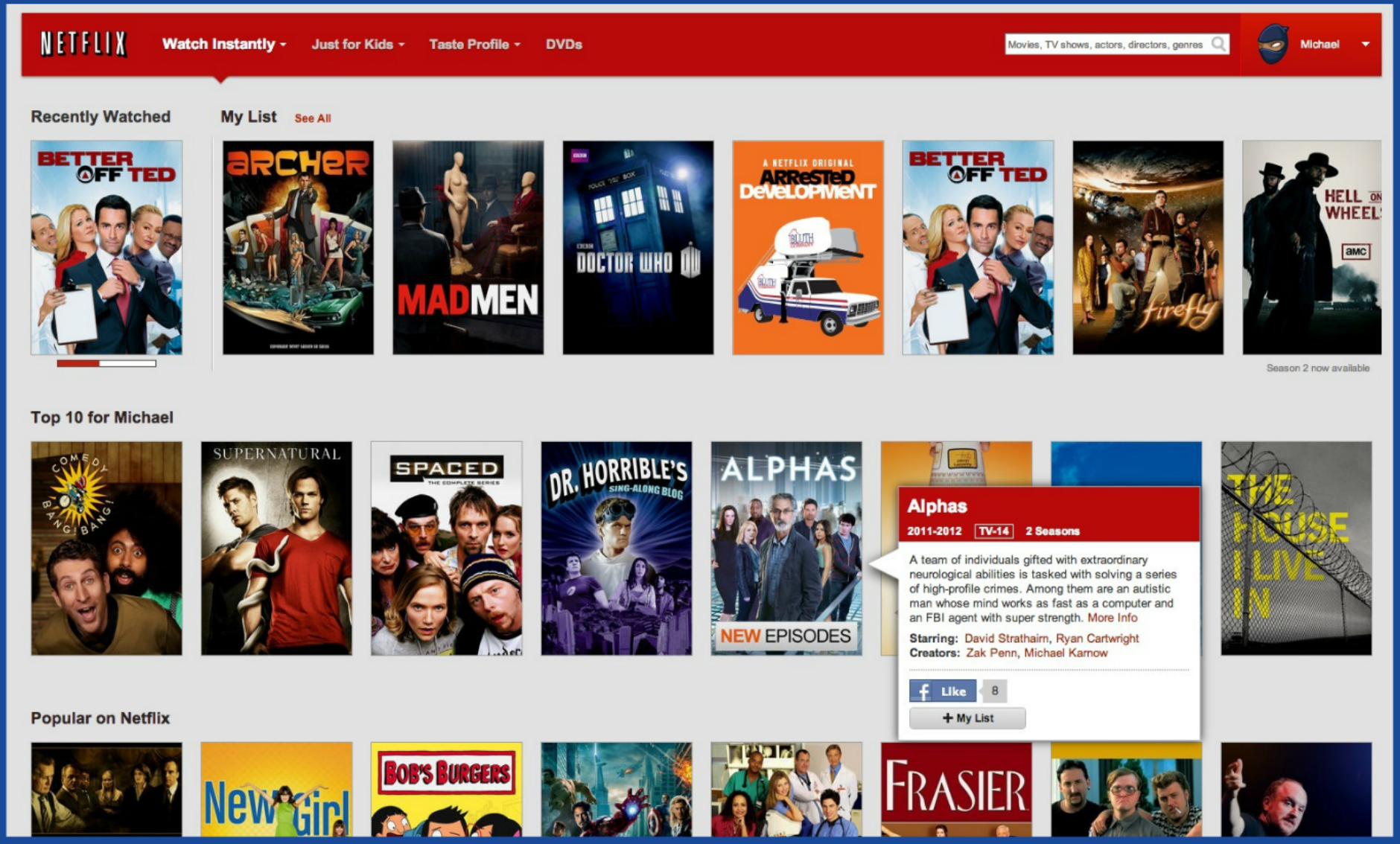
Hybrind Recommender Systems

- Combina as abordagens de Content-based Filtering e Collaborative Filtering;
- Diversas formas de implementação:
  - aplicação dos dois separados e juntar depois;
  - adicionando capacidade de content-based a Collaborative Filtering (ou vice-versa);
  - unificação das duas abordagens em um único modelo.



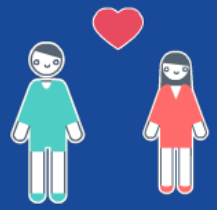
- Combina as abordagens de Content-based Filtering e Collaborative Filtering;

- Diversas formas de implementação:
  - aplicação dos dois separados e juntar depois;
  - adicionando capacidade de content-based a Collaborative Filtering (ou vice-versa);
  - unificação das duas abordagens em um único modelo.


NETFLIX
Watch Instantly
Just for Kids
Taste Profile
DVDs
Movies, TV shows, actors, directors, genres
Michael
Recently Watched
My List
See All
Season 2 now available
Top 10 for Michael
Alphas
2011-2012
TV-14
2 Seasons
A team of individuals gifted with extraordinary neurological abilities is tasked with solving a series of high-profile crimes. Among them are an autistic man whose mind works as fast as a computer and an FBI agent with super strength. More Info
Starring: David Strathairn, Ryan Cartwright
Creators: Zak Penn, Michael Karnow
Like
8
+ My List
Popular on Netflix

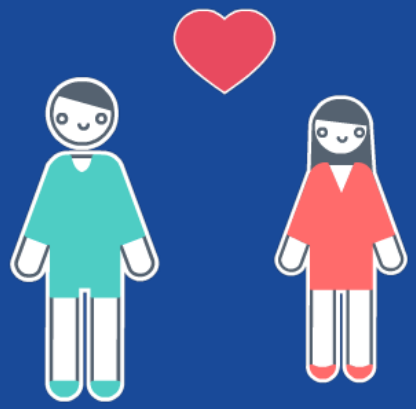
Context-aware Recommender Systems

- Técnica muito recente e em desenvolvimento;
- Gera recomendações muito mais personalizadas e abrangentes;
- Abrange além de dados de *User* e *Item*, dados do *Contexto* que levaram ao acontecimendo de dado evento.



- Técnica muito recente e em desenvolvimento;
- Gera recomendações muito mais personalizadas e abrangentes;
- Abrange além de dados de *User* e *Item*, dados do *Contexto* que levaram ao acontecimendo de dado evento.


"THE MOST AWESOME ACTION CAST EVER ASSEMBLED..."
SYLVESTER STALLONE
JASON STATHAM
JET LI
MICKEY ROURKE
DOLPH LUNDGREN
BRUCE WILLIS
AND
ARNOLD SCHWARZENEGGER
EMPIRE
STALLONE STATHAM LI
THE EXPENDABLES
IN CINEMAS THURS AUGUST 19, ADVANCE SCREENINGS AUGUST 13, 14 & 18




THE FAULT IN OUR STARS
MUSIC FROM THE MOTION PICTURE

LARGE-SCALE PARALLEL COLLABORATIVE FILTERING

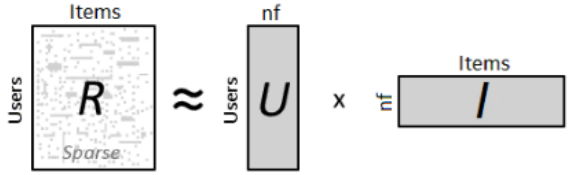
- Foi desenvolvido para a competição Netflix Prize e melhorou o desempenho do algoritmo que era utilizado pela Netflix (Cine-Match) em 5.91%;
- Utiliza um método simples e escalável para grandes volumes de dados;
- É paralelizável;
- Usa somente dados de User e Item.

Abordagem tradicional de algoritmos de recomendação:

f: User x Item -> Rating

Estimar os ratings que não foram dados através da função *f*.

Este algoritmo decompõe a matriz *R* de ratings em *U* e *I*, tal que:



nf é o número de variáveis consideradas no modelo

Passo a passo do algoritmo:

1- Inicializar a matriz *I*, atribuindo o rating médio do item na primeira coluna e valores aleatórios próximos de 0 nas outras colunas;

2- Fixa *I* e resolve *U*, minimizando a função objetivo;

3- Fixa *U* e resolve *I*, minimizando a função objetivo;

4- Repetir 2 e 3 até atingir o critério de parada.

O critério de parada utilizado é de Raíz Quadrada do Erro Quadrático Médio.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(r_i - \hat{r}_i)^2}$$

A decomposição em *U* e *I* se dá resolvendo a seguinte função:

$$f(U,I) = \sum_{(i,j)\in I}(r_{ij} - u_i^T i_j)^2 + \lambda\Big(\sum_i n_{u_i}\|u_i\|^2 + \sum_j n_{i_j}\|i_j\|^2\Big)$$

E para resolver *U* dado *I*, tem-se:

$$u_i = A_i^{-1}V_i, \quad \forall i$$

onde: $A_i = I_{N_i} I_{N_i}^T + \lambda n_{u_i} E$,
$V_i = I_{N_i} R^T(i, N_i)$,
E é a matriz identidade de $n_f \times n_f$,
I_{N_i} denota a submatriz de I onde as colunas $j \in N_i$ são selecionadas
$R(i, I_i)$ é a matriz onde as colunas $j \in I_i$ da linha i é selecionada.
N_i são os usuários com rating no item i
λ é o coeficiente de regularização, setado manualmente.

A resolução de *I* dado *U* é análoga, alternando-se User para Item

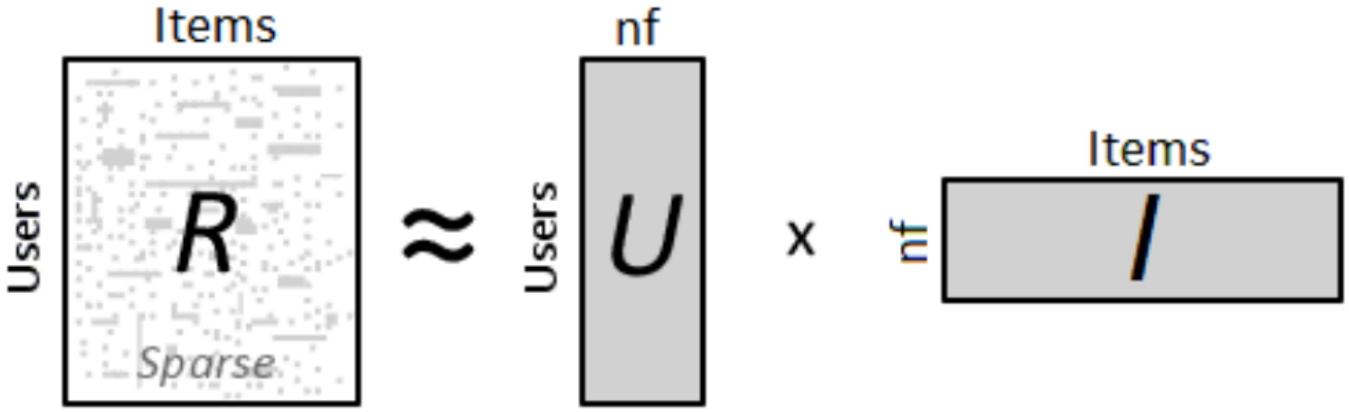
- Foi desenvolvido para a competição Netflix Prize e melhorou o desempenho do algoritmo que era utilizado pela Netflix (Cine-Match) em 5.91%;
- Utiliza um método simples e escalável para grandes volumes de dados;
- É paralelizável;
- Usa somente dados de User e Item.

Abordagem tradicional de algoritmos de recomendação:

f: User x Item -> Rating

Estimar os ratings que não foram dados através da função *f*.

Este algoritmo decompõe a matriz *R* de ratings em *U* e *I*, tal que:



nf é o número de variáveis consideradas no modelo

Passo a passo do algoritmo:

1- Inicializar a matriz *I*, atribuindo o rating médio do item na primeira coluna e valores aleatórios próximos de 0 nas outras colunas;

2- Fixa *I* e resolve *U*, minimizando a função objetivo;

3- Fixa *U* e resolve *I*, minimizando a função objetivo;

4- Repetir 2 e 3 até atingir o critério de parada.

O critério de parada utilizado é de Raíz Quadrada do Erro Quadrático Médio.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(r_i - \hat{r}_i)^2}$$

A decomposição em *U* e *I* se dá resolvendo a seguinte função:

$$f(U,I) = \sum_{(i,j)\in I} (r_{ij} - u_i^T i_j)^2 + \lambda\left(\sum_i n_{u_i}\|u_i\|^2 + \sum_j n_{i_j}\|i_j\|^2\right)$$

E para resolver *U* dado *I*, tem-se:

$$u_i = A_i^{-1} V_i, \quad \forall i$$

onde: $A_i = I_{N_i} I_{N_i}^T + \lambda n_{u_i} E$,
$V_i = I_{N_i} R^T(i, N_i)$,
E é a matriz identidade de $n_f \times n_f$,
I_{N_i} denota a submatriz de I onde as colunas $j \in N_i$ são selecionadas
$R(i, I_i)$ é a matriz onde as colunas $j \in I_i$ da linha i é selecionada.
N_i são os usuários com rating no item i
λ é o coeficiente de regularização, setado manualmente.

A resolução de *I* dado *U* é análoga, alternando-se User para Item

LARGE-SCALE PARALLEL COLLABORATIVE FILTERING

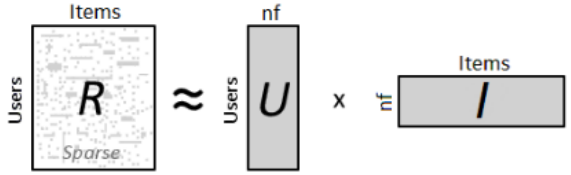
- Foi desenvolvido para a competição Netflix Prize e melhorou o desempenho do algoritmo que era utilizado pela Netflix (Cine-Match) em 5.91%;
- Utiliza um método simples e escalável para grandes volumes de dados;
- É paralelizável;
- Usa somente dados de User e Item.

Abordagem tradicional de algoritmos de recomendação:

f: User x Item -> Rating

Estimar os ratings que não foram dados através da função *f*.

Este algoritmo decompõe a matriz *R* de ratings em *U* e *I*, tal que:



nf é o número de variáveis consideradas no modelo

Passo a passo do algoritmo:

1- Inicializar a matriz *I*, atribuindo o rating médio do item na primeira coluna e valores aleatórios próximos de 0 nas outras colunas;

2- Fixa *I* e resolve *U*, minimizando a função objetivo;

3- Fixa *U* e resolve *I*, minimizando a função objetivo;

4- Repetir 2 e 3 até atingir o critério de parada.

O critério de parada utilizado é de Raíz Quadrada do Erro Quadrático Médio.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(r_i - \hat{r}_i)^2}$$

A decomposição em *U* e *I* se dá resolvendo a seguinte função:

$$f(U,I) = \sum_{(i,j)\in I}(r_{ij} - u_i^T i_j)^2 + \lambda\Big(\sum_i n_{u_i}\|u_i\|^2 + \sum_j n_{i_j}\|i_j\|^2\Big)$$

E para resolver *U* dado *I*, tem-se:

$$u_i = A_i^{-1}V_i, \quad \forall i$$

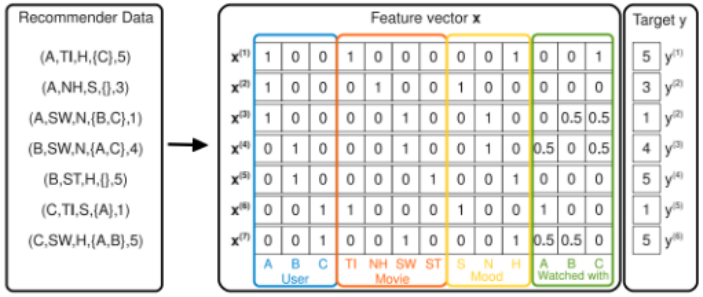
onde: $A_i = I_{N_i} I_{N_i}^T + \lambda n_{u_i} E$,
$V_i = I_{N_i} R^T(i, N_i)$,
E é a matriz identidade de $n_f \times n_f$,
I_{N_i} denota a submatriz de I onde as colunas $j \in N_i$ são selecionadas
$R(i, I_i)$ é a matriz onde as colunas $j \in I_i$ da linha i é selecionada.
N_i são os usuários com rating no item i
λ é o coeficiente de regularização, setado manualmente.

A resolução de *I* dado *U* é análoga, alternando-se User para Item

FAST CONTEXT-AWARE RECOMMENDATIONS WITH FACTORIZATION MACHINES

- Usa dados de User, Item e Context (praticamente tudo).
- Os algoritmos Context-aware costumam ser computacionalmente custosos, já que utilizam diversos tipos de dados;
- O algoritmo *Multiverse Recommendation*, que possuía um dos melhores desempenhos em recomendação tem complexidade $\mathrm{O}(\mathrm{k}^m)$
- Este algoritmo tem complexidade $\mathrm{O}(|S|\mathrm{mk})$
- Utiliza uma abordagem chamada Factorization Machine

A modelagem de dados deve ser da seguinte forma:



onde o input do algoritmo é o vetor x resultante.
Cada parâmetro a ser estimado é encontrado através da função:

$$\theta = -\frac{\sum_{(\mathbf{x},y)\in S}\left(g_{(\theta)}(\mathbf{x}) - y\right) h_{(\theta)}(\mathbf{x})}{\sum_{(\mathbf{x},y)\in S} h^2_{(\theta)}(\mathbf{x}) + \lambda_{(\theta)}}$$

Factorization Machines são modelos de classes genéricas que agregam e imitam diversos tipos de sistemas de recomendação.

Eles modelam todas interações entre pares de variáveis através de:

$$y(x) := w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n}\sum_{j=i+1}^{n} \hat{w}_{i,j} x_i x_j$$

onde:

$$\hat{w}_{i,j} :=< v_i, v_j >= \sum_{f=1}^{k} v_{i,f} \cdot v_{j,f}$$

é a interação entre os pares de variáveis e

$$w_0 \in \mathbb{R}, \quad w \in \mathbb{R}^n, \quad V \in \mathbb{R}^{n\times k}$$

são os parâmetros a serem estimados.

Passo a passo do algoritmo:

1- Inicializa *w0*, *w* e *V* com valores 0;

2- Computa matrizes auxiliares *e*, *q*;

3- Computa o valor de *w0*;

4- Computa os valores de *w*;

5- Computa os valores de *V*;

6- Repete 3, 4, 5 até atingir o critério de parada.

O critério de parada utilizado foram o número de máximo de iterações.

- Usa dados de User, Item e Context (praticamente tudo).
- Os algoritmos Context-aware costumam ser computacionalmente custosos, já que utilizam diversos tipos de dados;
- O algoritmo *Multiverse Recommendation*, que possuía um dos melhores desempenhos em recomendação tem complexidade $\mathrm{O}(\mathrm{k}^m)$
- Este algoritmo tem complexidade $\mathrm{O}(|\mathrm{S}|\mathrm{mk})$
- Utiliza uma abordagem chamada Factorization Machine

Factorization Machines são modelos de classes genéricas que agregam e imitam diversos tipos de sistemas de recomendação.

Eles modelam todas interações entre pares de variáveis através de:

$$y(x) := w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \hat{w}_{i,j} x_i x_j$$

onde:

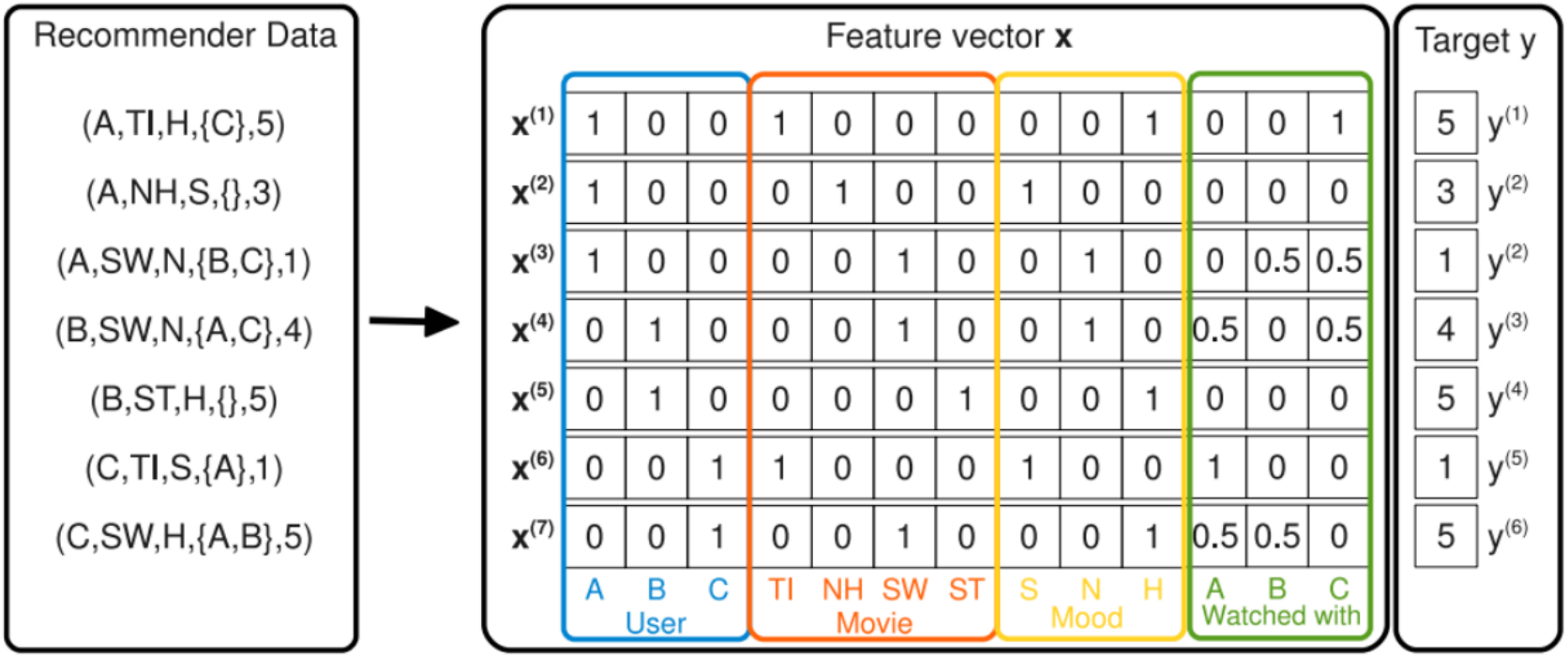
$$\hat{w}_{i,j} :=< v_i, v_j >= \sum_{f=1}^{k} v_{i,f} \cdot v_{j,f}$$

é a interação entre os pares de variáveis e

$$w_0 \in \mathbb{R}, \quad w \in \mathbb{R}^n, \quad V \in \mathbb{R}^{n \times k}$$

são os parâmetros a serem estimados.

A modelagem de dados deve ser da seguinte forma:



onde o input do algoritmo é o vetor x resultante.
Cada parâmetro a ser estimado é encontrado através da função:

$$\theta = -\frac{\sum_{(\mathbf{x},y)\in S} \left(g_{(\theta)}(\mathbf{x}) - y\right) h_{(\theta)}(\mathbf{x})}{\sum_{(\mathbf{x},y)\in S} h^2_{(\theta)}(\mathbf{x}) + \lambda_{(\theta)}}$$

Passo a passo do algoritmo:

1- Inicializa *w0*, *w* e *V* com valores 0;

2- Computa matrizes auxiliares *e*, *q;*

3- Computa o valor de *w0;*

4- Computa os valores de *w*;

5- Computa os valores de *V*;

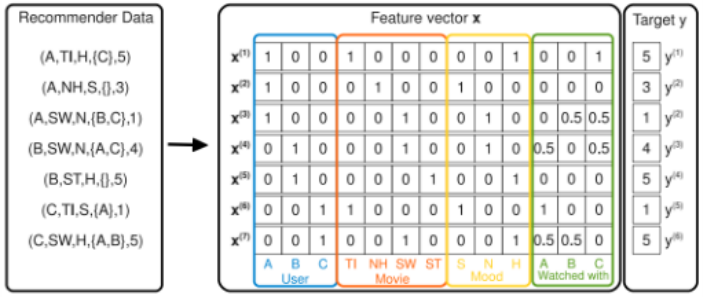
6- Repete 3, 4, 5 até atingir o critério de parada.

O critério de parada utilizado foram o número de máximo de iterações.

FAST CONTEXT-AWARE RECOMMENDATIONS WITH FACTORIZATION MACHINES

- Usa dados de User, Item e Context (praticamente tudo).
- Os algoritmos Context-aware costumam ser computacionalmente custosos, já que utilizam diversos tipos de dados;
- O algoritmo *Multiverse Recommendation*, que possuía um dos melhores desempenhos em recomendação tem complexidade $\mathrm{O}(\mathrm{k}^m)$
- Este algoritmo tem complexidade $\mathrm{O}(|S|\mathrm{mk})$
- Utiliza uma abordagem chamada Factorization Machine

A modelagem de dados deve ser da seguinte forma:



onde o input do algoritmo é o vetor x resultante.
Cada parâmetro a ser estimado é encontrado através da função:

$$\theta = -\frac{\sum_{(\mathbf{x},y)\in S}\left(g_{(\theta)}(\mathbf{x}) - y\right) h_{(\theta)}(\mathbf{x})}{\sum_{(\mathbf{x},y)\in S} h^2_{(\theta)}(\mathbf{x}) + \lambda_{(\theta)}}$$

Factorization Machines são modelos de classes genéricas que agregam e imitam diversos tipos de sistemas de recomendação.

Eles modelam todas interações entre pares de variáveis através de:

$$y(x) := w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n}\sum_{j=i+1}^{n} \hat{w}_{i,j} x_i x_j$$

onde:

$$\hat{w}_{i,j} :=< v_i, v_j >= \sum_{f=1}^{k} v_{i,f} \cdot v_{j,f}$$

é a interação entre os pares de variáveis e

$$w_0 \in \mathbb{R}, \quad w \in \mathbb{R}^n, \quad V \in \mathbb{R}^{n\times k}$$

são os parâmetros a serem estimados.

Passo a passo do algoritmo:

1- Inicializa *w0*, *w* e *V* com valores 0;

2- Computa matrizes auxiliares *e*, *q*;

3- Computa o valor de *w0*;

4- Computa os valores de *w*;

5- Computa os valores de *V*;

6- Repete 3, 4, 5 até atingir o critério de parada.

O critério de parada utilizado foram o número de máximo de iterações.

TESTES

- Todos os códigos e testes foram implementados em R e os dados utilizados são de uma empresa de comércio eletrônico do setor de Móveis e Utilidades Domésticas.

- Dados utilizados:
  - Collaborative Filtering: User (id_user), Item (id_item), Rating (#compras do item);

  - Context Recommendations: User (id_user, sexo, idade, estado(geográfico)), Item (id_item, categoria, cor), Contexto (proximidade da compra com datas especiais, uso de cupom de desconto, recompra, likes em tags de Facebook, agrupamento de dados de navegação com abandono de carrinho) e Rating (#compras do item).

- Collaborative Filtering:
  - base de 1 milhão de linhas;
  - aproximadamente 300000 Users x 50000 Item;

- Context Recommendations:
  - ordens - 1 milhão de linhas, intenção de compra - 200000 linhas, facebook - 30000 linhas
  - aproximadamente 1 milhão de linhas x 400000 variáveis

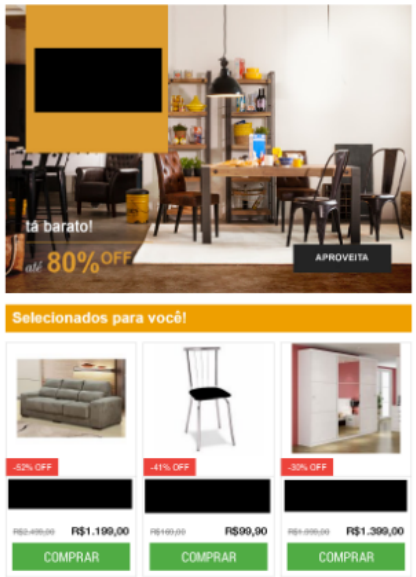
- Todos os testes rodaram em máquina com Ubuntu Server 64bits, 24 cores, 128GB RAM, 2 TB Disco

- Tempo de execução:
  - Collaborative Filtering - com nf = 100, 6h;
  - Context Recommendations - com nf = 5, 40h.

- Testes "reais":
  - Envio de campanhas de E-mail Marketing (dias 29, 30, 31 de outubro e 5 e 6 de novembro);

  - 22 grupos de aproximadamente 10000 pessoas, sendo:
    - 1 grupo selecionado pelo algoritmo Large-scale Parallel Collaborative Filtering;
    - 1 grupo selecionado pelo algoritmo Fast Context-aware Recommendations with Factorization Machines;
    - 20 grupos aleatórios sem aplicação de algoritmo.

  - Não houve sobreposição de pessoas dos mesmos algoritmos em disparos diferentes, ou seja, uma pessoa não recebeu mais de uma campanha de dado algoritmo.



- Todos os códigos e testes foram implementados em R e os dados utilizados são de uma empresa de comércio eletrônico do setor de Móveis e Utilidades Domésticas.

- Dados utilizados:
  - Collaborative Filtering: User (id_user), Item (id_item), Rating (#compras do item);

  - Context Recommendations: User (id_user, sexo, idade, estado(geográfico)), Item (id_item, categoria, cor), Contexto (proximidade da compra com datas especiais, uso de cupom de desconto, recompra, likes em tags de Facebook, agrupamento de dados de navegação com abandono de carrinho) e Rating (#compras do item).

- Collaborative Filtering:
  - base de 1 milhão de linhas;
  - aproximadamente 300000 Users x 50000 Item;

- Context Recommendations:
  - ordens - 1 milhão de linhas, intenção de compra - 200000 linhas, facebook - 30000 linhas
  - aproximadamente 1 milhão de linhas x 400000 variáveis

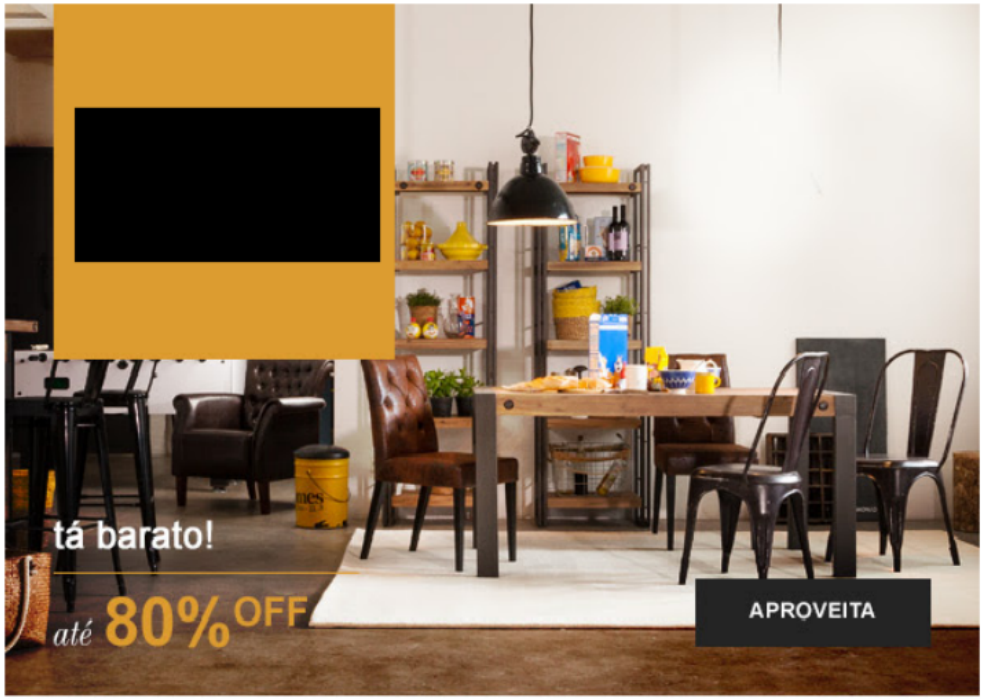
- Todos os testes rodaram em máquina com Ubuntu Server 64bits, 24 cores, 128GB RAM, 2 TB Disco

- Tempo de execução:
  - Collaborative Filtering - com nf = 100, 6h;
  - Context Recommendations - com nf = 5, 40h.

- Testes "reais":
  - Envio de campanhas de E-mail Marketing (dias 29, 30, 31 de outubro e 5 e 6 de novembro);

  - 22 grupos de aproximadamente 10000 pessoas, sendo:
    - 1 grupo selecionado pelo algoritmo Large-scale Parallel Collaborative Filtering;
    - 1 grupo selecionado pelo algoritmo Fast Context-aware Recommendations with Factorization Machines;
    - 20 grupos aleatórios sem aplicação de algoritmo.

  - Não houve sobreposição de pessoas dos mesmos algoritmos em disparos diferentes, ou seja, uma pessoa não recebeu mais de uma campanha de dado algoritmo.



Selecionados para você!

-52% OFF

[redacted]

~~R$2.499,00~~ **R$1.199,00**

COMPRAR

-41% OFF

[redacted]

~~R$169,00~~ **R$99,90**

COMPRAR

-30% OFF

[redacted]

~~R$1.999,00~~ **R$1.399,00**

COMPRAR

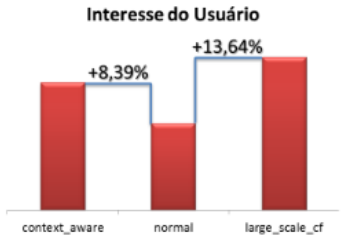
RESULTADOS

Métricas de E-mail Marketing

- Quantidade de e-mails enviados (sent)
- Quantidade de e-mails abertos (open)
- Quantidade de e-mails clicados (click)
- Quantidade de ordens geradas (orders)
- Receita gerada (revenue)

- open (%): open / sent
- CTR (%): click / sent
- click to open (%): click / open
- conversion (%): orders / click
- revenue / email: revenue / sent

base	CTR %	click to open %	revenue / email
normal	0.00%	0.00%	R$ 0.00
context_aware	8.39%	9.67%	R$ 0.01
large_scale_cf	13.65%	14.06%	R$ 0.04



data	base	CTR %	click to open %	revenue / email
29/10/2014	normal	0.00%	0.00%	R$ 0.00
29/10/2014	context_aware	1.94%	6.83%	-R$ 0.06
29/10/2014	large_scale_cf	15.92%	14.59%	R$ 0.04
30/10/2014	normal	0.00%	0.00%	R$ 0.00
30/10/2014	context_aware	5.68%	5.58%	-R$ 0.02
30/10/2014	large_scale_cf	8.54%	6.75%	-R$ 0.03
31/10/2014	normal	0.00%	0.00%	R$ 0.00
31/10/2014	context_aware	4.04%	11.91%	R$ 0.04
31/10/2014	large_scale_cf	6.65%	9.35%	R$ 0.02
05/11/2014	normal	0.00%	0.00%	R$ 0.00
05/11/2014	context_aware	24.51%	22.25%	R$ 0.08
05/11/2014	large_scale_cf	31.27%	25.17%	R$ 0.12
06/11/2014	normal	0.00%	0.00%	R$ 0.00
06/11/2014	context_aware	18.39%	14.28%	R$ 0.05
06/11/2014	large_scale_cf	16.85%	25.84%	R$ 0.09

Conclusão

- Recomendação melhorou o desempenho tanto em procura (clicks) quanto em receita;
- Collaborative Filtering teve um desempenho melhor;
- Context-aware precisa selecionar muito bem os contextos e também aumentar o número de features (nf);

Próximos passos

- Recomendação online;
- Incorporar dados de navegação e carrinho no Collaborative Filtering;
- Testar novos contextos e aumentar o número de features;
- Testar recomendação conjunta dos 2 algoritmos.

Métricas de E-mail Marketing

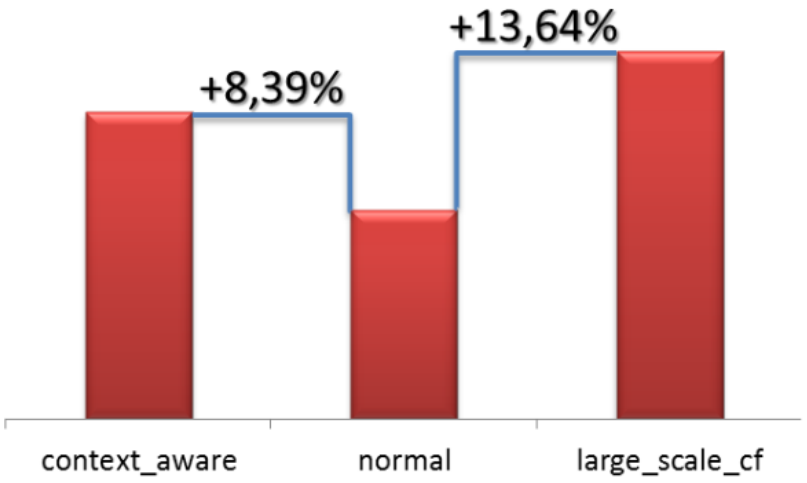
- Quantidade de e-mails enviados (sent)
- Quantidade de e-mails abertos (open)
- Quantidade de e-mails clicados (click)
- Quantidade de ordens geradas (orders)
- Receita gerada (revenue)

- open (%): open / sent
- CTR (%): click / sent
- click to open (%): click / open
- conversion (%): orders / click
- revenue / email: revenue / sent

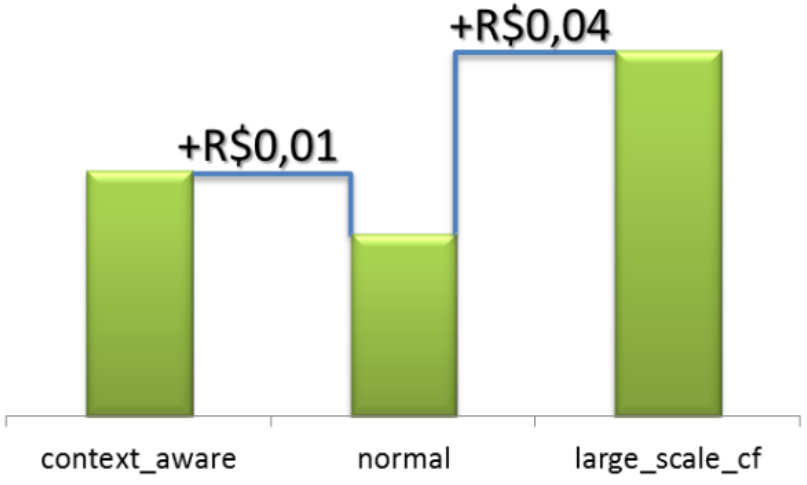
data	base	CTR %	click to open %	revenue / email
29/10/2014	normal	0.00%	0.00%	R$ 0.00
29/10/2014	context_aware	1.94%	6.83%	-R$ 0.06
29/10/2014	large_scale_cf	15.92%	14.59%	R$ 0.04
30/10/2014	normal	0.00%	0.00%	R$ 0.00
30/10/2014	context_aware	5.68%	5.58%	-R$ 0.02
30/10/2014	large_scale_cf	8.54%	6.75%	-R$ 0.03
31/10/2014	normal	0.00%	0.00%	R$ 0.00
31/10/2014	context_aware	4.04%	11.91%	R$ 0.04
31/10/2014	large_scale_cf	6.65%	9.35%	R$ 0.02
05/11/2014	normal	0.00%	0.00%	R$ 0.00
05/11/2014	context_aware	24.51%	22.25%	R$ 0.08
05/11/2014	large_scale_cf	31.27%	25.17%	R$ 0.12
06/11/2014	normal	0.00%	0.00%	R$ 0.00
06/11/2014	context_aware	18.39%	14.28%	R$ 0.05
06/11/2014	large_scale_cf	16.85%	25.84%	R$ 0.09

base	CTR %	click to open %	revenue / email
normal	0.00%	0.00%	R$ 0.00
context_aware	8.39%	9.67%	R$ 0.01
large_scale_cf	13.65%	14.06%	R$ 0.04

Interesse do Usuário



Receita (por e-mail enviado)



Conclusão

- Recomendação melhorou o desempenho tanto em procura (clicks) quanto em receita;
- Collaborative Filtering teve um desempenho melhor;
- Context-aware precisa selecionar muito bem os contextos e também aumentar o número de features (nf);

Próximos passos

- Recomendação online;
- Incorporar dados de navegação e carrinho no Collaborative Filtering;
- Testar novos contextos e aumentar o número de features;
- Testar recomendação conjunta dos 2 algoritmos.

DÚVIDAS?