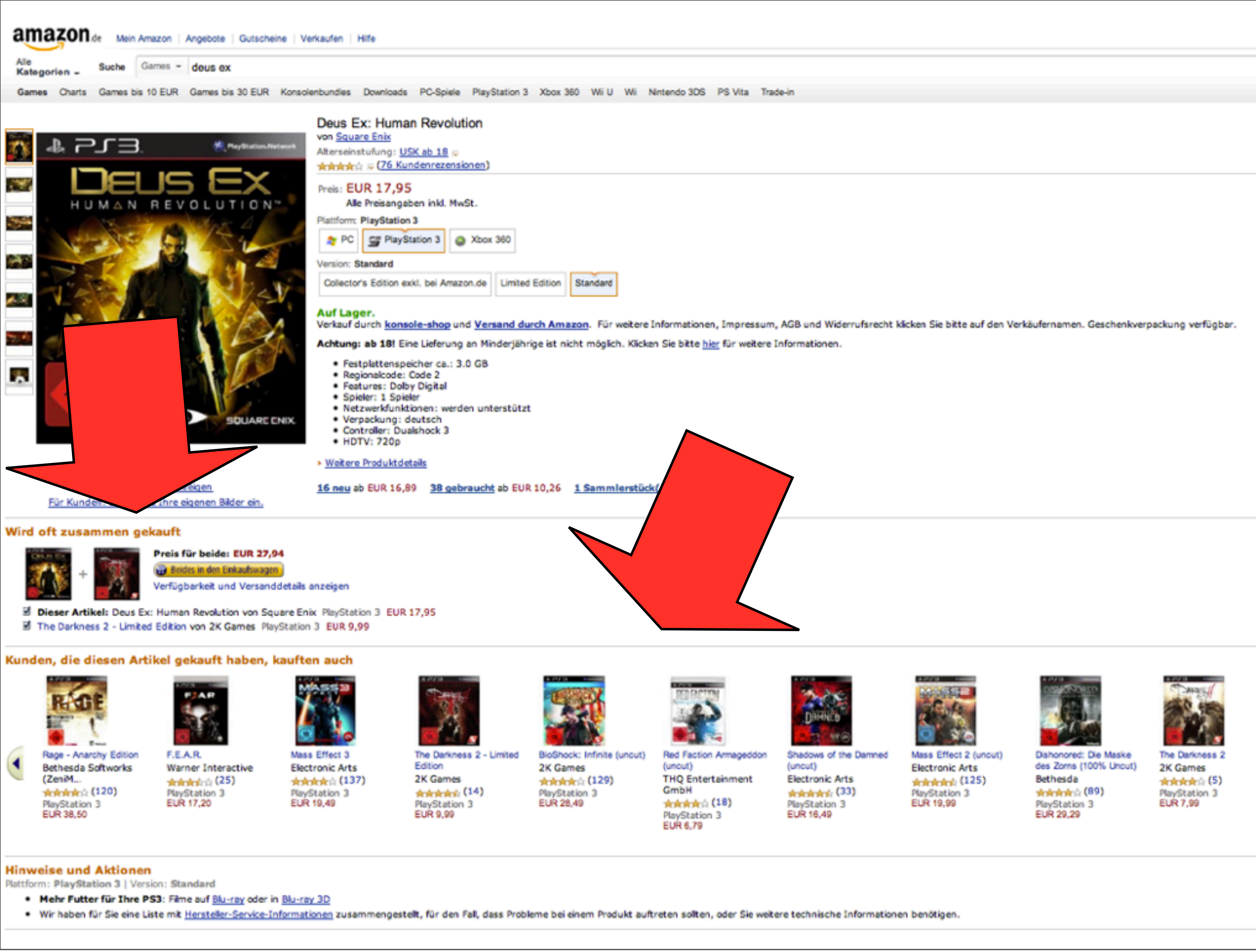
Large-scale Information Processing, Summer 2013

Recommender Systems

Ulf Brefeld
Knowledge Mining & Assessment
brefeld@kma.informatik.tu-darmstadt.de

TECHNISCHE UNIVERSITÄT DARMSTADT

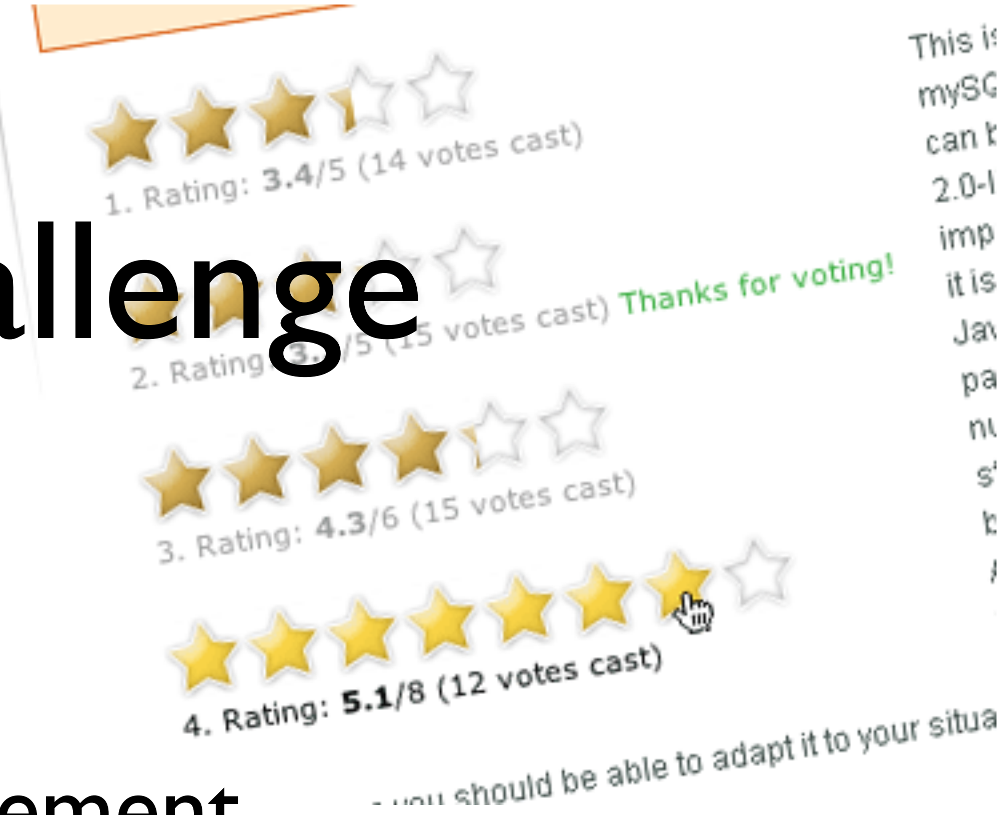
DIPF
Educational Research and Educational Information



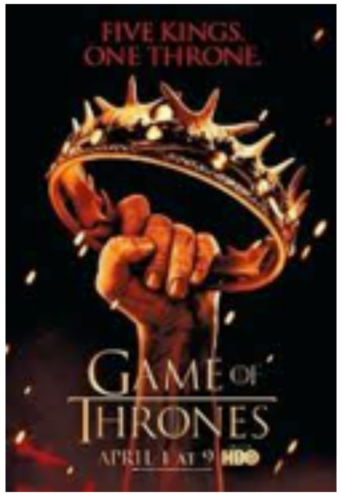
amazon.de
Mein Amazon | Angebote | Gutscheine | Verkaufen | Hilfe
Alle Kategorien
Suche
Games
deus ex
Games
Charts
Games bis 10 EUR
Games bis 30 EUR
Konsolenbundles
Downloads
PC-Spiele
PlayStation 3
Xbox 360
Wii U
Wii
Nintendo 3DS
PS Vita
Trade-in
PS3
DEUS EX
HUMAN REVOLUTION
SQUARE ENIX
Deus Ex: Human Revolution
von Square Enix
Alterseinstufung: USK ab 18
(76 Kundenrezensionen)
Preis: EUR 17,95
Alle Preisangaben inkl. MwSt.
Plattform: PlayStation 3
PC
PlayStation 3
Xbox 360
Version: Standard
Collector's Edition exkl. bei Amazon.de
Limited Edition
Standard
Auf Lager.
Verkauf durch konsole-shop und Versand durch Amazon. Für weitere Informationen, Impressum, AGB und Widerrufsrecht klicken Sie bitte auf den Verkäufernamen. Geschenkverpackung verfügbar.
Achtung: ab 18! Eine Lieferung an Minderjährige ist nicht möglich. Klicken Sie bitte hier für weitere Informationen.
Festplattenspeicher ca.: 3.0 GB
Regionalcode: Code 2
Features: Dolby Digital
Spieler: 1 Spieler
Netzwerkfunktionen: werden unterstützt
Verpackung: deutsch
Controller: Dualshock 3
HDTV: 720p
Weitere Produktdetails
16 neu ab EUR 16,89
38 gebraucht ab EUR 10,26
1 Sammlerstück
Für Kunden: Ihre eigenen Bilder ein.
Wird oft zusammen gekauft
Preis für beide: EUR 27,94
Beides in den Einkaufswagen
Verfügbarkeit und Versanddetails anzeigen
Dieser Artikel: Deus Ex: Human Revolution von Square Enix PlayStation 3 EUR 17,95
The Darkness 2 - Limited Edition von 2K Games PlayStation 3 EUR 9,99
Kunden, die diesen Artikel gekauft haben, kauften auch
Rage - Anarchy Edition
Bethesda Softworks (ZeniM...
(120)
PlayStation 3
EUR 38,50
F.E.A.R.
Warner Interactive
(25)
PlayStation 3
EUR 17,20
Mass Effect 3
Electronic Arts
(137)
PlayStation 3
EUR 19,49
The Darkness 2 - Limited Edition
2K Games
(14)
PlayStation 3
EUR 9,99
BioShock: Infinite (uncut)
2K Games
(129)
PlayStation 3
EUR 28,49
Red Faction Armageddon (uncut)
THQ Entertainment GmbH
(18)
PlayStation 3
EUR 6,79
Shadows of the Damned (uncut)
Electronic Arts
(33)
PlayStation 3
EUR 16,49
Mass Effect 2 (uncut)
Electronic Arts
(125)
PlayStation 3
EUR 19,99
Dishonored: Die Maske des Zorns (100% Uncut)
Bethesda
(89)
PlayStation 3
EUR 29,29
The Darkness 2
2K Games
(5)
PlayStation 3
EUR 7,99
Hinweise und Aktionen
Plattform: PlayStation 3 | Version: Standard
Mehr Futter für Ihre PS3: Filme auf Blu-ray oder in Blu-ray 3D
Wir haben für Sie eine Liste mit Hersteller-Service-Informationen zusammengestellt, für den Fall, dass Probleme bei einem Produkt auftreten sollten, oder Sie weitere technische Informationen benötigen.

Hilfe | Mein Benutzerkonto | Wunschzettel | Gutscheine | Newsletter | Jobs | Anmelden
zalando
Lieblingsprodukt suchen...
Warenkorb (0)
0,00 €
DAMEN HERREN KINDER MAGAZIN
Schuhe Bekleidung Sport Accessoires Premium Wohnen Marken Inspiration Fashion Deal SALE
Lounge
< Zurück | Startseite > ... > Sportschuhe > Fußballschuhe > Multinocken-Sohle >
Nike Performance
TIEMPO MYSTIC IV - Fußballschuh Multinocken - blue/white
Mehr Nike Performance | Mehr Multinocken-Sohle
Lieferbar innerhalb von 1-3 Werktagen
Gewählte Farbe
Mehr Farben
Wählen Sie Ihre Größe | Größentabelle
EU | Herstellergrößen
38 38,5 39 40 40,5 41 42 42,5 43
44 44,5 45 45,5 46 47 47,5 48,5 49,5
Gefällt mir
5 Personen gefällt das.
Bisher 74,95 €
Jetzt ab 44,95 €
Sie sparen 40%
Inkl. 19% MwSt.
Versand kostenlos
In den Warenkorb
Auf den Wunschzettel
Weiterempfehlen
Meine Größe anfragen
zalando lounge
Entdecken Sie jetzt den exklusiven Shopping-Club von Zalando!
Tägl. neue Verkaufsaktionen
Topmarken bis zu 70% reduziert
Jetzt kostenlos registrieren
HIER ANMELDEN
Produktdetails
Markendetails für Nike Performance
Kundenmeinungen (41)
Details für Nike Performance TIEMPO MYSTIC IV - Fußballschuh Multinocken - blue/white
Naturrasen kann ja ganz schön sein, aber auch Kunstrasen hat seine Vorteile. Zum Beispiel springt der Ball immer gleich hoch, egal ob es nass oder trocken ist, und er ist schneller. Das erfordert natürlich mehr Präzision beim Spielen. Damit diese nicht an den Schuhen scheitert, hat uns Nike mit dem TIEMPO MYSTIC IV bedacht.
multidirektionale Kunstrasenstollen mit Flexkerben und Pivot-Punkten für Traktion und Flexibilität
Innenmaterial: Textil
Innensohle - Technologie: EVA
Extras: verstärkte Ferse
Innensohle: konturiert, gepolstert
Außensohle: Multinocken (Turf), Flex-Kerben
Sportart: Fußball
Obermaterial: Leder/Synthetik
Artikelnummer: N1242A00J-550
Das könnte Sie auch interessieren
Nike Performance
TIEMPO NATURAL IV LTR TF - Fußballschuh Multinocken - black/white/electric green
49,95 €

The Netflix Challenge



- Movie recommendations
- $1,000,000 for 10% rmse improvement
- Data set: 100m ratings, 480k users, 18k movies
- Before: Cinematch score = 0.9525 rmse
- Winners score (after 2 years!) = 0.8567 rmse



Items

- Items: articles, ads, products, media, users, movies...
- Item/Domain characteristics:
  - Size of item pool (the Web, 2k documents)
  - Quality (e.g., editorial content, unstructured text)
  - Lifetime (e.g., constant, many new items)
  - Additional context (link graph, user/item attributes)

Properties

- Feedback:
  - Type (click, rating,...)
  - Latency (instant, 14 days)
  - Volume (200 per day, 300M per day)
- Training (online vs. offline)
- Constraints:
  - Relevance, diversity, freshness
  - Business rules (up-selling, cross-selling)

Optimization Criterion

- Goal: Display "best" items for each user visit
- Optimization criterions: click through rate (CTR), conversion-rate, revenue, engagement, relevance score, ...
- Example:
  - User satisfaction/engagement
  - BUT: no quick feedback
  - Approximate by #clicks (constrained by freshness, relevance, diversity, etc)

Personalization

- However: Every user is unique in her interests
- Remedy: Learn a personalized model for every user?
- Challenges:
    - Does it scale w/ the application?
    - Group users (e.g., clustering etc.)
    - How many clusters?
    - Interpretability?

Overview

- Apriori algorithm
- Collaborative filtering
-

The Apriori Algorithm

Agrawal & Srikant, 1994

- People who bought X also bought Y
- Mining frequent itemsets for association rules
- Key concepts:
  - Frequent itemsets = sets of items with a minimum support (e.g., frequency)
  - Any subset of a frequent itemset must be frequent

Apriori in a Nutshell

- Task: Find all frequent itemsets (= sets of items that have a minimum support)
- Subsets of frequent itemsets are also frequent
  - If {a,b} is frequent, so are {a} and {b}
- Iterative procedure to find frequent itemsets with cardinality 1 to k (k-itemset)
  - Dynamic programming!

Pseudo Code

- Join step: C_k is generated by joining L_{k-1} with itself
- Prune step: Any infrequent (k-1)-itemset cannot be a subset of a frequent k-itemset

```
L1 = {frequent items}
for (k=1; Lk!=∅; k++)
   Ck+1 = candidates generated from Lk
   for each transaction t in database do
      increment count of all candidates in
      Ck+1 that are contained in t
   Lk+1 = candidates in Ck+1 with min
   support
return ∪kLk
```

L_k = frequent itemset of size k
C_k = candidate itemset of size k

Example k=1

- Transaction database

w1 =
w2 =
w3 =
w4 =
w5 =
w6 =
w7 =

- Min support = 3/7

Example k=1

L_1

= 3/7
= 6/7
= 4/7
= 5/7

- Transaction database

w1 =
w2 =
w3 =
w4 =
w5 =
w6 =
w7 =

- Min support = 3/7

Example k=2

L1

= 3/7
= 6/7
= 4/7
= 5/7

- Transaction database

w1 =
w2 =
w3 =
w4 =
w5 =
w6 =
w7 =

L2

= 3/7
= 1/7
= 2/7
= 3/7
= 4/7
= 3/7

- Min support = 3/7

Example k=2

L1

- Transaction database

L2

- Min support = 3/7

Tuesday, July 16, 13

Example k=3

- Transaction database

- Min support = 3/7

Example k=3

L_1

= 3/7

= 6/7

= 4/7

= 5/7

- Transaction database

w1 =

w2 =

w3 =

w4 =

w5 =

w6 =

w7 =

L_2

+ = 3/7

[redacted]

+ = 3/7

+ = 4/7

+ = 3/7

L_3

[redacted]

- Min support = 3/7

Association Rules

- We identified the following frequent sets:

- Cannot make much out of L_1

Association Rules

- We identified the following frequent sets:

- Cannot make much out of L_1

Association Rules

- We identified the following frequent sets:

- For every non-empty subset of s of I generate rule:

$$s \rightarrow (I-s) \text{ if } support(I)/support(s) \geq min\ conf$$

Association Rules - Example

- min confidence = 0.75

(3/7) / (6/7) = 0.5

$$s \rightarrow (I - s) \text{ if support}(I)/\text{support}(s) \geq \text{min conf}$$

Association Rules - Example

- min confidence = 0.75

(3/7) / (6/7) = 0.5 < 0.75

$s \rightarrow (I-s)$ if $support(I)/support(s) \geq min\ conf$

Association Rules - Example

- min confidence = 0.75

[redacted]

[redacted] 0.75

s→(I-s) if support(I)/support(s)≥min conf

Association Rules - Example

- min confidence = 0.75

~~(3/7) / (6/7) = 0.5 < 0.75~~

 ⟶ 

(3/7) / (3/7) = 1 > 0.75

 +  = 3/7

 +  = 3/7

 +  = 4/7

 +  = 3/7

 = 3/7

 = 6/7

 = 4/7

 = 5/7

$s \rightarrow (I-s)$ if support(I)/support(s)≥min conf

Association Rules - Example

- min confidence = 0.75

When a user is about to buy (viewing)  recommend

~~(3/7) / (6/7) = 0.5 < 0.75~~

(3/7) / (3/7) = 1 > 0.75

$s \rightarrow (I-s)$ if $support(I)/support(s) \geq min\ conf$

Some Improvements

- Hash-based itemset counting (k-itemsets whose hash bucket count is below threshold cannot be frequent)
- Database reductions (transactions not containing a frequent k-itemset is useless for subsequent scans)
- Partitioning (Potentially frequent itemsets must be frequent in at least one partition of the database)
- ...

Apriori w/ Hadoop

Li & Zhang, 2011

- Naive approach: compute everything at once
- Use only a single map/reduce phase
- (Similar to the word count example)
- Pros: Easy to implement
- Cons: Slow!

Mapper

- Input: key = pos. in file, value = transaction t
- Output: key(s) = subset(s) of t, value = one (const.)

```
map (key: position, value: t){
  foreach itemset I in t:
    write(I,one)
  end
}
```

Reducer

- Input: key = itemset, value = one (const.)
- Output: key = itemset, value = frequency

```
reduce (key: I, value: c){
  sum = 0
  while c.hasNext():
    sum++
  end
  if (sum ≥ min support count)
    write(I,sum)
}
```

Sketch of a k-phase implementation

(e.g., Yang et al., 2010; Li et al., 2012; Lin et al., 2012)

- Mapper I computes candidate 1-itemset from transactions
- Reducer computes global frequencies of 1-itemsets and writes L_1 to distributed cache
- Mapper II
  - Setup: reads L_{k-1} from cache, generates C_k
  - Map: for each candidate c in subset (C_k,t) write (c,one)

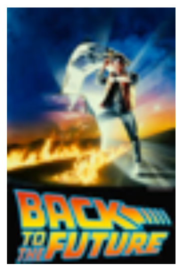
One-phase v. k-phase

- One-phase:
  - Uses a single pass over the database
  - Huge network load by generating ALL candidate pairs (also infrequent ones) at once
- k-phase:
  - Needs k passes over the database
  - Network load minimal as only frequent itemsets are passed on the next iteration

Collaborative Filtering

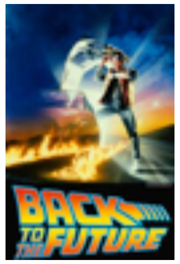
- Heuristic to generate "personalized" recommendations
- Here: personalized wrt similar users
- User-to-item similarities
- Personalized recommendations

User-Item Matrix



Paul							
Marie							
Susi							
Mike							
Anke							

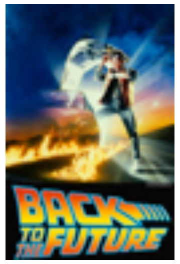
User-Item Matrix



Paul							
Marie							
Susi							
Mike							
Anke							

- Example: users rate movies with scores 1-5

Item-to-item similarities

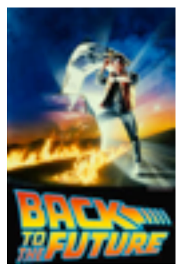


Paul	4	5		4	4		
Marie			3	1			
Susi		5	1				
Mike					5	5	1
Anke		1		5			

- Common choice: use cosine of column vectors

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

Item-to-item similarities

							
Paul	4	5		4	4		
Marie			3	1			
Susi		5	1				
Mike					5	5	1
Anke		1		5			

- Common choice: use cosine of column vectors

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

Item-to-item Similarities

cos( , ) = $25/(51^{.5} * 42^{.5})$
= 0.54

Mike					5	5	1
Anke		1		5			

- Common choice: use cosine of column vectors

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

Item-to-item Similarities

cos(, ) = 25/(51$^{.5}$*42$^{.5}$)
= 0.54

P							
M							
S							
Mike					[illegible]	[illegible]	1
Anke		1		5			

- Common choice: use cosine of column vectors

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

Item-to-item similarities




Alternatives to cosine, e.g., Pearson-correlation, Tanimoto coefficient, custom similarities,...

Paul							
Marie							
Susi		5	1				
Mike					5	5	1
Anke		1		5			

- Common choice: use cosine of column vectors

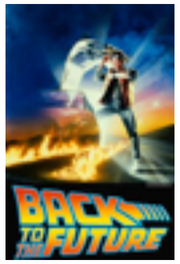
$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

Recommendations

Paul	4	5		4	4		
Marie			3	1			
Susi		5	1	???			
Mike					5	5	1
Anke		1		5			

- How would Susi like Rambo 3?

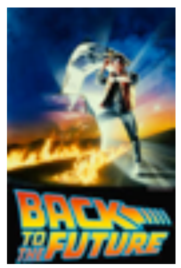
Recommendations



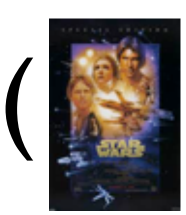
Paul	4	5		4	4		
Marie			3	1			
Susi		5	1	???			
Mike					5	5	1
Anke		1		5			

- How would Susi like Rambo 3?
- Use linear combination of preferences and similarities

Recommendations



Paul	4	5		4	4		
Marie			3	1			
Susi		5	1	???			
Mike					5	5	1
Anke		1		5			

like(Susi, ) = 5 * sim(, ) + 1 * sim(, )
= ...

CF w/ Hadoop

- Key ideas for efficiency:
  - Discard item pairs without ratings
- Similar to compute pairwise similarities of text documents (e.g., Jaccard similarity)
- CF with Hadoop needs two phases
- Mahout contains a readily available implementation

First phase

- Map:
  - Input: value = {user, item, rating}-triplets
  - Output: key = user, value = {item, rating}
- Reduce:
  - Input: key = user, value = {item, rating}
  - Output: key = user, value = {{item, rating}}

First phase

- Map:
  - Input: value = {user, item, rating}-triplets
  - Output: key = user, value = {item, rating}
- Reduce:
  - Input: key = user, value = {item, rating}
  - Output: key = user, value = {{item, rating}}

Inverted index!

key=Marie, value={{StarWars,3}{Rambo,1}}

Second phase

- Map:
  - Input: key = user, value = {{item, rating}}
  - Output: key = {item,item}, value = {rating,rating}
- Reduce:
  - Input: key = {item,item}, value = {rating,rating}
  - Output: key = {item,item}, value = similarity

Second phase

- Map: key={StarWars,Rambo}, value = {3,1}
  - Input: key = user, value = {{item, rating}}
  - Output: key = {item,item}, value = {rating,rating}
- Reduce:
  - Input: key = {item,item}, value = {rating,rating}
  - Output: key = {item,item}, value = similarity

CF Summary

- Pro's:
  - Easy to implement
- Con's:
  - Heuristic
  - Transparency issues (peers change)
  - New item/user problem

CF Extensions

- Matrix factorization
- Features
- Rating biases

Multi-armed Bandits

- Which machine has the highest expected reward?

Contextual Bandits

- Environment defines a context $\mathbf{x}_t \in X$
- Learner chooses an action $a_t \in \{1,...,k\}$
- Environment outputs a reward $r_t \in [0,1]$

- Goal: Learn a good policy $\pi: X \rightarrow \{1,...,k\}$ to minimize the regret

$$regret = \max_{\pi} \sum_{t=1}^{T} r_t(\pi(\boldsymbol{x}_t)) - \sum_{t=1}^{T} r_t(a_t)$$

Analogies/Differences

- Supervised learning:
  - Loss function is unknown as reward for untaken actions is unknown
- Unsupervised learning:
  - Not unsupervised as there is partial labels in form of rewards
- Reinforcement learning:
  - Easier than reinforcement learning as we can always assign rewards to actions

Observation

- Problem setting matches many online prediction problems:
  - Placing advertisements on Web pages
  - Recommendations
  - Reward could be click/no click or revenue of click, etc.
- Exploration/exploitation trade-off

UCB

- Estimate r_t of each arm a together with a confidence interval c of the estimation
- Choose arm a_t that realizes

$$a_t = \underset{a}{\operatorname{argmax}}\, \hat{r}_{t,a} + c_{t,a}$$

- For appropriately defined c, regret is only logarithmic in T (which is optimal!)
- Upper Confidence Bound (UCB) algorithms

The Model

- Every product/item is a contextualized bandit
- Choosing arm a is equal to
  - Recommend product a
  - Display advertisement a
  - ...
- Context describes the actual page, user history, etc.

LinUCB

Li et al., WWW2010

- Use disjoint linear models: $\mathbb{E}\left[r_{t,a}|\boldsymbol{x}_{t,a}\right] = \langle \boldsymbol{w}_a^*, \boldsymbol{x}_{t,a} \rangle$
  - Build matrix of contexts $X_a \in \mathbb{R}^{m \times d}$
  - Stack-up rewards into vector $\boldsymbol{b} = (\ldots, b_{t,a}, \ldots)^\top$
- Regularized least squares regression gives

$$\boldsymbol{w}_a = (X_a^\top X_a + \mathbf{1})^{-1} X_a \boldsymbol{b}_a$$

  - (see lecture on linear models)

Confidence RLSR

Li et al., WWW2010

- With probability at least 1-δ

$$|\langle \boldsymbol{w}_a, \boldsymbol{x}_{t,a} \rangle - \mathbb{E}\left[r_{t,a} | \boldsymbol{x}_{t,a}\right]| \leq \left(1 + \sqrt{\frac{\log(2/\delta)}{2}}\right) \sqrt{\boldsymbol{x}_{t,a}^\top (X_a^\top X_a + \mathbf{1})^{-1} \boldsymbol{x}_{t,a}}$$

Confidence RLSR

Li et al., WWW2010

- With probability at least 1-δ

$$|\langle \boldsymbol{w}_a, \boldsymbol{x}_{t,a}\rangle - \mathbb{E}\left[r_{t,a}|\boldsymbol{x}_{t,a}\right]| \leq \underbrace{\boxed{\left(1+\sqrt{\frac{\log(2/\delta)}{2}}\right)}}_{\alpha} \sqrt{\boldsymbol{x}_{t,a}^\top \underbrace{\boxed{(X_a^\top X_a + \mathbf{1})}}_{A_a}{}^{-1} \boldsymbol{x}_{t,a}}$$

- That is, according to UCB, choose the arm (item) according to:

$$a_t = \underset{a}{\operatorname{argmax}} \left(\langle \boldsymbol{w}_a, \boldsymbol{x}_{t,a} \rangle + \alpha \sqrt{\boldsymbol{x}_{t,a}^\top A^{-1} \boldsymbol{x}_{t,a}} \right)$$

Confidence RLSR

Li et al., WWW2010

- With probability at least 1-δ

$$\left|\langle \boldsymbol{w}_a, \boldsymbol{x}_{t,a}\rangle - \mathbb{E}\left[r_{t,a}|\boldsymbol{x}_{t,a}\right]\right| \leq \underbrace{\boxed{\left(1+\sqrt{\frac{\log(2/\delta)}{2}}\right)}}_{\alpha} \sqrt{\boldsymbol{x}_{t,a}^{\top} \underbrace{\boxed{(X_a^{\top} X_a + \mathbf{1})}}_{A_a}{}^{-1} \boldsymbol{x}_{t,a}}$$

- That is, according to UCB, choose the arm (item) according to:

$$a_t = \underset{a}{\operatorname{argmax}} \left(\langle \boldsymbol{w}_a, \boldsymbol{x}_{t,a}\rangle + \alpha \sqrt{\boxed{\boldsymbol{x}_{t,a}^{\top} A^{-1} \boldsymbol{x}_{t,a}}} \right)$$

predictive variance of expected reward

Confidence RLSR

Li et al., WWW2010

- With probability at least 1-δ

$$|\langle \boldsymbol{w}_a, \boldsymbol{x}_{t,a}\rangle - \mathbb{E}\left[r_{t,a}|\boldsymbol{x}_{t,a}\right]| \leq \overbrace{\left(1+\sqrt{\frac{\log(2/\delta)}{2}}\right)}^{\alpha}\sqrt{\boldsymbol{x}_{t,a}^\top \underbrace{(X_a^\top X_a + \mathbf{1})}_{A_a}{}^{-1}\boldsymbol{x}_{t,a}}$$

- That is, according to UCB, choose the arm (item) according to:

$$a_t = \operatorname*{argmax}_a \left(\langle \boldsymbol{w}_a, \boldsymbol{x}_{t,a}\rangle + \alpha \overbrace{\sqrt{\underbrace{\boldsymbol{x}_{t,a}^\top A^{-1}\boldsymbol{x}_{t,a}}_{\text{predictive variance of expected reward}}}}^{\text{standard deviation}}\right)$$

Confidence RLSR

Li et al., WWW2010

- With probability at least 1-δ

$$\left|\langle \boldsymbol{w}_a, \boldsymbol{x}_{t,a}\rangle - \mathbb{E}\left[r_{t,a}|\boldsymbol{x}_{t,a}\right]\right| \leq \overbrace{\boxed{\left(1+\sqrt{\frac{\log(2/\delta)}{2}}\right)}}^{\alpha} \sqrt{\boldsymbol{x}_{t,a}^\top \underbrace{\boxed{(X_a^\top X_a + \mathbf{1})}}_{A_a}{}^{-1} \boldsymbol{x}_{t,a}}$$

- That is, according to UCB, choose the arm (item) according to:

$$a_t = \underset{a}{\operatorname{argmax}} \left(\langle \boldsymbol{w}_a, \boldsymbol{x}_{t,a}\rangle + \boxed{\alpha} \boxed{\sqrt{\boxed{\boldsymbol{x}_{t,a}^\top A^{-1} \boldsymbol{x}_{t,a}}}}\right)$$

standard deviation

treat as parameter
if too conservative
(too large)

predictive variance of expected reward

Analysis

Li et al., WWW2010

- Linear in #arms, cubic in #dimensions
- Updates of A could be cached to reduce recomputations
- Trades-off exploitation/exploration automatically due to incorporation of confidence
- Perfect for dynamic domains with many new items, e.g., news articles

Algorithm

Li et al., WWW2010

Input $\alpha \in \mathbb{R}^+$
for $t = 1, 2, \ldots, T$
    compute contexts $\boldsymbol{x}_{a,t}$ for all arms a
    for all arms a
        if a is new
            $A_a \leftarrow \mathbf{1}$
            $b_a \leftarrow \mathbf{0}$
        end
        $\boldsymbol{w}_{t,a} \leftarrow A_a^{-1} \boldsymbol{b}_a$
        $p_{t,a} \leftarrow \langle \boldsymbol{w}_a, \boldsymbol{x}_{t,a} \rangle + \alpha \sqrt{\boldsymbol{x}_{t,a}^\top A_a^{-1} \boldsymbol{x}_{t,a}}$
    end
    Choose arm $a_t = \operatorname{argmax}_a p_{t,a}$
    Receive reward r_t
    $A_a \leftarrow A_a + \boldsymbol{x}_{t,a} \boldsymbol{x}_{t,a}^\top$
    $\boldsymbol{b} \leftarrow \boldsymbol{b}_{a,t} + r_t \boldsymbol{x}_{t,a}$
end

Some hints from Prüfungsbüro

http://www.intern.tu-darmstadt.de/dez_ii/pruefungsmanagement/pruefende/pruefende.de.jsp