

**“SORTING IT OUT”**

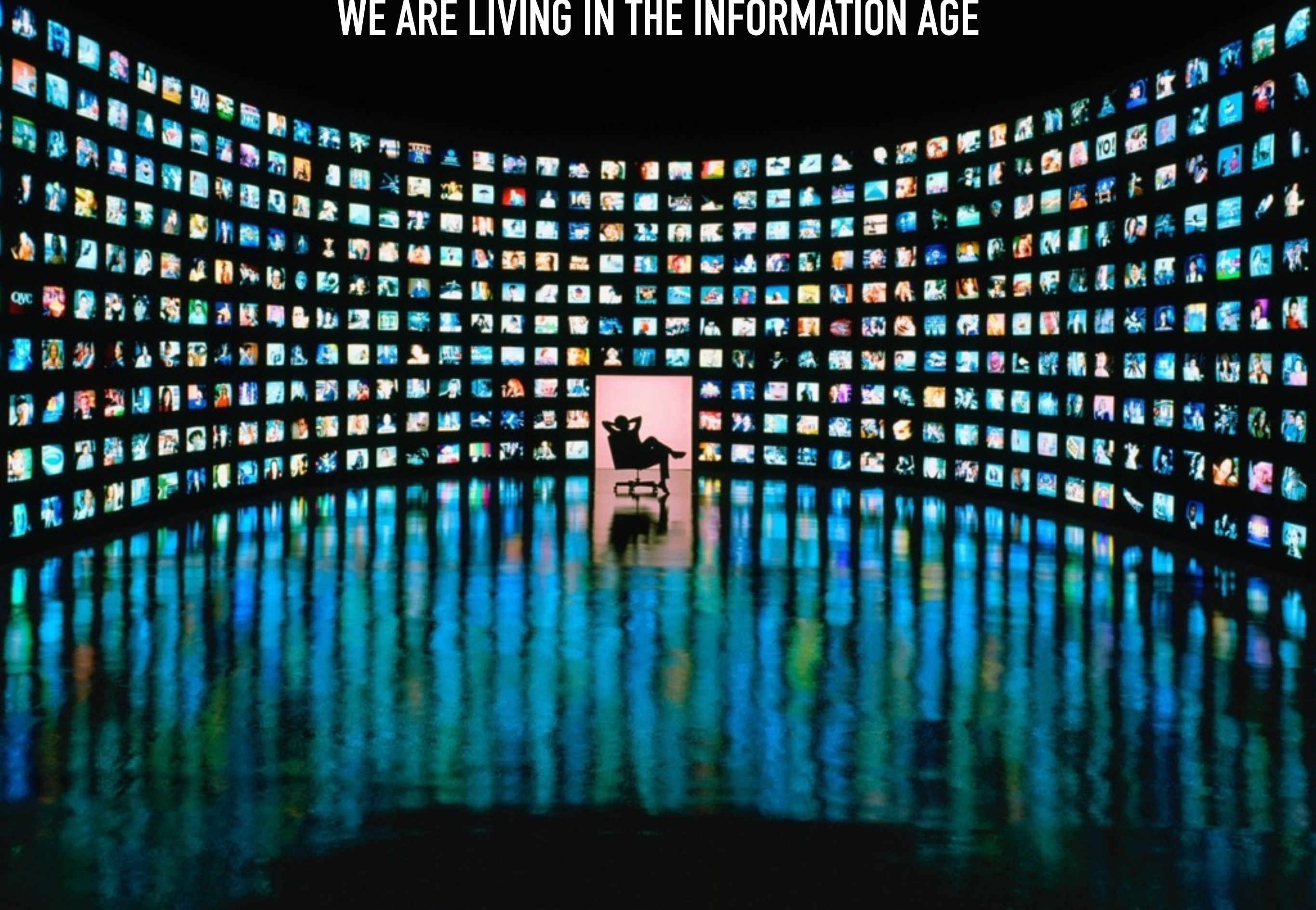
REAL WORLD CHALLENGES FOR

---

**RECOMMENDER SYSTEMS**



**WE ARE LIVING IN THE INFORMATION AGE**






**SEARCH**

**DISCOVERY**



# RECOMMENDATIONS ARE EVERYWHERE...



**recommendation ranking bias**

**recommendation position bias**

recommendation

receive

received

recommended

receipt

recruiting

recall

recent

Remove


Remove

Google Search


I'm Feeling Lucky

RECOMMENDATIONS ARE EVERYWHERE...

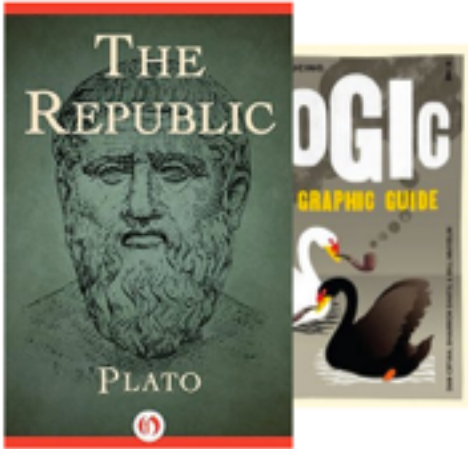
Recommended for you, aviad



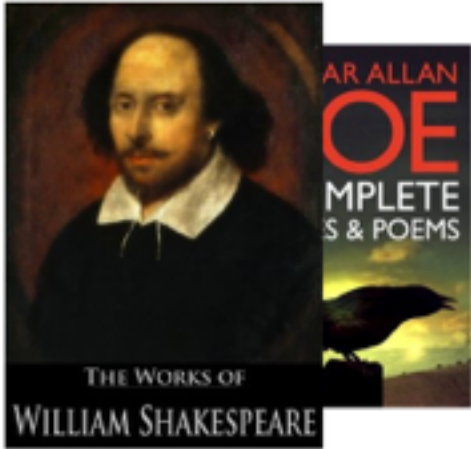
Religion & Spirituality Books  
65 ITEMS




Craft, Hobby & Home Books  
6 ITEMS




Historical Books  
14 ITEMS




Literature & Fiction  
88 ITEMS




Reference Books  
100 ITEMS



Mystery, Thriller & Suspense Books  
46 ITEMS



Stuffed Animals & Plush Toys  
5 ITEMS









Children's Books  
79 ITEMS

ITS ALL AROUND US

---

## RECOMMENDATIONS ARE EVERYWHERE...

You Might Like...  
Based on your recent installs

					
<b>Make7! Hexa Puzzle</b> ★★★★★ <b>FREE</b>	<b>Paper Toss 2.0</b> ★★★★★ <b>FREE</b>	<b>Troll Face Quest Unl</b> ★★★★★ <b>FREE</b>	<b>ChattingCat-English</b> ★★★★★ <b>FREE</b>	<b>Puzzles with Match</b> ★★★★★ <b>FREE</b>	<b>One touch Drawing</b> ★★★★★ <b>FREE</b>
<i>Popular with Brain It On! - Physics Puzzles users</i>	<i>Popular with Fishing Hook users</i>	<i>Popular with Brain It On! - Physics Puzzles users</i>	<i>Popular with WhiteSmoke Writing Assistant users</i>	<i>Popular with Brain It On! - Physics Puzzles users</i>	<i>Popular with Brain It On! - Physics Puzzles users</i>

# RECOMMENDATIONS ARE EVERYWHERE...

Customers who viewed Renaissance Hong Kong Harbour View Hotel also viewed:

## [Grand Hyatt Hong Kong](#)

★★★★★



Directly connected to the Hong Kong Convention and Exhibition Centre (HKCEC) and a 5-minute walk from Wanchai MTR subway station and Star Ferry Pier, the luxurious Grand Hyatt Hong Kong features a...

4 people are looking right now

Score from 574 reviews

**Excellent 8.9** /10

Total price from:

**THB 29,661**

[Book now](#)

## [InterContinental Hong Kong](#)

★★★★★



5-star InterContinental Hong Kong Hotel enjoys a waterfront location.

6 people are looking right now

Score from 1,457 reviews

**Wonderful 9** /10

Total price from:

**THB 30,333**

[Book now](#)

## [The Harbourview](#) ★★★★★



Overlooking Victoria Harbor, the Harbourview offers well-furnished rooms a 5-minute walk from Wan Chai MTR subway station. It features an on-site restaurant and Wi-Fi access in the entire hotel.

6 people are looking right now

Score from 5,442 reviews

**Good 7.7** /10

Total price from:

**THB 24,718**

[Book now](#)

## [JW Marriott Hotel Hong Kong](#)

★★★★★



Sitting atop the prestigious Pacific Place Complex, the JW Marriott Hotel Hong Kong has direct access to the Admiralty MTR Subway Station and Pacific Place Shopping Mall.

2 people are looking right now

Score from 1,256 reviews

**Excellent 8.7** /10

Total price from:

**THB 28,658**

[Book now](#)

## PROBLEM DEFINITION

Estimate a **utility function**  
to **predict** how  
a **user** will like an **item**.

Given a set of items  $I$  and a set of users  $U$   
find a utility function  $f : U \times I \rightarrow R$

## EXTENSION (CONTEXT)

Given  $I, U$  and a context  $C$   
find a utility function  $f : C \times U \times I \rightarrow R$





**JUST BEFORE WE DIVE IN  
THE ALGORITHMS....**

**A PRODUCT'S SUCCESS IS A  
PRODUCT OF IT'S ELEMENTS**

**DATA**

**CLEANING AND  
FILTERING**

**FEATURE  
EXTRACTION**

**MODELING**

**RECOMMENDER  
EVALUATION**

**PRODUCTION  
ENVIRONMENT**

**BUSINESS  
RULES**

**AB TESTING**

**BI**

**UX**

# WHAT ARE YOU TRYING TO ACHIEVE?

1. Increase volume/value of sales
2. Increase user satisfaction and loyalty (contradicts #1?)
3. Understand what users want



# HOW WELL CAN YOU IMPROVE?

## START SIMPLE

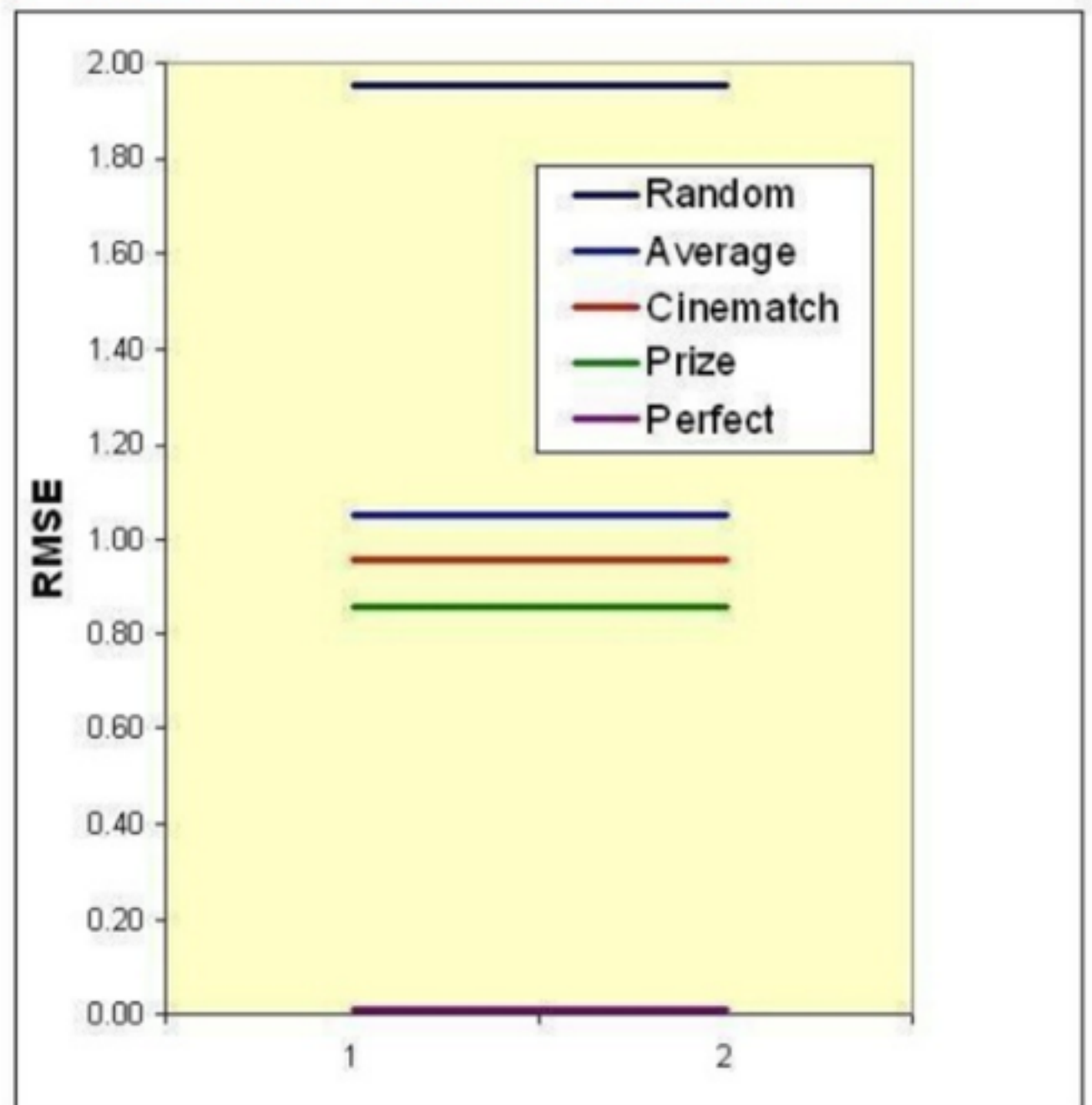
1. A random recommender is the worst you can do, having on knowledge or assumption on the data
2. Compare to item popularity - sort items by overall 'purchases' ever.
3. As you find better recommenders, always compare to the above baselines
4. Invest time and thought in a validation framework - more on this later...

# HOW WELL CAN YOU IMPROVE?

IT IS EASY TO CREATE SIMPLE RECOMMENDERS, BUT HARD TO BEAT THEM

Cinematch had an RMSE score of 0.9513 and the winning team reached 0.8728 which is a 8.26% improvement.

It took another 2 years to reach RMSE of 0.8567 which is 9.9% better

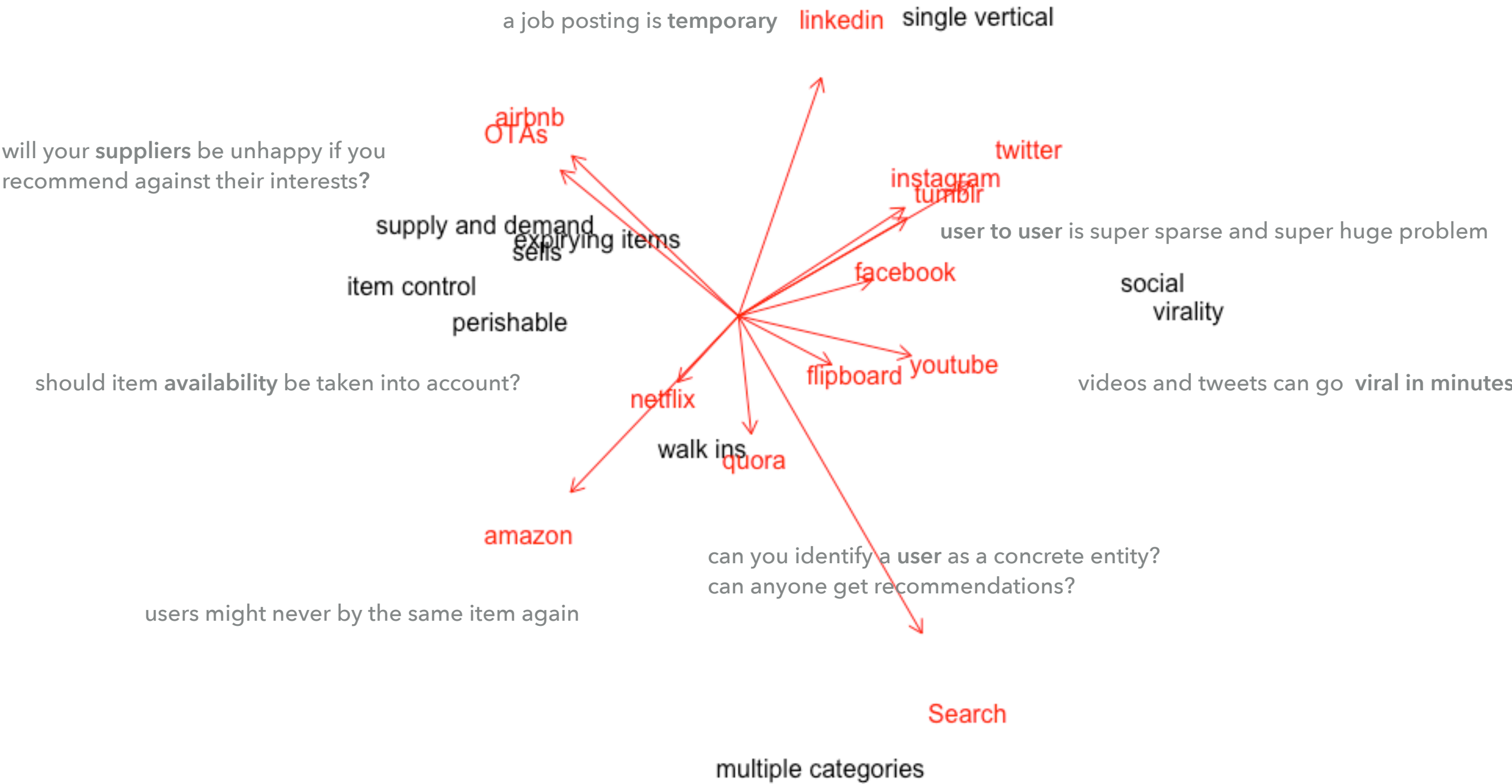


# WHAT ARE YOU RECOMMENDING? AND TO WHO?

1. Netflix - Movies to Watchers
2. Google Search - URLs/Images/News/... to Users submitting queries
3. Tumblr - Blogs to Blog Owners
4. Facebook - Users to Users, Content to Users
5. Amazon - Everything they can store to Everyone
6. LinkedIn - Jobs and Contacts to Users

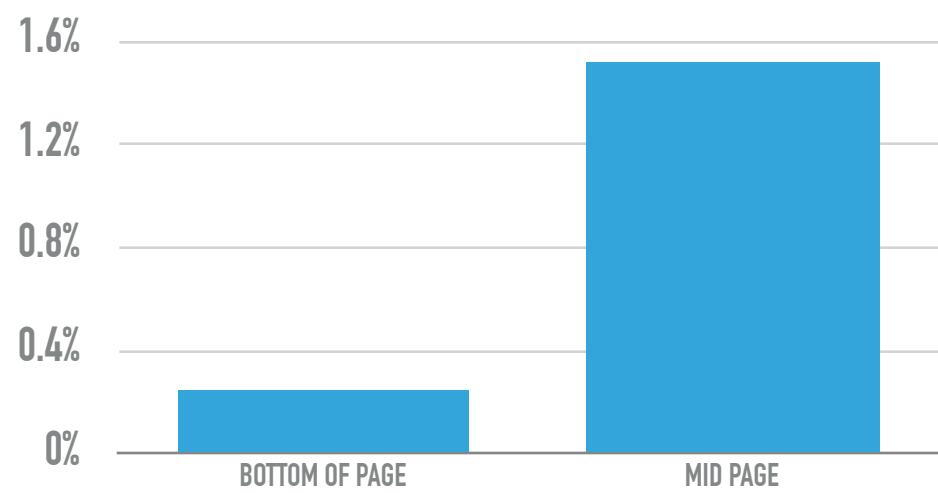


# WHAT ECOSYSTEM ARE YOU WORKING IN?

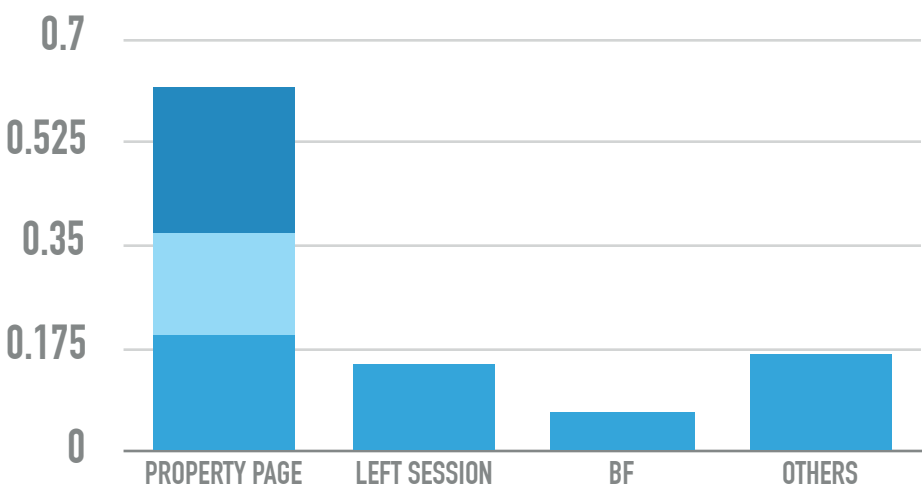


# MORE THAN AN ALGORITHM – HOW WILL IT BE USED?

CTR ~LOCATION




USER EXPERIENCE



## TITLES AND PRESENTATION


RECOMMENDED PROPERTIES



Centara Grand Beach Resort & Villas

★★★★★


Excellent 8.2 (1520 reviews)



The Tubkaak Boutique Resort

★★★★★


Excellent 8.9 (321 reviews)



Beyond Resort Krabi

★★★★★

Excellent 8.2 (1297 reviews)



Dusit Thani Krabi Beach Resort

★★★★★

Excellent 8.6 (102 reviews)

# WHAT ARE YOUR DATA SOURCES?

- ▶ Past behaviour - concrete user and items
- ▶ Relationships to other users via items or other sources - concrete user, social, CF
- ▶ Explicit signals vs Implicit signals - item reviews vs. on site behavior
- ▶ Item Content - when items expire or they are consumed one time
- ▶ Demography - when no login is available
- ▶ Context - when? where? with who?



# DATA SOURCES

Impression/Search based data  
very cumbersome, prone to bias

	<div><div>T</div>searchmessageid</div>	<div><div>📍</div>hotel_id</div>	<div><div>📍</div>rankinpage</div>	<div><div>📍</div>min_price_usd</div>	<div><div>☑️</div>breakfast</div>
1	00000efb-6607-473f-a34b-4638ecce2d25	527,794	1	109.67	true
2	00000efb-6607-473f-a34b-4638ecce2d25	501,851	2	52.55	false
3	00000efb-6607-473f-a34b-4638ecce2d25	91,269	3	113.73	false
4	00000efb-6607-473f-a34b-4638ecce2d25	10,860	4	136.23	false
5	00000efb-6607-473f-a34b-4638ecce2d25	8,350	5	67	false
6	00000efb-6607-473f-a34b-4638ecce2d25	161,872	6	134.58	true
7	00000efb-6607-473f-a34b-4638ecce2d25	263,304	7	26.82	false
8	00000efb-6607-473f-a34b-4638ecce2d25	66,525	8	144.59	true
9	00000efb-6607-473f-a34b-4638ecce2d25	197,206	9	231.21	false
10	00000efb-6607-473f-a34b-4638ecce2d25	9,388	10	56.66	false
11	00000efb-6607-473f-a34b-4638ecce2d25	240,525	11	31.86	false
12	00000efb-6607-473f-a34b-4638ecce2d25	10,850	12	57.64	true
13	00000efb-6607-473f-a34b-4638ecce2d25	108,498	13	58.97	false
14	00000efb-6607-473f-a34b-4638ecce2d25	91,270	14	64.89	false
15	00000efb-6607-473f-a34b-4638ecce2d25	149,080	15	28.4	false
16	00000efb-6607-473f-a34b-4638ecce2d25	1,061,229	16	120.66	false
17	00000efb-6607-473f-a34b-4638ecce2d25	240,277	17	40.93	false
18	00000efb-6607-473f-a34b-4638ecce2d25	148,785	18	31.85	false

**BENEFITS:**  
BI SOURCE TO  
UNDERSTAND WHAT WAS  
SHOWN, HOW ITEMS ARE  
RANKED, HOW USERS  
BEHAVE ON YOUR SITE

searchmessageid

hotel\_id

rankinpage

min\_price\_usd

breakfast

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

00000efb-6607-473f-a34b-4638ecce2d25

527,794

1

109.67

true

00000efb-6607-473f-a34b-4638ecce2d25

501,851

2

52.55

false

00000efb-6607-473f-a34b-4638ecce2d25

91,269

3

113.73

false

00000efb-6607-473f-a34b-4638ecce2d25

10,860

4

136.23

false

00000efb-6607-473f-a34b-4638ecce2d25

8,350

5

67

false

00000efb-6607-473f-a34b-4638ecce2d25

161,872

6

134.58

true

00000efb-6607-473f-a34b-4638ecce2d25

263,304

7

26.82

false

00000efb-6607-473f-a34b-4638ecce2d25

66,525

8

144.59

true

00000efb-6607-473f-a34b-4638ecce2d25

197,206

9

231.21

false

00000efb-6607-473f-a34b-4638ecce2d25

9,388

10

56.66

false

00000efb-6607-473f-a34b-4638ecce2d25

240,525

11

31.86

false

00000efb-6607-473f-a34b-4638ecce2d25

10,850

12

57.64

true

00000efb-6607-473f-a34b-4638ecce2d25

108,498

13

58.97

false

00000efb-6607-473f-a34b-4638ecce2d25

91,270

14

64.89

false

00000efb-6607-473f-a34b-4638ecce2d25

149,080

15

28.4

false

00000efb-6607-473f-a34b-4638ecce2d25

1,061,229

16

120.66

false

00000efb-6607-473f-a34b-4638ecce2d25

240,277

17

40.93

false

00000efb-6607-473f-a34b-4638ecce2d25

148,785

18

31.85

false

s1

s2: Sort by price

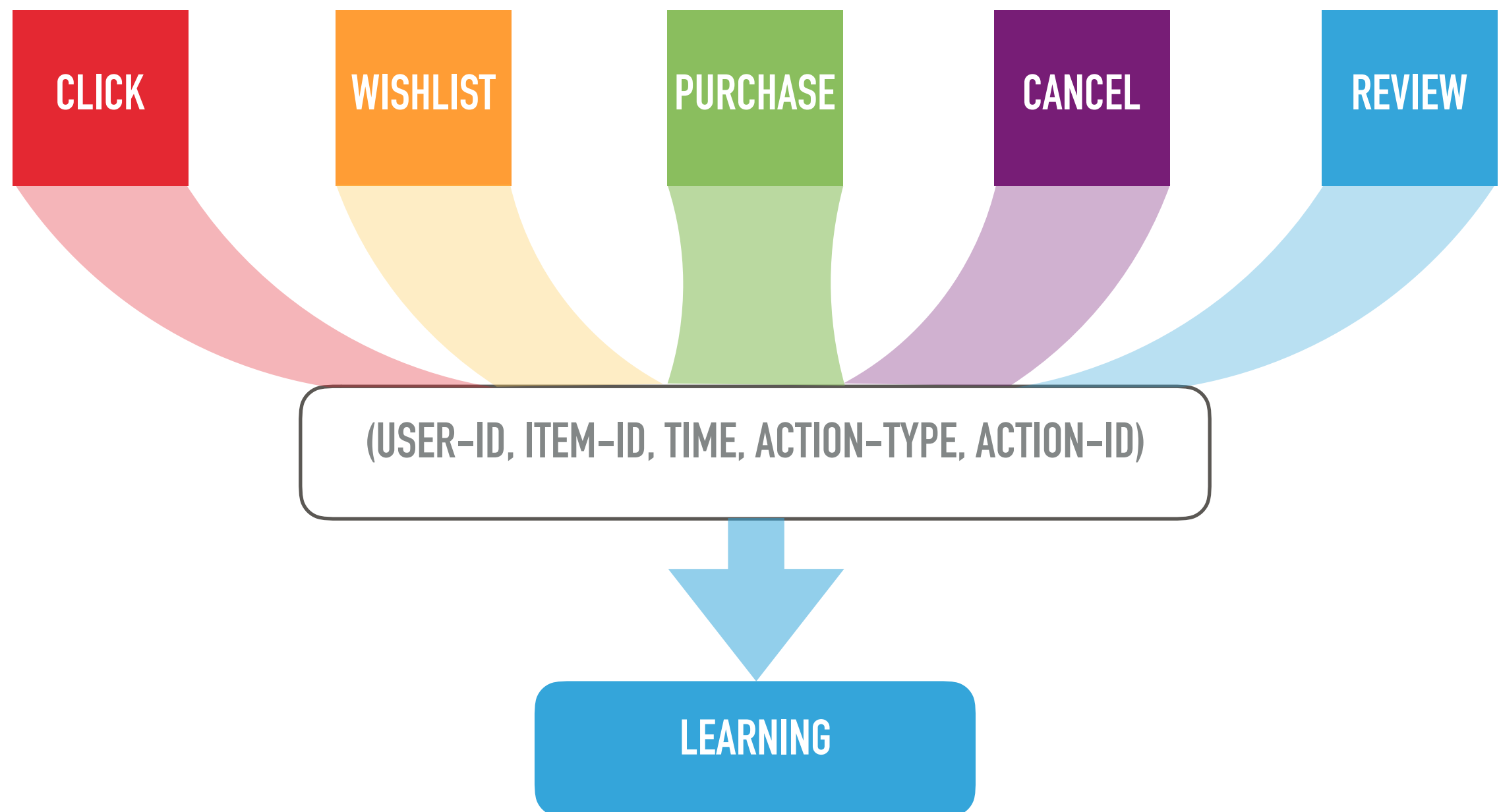
s3: Filter by area

s4: Filter by star rating

## DATA SOURCES

Action stream *(User Action)*

Light and effective, addresses most algorithm's needs



## EXPLICIT/IMPLICIT SIGNALS

- Hard to ask people to help
1. Explicit signal - a user review. ★★☆☆
  2. Implicit signal - item(s) viewed, clicked, songs listened to, movies watched, items added to favourites, added to shopping cart...  
→ Should use  $\{a_1, a_2, a_3, \dots\} \rightarrow \mathbb{R}^+$

Most papers suggest only positive re-enforcement while some cases suggest negative as well

Research focus is mostly on implicit - much more effective



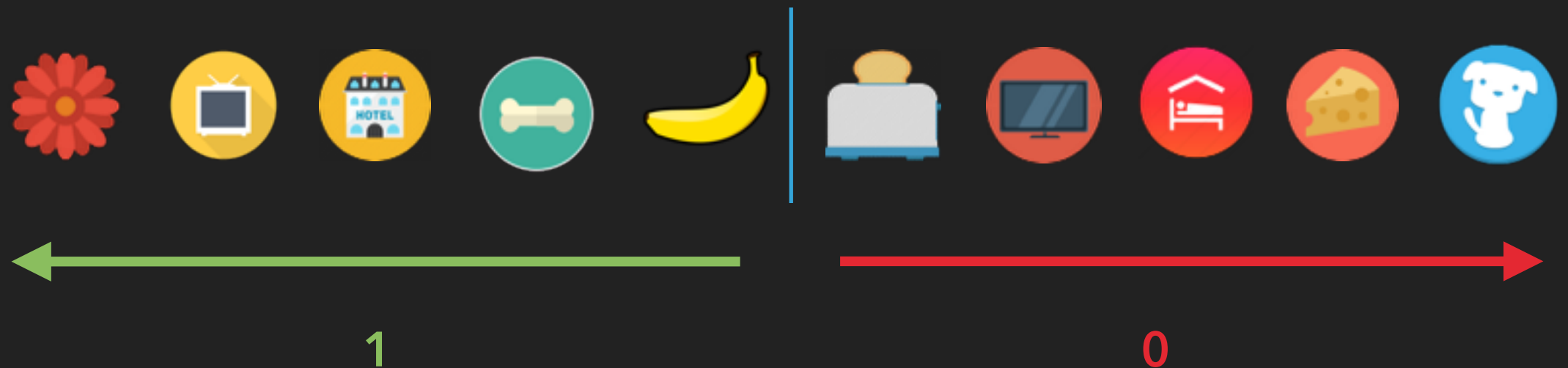
# ML FRAMEWORK

*\* Don't use RMSE : out of date \**

- ▶ forget about RMSE - focus on Information Retrieval metrics
- ▶ Since the response is a list, we are interested how high relevant items appear

# PRECISION @ K











"Precision at 5" corresponds to the number of relevant results on the top five recommendation



- ▶ Any relevance has an outcome of 0 or 1, not the best choice always
- ▶ Very simple to explain

# NORMALISED DISCOUNTED CUMULATIVE GAIN (NDCG)

$$NDCG@K(L) = \frac{DCG@K}{IDCG@K}$$

										
relevance		1		3	1					
gain		1		7	1					
discount	1	1.58	2	2.32	2.58					

$$DCG@k = 4.032$$

$$IDCG@k = 7 + \frac{1}{1.58} + \frac{1}{2} = 8.13$$

$$NDCG@k = 0.495$$

FINALLY

---

ALGORITHMS



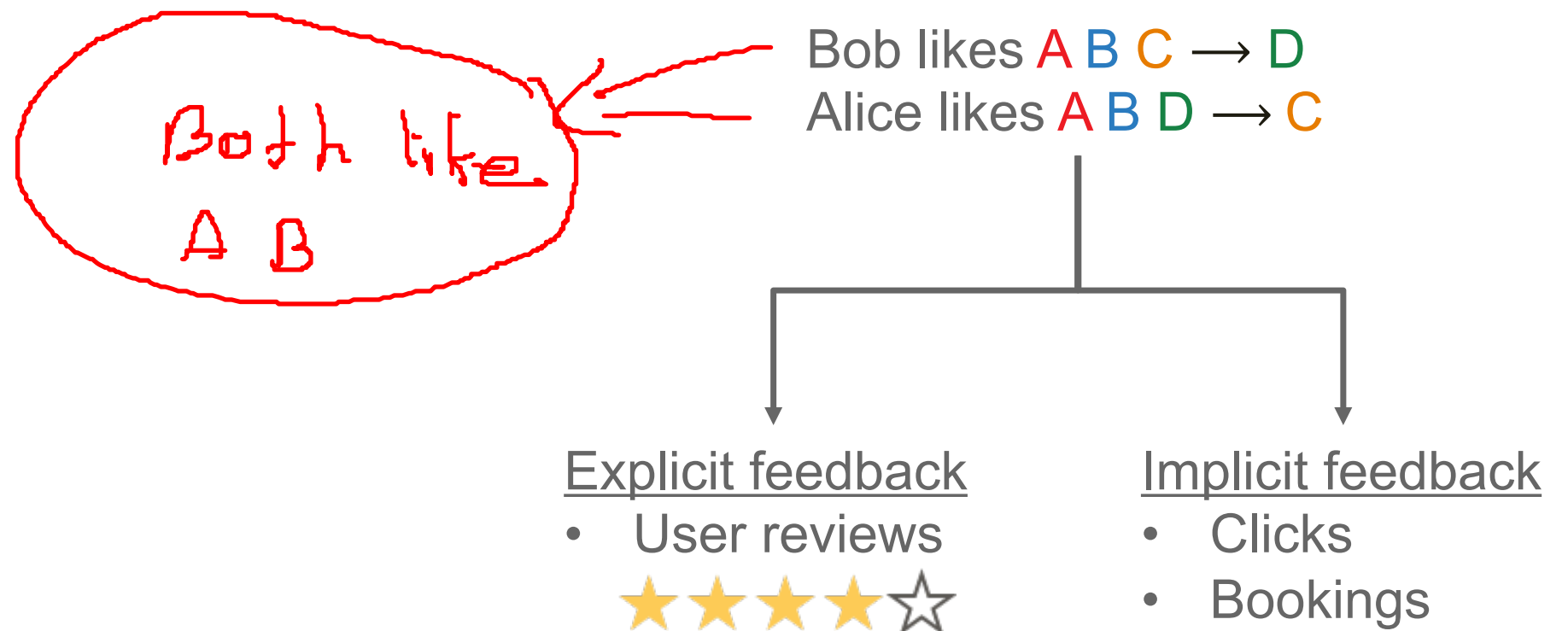
## APPROACH TO RECOMMENDATION

### Content based

- Predicts a user preference by analyzing the content of items the user has liked in the past

### Collaborative filtering

- Predicts a user preference by collecting taste information from other users (similar users)



# USER ITEM MATRIX

From a collection of users  $u=1,\dots,n$  and collection of items  $i=1,\dots,m$ , we can form a  $n\times m$  matrix of user preferences or ratings  $r_{ui}$

(John, Marriott, 7.8)  
(John, Lebua, 8.3)  
(Jane, Marriott, 6.7)  
(Jane, Bayioke, 6.0)  
(Adam, Bayioke, 5.5)  
(Adam, Shangri-La, 9.2)  
(Adam, Peninsula, 9.6)  
(Dean, Shangri-La, 9.5)  
(Dean, Movenpick, 8.8)  
:  
:  
:  
:

	Peninsula	Bayioke	Lebua	Marriott	Movenpick	Shangri-La
John			8.3	7.8		
Jane		6		5.5		
Adam	9.6	10				9.2
Dean					8.8	9.5
Aaron	2	5.5	3	3		
Lynda			2.2	7.4	7.6	3

# USER ITEM MATRIX

How will **Jane** rate the Lebua?

find some people like Jane  
- find neighbourhood  
or  
- Matrix factorization

	Peninsula	Bayioke	Lebua	Marriott	Movenpick	Shangri-La
John			8.3	7.8		
Jane		6	??	5.5		
Adam	9.6	10				9.2
Dean					8.8	9.5
Aaron	2	5.5	3	3		
Lynda			2.2	7.4	7.6	3

# USER BASED RECOMMENDATION

By knowing which users are similar to Jane we can compute 'neighbourhood' averages

$$sim(u,u') = \frac{\sum_i r_{ui}r_{u'i}}{\sqrt{\sum_i r_{ui}^2 \cdot \sum_i r_{u'i}^2}}$$

	Peninsula	Bayioke	Lebua	Marriott	Movenpick	Shangri-La	sim(jane,u)
John			8.3	7.8			0.462747565
Jane		6	??	5.5			1
Adam	9.6	10				9.2	0.44307298
Dean					8.8	9.5	0
Aaron	2	5.5	3	3			0.841335298
Lynda			2.2	7.4	7.6	3	0.444832653

# USER BASED RECOMMENDATION

Average top N members, assume N=2

$$\hat{r}_{uj} = \frac{1}{N} \sum_{u' \in S_N(U)} r_{u'i} = \frac{8.3 + 3}{2} = 5.65$$

	Peninsula	Bayioke	Lebua	Marriott	Movenpick	Shangri-La	sim(jane,u)
John			8.3	7.8			0.462747565
Jane		6	5.65	5.5			1
Adam	9.6	10				9.2	0.44307298
Dean					8.8	9.5	0
Aaron	2	5.5	3	3			0.841335298
Lynda			2.2	7.4	7.6	3	0.444832653



# USER BASED RECOMMENDATION

Average top N members, assume N=2, weight by similarity

$$\hat{r}_{uj} = K \sum_{u' \in S_N(U)} sim(u, u') r_{u'i} = 0.766 \cdot (8.3 \cdot 0.462 + 3 \cdot 0.841) = 4.88$$

$$K = 1 / \sum_{u' \in S_N(U)} |sim(u, u')|$$

	Peninsula	Bayioke	Lebua	Marriott	Movenpick	Shangri-La	sim(jane,u)
John			8.3	7.8			0.462747565
Jane		6	4.88	5.5			1
Adam	9.6	10				9.2	0.44307298
Dean					8.8	9.5	0
Aaron	2	5.5	3	3			0.841335298
Lynda			2.2	7.4	7.6	3	0.444832653

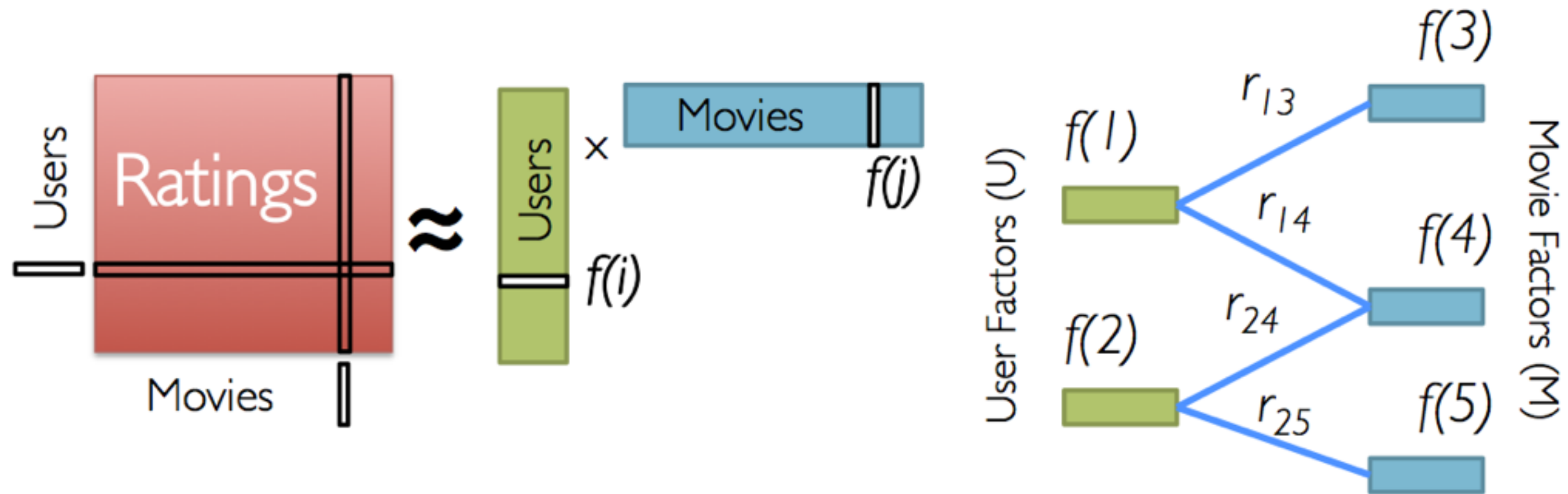
# MATRIX FACTORIZATION

Factorizing the matrix, by embedding the users and item in a latent space of arbitrary dimension  $x_u, y_i \in \mathbb{R}^d$

- (John, Marriott, 7.8)
- (John, Lebua, 8.3)
- (Jane, Marriott, 6.7)
- (Jane, Bayioke, 6.0)
- (Adam, Bayioke, 5.5)
- (Adam, Shangri-La, 9.2)
- (Adam, Peninsula, 9.6)
- (Dean, Shangri-La, 9.5)
- (Dean, Movenpick, 8.8)
- :
- :
- :

	Peninsula	Bayioke	Lebua	Marriott	Movenpick	Shangri-La
John			8.3	7.8		
Jane		6.0		5.5		
Adam	9.6	5.5				9.2
Dean					8.8	9.5

## MATRIX FACTORIZATION



Iterate:

$$f[i] = \arg \min_{w \in \mathbb{R}^d} \sum_{j \in \text{Nbrs}(i)} (r_{ij} - w^T f[j])^2 + \lambda ||w||_2^2$$

# MATRIX FACTORIZATION

We can then fill the whole matrix with our prediction

(John, Marriott, 7.8)  
(John, Lebua, 8.3)  
(Jane, Marriott, 6.7)  
(Jane, Bayioke, 6.0)  
(Adam, Bayioke, 5.5)  
(Adam, Shangri-La, 9.2)  
(Adam, Peninsula, 9.6)  
(Dean, Shangri-La, 9.5)  
(Dean, Movenpick, 8.8)  
:  
:  
:  
:

	Peninsula	Bayioke	Lebua	Marriott	Movenpick	Shangri-La
John	9.1	5.8	8.3	7.8	8.1	9.0
Jane	9.2	6.0	8.0	5.5	8.3	9.1
Adam	9.6	5.5	8.2	6.5	8.7	9.2
Dean	9.5	6.5	8.4	6.5	8.8	9.5

# MATRIX FACTORIZATION – IMPLICIT FEEDBACK

Much more powerful implementation using implicit feedback, has more examples and is less prone to rating biases.

It is costly to find the right weights for different action types

- (John, Peninsula, click)
- (John, Bayioke, click)
- (John, Lebua, click)
- (John, Marriott, book)
- (Jane, Lebua, click)
- (Jane, Movenpick, click)
- (Adam, Bayioke, book)
- (Adam, Shangri-La, book)
- (Dean, Shangri-La, book)
- :
- :
- :
- :

	Peninsula	Bayioke	Lebua	Marriott	Movenpick	Shangri-La
John	C	C	C	B		
Jane			C		C	
Adam		B				B
Dean						B

$$p_{u,i} = \begin{cases} 1 & r_{u,i} > 0 \\ 0 & r_{u,i} = 0 \end{cases}$$

$$w_{u,i} = 1 + \alpha \cdot r_{u,i}$$

$$\min \sum_{u,i} w_{u,i} (x'_u y_i - p_{u,i})^2 + reg(x'_u y_i)$$

## FACTORIZATION MACHINES

- ▶ A 2nd order Factorization Machine is a predictor that learns all 2nd order interactions between all  $n$  features.
- ▶ If there is an symmetric weight interaction matrix  $\mathbf{Z}$ , we try to approximate it with a low rank factorization  $\mathbf{V}$ .
- ▶ Some mathematical tricks allow FM to be solved in linear time
- ▶ This is highly effective since we are interested more in interactions between user and item than user or item features

$$\hat{y}(x) = w_0 + \mathbf{w}'x + \sum_{i=1}^n \sum_{j=i+1}^n z_{i,j} x_i x_j$$

$$\theta = (w_0, \mathbf{w}, \mathbf{Z}); \quad w_0 \in \mathbb{R}, \quad \mathbf{w} \in \mathbb{R}^n, \quad \mathbf{Z} \in \mathbb{S}^n$$

$$|\theta| = 1 + n + \binom{n}{2}$$

---

$$\mathbf{V} \in \mathbb{R}^{n \times k}, \quad k \ll n$$

$$\mathbf{V}\mathbf{V}^T = \mathbf{Z}$$

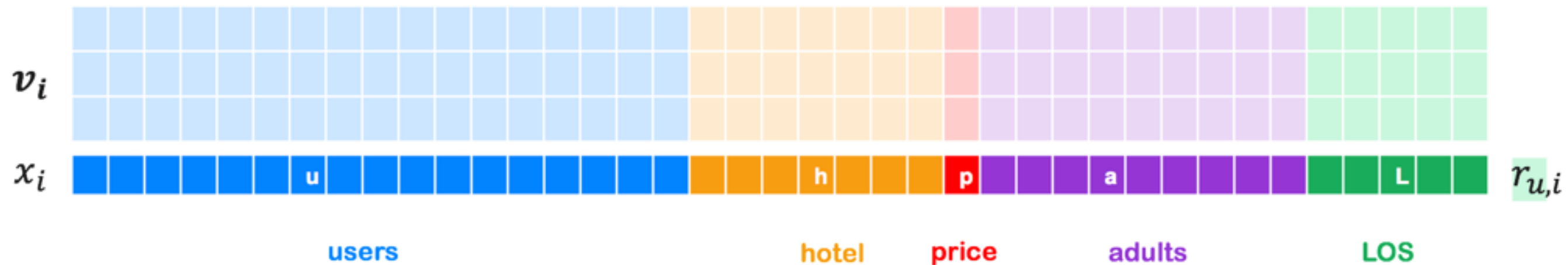
$$\hat{y}(x) = w_0 + \mathbf{w}'x + \sum_{i=1}^n \sum_{j=i+1}^n \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j$$

$$|\theta_{FM}| = 1 + (n + 1)k$$

\*see Rendle 2010 - Factorisation Machines



# FACTORIZATION MACHINES – CONTEXT



$$\hat{y} = \langle \mathbf{v}_u, \mathbf{v}_h \rangle + \langle \mathbf{v}_u, \mathbf{v}_p \rangle + \langle \mathbf{v}_u, \mathbf{v}_a \rangle + \langle \mathbf{v}_u, \mathbf{v}_L \rangle +$$

$$\langle \mathbf{v}_h, \mathbf{v}_p \rangle + \langle \mathbf{v}_h, \mathbf{v}_a \rangle + \langle \mathbf{v}_h, \mathbf{v}_L \rangle +$$

$$\langle \mathbf{v}_p, \mathbf{v}_a \rangle + \langle \mathbf{v}_p, \mathbf{v}_L \rangle +$$

$$\langle \mathbf{v}_a, \mathbf{v}_L \rangle +$$

## ITEM – ITEM CO-OCCURRENCE

Similar item

Session A



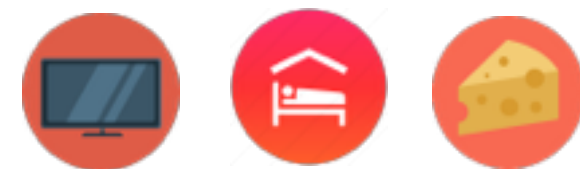
Session B



Session C



Session E



$$s(item_i, item_j) = \frac{c_{ij}}{f(item_i, item_j)}$$

Where  $c_{ij}$  is co-occurrence count and  $c_i$  and  $c_j$  are the total occurrence counts

$f(item_i, item_j)$  is a normalization function that takes the “global popularity” into account. the simplest normalization functions is to simply divide by the product of the items’ global popularity:  $f(item_i, item_j) = c_i \cdot c_j$

# ITEM – ITEM CO-OCCURRENCE

### Advantage

- ▶ Simple, easy to implement and yet very effective

### Drawback

- ▶ Prone to overfitting (consider a minimum threshold or a constant regularisation)
- ▶ Not personalised
- ▶ Cannot add more features !!!

# ITEM2VEC

**Word2vec (skip gram):** The method aims at finding word representations that captures the relation between a word to its surrounding words in a sentence.

Given a corpus of words  $\mathbf{w}$  and their contexts  $\mathbf{c}$

The goal is to set parameters  $\theta$  so as to maximize the corpus probability

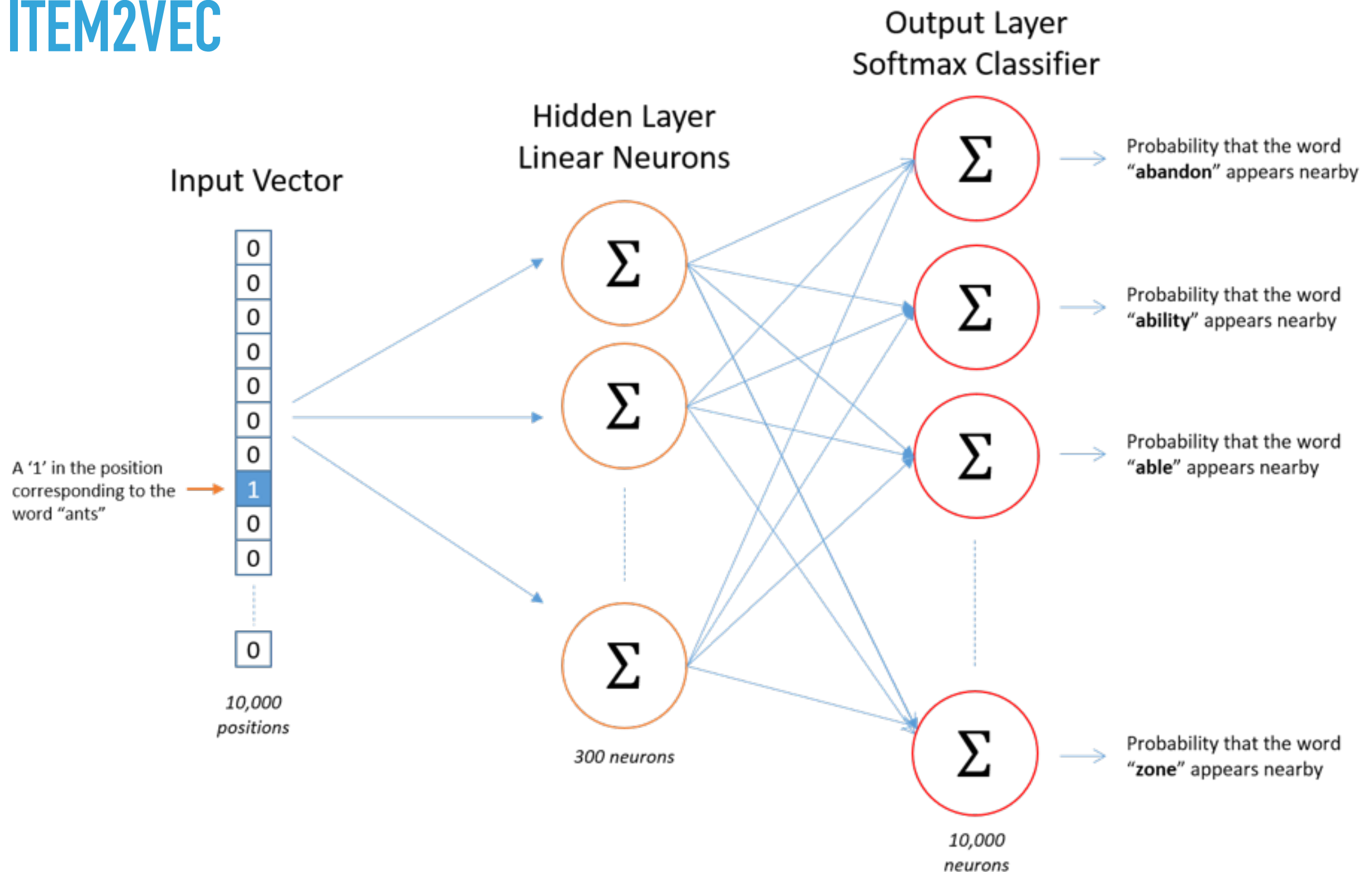
$$\arg \max_{\theta} \prod_{(\mathbf{w}, \mathbf{c}) \in D} p(\mathbf{c}|\mathbf{w}; \theta)$$

D is the set of all word and context pairs we extract from text and

$$p(\mathbf{c}|\mathbf{w}; \theta) = \frac{e^{\mathbf{v}_{\mathbf{c}} \cdot \mathbf{v}_{\mathbf{w}}}}{\sum_{\mathbf{c}' \in \mathbf{C}} e^{\mathbf{v}_{\mathbf{c}'} \cdot \mathbf{v}_{\mathbf{w}}}}$$

$\mathbf{v}_{\mathbf{c}}$  and  $\mathbf{v}_{\mathbf{w}}$  are vector representations for  $\mathbf{c}$  and  $\mathbf{w}$  respectively, and  $\mathbf{C}$  is the set of all available contexts

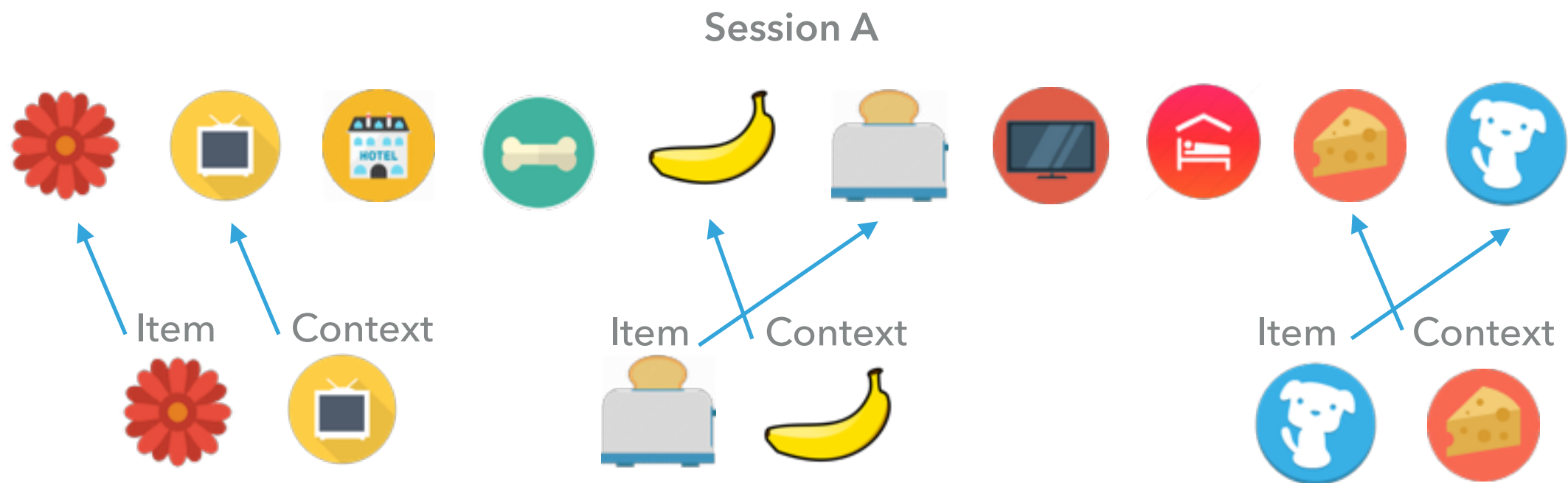
## ITEM2VEC



## ITEM2VEC

What we need to do is to define "Context  $c$  of an Item  $i$ "

The simplest one is to extract pairs from session data

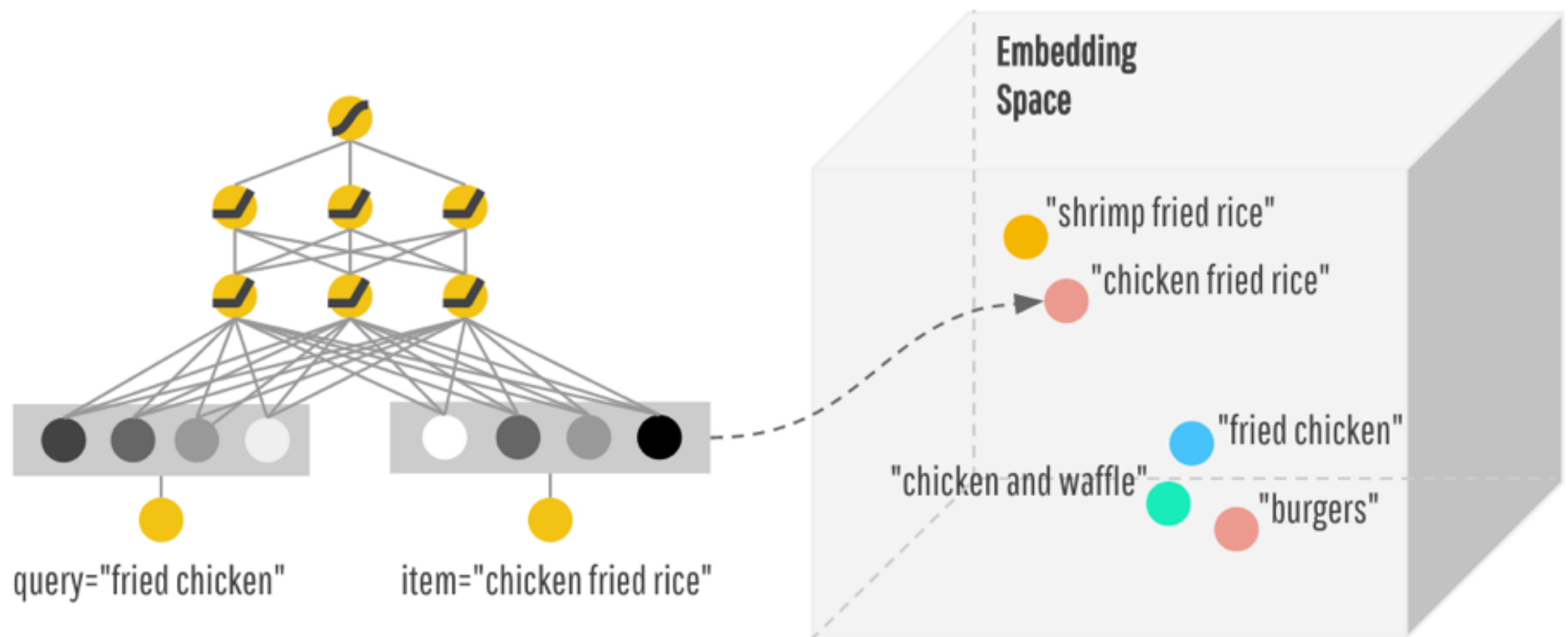




## DEEP MODEL

Embedding-based model (e.g. Factorisation Machine) has a main strength in its "**Generalization**" but it fails to capture some "exception rules" because it is too generalised

### Deep Model (Embedding-based model)

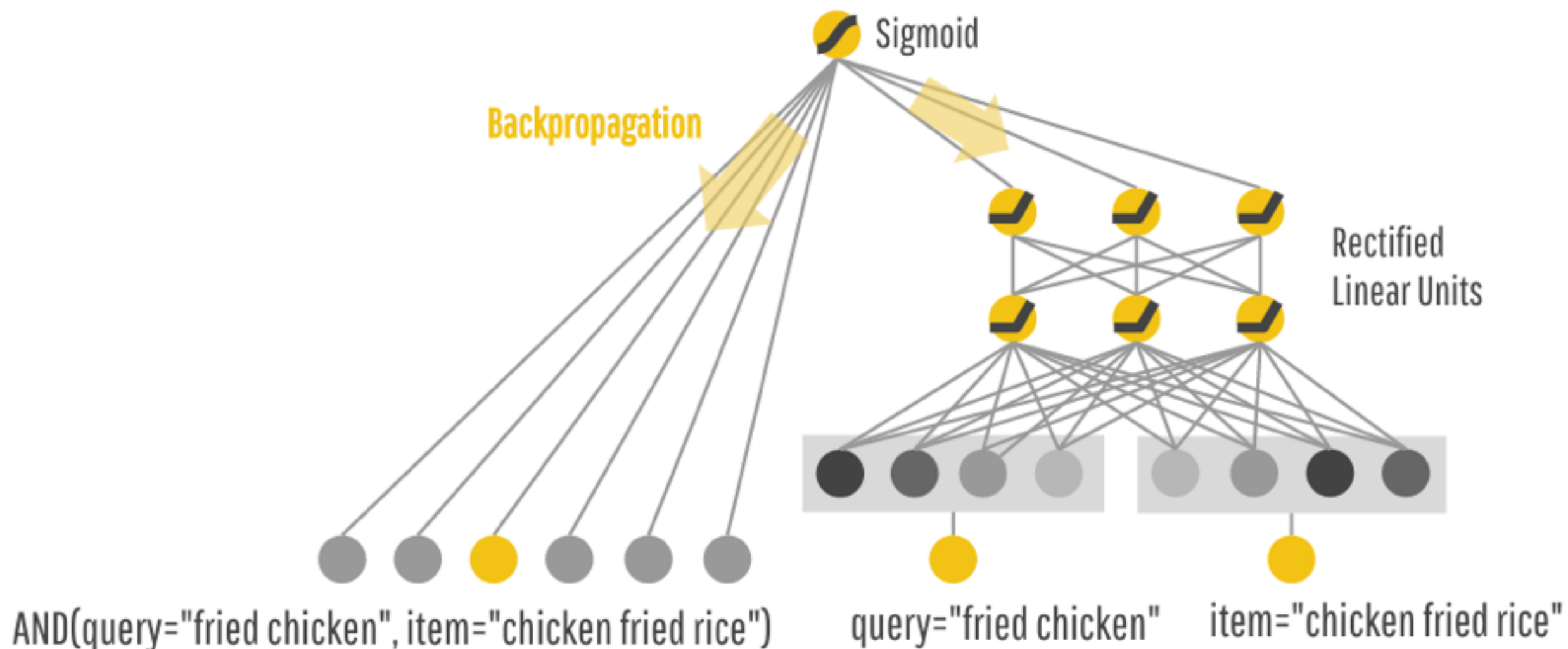


## WIDE AND DEEP LEARNING

Goal is to remember special event

The idea is to combine "wide" model with "deep model" so that the model can learn to "Generalize" and "Memorize" at the same time.

### Wide and Deep Model



## RECOMMENDED: MATRIX FACTORIZATION – IMPLICIT FEEDBACK

<https://gist.github.com/jbochi/2e8ddcc5939e70e5368326aa034a144e>