DIGINETICA
Retail Techology Company

Personalized E-commerce Search Challenge

epam Nikitko Dmitrii

8 October 2016

Data

- Anonymized user activity (purchases, views, clicks)
- Search engine queries, hashed query terms, category id
- Hashed product descriptions and meta-data

Dataset Statistics

- The number of queries: 923 127
- The number of unique sessions: 573 935
- The number of products: 184 047

Sample

queryId	sessionId	userId	timeframe	duration	Searchstring tokens	categoryId	items	is.test
43491	33900	NaN	3766699	451	42423,37748, 2886,215391	0	*91816,81796, 8517,72561...*	False
226872	164307	NaN	0	1638	NaN	571	*109677,3813 1,72578,...*	False
681772	50411	NaN	345234	2863	NaN	1009	*8392,206727, 314260,...*	True

Reranking

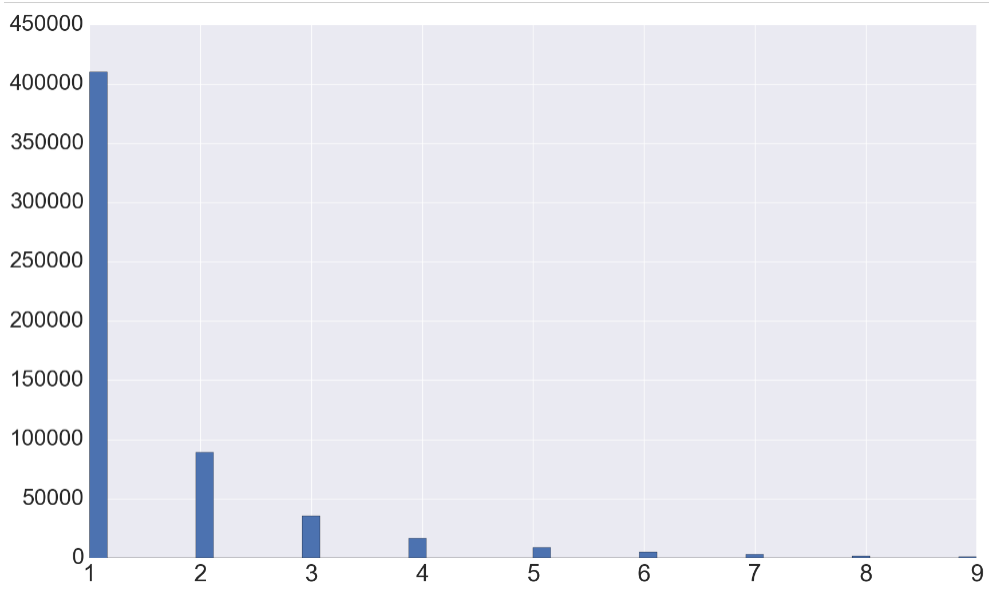
Metric

NDCG (Normalized Discounted Cumulative Gain)

$$\mathrm{nDCG_p} = \frac{DCG_p}{IDCG_p}, \qquad \mathrm{DCG_p} = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

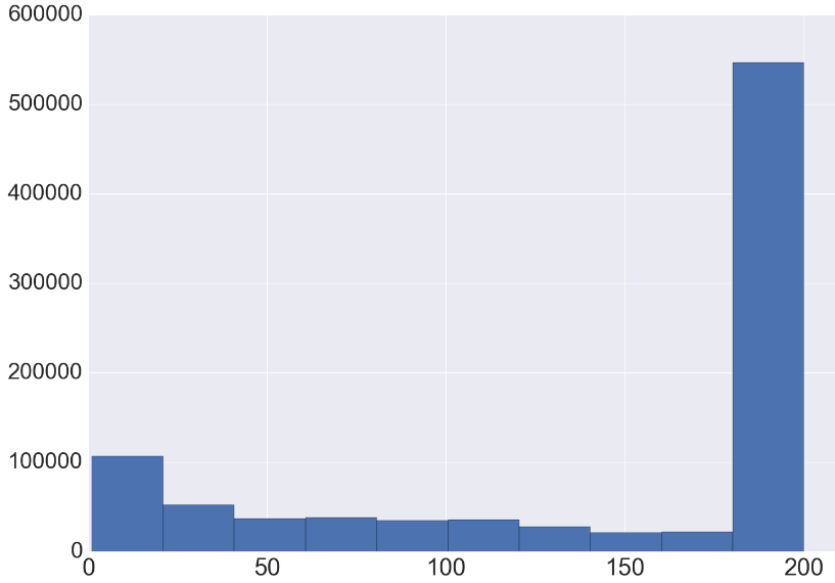
- Highly relevant documents are more useful than marginally relevant documents, which are in turn more useful than irrelevant documents.
- Relevance value is reduced logarithmically proportional to the position of the result

Queries per session



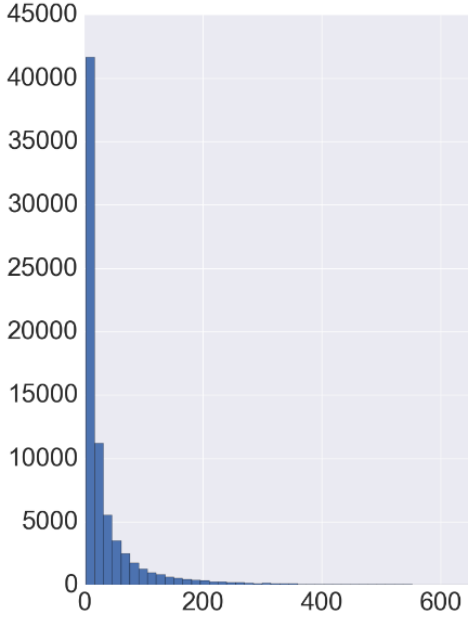
- 75% sessions have only one query

Items per query



- 83% non-textual sessions with large search results (items)

Products views



- Long tail

Recover item relevance for every query

- Purchases have session information, timeframe and item id.
- Clicks information provide a query id, timeframe, item id.
- Combine it! Its improve query-level relevance information.

Query					
Item id	4355	2344	34534	34553	…
Relevance	0	1	0	3	…

Sparse 184047-D vector

Possible solutions

Pointwise regression

- Linear SGD regression with L1 regularization and sparsify after fit.
- For every item train own model only on positive items, for negative sampling choose random query.
- Gaussian random projection for reduce the dimensionality

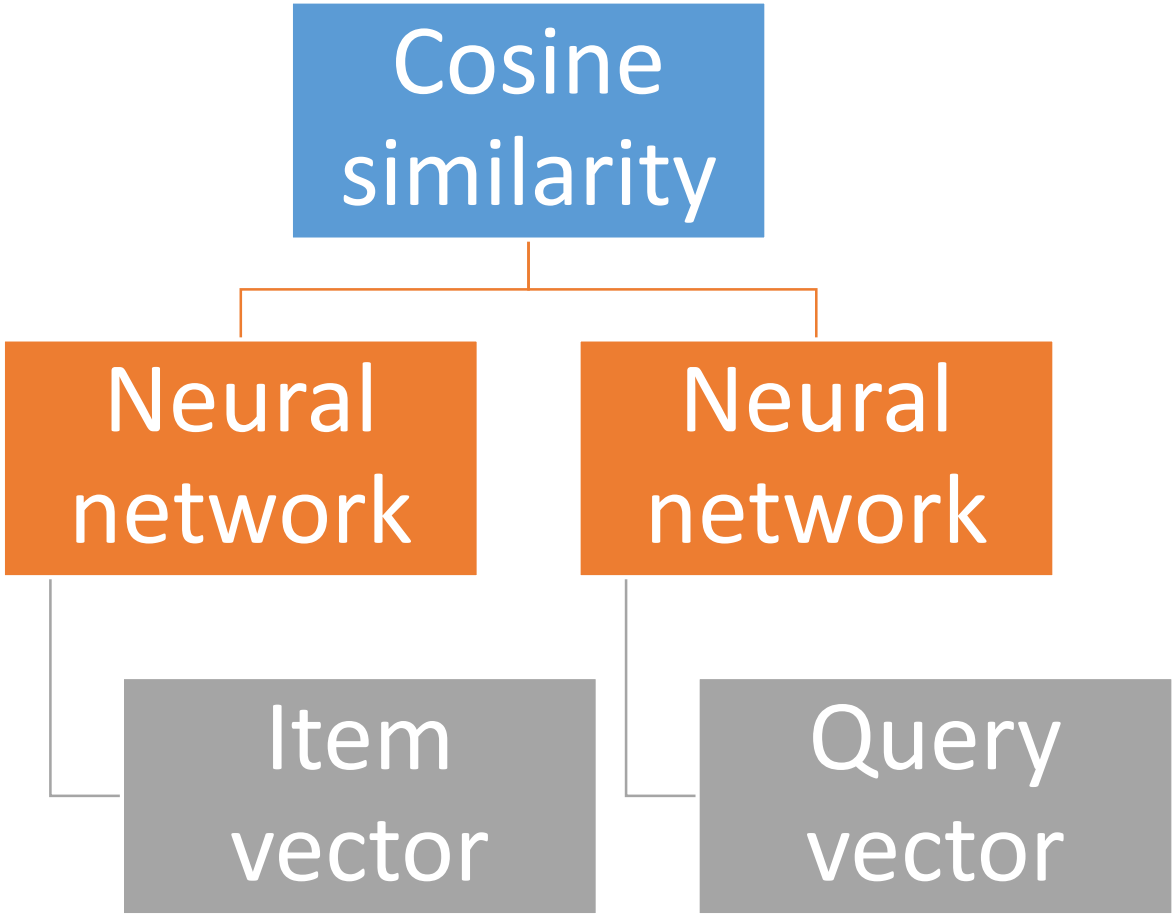
Query similarity

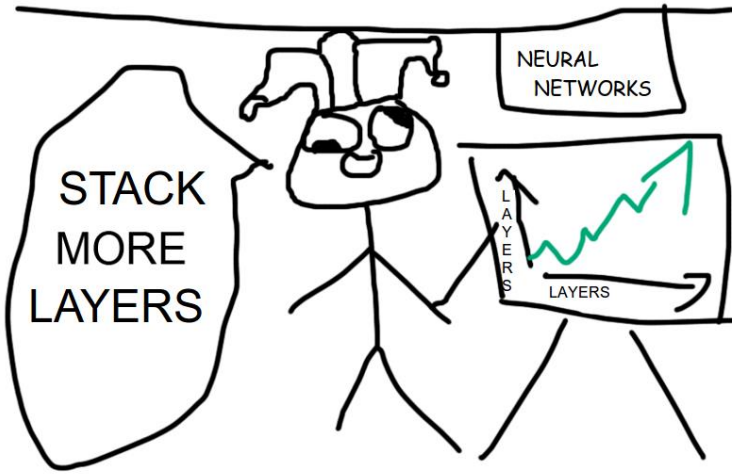
- Locality Sensitive Hashing MinHash (**Jaccard** similarity) (**datasketch**)
- Cosine similarity (fast **sklearn pairwise_distances**)
- Split items by category

Personalized recommendations

- Group queries by session => user profile with items relevance
- Now we can use collaborative filtering technics
  - Matrix factorization
    - SVD
    - NMF
    - Implicit ALS
  - Pairwise loss (BPR, WARP) (**LightFM**)
  - Factorization Machines

Neural networks similarity learning





Results

RESULTS

	User	Team Name	FinalNDCG (weighted average)	SearchNDCG (query-full; textual queries)	CategoryNDCG (query-less; category facets)
1	minerva	Ali-Search	0.4262 (1)	0.5574 (1)	0.3935 (1)
2	Dmitrii_Nikitko		0.4149 (2)	0.5301 (2)	0.3861 (3)
3	joaopalotti		0.3712 (5)	0.4860 (3)	0.3425 (5)
4	wistuba		0.3769 (4)	0.4495 (4)	0.3588 (4)
5	tjy	red fruit yard	0.4015 (3)	0.4364 (5)	0.3928 (2)

Tricks

- Cut-off long tail – use only most popular items
- Write extension methods and classes
- Logging experiments with code backups
- Use already implemented solutions and algorithms

DIGINETICA
Retail Techology Company

Personalized E-commerce Search Challenge

epam Nikitko Dmitrii

8 October 2016