

Quality assessment

Stanislav Protasov

Agenda

Metrics:

- Offline
- Online
- Other aspects

Q: what is the target metric of the service?

What is the target metric of a service?

- Money
- Traffic share (yandex vs google)
- User satisfaction
- User happiness
- Logins/Subscriptions
- ...

Unfortunately, we cannot re-run ML models with complex human targets (e.g. *if we had this feature, we would attract \$\$*)

What we can?

Create a set of easily **computable** metrics which correlate with target ones.

Satisfaction vs *probability to find relevant doc*

Traffic Share vs *MAU*

High-level evaluation techniques

Offline evaluation — result of a new model is compared on *manually preprocessed data*

- On a **bucket**: accuracy, precision, recall
- By assessors: relevance scale
 - (assessors are humans: **kappa stats**, weighted, majority voting, ...)

Online evaluation — a new model is *compared* with an old one by some *target metric* on a subset of users. Most widely used approach is **A/B testing**



BTW, what is *relevance*?

Information need is encoded in a **query**.



Query is used to get **results**.

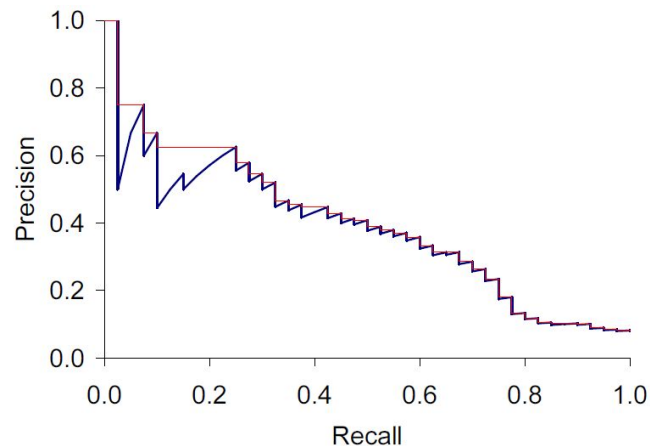
But do results **satisfy information need**?

{0, 1} or [0..1]

Offline: accuracy, precision, recall, F_1

- **Accuracy** is not the case
- **Precision** is how many relevant out of retrieved
- **Recall** is how many relevant retrieved out of all relevant
- **Precision-recall curve** can be used for ranked results
- F_1 is to come up with a single number

Users are *tolerant* to irrelevant results



Mean Average Precision

Precision@K == TruePositive / K

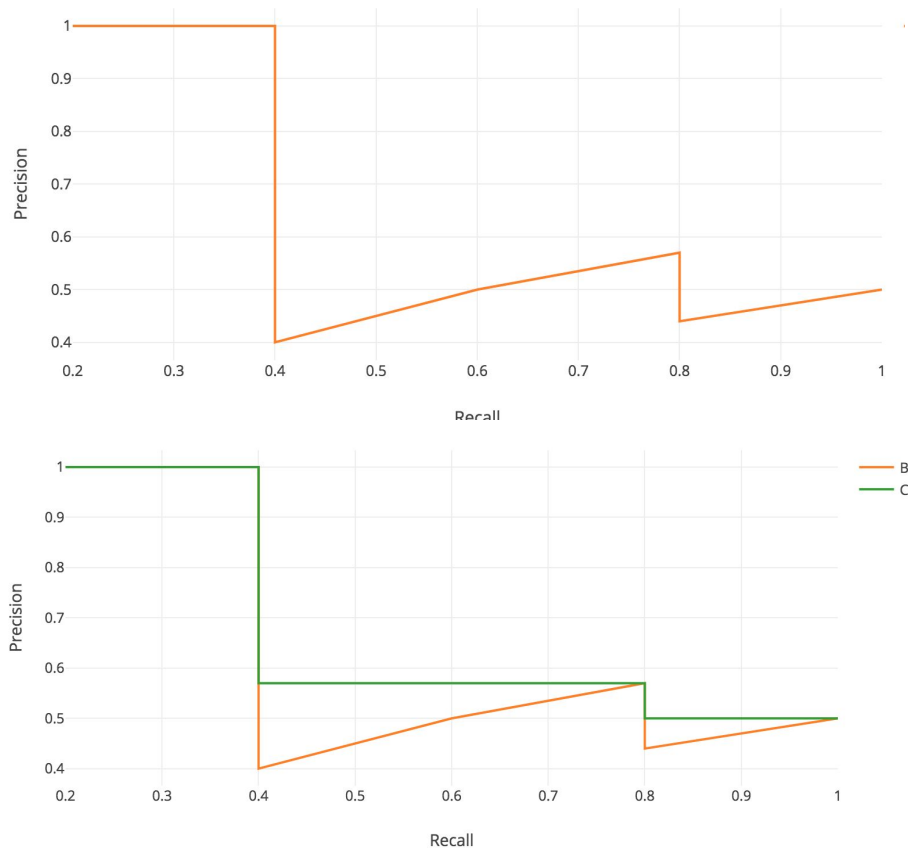
Recall* == TruePositive / AllPositive

Average Precision: $AP = \int_0^1 p(r)dr$

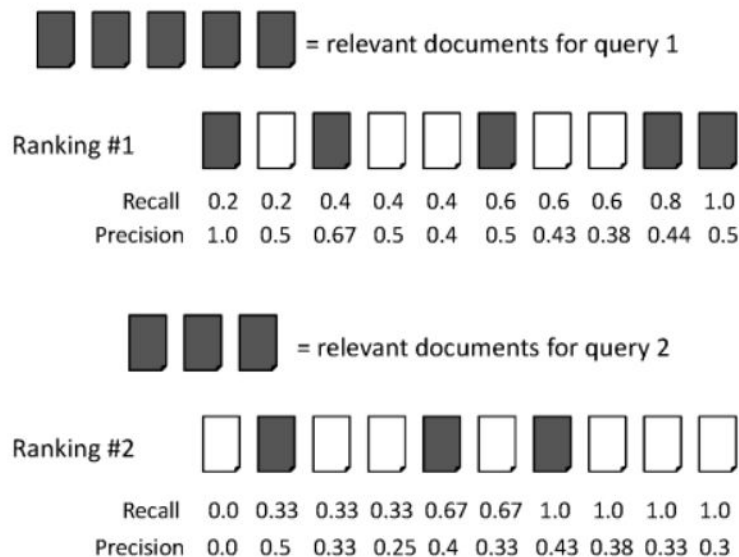
1. Improve “steps”

2. AP@n $AveP = \sum_{k=1}^n P(k) \Delta r(k)$

$$MAP = \frac{\sum_{q=1}^Q AveP(q)}{Q}$$



MAP



$$\text{average precision query 1} = (1.0 + 0.67 + 0.5 + 0.44 + 0.5)/5 = 0.62$$

$$\text{average precision query 2} = (0.5 + 0.4 + 0.43)/3 = 0.44$$

$$\text{mean average precision} = (0.62 + 0.44)/2 = 0.53$$

$$\text{AveP} = \sum_{k=1}^n P(k) \Delta r(k)$$

Whiteboard time (MAP)

Q	<i>“vk”</i>	<i>“Good search engine”</i>	<i>“Fanny Animal Imajes”</i>	<i>“Who let the dogs out?”</i>
1	vk.com	Good engine repair	Images of Fanny Ardant	Baha Men - Who Let The Dogs Out (Youtube)
2	vkusvill	<i>yahoo.com</i>	9GAGs	CNN: old men let the dogs out to prevent robbery
3	МЛ	google.com	fishki.net	<i>Who Let The Dogs Out Wikipedia</i>
4	Some trash	yandex.ru	CNN.com	Funny Dogs website
5	<i>facebook.com</i>	<i>altavista</i>	<i>Moscow Zoo</i>	Baha Men - Who Let The Dogs Out (Lyrics)

Offline: simple is ok

Mean reciprocal rank (MRR):
$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i}.$$

Q — some queries

rank_i — rank of **first** relevant document

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i}.$$

Whiteboard time (MRR)

Q	“vk”	“Good search engine”	“Fanny Animal Imajes”	“Who let the dogs out?”
1	vk.com	Good engine repair	Images of Fanny Ardant	Baha Men - Who Let The Dogs Out (Youtube)
2	vkusvill	<i>yahoo.com</i>	9GAGs	CNN: old men let the dogs out to prevent robbery
3	МЛ	google.com	fishki.net	<i>Who Let The Dogs Out Wikipedia</i>
4	Some trash	yandex.ru	CNN.com	Funny Dogs website
5	<i>facebook.com</i>	<i>altavista</i>	<i>Moscow Zoo</i>	Baha Men - Who Let The Dogs Out (Lyrics)

Offline: discounted gain model

Cumulative gain CG@p: $CG_p = \sum_{i=1}^p rel_i$ (p — e.g. 10 items on SERP, rel_i - 0/1 or 0..1)

Discounted CG@p: $DCG_p = \sum_{i=1}^p \frac{rel_i}{\log_2(i+1)} = rel_1 + \sum_{i=2}^p \frac{rel_i}{\log_2(i+1)}$ $DCG_p = \sum_{i=1}^p \frac{2^{rel_i} - 1}{\log_2(i+1)}$

Normalized DCG: divide DCG by the best possible achievable (Ideal) DCG:

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

$$DCG_p = \sum_{i=1}^p \frac{rel_i}{\log_2(i+1)} = rel_1 + \sum_{i=2}^p \frac{rel_i}{\log_2(i+1)}$$

DCG example

	Relevance		Discounted Gain	
1	R	1	1	
2	R	1	0,63	
3	N	0	0	
4	N	0	0	
5	R	1	0,38	
6	R	1	0,35	
7	N	0	0	
8	R	1	0,31	
9	N	0	0	
10	N	0	0	
...	...			

$$\text{DCG}_p = \sum_{i=1}^p \frac{rel_i}{\log_2(i+1)} = rel_1 + \sum_{i=2}^p \frac{rel_i}{\log_2(i+1)}$$

* Whiteboard time (DCG)

Q	<i>“vk”</i>	<i>“Good search engine”</i>	<i>“Fanny Animal Imajes”</i>	<i>“Who let the dogs out?”</i>
1	vk.com	Good engine repair	Images of Fanny Ardant	Baha Men - Who Let The Dogs Out (Youtube)
2	vkusvill	<i>yahoo.com</i>	9GAGs	CNN: old men let the dogs out to prevent robbery
3	МЛ	google.com	fishki.net	<i>Who Let The Dogs Out Wikipedia</i>
4	Some trash	yandex.ru	CNN.com	Funny Dogs website
5	<i>facebook.com</i>	<i>altavista</i>	<i>Moscow Zoo</i>	Baha Men - Who Let The Dogs Out (Lyrics)

Offline: pFound

pFound idea is similar. Estimate probability to find relevant item.

Presets:

- **max relevance is 0.4** (R), otherwise is 0 (NR, maybe R);
- probability to quit with no reason **pBreak** - 0.15;
- User watches from top to bottom, stops if *found* relevant OR if *tired*;

$$pLook[i] = pLook[i-1] * (1 - pRel[i-1]) * (1 - pBreak)$$

$$pfound = \sum_{i=1}^n pLook[i] * pRel[i]$$

$$pLook[i] = pLook[i-1] * (1 - pRel[i-1]) * (1 - pBreak)$$

$$pfound = \sum_{i=1}^n pLook[i] * pRel[i]$$

Whiteboard time

Q	<i>“Fanny Animal Images”</i>
1	Images of Fanny Ardant
2	9GAGs
3	fishki.net
4	CNN.com
5	<i>Moscow Zoo</i>

$$pLook[1] = 1$$

$$pLook[2] =$$

$$pLook[3] =$$

$$pLook[4] =$$

$$pLook[5] =$$

$$pFound =$$

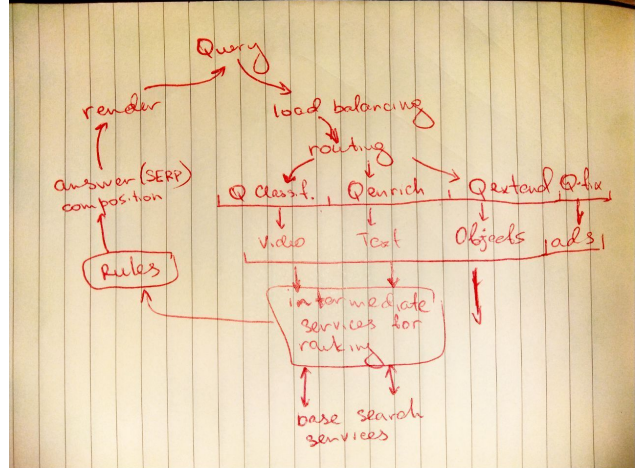
Online: A/B testing

Online testing allows to test of a *target metric*!

1. Is *relative*. Create **alternative** model
2. Formulate **target metric** (CTR, dwell-time, total revenue, ARPU)
3. Prepare **representative subset of users** (for complex setups use multinomial experiments)
4. Run controlled experiment, **aggregate statistics** on your metric
5. Run **statistical test** (Welch, Fisher, Student, ...) to see that **mean shift** for a distribution is significant*
6. Make a **decision** to accept (no always “better” for new features)

A/B testing infrastructure

- User **identification** method (logged in, cookies, fingerprints)
- **Experiment support** in the whole infrastructure (should not be as usual traffic) with flags/parameters
- Mapping **registry** (users to experiments)
- **Statistical tools** ready before you start experiment



Use a platform: Google Analytics, Optimizely, VWO, ...

Other quality aspects

New users should get most ranked items, whereas old users should be surprised with quality items from “long tail” [[ref](#)]

Add diversity to recommendations, get out of the bubble [[ref](#)]

Novelty issue (did I see this before?) [[ref](#)]

Stay legally and ethically safe (no medical questions, no porn, no swearing)

~~Search engine~~ One company's target

User should **solve a problem** with a service: get an answer to the question, a service or an item without leaving portal. Steps to achieve:

- Keep users *logged in*
- Evaluate user intent (surfing, buying, asking, ...)
- Provide a *quality service* for each intent (first, specific search, then specific service)
- Don't be evil

**DON'T BE
EVIL***

***Unless It's Profitable**

