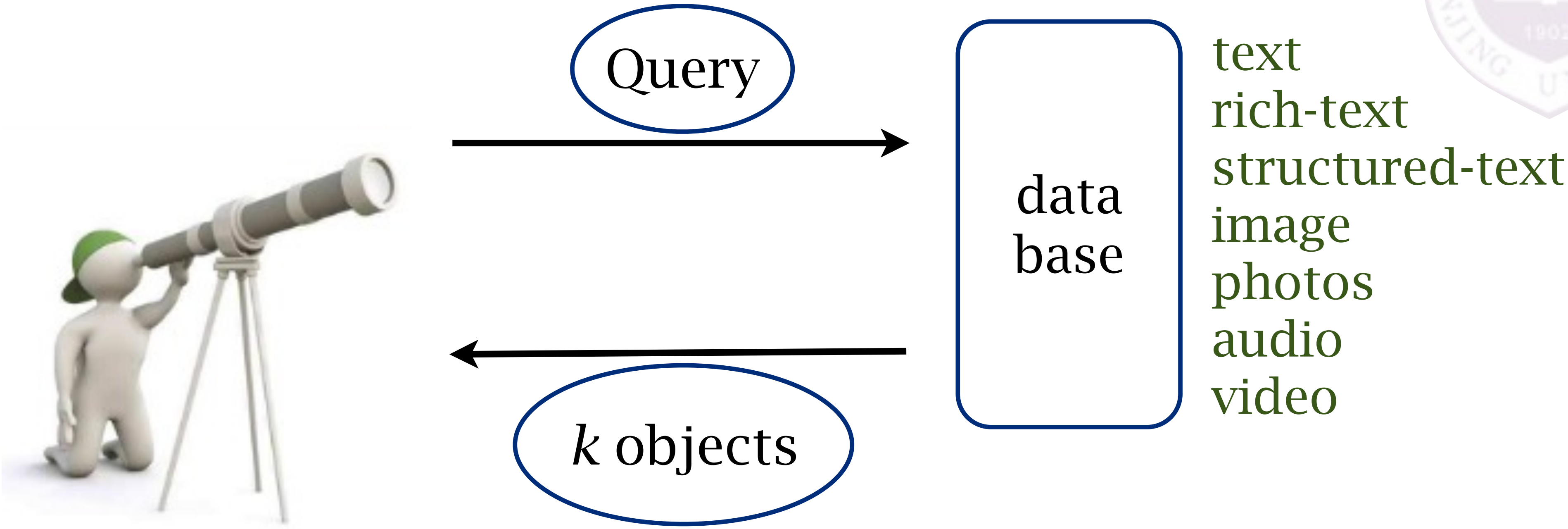
Lecture 13: Data Mining IV
Information Retrieval Systems

http://cs.nju.edu.cn/yuy/course_dm13ms.ashx

Information retrieval systems

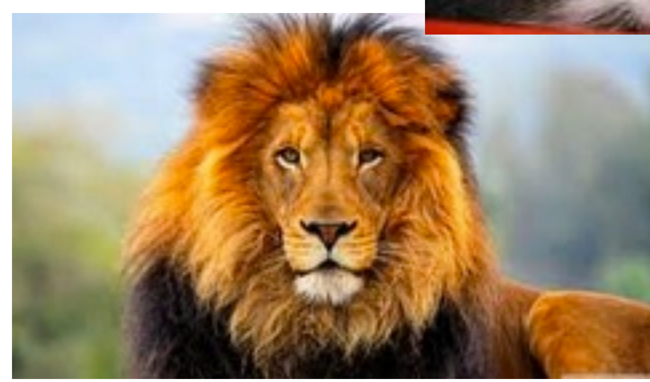


Content-based information retrieval:
for objects with rich semantics
find top *k* objects most similar to the query

- searching historical records of the Dow Jones index for past occurrences of a particular time series pattern
- searching a database of satellite images for any images which contain evidence of recent volcano eruptions in Central America
- searching the Internet for online documents that provide reviews of restaurants in Helsinki

Evaluation

how good is an retrieval system?



unlike classification where labels are given

Evaluation

how good is an retrieval system?

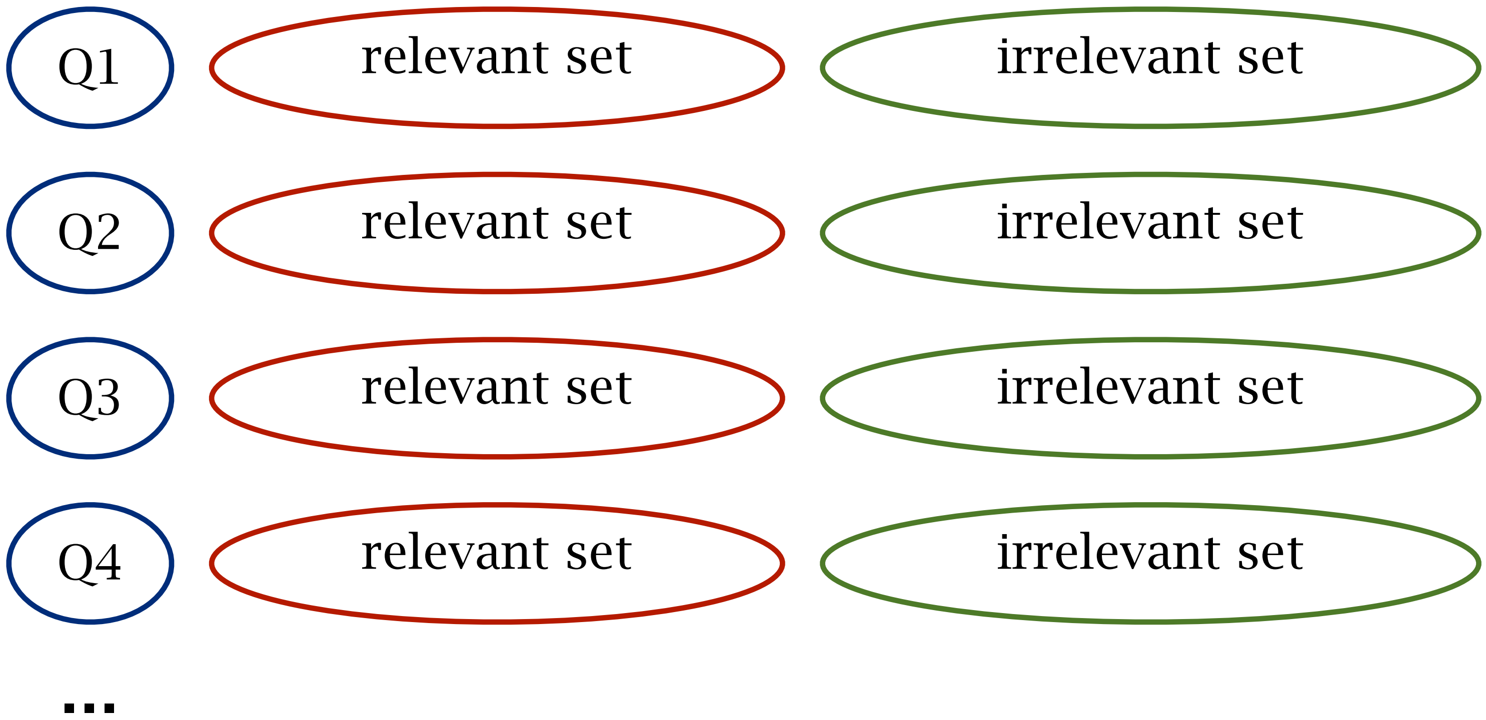
QUERY

N/R R

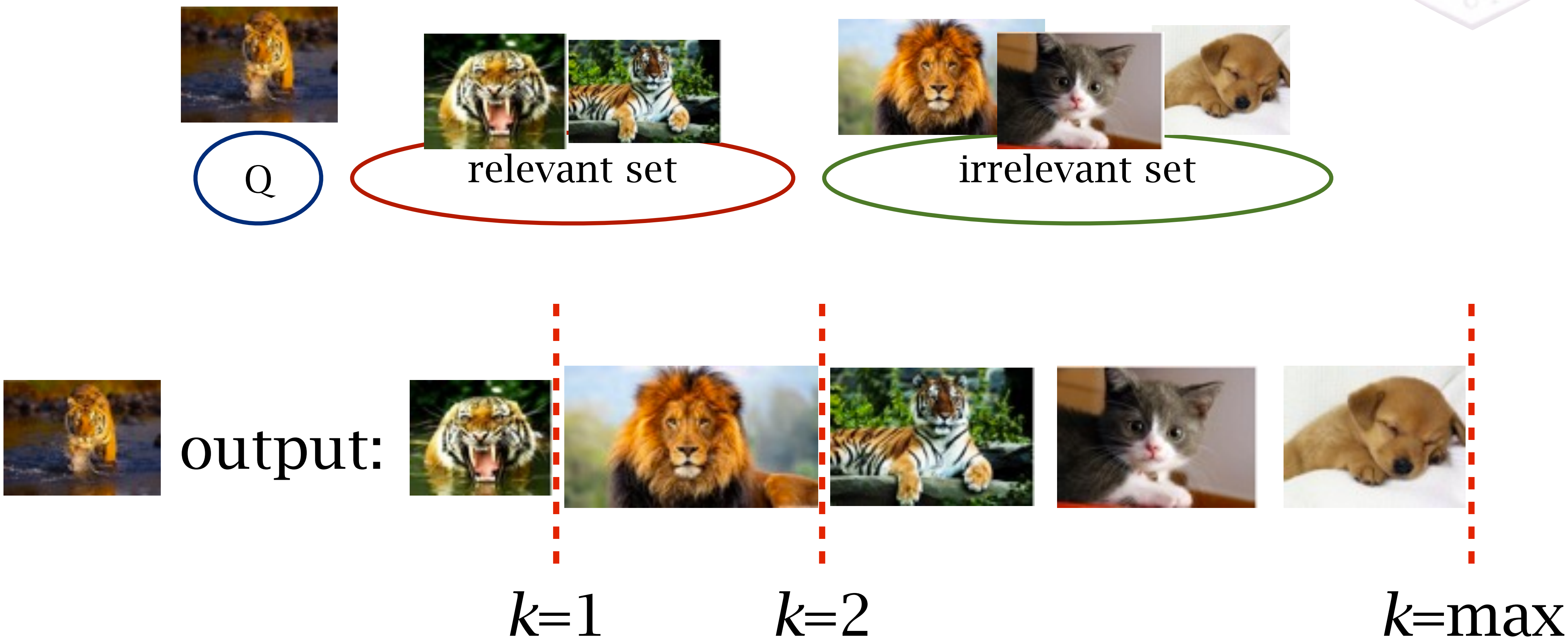
for a particular query, objects can be categorized into “relevant” and “irrelevant”

Evaluation

a set queries and pre-labeled relevant/ irrelevant objects



Configurable output

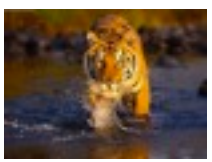
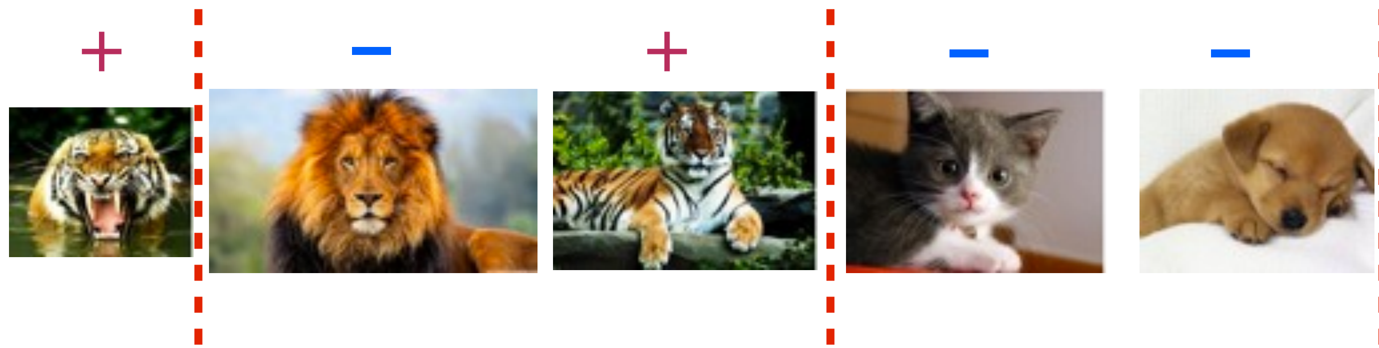


usually a retrieval system evaluates all objects and rank them according to the similarity

classification error?

Precision and recall

Precision: relevant outputs / all outputs
Recall: relevant outputs / all relevant objects

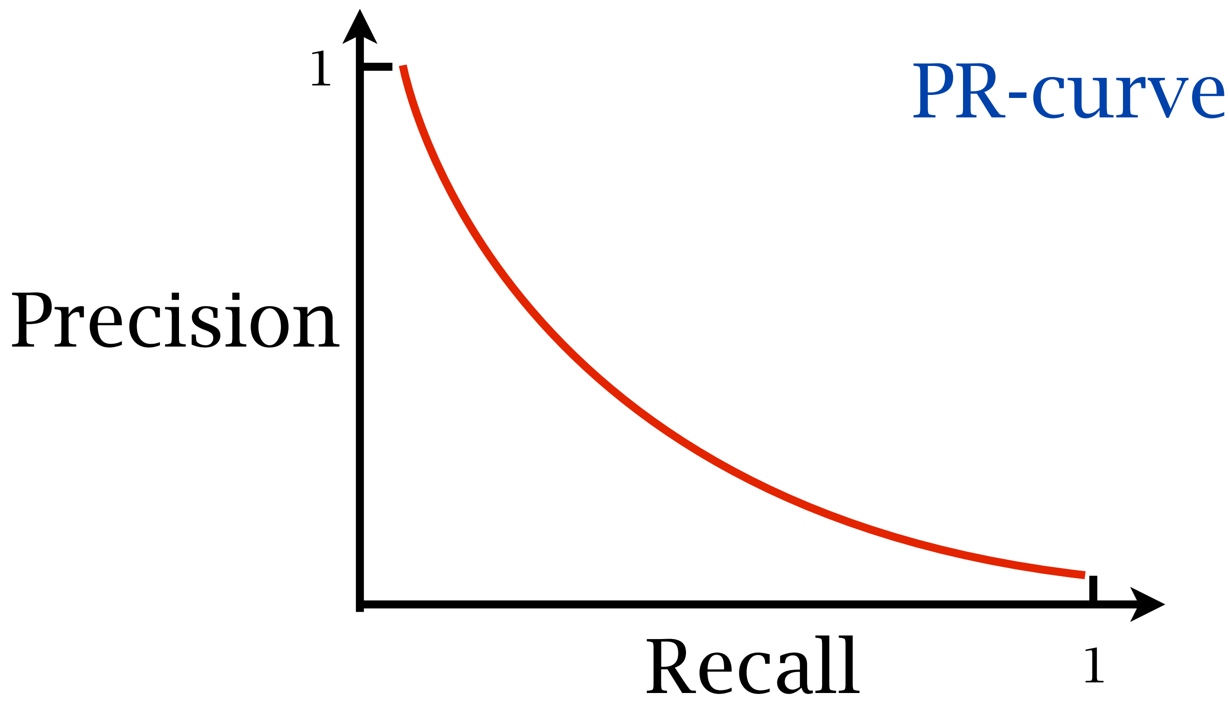
 output: 

k=1	k=3	k=max
P: 1	P: 0.66	P: 0.4
R: 0.5	R: 1	R: 1

should be averaged over all test queries

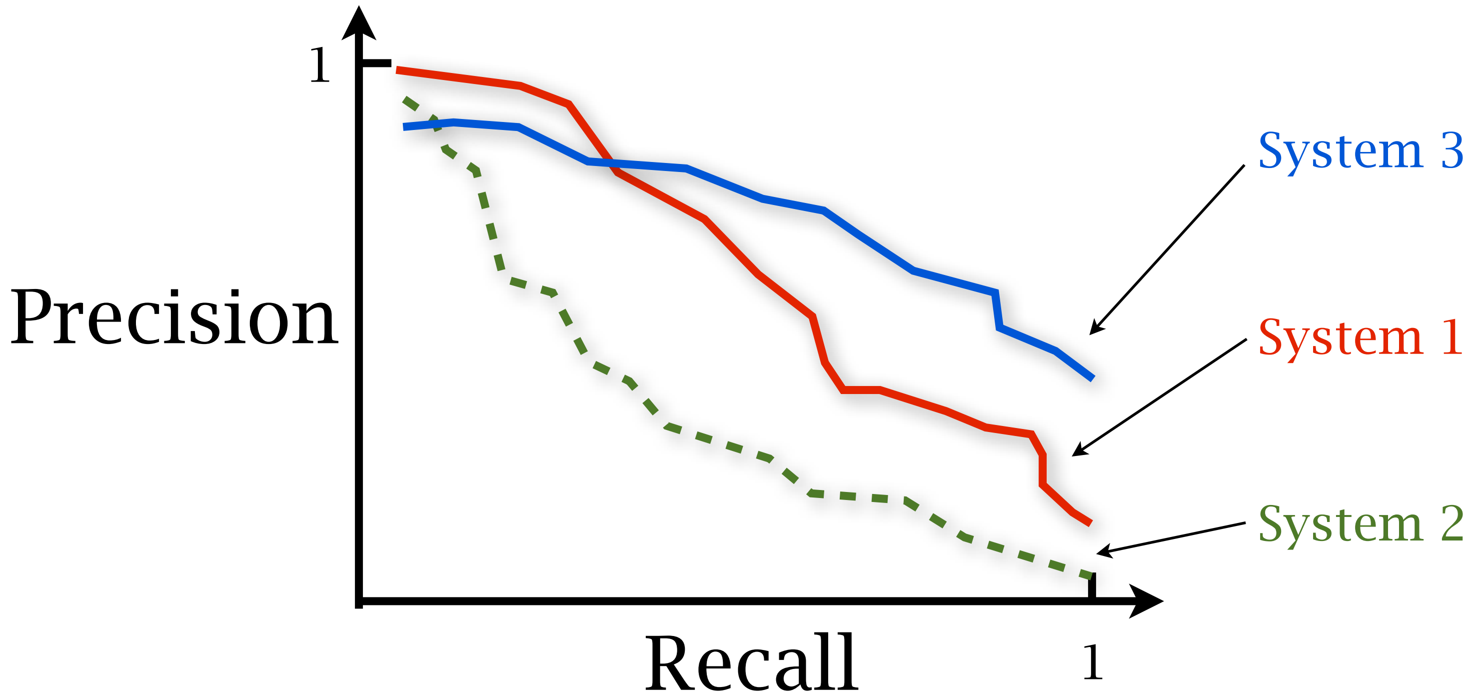
Precision and recall

Enumerate all *k* to produce a set of (P,R) pairs



Precision and recall

Compare retrieval systems



System 1 is better than System 2
System 1 v.s. System 3?

Precision and recall

Compare retrieval systems

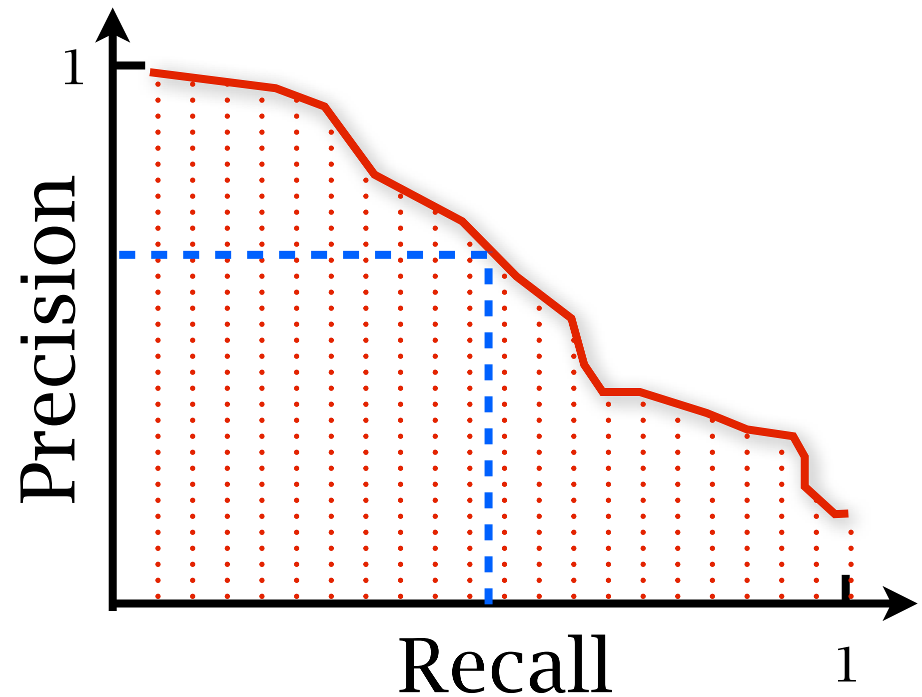
Precision/recall at a fixed *k*

Area under PR-Curve:

Position where P=R

F-measure:

for arbitrary cut-point



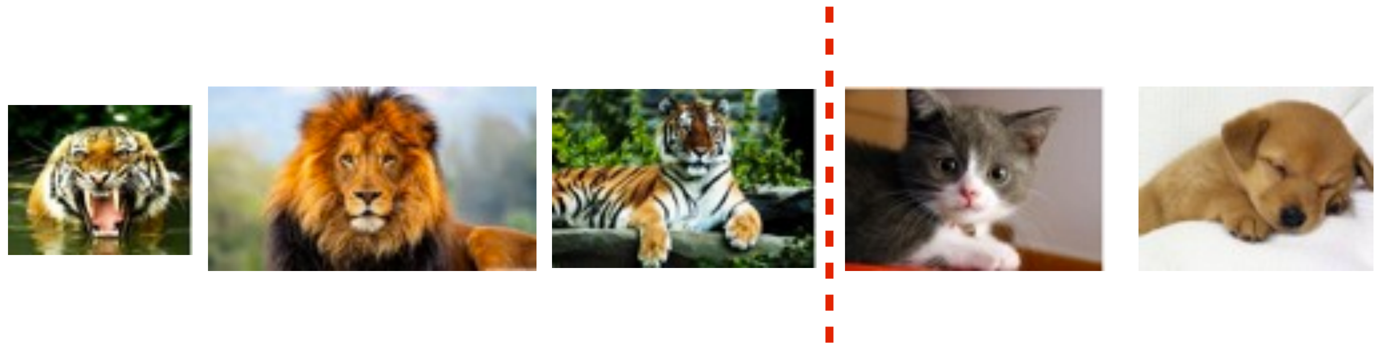
$$F = \frac{1}{\frac{1}{2}\left(\frac{1}{P} + \frac{1}{R}\right)}$$

Harmonic mean: the probability of the binary random variable whose expectation equals the average expectation of two binary random variables

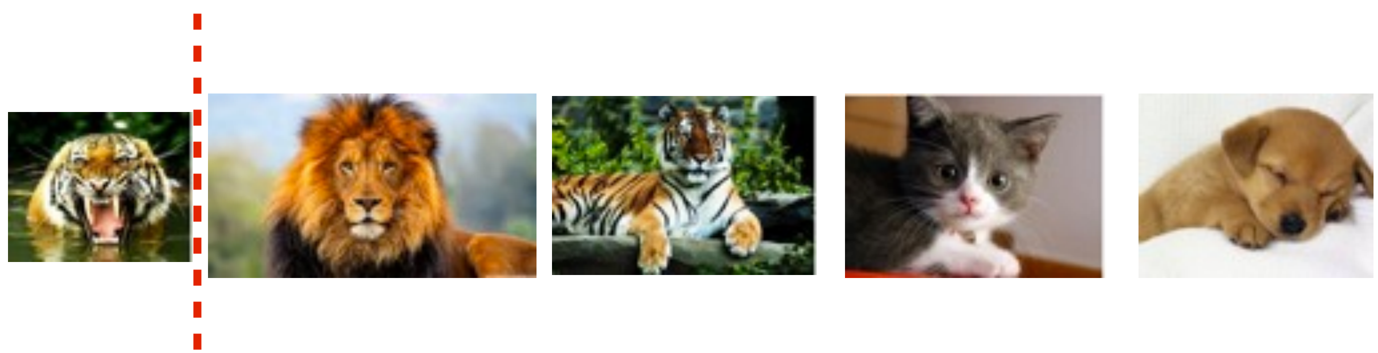
Precision v.s. recall

application dependent

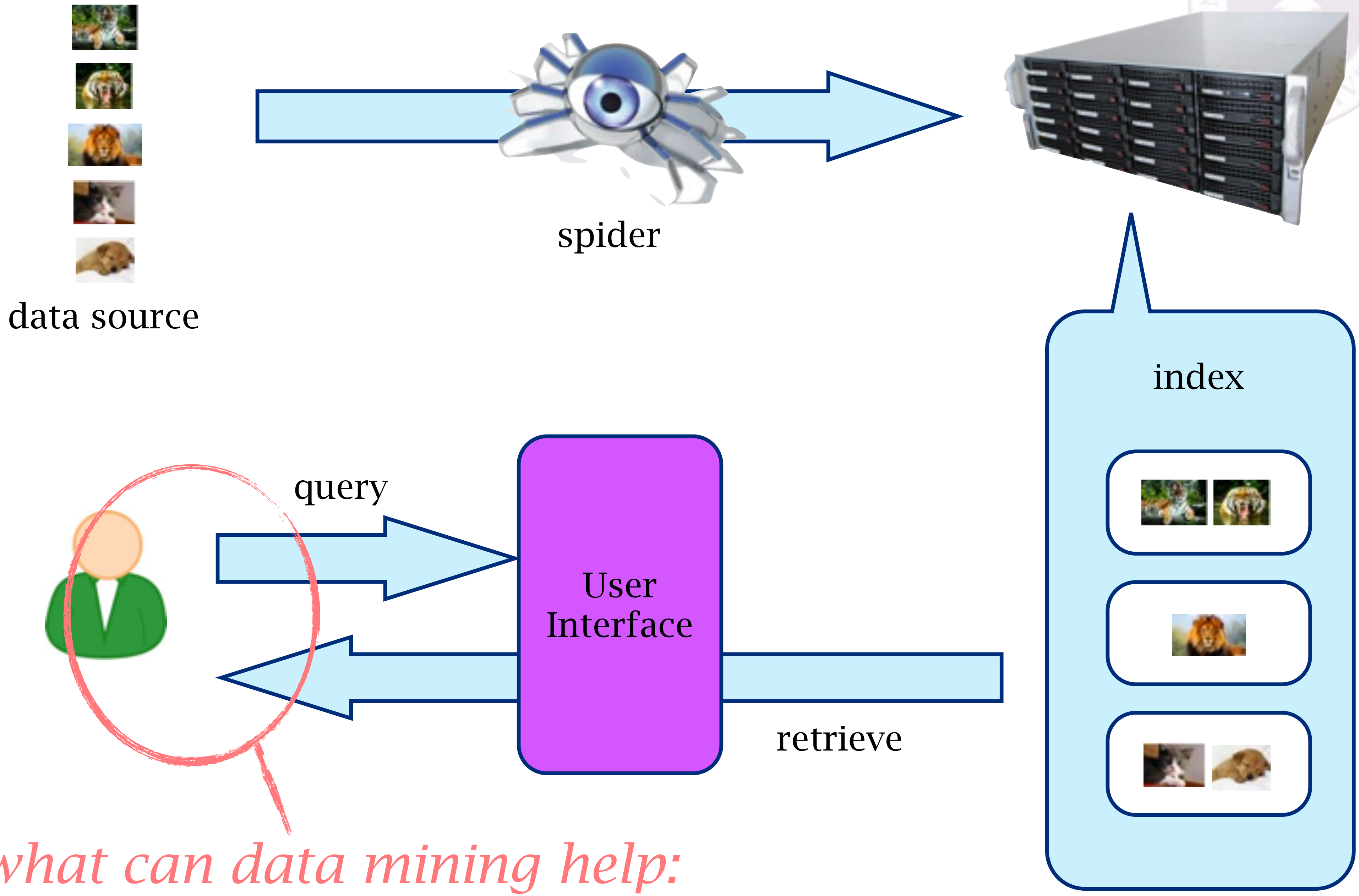
Criminal face retrieval: high recall

 output: 

Recommendation in social network: high precision

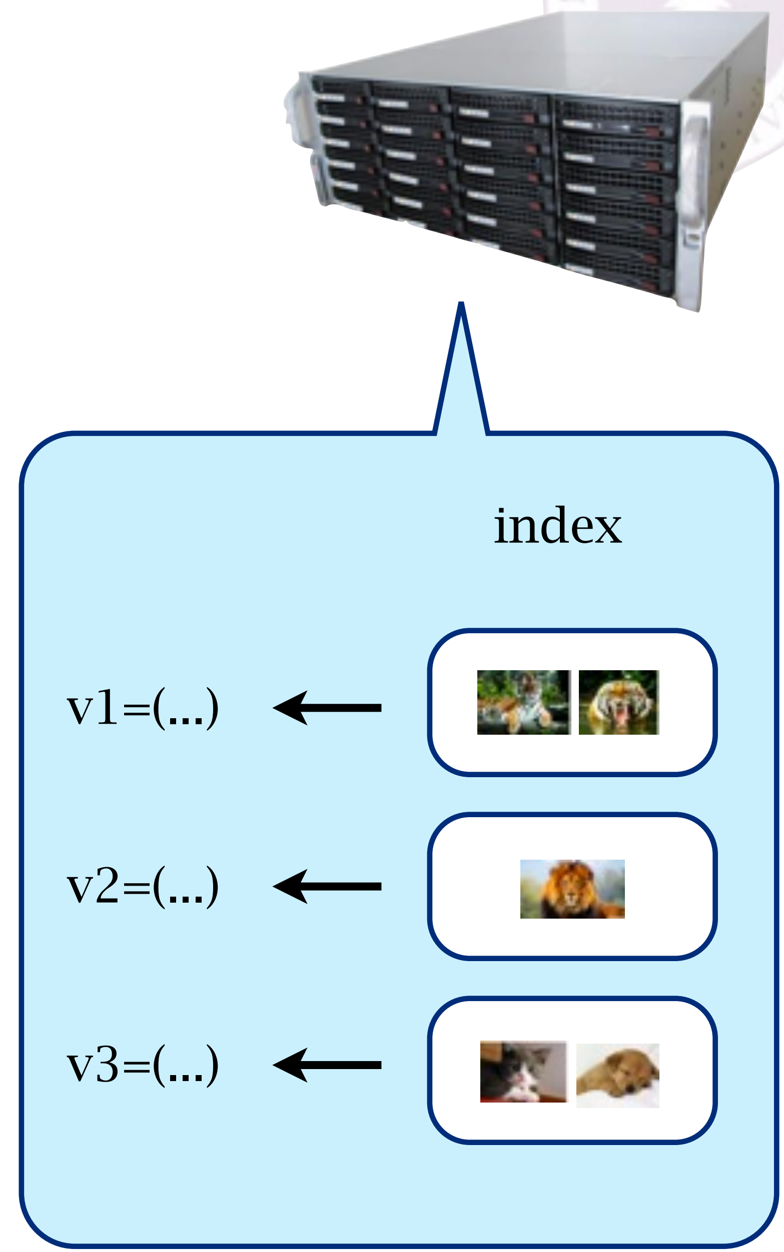
 output: 

IR Systems



what can data mining help: find better match

Understand the content: feature extraction



Text: vector representation

Dictionary: (1,苏州) (2. 南京) (3. 孔子) (4. 老子)...

苏州老子雕塑卖萌 背对裤衩楼“吐舌扮鬼脸”（图）

老子雕像继裸女座椅雕塑之后，苏州金鸡湖畔的一尊老子雕塑再度引发争议。道家创始人老子以朴素辩证法思想和无为而治的政治主张，润泽千年，成为中华文化不可或缺的瑰宝。然而，就是这样一个万民敬仰的圣贤，在这尊雕塑上却眼睛紧闭，舌头伸出，露出嘴中一个大门牙，作出一副“龇牙吐舌”的怪状，雷倒了许多路过的市民和游客。昨日，这尊老子“龇牙吐舌”的雕塑在微博上被众多网友转发，一度引起广泛关注。

(2,0,0,3 ...)

Text: vector representation

Document-term frequency matrix

	t1	t2	t3	t4	t5
D1	24	21	9	0	0
D2	32	10	5	0	3
D3	12	16	5	0	0
D4	6	7	2	0	0
D5	43	31	20	0	3
D6	2	0	0	18	7
D7	0	0	1	32	12
D8	3	0	0	22	4
D9	1	0	0	34	27

cosine similarity:

$$\cos(q, x) = \frac{q^\top x}{\|q\| \cdot \|x\|}$$

Query:

(0,0,1,1,0)

features are important to the
performance of a retrieval system

Text: vector representation

Inverse Document Frequency

$$IDF(t) = \log\left(\frac{\text{Number of total documents}}{\text{Number of documents containing } t}\right)$$

Document-term frequency (TF)

	t1	t2	t3	t4	t5	t6
D1	24	21	9	0	0	3
D2	32	10	5	0	3	0
D3	12	16	5	0	0	0
D4	6	7	2	0	0	0
D5	43	31	20	0	3	0
D6	2	0	0	18	7	16
D7	0	0	1	32	12	0
D8	3	0	0	22	4	2
D9	1	0	0	34	27	25
D10	6	0	0	17	4	23

$$IDF(t1) = log\frac{10}{9} = 0.1520$$

$$IDF(t6) = \log_2\frac{10}{5} = 1$$

multiply

Document-term TF-IDF matrix

	t1	t2	t3	t4	t5	t6
D1	3.7	21	6.6	0	0	3
D2	4.9	10	3.7	0	1.5	0
D3	1.8	16	3.7	0	0	0
D4	0.9	7	1.5	0	0	0
D5	6.5	31	15	0	1.5	0
D6	0.3	0	0	18	3.6	16
D7	0	0	0.7	32	6.2	0
D8	0.5	0	0	22	2.1	2
D9	0.2	0	0	34	14	25
D10	0.9	0	0	17	2.1	23

Text: vector representation

Many ways to form features

Table 4. Performance results for eight term-weighting methods averaged over 5 collections

Term-weighting methods	Rank of method and ave. precision	CACM 3204 docs 64 queries	CISI 1460 docs 112 queries	CRAN 1397 docs 225 queries	INSPEC 12,684 docs 84 queries	MED 1033 docs 30 queries	Averages for 5 collections
1. Best fully weighted (*tfc·nfx*)	Rank *P*	1 0.3630	14 0.2189	19 0.3841	3 0.2626	19 0.5628	11.2
2. Weighted with inverse frequency *f* not used for docs (*txc·nfx*)	Rank *P*	25 0.3252	14 0.2189	7 0.3950	4 0.2626	32 0.5542	16.4
3. Classical tf × idf No normalization (*tfx·tfx*)	Rank *P*	29 0.3248	22 0.2166	219 0.2991	45 0.2365	132 0.5177	84.4
4. Best weighted probabilistic (*nxx·bpx*)	Rank *P*	55 0.3090	208 0.1441	11 0.3899	97 0.2093	60 0.5449	86.2
5. Classical idf without normalization (*bfx·bfx*)	Rank *P*	143 0.2535	247 0.1410	183 0.3184	160 0.1781	178 0.5062	182
6. Binary independence probabilistic (*bxx·bpx*)	Rank *P*	166 0.2376	262 0.1233	154 0.3266	195 0.1563	147 0.5116	159
7. Standard weights cosine normalization (original Smart) (*txc·txx*)	Rank *P*	178 0.2102	173 0.1539	137 0.3408	187 0.1620	246 0.4641	184
8. Coordination level binary vectors (*bxx·bxx*)	Rank *P*	196 0.1848	284 0.1033	280 0.2414	258 0.0944	281 0.4132	260

[Salton and Buckley, 88]

Text: vector representation

the vector representation usually results high dimensional features

TF-IDF + PCA = LSA (Latent Semantic Analysis)

a dimension in LSA is a weighted combination of words

indexing using LSA implicitly involves more key words

Image: features

common ingredient:

colors

RGB, HSV, LIB...

texture

Fourier transformation, wavelets

gradients

edges, descriptors

Image: features

Global features

1. 3-D color feature vector
   – Spatially averaged over the whole image
   – Euclidean distance
2. k-dimensional color histogram
   – bins selected by partition based-based clustering algorithm such as k means
   – k is application dependent
   – Mahanalobis distance using inverse variances
3. 3-D Texture Vector
   – coarseness/scale, directionality, contrast
4. shape feature based on area, circularity, eccentricity, axis orientation, moments

Image: features

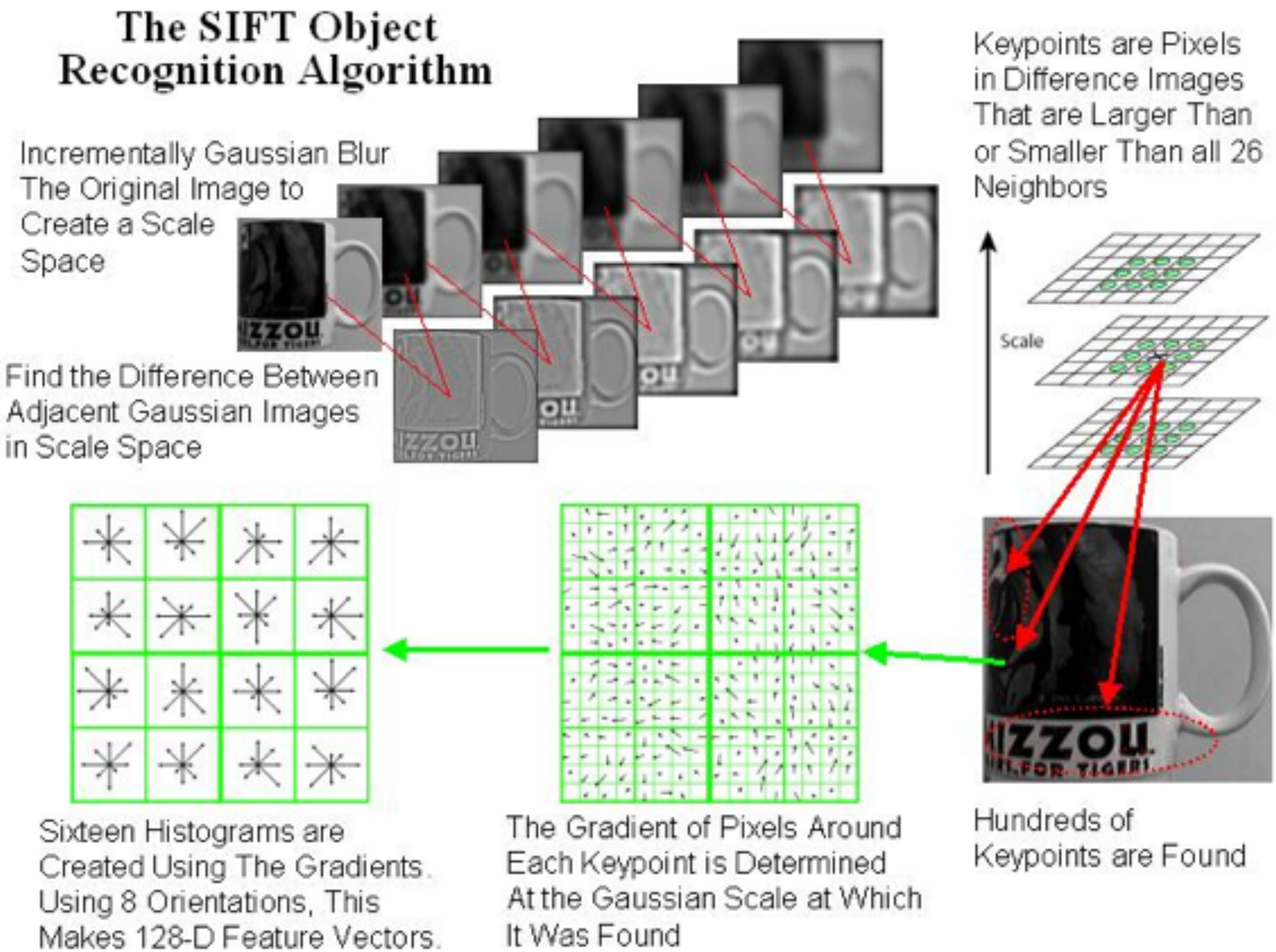
Local features

bag-of-words

split the images into small pieces
extract a feature vector per piece
clustering to find centers of feature vectors
each image by a vector of frequency of centers

Image: features

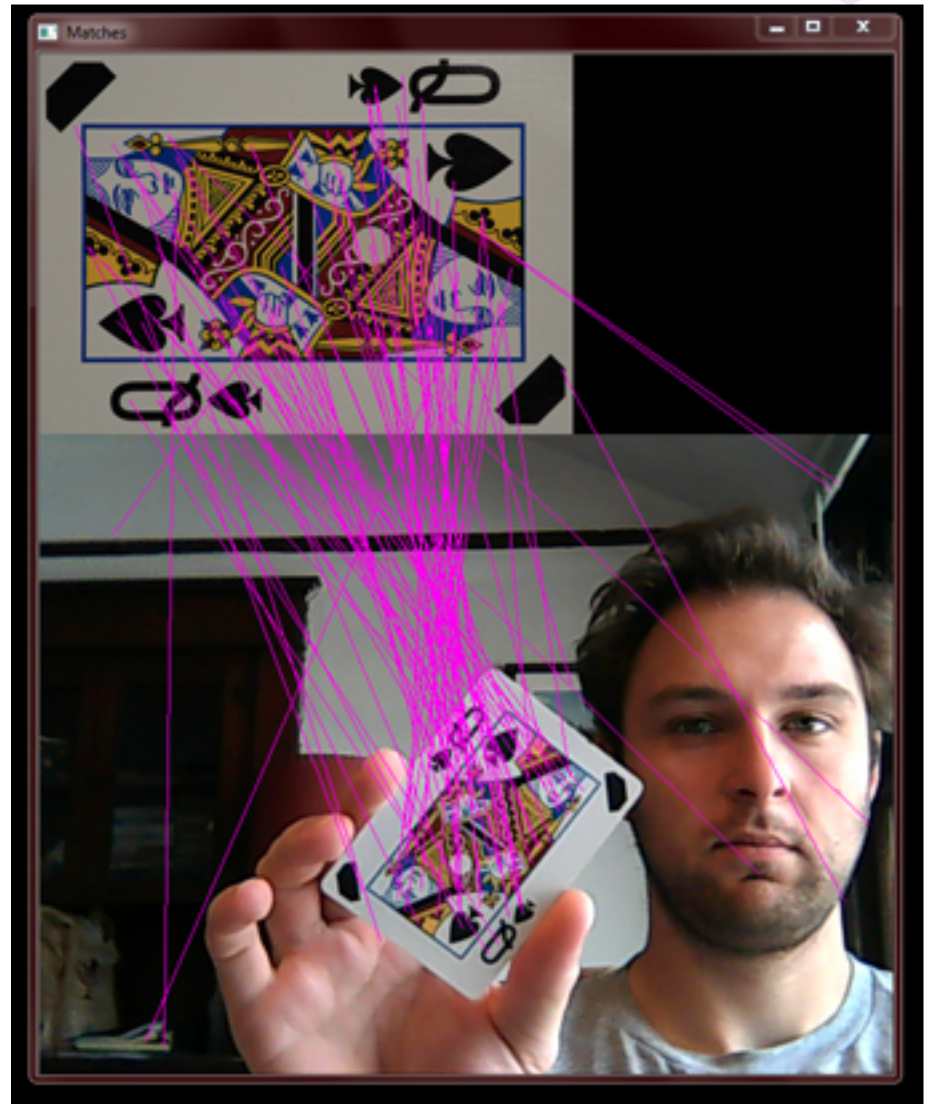
Local features



Pic from http://eecs.vanderbilt.edu/CIS/CRL/wm.shtml

Image: features

- Local features
  - Bag of words of SIFT vectors



Pic from http://blogs.oregonstate.edu/hess/sift-library-places-2nd-in-acm-mm-10-ossc/

Audio: features

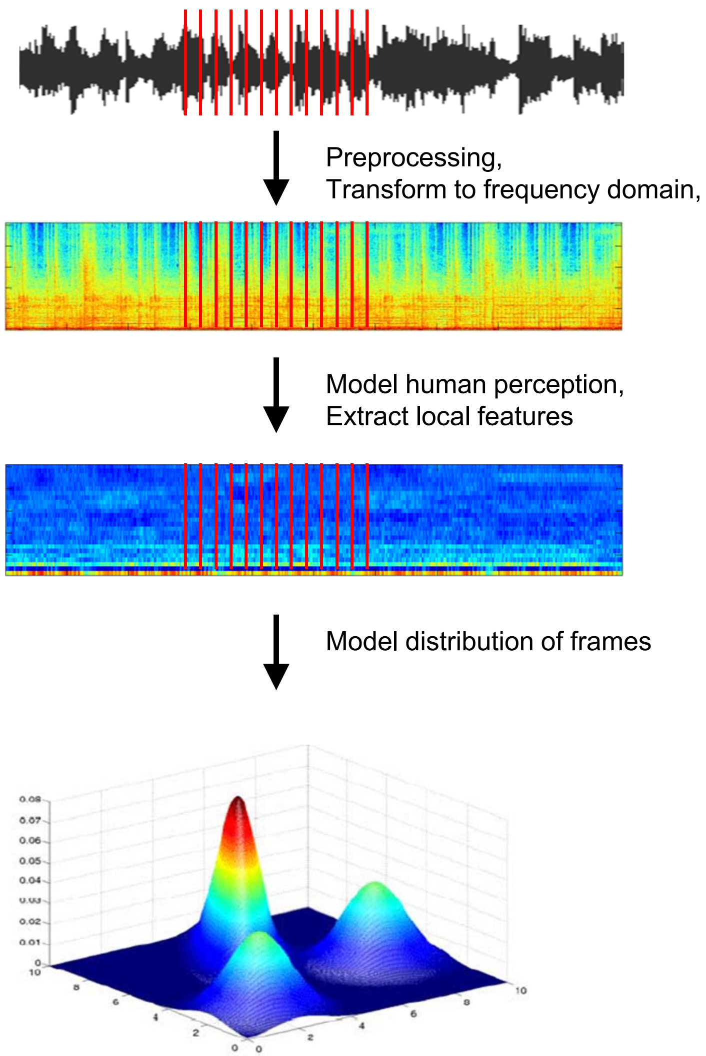
voice audio: speech-to-text transformation

music audio: extract semantic features

Music: features

frame-level processing

- cut frames out
- extract frame features
- bag-of-frame distribution

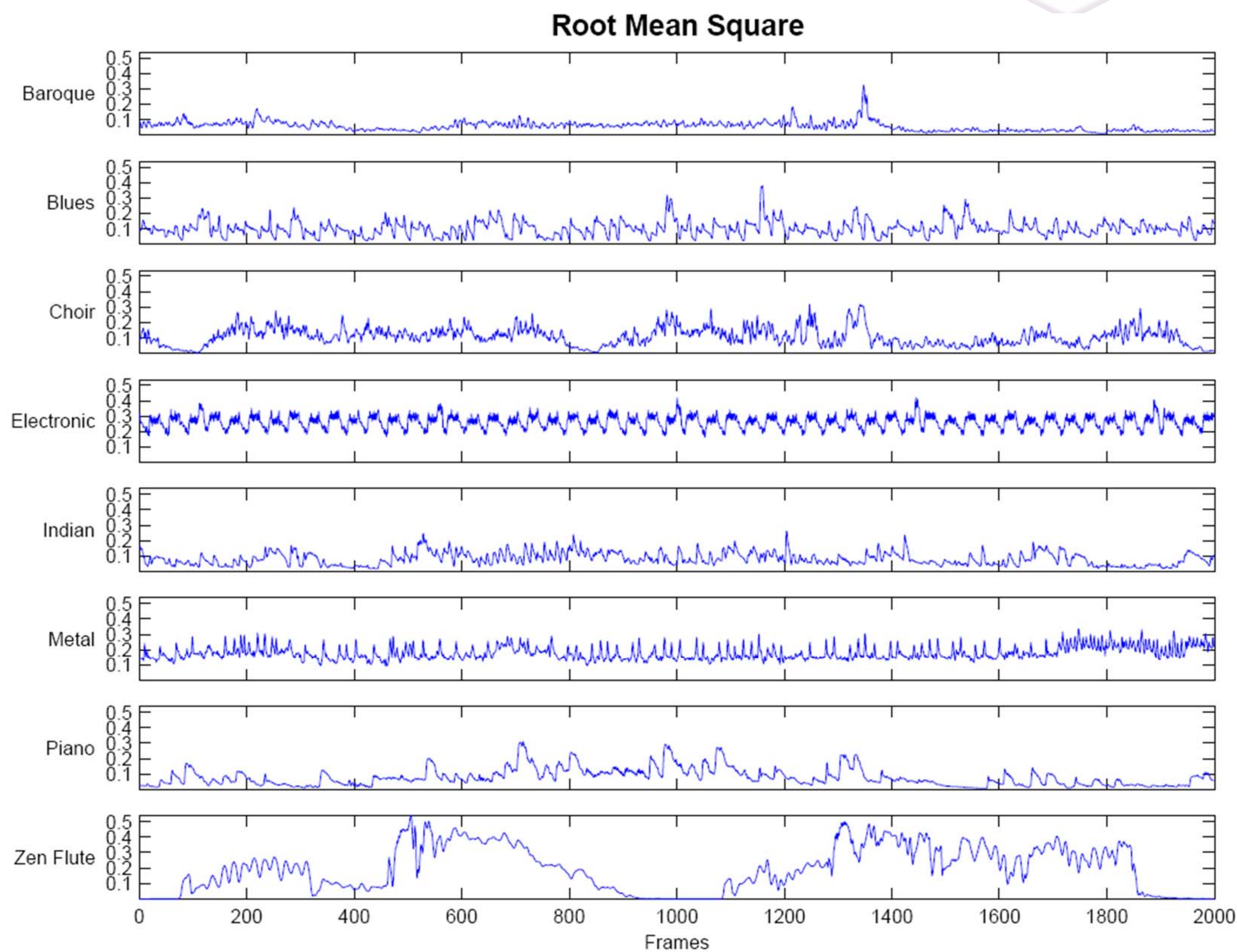


Music: features

Root-Mean-Square (RMS) Energy

$$RMS_t = \sqrt{\frac{1}{K} \cdot \sum_{k=t\cdot K}^{(t+1)\cdot K-1} s(k)^2}$$

s(*k*) is the signal value in time domain



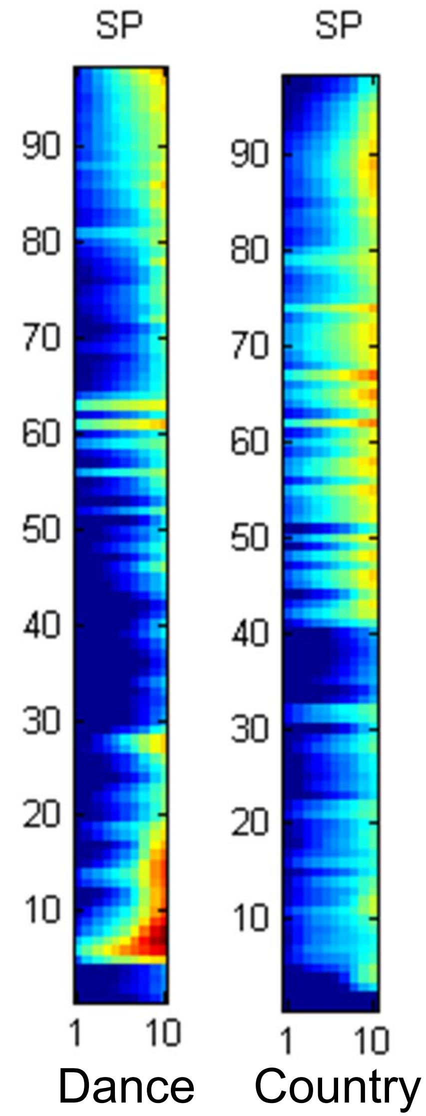
[from Markus Schedl and Peter Knees: Music Information Retrieval 2.0, ECIR'12 Tutorial.]

Music: features

Spectral Pattern

transform into spectrum domain

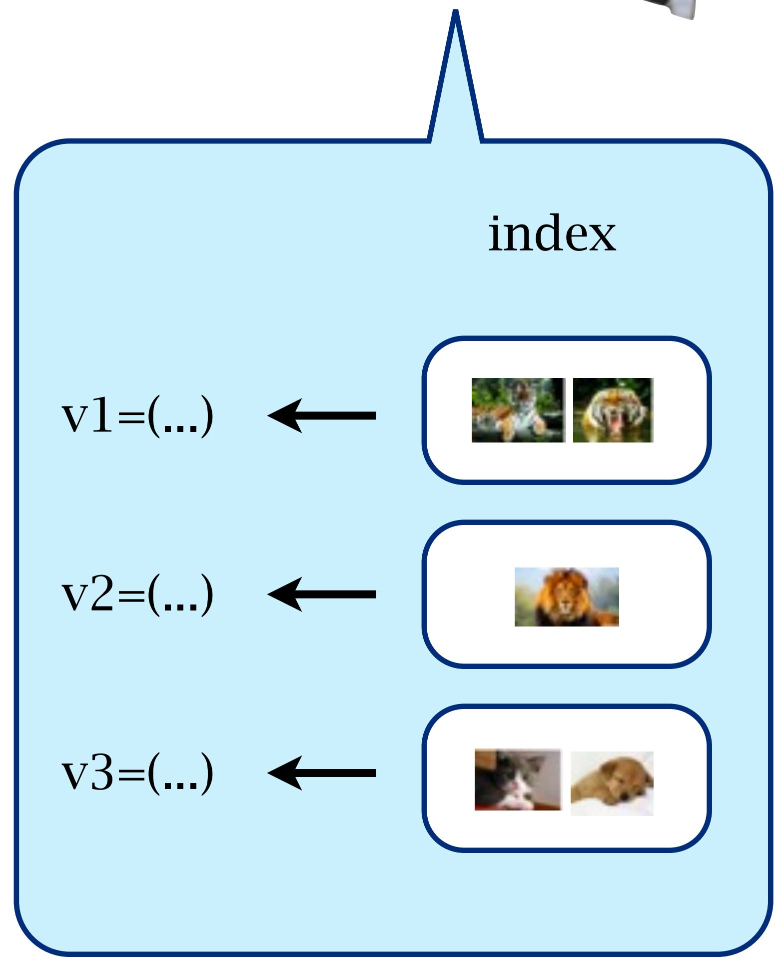
sort the energy in each frequency band of a block of frames



[from Markus Schedl and Peter Knees: Music Information Retrieval 2.0, ECIR'12 Tutorial.]

Understand the user

PageRank is a heuristic,
data reflect the real needs of users



Learning to rank

Transform to binary classification

 output:

\+ 

− 

\+ 

− 

− 

Learning to rank

Transform to binary classification

+ − + − −

 output: 

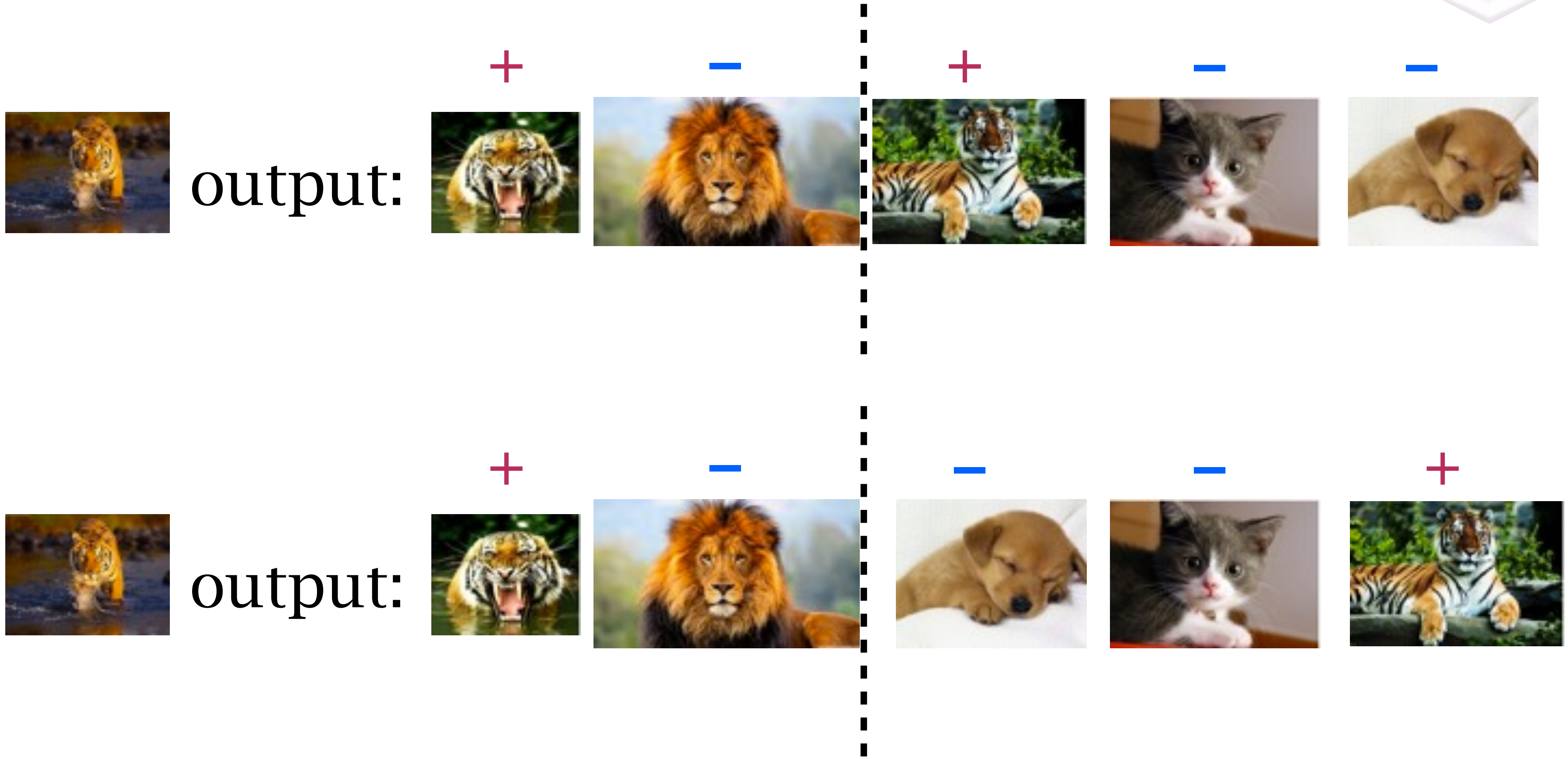
learn a binary classifier

weight items by the confidence of the classifier

+ −

Learning to rank

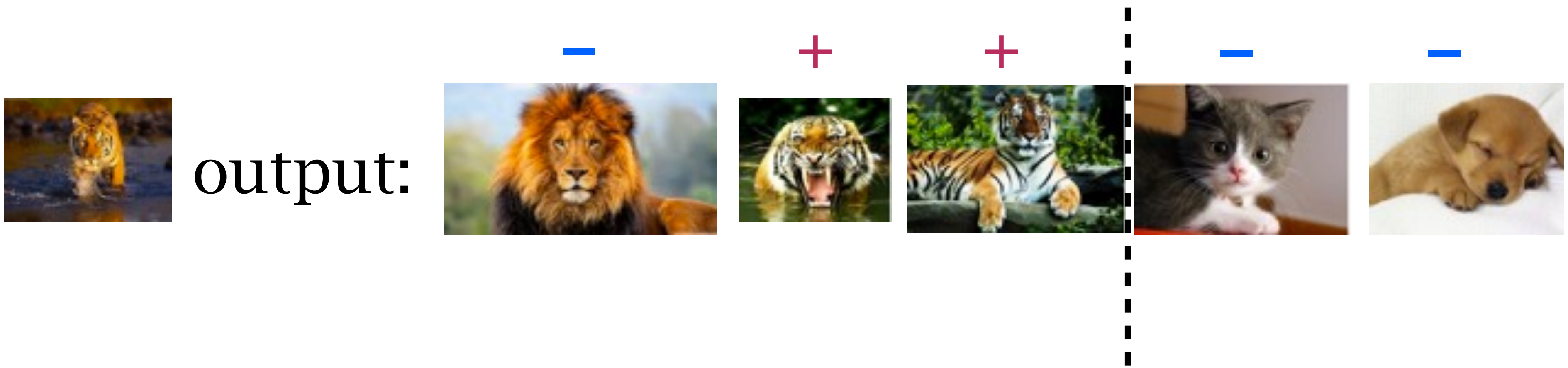
Binary classification ≠ ranking



same classification error
different ranking error

Learning to rank

Binary classification ≠ ranking



same ranking error (certain criterion)
different classification error

Learning to rank

Learning with ranking loss

$$\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} I[f(x_i^+) < f(x_i^-)]$$

Learning to rank

Learning with ranking loss:

$$\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} I[f(x_i^+) < f(x_i^-)]$$

RankSVM: using hinge loss [Herbrich et al, 2000; Joachims, 2002; Rakotomamonjy, 2004]

$$\min_{w}\left(\|w\|_2 + C\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\max\{0, 1-(f(x_i^+)-f(x_i^-))]\right)$$

Learning to rank

Learning with ranking loss:

$$\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} I[f(x_i^+) < f(x_i^-)]$$

RankBoost: using exp-loss [Freund et al, 2003]

Algorithm **RankBoost**
Given: initial distribution D over $\mathcal{X} \times \mathcal{X}$.
Initialize: $D_1 = D$.
For $t = 1, \ldots, T$:

- Train weak learner using distribution D_t.
- Get weak ranking $h_t : \mathcal{X} \to \mathbb{R}$.
- Choose $\alpha_t \in \mathbb{R}$.
- Update: $D_{t+1}(x_0, x_1) = \dfrac{D_t(x_0, x_1)\exp(\alpha_t(h_t(x_0) - h_t(x_1)))}{Z_t}$
  where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final ranking: $H(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$

Learning to rank

Learning with ranking loss

 output: 

+ −

learn a ranker

weight items by the ranker output value

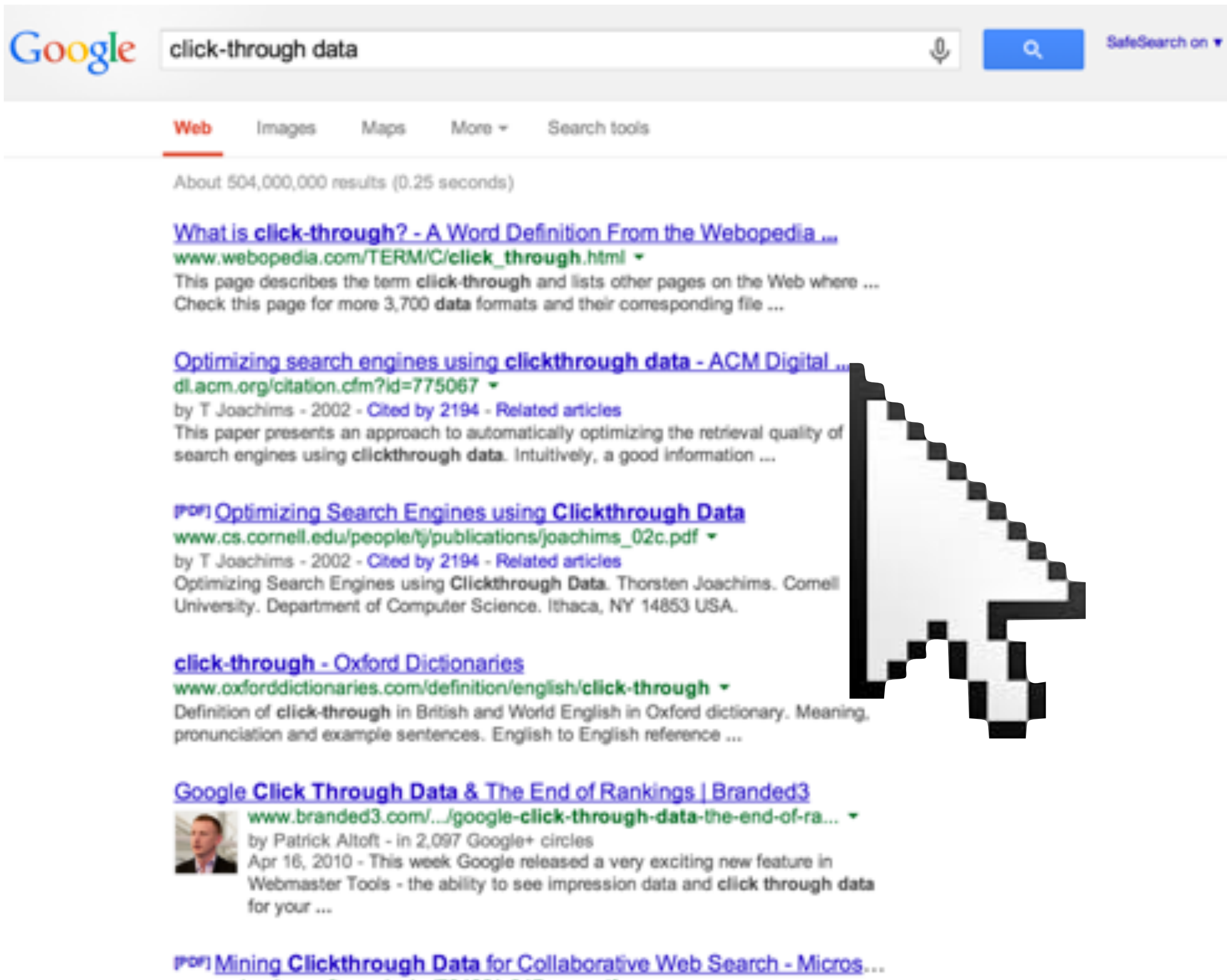
Relevance feedback

The lack of labels

a) traditional Web search　　*b) WebSIS*

Implicit feedback

Click-through data



Involve user features

different users may use the same keywords for different purpose

  output:

\+ − + − −

geographic data

computer configurations

sites visited

习题

对于用户的一条查询，数据库中总共有100个相关对象，系统返回了10个对象，其中不相关的有3个，请问对于这一条查询，系统的查准率（Precision）和查全率（Recall）各是多少?