IIIT-H

Web Mining
Lecture 22: Query Understanding by Log Mining

Manish Gupta

26th Oct 2013

Slides borrowed (and modified) from
Daxin Jiang, Jian Pei, Hang Li. Web Search/Browse Log Mining: Challenges, Methods, and Applications. Tutorial at WWW 2010

Recap of Lecture 21: Introduction to Web Search Query Log Mining

- Search and browse logs
- Log mining applications
- Four data structures
- Query Statistics
- Query Classification

Announcements

Today's Agenda

- Query Expansion, Refinement, and Suggestion
- Temporal and Spatial Aspects of Queries
- Text Mining from Query Logs

Today's Agenda

- **Query Expansion, Refinement, and Suggestion**
- Temporal and Spatial Aspects of Queries
- Text Mining from Query Logs

Outline of Query Expansion, Refinement, and Suggestion

- Overview of Query Expansion
- Methods for Query Expansion
- Overview of Query Refinement
- Methods for Query Refinement
- Overview of Query Suggestion
- Methods for Query Suggestion
- Summary of Query Expansion, Refinement, and Suggestion

Overview of Query Expansion

- Re-write query to increase search recall
- Example: ‘ny times’ → ‘ny times new york’
- Has been studied in IR from many years

Methods for Query Expansion

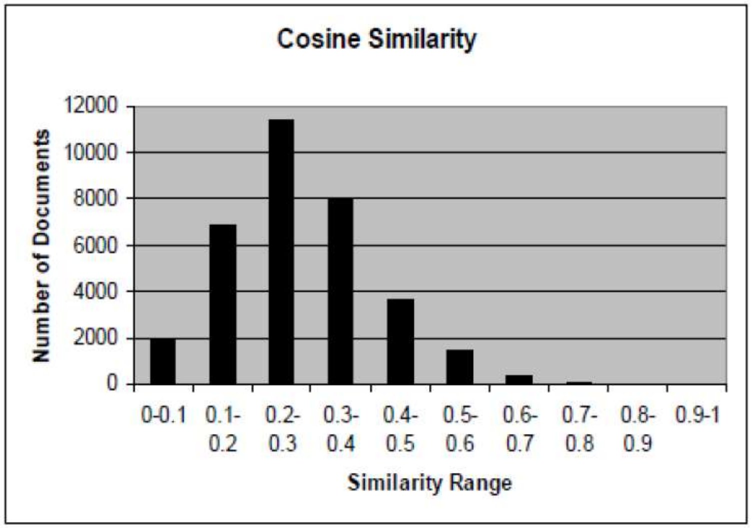
- Traditional approach
  - – Global methods
    - Thesauri (e.g., Longman dictionary or WordNet)
    - Automatic thesaurus generation
  - – Local methods
    - Explicit relevance feedback
    - Pseudo relevance feedback
  - – Combination of global and local methods
- Using log data
  - – Using click-through data [Cui02]
  - – Using session data [Fonseca05]

Query Expansion Using Click-Through Data [Cui02]

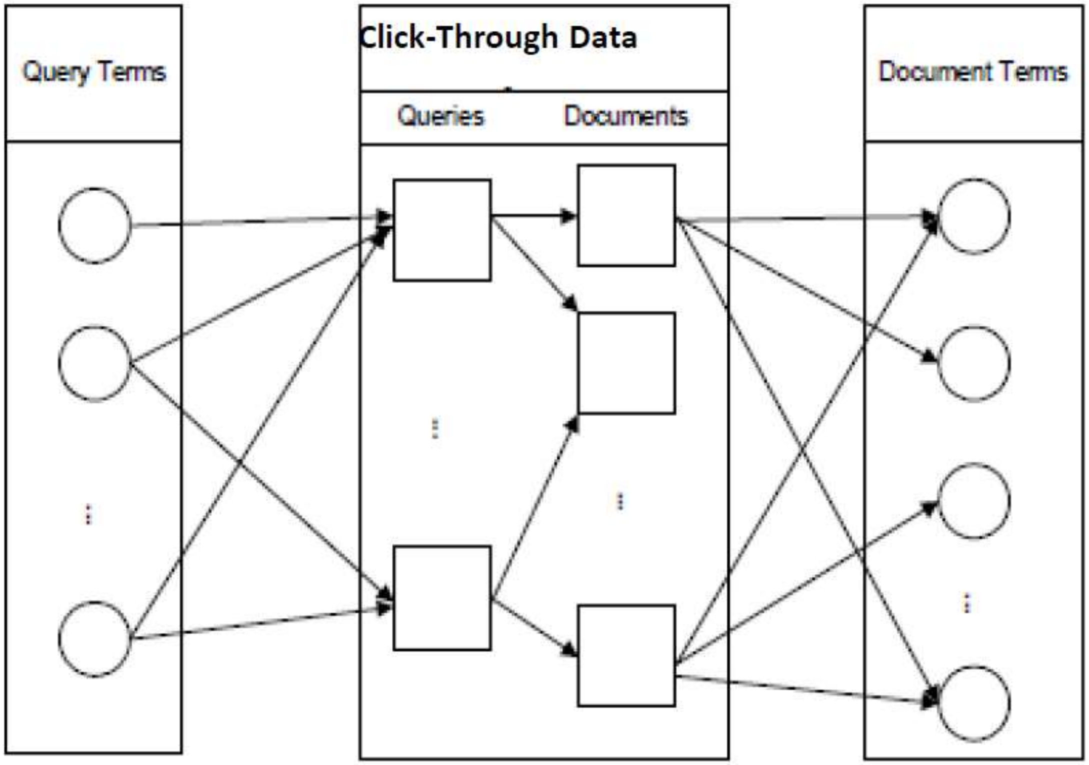
- There is gap between query space and document space
  - Web queries are often short and ambiguous
  - Users may not use the same terms appearing in documents as search keywords
- Query terms are linked to document terms by click-through data
  - If a set of documents is often linked to a set of queries, then the terms in the documents are strongly related to the terms of the queries

Gap between Query Space and Document Space

- Each document *d* is represented by
  - *W(d)*: terms in *d*
  - *W(q)*: queries for which document is clicked on
- Calculating cosine similarity between *W(d)* and *W(q)*
- Few documents have similarity values above 0.8
- Average similarity value is 0.28
- Large gap between two spaces



Mapping Query Terms to Document Terms



$$P\left(w_j(d), w_i(q)\right) = \sum_{D_k} P\left(w_j(d) \middle| D_k\right) P(D_k | w_i(q))$$

where $w_j(d)$ is the document term and $w_i(q)$ is the query term

Query Expansion by Term Correlations

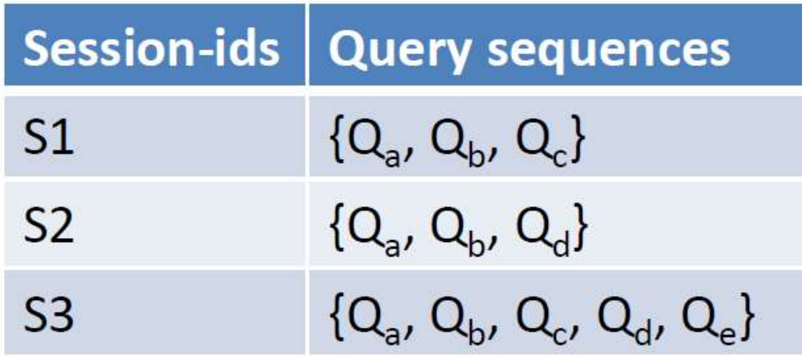
- Given a query Q, calculate the weight for each term by $Weight\left(w_j(d)\right) = \ln(\prod_{w_i(q)\in Q}(P\left(w_j(d)|w_i(q)\right) + 1))$
- Use the top terms for expansion
- Example: top terms of query ‘Steve Jobs’
  – Apple, personal computer, computer

Query Expansion Using Session Data [Fonseca05]

- Offline part
  - Find all queries "associated" with query q
  - Group associated queries into "concepts"
- Online part
  - Given query q, find all concepts of q
  - Ask user to select concept
  - Expand q with the other queries in selected concept

Offline Step 1: Association Rule Mining

- Session data

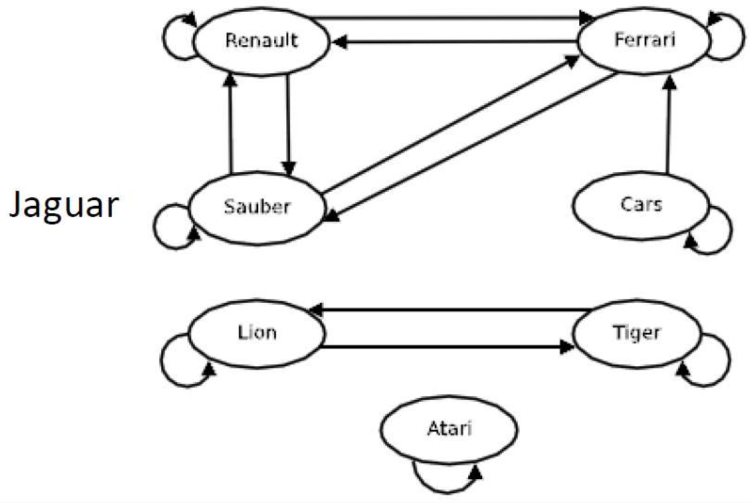


Session-ids	Query sequences
S1	$\{Q_a, Q_b, Q_c\}$
S2	$\{Q_a, Q_b, Q_d\}$
S3	$\{Q_a, Q_b, Q_c, Q_d, Q_e\}$

- Mining length-2 frequent sequential patterns
  - Many methods in data mining
- Deriving association rules
  - For frequent pattern $\{Q_a, Q_b\}$, if confidence is greater than threshold, generate rule: $Q_b \rightarrow Q_a$

Offline Step 2: Finding Concepts

- For each query Q_a, create query set R_a such that for any query $Q_i \in R_a$, rule $Q_i \rightarrow Q_a$ exists
- Build query relation graph G_a with respect to Q_a
  - Each query $Q_i \in R_a$ is vertex
  - Two queries Q_i, $\mathrm{Q_j} \in R_a$ are connected with directed edge from Q_i to Q_j if there is rule $Q_j \rightarrow Q_i$
- Concept = strongly connected



Online Part

- Given query Q, return concepts to user
- Ask user
  - Which concept she is interested in
  - Type between Q and selected concepts
- Expand query with terms in selected concept
  - Take different approaches if query-concept type is specified

Overview of Query Refinement

- Reformulate query to better represent search intent
  - Spelling error correction: ‘machin learning’ → ‘machine learning’
  - Stemming
  - Acronym expansion
- Challenges: mapping from X to Y, huge spaces
  - ‘papers on machin learn’ → ‘paper on machine learning’

Methods for Query Refinement

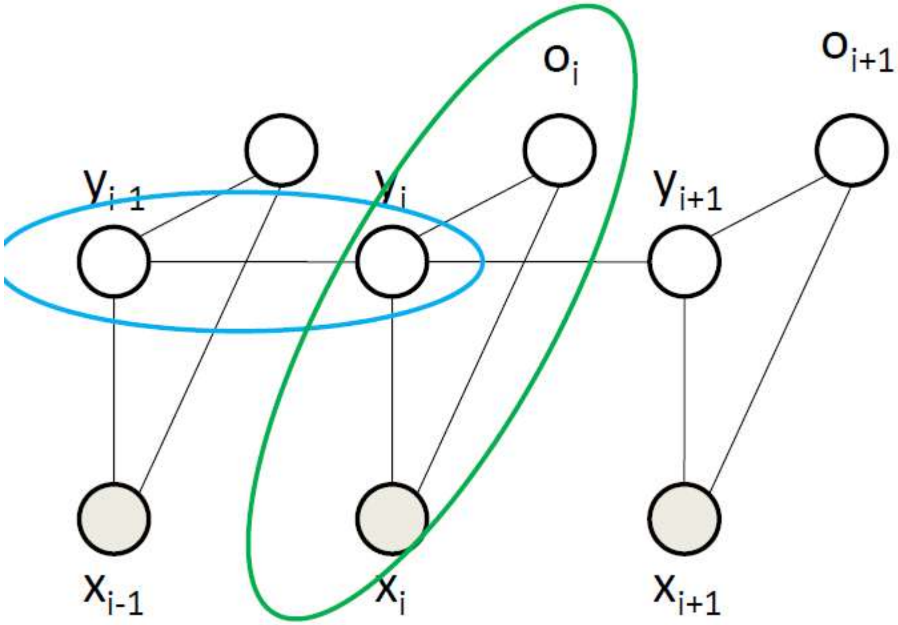
- Spelling error correction
  - Maximum Entropy Model [Li06]
  - Source Chanel Model [Cucerzen04]
- Unified and discriminative model learned from query log
  - CRF Model [Guo08]

Query Refinement Using Conditional Random Fields Model

- View query refinement as mapping from space of original queries *X* to space of refined queries *Y*
- Directly using *P(y|x)* is not practical as *X* and *Y* are huge
- Employ model *P(y, o|x)* where *O* denotes operations, reduce the output space
- Define *P(y,o|x)* as CRF
- Multiple layers of CRF is employed

Conditional Random Fields for Query Refinement [Guo08]

- $\Pr(y, o|x) = \frac{1}{Z(x)} \exp(\sum_{i=1}^{n} (\sum_k \lambda_k f_k(y_{i-1}, y_i) + \sum_k \lambda_k h_k(y_i, o_i, x)))$
- Basic CRF-QR model

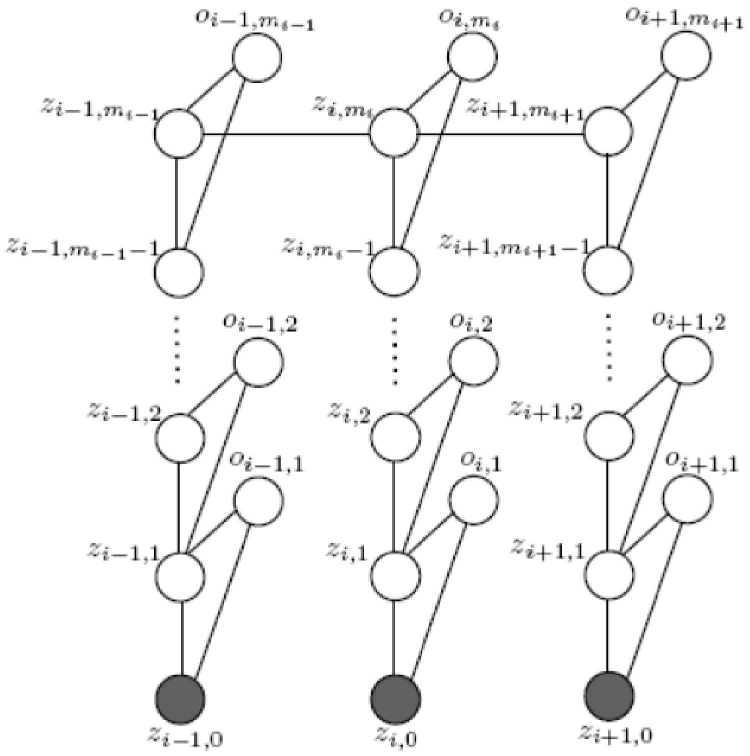


Refinement Operations

Task	Operation	Description
Spelling Error Correction	Deletion	Delete a letter in the word
	Insertion	Insert a letter into the word
	Substitution	Replace a letter in the word with another letter
	Transposition	Switch two letters in the word
Word Splitting	Splitting	Split one word into two words
Word Merging	Merging	Merge two words into one word
Phrase Segmentation	Begin	Mark a word as beginning of phrase
	Middle	Mark a word as middle of phrase
	End	Mark a word as end of phrase
	Out	Mark a word as out of phrase
Word Stemming	+s/-s	Add or Remove suffix `-s'
	+ed/-ed	Add or Remove suffix `-ed'
	+ing/-ing	Add or Remove suffix `-ing'
Acronym Expansion	Expansion	Expand acronym

CRF-QR Extended Model

- Multiple Refinement Tasks



$$\Pr(\boldsymbol{y}, \vec{o}, \vec{z} | \boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \prod_{i=1}^{n} (\phi(y_{i-1}, y_i) \prod_{j_i=1}^{m_i} \phi(z_{i,j_i}, o_{i,j_i}, z_{i,j_i-1}))$$

Query Suggestion

- Suggest queries in two types
  - Same search intent, better form
  - Related searches
- Methods
  - Using click-through data
  - Using session data
  - Context aware query suggestion [Cao08]

Methods Using Click-Through Data

- Use similar queries as suggestions for each other
- Measure similarity of queries
  - Overlap of clicked document [Beeferman00], [Wen01]
  - Similarity of category or content of clicked documents, [Wen01], [Yates04],
- Cluster queries
  - Agglomerative hierarchical method [Beeferman00]
  - DBScan [Wen01]
  - K-means [Yates04]

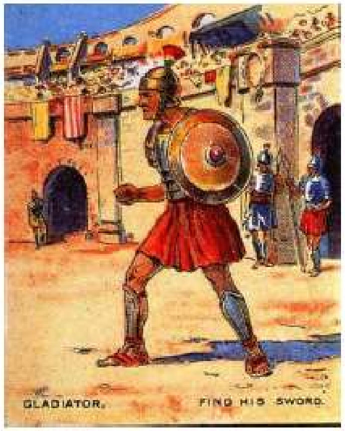
Methods Using Session Data

- Co-occurrence or adjacency in sessions
  - If Q_a and Q_b often co-occur in the same session, they can be suggestions for each other
  - If Q_b often appear immediately after Q_a in the same session, Q_b is a suggestion for Q_a
- Measures to represent correlation between Q_a and Q_b
  - Number of sessions where Qa and Qb co-occur (or are adjacent) [Jensen06][Huang03][Jones06]
  - Mutual information, Weighted mutual information [Jensen06]
  - Jaccard similarity, dependency, cosine similarity [Huang03]
  - Log likelihood ratio [Jones06]

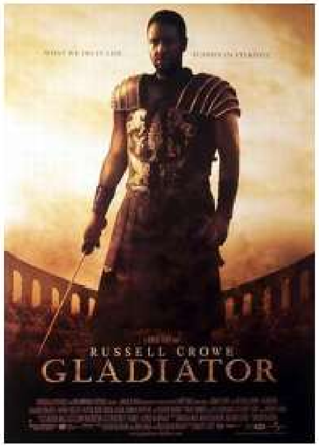
Context-Aware Query Suggestion

- User raises query “*gladiator*”

History?



People?



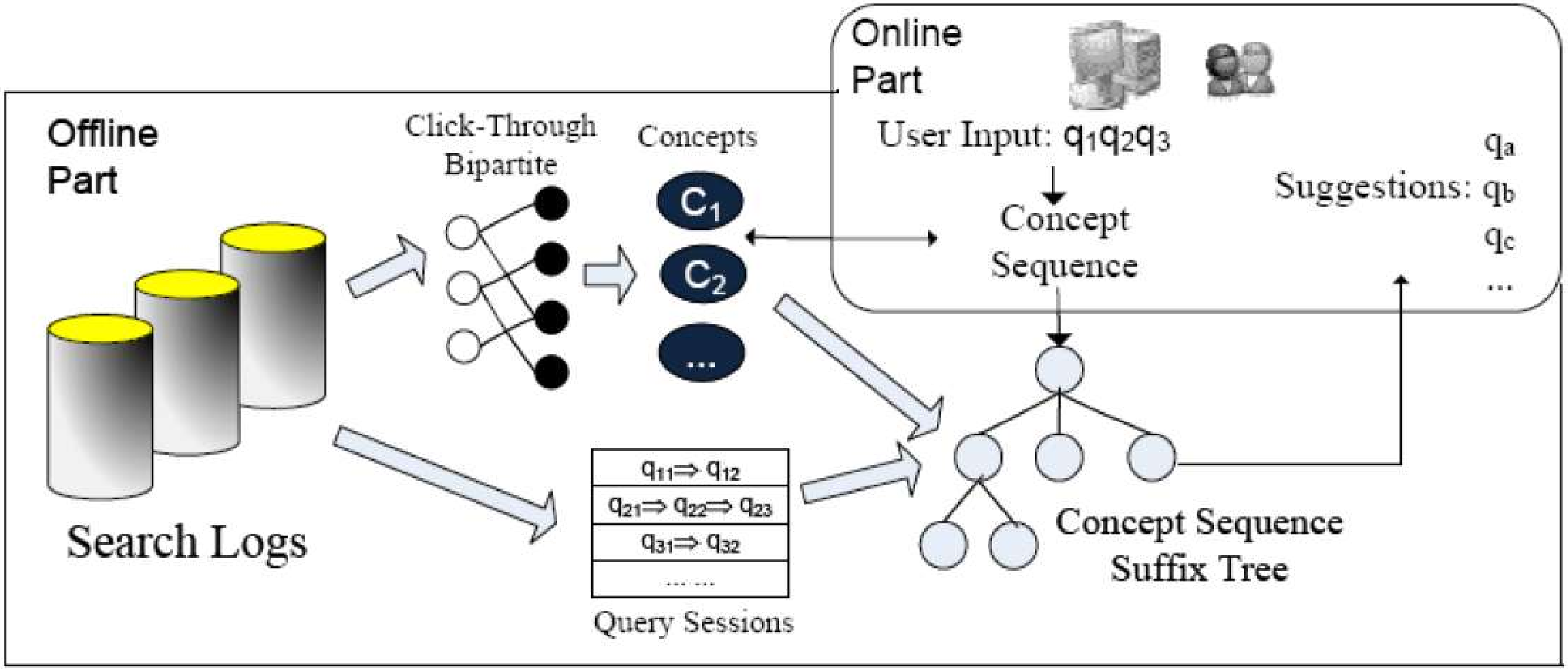
Film?

- If user raises query *“beautiful mind”* before *“gladiator”*
- Then user is likely to be interested in the film

Context-Aware Query Suggestion

- A naïve formulation
  - Given user query q_n
  - Find sequence of queries $q_1 \dots q_{n-1}$ submitted by users immediately before q_n
  - Scan log data and find out that in the same context $q_1 \dots q_{n-1}$, what queries people often ask after q_n
  - Output results as query suggestion
- Challenges: data sparseness, large scale

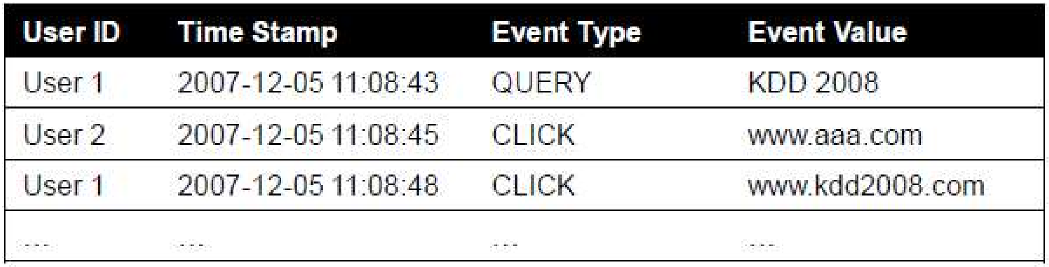
Method of Context-Aware Query Suggestion [Cao08]



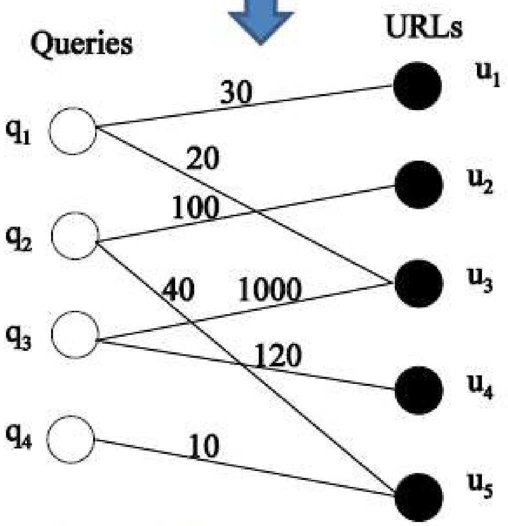
- Offline part: model learning
  - Summarizing queries into concepts by clustering queries on click-through bipartite
  - Mining frequent patterns from session data and building concept sequence suffix tree
- Online part: query suggestion

Finding Concepts from Click-Through Data

Search log



User ID	Time Stamp	Event Type	Event Value
User 1	2007-12-05 11:08:43	QUERY	KDD 2008
User 2	2007-12-05 11:08:45	CLICK	www.aaa.com
User 1	2007-12-05 11:08:48	CLICK	www.kdd2008.com
...	...	...	...



click-through bipartite

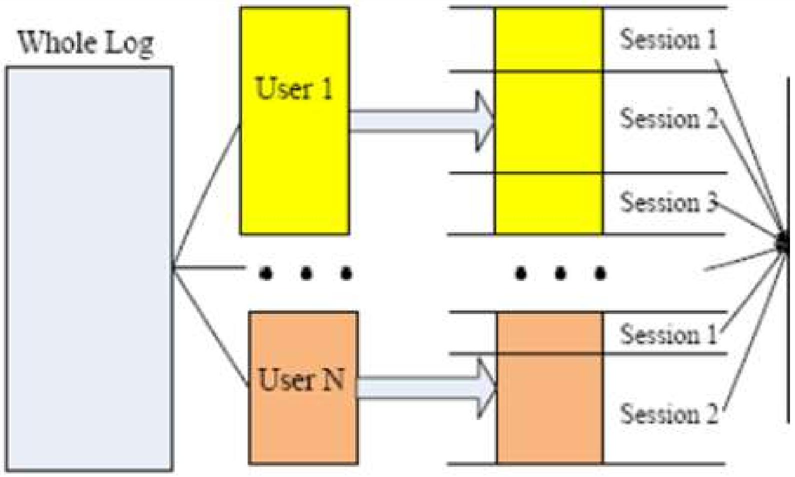
Query is represented by feature vector of URLs.

$$\vec{q_i}[j] = \begin{cases} \dfrac{w_{ij}}{\sqrt{\sum_{\forall e_{ik}} w_{ik}^2}} & \text{if } e_{ik} \text{ exists} \\ 0 & \text{otherwise} \end{cases} .$$

Distance between two queries.

$$\operatorname{dist}(q_i, q_j) = \sqrt{\sum_{u_k} (\vec{q_i}[k] - \vec{q_j}[k])^2} .$$

Finding Concept Sequences from Session Data



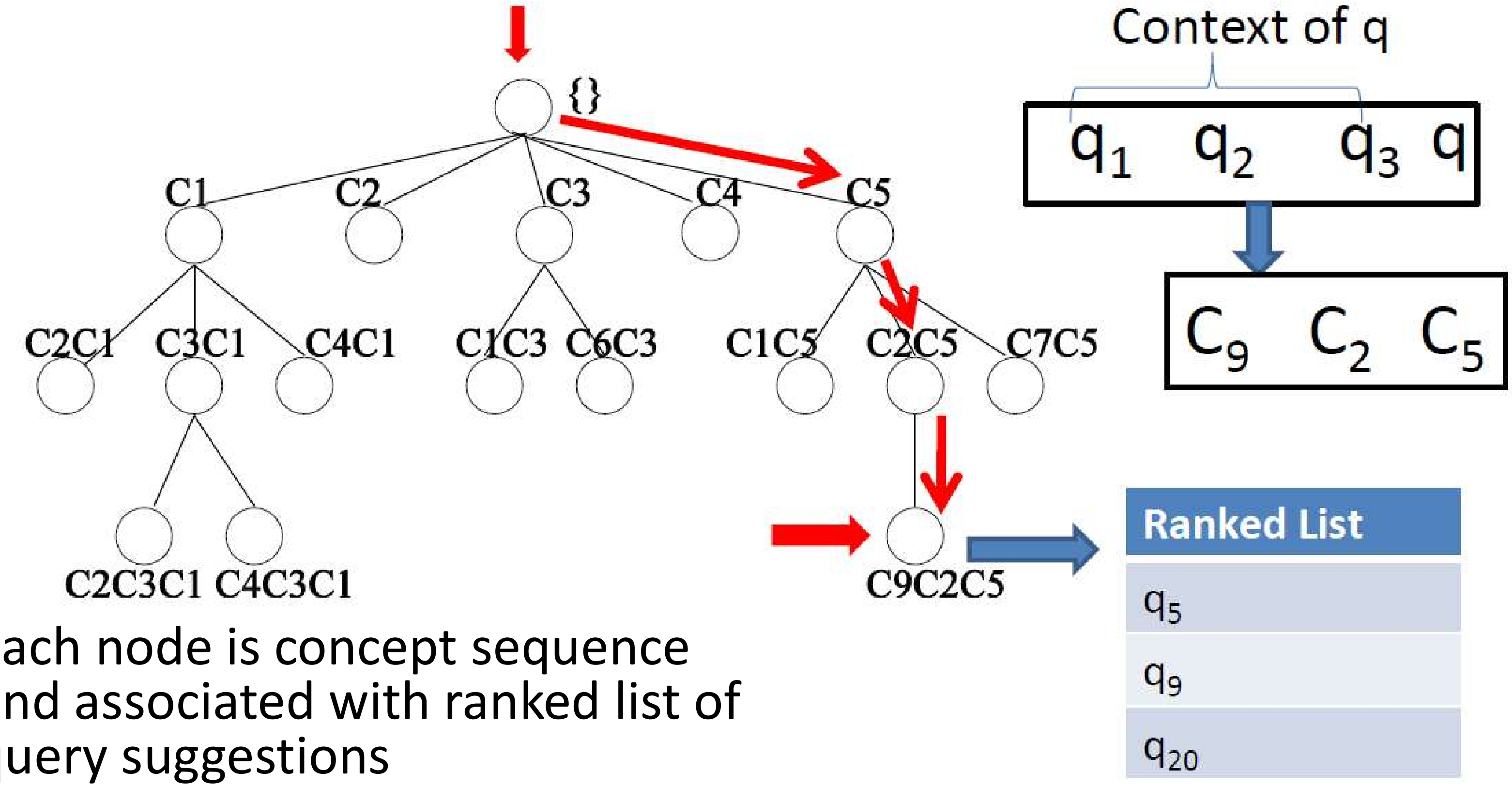
Query sequences
$q_1 \Rightarrow q_2 \Rightarrow q_3$
q_1
$q_4 \Rightarrow q_3 \Rightarrow q_7 \Rightarrow q_{11}$
……
$q_1 \Rightarrow q_2 \Rightarrow q_9$

Concept sequences
$C_1 \Rightarrow C_2 \Rightarrow C_3$
C_2
$C_4 \Rightarrow C_3 \Rightarrow C_3$
……
$C_1 \Rightarrow C_2 \Rightarrow C_3$

Examples of query sequences

Query sequences
SMTP ⇒ POP3
BAMC ⇒ Brooke Army Medical Center
Nokia N73 ⇒ Nokia N73 themes ⇒ free themes Nokia N73

Building Concept Sequence Suffix Tree



Ranked List
q_5
q_9
q_{20}

- Each node is concept sequence and associated with ranked list of query suggestions
- Each parent node is maximal suffix of its child nodes

Summary of Query Expansion, Refinement, and Suggestion

- Different methods to help users to search
  - Query expansion, refinement, and suggestion
- Click-through data and session data are useful for query expansion, refinement, and suggestion
- Using click-through data
  - Finding similar queries based on co-clicks
- Using session data
  - Finding frequent co-occurring or adjacent queries

Today’s Agenda

- Query Expansion, Refinement, and Suggestion
- **Temporal and Spatial Aspects of Queries**
- Text Mining from Query Logs

Outline of Temporal and Spatial Aspects of Queries

- Overview of Temporal Aspect of Queries
- Analysis of Query Temporal Trends
- Query Temporal Models
- Summary of Temporal Aspect of Queries
- Overview of Spatial Aspect of Queries
- Summary of Spatial Aspect of Queries

Overview of Temporal Aspects of Queries

- Temporal trends of queries
  - How queries change over time
- Query temporal models
  - Periodic query identification
  - Burst query identification

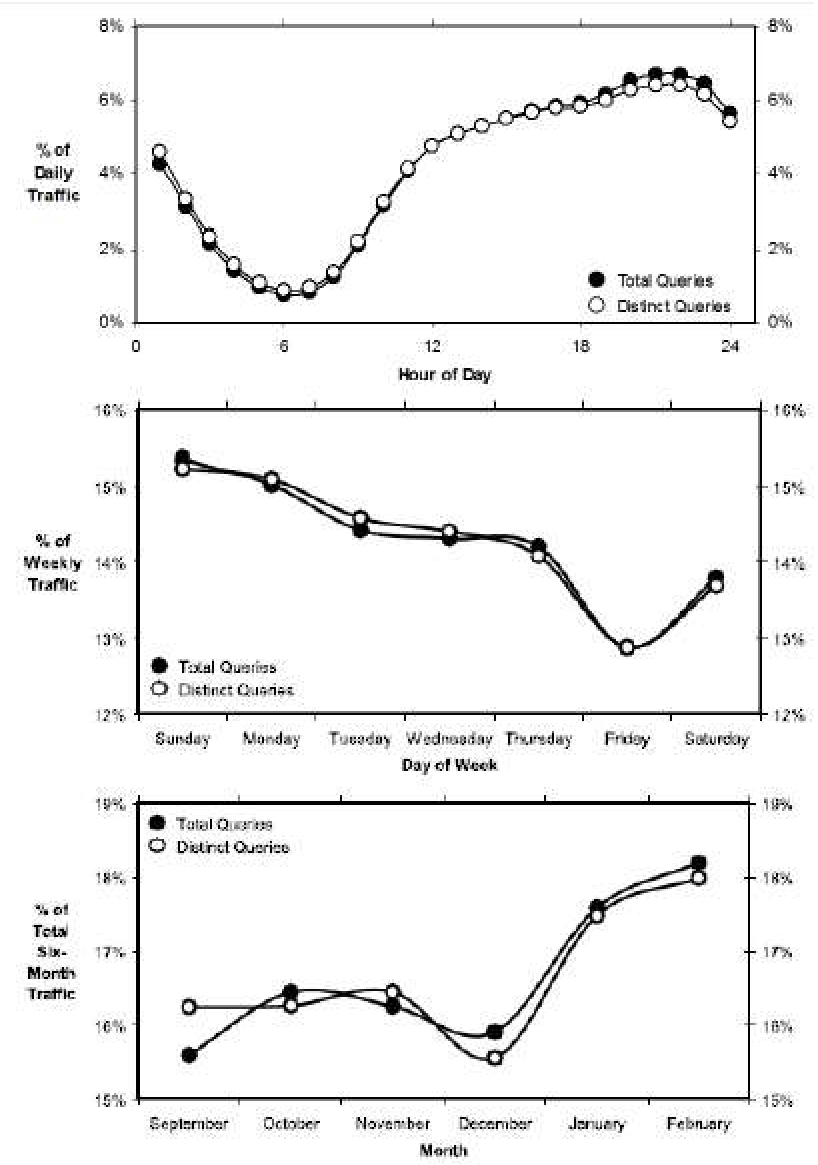
Analysis of Query Temporal Trends

- Examine several aspects of query stream over time (hourly, daily, and monthly):
  - Query volume: overall and by category
  - Query type: overall and by category
- Result of temporal trend analysis: [Beitzel07]
- Applications
  - Query classification
  - Caching strategy

Query Log Data

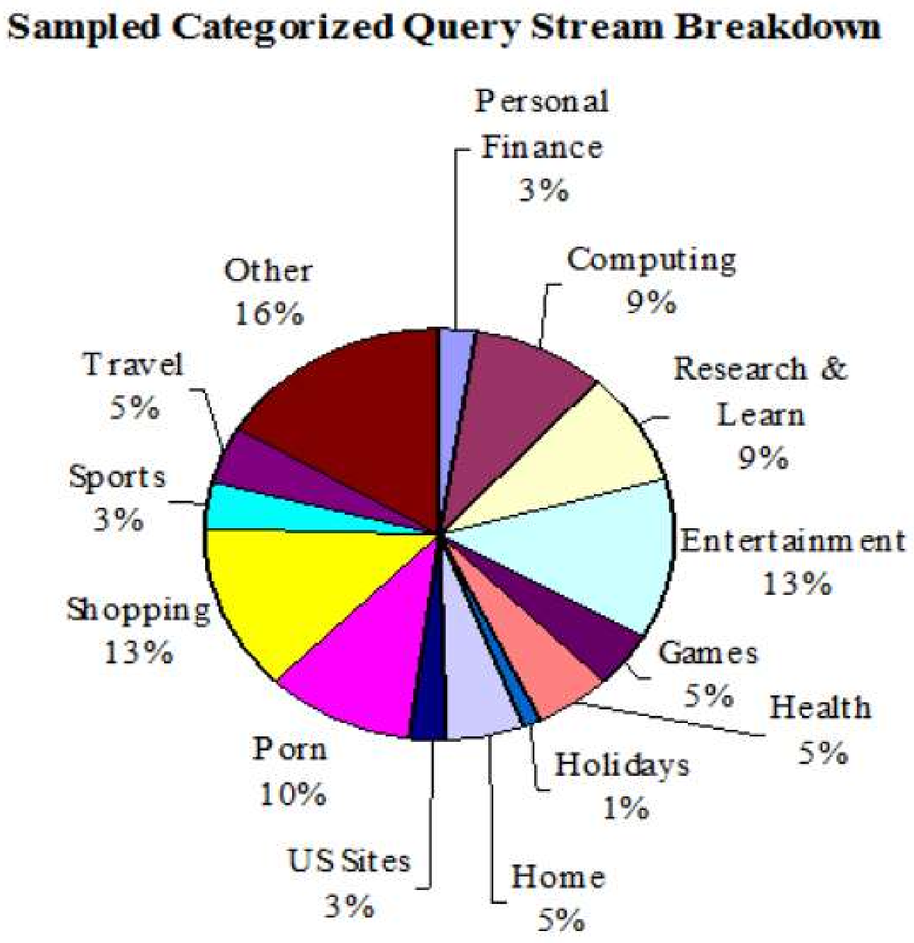
- Analyzed two AOL search logs:
  - One week of queries in December, 2003
  - Six months of queries: Sept. 2004-Feb. 2005
- Light pre-processing was done:
  - Case differences, punctuation, & special operators removed; whitespace trimmed
- Basic statistics:
  - Queries average 2.2 terms in length
  - Only one page of results is viewed 81% of the time; Two pages: 18%; Three or more: 1%
  - Consistent with previous studies

Traffic Volume Over Time



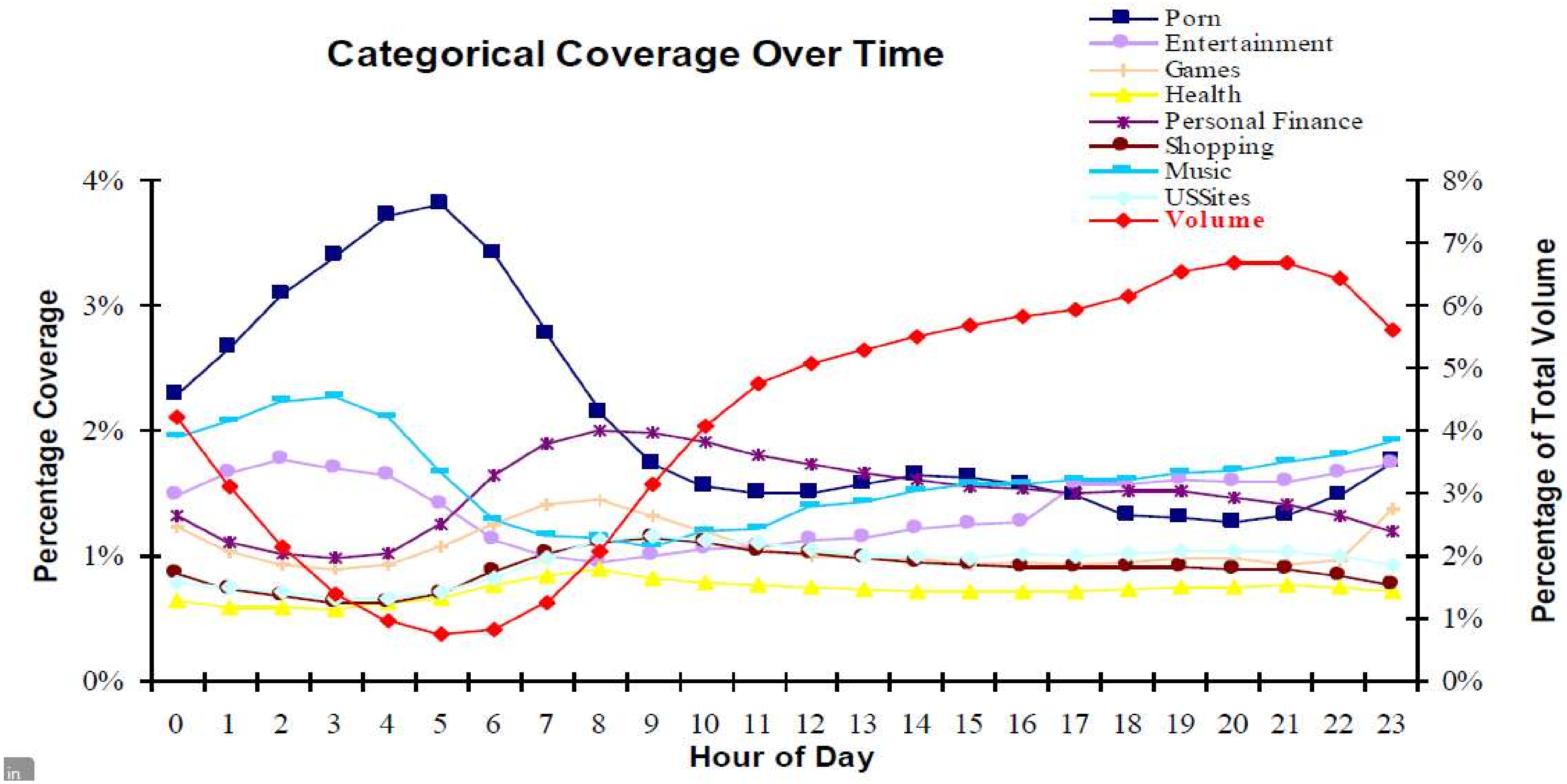
- Temporal trend of overall query volume
  - Hourly: 5-6 am lowest, 9-10 pm highest
  - Daily: drastic drop on Friday
  - Monthly: may be influenced by many other factors
- Trend of total queries matches with that of distinct queries
- Trend over months may be influenced by other factors than time

Category Breakdown



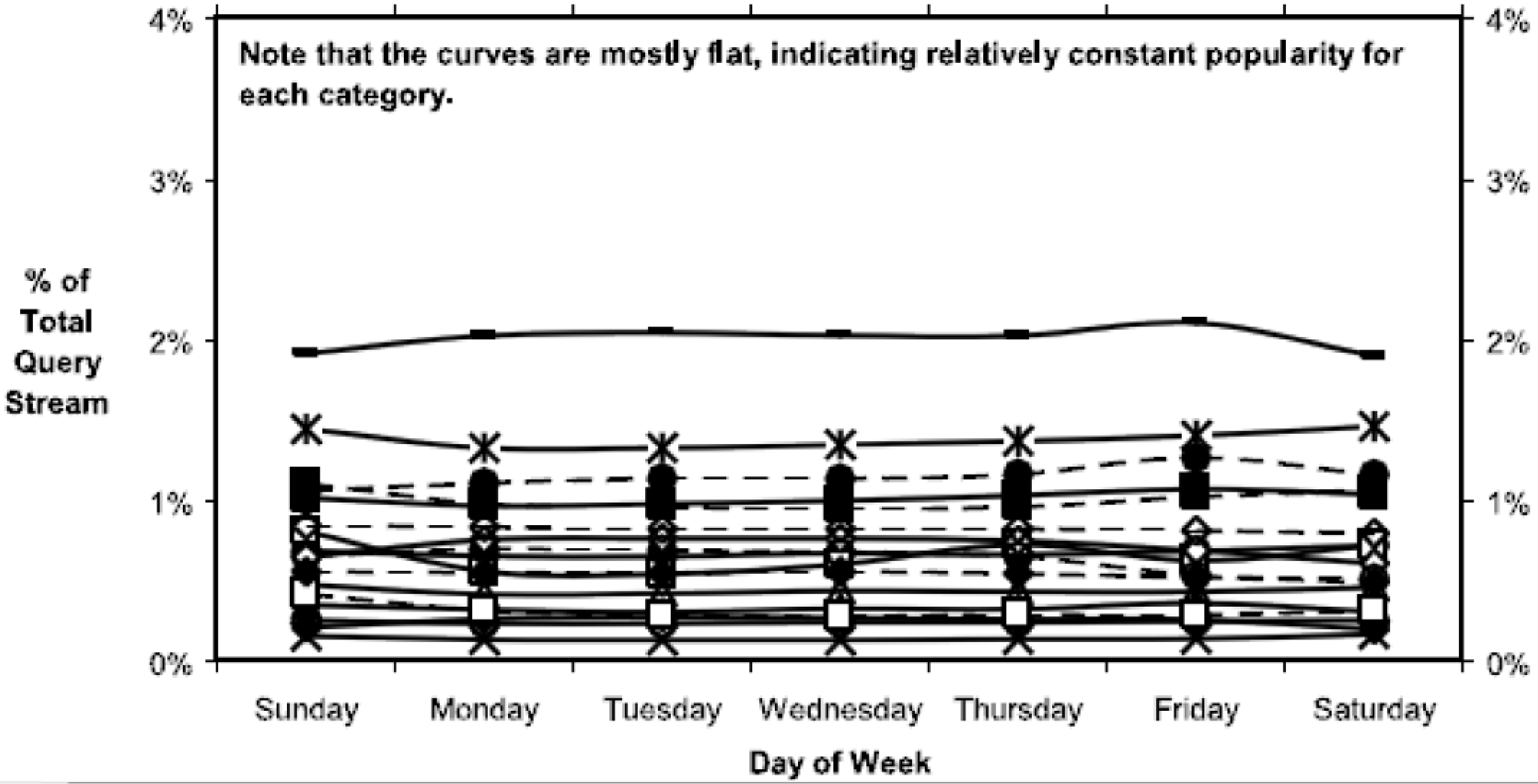
- Query lists for each category created by human editors
- Query stream classified by exactly matching each query to category lists
- Cover 13% of total query traffic

Category Popularity Over A Day



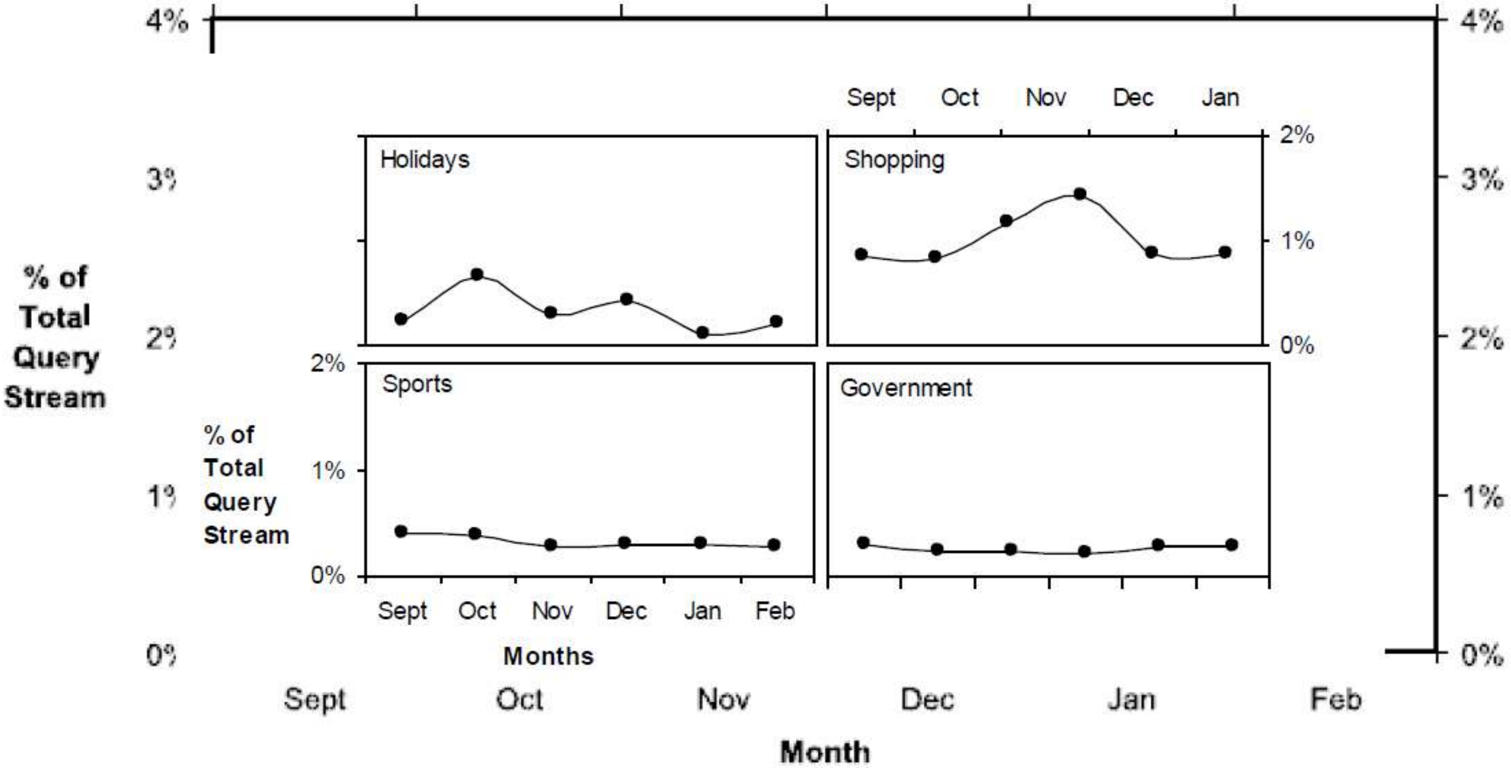
- Hourly: some topical categories (e.g., entertainment) vary substantially more than others

Category Popularity Over Week



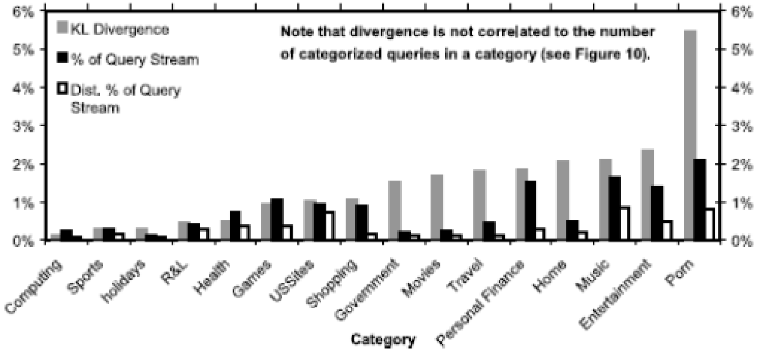
- Daily: categories are relatively stable

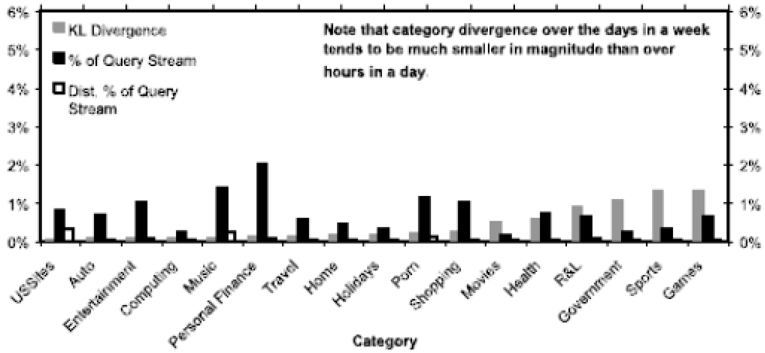
Category Popularity Over Six Months

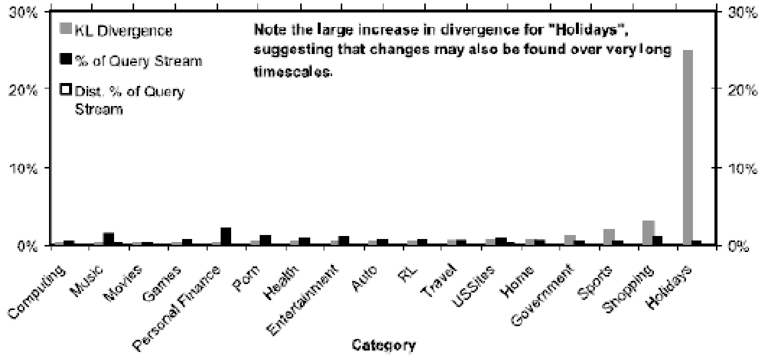


- Monthly: most categories are stable, while some others show seasonal changes (sports, holidays)

KL Divergence for Categories







- Whether distribution of queries within each category change over time
- $KL\big(p(q|t)\big|\big|p(q|c,t)\big) = \sum_q p(q|t) \log \frac{p(q|t)}{p(q|c,t)}$
- Categories with largest variance
  - Hourly: porn, entertainment, music, home
  - Daily: games, sports, government, research and learn
  - Monthly: holidays, shopping, sports, government

Query Temporal Models

- Query temporal models
  - Find periodic queries [Vlachos04]
  - Find burst queries [Dong10]
  - Find temporal relations between queries [Chien05]
- Temporal models are useful
  - Search results ranking (ranking based on recency, ranking based on periods)
  - Online advertisement

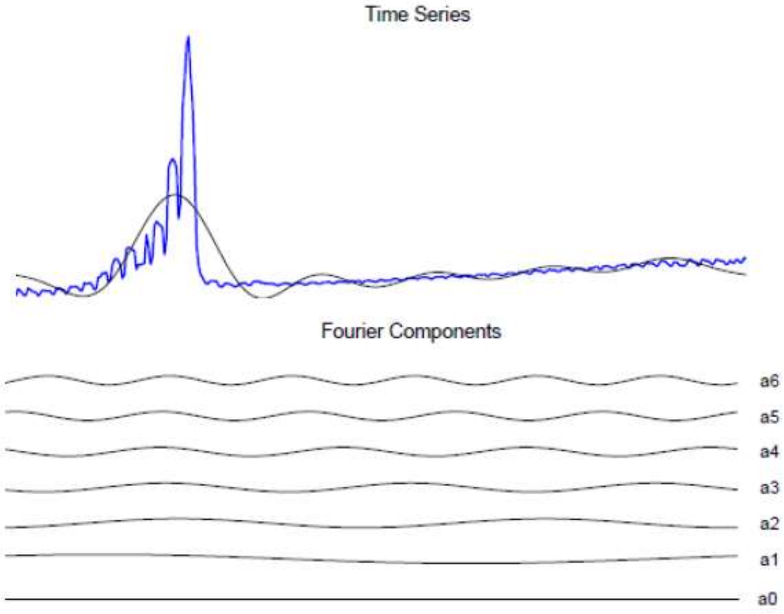
Periodic Query Identification Using Discrete Fourier Transformation [Vlachos04]

- Consider query stream as time series
- Represent time series as linear combination of complex sinusoids
  - $X\left(f_{k/N}\right) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) e^{-\frac{j2\pi kn}{N}}$, k=0, … N−1
- Find top K coefficients with highest magnitudes
- Represent power of each frequency

$$P\left(f_{\frac{k}{N}}\right) = \left\|X\left(f_{k/N}\right)\right\|^2, k$$

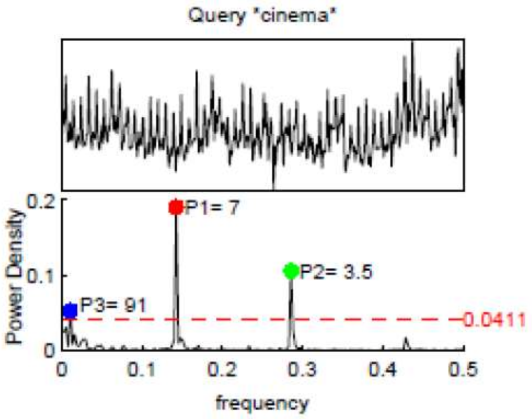
$$= 0,1, \dots, \left\lceil \frac{N-1}{2} \right\rceil$$

- Power spectrum for query

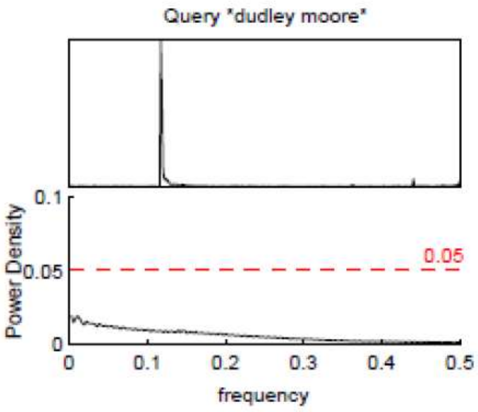


Periodic Queries vs Non-Periodic Queries

- Time periods can be found from power spectrum (period = 1 / frequency)
- Power spectrums of random (non-periodic) queries follow exponential distribution
- Hypothesis testing: find significant periods of query using power spectrum of query



Most significant period for query "cinema" is 7



No significant period found for query "dudley moore"

Burst Query Identification Using Language Model [Dong10]

- Calculating probabilities for generating query at current time slot $P(q \mid M_{C,t})$ $P(q \mid M_{Q,t})$
- Calculating probabilities for generating query from previous time slot to current time slot $P(q \mid M_{C,t-r_i})$ $P(q \mid M_{Q,t-r_i})$
- Calculating buzziness of query from two language models and linearly combining them

$$\text{buzz(q,t,C)} = \max_i \log P(q \mid M_{C,t}) - \log P(q \mid M_{C,t-r_i})$$

$$\text{buzz(q,t,Q)} = \max_i \log P(q \mid M_{Q,t}) - \log P(q \mid M_{Q,t-r_i})$$

Summary of Temporal Aspect of Queries

- Analysis of Query Temporal Trends
- Periodic Query Identification
- Burst Query Identification

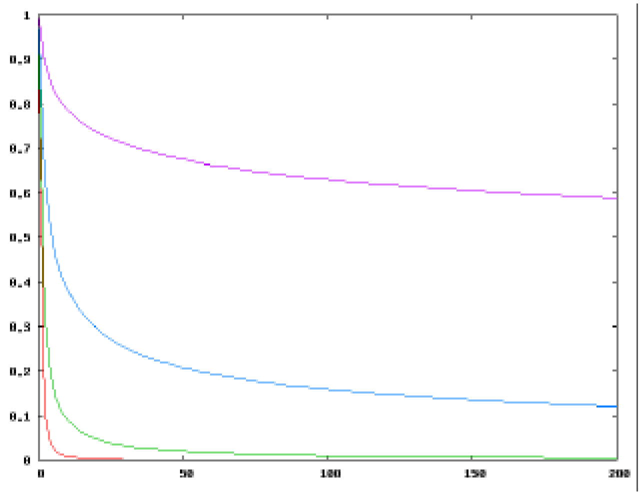
Overview Spatial Aspects of Queries

- Queries can be modeled from location perspective
- Two types
  - Local interest queries [Backstrom08]
    - Queries only interested by users at particular location
    - e.g., name of local high school, newspaper
  - Localizable queries [Welch08][Yi09]
    - Users at different locations may issue the same query, but referring to different things
    - e.g., pizza hut, house for rent

Modeling Local Interest Queries [Backstrom08]

- Identify and characterize geo-features of topics
  - Find center of geographic focus for topic
  - Determine if topic is tightly concentrated or spread diffusely geographically
- Answer two types of question
  - Given query, what is center and dispersion?
  - Given region, what are local queries?
- Applications
  - Business intelligence
  - Re-ranking search results

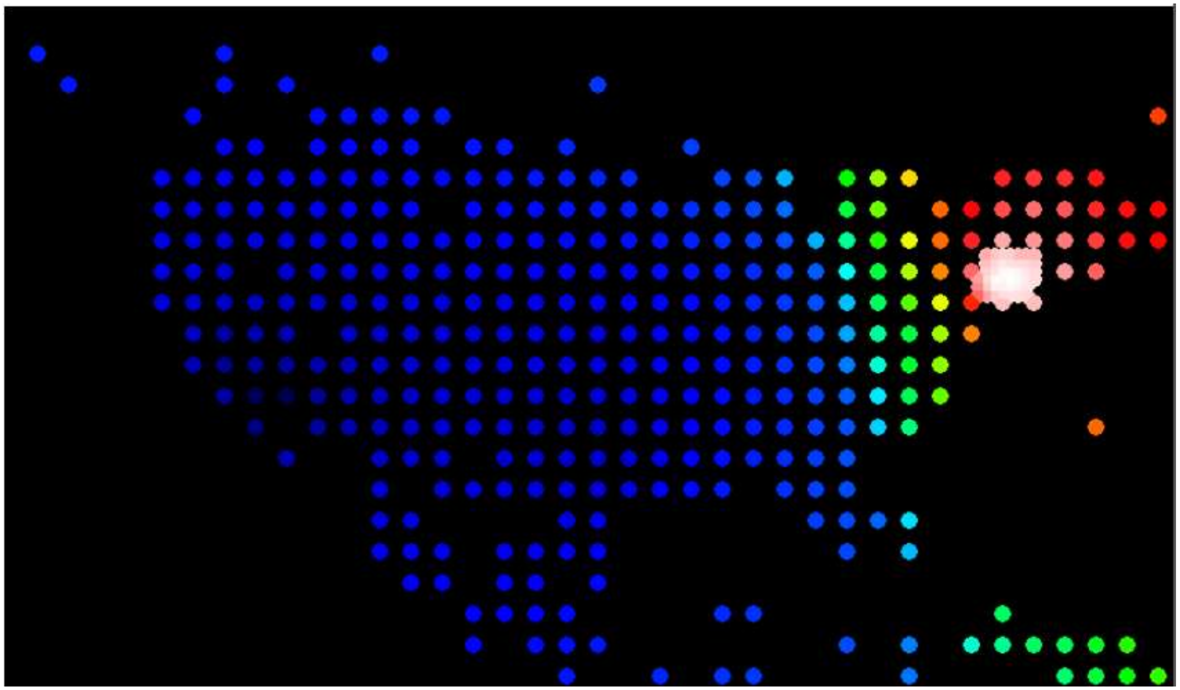
Probabilistic Model



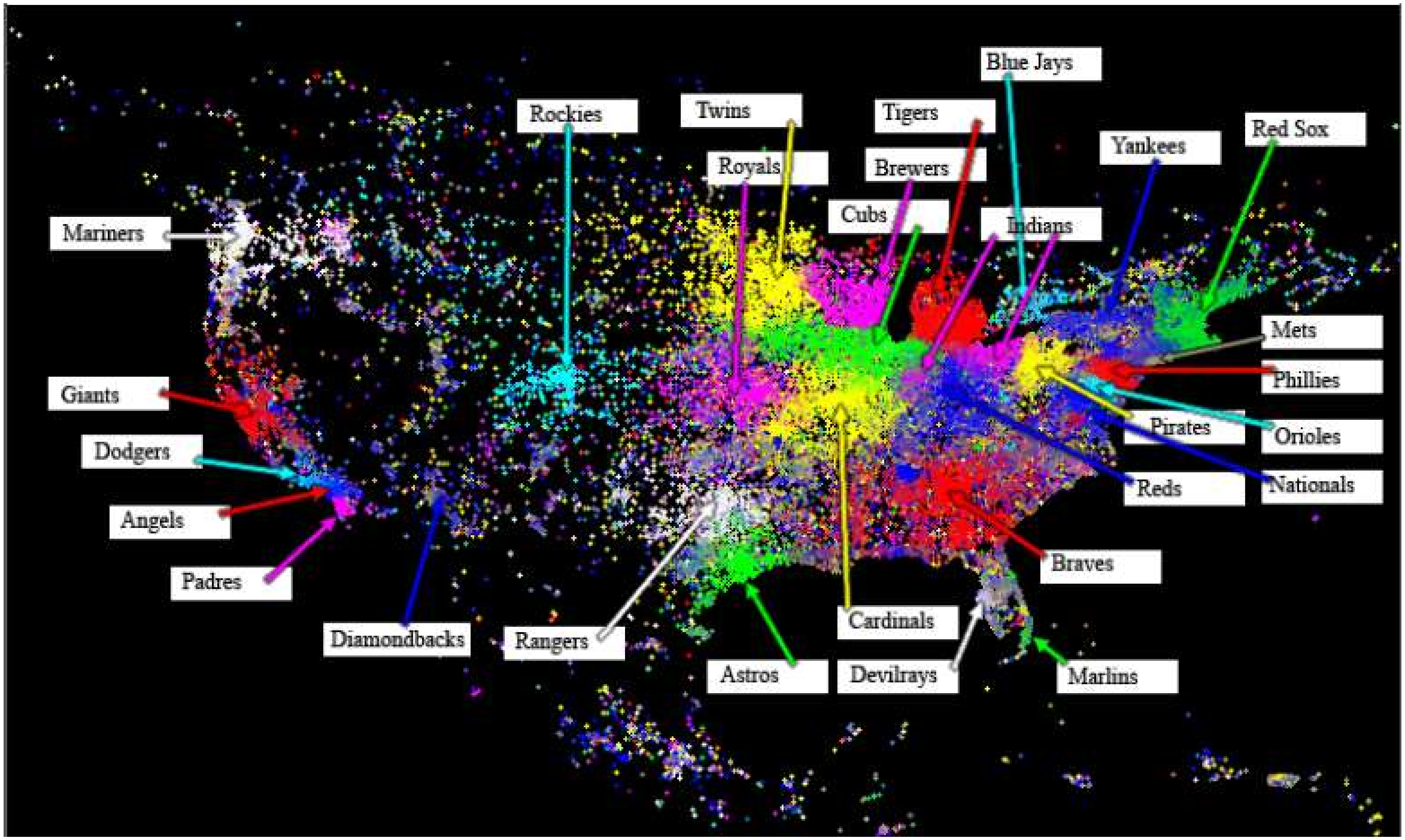
- Consider query topic *t*
  - e.g. ‘red sox’
- For each location *x*, query coming from *x* has probability p_x with respect to *t*
- There exists center *z*.
  - Probability is highest at z
  - *px* is decreasing function of ||x-z||
- Probability density function:
  - Query coming from x has probability $p_x = C d\text{-}\alpha$
  - Ranges from non-local (α = 0) to extremely local (large α)

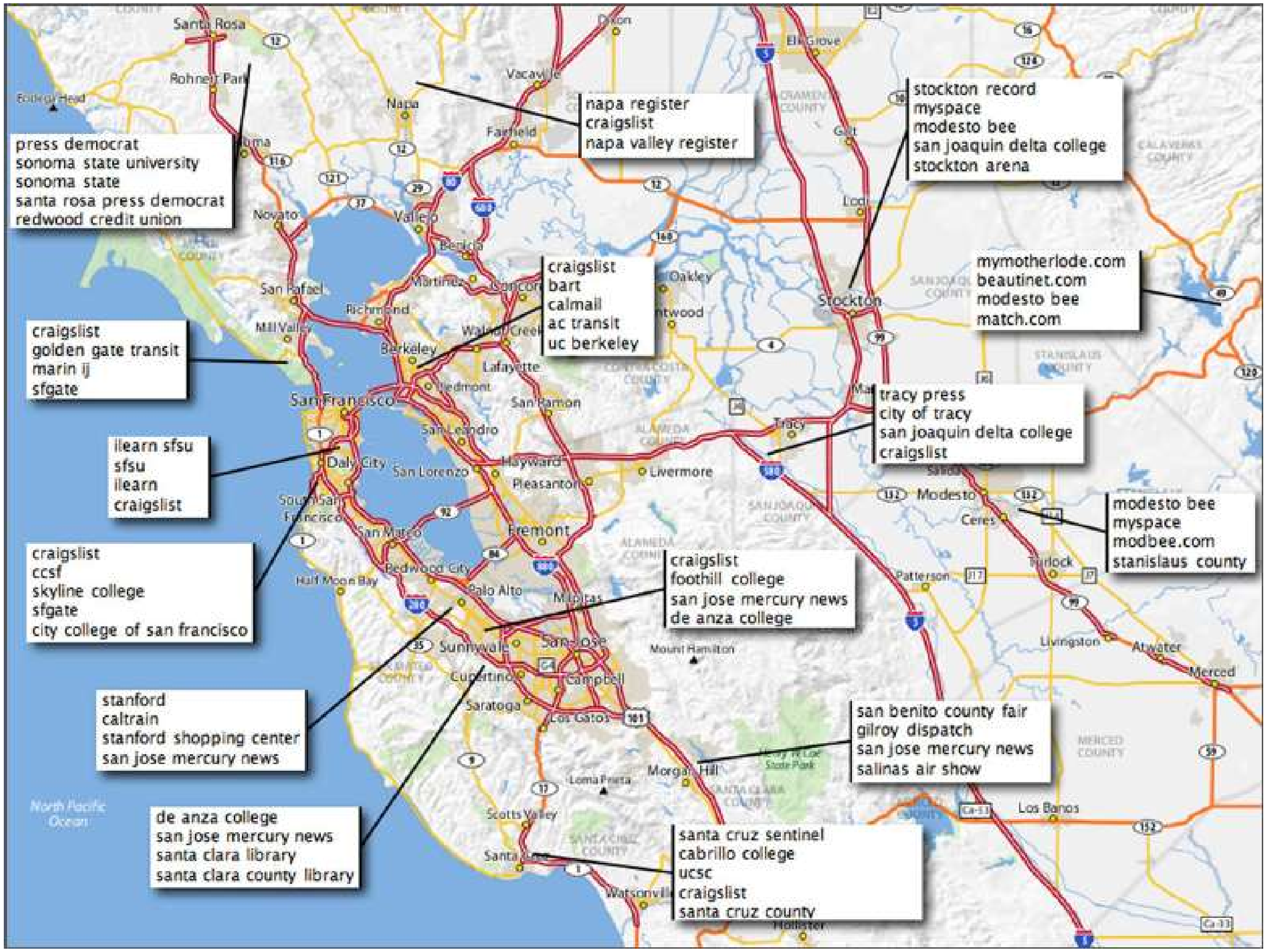
Algorithm

- Employing maximum likelihood estimation to estimate center, C and α
- Simple algorithm finds parameters which maximize likelihood
  - For given center, likelihood is unimodal and simple search algorithms find optimal C and α
  - Consider all centers on course mesh, optimize C and α for each center
  - Find best center, consider finer mesh



Baseball Team Queries





napa register
craigslist
napa valley register

stockton record
myspace
modesto bee
san joaquin delta college
stockton arena

press democrat
sonoma state university
sonoma state
santa rosa press democrat
redwood credit union

craigslist
bart
calmail
ac transit
uc berkeley

mymotherlode.com
beautinet.com
modesto bee
match.com

craigslist
golden gate transit
marin ij
sfgate

tracy press
city of tracy
san joaquin delta college
craigslist

ilearn sfsu
sfsu
ilearn
craigslist

modesto bee
myspace
modbee.com
stanislaus county

craigslist
ccsf
skyline college
sfgate
city college of san francisco

craigslist
foothill college
san jose mercury news
de anza college

stanford
caltrain
stanford shopping center
san jose mercury news

san benito county fair
gilroy dispatch
san jose mercury news
salinas air show

de anza college
san jose mercury news
santa clara library
santa clara county library

santa cruz sentinel
cabrillo college
ucsc
craigslist
santa cruz county

Identifying Localizable Queries

- Traditional work [Gravano03]
  - Using search results
  - If multiple locations distribute evenly in search result, the query is likely to be localizable query
- Recent work [Welch08][Yi09]
  - Using query log data
  - A localizable query is likely to appear as a sub query in other queries, associating with different locations
  - For example, some users may issue “car rental”, while others may issue “car rental california”, “car rental new york”, etc

Identifying Localizable Queries Using Query Log [Welch et al 08]

- An empirical study
  - Significant fraction of queries are localizable
  - Roughly 30%, but users only explicitly localize them about half of the time
  - Users exhibit consensus on which queries are localizable
- Approach
  - Identify candidate localizable queries
  - Select relevant features
  - Train and evaluate classifier

Identify Candidate Localizable Queries

- Use U.S. Census Bureau data as an address book
- For each query Q, look up the address book
  - If a match is found, the matched part is Q_l, the remaining part is Q_b
  - Q is a localized query of Q_b
- Aggregate all Q_l for each Q_b
  - The set of Q_l for each Q_b is denoted by $L(Q_b)$
  - Q_b is candidate localizable query if it often localized

Major Features of Classifier

- Localization ratio
  - How often query is localized
- Location distribution
  - Query should be localized with evenly distributed locations
- Click-through rates
  - Click-through rate should be higher in localized query

Summary of Spatial Aspect of Queries

- Two directions to analyze queries from location perspective
- Local interest queries: identifying center and dispersion of local interest queries
- Localizable queries: identifying localizable queries using term co-occurrences

Today’s Agenda

- Query Expansion, Refinement, and Suggestion
- Temporal and Spatial Aspects of Queries
- **Text Mining from Query Logs**

Outline of Text Mining from Query Logs

- Overview of Text Mining from Query Logs
- Named Entity Mining from Query Logs

Overview of Text Mining from Query Logs

- View search log data as `texts’
- Conduct text mining on log data
- Named entity mining
  - Entity extraction
  - Attribute extraction

Named Entity Mining from Query Logs

- To mine information about named entities in a class
- Examples
  - Cities: new york, los angeles, london, san francisco, dallas, boston, phoenix
  - Universities: harvard, stanford, michigan state university, oxford, mit, columbia, cambridge
- Over 70% queries contain named entities.

Methods for Named Entity Mining from Query Logs

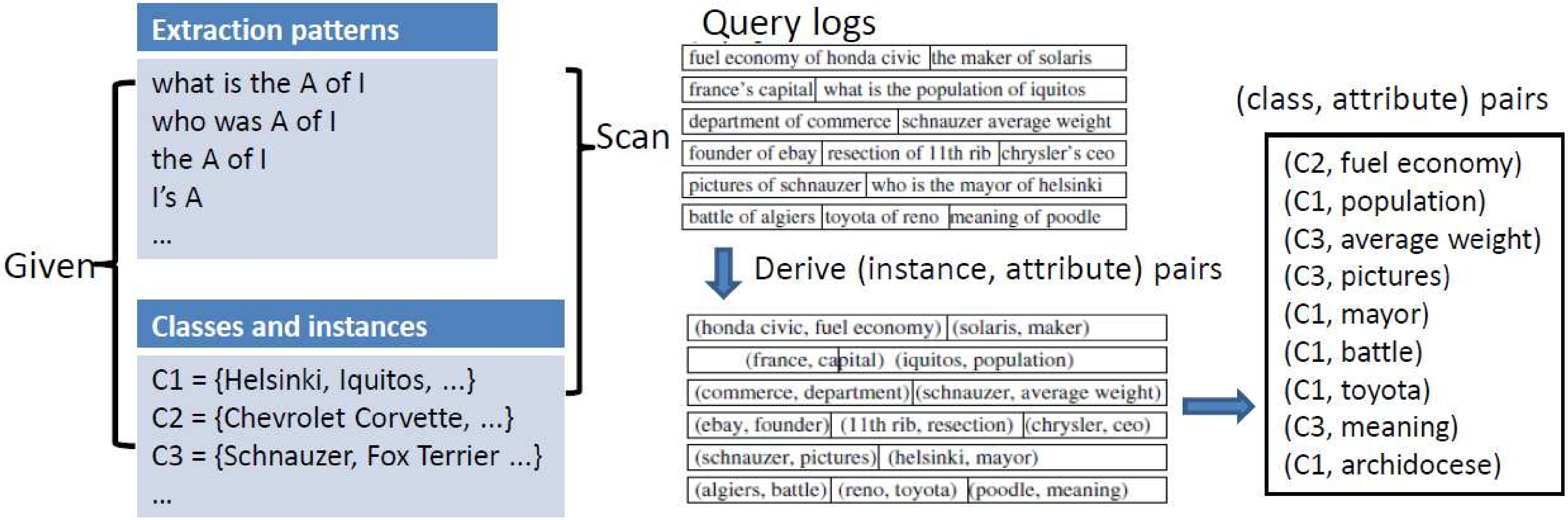
- Named entity mining using query log and weak supervision [Pasca07]
- Attribute mining using session data [Wang09]
- Named entity mining using topic model [Guo08, Xu08]

Named Entity Mining from Query Logs and Weak Supervision [Pasca07a]

- Assumption
  - Let C be a class, and I ∈ C be instances of the class.
  - If A is a prominent attribute of C, then a fraction of queries about I are likely to ask both A and I
- Framework
  - Step 1: attribute generation
  - Step 2: attribute filtering
  - Step 3: attribute ranking

Attribute Generation

- Match queries with a set of seed attributes (patterns)
- Derive a list of (instance, attribute) pairs
- Replace (instance, attribute) pairs with (class, attribute) pairs
- Count frequency for each (class, attribute) pair



Extraction patterns

what is the A of I
who was A of I
the A of I
I's A
...

Given

Classes and instances

C1 = {Helsinki, Iquitos, ...}
C2 = {Chevrolet Corvette, ...}
C3 = {Schnauzer, Fox Terrier ...}
...

Scan

Query logs

fuel economy of honda civic | the maker of solaris
france's capital | what is the population of iquitos
department of commerce | schnauzer average weight
founder of ebay | resection of 11th rib | chrysler's ceo
pictures of schnauzer | who is the mayor of helsinki
battle of algiers | toyota of reno | meaning of poodle

Derive (instance, attribute) pairs

(honda civic, fuel economy) | (solaris, maker)
(france, capital) (iquitos, population)
(commerce, department) | (schnauzer, average weight)
(ebay, founder) | (11th rib, resection) | (chrysler, ceo)
(schnauzer, pictures) | (helsinki, mayor)
(algiers, battle) | (reno, toyota) | (poodle, meaning)

(class, attribute) pairs

(C2, fuel economy)
(C1, population)
(C3, average weight)
(C3, pictures)
(C1, mayor)
(C1, battle)
(C1, toyota)
(C3, meaning)
(C1, archidocese)

Attribute Filtering and Ranking

- Discard attributes that are proper nouns or part of proper nouns
  - With a large Web corpus
- Remove too generic attributes
  - e.g., meaning, story, summary, pictures
- Remove near-duplicate attributes
  - Two attributes are considered redundant if they have small edit distance
- Ranking attributes based on their frequencies

Named Entity Mining Using Query Log Data and Topic Model [Guo09]

- Using Query Log Data (or Click-through Data)
- Using Topic Model
- Weakly Supervised Latent Dirichlet Allocation
- vs Pasca's work (named entity mining from log data, deterministic approach)

Offline Step 1: Seed and Query Log

final fantasy
Movie Game

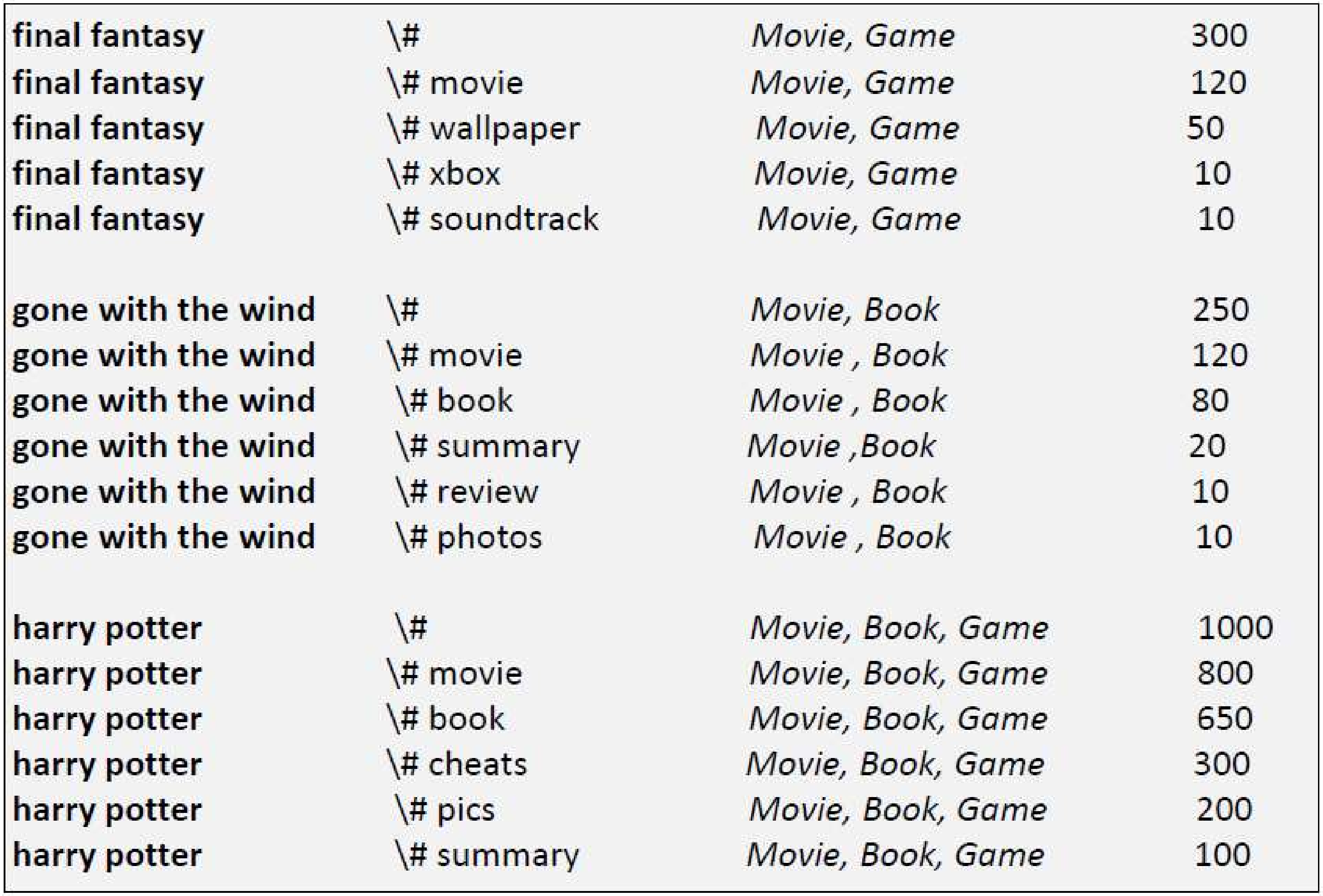
gone with the wind
Movie Book

harry potter
Movie Book Game

Named entity can belong to several classes
→ probabilistic approach

final fantasy	300
final fantasy movie	120
final fantasy wallpaper	50
gone with the wind movie	120
gone with the wind review	10
gone with the wind photos	10
harry potter	1000
harry potter book	650
gone with the wind book	80
gone with the wind summary	20
harry potter cheats	300
harry potter pics	200
harry potter summary	100
final fantasy xbox	10
final fantasy soundtrack	10
gone with the wind	250
harry potter movie	800
……	

Named Entity, Context, Class, and Frequency



final fantasy	\#	*Movie, Game*	300
final fantasy	\# movie	*Movie, Game*	120
final fantasy	\# wallpaper	*Movie, Game*	50
final fantasy	\# xbox	*Movie, Game*	10
final fantasy	\# soundtrack	*Movie, Game*	10
gone with the wind	\#	*Movie, Book*	250
gone with the wind	\# movie	*Movie , Book*	120
gone with the wind	\# book	*Movie , Book*	80
gone with the wind	\# summary	*Movie ,Book*	20
gone with the wind	\# review	*Movie , Book*	10
gone with the wind	\# photos	*Movie , Book*	10
harry potter	\#	*Movie, Book, Game*	1000
harry potter	\# movie	*Movie, Book, Game*	800
harry potter	\# book	*Movie, Book, Game*	650
harry potter	\# cheats	*Movie, Book, Game*	300
harry potter	\# pics	*Movie, Book, Game*	200
harry potter	\# summary	*Movie, Book, Game*	100

Pseudo Documents of Named Entities

final fantasy

\#	300
\# movie	120
\# wallpaper	50
\# xbox	10
\# soundtrack	10

Movie, Game

Labels of document:
Topics

gone with the wind

\#	250
\# movie	120
\# book	80
\# summary	20
\# review	10
\# photos	10

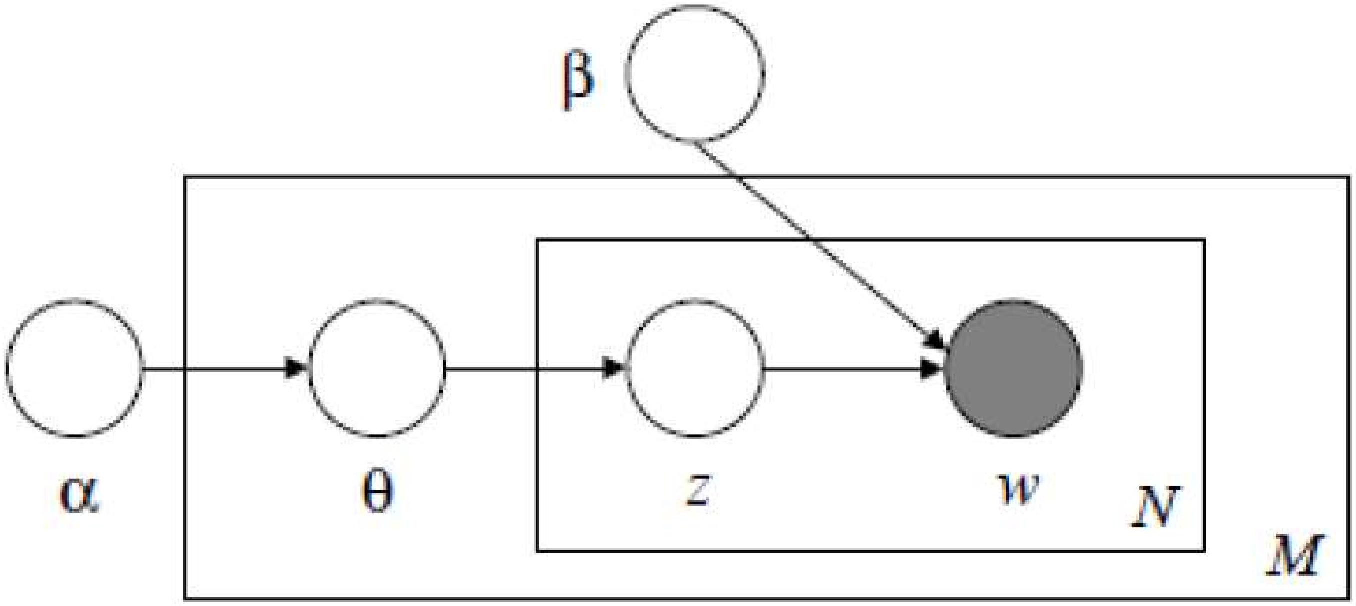
Movie, Book

harry potter

\#	1000
\# movie	800
\# book	650
\# cheats	300
\# pics	200
\# summary	100

Movie, Book, Game

Offline Step 2: Building Latent Dirichlet Allocation Model



z: *Movie, Book, Game*

w: \#, \# movie, \# book, ….

θ : distribution of classes for named entity

β : distribution of contexts for class

Offline: Weakly Supervised Latent Dirichlet Allocation

$$p(\mathcal{D}|\Theta)=\prod_{d=1}^{M}\int p(\boldsymbol{\theta}_d|\boldsymbol{\alpha})\Big(\prod_{n=1}^{N_d}\sum_{z_{dn}}p(z_{dn}|\boldsymbol{\theta}_d)p(w_{dn}|z_{dn},\boldsymbol{\beta})\Big)d\boldsymbol{\theta}_d$$

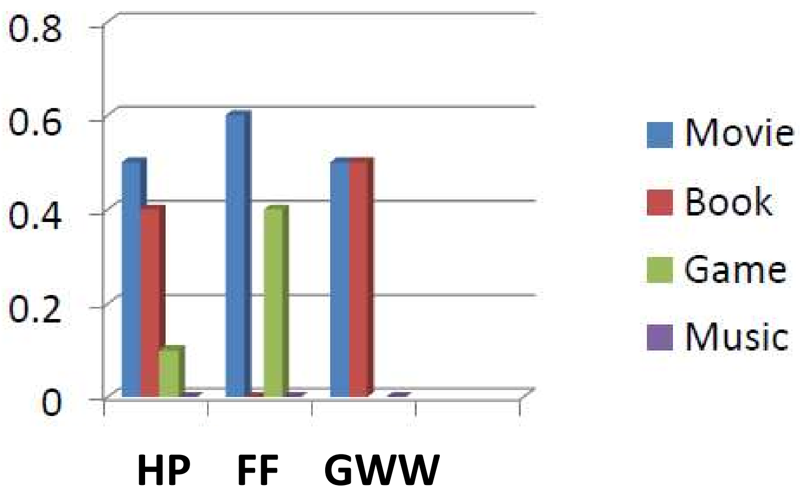
$$\log p(D\,|\,\Theta)+\lambda C(\Theta,y)$$

$$=\sum_{d=1}^{M}\log\int p(\boldsymbol{\theta}_d|\boldsymbol{\alpha})\Big(\prod_{n=1}^{N_d}\sum_{z_{dn}}p(z_{dn}|\boldsymbol{\theta}_d)p(w_{dn}|z_{dn},\boldsymbol{\beta})\Big)d\boldsymbol{\theta}_d$$

$$+\sum_{d=1}^{M}\lambda\sum_{i=1}^{K}y_{di}\bar{z}_{di}$$

$$\bar{z}_i=\frac{1}{N}\sum\nolimits_{n=1}^{N}z_n^i;$$

constraints



HP=Harry Potter, FF=Final Fantasy, GWW=Gone with the wind

Learned Probabilities

$P(w \mid z, \beta)$

\#	0.5
\# movie	0.2
\# review	0.1
\# wallpaper	0.1
\# photos	0.1

Movie

\#	0.8
\# book	0.1
\# summary	0.05
\# review	0.05

Book

\#	0.6
\# pics	0.2
\# cheats	0.1
\# xbox	0.05
\# soundtrack	0.05

Game

$P(z \mid \theta)$

final fantasy

Movie	0.5
Game	0.5

gone with the wind

Movie	0.6
Book	0.4

harry potter

Movie	0.6
Book	0.3
Game	0.1

Online: Inference

kung fu panda

\#	250
\# movie	100
\# wallpaper	20
\# walkthrough	10
\# review	10

Movie, Game?

beautiful mind

\#	200
\# movie	150
\# summary	60
\# review	40
\# book	80

Movie, Book?

kung fu panda

Movie	0.9
Game	0.1

beautiful mind

Movie	0.7
Book	0.3

Summary of Text Mining from Query Logs

- Named entity mining using query log and patterns
- Attribute mining using session data
- Named entity mining using topic model

Take-away Messages

- At macro level
  - Query Statistics
  - Query Temporal Trend Analysis
- At micro level
  - Query Classification (Intent Understanding)
  - Task, Topic, Entity & Attribute, Time, Location
  - Offline: Large Scale Log Mining
  - Online: Query Classification, Query Expansion, Refinement, Suggestion
- Click-through data and session data are very useful

Further Reading

- Daxin Jiang, Jian Pei, Hang Li. Web Search/Browse Log Mining: Challenges, Methods, and Applications. Tutorial at WWW 2010
- Daxin Jiang, Jian Pei, Hang Li. Mining Search and Browse Logs for Web Search: A Survey. ACM Transactions on Computational Logic, Vol. V, No. N, February 2013, Pages 1–42.
- Maristella Agosti, Franco Crivellari, Giorgio Maria Di Nunzio. Web log analysis: a review of a decade of studies about information acquisition, inspection and interpretation of user interaction. Data Min Knowl Disc (2012) 24:663–696
- Fabrizio Silvestri. Mining Query Logs: Turning Search Usage Data into Knowledge. Foundations and Trends in Information Retrieval. Vol. 4, Nos. 1–2 (2010) 1–174
- Marius Pasca. Tutorial. Web Search Queries as a Corpus. ACL 2011
- Ricardo Baeza-Yates, Fabrizio Silvestri. Query Log Mining.

Preview of Lecture 23: Document Understanding by Log Mining

- Motivation
- Enriched models using log data
- Tackling sparsity
- Application examples

Disclaimers

- This course represents opinions of the instructor only. It does not reflect views of Microsoft or any other entity (except of authors from whom the slides have been borrowed).
- Algorithms, techniques, features, etc mentioned here might or might not be in use by Microsoft or any other company.
- Lot of material covered in this course is borrowed from slides across many universities and conference tutorials. These are gratefully acknowledged.

Thanks!