Mining Deep Web Repositories

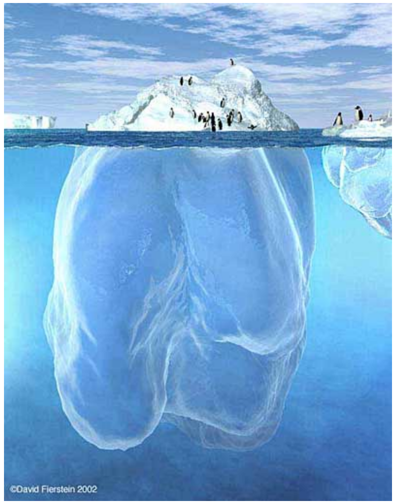
Nan Zhang, George Washington University
Gautam Das, University of Texas at Arlington and Qatar Computing Research Institute

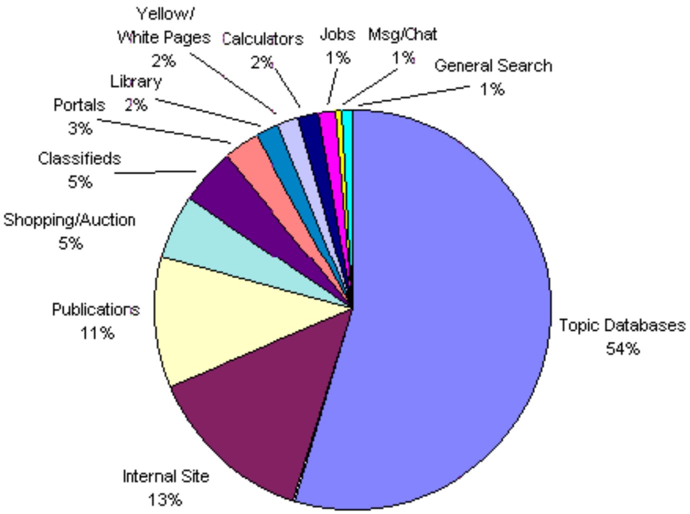
Outline

- Introduction
- Resource Discovery and Interface Understanding
- Technical Challenges for Data Mining
- Crawling
- Sampling
- Data Analytics
- Final Remarks

The Deep Web

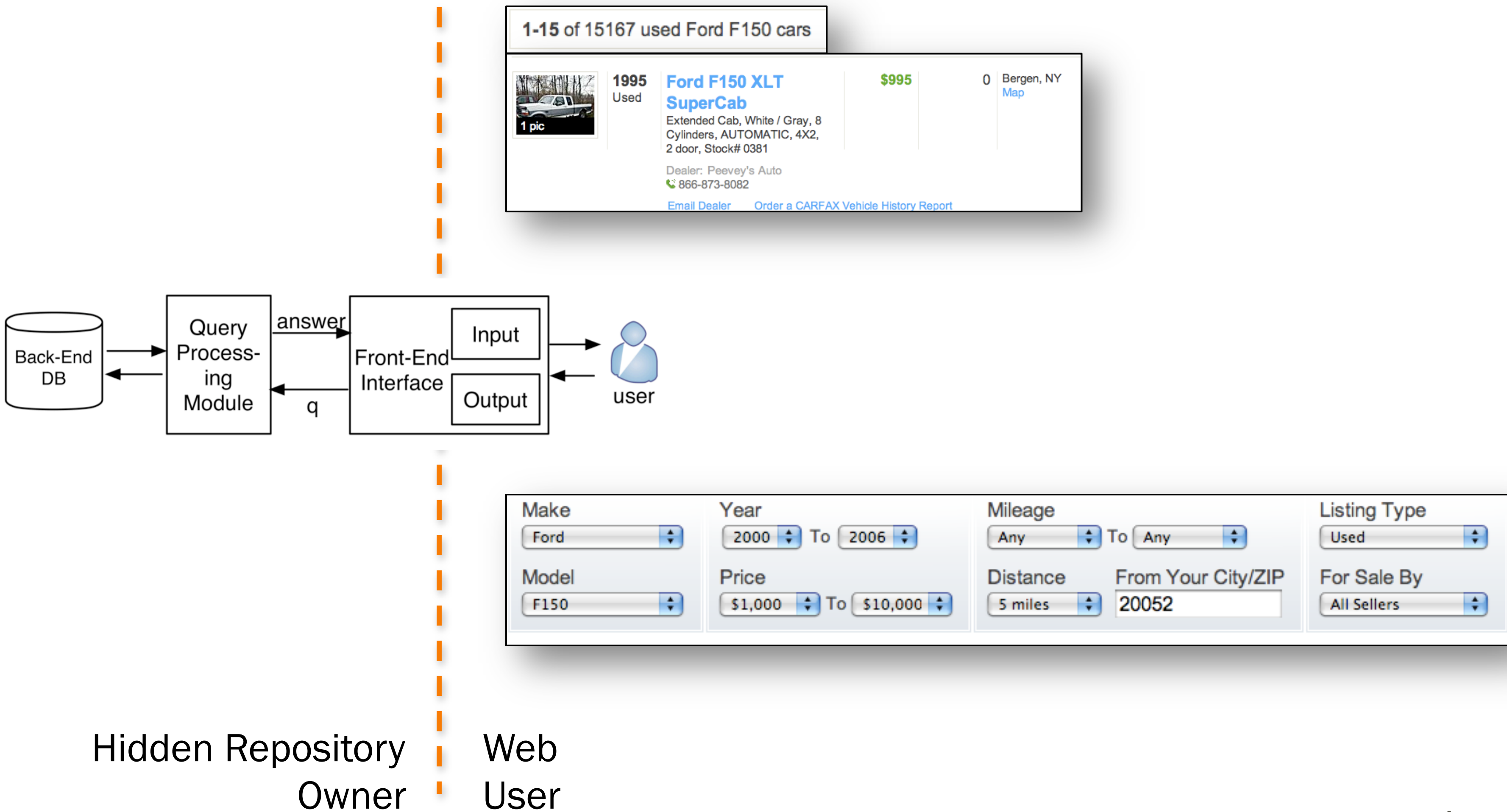
- Deep Web vs Surface Web
  - Dynamic contents, unlinked pages, private web, contextual web, etc
  - Estimated size: 91,850 vs 167 tera bytes[1], hundreds or thousands of times larger than the surface web[2]





[1] SIMS, UC Berkeley, How much information? 2003
[2] Bright Planet, Deep Web FAQs, 2010, http://www.brightplanet.com/the-deep-web/

Hidden Web Repositories



Deep Web Repository: Example I

Enterprise Search Engine's Corpus

Unstructured data | Keyword search | Top-k

Asthma

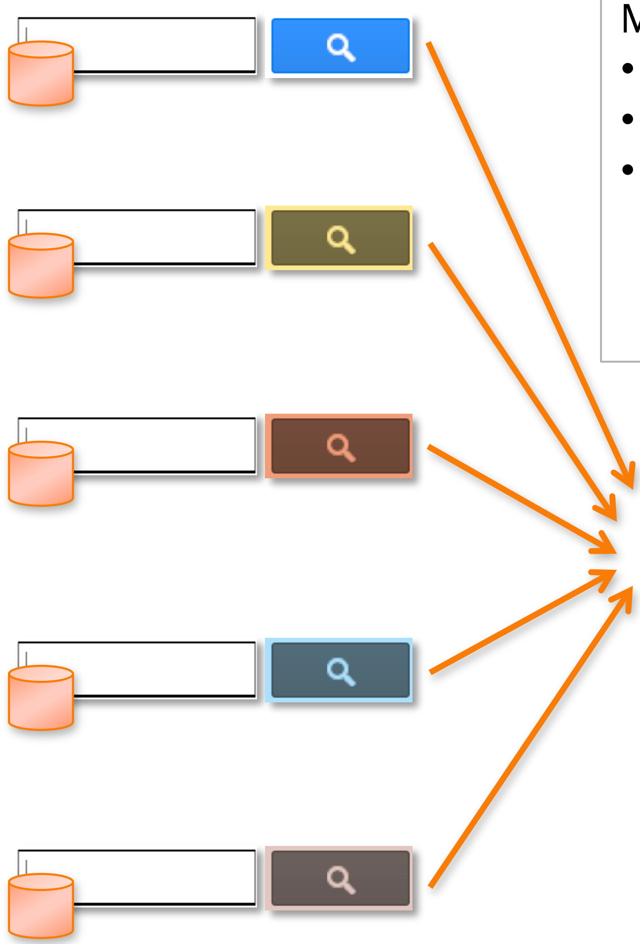
CDC - **Asthma** and Allergies - Prevention of Occupational ...
ASTHMA AND ALLERGIES. Prevention of Occupational **Asthma**: Introduction. **...** Smith AM, Bernstein DI. Management of work-related **asthma**. **...**
www.cdc.gov/niosh/topics/asthma/OccAsthmaPrevention.html
[More results from www.cdc.gov/niosh/topics/asthma]

Lower Airway Rhinovirus Burden and the Seasonal Risk of **Asthma** Exacerbation.
Denlinger LC, Sorkness RL, Lee WM, Evans M, Wolff M, Mathur S, Crisafi G, Gaworski K, Pappas
Am J Respir Crit Care Med. 2011 Aug 4. [Epub ahead of print]
PMID: 21816938 [PubMed - as supplied by publisher]
Related citations

First Aid/CPR/AED - Professional Rescuers
... one- and two-rescuer); AED; Optional training in use of epinephrine auto-injectors and **asthma** inhalers available. Course ...

- Brand Name: Foradil Aerolizer
  Generic Name: Formoterol Fumarate Inhalation Powder
- Brand Name: Qvar
  Generic Name: Beclomethasone Dipropionate HFA

Exploration: Example I



Metasearch engine
- Discovers deep web repositories of a given topic
- Integrate query answers from multiple repositories
- For result re-organization, evaluate the quality of each repository through data analytics and mining
  - e.g., how large is the repository?
  - e.g., clustering of documents

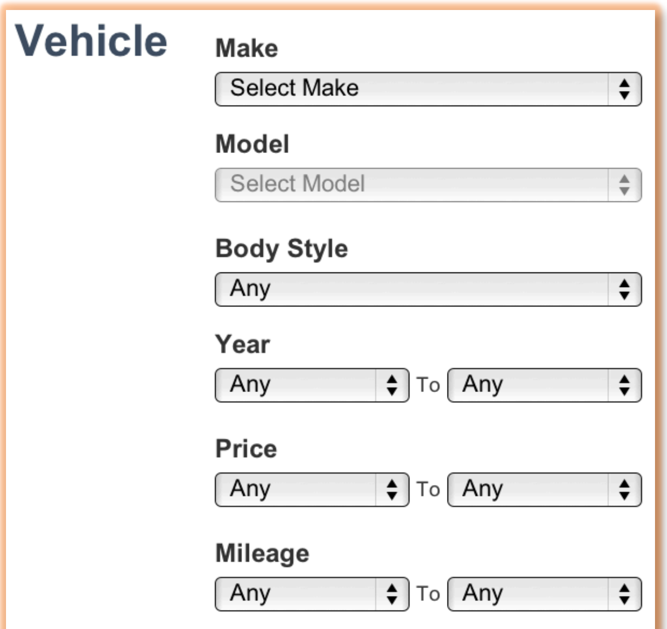
Example II

Yahoo! Auto, other online e-commerce websites

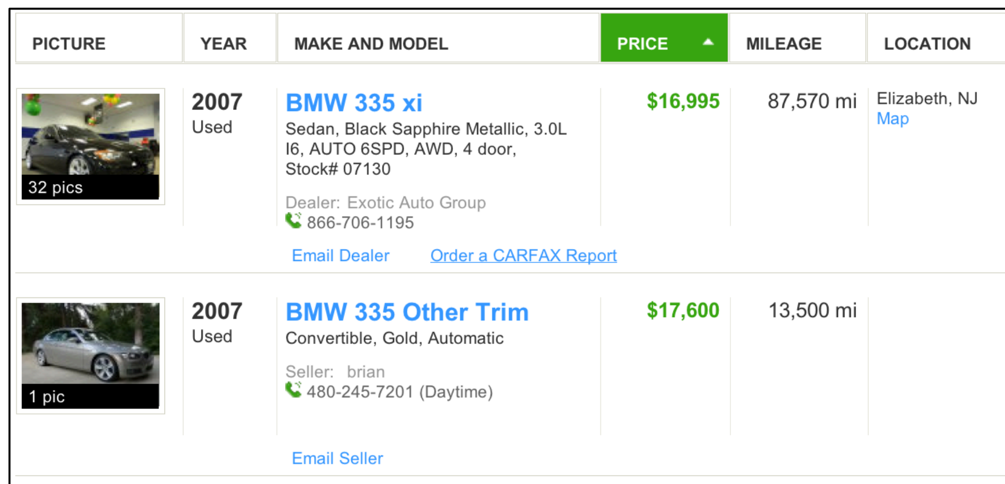
Structured data

Form-like search

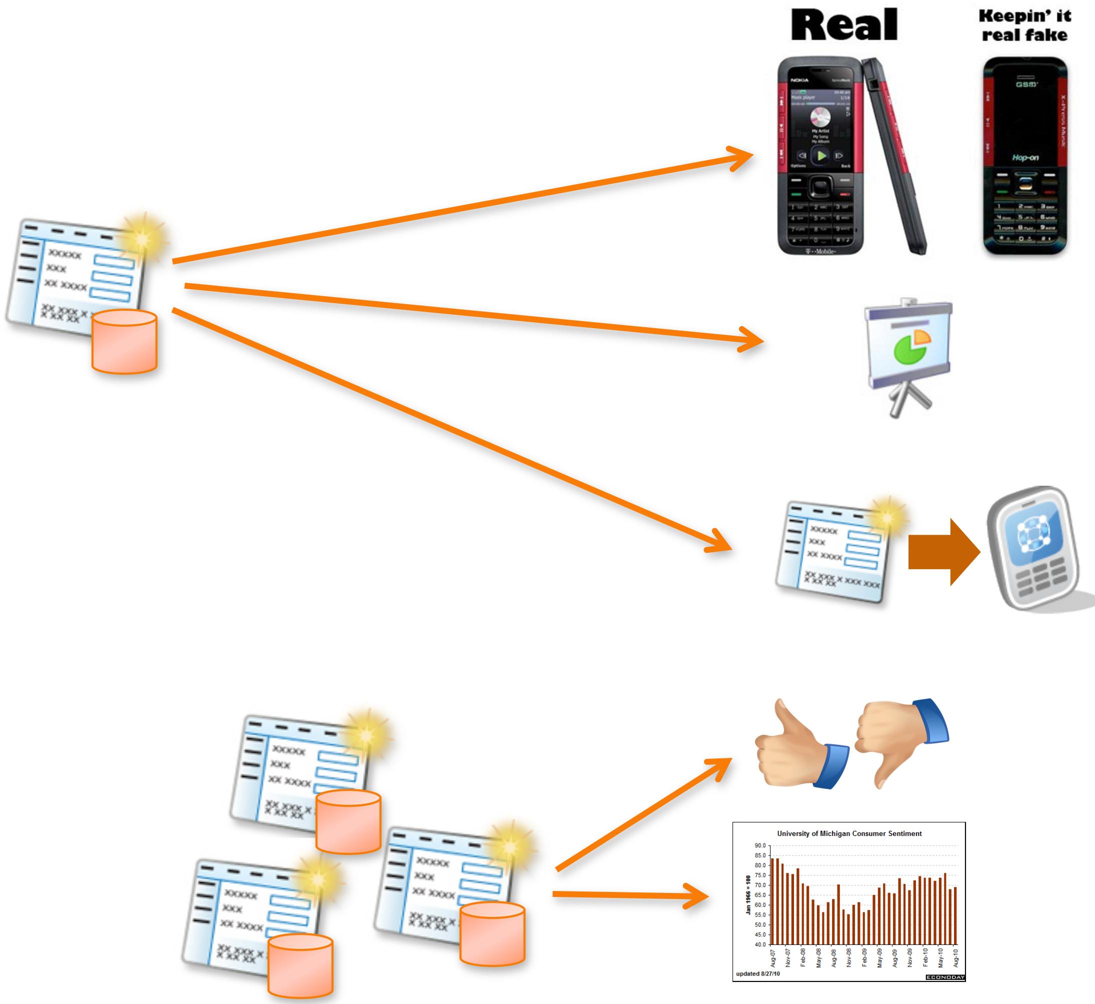
Top-1500



YAHOO! AUTOS



Exploration: Example II



Third-party analytics & mining of an individual repository

- Price distribution
- Price anomaly detection
- Classification: fake or real?

Third-party mining of multiple repositories

- Repository comparison
- Consumer behavior analysis

Main Tasks

- Resource discovery
- Data integration
- Single-/Cross- site mining

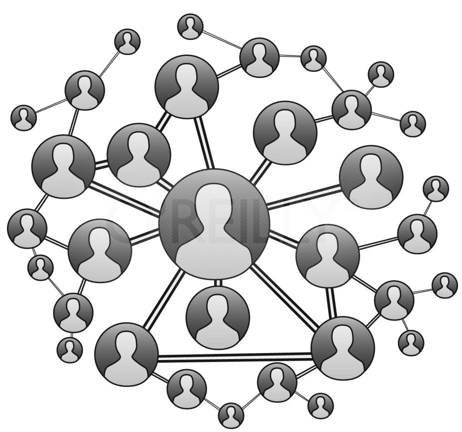
KAYAK
TERAPEAK
pricegrabber
Smart Shopping Anytime Anywhere

Example III

Semi-structured data

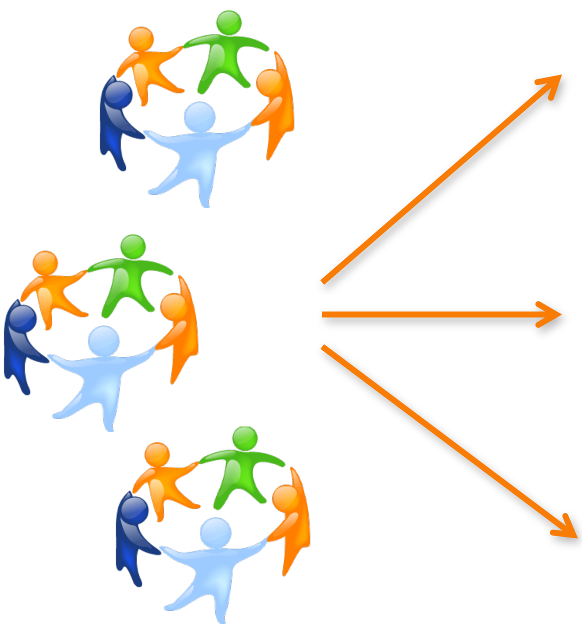
Graph browsing

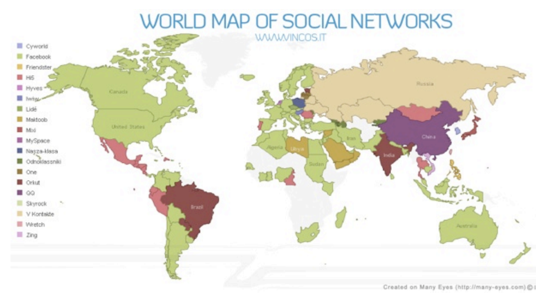
Local view

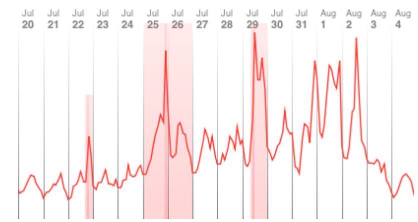


Picture from Jay Goldman, Facebook Cookbook, O'Reiley Media, 2008.

Exploration: Example III







TweetEffect beta

TWITALYZER

PageLever

For commercial advertisers:
- Market penetration of a social network
- “buzz words” tracking

For private detectors:
- Find pages related to an individual

For individual page owners:
- Understand the (relative) popularity or followers of ones own page
- Understand how new posts affect the popularity
- Understand how to promote the page

Main Tasks: resource discovery and data integration less of a challenge, analytics and mining of very large amounts of data becomes the main challenge.

Summary of Main Tasks/Obstacles

- Find where the data are
  - Resource discovery: find URLs of deep web repositories
  - Required by: Metasearch engine, shopping website comparison, consumer behavior modeling, etc.
- Understand the web interface
  - Required by almost all applications.
- Mine the underlying data
  - Through crawling, sampling, and/or analytics
  - Required by: Metasearch engine, keep it real fake, price prediction, universal mobile interface, shopping website comparison, consumer behavior modeling, market penetration analysis, social page evaluation and optimization, etc.

Covered by many recent tutorials [Weikum and Theobald PODS 10, Chiticariu et al SIGMOD 10, Dong and Nauman VLDB 09, Franklin, Halevy and Maier VLDB 08]

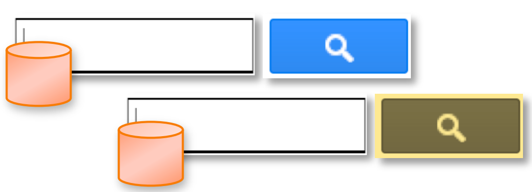
Demoed by research prototypes and product systems

DBLife WEBTABLES

TEXTRUNNER

Arnetminer Cazoodle

PATENTSSEARCHER



Focus of This Tutorial

- Brief Overview of:
  - Resource discovery
  - Interface understanding
  - i.e., where to, and how to issue a search query to a deep web repository?
- Our focus: Mining through crawling, sampling, analytics

Which individual search and/or browsing requests should a third-party explorer issue to the web interface of a given deep web repository, in order to enable efficient data mining?

Outline

- Introduction
- Resource Discovery and Interface Understanding
- Technical Challenges for Data Exploration
- Crawling
- Sampling
- Data Analytics
- Final Remarks

Resource Discovery

- Objective: discover resources of “interest”
  - Task 1: is an URL of interest?
    - Criteria A: is a deep web repository
    - Criteria B: belongs to a given topic
  - Task 2: Find all interesting URLs
- Task 1, Criteria A
  - Transactional page search [LKV+06]
    - Pattern identification – e.g., “Enter keywords”, form identification
    - Synonym expansion – e.g., “Search” + “Go” + “Find it”
- Task 1, Criteria B:
  - Learn by example
- Task 2
  - Topic distillation based on a search engine
    - e.g., “used car search”, “car * search”
    - Alone not suffice for resource discovery [Cha99]
  - Focused/Topical “Crawling”
    - Priority queue ordered by importance score
    - Leveraging locality
    - Often irrelevant pages could lead to relevant ones
      - Reinforcement learning, etc.

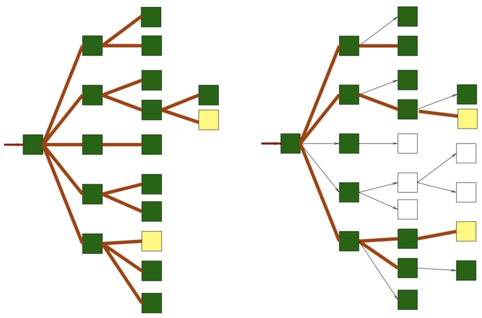


Figure from [DCL+00]

[DCL+00] M. Diligenti, F. M. Coetzee, S. Lawrence, C. L. Giles, and M. Gori, "Focused crawling using context graphs", VLDB, 2000.

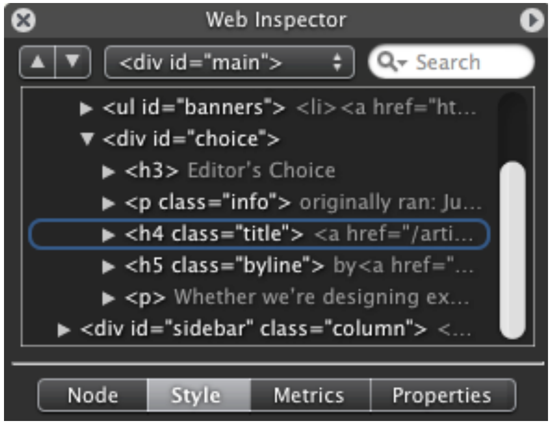
[LKV+06] Y. Li, R. Krishnamurthy, S. Vaithyanathan, and H. V. Jagadish, "Getting Work Done on the Web: Supporting Transactional Queries", SIGIR, 2006.

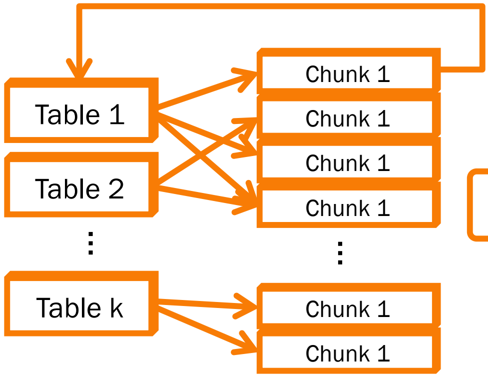
[Cha99] S. Chakrabarti, "Recent results in automatic Web resource discovery", ACM Computing Surveys, vol. 31, 1999.

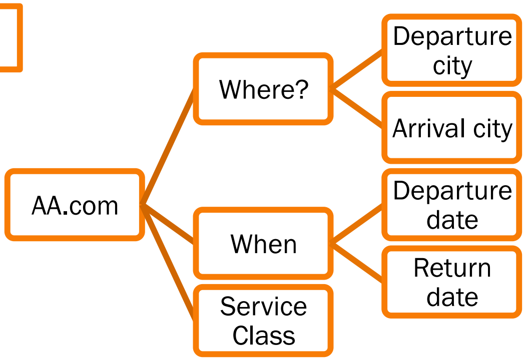
Interface Understanding

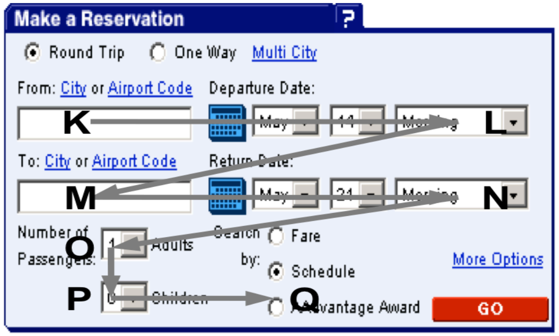
Modeling Web Interface

- Generally easy for keyword search interface, but can be extremely challenging for others (e.g., form-like search, graph-browsing)
- What to understand?
  - Structure of a web interface
- Modeling language
  - Flat model e.g., [KBG+01]
  - Hierarchical model e.g., [ZHC04, DKY+09]
- Input information
  - HTML Tags e.g., [KBG+01]
  - Visual layout of an interface e.g., [DKY+09]







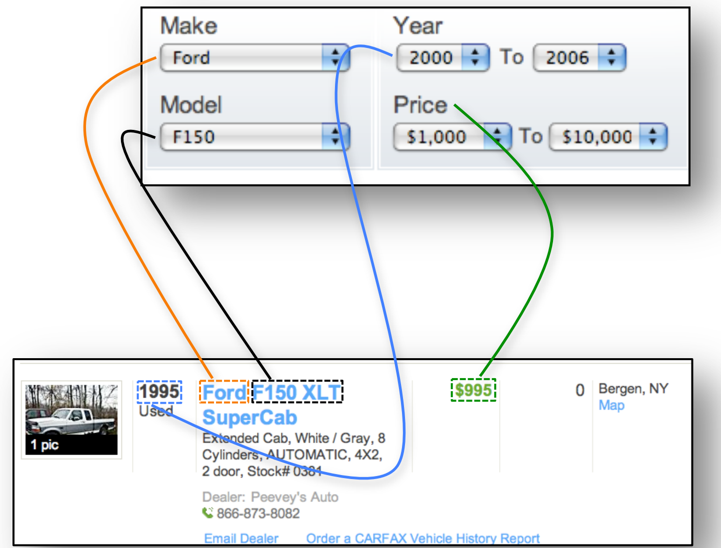


[KBG+01] O. Kaljuvee, O. Buyukkokten, H. Garcia-Molina, and A. Paepcke, "Efficient Web Form Entry on PDAs", WWW 2001.
[ZHC04] Z. Zhang, B. He, and K. C.-C. Chang, "Understanding Web Query Interfaces: Best-Effort Parsing with Hidden Syntax", SIGMOD 2004
[DKY+09] E. C. Dragut, T. Kabisch, C. Yu, and U. Leser, "A Hierarchical Approach to Model Web Query Interfaces for Web Source Integration", VLDB, 2009.

Interface Understanding

Schema Matching

- What to understand?
  - Attributes corresponding to input/output controls on an interface
- Modeling language
  - Map schema of an interface to a mediated schema (with well understood attribute semantics)
- Key Input Information
  - Data/attribute correlation [SDH08, CHW+08]
  - Human feedback [CVD+09]
  - Auxiliary sources [CMH08]



[CHW+08] M. J. Cafarella, A. Halevy, D. Z. Wang, E. Wu, and Y. Zhang, "WebTables: exploring the power of tables on the web", VLDB, 2008.
[SDH08] A. D. Sarma, X. Dong, and A. Halevy, "Bootstrapping Pay-As-You-Go Data Integration Systems", SIGMOD, 2008.
[CVD+09] X. Chai, B.-Q. Vuong, A. Doan, and J. F. Naughton, "Efficiently Incorporating User Feedback into Information Extraction and Integration Programs", SIGMOD, 2009.
[CMH08] M. J. Cafarella, J. Madhavan, and A. Halevy, "Web-Scale Extraction of Structured Data", SIGMOD Record, vol. 37, 2008.

Related Tutorials

- [FHM08] M. Franklin, A. Halevy, and D. Maier, "A First Tutorial on Dataspaces", VLDB, 2008.
- [GM08] L. Getoor and R. Miller, "Data and Metadata Alignment: Concepts and Techniques", ICDE, 2008.
- [DN09] X. Dong and F. Nauman, "Data fusion - Resolving Data Conflicts for Integration", VLDB, 2009.
- [CLR+10] L. Chiticariu, Y. Li, S. Raghavan, and F. Reiss, "Enterprise Information Extraction: Recent Developments and Open Challenges", SIGMOD, 2010.
- [WT10] G. Weikum and M. Theobald, "From Information to Knowledge: Harvesting Entities and Relationships from Web Sources", PODS, 2010.

Outline

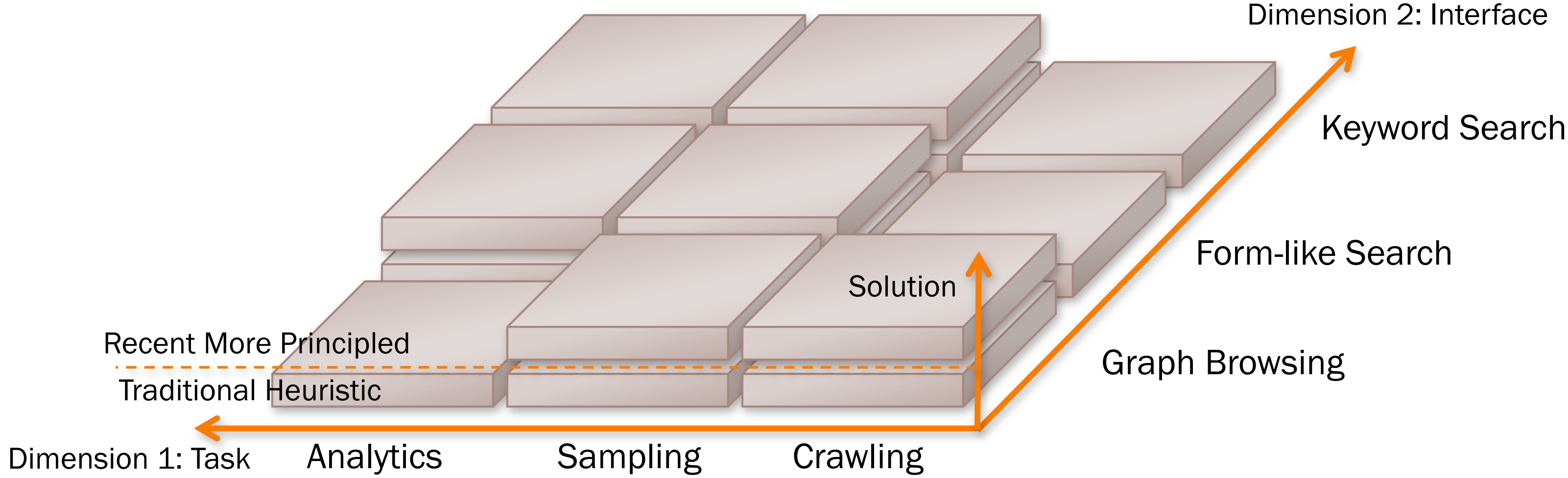
- Introduction
- Resource Discovery and Interface Understanding
- Technical Challenges for Data Mining
- Crawling
- Sampling
- Data Analytics
- Final Remarks

Mining a Deep Web Repository

Once the interface is properly understood…

- Assume that we are now given
  - A URL for a deep web repository
  - A wrapper for querying the repository (still limited by what queries are accepted by the repository – see next few slides)
- What's next?
  - We still need to address the following challenge: which queries or browsing requests should we issue in order to efficiently support data mining?
- Main source of challenge
  - restrictions on query interfaces
  - Orthogonal to the interface understanding challenge, and remains even after an interface is fully understood.
  - e.g., how to estimate COUNT(*) through an SPJ interface

Problem Space and Solution Space



Problem Space

Solution Space

Around 2000

~ 2005 - now

Traditional Heuristic Approaches

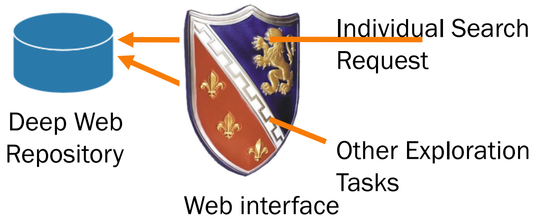
Recent Approaches with Theoretical Guarantees

- e.g., seed-query based bootstrapping for crawling
- e.g., query sampling for repository sampling
- No guarantee on query cost, accuracy, etc.

- e.g., performance-bounded crawlers
- e.g., unbiased samplers and aggregate estimators
- Techniques built upon sampling theory, etc.

Dimension 1. Task

- Crawling
  - Objective: download as many elements of interest (e.g., documents, tuples, metadata such as domain values) from the repository as possible.
  - Applications: building web archives, private directors, etc.
- Sampling
  - Draw sample elements from a repository according to a pre-determined distribution (e.g., uniform distribution for simple random sampling)
  - Why? Because crawling is often impractical for very large repositories because of practical limitations on the number of web accesses.
  - Collected sample can be later used for analytical processing, mining, etc.
  - Applications: Search-engine quality evaluation for meta-search-engines, price distribution, etc.
- Data Analytics
  - Directly support online analytics over the repository
  - Key Task: efficiently answer aggregate queries (COUNT, SUM, MIN, MAX, etc.)
  - Overlap with sampling, but a key difference on the tradeoff of versatility vs. efficiency.
  - Applications: consumer behavior analysis, etc.



Why The Three Tasks?

- Data mining can be enabled by
  - Crawling: the crawled dataset can be treated as a local database
  - Sampling: see the following slides for sample-based/facilitated data mining
  - Data analytics: provides an API for data mining algorithms to call

Sample-Based / Facilitated Data Mining

- Two general methods:
  - Black-box approach: First generate a sample, and then apply data mining over the sample rather than the entire dataset.
    - Transparency can also be achieved at the OLAP level [LHY+08]
  - White-box approach: use sample in selected steps (even preprocessing) of the data mining algorithm.
- Surveys
  - Baohua Gu, Feifang Hu and Huan Liu, Sampling and Its Application in Data Mining: A Survey, Technical Report TRA 6/00, National University of Singapore, 2000.
  - Sameep Mehta and Vinayaka Pandit, Survey of Sampling Techniques for Data Mining, Tutorial, COMAD 2010.

Generic Methods

- Input Reduction (Black-box)
  - Sample from the input dataset the most important tuples for data mining
- Divide-and-Conquer (White-box)
  - Mine one sample set at a time
  - Combine results to produce the final mining results
- Bootstrapping (White-box)
  - Use sample to "guide" data mining over the entire dataset (e.g., as initialization settings)

Sampling for Classification

- Divide-and-Conquer: Windowing in ID3 [Qui86]
  - first use a subset of the training set (i.e., a "window") to construct the decision tree
  - then test it using the remainder of the training set, append mis-classified tuples to the window, and repeat the process until no mis-classification
- Input Reduction: with stratified sampling [Cat91]
  - esp. when the distribution of class labels is far from uniform in the training dataset

Sampling for Association Rule Mining

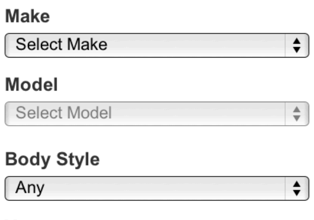
- Bootstrapping: find candidates from samples
  - first use samples to find approximate frequencies / candidate itemsets
  - then use the entire dataset to get the exact frequencies / verify candidates
  - possible to guarantee the discovery of all frequent itemsets (i.e., Las Vegas algorithm)
  - [AMS+96] [Toi96] [ZPLO97] [LCK98] [CHH+05] [CGG10]

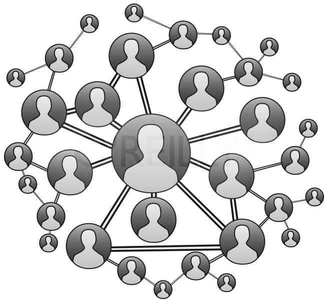
Sampling for Clustering

- Bootstrapping: use sample for initial settings
  - HAC on sample to bootstrap EM [MH98]
- Input Reduction
  - use sampling to neglect small clusters
  - density based sampling (oversample in sparse areas, undersample in dense ones) [PF00]

Dimension 2. Interface

- Keyword-based search
  - Users specify one or a few keywords
  - Common for both structured and unstructured data
  - e.g., Google, Bing, Amazon.
- Form-like search
  - Users specify desired values for one or a few attributes
  - Common for structured data
  - e.g., Yahoo! Autos, AA.com, NSF Award Search.
  - A similar interface: hierarchical browsing
- Graph Browsing
  - A user can observe certain edges and follow through them to access other users' profiles.
  - Common for online social networks
  - e.g., Twitter, Facebook, etc.
- A Combination of Multiple Interfaces
  - e.g., Amazon (all three), eBay (all three).

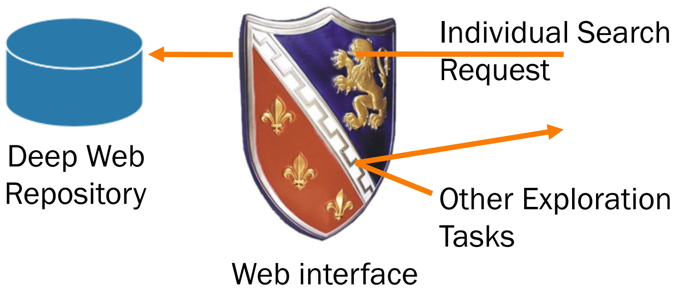




Key Challenge

Restrictive Input Interface

- Restrictions on what queries can be issued
  - Keyword Search Interface: nothing but a set of keywords
  - Form-like Interface: only conjunctive search queries
    - e.g., List all Honda Accord cars with Price below $10,000
  - Graph Browsing Interface
    - only select one of the neighboring nodes
- We do not have complete access to the repository. No complete SQL support
  - e.g., we cannot issue “big picture” queries: e.g., SUM, MIN, MAX aggregate queries
  - e.g., we cannot issue “meta-data” queries: e.g., keyword such as DISTINCT (handy for domain discovery)



Key Challenge

Restrictive Output Interface

- Restrictions on how many tuples will be returned
  - Top-k restriction leads to three types of queries:
    - overflowing (> k): top-k elements (documents, tuples) will be selected according to a (sometimes secret) scoring function and returned
    - valid (1..k element)
    - underflowing (0 element)
  - COUNT vs. ALERT
    - An alert of overflowing can always be obtained through a web interface

> A maximum of 3000 awards are displayed. If you did not find the information you are looking for, please refine your search.

  - Page turn
    - Limited number of page turns allowed (e.g., 10-100 for Google)
      - Essentially the same as top-k restriction

> Your search returned 41427 results. The allowed maximum number of results is 1000. Please narrow down your search criteria and try your search again.

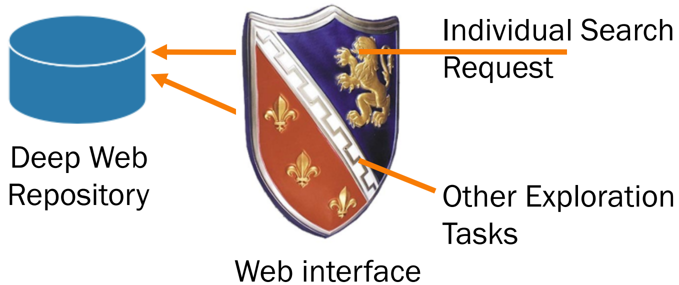
    - Unlimited page turns
      - But a page turn also consumes a web access

> **1-15** of 15167 used Ford F150 cars

Key Challenge

Implications of Interface Restrictions

- Two ways to address the input/output restrictions
  - Direct negotiation with the owner of the deep web repository
    - Crawling, sampling and analytics can all be supported (if necessary)
    - Used by many real-world systems - e.g., Kayak
  - Bypass the interface restrictions
    - By issuing a carefully designed sequence of queries
    - e.g., for crawling: these queries should recall as many tuples as possible
      - or even “prove” that all tuples/documents returnable by the output interface are crawled.
    - e.g., for analytics: one should be able to infer from these queries an accurate estimation of an aggregate that cannot be directly issued because of the input interface restriction.

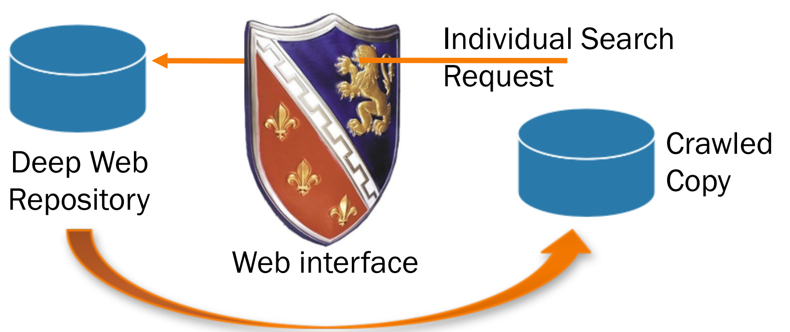


Outline

- Introduction
- Resource Discovery and Interface Understanding
- Technical Challenges for Data Exploration
- Crawling
- Sampling
- Data Analytics
- Final Remarks

Overview of Crawling

- Motivation for crawling
  - Enable third-party web services - e.g., mash-up
  - A pre-processing step for answering queries not supported by the web interface
    - e.g., count the percentage of used cars which have GPS navigation; find all documents which contain the term “DBMS” and were last updated after Aug 1, 2011.
    - Note: these queries cannot be directly answered because of the interface restrictions.
  - Note the key differences with web crawling
- Taxonomy of crawling techniques
  - Interfaces: (a) (keyword and form-like) search interface, (b) browsing interface
  - Technical challenges: (1) find a finite set of queries that recall most if not all tuples (a challenge only for search interfaces), (2) find a small subset while maintaining a high recall, (3) issue the small subset in an efficient manner (i.e., system issues).
- Our discussion order
  - (a1), (a2), (b2), (*3)



Crawling Over Search Interfaces

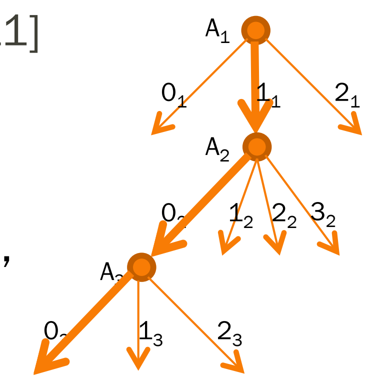
(a1) Find A Finite Set of Search Queries with High Recall

- Keyword search interface
  - Use a pre-determined query pool: e.g., all English words/phrases
  - Bootstrapping technique: iterative probing [CMH08]
- Form-like search interface
  - If all attributes are represented by drop-down boxes or check buttons
    - Solution is trivial 

  - If certain attributes are represented by text boxes
    - Prerequisite: attribute domain discovery 

    - Nearly impossible to guarantee complete discovery [JZD11]
      - Reason: top-k restriction on output interface
      - k: $\Omega(|V|^m)$; query cost: $\Omega(m^2|V|^3)$
      - Probabilistic guarantee achievable
    - Note: domain discovery also has other applications – e.g., as a preprocessor for sampling, or standalone interest.

Query: SELECT * FROM D
Answer: $\{0_1, 0_2, \ldots, 0_m\}$

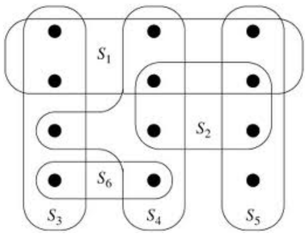


[CMH08] M. J. Cafarella, J. Madhavan, and A. Halevy, "Web-Scale Extraction of Structured Data", SIGMOD Record, vol. 37, 2008.
[JZD11] X. Jin, N. Zhang, G. Das, "Attribute Domain Discovery for Hidden Web Databases", SIGMOD 2011.

Crawling Over Search Interfaces

(a2) How to Efficiently Crawl - Heuristics

- Motivation: Cartesian product of attribute domains often orders of magnitude larger than the repository size
  - e.g., cars.com: 5 inputs, 200 million combinations vs. 650,000 tuples
- How to use the minimum number of queries to achieve a significant coverage of underlying documents/tuples
  - Essentially a set cover problem (but inputs are not properly known before hand)
- Search query selection
  - Keyword search: a heuristic of maximizing #new_elements/cost [NZC05]
    - #new_elements: not crawled by previously issued queries
    - Cost may include keyword query cost + cost for downloading details of an element
  - Form-like search: find “binding” inputs [MKK+08]
    - Informative query template: grow with increasing dimensionality
    - Good news: #informative templates grows proportionally with the database size, not #input combinations.



Make:Toyota
Type:Hybrid

~~Make:Jeep~~
~~Type:Hybrid~~

[NZC05] A. Ntoulas, P. Zerfos, and J. Cho, "Downloading Textual Hidden Web Content through Keyword Queries", JCDL, 2005.
[MKK+08] J. Madhavan, D. Ko, L. Kot, V. Ganapathy, A. Rasmussen, and A. Halevy, “Google’s Deep-Web Crawl”, VLDB 2008.

Crawling Over Search Interfaces

(a2) How to Efficiently Crawl - Theoretical Bounds

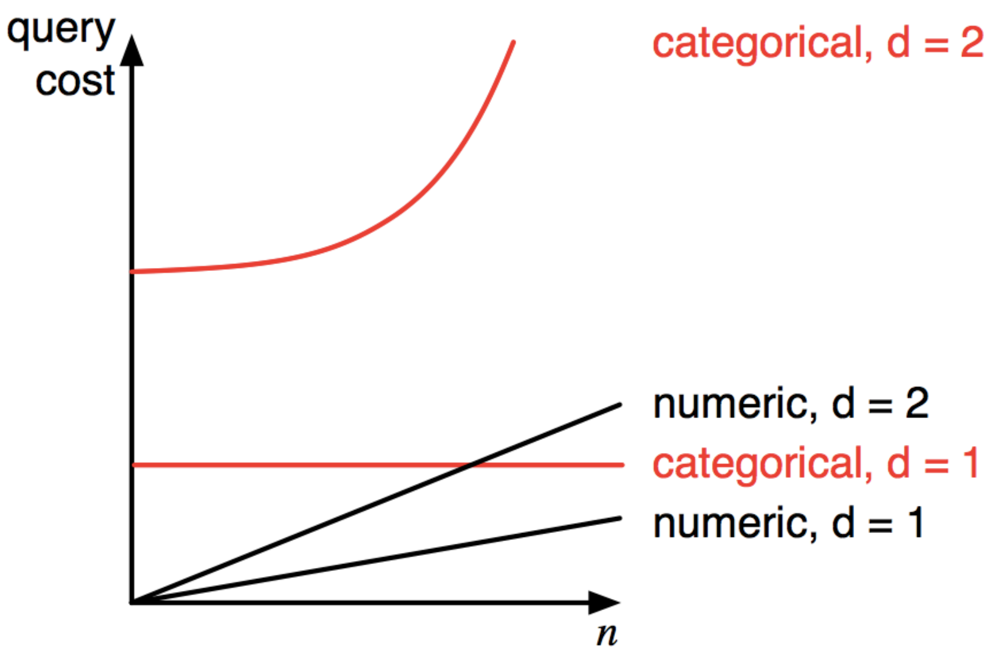
- Crawling Algorithms for Form-Like Search [SZT+12]
  - $O(mn/k)$ for a numeric database.
  - U_1 when there is only one categorical attribute.
  - $n/k * \Sigma^m_{i=1} \min(U_i, n/k) + \Sigma^m_{i=1} U_i$ for a categorical database.
  - $U_1 + O(mn/k)$ for a mixed database with one categorical attribute.
  - $n/k * \Sigma^m_{i=1} \min(U_i, n/k) + \Sigma^m_{i=1} U_i + O((m\text{-}cat)*n/k)$ for a mixed database with cat (cat > 1) categorical attributes.
- None of these can be improved beyond a constant factor!

m: # attributes
n: # tuples
U_i: attr domain size

[SZT+12] Cheng Sheng, Nan Zhang, Yufei Tao, and Xin Jin, "Optimal Algorithms for Crawling a Hidden Database in the Web", VLDB 2012.

Crawling Over Search Interfaces

(a2) How to Efficiently Crawl - Theoretical Bounds



Summary:

We pay $O(n/k)$ for each numeric attribute

We pay $n/k * \min(U_i, n/k) + U_i$ for each categorical attribute.

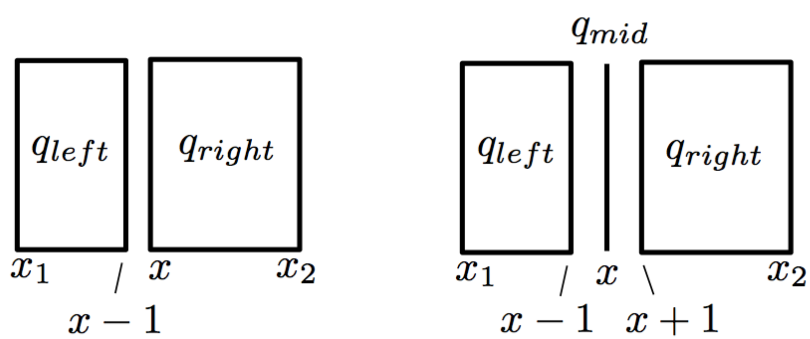
Only exception: when there is only one categorical attribute.

Crawling Over Search Interfaces

(a2) How to Efficiently Crawl – Numeric DB

- Baseline Algorithm: Recursive equal-width binary split.
  - Problem: Worst-case query cost is domain-size dependent.
  - Pitfall: Many wasted queries - overflow and return no new tuple beyond previously seen.

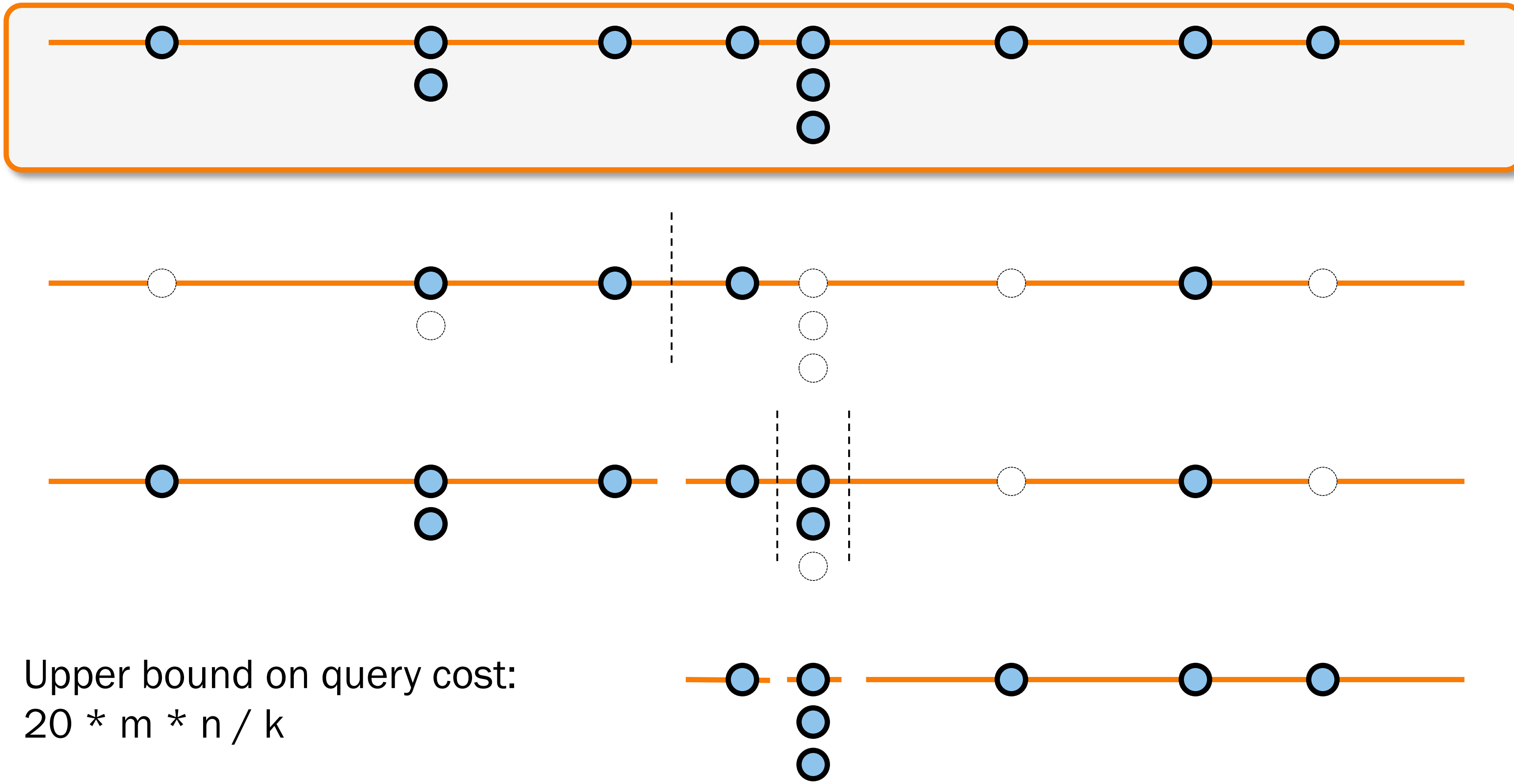
- Key Idea of Rank-Shrink: Make sure no query is wasted.
  - each post-split query either is valid, or returns at least k/4 new tuples.



- How? Consider the ordered set of k tuples returned by q:
  - 1 The ⌈k/2⌉-th value shared by more than k/4 (identical) tuples?
  - 2 YES → ternary split, NO → binary split.
  - Worst-case query cost: 12n/k.

Crawling Over Search Interfaces

(a2) How to Efficiently Crawl - RANK-SHRINK for Numeric DB



Upper bound on query cost:
20 * m * n / k

[SZT+12] Cheng Sheng, Nan Zhang, Yufei Tao, and Xin Jin, “Optimal Algorithms for Crawling a Hidden Database in the Web”, VLDB 2012.

Crawling Over Search Interfaces

(a2) How to Efficiently Crawl – Worst-Case Numeric DB

- Key Observation: No resolved query covers more than 1 tuple.

	A_1	A_2	$\cdots$	A_d	
	1	1	$\cdots$	1	
		$\cdots$			k tuples
	1	1	$\cdots$	1	
Group 1	2	1	$\cdots$	1	
	1	2	$\cdots$	1	d tuples
		$\cdots$			
	1	1	$\cdots$	2	
	2	2	$\cdots$	2	
		$\cdots$			k tuples
	2	2	$\cdots$	2	
Group 2	3	2	$\cdots$	2	
	2	3	$\cdots$	2	d tuples
		$\cdots$			
	2	2	$\cdots$	3	
$\vdots$		$\cdots$			
	m	m	$\cdots$	m	
		$\cdots$			k tuples
	m	m	$\cdots$	m	
Group m	$m+1$	m	$\cdots$	m	
	m	$m+1$	$\cdots$	m	d tuples
		$\cdots$			
	m	m	$\cdots$	$m+1$	

- Implication: $\exists\, \varepsilon = k/d > 0$, s.t. query cost $\geq dn/k \cdot \varepsilon$.

Crawling Over Search Interfaces

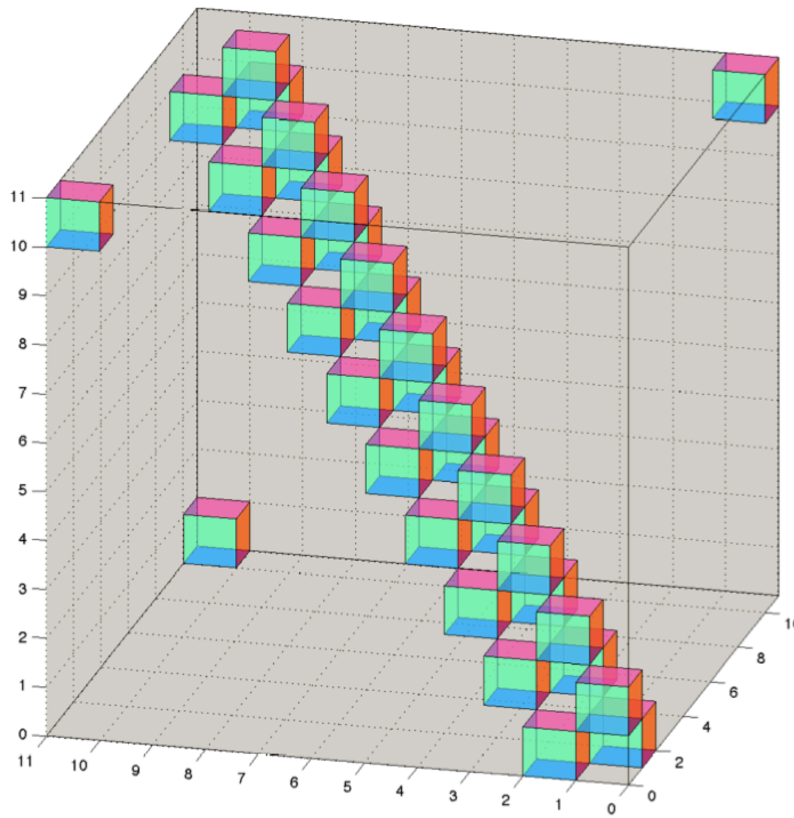
(a2) How to Efficiently Crawl – Categorical DB

- Baseline strategy: Depth-first search
  - Problem: Query cost (i.e., tree size) depends on domain size which can be unbounded.
- Key Observation: "Essential" domain size is indeed bounded.
  - 1 "Cover" Ai - i.e., query $A_i = v_1$, $A_i = v_2$,..., $A_i = v_{Ui}$ for all domain values for Ai respectively.
  - 2 Ignore all v_j for which $A_i = v_j$ returns valid.
  - 3 At most $\min\{U_i, n/k\}$ values left in the essential domain.
- Key Idea of Slide-Cover: Cover-Then-Slice
  - 1 "Cover" to find the essential domains of all attributes.
  - 2 "Slice" by performing DFS over a tree constructed from the essential domains.
- Worst-case query cost
  - U_1 when there is only one categorical attribute.
  - $n/k * \Sigma^m_{i=1}\min(U_i, n/k) + \Sigma^m_{i=1} U_i$ otherwise.

Crawling Over Search Interfaces

(a2) How to Efficiently Crawl – Worst-Case Categorical DB

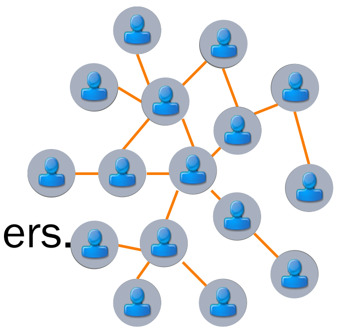
	A_1	A_2	$\cdots$	A_d
Group 0	1	0	$\cdots$	0
	0	1	$\cdots$	0
	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	0	0	$\cdots$	1
Group 1	2	1	$\cdots$	1
	1	2	$\cdots$	1
	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	1	1	$\cdots$	2
$\vdots$		$\cdots$		
Group $U-1$	0	$U-1$	$\cdots$	$U-1$
	$U-1$	0	$\cdots$	$U-1$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	$U-1$	$U-1$	$\cdots$	0



Crawling Over Browsing Interfaces

(b2) How to Efficiently Crawl

- Technical problem
  - Hierarchical browsing: Traverse vertices of a tree
  - Graph browsing: Traverse vertices of a graph
    - Starting with a seed set of users (resp. URLs).
    - Recursively follows relationships (resp. hyperlinks) to others.
  - Exhaustive crawling vs. Focused crawling



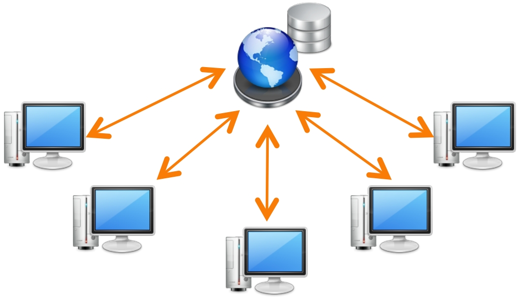
- Findings
  - Are real-world social networks indeed connected?
    - It depends – Flickr ~27%, LiveJournal ~95% [MMG+07]
  - How to select "seed(s)" for crawling?
    - Selection does not matter much as long as the number of seeds is sufficiently large (e.g., > 100) [YLW10]

[MMG+07] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and Analysis of Online Social Networks", IMC, 2007.
[YLW10] S. Ye, J. Lang, F. Wu, "Crawling Online Social Graphs", APWeb, 2010.

System Issues Related to Crawling

(*3) how to issue queries efficiently

- Using a cluster of machines for parallel crawling
  - Imperative for large-scale crawling
  - Extensively studied for web crawling
    - But are the challenges still the same for crawling deep web repositories?
- Independent vs. Coordination
  - Overlap vs. (internal) communication overhead
  - How much coordination? Static vs. dynamic

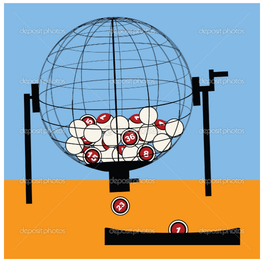


- Politeness, or server restriction detection
  - e.g., some repositories block an IP address if queries are issued too frequently – but how to identify the maximum unblocked speed?

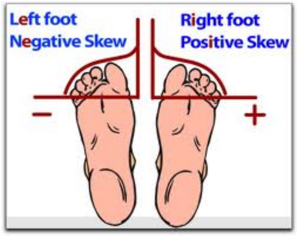
Outline

- Introduction
- Resource Discovery and Interface Understanding
- Technical Challenges for Data Exploration
- Crawling
- Sampling
- Data Analytics
- Final Remarks

Overview of Sampling



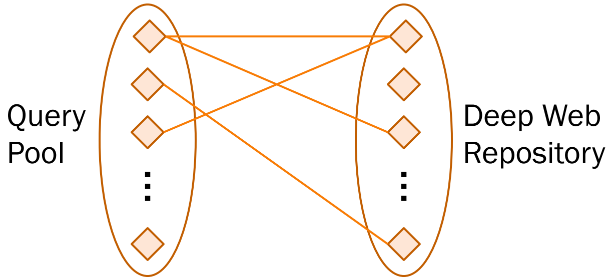
- Objective: Draw representative elements from a repository
  - Quality measure: sample skew
  - Efficiency measure: number of web accesses required
- Moti
  - U
    g
    [
  - S
    p
  - N data
    n
- Cent
  - Skew reduction: make the sampling distribution as close to a target distribution as possible
    - Target distribution is often the uniform distribution – in this case, the objective is to make the probability of retrieving each document as uniform as possible.



[IG02] P. G. Iperirotis and L. Gravano, "Distributed Search over the Hidden Web: Hierarchical Database Sampling and Selection", VLDB, 2002.
[SZS+06] M. Shokouhi, J. Zobel, F. Scholer, and S. Tahaghoghi, "Capturing collection size for distributed non-cooperative retrieval", SIGIR, 2006.
[BB98] K. Bharat and A. Broder, "A technique for measuring the relative size and overlap of public Web search engines", WWW, 1998.
[BG08] Z. Bar-Yossef and M. Gurevich, "Random sampling from a search engine's index", JACM, vol. 55, 2008.
[Das03] G. Das, "Survey of Approximate Query Processing Techniques (Tutorial)", SSDBM, 2003.
[GG01] M. N. Garofalakis and P. B. Gibbons, "Approximate Query Processing: Taming the TeraBytes", VLDB, 2001.

Sampling Over Keyword-Search Interfaces
Pool-Based Sampler: Basic Idea

- Query-pool based sampler
  - Assumption: there is a given (large) pool of queries which, once being issued through the web interface, can recall the vast majority of elements in the deep web repository
  - e.g., for unstructured data, a pool of English phrases
- Two types of sampling process
  - Heuristic: based on an observation that the query pool is too large to enumerate – so we have to (somehow) choose a small subset of queries (randomly or in a heuristic fashion) [IG02, SZS+06, BB98]
    - Problem: no guarantee on the “quality” (i.e., skew) of retrieved sample elements – e.g., if one randomly chooses a query and then randomly selects a document from the returned result [BB98], then longer documents will be favored over shorter ones.
  - Skew reduction: identify the source of skew and use skew-correction techniques, e.g., rejection sampling, to remove the skew.
- Interesting observation: relationship b/w keyword and sampling a bipartite graph



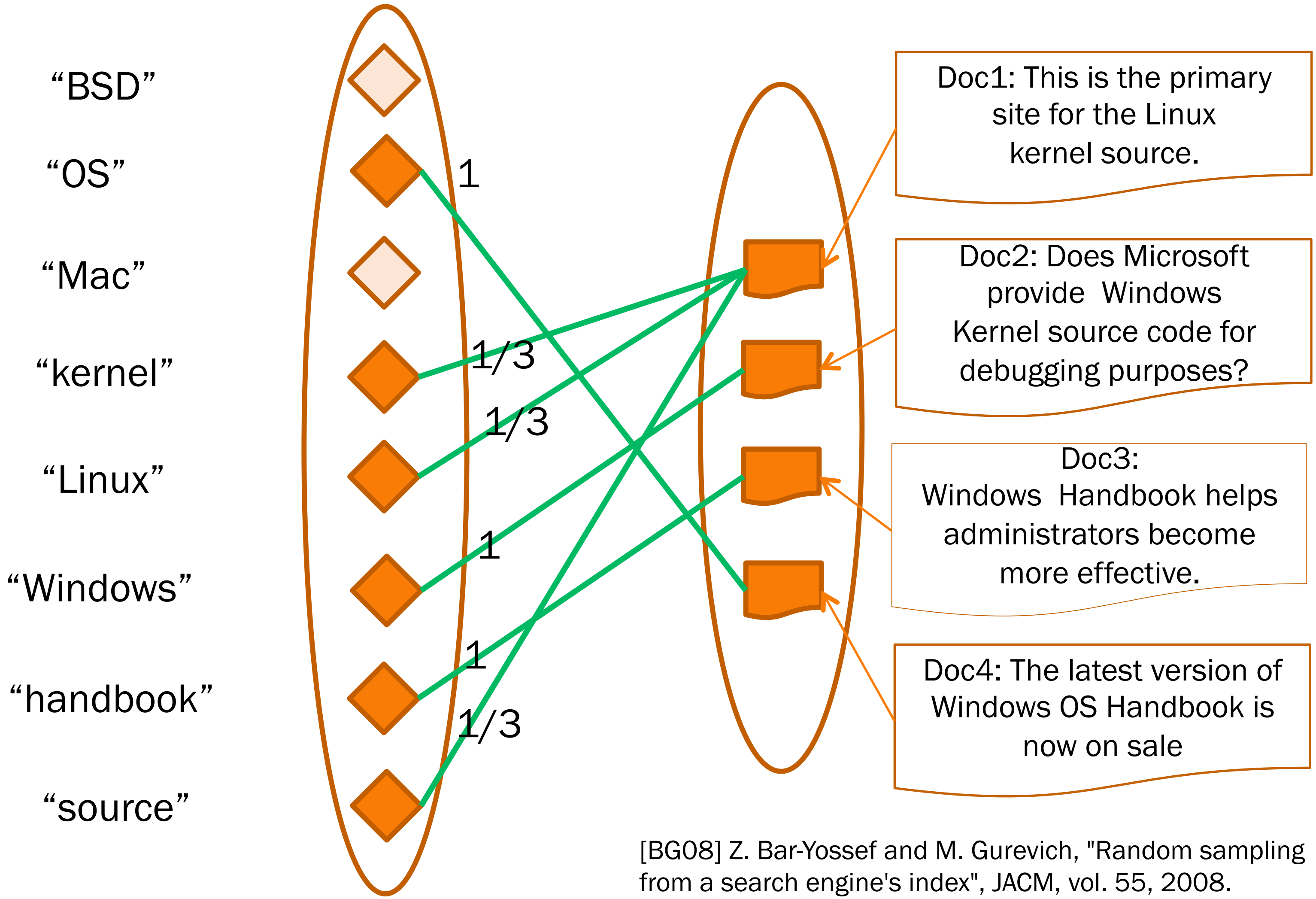
[IG02] P. G. Iperirotis and L. Gravano, "Distributed Search over the Hidden Web: Hierarchical Database Sampling and Selection", VLDB, 2002.

[SZS+06] M. Shokouhi, J. Zobel, F. Scholer, and S. Tahaghoghi, "Capturing collection size for distributed non-cooperative retrieval", SIGIR, 2006.

[BB98] K. Bharat and A. Broder, "A technique for measuring the relative size and overlap of public Web search engines", WWW, 1998.

Sampling Over Keyword-Search Interfaces

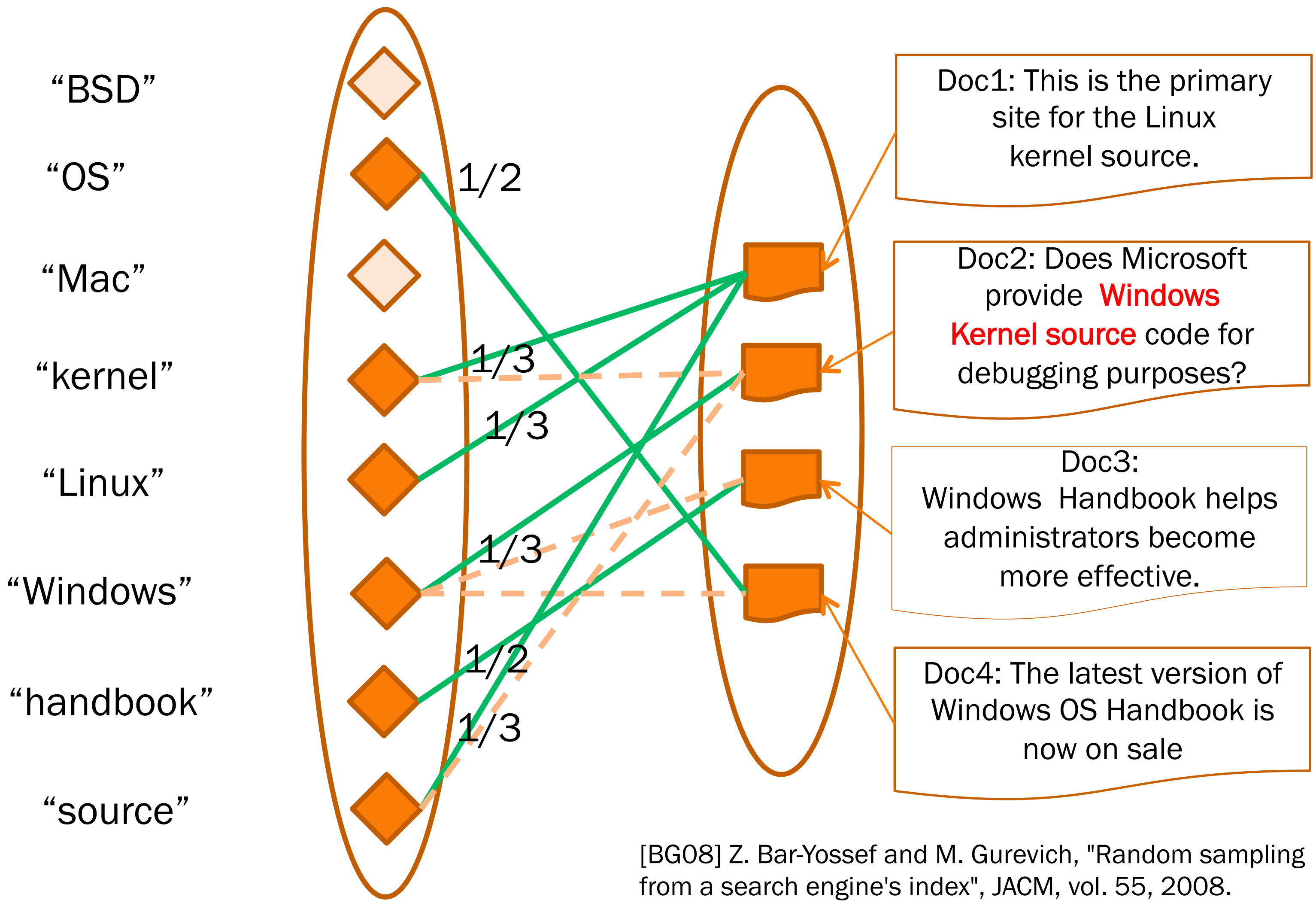
Pool-Based Sampler: Reduce Skew



[BG08] Z. Bar-Yossef and M. Gurevich, "Random sampling from a search engine's index", JACM, vol. 55, 2008.

Sampling Over Keyword-Search Interfaces

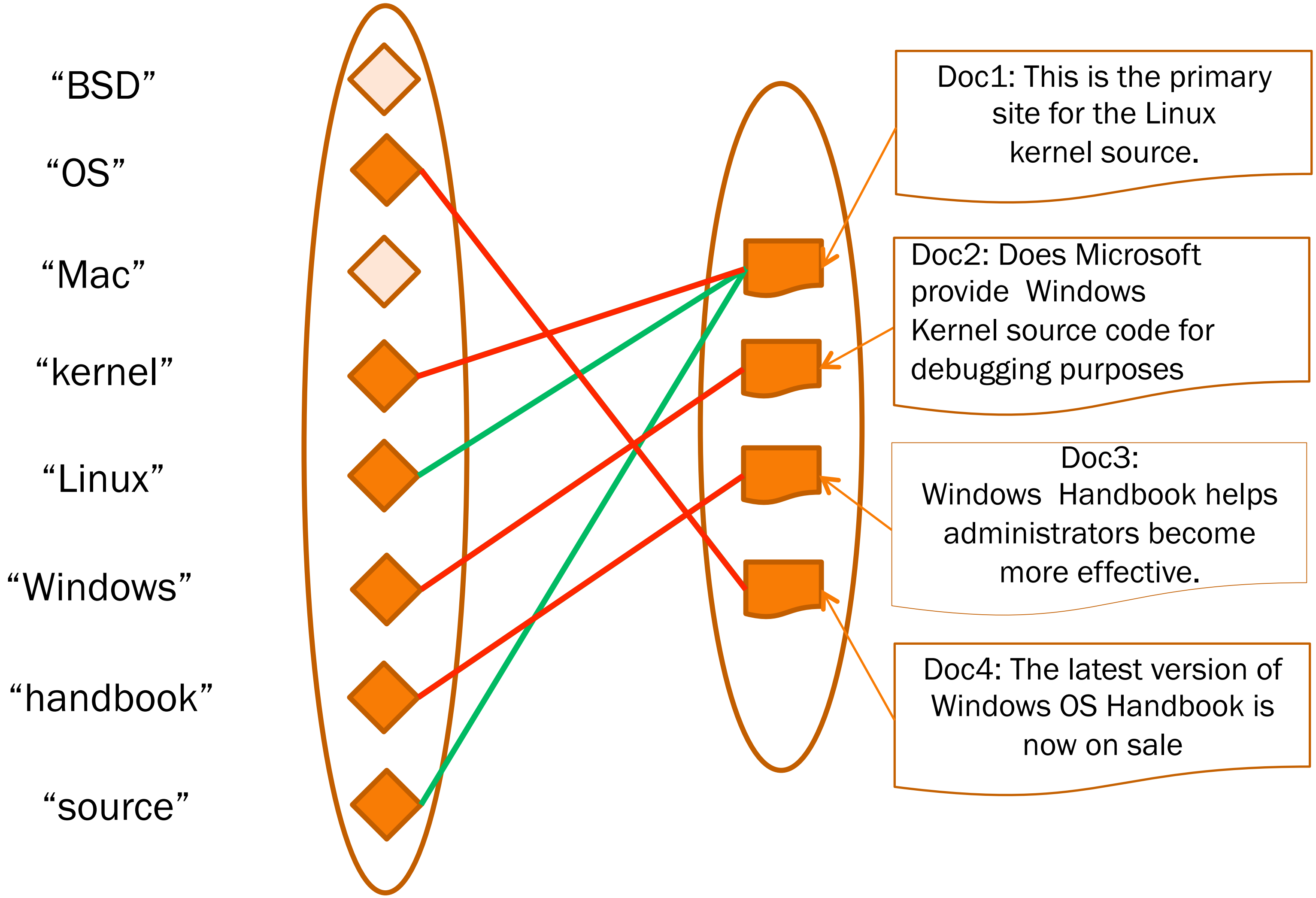
Pool-Based Sampler: Reduce Skew



[BG08] Z. Bar-Yossef and M. Gurevich, "Random sampling from a search engine's index", JACM, vol. 55, 2008.

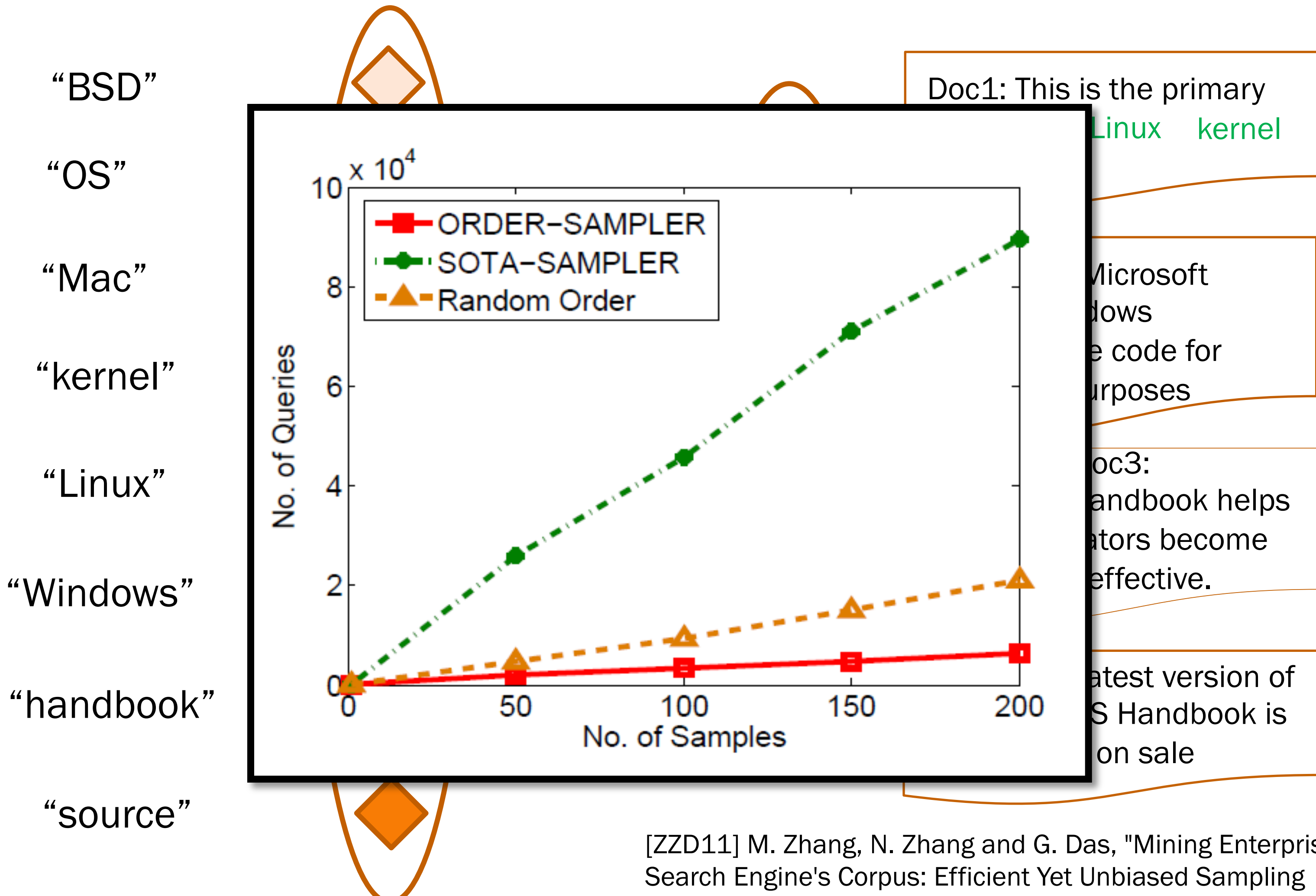
Sampling Over Keyword-Search Interfaces

Pool-Based Sampler: Remove Skew



Sampling Over Keyword-Search Interfaces

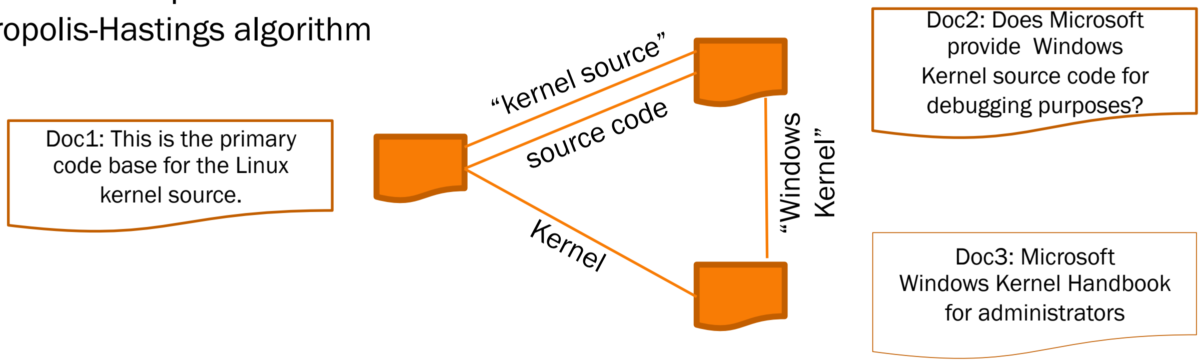
Pool-Based Sampler: Remove Skew



[ZZD11] M. Zhang, N. Zhang and G. Das, "Mining Enterprise Search Engine's Corpus: Efficient Yet Unbiased Sampling and Aggregate Estimation", SIGMOD 2011.

Sampling Over Keyword-Search Interfaces
Other Sampling Methods

- Pool-free random walk [BG08]
  - A graph model
    - Each element in the repository is a vertex
    - Two elements are connected if they are returned by the same query
  - Random walk over the graph, two enabling factors:
    - Given an element, we can sample uniformly at random a query which returns the document. (YEA for almost all keyword search interfaces).
    - Given an element, we can find the number of queries which return the document (may incur significant query cost)
  - Challenge 1: is the graph connected?
    - Note: the set of all possible queries which might return a document can be extremely large
      - 2^n queries for a document with n words
    - Thus, we have to limit our attention to a subset of queries
      - e.g., only consecutive phrases
      - Problem: too restricted – disconnected graph, too relaxed – high cost for sampling
  - Challenge 2: how to perform random walk?
    - Metropolis-Hastings algorithm



[BG08] Z. Bar-Yossef and M. Gurevich, "Random sampling from a search engine's index", JACM, vol. 55, 2008.

Sampling Over Form-Like Interfaces
Source of Skew

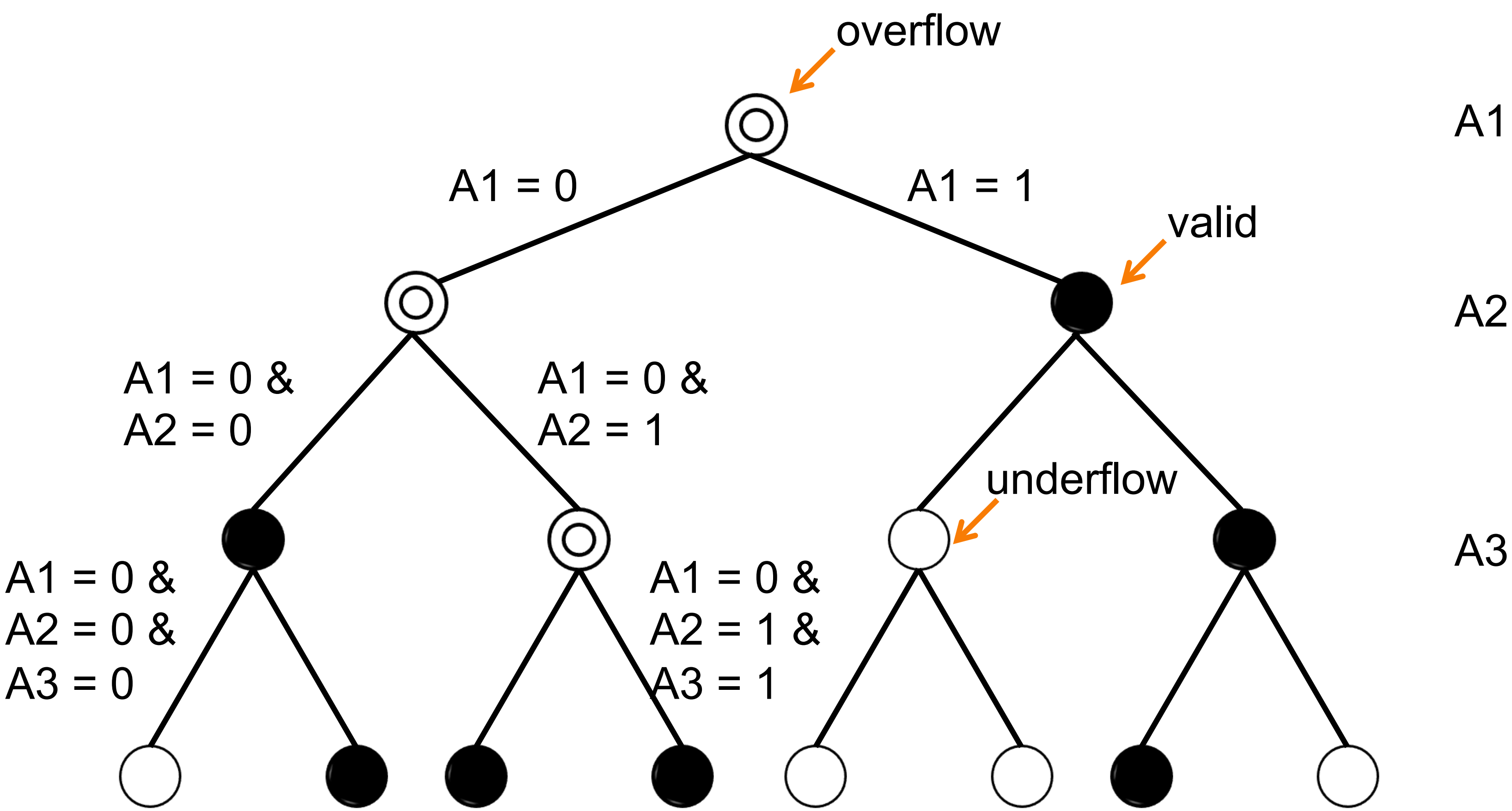
- Recall: Restrictions for Form-Like Interfaces
  - Input: conjunctive search queries only
  - Output: return top-k tuples only (with or without the COUNT of matching tuples)
- Good News
  - Defining “designated queries” no longer a challenge
  - e.g., consider all fully specified queries – each tuple is returned by one and only one of them

Sampling Over Form-Like Interfaces
Source of Skew

- Bad News: A New Source of Skew
  - We cannot really use fully specified queries because sampling would be really like search for a needle in a haystack
  - So we must use shorter, broader queries
    - But such queries may be affected by the top-k output restriction
    - Skew may be introduced by the scoring function used to select top-k tuples
    - e.g., skew on average price when the top-k elements are the ones with the lowest prices
- Basic idea for reducing/removing skew
  - Find non-empty queries which are not affected by the scoring function – i.e., queries which return 1 to k elements

Sampling Over Form-Like Interfaces

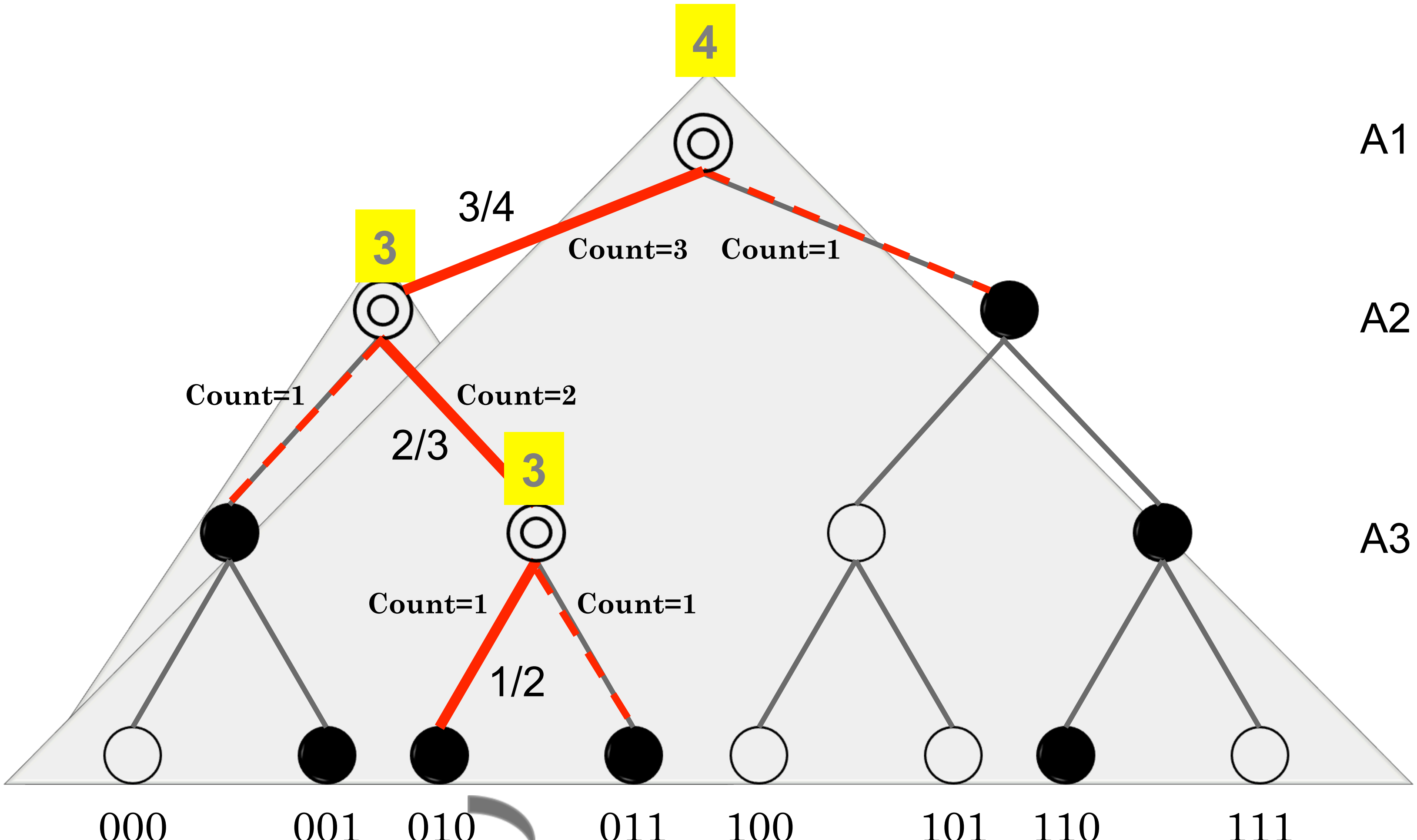
COUNT-Based Skew Removal



[DZD09] A. Dasgupta, N. Zhang, and G. Das, Leveraging COUNT Information in Sampling Hidden Databases, ICDE 2009.

Sampling Over Form-Like Interfaces

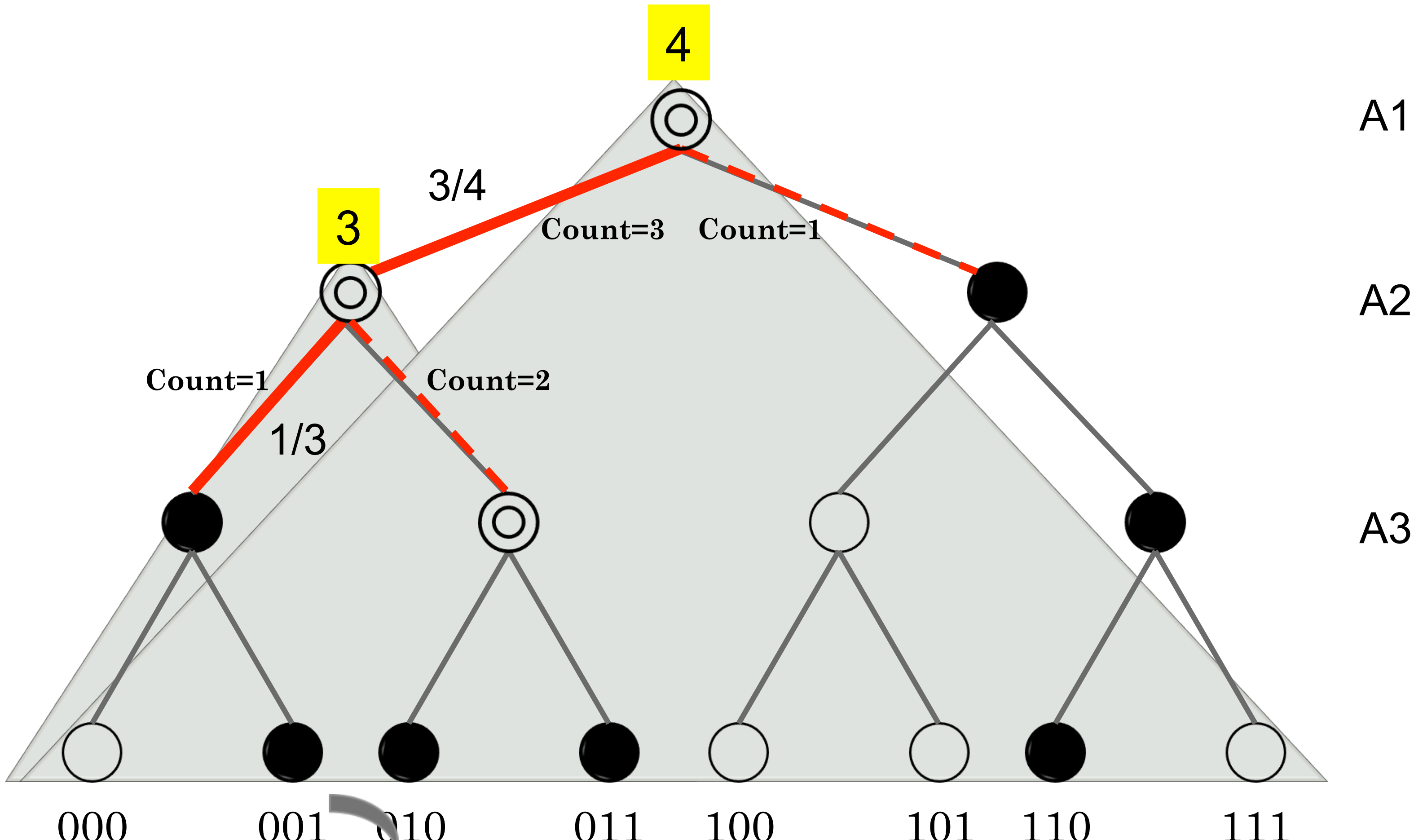
COUNT-Based Skew Removal



3/4 * 2/3 * 1/2 = 1/4

[DZD09] A. Dasgupta, N. Zhang, and G. Das, Leveraging COUNT Information in Sampling Hidden Databases, ICDE 2009.

Sampling Over Form-Like Interfaces
COUNT-Based Skew Removal

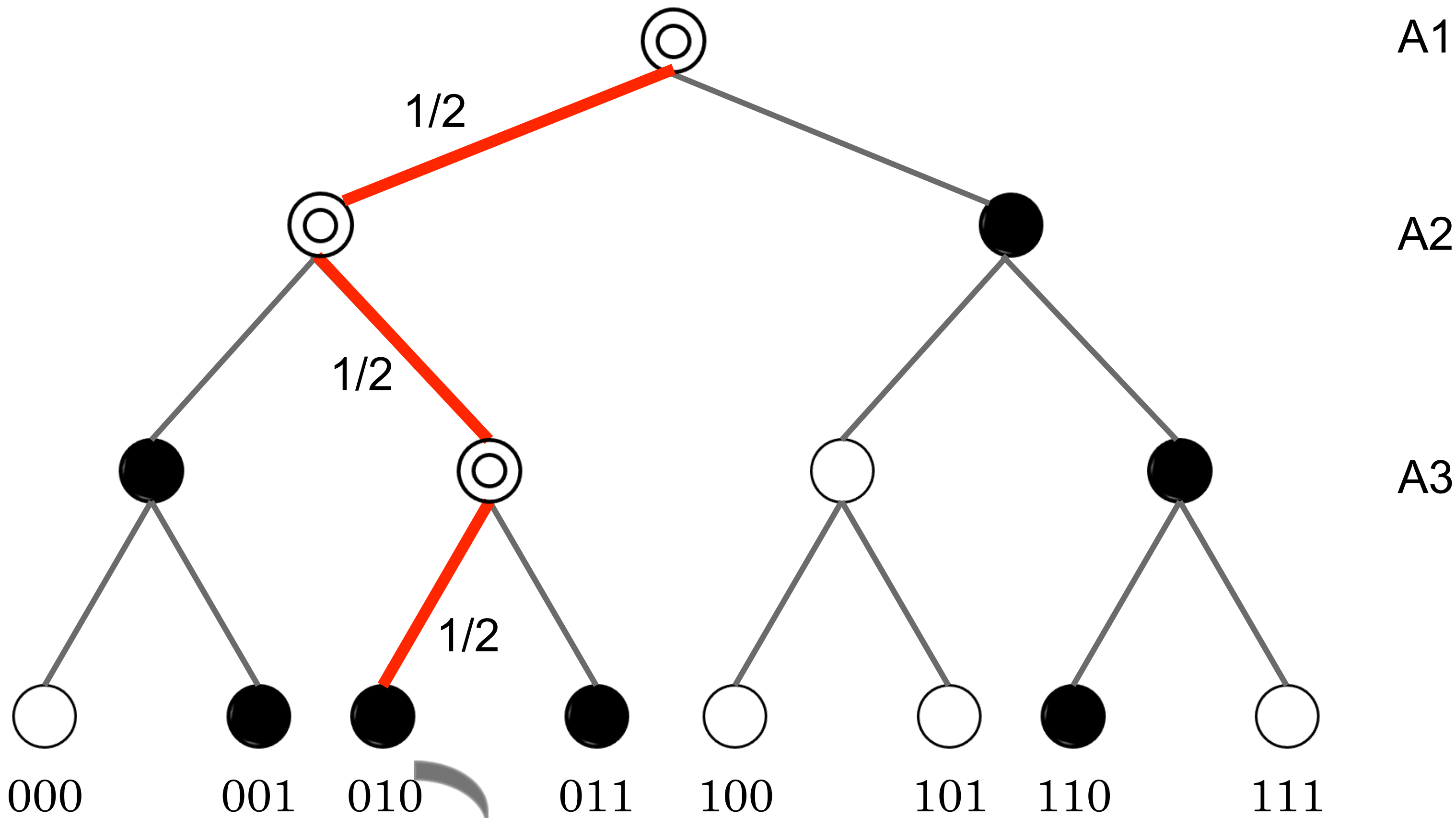


3/4 * 1/3 = 1/4

[DZD09] A. Dasgupta, N. Zhang, and G. Das, Leveraging COUNT Information in Sampling Hidden Databases, ICDE 2009.

Sampling Over Form-Like Interfaces

Skew Reduction for Interfaces Sans COUNT

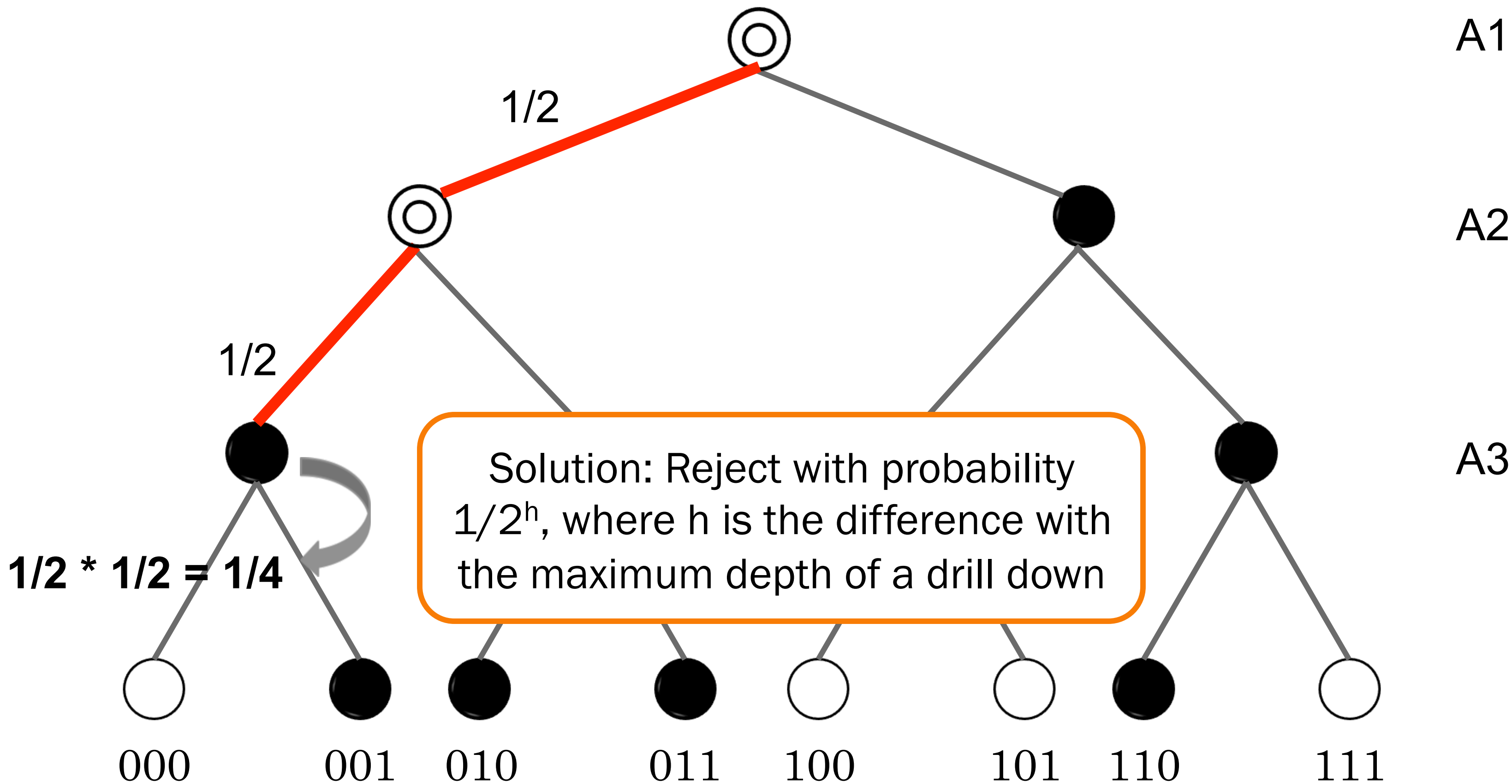


1/2 * 1/2 * 1/2 = 1/8

[DDM07] A. Dasgupta, G. Das, and H. Mannila, A Random Walk Approach to Sampling Hidden Databases, SIGMOD 2007.

Sampling Over Form-Like Interfaces

Skew Reduction for Interfaces Sans COUNT



Solution: Reject with probability $1/2^h$, where h is the difference with the maximum depth of a drill down

[DDM07] A. Dasgupta, G. Das, and H. Mannila, A Random Walk Approach to Sampling Hidden Databases, SIGMOD 2007.

Sampling Over Graph Browsing Interfaces
Sampling by exploration

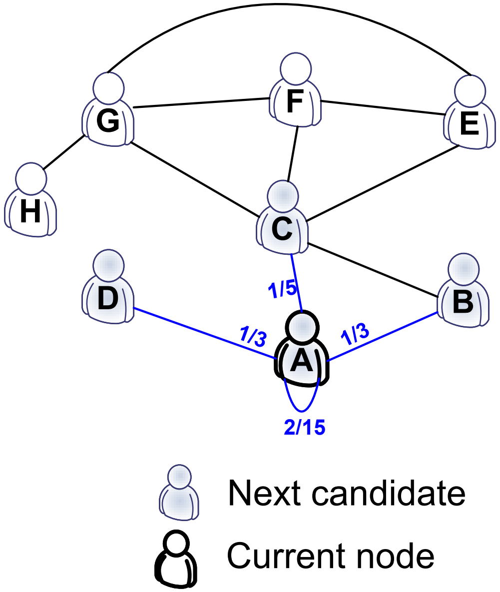
- Note: Sampling is a challenge even when the entire graph topology is given
  - Reason: Even the problem definition is tricky
    - What to sample? Vertices? Edges? Sub-graphs?
- Methods for sampling vertices, edges, or sub-graphs
  - Snowball sampling: a nonprobability sampling technique
  - Random walk with random restart
  - Forest Fire
  - ...

- What are the possible goals of sampling? [LF06]
  - Criteria for a static snapshot
    - In-degree & out-degree distributions, distributions of weakly/strongly connected components (for directed graphs), distribution of singular values, clustering coefficient, etc.
  - Criteria for temporal graph evolution
    - #edges vs. #nodes over time, effective diameter of the graph over time, largest connected component size over time,

[LF06] J Leskovec and C Faloutsos, Sampling from Large Graph, KDD 2006.

Sampling Over Graph Browsing Interfaces
Unbiased Sampling

- Survey and Tutorials for random walks on graphs
  - [Lov93], [LF08], [Mag08]
- Simple random walk is inherently biased
  - Stationary distribution: each node v has probability of d(v)/(2|E|) of being selected, where d(v) is the degree of v and |E| is the total number of edges – i.e., p(v) ~ d(v)
- Skew correction
  - Re-weighted random walk [VH08]
    - Rejection sampling
    - Or, if the objective is to use the samples to estimate an aggregate, then apply Hansen-Hurwitz estimator after a simple random walk.
  - Metropolis-Hastings random walk [MRR+53]
    - Transition probability from u to its neighbor v: min(1, d(u)/d(v))/d(u)
    - Stay at u with the remaining probability
    - Leading to a uniform stationary distribution



Example taken from the slides of M Gjoka, M Kurant, C Butts, A Markopoulou, "Walking in Facebook: Case Study of Unbiased Sampling of OSNs", INFOCOM 2010

[Mag08] M. Maggioni, Tutorial - Random Walks on Graphs Large-time Behavior and Applications to Analysis of Large Data Sets, MRA 2008.
[LF08] J. Leskovec and C. Faloutsos, "Tools for large graph mining: structure and diffusion", WWW (Tutorial), 2008.
[Lov93] L. Lovasz, "Random walks on graphs: a survey", Combinatorics, Paul Erdos is Eighty, 1993.
[VH08] E. Volz and D. Heckathorn, "Probability based estimation theory for respondent-driven sampling," J. Official Stat., 2008.
[MRR+53] N. Metropolis, M. Rosenblut, A. Rosenbluth, A. Teller, and E. Teller, Equation of state calculation by fast computing machines, J. Chem. Phys., vol. 21, 1953.

Outline

- Introduction
- Resource Discovery and Interface Understanding
- Technical Challenges for Data Exploration
- Crawling
- Sampling
- Data Analytics
- Final Remarks

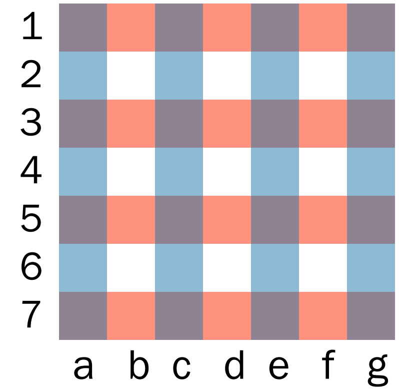
Overview of Data Analytics

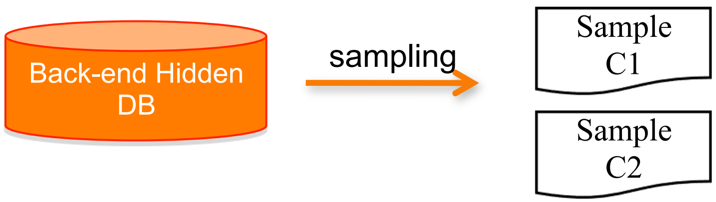
- Objective: Directly estimate aggregates over a deep web repository
- Motivating Applications
  - Unstructured data: Google vs. Bing, whose repository is more comprehensive?
  - Structured data: Total price of all cars listed at Yahoo! Autos?
- Sampling vs. Data Analytics
  - Data analytics requires the target aggregate to be known a priori. Samples can support multiple data analytics tasks
  - while samples may also be used to estimate (some, not all) aggregates, direct estimation is often more efficient because the estimation process can be tailored to the aggregate being estimated.
- Performance Measures
  - Quality measure: MSE = Bias^2 + Var:
    - Reduction of both bias and variance.
  - Efficiency measure: number of web accesses required

Analytics Over Keyword Search Interfaces

Leveraging Samples: Mark-and-Recapture

- Used for estimating population size in ecology.
- Recently used (in various forms) for estimating the corpus size of a search engine
  - Absolute size: [BFJ+06] [ZSZ+06] [LYM02]
  - Relative size (among search engines): [BB98] [BG08]





Lincoln-Petersen model

$$\tilde{m} = \frac{|C1| \times |C2|}{|C1 \cap C2|}$$

$$\tilde{m} = \frac{|C1| \times |C2|}{|C1 \cap C2|} = \frac{28 \times 28}{16} = 49$$

- Note: only requires C1 and C2 to be uncorrelated - i.e., the fraction of documents in the corpus that appears in C1 should be the same as the fraction of documents in C2 that appear in C1

[BB98] K. Bharat and A. Broder, "A technique for measuring the relative size and overlap of public Web search engines", WWW, 1998.
[BG08] Z. Bar-Yossef and M. Gurevich, "Random sampling from a search engine's index", JACM, vol. 55, 2008.
[BFJ+06] A. Broder, M. Fontura, V. Josifovski, R. Kumar, R. Motwani, S. Nabar, R. Panigrahy, A. Tomkis, and Y. Xu, "Estimating corpus size via queries", CIKM, 2006.
[SZS+06] M. Shokouhi, J. Zobel, F. Scholer, and S. Tahaghoghi. Capturing collection size for distributed non-cooperative retrieval. In *SIGIR*, 2006.
[LYM02] Y. C. Liu, K. Yu and W. Meng. Discovering the representative of a search engine. In *CIKM*, 2002.

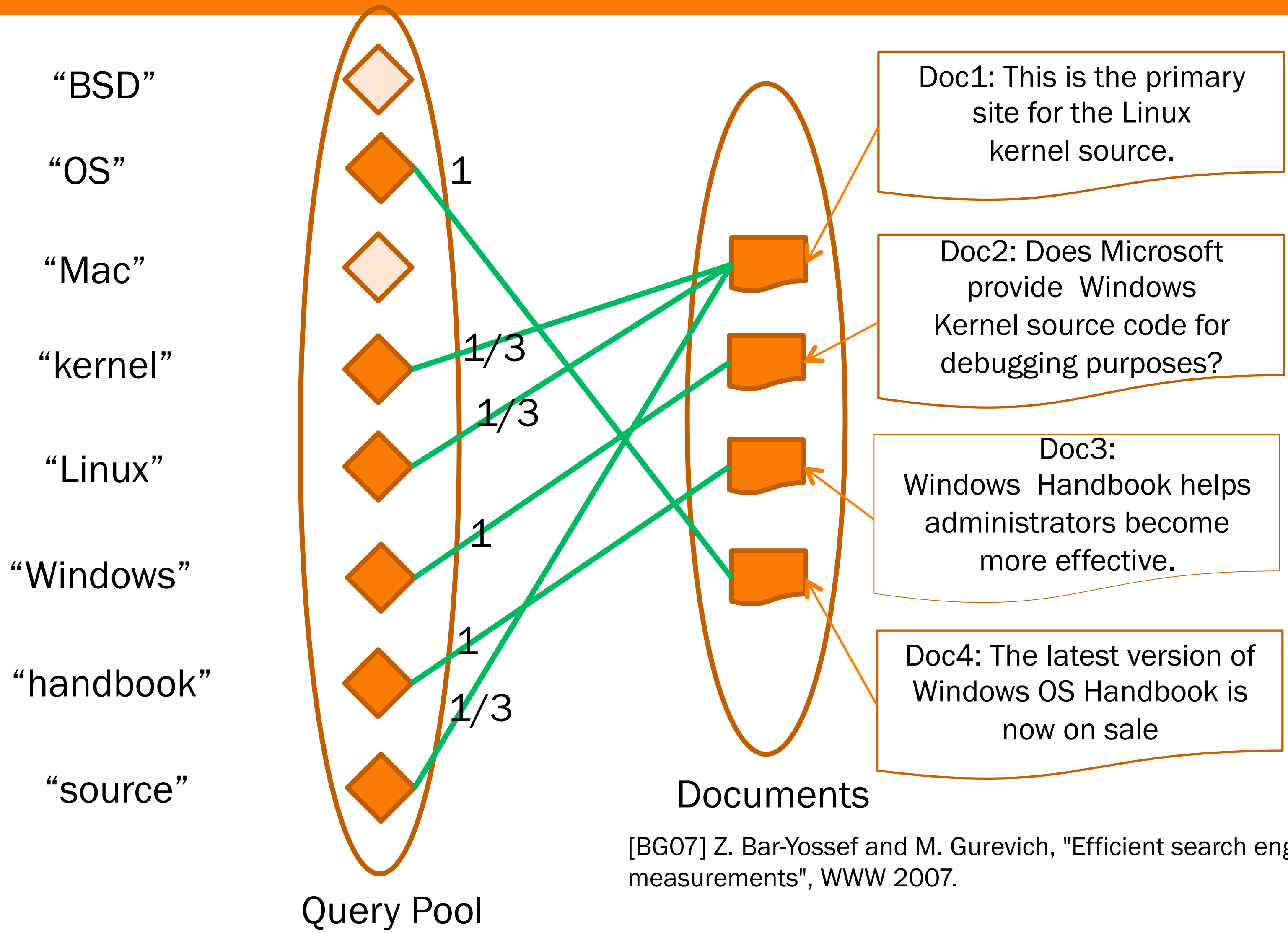
Problems with Mark-and-Recapture

- Problems
  - Correlation determination can be a tricky issue [BFJ+06]
    - e.g., C1: documents matching any five-digit number, C2: documents matching any medium frequency word – correlated
    - But – C1: documents matching exactly one five-digit number, C2 … exactly one medium frequency word – little correlation
  - Estimation bias
    - When using simple random samples, mark-and-recapture tends to be positively skewed [AMM05]
  - (In-) Efficiency: at least an expected number of $m^{1/2}$ samples required for a population of size m

[AMM05] S. C. Amstrup, B. F. J. Manly, and T. L. McDonald. *Handbook of capture-recapture analysis.* Princeton University Press, 2005.

Analytics Over Keyword Search Interfaces

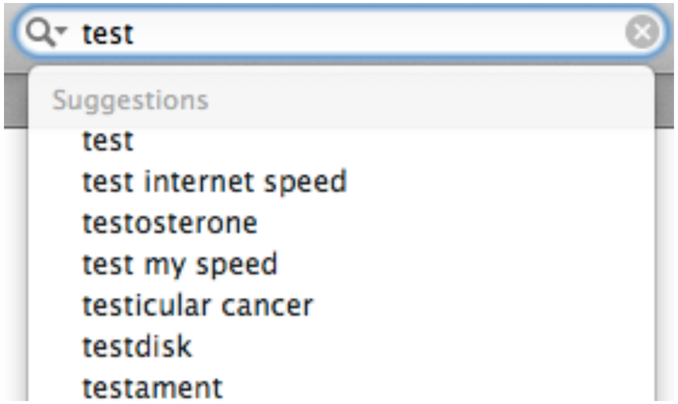
An Unbiased Estimator for COUNT and SUM

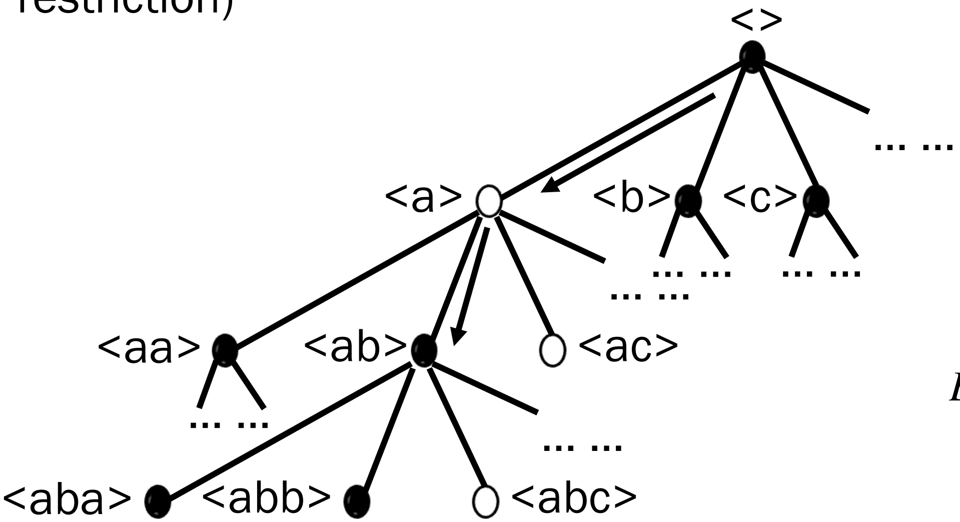


[BG07] Z. Bar-Yossef and M. Gurevich, "Efficient search engine measurements", WWW 2007.

Suggestion Sampling

Objective: perform analytics over a search engine's user query log, based on the auto-completion feature provide by the search engine (essentially an interface with prefix-query input restriction and top-k output restriction)





$$\text{When random walk stops at node } x$$

$$\text{Estimation for \# of search strings}: \frac{1}{p(x)}$$

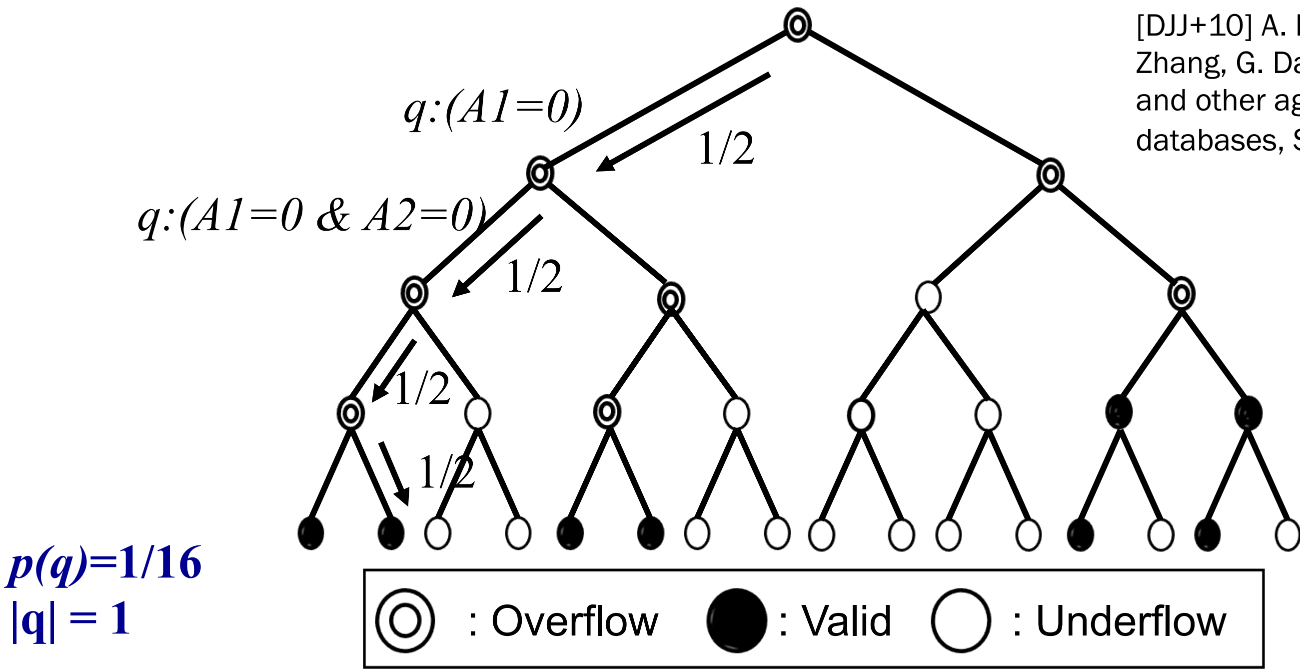
$$E\left[\frac{1}{p(x)}\right] = \sum_{x \text{ is marked}} p(x) . \frac{1}{p(x)} = \text{\# of marked nodes}$$

Z. Bar-Yossef and M. Gurevich. Mining search engine query logs via suggestion sampling. In *VLDB*, 2008.

Analytics Over Form-Like Interfaces

An Unbiased Estimator for COUNT and SUM

[DJJ+10] A. Dasgupta, X. Jin, B. Jewell, N. Zhang, G. Das, Unbiased estimation of size and other aggregates over hidden web databases, SIGMOD 2010.



Basic Ideas

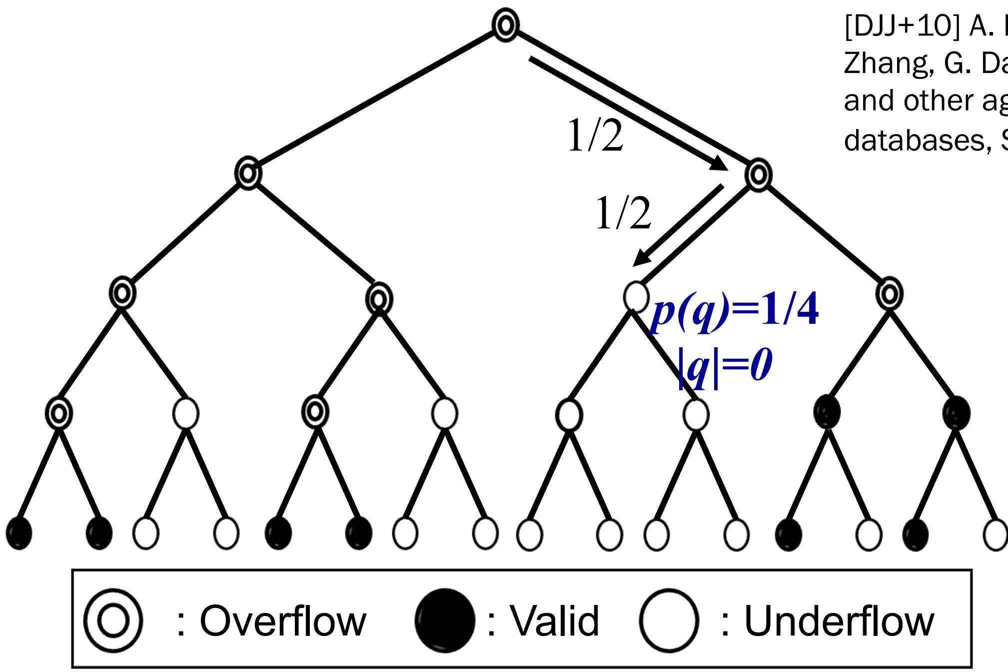
✓Continue drill down till valid or underflow is reached

✓ Size estimation as $\frac{|q|}{p(q)}$ (Hansen-Hurwitz Estimator)

✓ **Unbiasedness** of estimator $E\left[\frac{|q|}{p(q)}\right] = \sum_{q \in \Omega_{TV}} p(q).\frac{|q|}{p(q)} = m$

Analytics Over Form-Like Interfaces

An Unbiased Estimator for COUNT and SUM



[DJJ+10] A. Dasgupta, X. Jin, B. Jewell, N. Zhang, G. Das, Unbiased estimation of size and other aggregates over hidden web databases, SIGMOD 2010.

Basic Ideas

✓Continue drill down till valid or underflow is reached

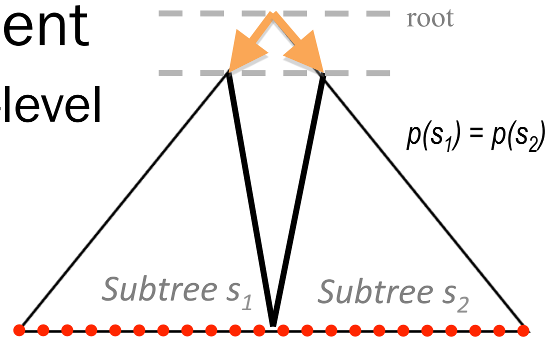
✓ Size estimation as $\frac{|q|}{p(q)}$ (Hansen-Hurwitz Estimator)

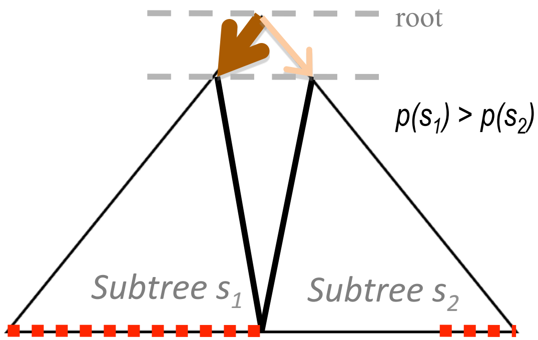
✓ **Unbiasedness** of estimator $E\left[\frac{|q|}{p(q)}\right] = \sum_{q \in \Omega_{TV}} p(q).\frac{|q|}{p(q)} = m$

Analytics Over Form-Like Interfaces

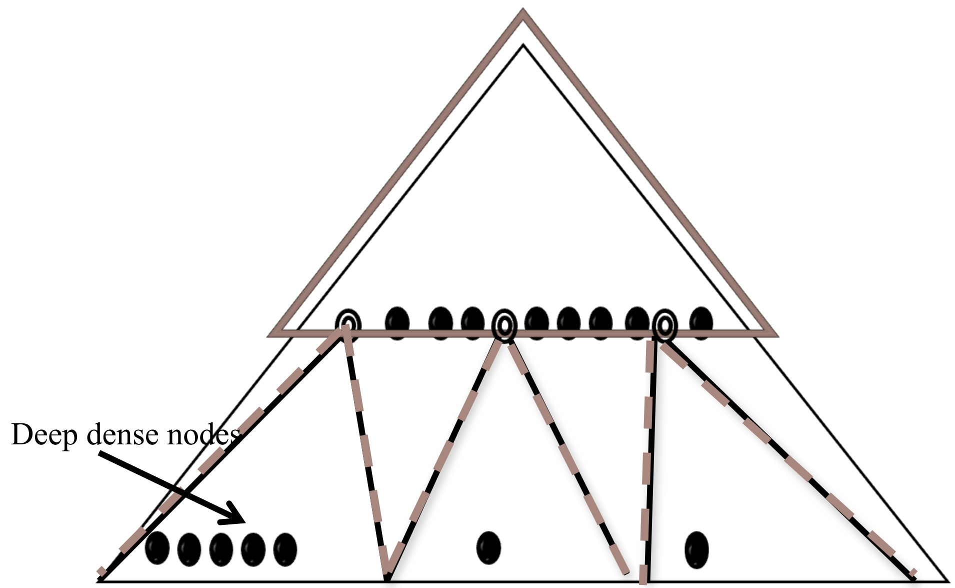
Variance Reduction

- Weight Adjustment
  - Addresses low-level low-cardinality nodes





- Divide-and-Conquer
  - Addresses deep-level dense nodes



[DJJ+10] A. Dasgupta, X. Jin, B. Jewell, N. Zhang, G. Das, Unbiased estimation of size and other aggregates over hidden web databases, SIGMOD 2010.

Analytics Over Form-Like Interfaces
Variance Reduction

- Stratified Sampling [LWA10]
- Adaptive sampling
  - e.g., adaptive neighborhood sampling: start with a simple random sample, then expand it with adding tuples from the neighborhood of sample tuples [WA11]
- Analytics Support for Data Mining Tasks
  - Frequent itemset mining [LWA10, LA11], differential rule mining [LWA10]

[LWA10] Tantan Liu, Fan Wang, Gagan Agrawal: Stratified Sampling for Data Mining on the Deep Web. ICDM 2010
[WA11] Fan Wang, Gagan Agrawal: Effective and efficient sampling methods for deep web aggregation queries. EDBT 2011
[LA11] Tantan Liu, Gagan Agrawal: Active learning based frequent itemset mining over the deep web. ICDE 2011

Analytics Over Graph Browsing Interfaces

Uniqueness of Graph Analytics

- Observation: uniqueness of analytics over graph browsing
  - Aggregates over a graph browsing interface may be defined on not only the underlying tuples (i.e., each user's information), but also the graph topology itself (i.e., relationship between users)
  - Examples: Graph cut, size of max clique, other topological measures
- Implication of the uniqueness
  - It is no longer straightforward how a sample of nodes can be used to answer aggregates
  - Efficiency and accuracy of analytics now greatly depend on what topological information the interface reveals, e.g.,
    - Level 1: a query is needed to determine whether user A befriends B.
    - Level 2: a query reveals the list of user A's friends.
    - Level 3: a query reveals the list of user A's friends, as well as the degree of each friend.

Analytics Over Graph Browsing Interfaces
Relationship with Graph Testing

- Graph Testing [GGR98, TSL10]
  - Input: a list of vertices
  - Interface: a query is needed to determine if there is an edge between two vertices
  - Objective: Approximately answer certain graph aggregates (e.g., k-colorability, size of max clique) while minimizing the number of queries issued.
- Differences with Graph Testing
  - The list of vertices is not pre-known
  - More diverse interface models
  - More diverse aggregates
    - e.g., on user attributes
    - e.g., defined over a local neighborhood

Example: k-colorability [GGR98].

A simple algorithm of sampling $O(k^2\log(k/\delta)/\varepsilon^3)$ vertices and testing each pair of them can construct a k-coloring of all n vertices such as at most εn^2 edges violate coloring rule.

[GGR98] O. Goldreich, S. Goldwasser, and D. Ron, "Property testing and its connection to learning and approximation", JACM, vol. 45, 1998.
[TSL10] Y. Tao, C. Sheng, and J. Li, "Finding Maximum Degrees in Hidden Bipartite Graphs", SIGMOD 2010.

Outline

- Introduction
- Resource Discovery and Interface Understanding
- Technical Challenges for Data Exploration
- Crawling
- Sampling
- Data Analytics
- Final Remarks

Conclusions

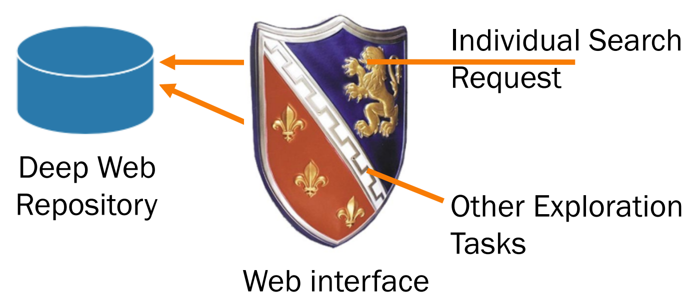
- Challenges
  - Resource discovery
  - Interface understanding
  - Data exploration
- Enabling Data Mining
  - Tasks: Crawling, Sampling, Analytics
  - Interfaces: Keyword search, form-like search, graph browsing

Traditional Heuristic Approaches

- e.g., seed-query based bootstrapping for crawling
- e.g., query sampling for repository sampling
- No guarantee on query cost, accuracy, etc.

Recent Approaches with Theoretical Guarantees

- e.g., performance-bounded crawlers
- e.g., unbiased samplers and aggregate estimators
- Techniques built upon sampling theory, etc.



Open Challenges

- Is the black-box approach still viable?
  - high cost of acquiring samples => significantly smaller sample size
  - poor performance of small-sized simple random sample [PK99]
- Two key challenges
  - Deeper integration of sampling and data mining algorithms
  - Workload-aware sampling / aggregate estimation algorithms for deep web databases

Open Challenges

- Website-Imposed Challenge
  - Dynamic data - when aggregates change rapidly
    - e.g., Twitter, financial data, etc.
  - Hybrid of interfaces
  - Many others…
- Privacy Challenge
  - From an owner's perspective: should aggregates be disclosed?
  - This challenge forms a sharp contrast with most existing work on data privacy (which focuses on protecting individual tuples while properly disclosing aggregate information for analytical purposes)
    - Here we must disclose individual tuples while suppressing access to aggregates
    - Recent work: dummy tuple insertion [DZDC09], correlation detection [WAA10], randomized generalization [JMZD11], adaptive query processing [ZZD12]

[DZD09] A. Dasgupta, N. Zhang, G. Das, and S. Chaudhuri, Privacy Preservation of Aggregates in Hidden Databases: Why and How? SIGMOD 2009.
[WAA10] S. Wang, D. Agrawal, and A. E. Abbadi, "HengHa: Data Harvesting Detection on Hidden Databases", CCSW 2010.
[JMZD11] X. Jin, A. Mone, N. Zhang, and G. Das, Randomized Generalization for Aggregate Suppression Over Hidden Web Databases, PVLDB 2011.
[ZZD12] M. Zhang, N. Zhang, and G. Das, Aggregate Suppression for Enterprise Search Engines, SIGMOD 2012.

References

- [AHK+07] Y. Ahn, S. Han, H. Kwak, S. Moon, and H. Jeong, "Analysis of Topological Characteristics of Huge Online Social Networking Services", WWW, 2007.
- [AMS+96] R Agrawal, H Mannila, R Srikant, H Toivonen, A I Verkamo, Fast Discovery of Association Rules, Advances in knowledge discovery and data mining, 1996.
- [BB98] K. Bharat and A. Broder, "A technique for measuring the relative size and overlap of public Web search engines", WWW, 1998.
- [BFJ+06] A. Broder, M. Fontura, V. Josifovski, R. Kumar, R. Motwani, S. Nabar, R. Panigrahy, A. Tomkis, and Y. Xu, "Estimating corpus size via queries", CIKM 2006.
- [BG07] Z. Bar-Yossef and M. Gurevich, "Efficient search engine measurements", WWW, 2007.
- [BG08] Z. Bar-Yossef and M. Gurevich, "Random sampling from a search engine's index", JACM, vol. 55, 2008.
- [BGG+03] M. Bawa, H. Garcia-Molina, A. Gionis, and R. Motwani, "Estimating Aggregates on a Peer-to-Peer Network," Stanford University Tech Report, 2003.
- [Cat91] J. Catlett, Megainduction: Machine Learning on Very Large Database. PhD thesis, School of Computer Science, University of Technology, Sydney, Australia, 1991
- [CD09] S. Chaudhuri and G. Das, "Keyword querying and Ranking in Databases", VLDB, 2009.
- [CGG10] Toon Calders, Calin Garboni, Bart Goethals, Efficient Pattern Mining of Uncertain Data with Sampling, PAKDD 2010.
- [CHH+05] S Cong, J Han, J Joeflinger, D Padua, A sampling-based framework for parallel data mining, PPoPP 2005.
- [CHW+08] M. J. Cafarella, A. Halevy, D. Z. Wang, E. Wu, and Y. Zhang, "WebTables: exploring the power of tables on the web", VLDB, 2008.
- [CLR+10] L. Chiticariu, Y. Li, S. Raghavan, and F. Reiss, "Enterprise Information Extraction: Recent Develop-ments and Open Challenges", SIGMOD, 2010.
- [CM10] A. Cali and D. Martinenghi, "Querying the Deep Web (Tutorial)", EDBT, 2010.
- [CMH08] M. J. Cafarella, J. Madhavan, and A. Halevy, "Web-Scale Extraction of Structured Data", SIGMOD Record, vol. 37, 2008.
- [CPW+07] D. H. Chau, S. Pandit, S. Wang, and C. Faloutsos, "Parallel Crawling for Online Social Networks", WWW, 2007.
- [CVD+09] X. Chai, B.-Q. Vuong, A. Doan, and J. F. Naughton, "Efficiently Incorporating User Feedback into Information Extraction and Integration Programs", SIGMOD, 2009.

References

- [CWL+09] Y. Chen, W. Wang, Z. Liu, and X. Lin, "Keyword Search on Structured and Semi-Structured Data (Tutorial)", SIGMOD, 2009.
- [Das03] G. Das, "Survey of Approximate Query Processing Techniques (Tutorial)", SSDBM, 2003.
- [DCL+00] M. Diligenti, F. M. Coetzee, S. Lawrence, C. L. Giles, and M. Gori, "Focused crawling using context graphs", VLDB, 2000.
- [DDM07] A. Dasgupta, G. Das, and H. Mannila, "A random walk approach to sampling hidden databases", SIGMOD, 2007.
- [DJJ+10] A. Dasgupta, X. Jin, B. Jewell, and G. Das, "Unbiased estimation of size and other aggregates over hidden web databases", SIGMOD, 2010.
- [DKP+08] G. Das, N. Koudas, M. Papagelis, and S. Puttaswamy, "Efficient Sampling of Information in Social Networks", CIKM/SSM, 2008.
- [DKY+09] E. C. Dragut, T. Kabisch, C. Yu, and U. Leser, "A Hierarchical Approach to Model Web Query Interfaces for Web Source Integration", VLDB, 2009.
- [DN09] X. Dong and F. Nauman, "Data fusion - Resolving Data Conflicts for Integration", VLDB, 2009.
- [DZD09] A. Dasgupta, N. Zhang, and G. Das, "Leveraging COUNT Information in Sampling Hidden Databases", ICDE, 2009.
- [DZD10] A. Dasgupta, N. Zhang, and G. Das, "Turbo-charging hidden database samplers with overflowing queries and skew reduction", EDBT, 2010.
- [DZD+09] A. Dasgupta, N. Zhang, G. Das, and S. Chaudhuri, "Privacy Preservation of Aggregates in Hidden Databases: Why and How?", SIGMOD, 2009.
- [FHM08] M. Franklin, A. Halevy, and D. Maier, "A First Tutorial on Dataspaces", VLDB, 2008.
- [GG01] M. Garofalakis, P. Gibbons: Approximate Query Processing: Taming the TeraBytes. VLDB 2001.

References

- [GGR98] O. Goldreich, S. Goldwasser, and D. Ron, "Property testing and its connection to learning and approximation", JACM, vol. 45, 1998.
- [GKBM10] M. Gjoka, M. Kurant, C. Butts, and A. Markopoulou, "Walking in Facebook: A Case Study of Unbiased Sampling of OSNs", INFOCOM, 2010.
- [GM08] L. Getoor and R. Miller, "Data and Metadata Alignment: Concepts and Techniques)", ICDE, 2008.
- [GMS06] C. Gkantsidis, M. Mihail, and A. Saberi, "Random walks in peer-to-peer networks: algorithms and evaluation", Performance Evaluation - P2P computing systems, vol. 63, 2006.
- [IG02] P. G. Iperirotis and L. Gravano, "Distributed Search over the Hidden Web: Hierarchical Database Sampling and Selection", VLDB, 2002.
- [JZD11] X. Jin, N. Zhang, G. Das, “Attribute Domain Dis-covery for Hidden Web Databases”, SIGMOD 2011.
- [KBG+01] O. Kaljuvee, O. Buyukkokten, H. Garcia-Molina, and A. Paepcke, "Efficient Web Form Entry on PDAs", WWW, 2001.
- [LCK98] S.D Lee, D.W. Cheung, and B. Kao. Is sampling useful in data mining? a case in the maintenance of discovered association rules. Data Mining and Knowledge Discovery, 2:233-262, 1998.
- [LHY+08] X Li, J Han, Z Yin, J-G Lee, Y Sun, Sampling cube: a framework for statistical OLAP over sampling data, SIGMOD 2008.
- [LWA10] T. Liu, F. Wang, and G. Agrawal, "Stratified Sampling for Data Mining on the Deep Web", ICDM, 2010.
- [LYM02] K.-L. Liu, C. Yu, and W. Meng, "Discovering the representative of a search engine", CIKM, 2002.
- [MAA+09] J. Madhavan, L. Afanasiev, L. Antova, and A. Halevy, "Harnessing the Deep Web: Present and Future", CIDR, 2009.
- [MH98] M. Meila and D. Hecherman. An experimental comparison of several clustering and ini- tialization methods. Technical report, Center for Biological and Computational Learn- ing, MIT, 1998.
- [MMG+07] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and Analysis of Online Social Networks", IMC, 2007.
- [NZC05] A. Ntoulas, P. Zerfos, and J. Cho, "Downloading Textual Hidden Web Content through Keyword Queries", JCDL, 2005.
- [PF00] C. R. Palmer and C. Faloutsos. Density biased sampling: An improved method for data mining and clustering. SIGMOD 2000.
- [PK99] F. Provost and V. Kolluri. A survey of methods for scaling up inductive algorithms. Machine Learning, pages 1-42, 1999.
- [Qui86] J. Quinlan. Induction of decision trees. Machine Learning, pages 81 - 106, 1986.
- [RG01] S. Raghavan and H. Garcia-Molina, "Crawling the Hidden Web", VLDB, 2001.
- [RT10] B. Ribeiro and D. Towsley, "Estimating and sampling graphs with multidimensional random walks", IMC, 2010.
- [SDH08] A. D. Sarma, X. Dong, and A. Halevy, "Bootstrapping Pay-As-You-Go Data Integration Systems", SIGMOD, 2008.

References

- [SZS+06] M. Shokouhi, J. Zobel, F. Scholer, and S. Tahaghoghi, "Capturing collection size for distributed non-cooperative retrieval", SIGIR, 2006.
- [Toi96] H. Toivonen. Sampling large databases for association rules. In Proceedings of the 22nd International Conference on Very Large Data Base (VLDB'96). Morgan Kaufmann, 1996.
- [TSL10] Y. Tao, C. Sheng, and J. Li, "Finding Maximum Degrees in Hidden Bipartite Graphs", SIGMOD 2010.
- [WA11] F. Wang, G. Agrawal, “Effective and Efficient Sampling Methods for Deep Web Aggregation Queries”, EDBT 2011.
- [WAA10] S. Wang, D. Agrawal, and A. E. Abbadi, "HengHa: Data Harvesting Detection on Hidden Databases", ACM Cloud Computing Security Workshop, 2010.
- [WT10] G. Weikum and M. Theobald, "From Information to Knowledge: Harvesting Entities and Relationships from Web Sources (Tutorial)", PODS, 2010.
- [YHZ+10] X. Yan, B. He, F. Zhu, J. Han, "Top-K Aggregation Queries Over Large Networks", ICDE, 2010
- [ZHC04] Z. Zhang, B. He, and K. C.-C. Chang, "Understanding Web Query Interfaces: Best-Effort Parsing with Hidden Syntax", SIGMOD, 2004.
- [ZPLO97] M.J. Zaki, S. Parthasarathy, W. Li, and M. Ogihara. Evaluation of sampling for data mining of association rules. In Proceedings of the 7th Worhshop on Research Issues in Data Engineering, 1997.
- [ZZD11] M. Zhang, N. Zhang, and G. Das, Mining Enterprise Search Engine's Corpus: Efficient Yet Unbiased Sampling and Aggregate Estimation, SIGMOD 2011.

Thank you

Questions?

Contact: nzhang10@gwu.edu, gdas@uta.edu