Not Quite Frequent Itemsets

Jouni K Seppänen

T-122.103 Algorithmic methods in data mining

October 17, 2003

(Heikki Mannila had to go to an important meeting,
so I am giving today's lecture.)

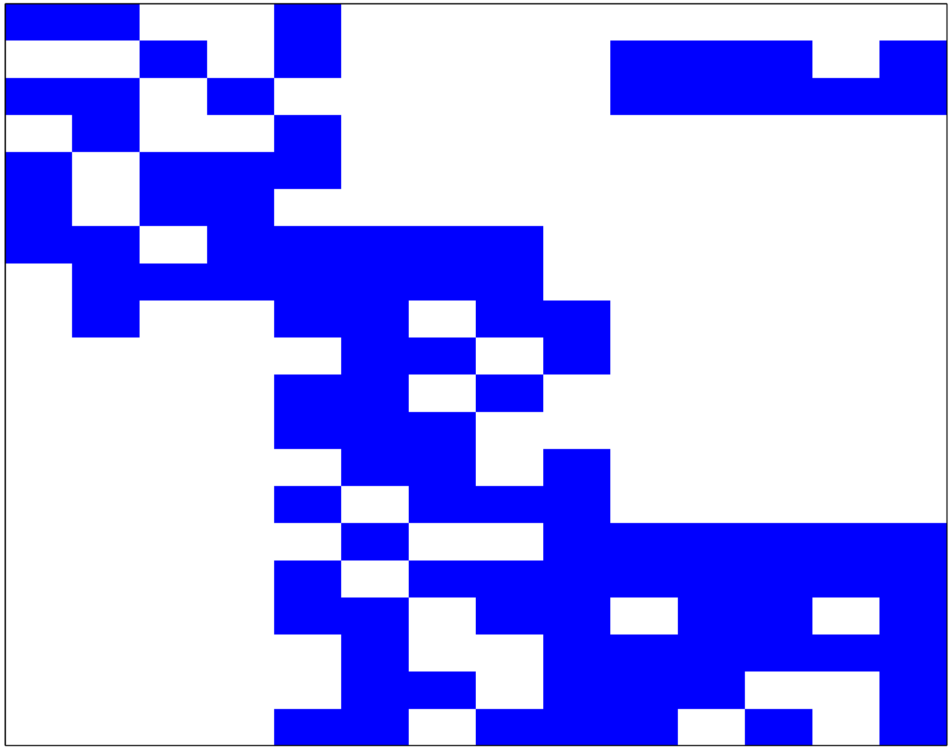
Problems with frequent itemsets

- Number of sets is large,
  potentially exponential
  (mining maximal sets helps a little)

- Unrealistic assumption that
  all connected attributes always co-occur

- In example picture, the three (intuitively)
  best-connected attribute sets are

$$\{1, 2, 3, 4, 5\}$$
$$\{5, 6, 7, 8, 9\}$$
$$\{10, 11, 12, 13, 14\}$$



Error-Tolerant Frequent Itemsets

- Cheng Yang, Usama Fayyad, & Paul S. Bradley. **Efficient Discovery of Error-Tolerant Frequent Itemsets in High Dimensions**. KDD 2001. `http://www-db.stanford.edu/~yangc/pub/cy-kdd01.pdf`

- Notation: r is a binary relation over R, with $|r| = n$; we denote the value of an item $A \in R$ in a transaction $T \in r$ by $r[T, A]$

- Definition: An itemset $E \subseteq I$ is an **error-tolerant itemset** (ETI) with error ϵ and support κ with respect to a database D that has n transactions, if in at least κn transactions of r at least a fraction $1 - \epsilon$ of the items in E are present.

- Maximal ETIs: ETIs whose supersets are not ETIs

- Immediate problem: if some attributes form an ETI E_0 with error $\ll \epsilon$, then other attributes can get a free ride along with E_0 to generate lots of spurious ETIs $E_j = E_0 \cup \{ A_j \}$

Algorithmic problem

- Unlike the ordinary support of frequent itemsets, the ETI property is not monotonic!

A	B	C	D
1	1	0	0
1	0	1	0
1	0	0	1
0	1	1	0
0	1	0	1
0	0	1	1

- With $\kappa = 1$, the set A has error 1, the set AB has error 1, the set ABC has error 0.67, and the set $ABCD$ has error 0.5. Thus a levelwise algorithm with $\kappa = 1$, $\epsilon = 0.5$ would discard all (proper nonempty) subsets of $ABCD$, although $ABCD$ is an ETI.

Solution to problem

- Weaken ETI definition: A **weak ETI** consists of a set of items $E \subseteq R$ and a set $T \subseteq r$ of transactions such that $|T| \geq \kappa n$ and

$$\frac{\sum_{X \in T, A \in E} r[X, A]}{|T| \cdot |E|} \geq 1 - \epsilon.$$

- An ETI is always a weak ETI, so finding weak ETIs and then filtering is enough

- Weak ETIs are not monotonic in the usual manner either,
  but the following result can be proved:

  Lemma ETI1. If E is a weak ETI with $|E| = m$, there is a weak ETI $E' \subseteq E$ with $|E'| = m - 1$.

  (Idea: remove the item that has the fewest 1s in T.)

Finding weak ETIs

- Itemset support is easy to compute in one pass through the database: count how many transactions include all the items in the candidate itemset.

- Weak ETIs seem more complicated, but a simple one-pass algorithm is possible:

  Lemma ETI2. The following algorithm computes the error rate of a weak ETI E with $|E| = m$: Keep counters C_j $(j = 0, \ldots, m)$, recording in C_j the number of transactions that have exactly j of the items in m. Then the error rate of E at support κ is

$$\delta(E) = \frac{1}{\kappa n m}\left[(m-t)\left(\kappa n - \sum_{j=t+1}^{m} C_j\right) + \left(\sum_{j=t+1}^{m}(m-j)C_j\right)\right],$$

  where t is the largest number such that $\sum_{j=t}^{m} C_j \geq \kappa n$.

  (Intuition: take the transactions that have the largest intersection with E.)

Putting it all together

- Algorithm to find all (strong) ETIs:
  1. Make all singletons into candidate sets.
  2. Find the weak-ETI error rate $\delta(E)$ of each candidate E using Lemma ETI2.
  3. Prune the candidates that have $\delta(E) < \epsilon$.
  4. Remember remaining candidates as weak ETIs.
  5. Form new candidates by adding into each remaining candidate E each item $A \in R \setminus E$. (Lemma ETI1)
  6. If there are candidates, go to 2.
  7. Go through the database and check for each weak ETI whether it is a strong ETI.

Unfortunately. . .

- The number of candidates examined is huge
- The free-rider problem is even worse with weak ETIs:
  it is also possible to pad "dense but small" weak ETIs with all-zero transactions.
- Solution: heuristics and approximations

Approximate algorithm

- Change algorithm:

  1. When adding items (dimensions), consider only those items that have support $\geq \kappa n$.
  2. When adding items to E, check that the new item has a fraction $\leq \epsilon$ of zeros within the transactions of E.
  3. Do not add all possible dimensions, only the best one.

- Thus there will at all times be at most $|R|$ candidates, and since the number of rounds through the database is restricted by $|R|$, the algorithm is worst-case quadratic wrt $|R|$.

- Of course, some ETIs will be missed...

Iterative approximate algorithm

- Run approximate algorithm many times. After each run, remove from the database those transactions that are covered by the found ETIs, and also reduce the support threshold by a factor of $\lambda(= 2)$.

- Now those sets may be missed that occur mostly in conjunction with other sets that have smaller error rates.

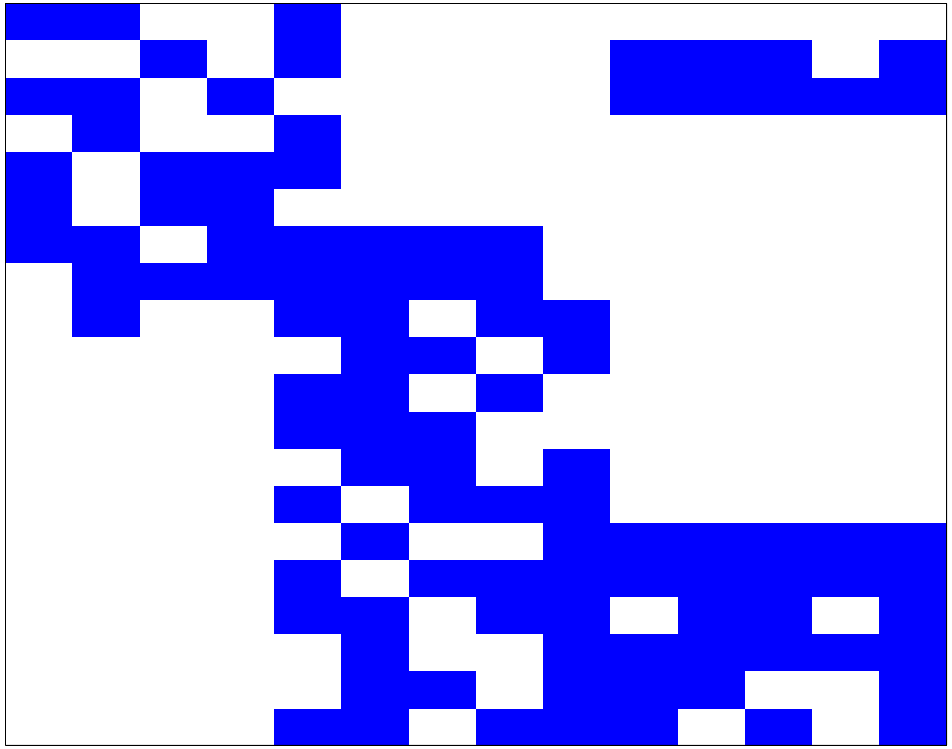
- This is not a big problem for the main application of Yang et al., which is to initialize a mixture model.

Iterative sampling and validation

- Speed up the database pass
- Instead of running the algorithm on the whole database, sample two non-overlapping subdatabases, run the greedy algorithm on one and validate on the other.
- Wrap the sampling algorithm within the iterative layer

Topic Models

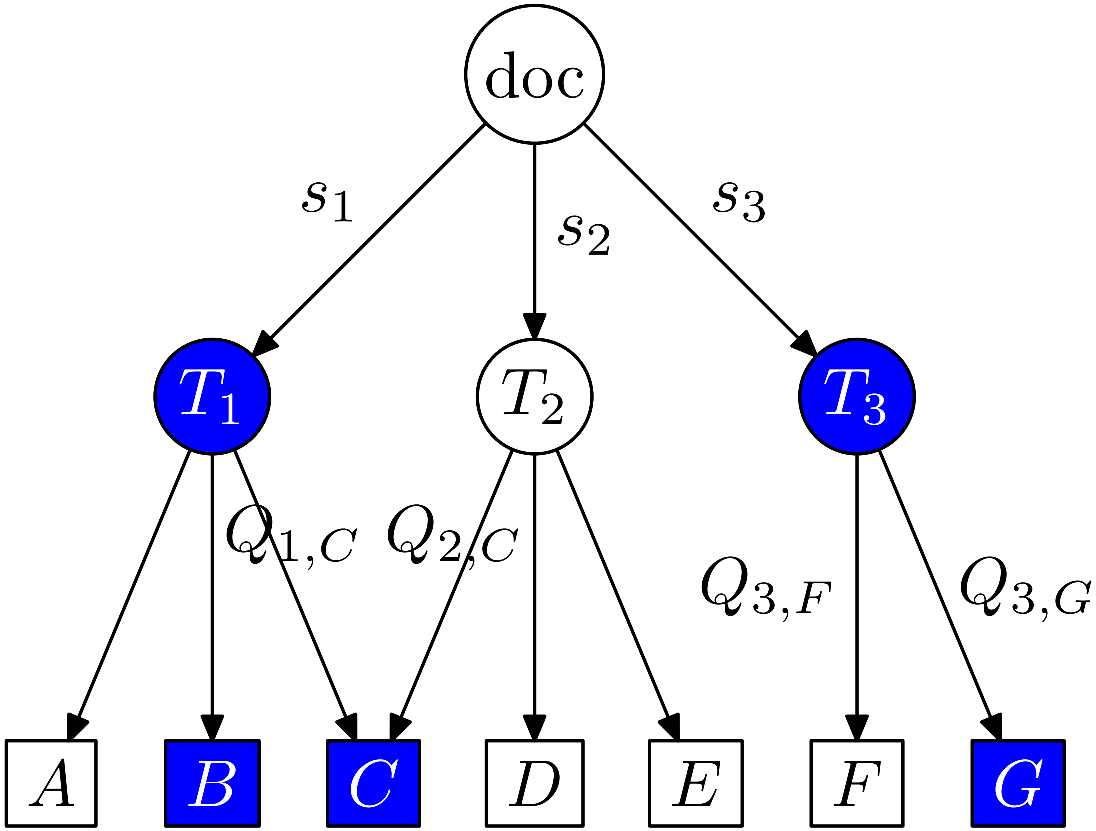
- Jouni K. Seppänen, Ella Bingham, & Heikki Mannila. **Simple algorithms for topic identification in 0–1 data**. PKDD 2003. `http://www.cis.hut.fi/ella/publications/cameraready.pdf`



- Model for binary data
  - E.g. word/document
  - Focus: positive connections between variables
- Topic ≈ a set of variables that tend to occur together

Topic Models

- In any given document, some topics T_i are active with some probabilities s_i, independently of each other.
- If topic T_i is active, it produces attribute A with probability $Q_{i,A}$.
- Topics act independently.



Other approaches

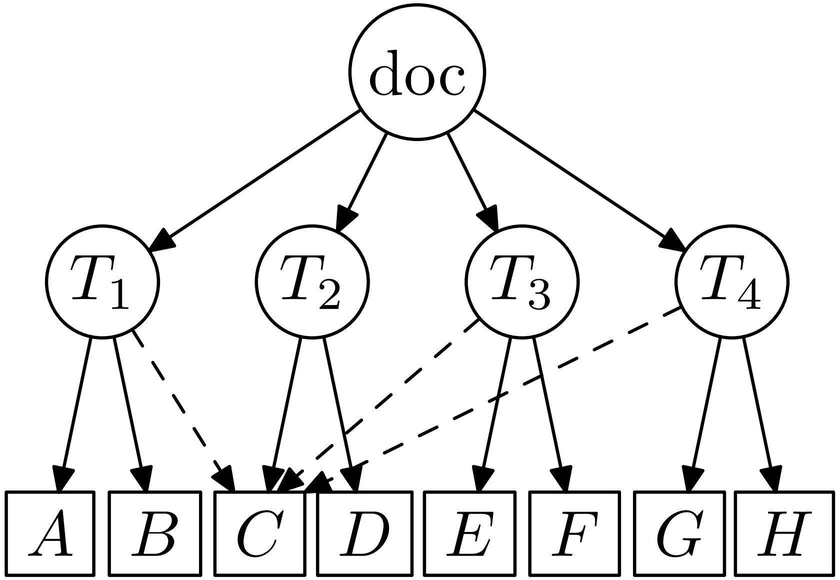
- Latent semantic analysis (SVD)
- Independent component analysis
- Nonnegative matrix factorization
- Probabilistic latent semantic analysis (EM algorithm)
- Latent Dirichlet allocation (variational method)
- Multinomial PCA
- Mixtures of Bernoulli distributions
- Frequent itemsets

Difficulty of topic assignment

- The following problem is NP-complete.
  - Given: topic model parameters, single data vector ("document"), threshold ρ
  - Question: is there a combination of topic activations that can explain the observed data so that the probability of the data is $\geq \rho$?
- Model structure is (to us) more interesting than the topics that explain individual observations

Assumptions on topic models

- Small topic probabilities (but not too small)
- Every attribute should have a **primary topic** to which it is most strongly connected
  - θ-bounded conspiracy: $\sum_{j \neq i} Q_{j,A} \leq \theta Q_{i,A}$



Lift statistic

- What kind of information can tell us that attributes A, B belong to the same topic?
  - Idea: if $P(A \mid B) \gg P(A)$, this should be the case.
  - Define
$$\mathrm{Lift}(A, B) = \frac{P(A \mid B)}{P(A)} = \frac{P(AB)}{P(A)P(B)}.$$
- Assume that A is a **core attribute** of topic T_i, i.e., that only topic T_i can generate A.
  - For any word B,
$$\mathrm{Lift}(A, B) = \frac{P(T_i \mid B)}{P(T_i)}.$$
  - If B is also a core attribute of T_i, then $P(T_i \mid B) = 1$.

Lift statistic

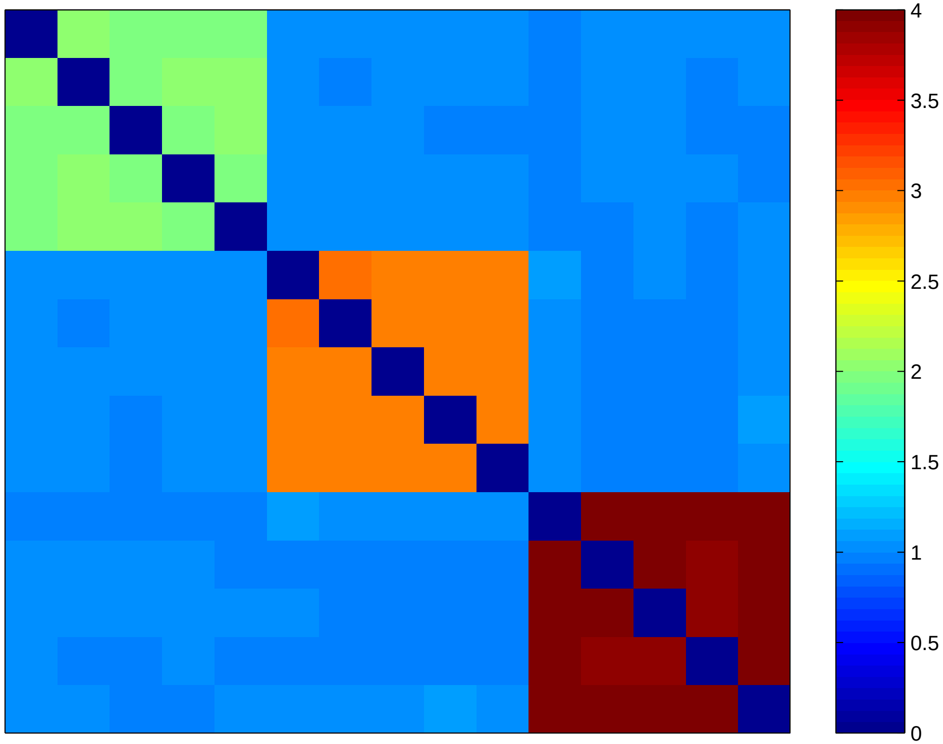
- Thus, if A and B are core attributes of topic T_1,

$$\mathrm{Lift}(A, B) = P(T_1)^{-1}.$$

- If A is a core attribute of T_1 and B a core attribute of T_2,

$$\mathrm{Lift}(A, B) = 1.$$

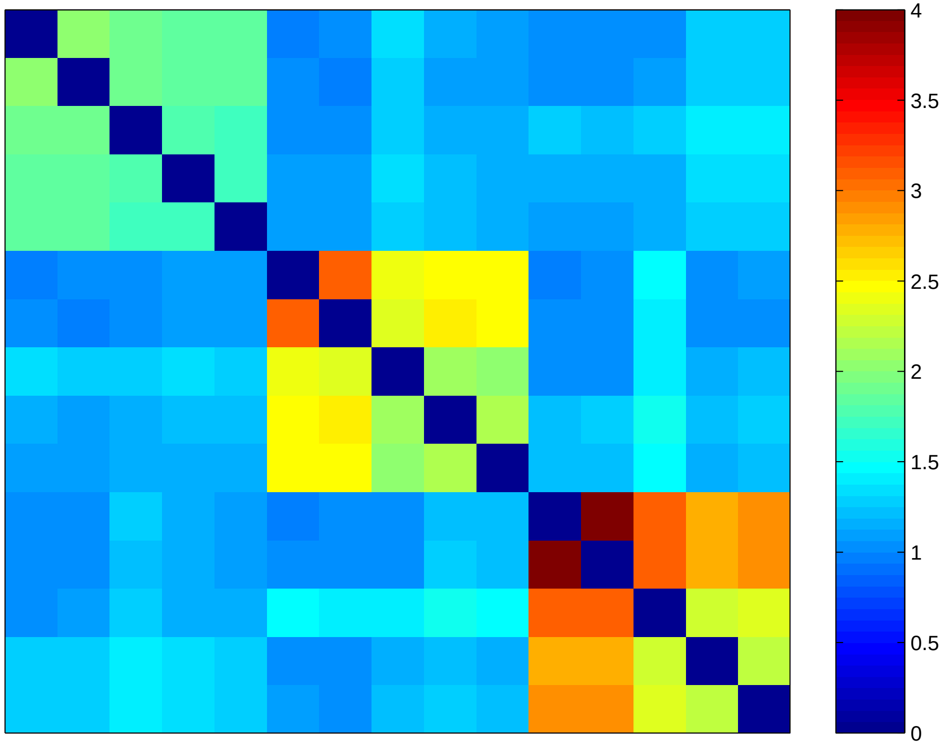
- Thus, if $P(T_j) \ll 1$ for all j, and if each attribute belongs to a single topic only, we can recognize the topics by looking at the lift statistics.



Lift statistic

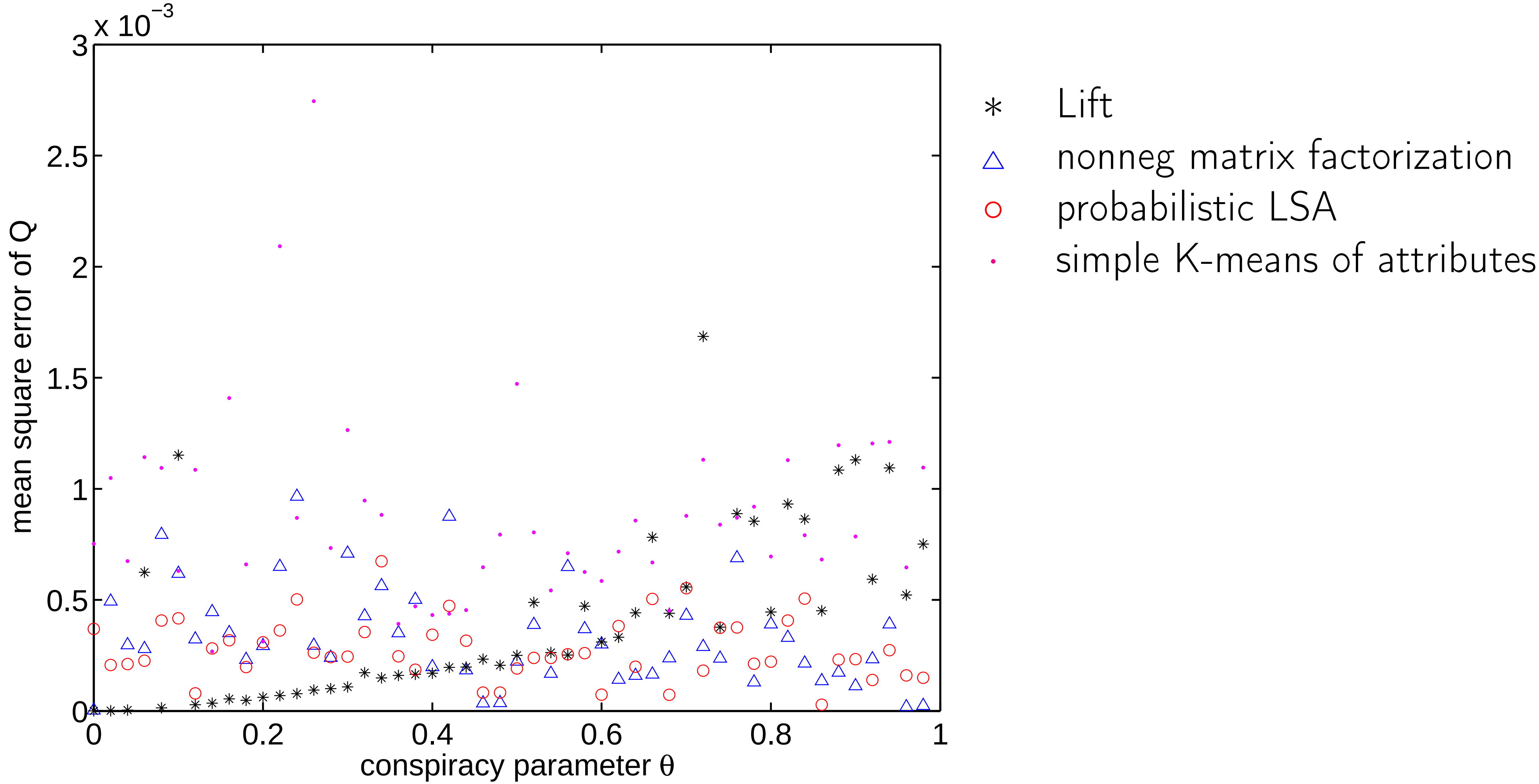
Lift statistic

- With non-core attributes, the situation is more complicated.
- However, if some core attributes can be identified, the others can be assigned in a separate pass.
- Algorithm:
  1. Identify the core attributes by their lifts
  2. Cluster the core attributes (into non-overlapping sets)
  3. Assign the other attributes to the clusters (that now may overlap)



Lift statistic

Experiments: generated data



Experiments: topics from CS bibliography data

- algorithms approximation damath problems scheduling some tree two
- analysis distributed libtr probabilistic systems
- bounds communication complexity focs lower
- algorithm efficient fast ipl matching problem set simple
- design ieeetc network networks optimal parallel routing sorting
- note tcs
- finding graphs minimum planar polynomial sets sicomp time
- graph number properties random tr
- from information learning lncs theory
- approach jacm linear new programming system
- actainf binary search trees
- abstract computation extended model stoc
- automata finite languages mfcs
- data dynamic infctrl logic programs structures using
- applications icalp theorem
- cacm computer computing science
- crypto functions
- jcss machines
- algebraic beatcs computational geometry
- de stacs van
- codes dmath

Probe measure

- Gautam Das, Heikki Mannila, & Pirjo Ronkainen. **Similarity of attributes by external probes.** Knowledge Discovery and Data Mining, pp. 23–29, 1998. `http://citeseer.nj.nec.com/das97similarity.html`

- Lift (like correlation, or co-occurrence) is an **internal** measure of similarity, depending only on the values of the two attributes being compared.

- **External** similarity measures look at the values of other attributes. Classic example:
  - Pepsi is similar to Coke is similar to Generic Brand Cola
  - However, any two of the attributes co-occur very rarely
  - The **context** where the attributes appear is important: perhaps any of the three is usually bought together with chips

Probe measure

- Look at the distribution of attributes C that are external to the two attributes being compared; the attributes C serve as "probes":

$$\text{Probe}(A, B) = \sum_{\substack{C \in R \\ A \neq C \neq B}} \left| P(C|A) - P(C|B) \right|$$

Again: topics from CS bibliography data

- algorithm algorithms efficient fast graph graphs matching optimal parallel problem set simple
- actainf beatcs damath dmath focs geometry icalp infctrl ipl jacm jcss libtr mfcs sicomp stacs stoc tcs tr
- complexity functions machines probabilistic
- applications problems some
- approach de logic model programming programs system systems van
- network networks routing sorting
- computational information theory
- linear new two
- binary search tree trees
- polynomial time
- algebraic automata finite languages note properties sets theorem
- data structures
- analysis design distributed using
- computation computing
- bounds lower
- computer science
- from learning
- cacm crypto ieeetc lncs
- number random
- abstract extended
- finding minimum planar

Future work

- Computing all $|R|^2$ pairwise lift values takes time; computing the probe distances is even slower, since all external attributes are checked. Could the sparseness of the data be exploited?
- Is the model a good one? For what kind of data?