Addendum to the proof of log n approximation ratio for the greedy set cover algorithm

- (From Vazirani’s very nice book ”Approximation algorithms”)
- Let x_1, x_2,...,x_n be the order in which the elements are covered (break ties arbitrarily)
- Lemma: $c(x_i)<=C^*/(n-i+1)$
- Proof. Suppose we are selecting a set that will cover x_i.

  The remaining elements can be covered with C* sets.

  Thus there largest set in C*, the optimal solution, will cover at least (n-i+1)/C* elements.

  I.e., The cost per element is at most C*/(n-i+1)

Thus

- Theorem. The approximation cost is at most H(n)
- Proof. The cost is at most the sum of the costs c(x_i)

$$\sum_{i=1}^{n} c(x_i) \leq \frac{C*}{n-i+1} \leq C*\sum_{i=1}^{n}\frac{1}{n} \leq C*H(n)$$

- Proving the bound H(s) is more tedious.

Finding fragments of orders, partial orders, and total orders from 0-1 data

Themes of the chapter

- Given a 0/1 a matrix
- Rows: observations, columns variables
- Can one find ordering information for the observations?
- Without additional assumptions, no; with some assumptions, yes

- Paleontological application:
  - find orders for subsets of fossil sites
  - a good ordering for (a subset of) the rows is one where the 1s are consecutive

- Also other applications

Themes of the chapter

- Finding small total orders (fragments) from 0-1 data
  - Local models/patterns
- Finding partial orders from 0-1 data
  - A global model
- Find total orders for 0-1 data
  - A global model

Finding small total orders (fragments) from 0-1 data

- Model: a subset of observations and a total order on the subset

- Task: find **all** such models fulfilling certain criteria

- Algorithm: a pattern discovery algorithm (levelwise search)

Finding partial orders from 0-1 data

- Model: a partial order over all observations
- Loglikelihood: proportional to the number of cases the observed occurrence patterns violate the continuity of species
- Prior: prefer partial orders that are as specific as possible
- Task: find **a** model with high likelihood * prior
- Algorithm: Find fragments and use heuristic search to build a good partial order

Find total orders for 0-1 data

- Model: a total order
- Loglikelihood: how many cases the observed occurrence patterns violate the continuity of species
- Task: find **the** best total order for the observations
- Algorithm: spectral method

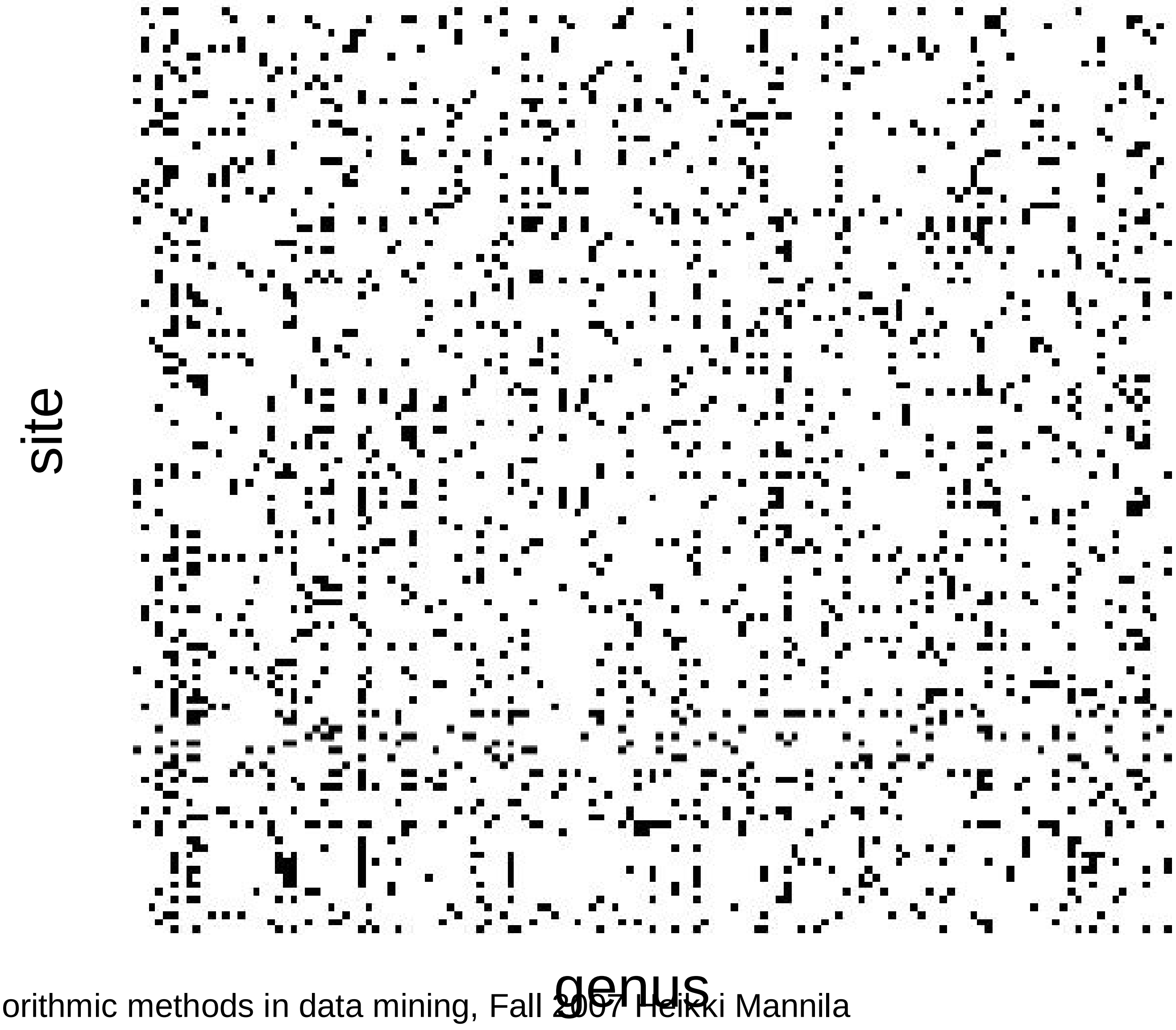
Type of data

- 0-1 data, large number of variables
- Examples:
  - Occurrences of words in documents
  - Occurrences of species in paleontological sites
  - Occurrence of a particular motif in a promoter region of a gene
- Typically the data is sparse: only a few 1s
- Asymmetry between 0s and 1s
  - A ”1” means that there really was something
  - A ”0” has less information (in a way)

Example

- Paleontological data from the NOW (Neogene Mammal Database)
- Fossil **sites** (one location, one layer)
- Each site contains fossils that are about the same age (+- 1 Ma)
- Variables: species/genera
- A ”1” is reasonably certain
- A ”0” might be due to several reasons
  - The species was not extant at that time
  - The remains did not fossilize
  - The tooth was overlooked
  - …

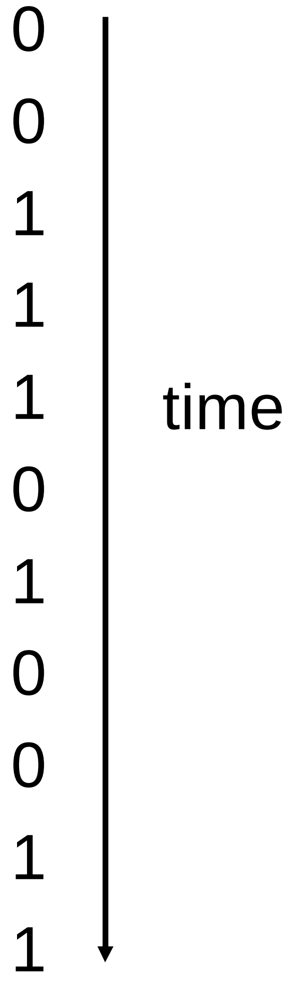
Site-genus -matrix



Background knowledge

- Species do not vanish and return

- An ordering of the sites with a ”0” between ”1”s is improbable

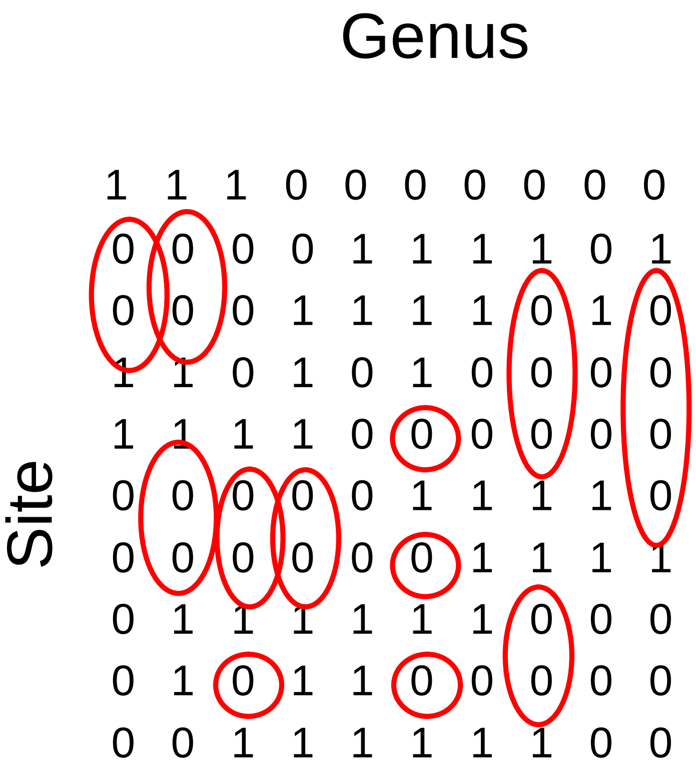


0
0
1
1
1
0
1
0
0
1
1

time

Example: seriation in paleontological data

- Given data about the occurrences of genera in fossil sites
- Want to find an ordering in which occurrences of a genus are consecutive
- **Lazarus count**: how many 0s are between 1s



Genus

Site

1	1	1	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	0	1
0	0	0	1	1	1	1	0	1	0
1	1	0	1	0	1	0	0	0	0
1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1	0
0	0	0	0	0	0	1	1	1	1
0	1	1	1	1	1	1	0	0	0
0	1	0	1	1	0	0	0	0	0
0	0	1	1	1	1	1	1	0	0

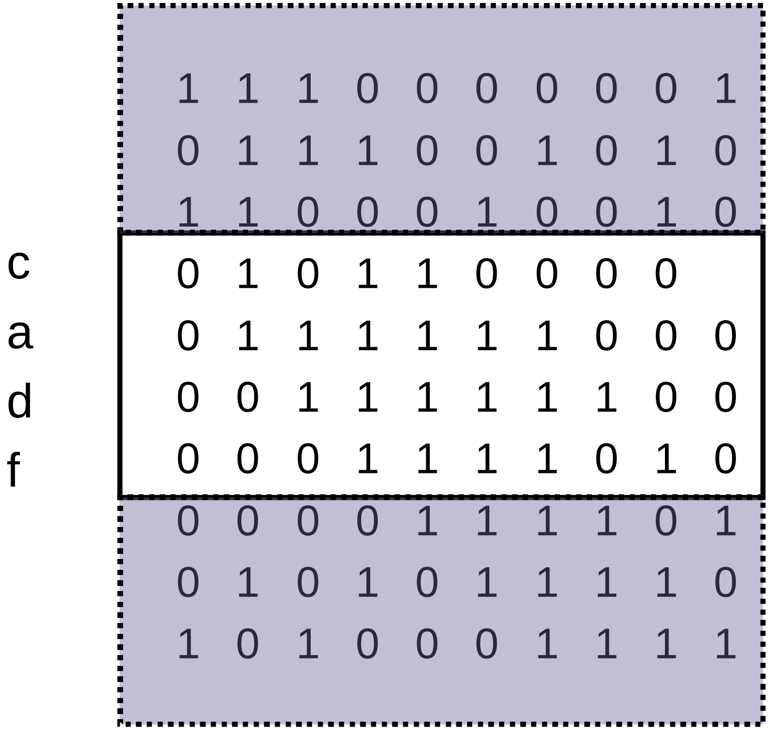
A better ordering

A smaller Lazarus count

$$\begin{array}{cccccccccc}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & \textcircled{0} & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & \textcircled{0} & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & \textcircled{0} & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & \textcircled{0} & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & \textcircled{0} \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1
\end{array}$$

Find small total orders (fragments) from 0-1 occurrence data

- Fragment: a total ordering of **a subset** of observations
- E.g., c<a<d<f
- Intuitive interpretation:
- For most variables the sequence of observations has no pattern of the form …1…0…1…



	1	1	1	0	0	0	0	0	0	1
	0	1	1	1	0	0	1	0	1	0
	1	1	0	0	0	1	0	0	1	0
c	0	1	0	1	1	0	0	0	0	
a	0	1	1	1	1	1	1	0	0	0
d	0	0	1	1	1	1	1	1	0	0
f	0	0	0	1	1	1	1	0	1	0
	0	0	0	0	1	1	1	1	0	1
	0	1	0	1	0	1	1	1	1	0
	1	0	1	0	0	0	1	1	1	1

Fragments of order

- 0/1 data set
- Fragment of order *f* is a sequence of observations

  $t_1 < t_2 < t_3 < \ldots < t_k$

- An variable *A* **disagrees** with fragment *f*, if for some $i<j<h$ we have $t_i(A)=t_h(A)$=1, but $\mathrm{t}_j(\mathrm{A})$=0.
- Otherwise t **agrees** with f:
- Then the column for *A* has the form

  0 0 …0 0 1 1 … 1 1 0 0 … 0 0

  for the observations in *f*

Example

A	1	0	0	1
B	1	1	1	0
C	0	0	0	1
D	1	0	1	0
E	1	0	1	1
F	1	1	1	1

a<b<c<d:	dis 1101	ag 0100	dis 0101	dis 1010
b<d<f<a:	ag 1111	dis 1010	ag 1110	ag 0011

What is a good fragment of order?

- A sequence f of rows, say, $u<v<w<t$
- Da(f): the number of variables disagreeing with the ordering
- Fr(f): the number of variables having at least 2 ones in the rows of f
- A good fragment has high Fr(f) and low Da(f)

Problem statement

- Given thresholds σ and γ
- Find all fragments of order f such that in the data

  $Fr(f) > \sigma$

  $Da(f) < \gamma$
- and all subfragments of f satisfy these
- and the fragment has smaller Da value than its peers
  - Any other fragments from the same set of objects

Algorithm

- How to find fragments with the specific properties?
- Start from fragments of length 2
  - No disagreements are possible
  - Only the bound $Fr(f)>\sigma$ needs to be tested

- Iteration:
  - Assume fragments of length k-1 are known
  - Then we can build candidate fragments of length k
  - Continue until no new patterns are found

- A complete algorithm: all fragments will be found

Monotonicity property

- Fragment $t_1 < t_2 < t_3 < \ldots < t_k$ can satisfy the requirements only if all subfragments of length k-1 satisfy them
- All these have to be in the collection of fragments of size k-1
- The levelwise algorithm

Algorithm

```
• Find F2, fragments of size 2
• C = all triples A<B<C such that A<B, A<C, and B<C are in F2
• k ß 3

• While C is not empty
      compute Da(f) for all f in C
      Fk ß {f in C | Fr(f)> $\sigma$ and Da(f)< $\gamma$ }
      k ß k+1
      C ß all fragments of length k such that all the subfragments of
            length k-1 are in Fk
```

Complexity of the algorithm

- Potentially exponential in the number of variables

- |F+C| = the size of the answer + all the candidates

- Proportional to

  $$|F+C|\ n\ m$$

  for a matrix with n rows and m columns

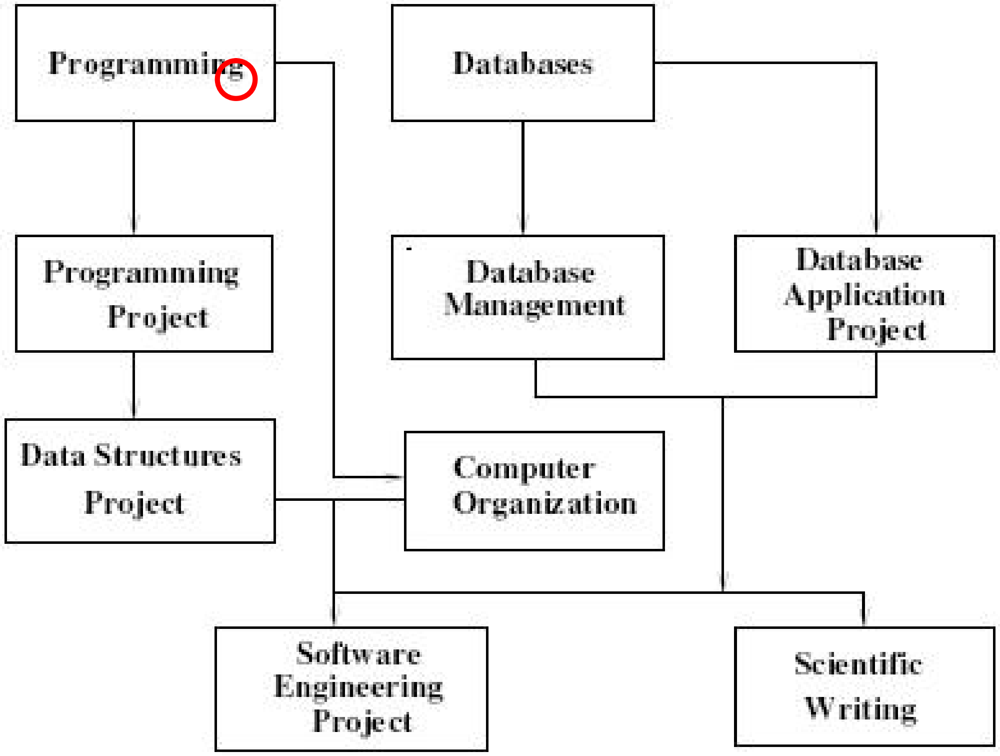
- Too low values of σ or too high values of γ will lead to huge outputs

Experimental results

- Data about students and courses
- Columns: students
- Rows: courses
- D(s,c)=1 if student s has taken course c
- Here we know the true ordering
  - Or actually two: official ordering
  - Real order in which the student took the courses

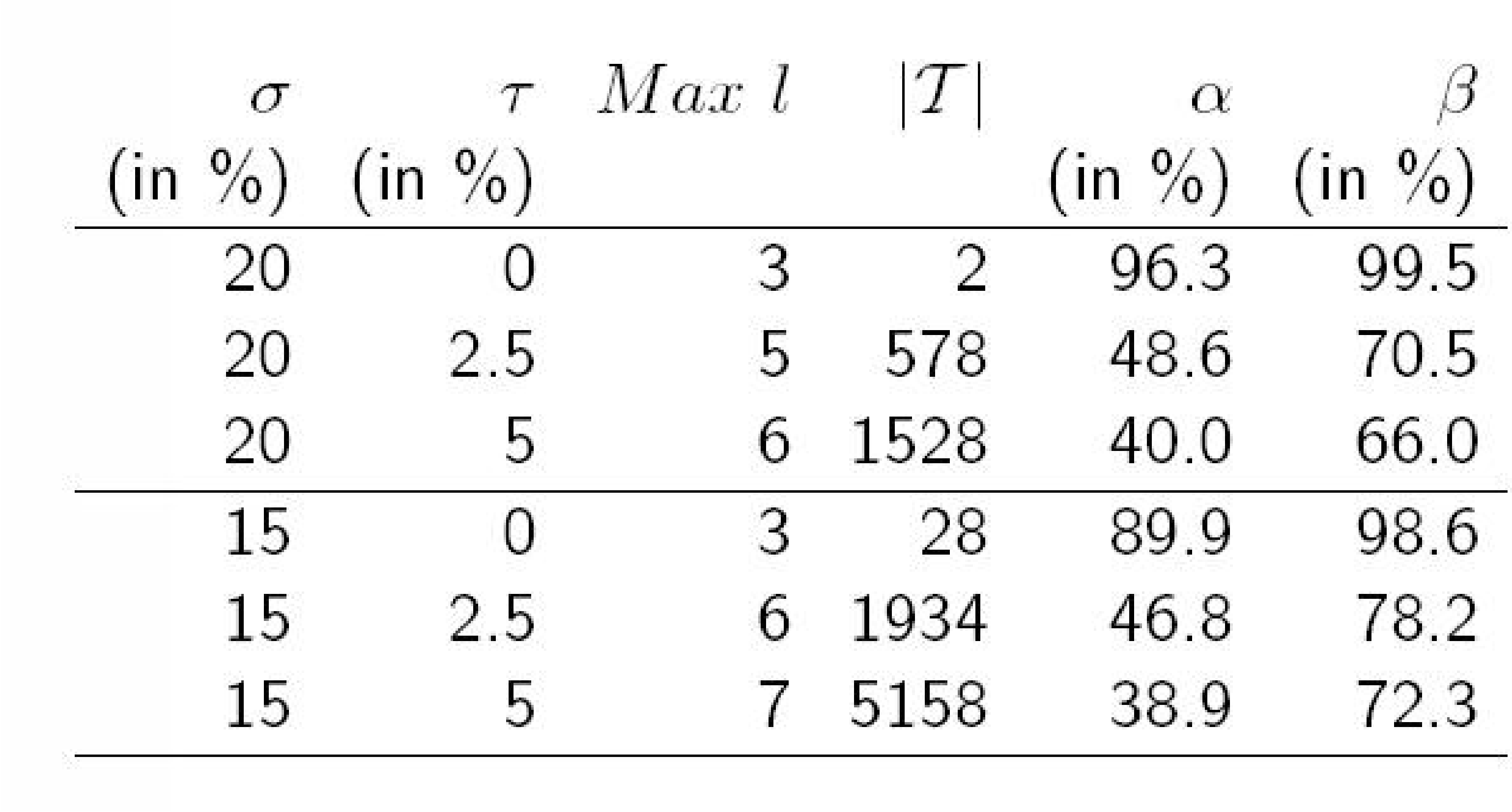
Part of the recommendations

Discovered fragment f
Fr(f)=1361, Da(f)=3.2%



⟨Programming,
Computer Organization,
Programming Project,
Data Structures Project,
Scientific Writing⟩

Results



σ	τ	$Max\ l$	$\lvert\mathcal{T}\rvert$	α	β
(in %)	(in %)			(in %)	(in %)
20	0	3	2	96.3	99.5
20	2.5	5	578	48.6	70.5
20	5	6	1528	40.0	66.0
15	0	3	28	89.9	98.6
15	2.5	6	1934	46.8	78.2
15	5	7	5158	38.9	72.3

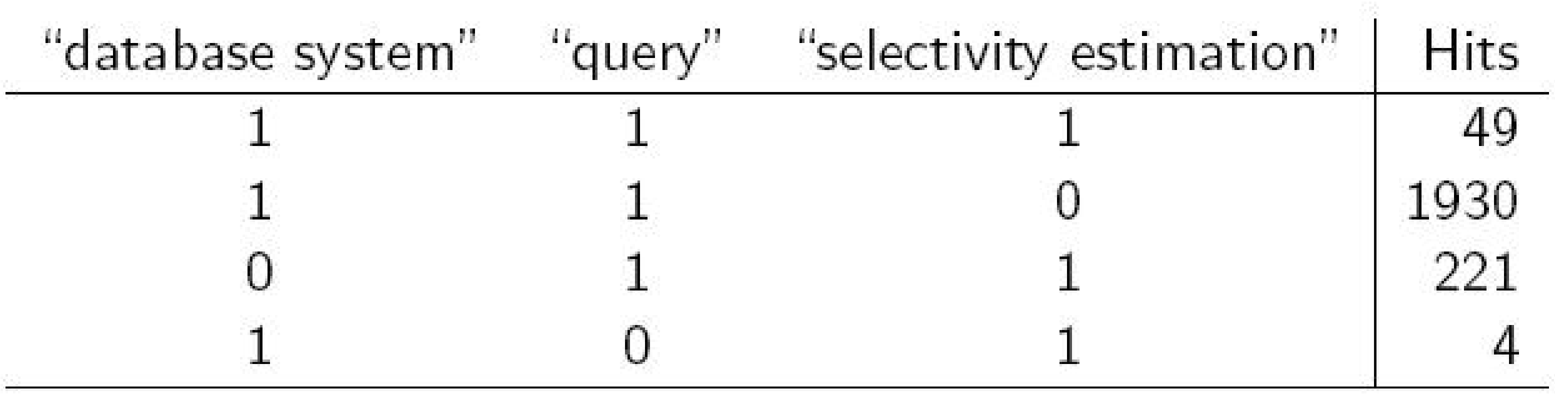
Results (paleontological data)

- Fragments for sites
- Or transpose the matrix: fragments for species
- Sequences of sites such that there are very few Lazarus events
- Provide ways of looking at projections of the data
- Can be used to find partial orders

Example: words in documents

- Represent collections of documents as term vectors
- Which words occur (1) in the document or not (0)
- Very large dimensionality, lots of observations

Example from Citeseer (in 2005)



“database system”	“query”	“selectivity estimation”	Hits
1	1	1	49
1	1	0	1930
0	1	1	221
1	0	1	4

What does this tell us about these terms? *Databases* and *selectivity estimation* together do not occur without *queries*

Databases < *queries* < *selectivity estimation*

Old (2005) example from Google Scholar

- prior distribution – MCMC
  151,000 documents

- prior distribution MCMC
  2950 documents

- – prior distribution MCMC
  1050 documents

- prior – distribution MCMC
  165 documents

prior < distribution < MCMC

Example from Google Scholar, Nov. 24, 2007

- prior distribution – MCMC
  2,220,000 documents

- prior distribution MCMC
  16,300 documents

- – prior distribution MCMC
  6,030 documents

- prior – distribution MCMC
  1,230 documents

prior < distribution < MCMC

An aside: have the ratios of the frequencies changed?

Query	2005	2007	Ratio
p d m	2950	16300	5.5
p d –m	151000	2220000	14.7
–p d m	1050	6030	5.7
p –d m	165	1230	7.5

Next theme

- Find small total orders from 0-1 data
- Finding partial orders from 0-1 data
- Find total orders for 0-1 data

Finding partial orders from 0-1 data

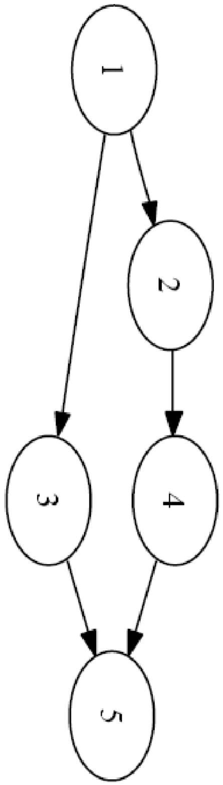
- Model: a partial order over all observations
- Loglikelihood: proportional to the number of cases the observed occurrence patterns violate the continuity of species
- Prior: prefer partial orders that are as specific as possible
- Task: find a model with high likelihood * prior
- Algorithm: Find fragments and use heuristic search to build a good partial order

Why partial orders?

- Determining the ages of sites is difficult
- Radioisotope methods apply only to few sites
- In paleontology the so-called MN system: 18 classes for the last 25 Ma
- Classes are assigned by ad hoc methods

- Searching for a total order might not be a good idea
- The MN system is a partial order

Finding partial orders from data

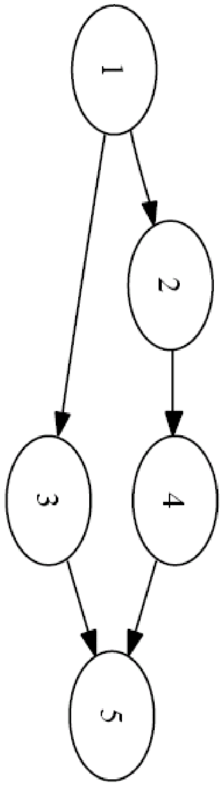


- How to find a partial order that fits well with the data?
- What does this mean?

What is a good partial order?

- The Lazarus count of a species with respect to a partial order P:
  - For how many sites the species was extinct at the site, but extant before and after it (as determined by P)
  - The same definition as for total orders
- A good partial order has small Lazarus count

- Can be formulated as a likelihood (a Lazarus event is a false positive)



1	1	0	0
2	1	1	0
3	0	0	1
4	1	1	1
5	1	1	1
	Laz	No Laz	No Laz

What is a good partial order?

- Find a partial order that has a low Lazarus count
- The trivial partial order has Lazarus count 0
- Want to find a partial order that is specific (close to a total order) and agrees with the data
- Measures of specificity:
  - the number of linear extensions of P (hard to compute)
  - number of edges in P
- Find a partial order that has high

  specificity * likelihood

Algorithm for finding partial orders

- Compute fragments from the unordered data
- E.g., $a < d < b < e < f$ and $b < e < c$ and $b < a < c < f$ and …
- Form a precedence matrix: in what fraction of the fragments does a precede b
- Form a partial order that approximates the precedence matrix (heuristic search)

Fragments and reverse fragments

- The fragment generation will produce for each fragment f also its reverse f^R

- The pairwise precedence matrix would be useless

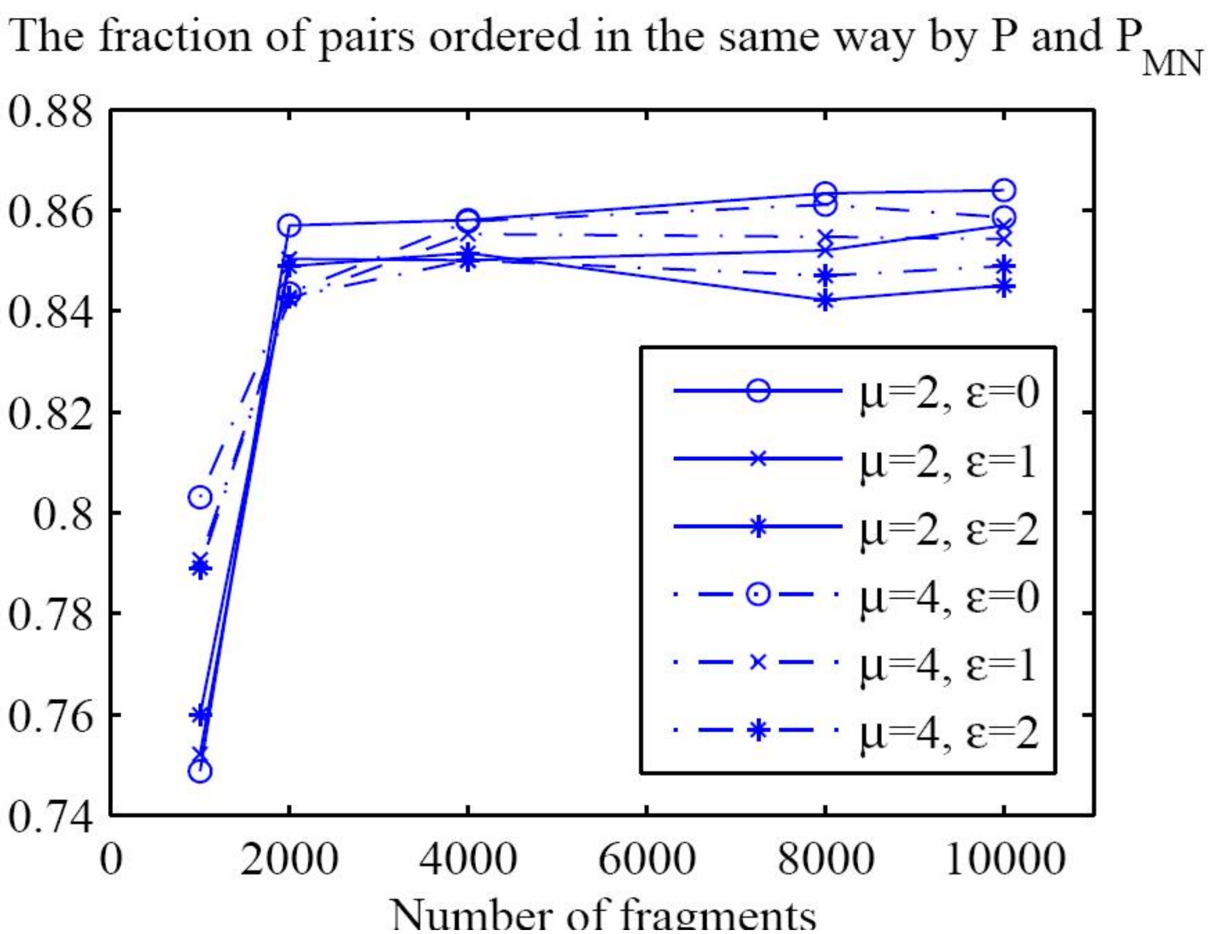
- Divide the fragments into two classes (graph cutting)
- Discard one class
- Build the precedence matrix

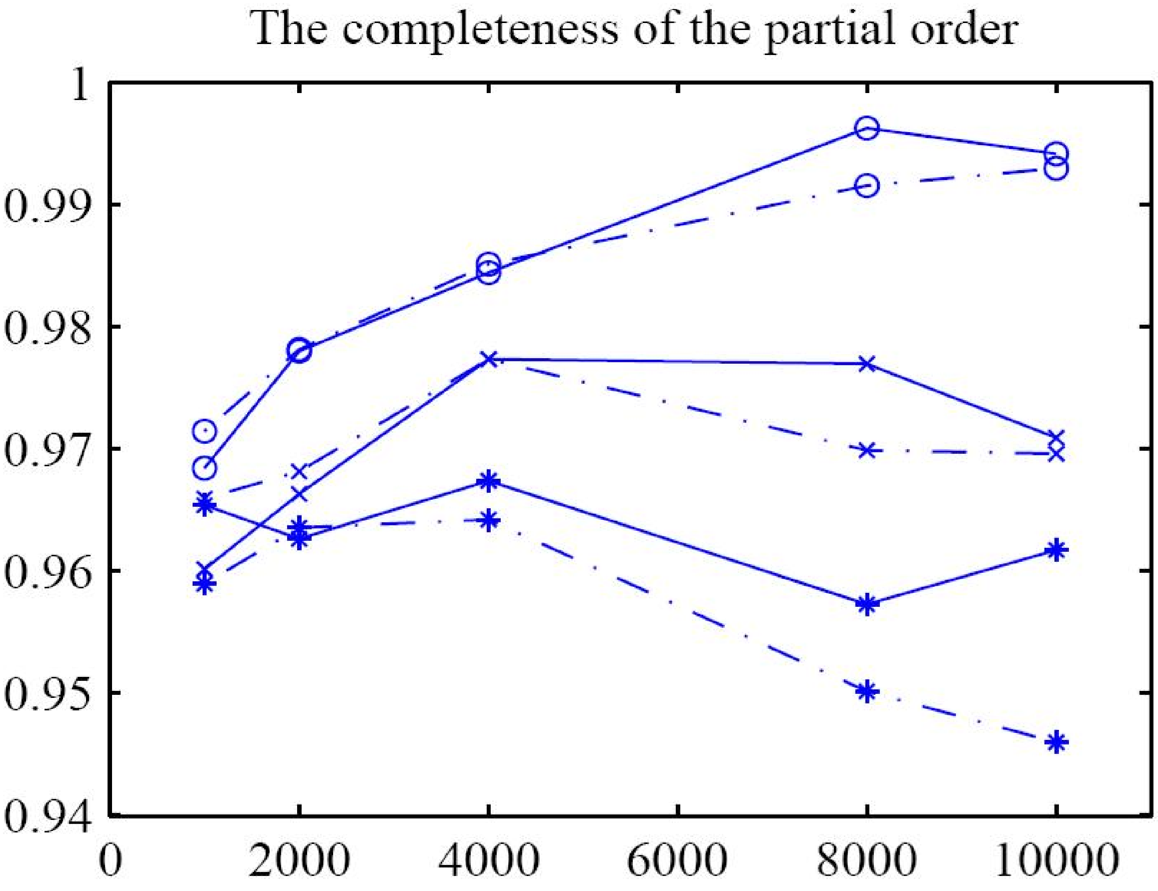
From precedence matrix to partial order

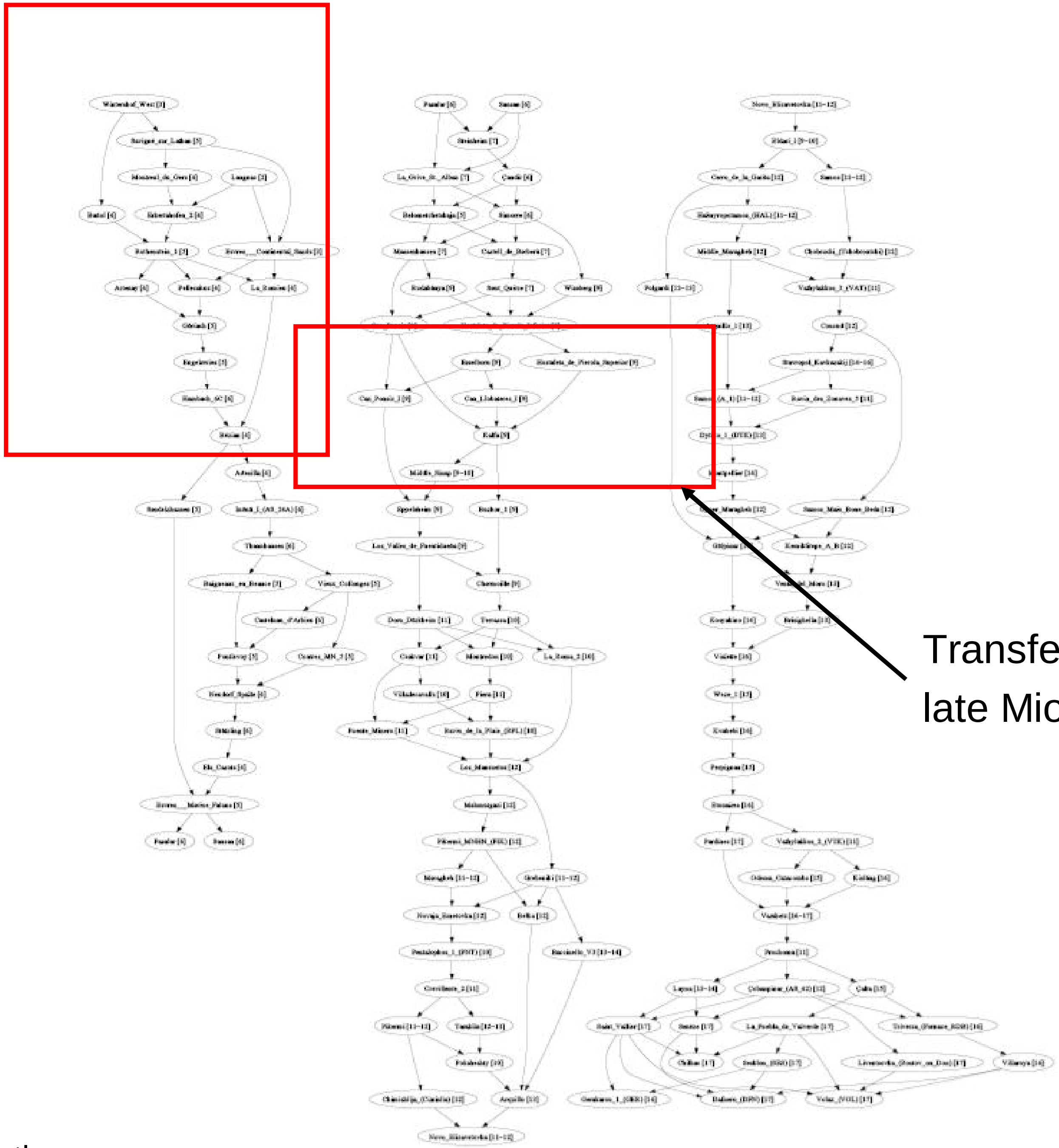
- Heuristic search
- Add edges to the partial order so that the match with the precedence matrix improves
- Keep track of transitivity

- Difficult (and interesting) algorithmic problem
- Empirical results look good

- Very recent theoretical results

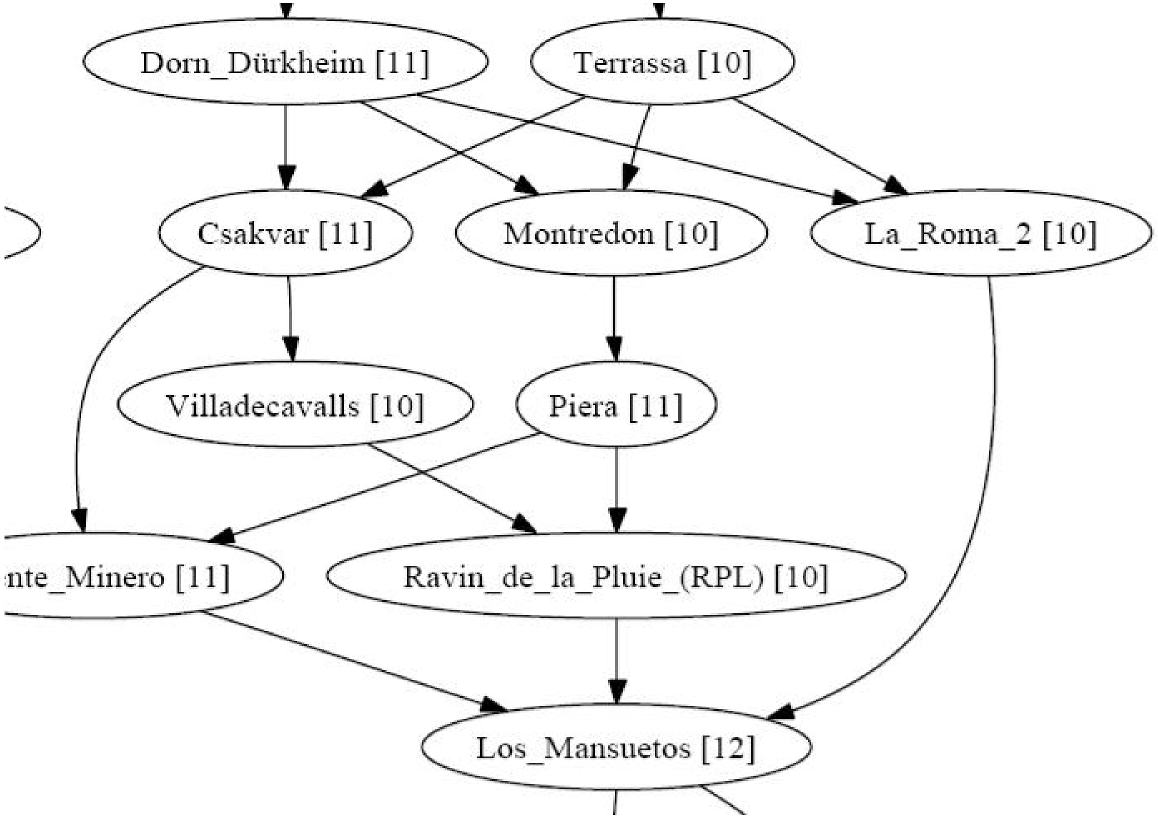

The fraction of pairs ordered in the same way by P and P_{MN}
0.88
0.86
0.84
0.82
0.8
0.78
0.76
0.74
0
2000
4000
6000
8000
10000
Number of fragments
μ=2, ε=0
μ=2, ε=1
μ=2, ε=2
μ=4, ε=0
μ=4, ε=1
μ=4, ε=2


The completeness of the partial order
1
0.99
0.98
0.97
0.96
0.95
0.94
0
2000
4000
6000
8000
10000



Early Miocene

Transfer to late Miocene



approximately 6–7 MN classes of sites will be re-evaluated on the basis of the partial order.

Themes of the talk

- Find small total orders from 0-1 data
- Finding partial orders from 0-1 data
- Find total orders for 0-1 data

Finding good total orders for a matrix

- Given a site-genus matrix
- What is a good total ordering for the rows?
- One in which there are as few Lazarus events as possible
- Model class: total orders
- Loglikelihood proportional to the number of Lazarus events

How to find such an ordering of the rows?

- If there is an ordering that has no Lazarus events, it can be found in linear time (Booth & Lueker)
  - consecutive ones property

- But normally there are (lots of) Lazarus events

Finding good total orders for a matrix

- The problem of finding the best ordering of the matrix is NP-hard

- Finding whether there is a submatrix of size k that has no Lazarus events is NP-hard
- The fragment method finds such submatrices

- Local search, traveling salesperson approaches

- Spectral methods

Spectral ordering for finding good total orders for a matrix

- Spectral ordering
- Compute a similarity measure *s(i,j)* between sites (e.g., dot product)
- Laplacian L(i,j)

$$L(i,j) = \begin{cases} -s(i,j), & i \neq j \\ \sum_k s(i,k), & i = j \end{cases}$$

- The eigenvector *v* corresponding to the second smallest eigenvalue of L satisfies

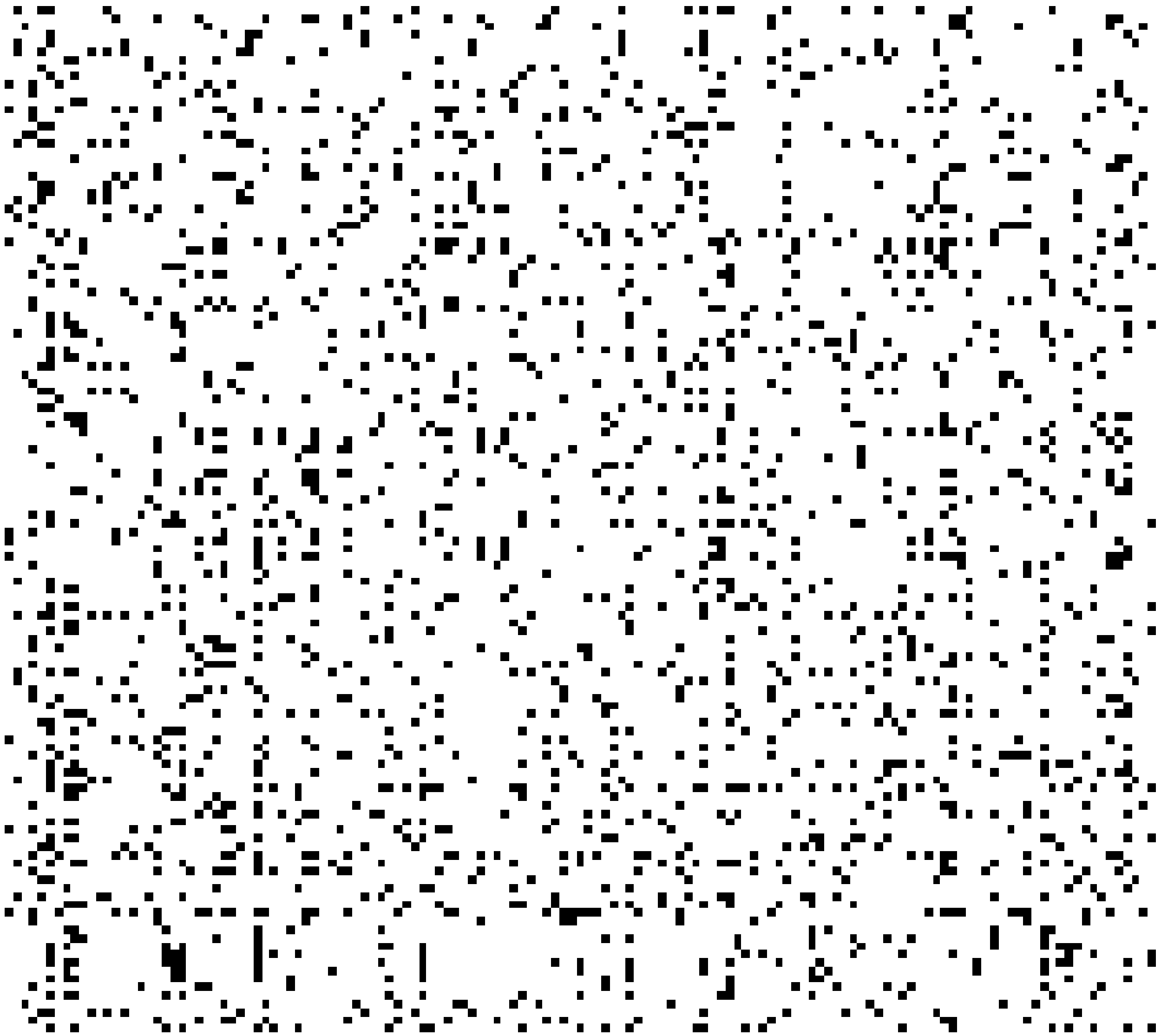
$$\sum_i v_i = 0,\ \ \sum_i v_i^2 = 1,\ \text{and}\ \sum_i s(i,j)(v_i - v_j)^2 = 1\ \text{is minimized.}$$

- Maps the points to 1-d, keeping similar points close to each other
- The values v_i can be used to order the points

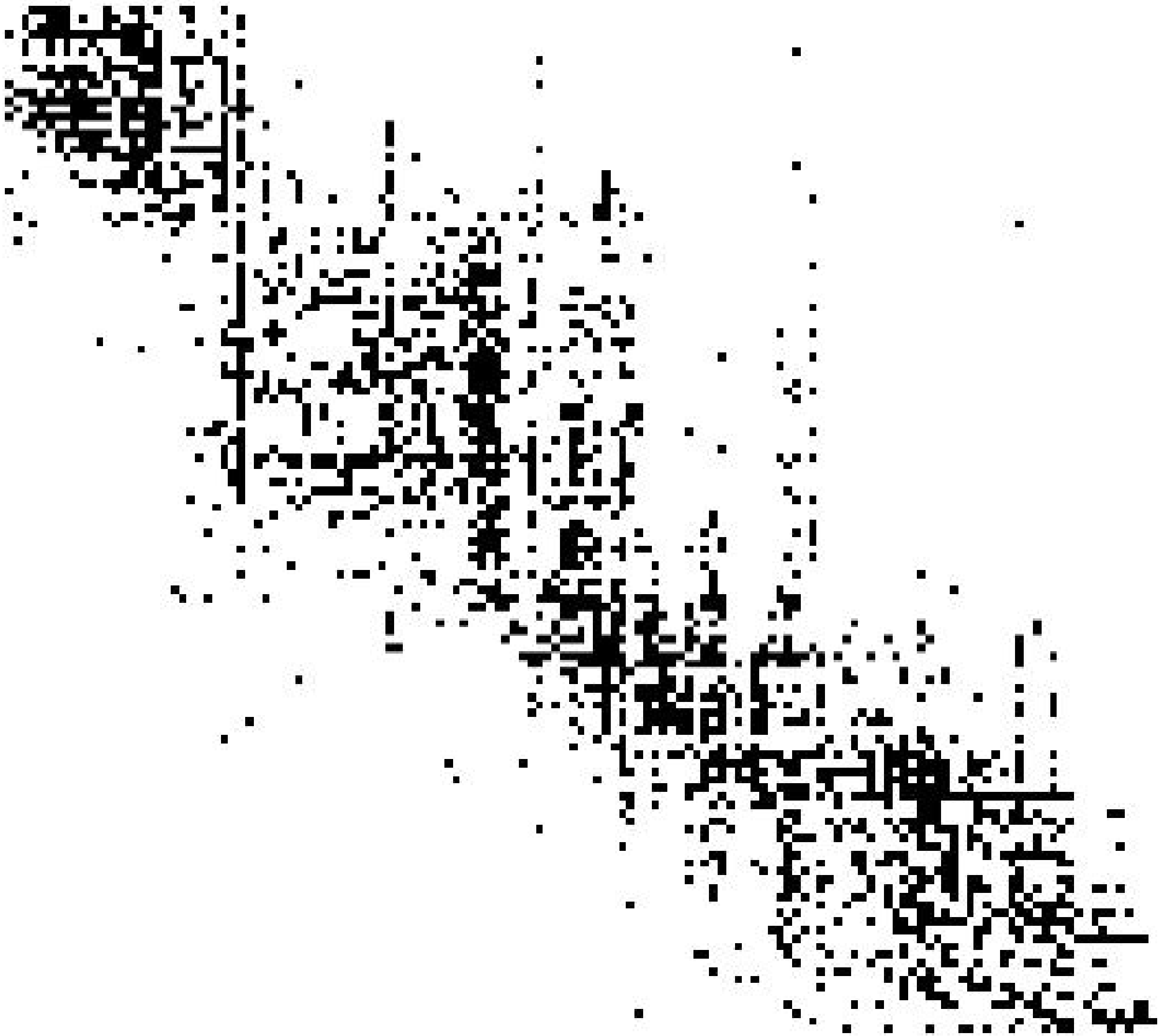
Empirical observation

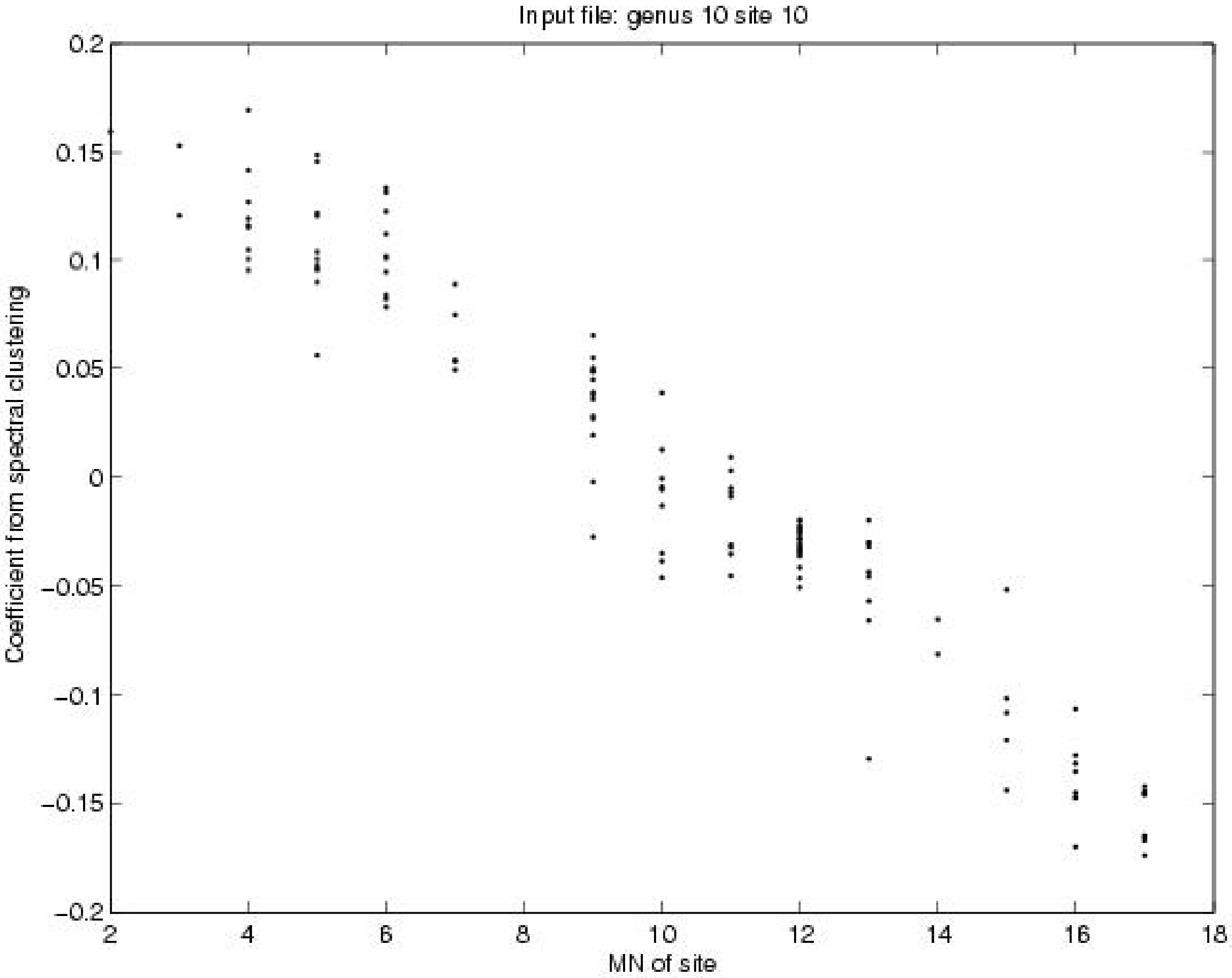
- The eigenvector seems to minimize also Lazarus events
- Even better than some combinatorial algorithms
- Why?
- No really good intuitive theoretical understanding
  - Related to mixing time of Markov chains etc.

Site-genus -matrix

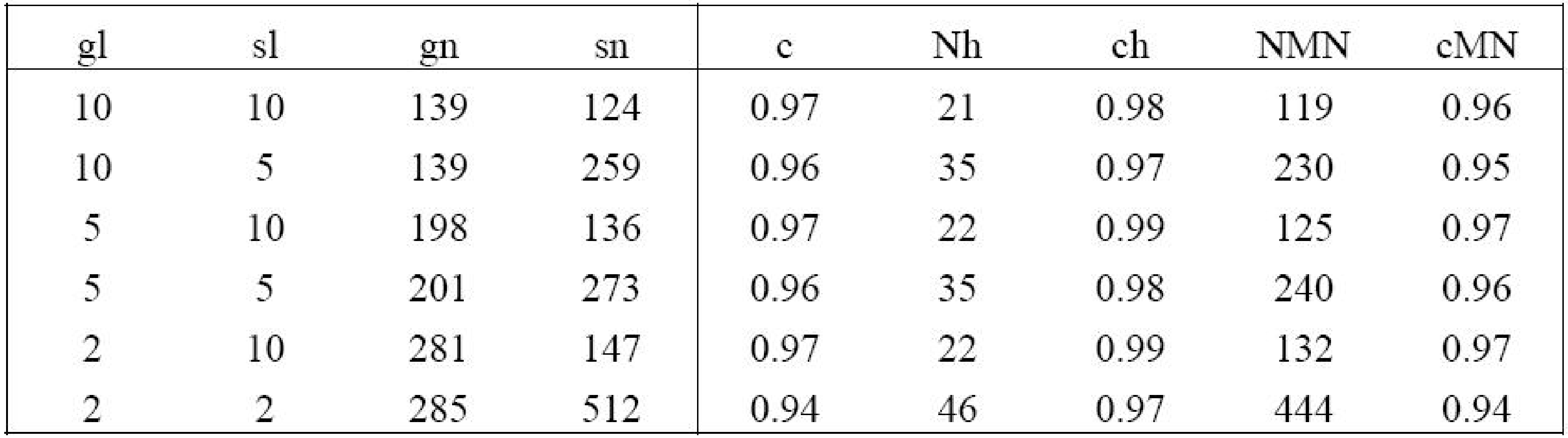


After spectral ordering

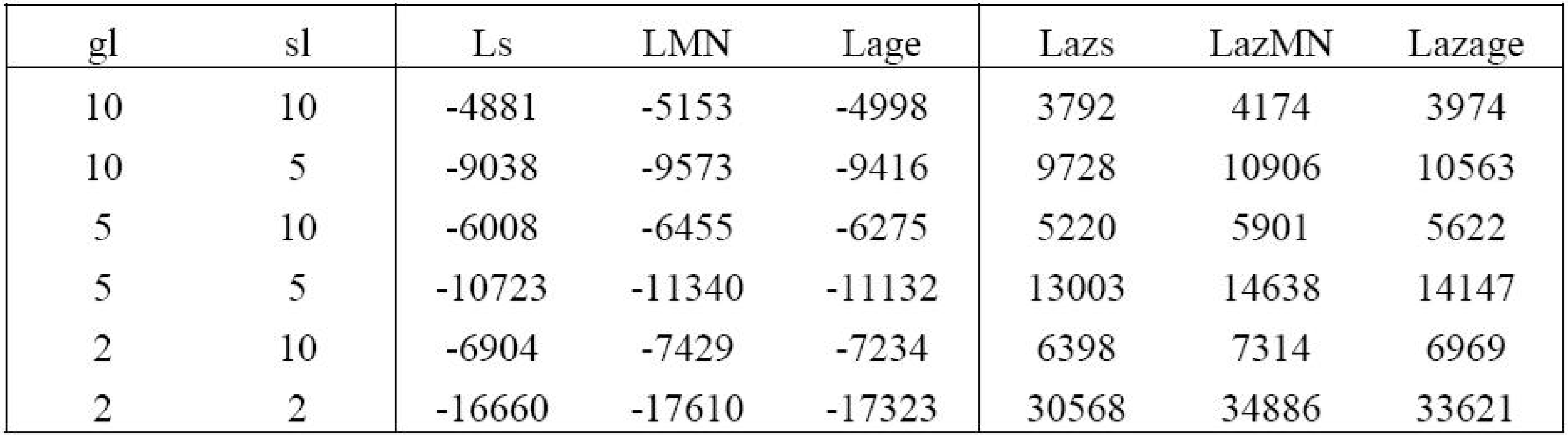




Fortelius, Jernvall, Gionis, Mannila, Paleobiology 32 (2006)



gl	sl	gn	sn	c	Nh	ch	NMN	cMN
10	10	139	124	0.97	21	0.98	119	0.96
10	5	139	259	0.96	35	0.97	230	0.95
5	10	198	136	0.97	22	0.99	125	0.97
5	5	201	273	0.96	35	0.98	240	0.96
2	10	281	147	0.97	22	0.99	132	0.97
2	2	285	512	0.94	46	0.97	444	0.94



gl	sl	Ls	LMN	Lage	Lazs	LazMN	Lazage
10	10	-4881	-5153	-4998	3792	4174	3974
10	5	-9038	-9573	-9416	9728	10906	10563
5	10	-6008	-6455	-6275	5220	5901	5622
5	5	-10723	-11340	-11132	13003	14638	14147
2	10	-6904	-7429	-7234	6398	7314	6969
2	2	-16660	-17610	-17323	30568	34886	33621

Questions

- Computational
  - Why does it work so well?
  - How well does it actually work (what is the smallest number of Lazarus events for this data?)
  - How to interpret the coefficients?

- Paleontological
  - Fully based on the occurrence matrix (excellent and bad)
  - Site-species data is only one type of data; how to use other types of data for the ordering?
  - …

Rough estimates of the sizes of the model classes

- N observations
- Fragments of size at most k
  - N^k individual fragments
  - 2^{N^k} sets of fragments
- Partial orders $2^{O(N^2)}$
- Total orders $N!$

Concluding remarks

- General task: finding order from unordered data
- Here using species continuity as the additional information
- Other applications are possible

- Model classes
  - Fragments
  - Partial orders
  - Total orders

Lots of open questions

- The unreasonable effectiveness of spectral methods on discrete optimization task
- Approximation guarantees
- Fragments from other applications
- MDL description of sequences via partial orders
- Etc.

References

- A. Gionis, T. Kujala and H. Mannila: Fragments of order. *ACM SIGKDD 2003*, p. 129-136.
- A. Ukkonen, M. Fortelius, H. Mannila: Finding partial orders from unordered 0-1 data. *ACM SIGKDD 2005*, p. 285-293.
- M. Fortelius, A. Gionis, J. Jernvall, H. Mannila, Spectral Ordering and Biochronology of European Fossil Mammals. *Paleobiology* 32, 2, 206-214 (2006).
- K. Puolamäki, M. Fortelius, H. Mannila: Seriation in Paleontological Data Using Markov Chain Monte Carlo Methods. *PLoS Comput Biol* 2(2): e6
- A. Gionis, H. Mannila, K. Puolamaki, and A. Ukkonen, Algorithms for Discovering Bucket Orders from Data, 12th International Conference on Knowledge Discovery and Data Mining (KDD) 2006, p. 561-566.