Constraint-Based Data Mining and an Application in Molecular Feature Mining

Luc De Raedt

Chair of Machine Learning and Natural Language Processing

Albert-Ludwigs-University Freiburg

Joint work with

Lee Sau Dan, Christoph Helma, Manfred Jaeger, Stefan Kramer, Heikki Mannila

Three Parts

- Introduction to Inductive Databases
- Inductive Database Systems
  - MolFea : Mining features in Molecules
- Constraint Based Mining
  - Integrate data mining with databases
  - Querying for patterns using constraints

Inductive databases

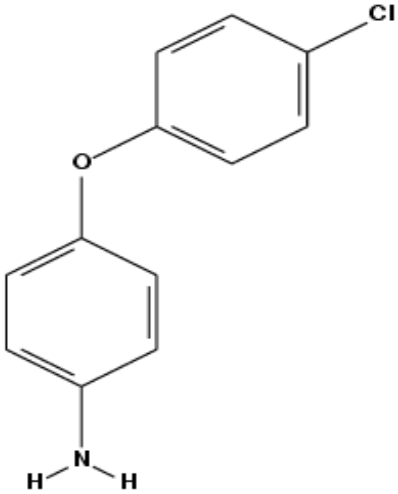
- Data mining
  - search for interesting and understandable patterns in data
- State-of-the-art in data mining ~ databases in the early days
- A theory of data mining is lacking
- View by Iemielinski and Mannila (CACM 96)
  - Make first class citizens out of patterns
  - Query not only the data but also the patterns
  - Tightly integrate data mining and databases

Inductive querying

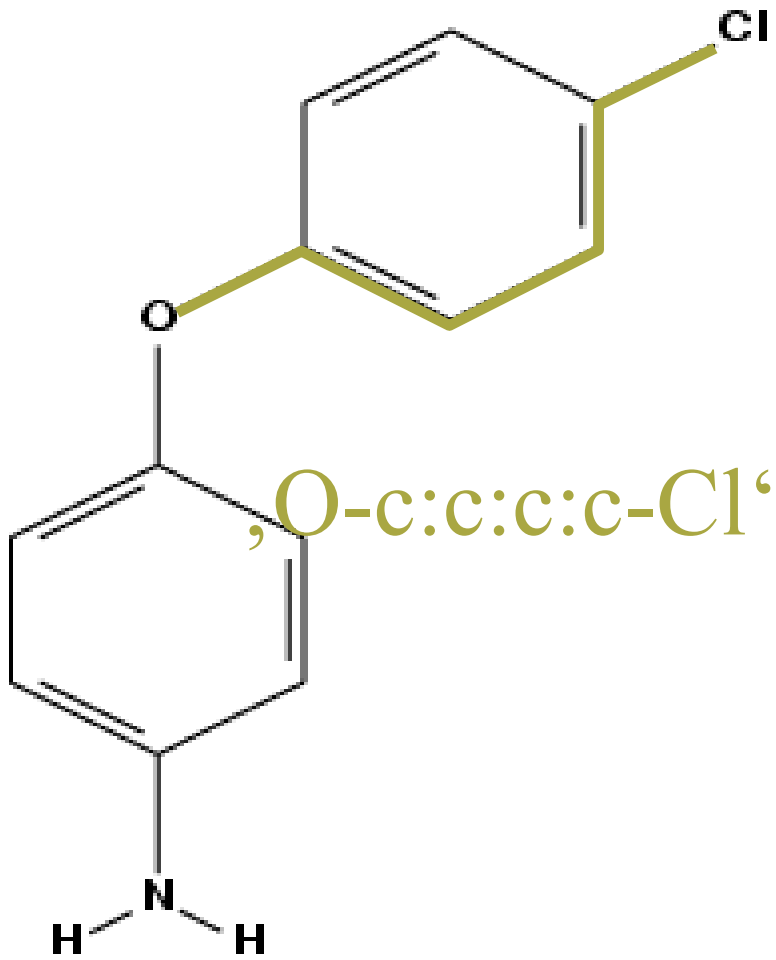
- The need to actively mine / analyze scientific databases in biology, chemistry
  - “Understandable” patterns needed
  - Scientist wants control of mining process
  - Constraint based mining
    - Constraints specify patterns of interest
    - E.g. *find all patterns that occur in at least 30 % the actives and at most 3% of the inactives and contain a benzene ring*
  - Mining becomes a querying process
    - *«There is no such thing as real discovery, just a matter of the expressive power of the query languages»* Iemielinski & Mannila, CACM 96

Molecular Feature Mining

- What ?
  - Find fragments (substructures) of interest in sets of molecules
- Why ?
  - Discover new knowledge
  - Use in predictive models
    - SAR (Structure Activity Relationship)



Molecules and Fragments



‚O-c:c:c:c-Cl‘

- 2D-structure
  - essentially Graphs
- Fragments
  - substructres
  - We : *linear* fragments
  - Sequence of atoms and bonds
- Linear fragments
  - ‚o‘, ‚c‘, ‚cl‘, ‚n‘, ‘s‘,... denote elements
  - ‚-‘ ... single bond
‚=, ... double bond
‚#‘ ... triple bond
‚:‘ ... aromatic bond
  - (hydrogens implicit)
- Smarts encoding

Constraint-based Data Mining

- What ?
  - Use constraints to specify which fragments/patterns are interesting
  - E.g. Frequency and syntax
- Why ?
  - Declarative Querying
  - Interactive Process
  - Inductive database idea

Constraint-based data mining

- Generality
  - One fragment *is more general* than another one if it is a substructure of the other one
  - Notation : *g ≤ s (g is more general than s; i.e. g will match a graph/string whenever s does)*
  - Graphs : ~ subgraph relationship
  - Strings : substring / subsequence relationship
    - E.g. *aabbcc is more general than ddaabbccee (substring)*
    - E.g. *abc is more general than aabbcc (subsequence)*
  - (Item)sets : subset relation, e.g. {a,b} subset {a,b,c}

Search Space for Strings

$$\varepsilon$$

$$a \quad b$$

$$aa \quad ab \quad ba \quad bb$$

$$aaa \quad aab \quad aba \quad baa \quad abb \quad bab \quad bba \quad bbb$$

$$\ldots$$

Every string has max two fathers

Observe that Σ^* is not a lattice !

mgg can contain more than element

mgs may be infinite

Primitives

- *Generality MolFea Symmetry !*
  - g is *equivalent* to s (*syntactic variants*) only when they are a reversal of one another
    - *E.g. ,C-O-S'* and *,S-O-C'* denote the same substructure
  - g *is more general than* *s* if and only if *g* is a subsequence of *s* or *g* is a subsequence of the reversal of *s*
    - *E.g. ,Cl-O-S'* ≤ *,Cl-O-S-c:c:c'*
    - *E.g., ,O-Cl'* ≤ *,Cl-O-S'*
- Frequency of a fragment f on a data set D
  - The percentage of data points in D that f occurs in
  - E.g let f be aa and let D={abaa,acc, caa}; freq(f,D) = .66=2/3

Primitive Constraints

- $f \leq P$, $P \leq f$, *not* $(f \leq P)$ and *not* $(P \leq f)$:
  f ... unknown target fragment,
  P ... a specific fragment
  e.g. *abbaa* $\leq f$
- *freq(f, D)*
  relative frequency of a fragment f on a data set D
- $freq(f, D1) \geq t$, $freq(f, D2) \leq t$,
  t ... positive real number between 0 and 1
  $D1$, $D2$ ... Data sets
  e.g. $freq(f, Pos) \geq 0.20$

Example query

- Let E1 = {aabbcc,abbc,bb}
- Let E2 = {abc,bc,cc}
- freq(f,E1) $\geq$ 2 and freq(f,E2) = 0 and “a “ < f
- Solutions : abb and abbc

Example Queries

- *(`N-O'≤ f) ∧*
  (freq(f, Act) ≥ 0.1) ∧
  (freq(f, Inact) ≤ 0.01)

- *not(,F' ≤ f) ∧ not (,Cl' ≤ f) ∧*
  not (,Br' ≤ f) ∧ not (,I' ≤ f) ∧
  (freq(f, Act) ≥ 0.05) ∧
  (freq(f, Inact) ≤ 0.02)

- *Queries are conjunctions of primitive constraints*

Representing Solutions

- Traditional min. frequency constraint
  - *Let c be freq(f, Act) ≥ x*
  - *c satisfies <u>Anti Monotonicity</u> property*
    - *If we have a fragment g ≤ s,*
    - *Then if s is a solution then g is a solution as well*
  - *Imposes a lower border S=max(Sol) on the space of solutions*

A String Example

freq(f,D) ≥ 2 where D= $ABCD \quad BDEF$

$ABDF \quad ABCF$

ε

$A \quad B \quad C \quad D \quad F$

$AB \quad BC \quad BD$

ABC

Consider E

E is not frequent,

Therefore no string containing E is frequent

Consider ABC

ABC is frequent

Therefore all substrings of ABC are frequent

Characterized by $S = \{ABC, BD, F\} = \max(Sol)$

Another String Example

Let $f \leq$ *ABD*

$$\varepsilon$$

$$A \quad B \quad D$$

$$AB \quad BD$$

$$ABD$$

Characterized by $S = \{ABD\} = \max(Sol)$

Representing Solutions

- Traditional max frequency constraint
  - *Let c be freq(f, Act) < x*
  - *c satisfies <u>Monotonicity</u> property*
    - *If we have a fragment g ≤ s,*
    - *Then if g is a solution then s is a solution as well*
  - *Imposes an upper border G=min(Sol) on the space of solutions*

A String Example

$ABCD \quad BDEF$

$ABDF \quad ABCF$

Consider *"B"* $\leq$ *f and freq(f,D)* $\geq$ 2 with D=

ε

$A \quad B \quad C \quad D \quad F$

$AB \quad BC \quad BD$

ABC

B

$AB \; BC$

ABC

Characterized by $S = \{ABC\}$

Characterized by $S = \{ABC, BD, F\}$

and $G = \{B\}$

Constraints

Anti-monotonic

$$freq(f, D) \geq x$$

$$f \leq P$$

$$not(P \leq f)$$

Monotonic

$$freq(f, D) \leq x$$

$$f \geq P$$

$$not(P \geq f)$$

In ML

$$f \leq P$$

$$\sim$$

P is a positive example

In ML

$$not(f \leq P)$$

$$\sim$$

P is a negative example

Mitchell's Version Space

- Consider now a conjunctive query

$$a_1 \wedge \ldots \wedge a_n \wedge m_1 \wedge \ldots \wedge m_k$$

$$c_1 = freq(f, D) \geq x$$

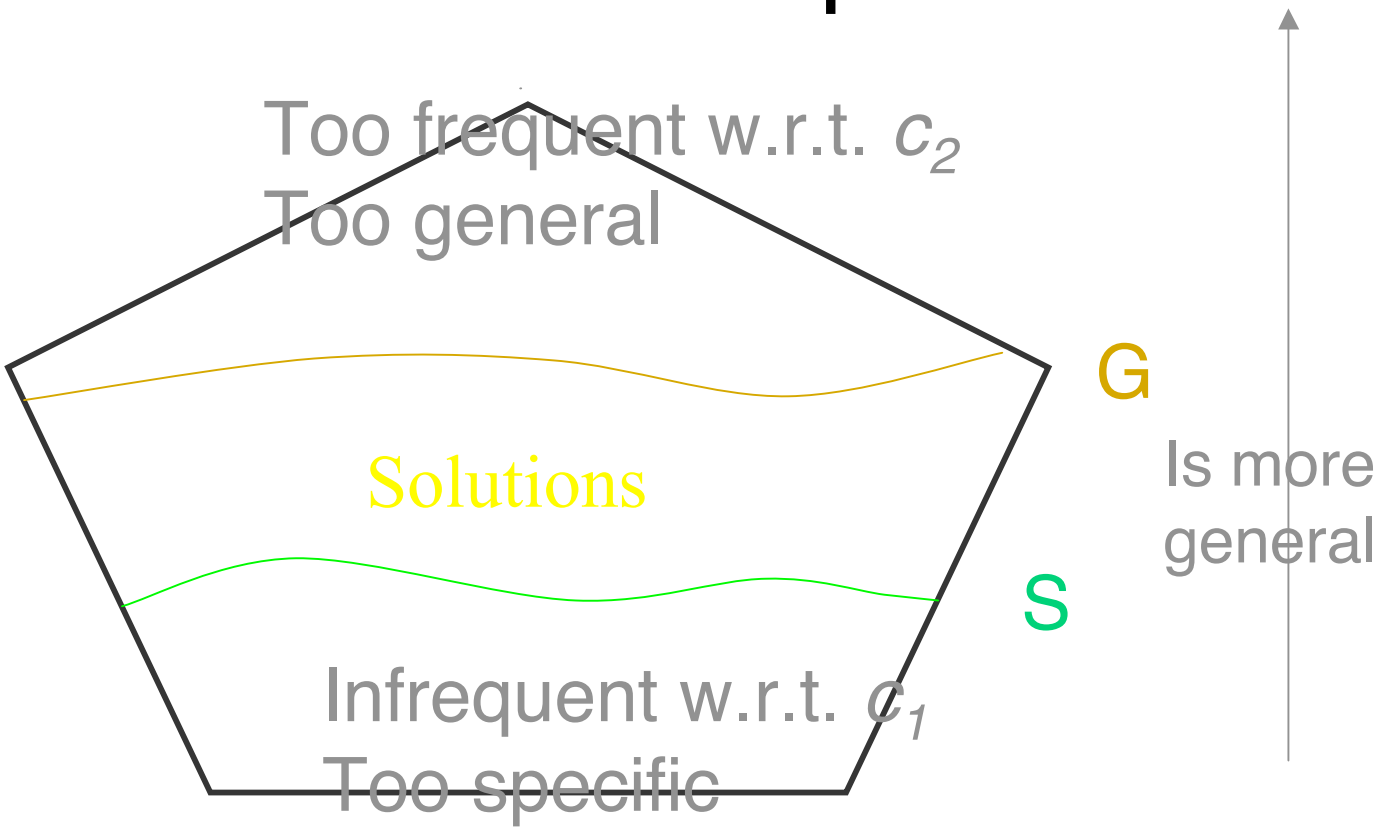
$$c_2 = freq(f, E) \leq y$$

- We want to compute

$$sol(a_1 \wedge \ldots \wedge a_n \wedge m_1 \wedge \ldots \wedge m_k) = \{f \mid \exists s \in S, g \in G : g \leq f \leq s\}$$

where S and G are defined w.r.t. $a_1 \wedge \ldots \wedge a_n \wedge m_1 \wedge \ldots \wedge m_k$

Mitchell's Version Spaces



Some problems

- There exist conjunctive queries *q* such that *Sol(q)* is not boundary set representable; these queries are not safe
  - Boundary sets may be infinite
  - Or may not be complete

Consider $\neg(\mathrm{a} \leq f)$ and let $\Sigma = \{a, b\}$

Then $S(\neg(\mathrm{a} \leq f)) = \{\}$

Consider $(\mathrm{a} \leq f) \wedge (\mathrm{b} \leq f)$ and let $\Sigma = \{a, b, c\}$

Then $G = \{ab, ba, acb, bca, accb, bcca, \ldots\}$

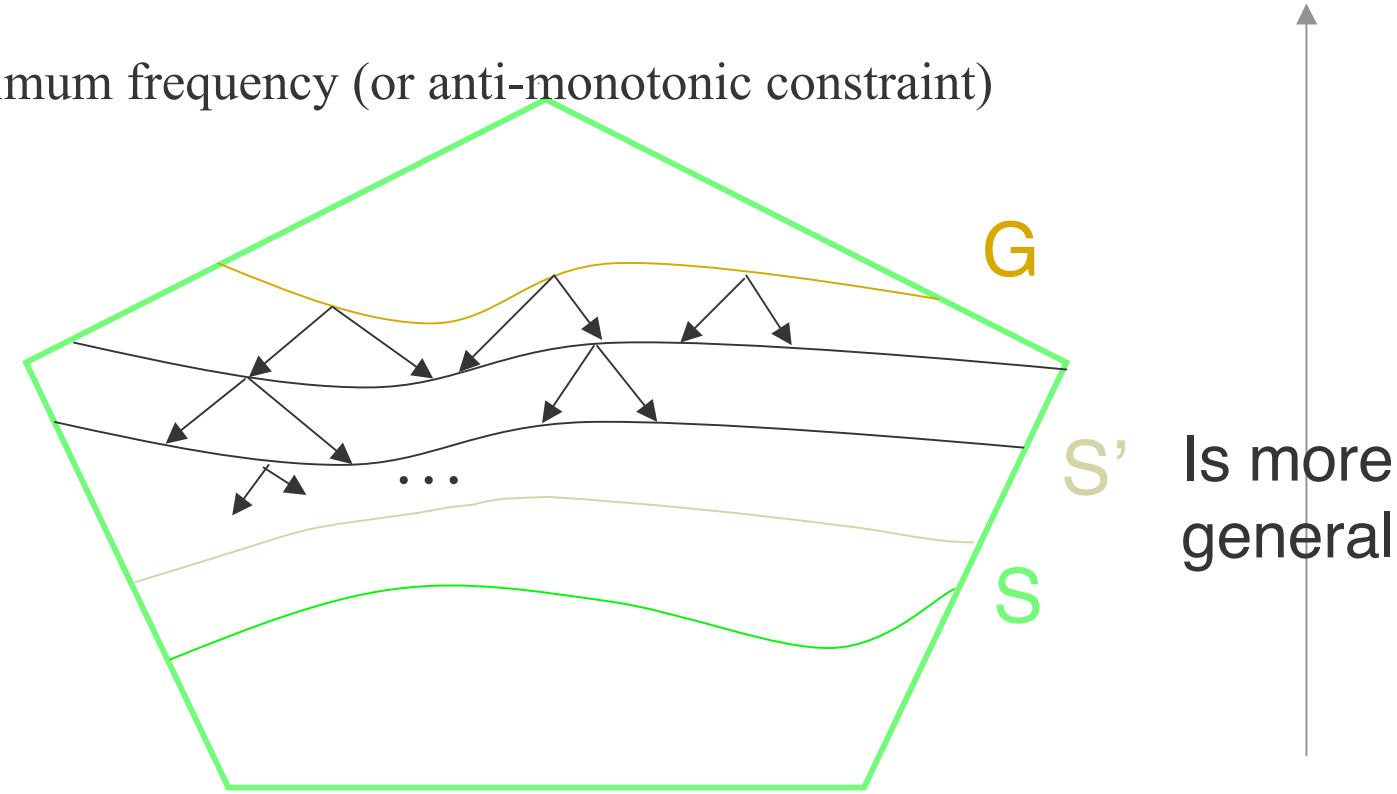
Computing Borders

- Borders completely characterize the set of solutions for *safe* queries
- If solution set is finite, then query is safe
- Combination of well-known algorithms to compute border wrt
  - Level wise algorithm by Agrawal et al., Mannila and Toivonen
  - Mitchell's and Mellish's version space algorithms
  - In our *level wise version space* algorithm

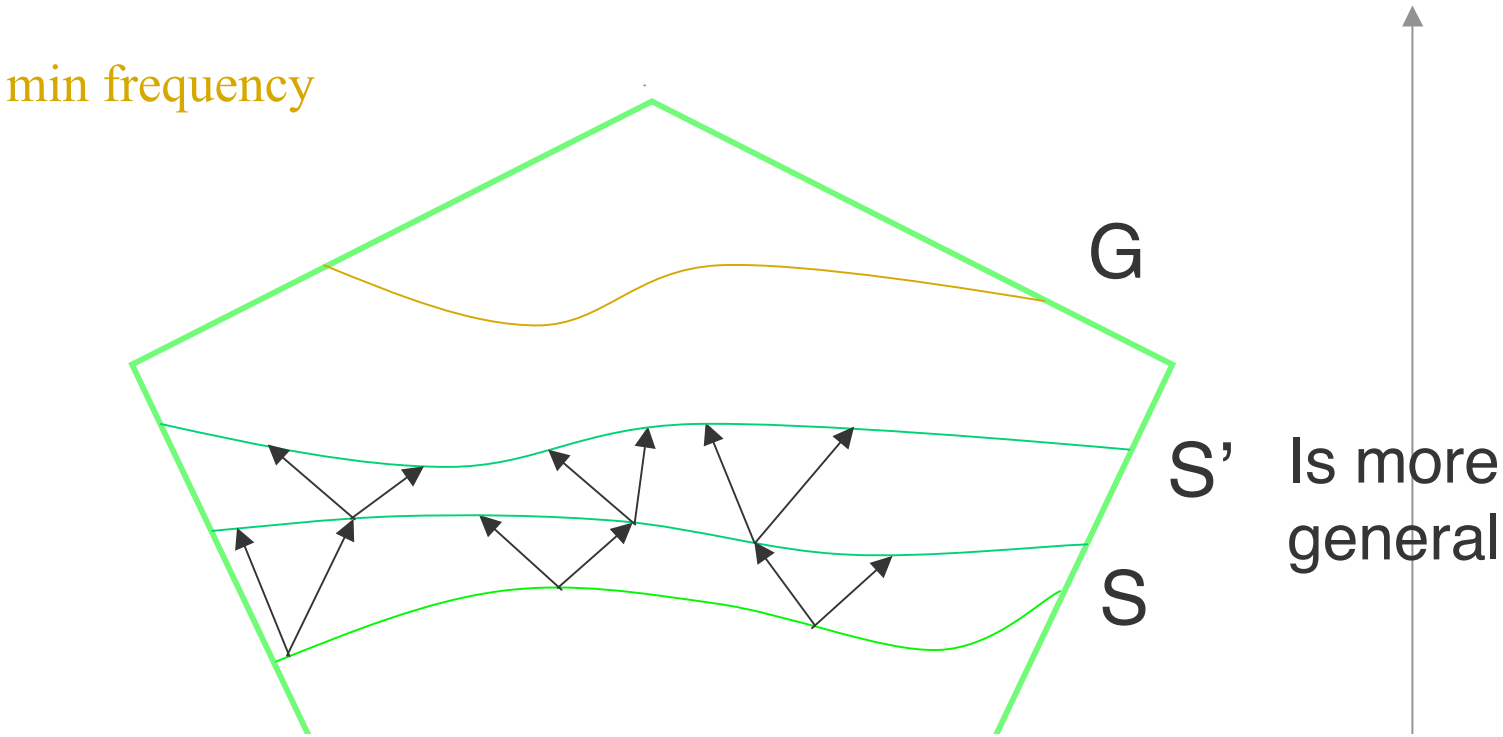
$$a_1 \wedge \ldots \wedge a_n \wedge m_1 \wedge \ldots \wedge m_k$$

Levelwise Version Spaces

Minimum frequency (or anti-monotonic constraint)

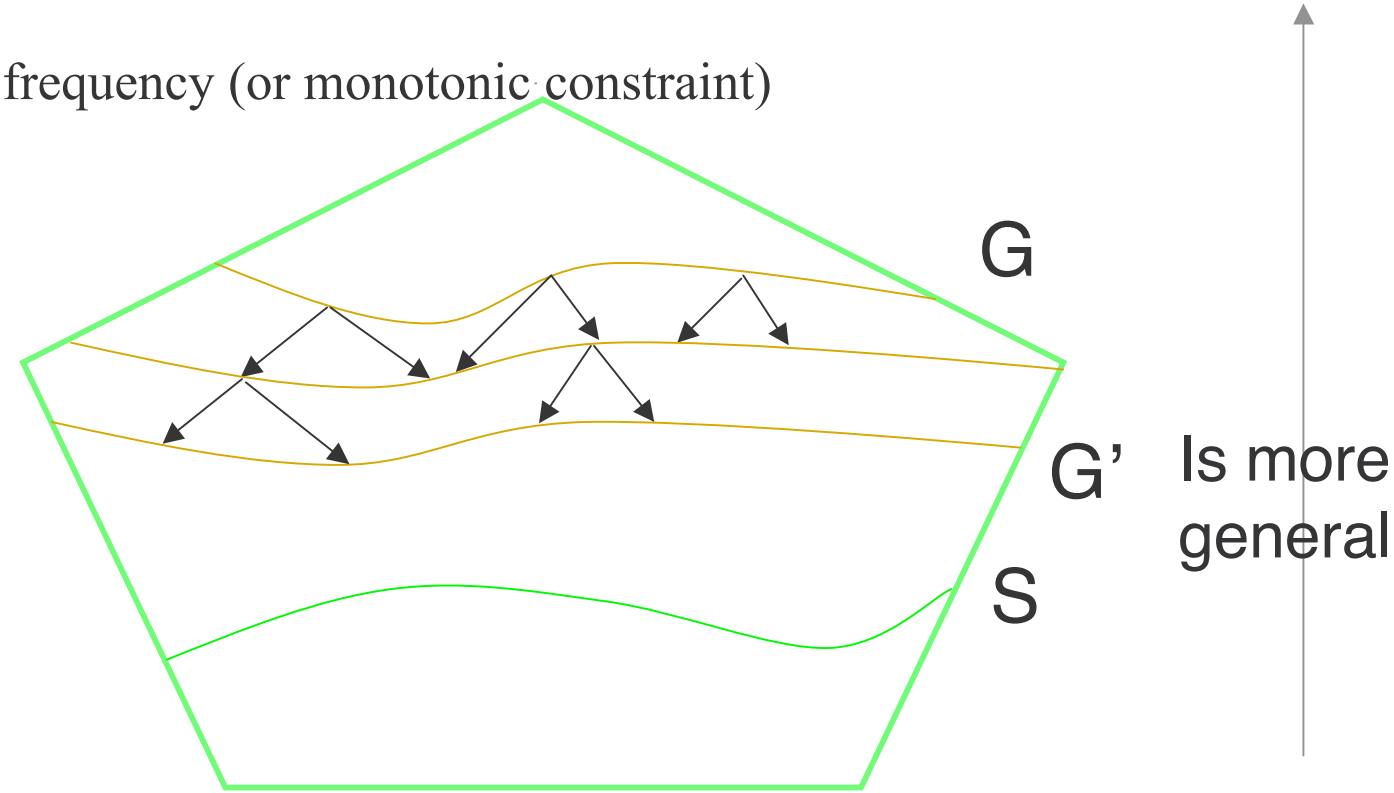


Dual computation

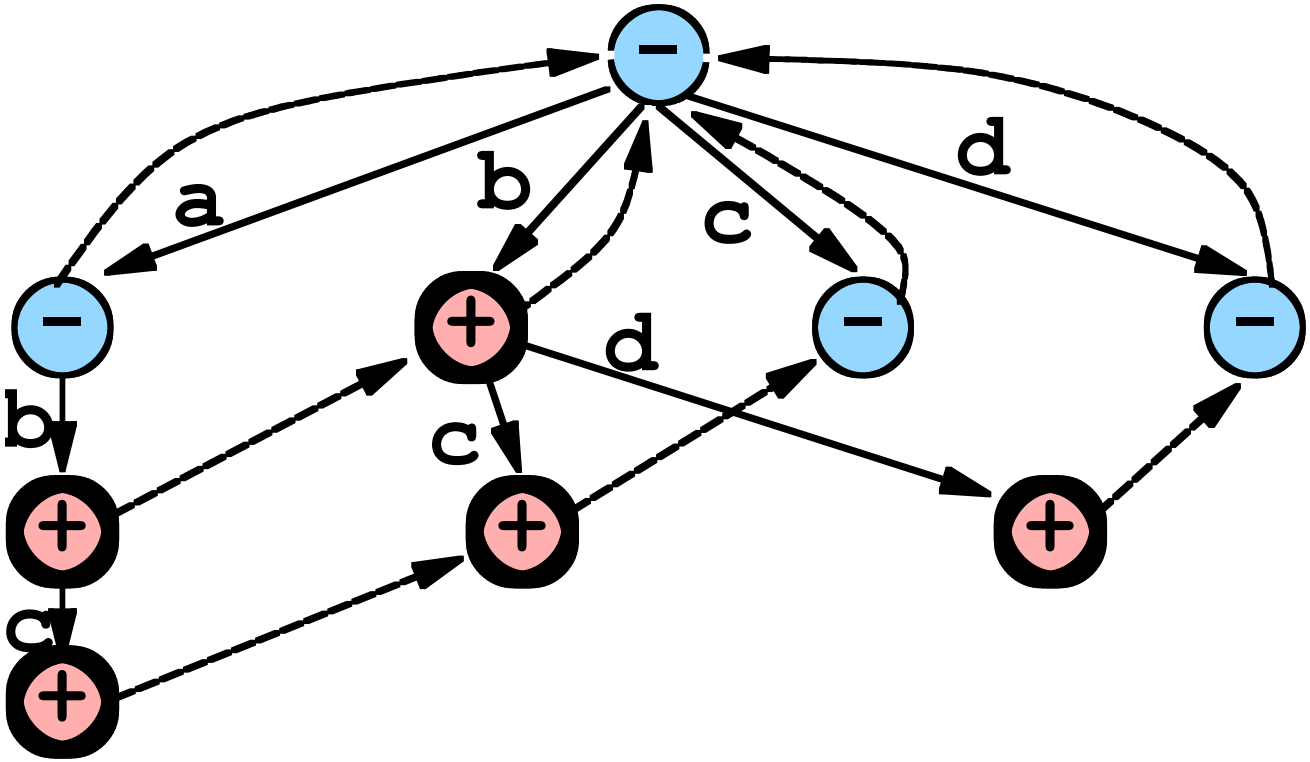


Level Wise Version Space Algorithm

max frequency (or monotonic constraint)



Version space tree



The HIV Data Set

- Developmental Therapeutics Program's AIDS Antiviral Screen Database (http://dtp.nci.nhi.gov)
- One of the largest public domain databases of this type
- Measures protection of human CEM cells from HIV-1 infection using a soluble formazan assay
- We retained 41768 compounds (after pre-processing the whole data set of 43382 ones)
  - 40282 Confirmed Inactive
  - 1069 Confirmed Moderately Active
  - 417 Confirmed Active

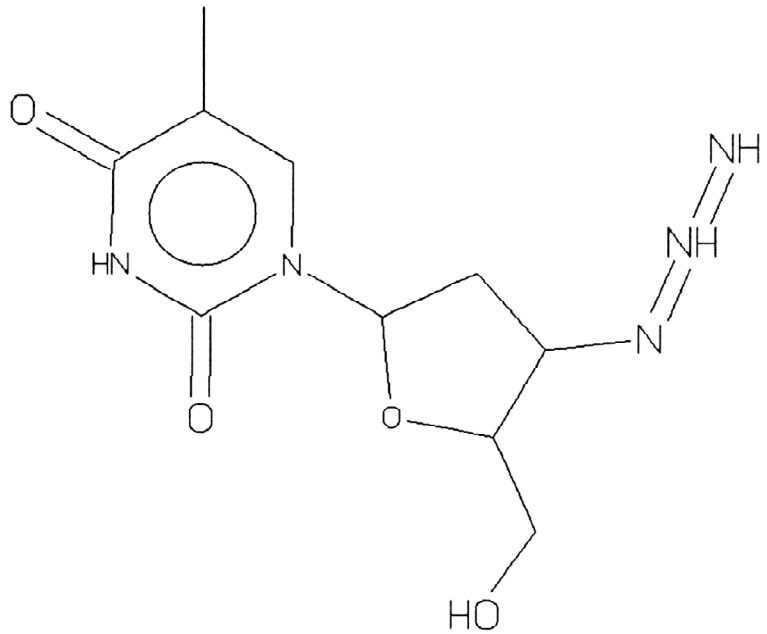
Experimental Setup

- Discover patterns that are, statistically significant, over-represented in the active compounds and under-represented in the inactive ones
- Minimum frequency in actives 3%, i.e. 13 compounds
- Maximum frequency on inactives computed using χ^2 (0.999) and size of classes
  - For CM :8; CI : 516
- Matching Smiles and Smarts using Daylight Tool !

Discovered Fragments (Actives vs. Inactives)

Shorth.	Fragm.	#CA	#CI	G/S	crit.
a	N-C-c:c:c:o	21	25	G	acc.
b	N=N=N-C-C-C-n:c:c:c=O	51	11	S	χ^2
c	N=N=N-C-C-C-n:c:n:c=O	51	11	S	χ^2
d	C-C-C-C-C-C-C-C-C-C-C-C-O-C-C-O-C-C-C-N=N=N	15	0	S	acc.
e	C-C-C-C-C-C-C-C-C-C-C-C-O-C-C-C-N=N=N	15	0	S	acc.
f	O=C-C-C-C-C-O-C-C-O-C-C-O-C=O	14	1	S	acc.
g	N=N=N-C-C-C-O-C-C-O-P=O	22	2	S	acc.
h	N=N=N-C-C-C-O-P=O	22	2	S	acc.

AZT (Azidothymidine)



The majority of these fragments are derivatives of AZT.

Gives insight into the structural requirements for anti-HIV activity.

A rediscovery that proves the principle

Post-processing

Combine fragments ?

Use of Fragments : SAR

- Use as fingerprints/descriptors for SAR model building
- Feed data into your favorite data mining/statistical package
  - Neural Nets
  - Decision Trees
  - (Logistic) Regression
  - Support Vector Machines
  - Bayesian Methods
  - Principal Component Analysis
  - …

Fragment Fingerprints

c:c-c:c	c:(6)-c(6)	Br-C	Br	C-O-c:(6)-N	C-O-c:c-N	Class
1	1	0	0	0	1	+1
0	0	0	0	0	1	-1
1	0	0	0	1	1	+1
...	...	...	...	...	...	...

```
1.6274328192175296 * c:c:c:c:c:c:c:c:c
1.4455302626881337 * C-Cl
1.3226667063998578 * C-C-C-C-N-C
1.310524380418045  * C-C-C-O
0.9516054404252757 * C-C=C
0.8654786477941714 * c:c:c:c:c:n
0.8243351055367271 * C-C-C-C=C
0.8197902253156605 * C-C-C-N-C
0.7969086522621357 * c:c:c-C=O
0.7819601605449131 * C-N-C
0.7796980414561107 * N-N
0.7498673413287917 * C-C-C-C-O
0.7276759799450657 * C-C-N-N
0.727514353351238  * N-O
0.7167965121501293 * C-O-C
0.6784153103780268 * C
0.6744410897500348 * C-N-c:c:c:c:c:c
0.6744410897500348 * C-N-c:c:c:c:c
0.6716119052528489 * c:c-N
0.5686660779334143 * C-C-N
```

Figure 4: The 20 strongest activating fragments for *Salmonella* mutagenicity derived from linear Support Vector Machines. Fragments are written in SMARTS notation: uppercase letters: aliphatic atoms, lowercase letters: aromatic atoms, - single bond, : aromatic bond, = double bond; baseline value: -0.24

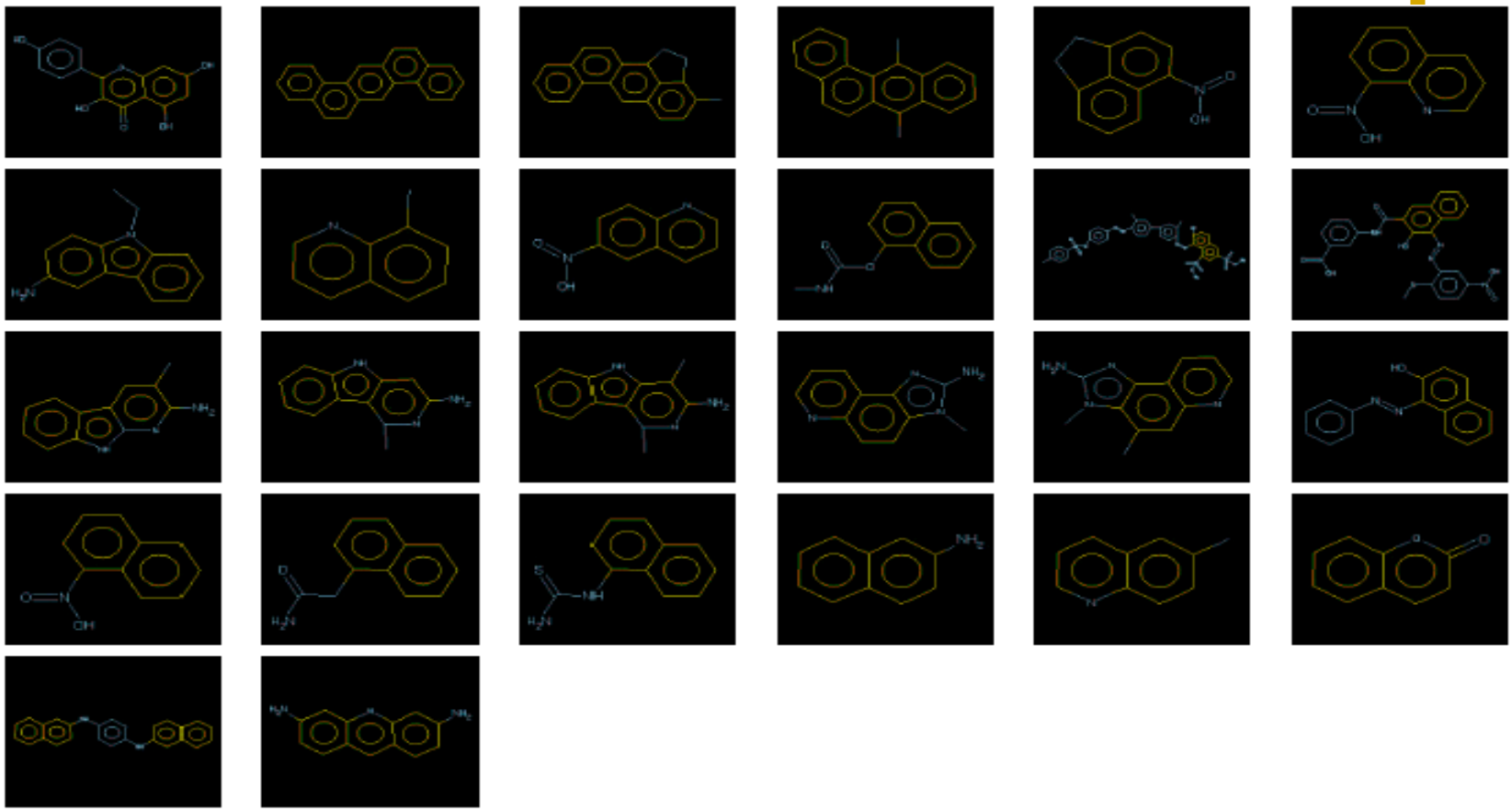
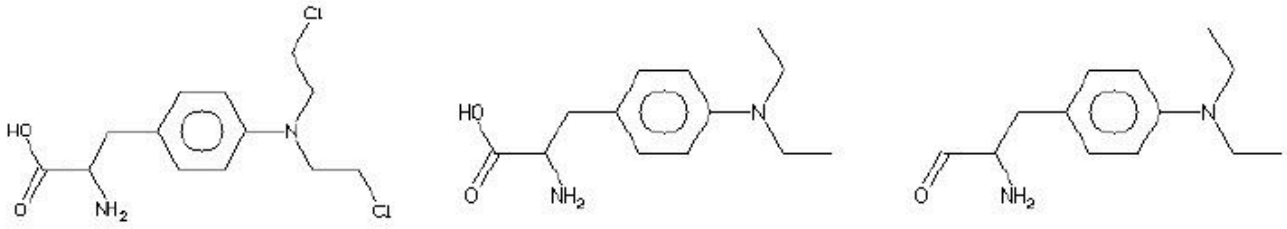


Figure 6: Mutagenic compounds containing the fragment C:C:C:C:C:C:C:C:C. Atoms matching this fragment are marked in yellow.

```
-1.479449078121286  * Cl-C-Cl
-1.4528269249653274 * C-C-C=C-C
-1.0145947939115687 * C-N-c:c
-1.0145947939115687 * C-N-c:c:c
-0.9492959881012157 * C-C
-0.9474885039899876 * C-C-N-C
-0.9402207493855474 * C-O-C=O
-0.937214552267573  * c:c:c:c:c:c-S
-0.937214552267573  * c:c:c:c:c-S
-0.937214552267573  * c:c:c:c-S
-0.9115486314638905 * C-C-C-C=O
-0.8877782140374197 * C-C-C-C
-0.8678653536715137 * c:c:c:c:c:c:c:c:c:c
-0.8568018292049271 * c:c:n:c:c
-0.7574483341970001 * Cl
-0.7529686472886363 * O-C=O
-0.7447971289365931 * C-C-C-N
-0.7285699786145916 * O
-0.7168970154384797 * C-C-C-C-C
-0.6759056107684382 * c:n
```

Figure 5: The 20 strongest deactivating fragments for bacterial mutagenicity derived from linear Support Vector Machines. Fragments are written in SMARTS notation: uppercase letters: aliphatic atoms, lowercase letters: aromatic atoms, - single bond, : aromatic bond, = double bond; baseline value: -0.24

Original Structure | **Modification 1** | **Modification 2**



10 most important activating fragments

```
1.4455302626881337 * C-Cl
1.310524380418045 * C-C-C-O
0.7819601605449131 * C-N-C
0.6784153103780268 * C
0.6744410897500348 * C-N-C:C:C:C:C:C
0.6744410897500348 * C-N-C:C:C:C:C
0.6716119052528489 * C:C-N
0.5686660779334143 * C-C-N
0.5402835372535206 * N
0.4510768731408878 * C:C
```

```
1.310524380418045 * C-C-C-O
0.7819601605449131 * C-N-C
0.6784153103780268 * C
0.6744410897500348 * C-N-C:C:C:C:C:C
0.6744410897500348 * C-N-C:C:C:C:C
0.6716119052528489 * C:C-N
0.5686660779334143 * C-C-N
0.5402835372535206 * N
0.4510768731408878 * C:C
0.4276304505150429 * C:C:C:C-N
```

```
0.7819601605449131 * C-N-C
0.6784153103780268 * C
0.6744410897500348 * C-N-C:C:C:C:C:C
0.6744410897500348 * C-N-C:C:C:C:C
0.6716119052528489 * C:C-N
0.5686660779334143 * C-C-N
0.5402835372535206 * N
0.4510768731408878 * C:C
0.4276304505150429 * C:C:C:C-N
0.3873012410340486 * C-C:C:C:C:C
```

Prediction

Mutagen (1.00)	Unclassifyable (0.007)	Nonmutagen (-0.57)

Figure 7: The consequences of removing activating fragments from Melphalan (CAS 148-82-3)

Chemical Similarity

- Comparison of different chemical databases
  - Characterize differences among different data sets using differences in fragments
- Special case :
  - Compare one compound against a database (wrt the fragments that occur)
- Lazar system (Christoph Helma)

Use of Fragments : SAR

- Several experiments reported on problems from predictive toxicology, cf. Kramer and De Raedt, ICML 01
  - Best results in combination with SVMs
  - 2 year rodent carcinogenicity assay (NTP) ~ 70 % ~ 500 compounds
  - Mutagenicity (Ames Test) ~ 80% ~ 800 compounds
- Method has proven its use in several benchmarks problems

Ongoing Work MolFea

- Work with branched fragments instead of linear sequences
  - conceptually easy, computationally more expensive
- Use abstractions, e.g. H-bond-donor/acceptor; lipophilic center, …
- Deriving 3D fragments
  - Annotate fragments with 3D information
  - Initial implementation works
  - Goal : mining for pharmacophores
- Integrate MolFea in existing chemical databases with GUI for interactive exploration
- Various activities on the solver side
- Applications to strings of proteins, genes, dna

Boolean Inductive Queries

Any monotonic or anti-monotonic constraint c,

and any membership function (e.g. $f \in P$)

is an atom.

An inductive query is a boolean formula over atoms.

E.g. $(f \in P)$ and $[freq(f, D1) > x$ or $freq(f, D2) < y]$ and $f < abbbcccc$

The query evaluation problem

Given

an inductive database

an inductive query q

Find a characterisation of $sol(q)$

Query optimization problem

- Evaluation of a primitive p has associated cost c(p)
- Find : a strategy to compute all solutions whose expected cost is minimal

- Open problem
- Needs estimates for expected number of solutions
- Database theory ….

Reasoning

Claim (subsumption)

Let q_1 and q_2 be two queries such that $q_1 \models q_2$.

Then $sol(q_1) \subseteq sol(q_2)$

Background knowledge can also be used in this process.

E.g. $freq(f, D) > x$ and $x \geq y \rightarrow freq(f, D) > y$

E.g. $freq(f, D1) > x$ and $D1 \subseteq D2 \rightarrow freq(f, D2) > x$

E.g. $freq(f2, D) > x$ and $f1 \leq f2 \rightarrow freq(f1, D) > x$

Useful :

axioms about sets, generality, number theory

Subsumption is useful in the light of interactive querying and reuse of the results of previous queries

Memory organisation

- Consider
  - q1 : freq(f,D) > m
  - q2: freq(f,D U M) > m (q1 |= q2)
  - q3:freq(f,D) > m OR freq(f,M) > m (q3 |= q2)
- Scenario's
  - q1 answered and stored; q2 asked
  - q2 answered and stored; q1 asked
- Keep track of subset relations among pattern sets / data sets
- Keep track of relations among patterns (generality structure) within given pattern set

What can we identify ?

- Pattern domain
  - Language of patterns
    - (e.g., itemsets, association rules, sequences, graphs, dependencies, decision trees, clusters)
  - Evaluation functions
    - (e.g., frequency, closures, generality, validity, accuracy)
  - Primitive constraints
    - (e.g., minimal and maximal frequency, freeness, syntactic constraints, minimal accuracy)
- DM settings
  - local pattern mining (as here)

Other settings

- Given
  - Database D
  - Language of patterns L
  - Convex scoring function s
- Find: k patterns p in L whose score s(p,D) is maximal
- Convex criteria allows for branch-and-bound algorithm

Branch-and-bound

- Consider the following task
  - two data sets D1 and D2
  - find patterns p such that
    - d(p) = freq(p,D1) – freq(p,D2) and d(p) > x or d(p) is maximal
    - let's assume absolute frequencies
- Property
  - For any pattern q that is more specific than p, we have that d(q) ≤ freq(p,D1)
  - So, knowledge about the frequencies of p imposes an (upper) bound on d(q) for any more specific pattern q
  - This bound can be used for pruning together with the demand of maximality or the constraint x < d(q)
  - optimal, k best, specific bound

Principles

- Morishita et al. have shown that this works for
  - significant patterns using chi-square test, entropy gain, gini-index
  - have also shown that it can be paralellized
  - impressive experiments
- Extended towards multiple dimensions by Zimmermann-De Raedt

Constraint-Based Clustering

- Queries generate data sets rather than patterns (work by Albrecht Zimmermann)
- Imagine constraints on data sets instead of on patters
  - E.g., insame(e1,e2)
  - E.g., indiff(e1,e2)
  - freq(p,C1) > x and freq(p,C2) < y and …
  - card(C1) > y
  - now p is given and the Ci are being queried
- Mathematical programming
  - reformulate constraints +
  - optimization criterion
  - Problem : non-linearity

Where to go from here ?

- Other forms of tasks ?
  - Clustering (some initial works exist)
    - Formulate constraints on no. of desired clusters, and cluster membership
  - Prediction
    - Some approaches to decision tree learning exist
- Other forms of algorithms ?
  - Instead of “all solutions” find “best” or “plausible” solutions
  - Approximation/heuristic algorithms
  - Cf. constraint programming
- Integration in databases
  - Has received some attention for SQL, LDL, relational algebra though much of it as syntactic sugar

Conclusions

- Constraint based mining
  - Inductive queries
  - Various types / problems / approaches
  - Largely local pattern mining
- Illustration of use
  - Molecular feature mining as an appli
- Many remaining open problems and opportunities for research