Euclidean Embedding of Co-Proven Queries

Presentation of Master Thesis

Hannes Schulz

Albert-Ludwigs-Universität Freiburg

2009-5-20

THE BLIND AND THE ELEPHANT

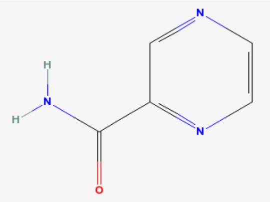
THE BLIND AND THE ELEPHANT

Outline

Outline

Outline

Relational Databases



atom

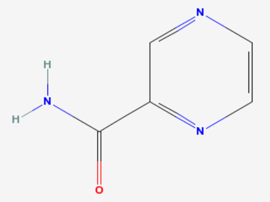
id	name
1	h
2	h
3	n
4	c
5	o

bond

id	a1	a2	type
1	1	3	1
2	2	3	1
3	3	4	1
4	4	5	2

...

Relational Databases



atom

id	name
1	h
2	h
3	n
4	c
5	o

bond

id	a1	a2	type
1	1	3	1
2	2	3	1
3	3	4	1
4	4	5	2

. . .

- Entities are discrete
- Elements part of arbitrary number of relations
- Implicit information in views
  - No mapping to feature vectors

Inductive Logic Programming to the Rescue

```
atom(1,d1_1,c,22,−0.117).
atom(1,d1_2,c,22,−0.117).
atom(1,d1_3,c,22,−0.117).
...
bond(1,d1_1,d1_2,7).
bond(1,d1_2,d1_3,7).
bond(1,d1_3,d1_4,7).
bond(1,d1_4,d1_5,7).
...
```

Inductive Logic Programming

- Retains complexity of DB in interpretations

Inductive Logic Programming to the Rescue

```
ring(K) :-
   atom(K,A,c,_,_),
   bond(K,A,B,_),
   atom(K,B,c,_,_),
   bond(K,B,C,_),
   ...
   bond(K,F,A,_).
```

Inductive Logic Programming

- Retains complexity of DB in interpretations
- Uses rules to reflect implicit information

Inductive Logic Programming to the Rescue

```
atom(K,A,c,_,_),
bond(K,A,B,_),
atom(K,B,c,_,_),
bond(K,B,C,_)
```

is frequent in organic molecules

Inductive Logic Programming

- Retains complexity of DB in interpretations
- Uses rules to reflect implicit information
- Abstracts using variables in queries

Outline

ILP Algorithms: Lifted from Propositional Case

One table to *n* tables

- Pattern miners (Warmr)
- Rule learners (Aleph)
- Decision Trees (Tilde)
- ...

ILP Algorithms: Lifted from Propositional Case

One table to *n* tables

- Pattern miners (Warmr)
- Rule learners (Aleph)
- Decision Trees (Tilde)
- ...

Usually,

- Input: interpretations, rules (“background knowledge”)
- Search (sub-)space of logical formulæ
- Output:
  - Pattern miners: interesting patterns
  - Classification: concept definitions

Back to the Elephant

Input and output of ILP algorithms feel like elephant parts.

Mutagenesis Pattern Mining

```
key(A),attyp(A,B,28),attyp(A,C,28),sbond(A,B,C,1)
key(A),atel(A,B,c),atel(A,C,c),attyp(A,D,27),sbond(A,
    B,C,1),sbond(A,B,D,7),carbon_6_rings(A)
key(A),attyp(A,B,22),methyls(A)
key(A),atel(A,B,c),atel(A,C,c),atel(A,D,h),atel(A,E,n
    ),sbond(A,B,C,7),sbond(A,B,D,1),sbond(A,C,E,1),
    benzenes(A),ring_size_5s(A)
key(A),atel(A,B,c),atel(A,C,c),atel(A,D,h),atel(A,E,n
    ),attyp(A,F,10),sbond(A,B,C,7),sbond(A,B,D,1),
    sbond(A,C,E,1)
key(A),atel(A,B,c),attyp(A,C,10),attyp(A,D,27),attyp(
    A,E,27),sbond(A,B,C,1),sbond(A,D,E,7)
key(A),atel(A,B,c),attyp(A,C,21),attyp(A,D,26),attyp(
    A,E,26),sbond(A,B,C,7),sbond(A,B,D,7),sbond(A,C,
    E,7)
key(A),atel(A,B,c),atel(A,C,c),atel(A,D,n),attyp(A,E
    ,22),sbond(A,B,D,1),sbond(A,B,E,7),sbond(A,C,E
    ,1)
```

Back to the Elephant

Input and output of ILP algorithms feel like elephant parts.

Mutagenesis Pattern Mining

```
key(A),attyp(A,B,28),attyp(A,C,28),sbond(A,B,C,1)
key(A),atel(A,B,c),atel(A,C,c),attyp(A,D,27),sbond(A,
    B,C,1),sbond(A,B,D,7),carbon_6_rings(A)
key(A),attyp(A,B,22),methyls(A)
key(A),atel(A,B,c),atel(A,C,c),atel(A,D,h),atel(A,E,n
    ),sbond(A,B,C,7),sbond(A,B,D,1),sbond(A,C,E,1),
    benzenes(A),ring_size_5s(A)
key(A),atel(A,B,c),atel(A,C,c),atel(A,D,h),atel(A,E,n
    ),attyp(A,F,10),sbond(A,B,C,7),sbond(A,B,D,1),
    sbond(A,C,E,1)
key(A),atel(A,B,c),attyp(A,C,10),attyp(A,D,27),attyp(
    A,E,27),sbond(A,B,C,1),sbond(A,D,E,7)
key(A),atel(A,B,c),attyp(A,C,21),attyp(A,D,26),attyp(
    A,E,26),sbond(A,B,C,7),sbond(A,B,D,7),sbond(A,C,
    E,7)
key(A),atel(A,B,c),atel(A,C,c),atel(A,D,n),attyp(A,E
    ,22),sbond(A,B,D,1),sbond(A,B,E,7),sbond(A,C,E
    ,1)
```

See the Whole Picture

Idea:

- Embed queries and interpretations into common 2D space

Mutagenesis Pattern Mining

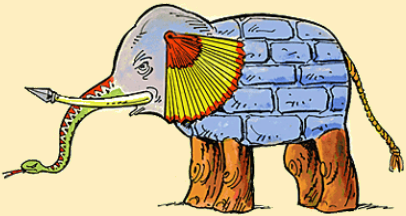
```
key(A),attyp(A,B,28),attyp(A,C,28),sbond(A,B,C,1)
key(A),atel(A,B,c),atel(A,C,c),attyp(A,D,27),sbond(A,
        B,C,1),sbond(A,B,D,7),carbon_6_rings(A)
key(A),attyp(A,B,22),methyls(A)
key(A),atel(A,B,c),atel(A,C,c),atel(A,D,h),atel(A,E,n
        ),sbond(A,B,C,7),sbond(A,B,D,1),sbond(A,C,E,1),
        benzenes(A),ring_size_5s(A)
key(A),atel(A,B,c),atel(A,C,c),atel(A,D,h),atel(A,E,n
        ),attyp(A,F,10),sbond(A,B,C,7),sbond(A,B,D,1),
        sbond(A,C,E,1)
key(A),atel(A,B,c),attyp(A,C,10),attyp(A,D,27),attyp(
        A,E,27),sbond(A,B,C,1),sbond(A,D,E,7)
key(A),atel(A,B,c),attyp(A,C,21),attyp(A,D,26),attyp(
        A,E,26),sbond(A,B,C,7),sbond(A,B,D,7),sbond(A,C,
        E,7)
key(A),atel(A,B,c),atel(A,C,c),atel(A,D,n),attyp(A,E
        ,22),sbond(A,B,D,1),sbond(A,B,E,7),sbond(A,C,E
        ,1)
```

See the Whole Picture

Idea:

- Embed queries and interpretations into common 2D space
- Show – at a glance! – how they are related

Mutagenesis Pattern Mining



SEE THE WHOLE PICTURE

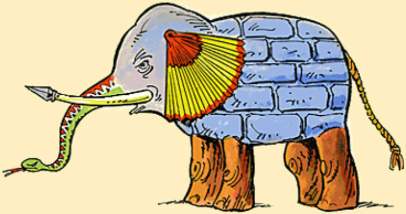
Idea:

- Embed queries and interpretations into common 2D space
- Show – at a glance! – how they are related

However:

- Embedding algorithms need a distance measure

MUTAGENESIS PATTERN MINING

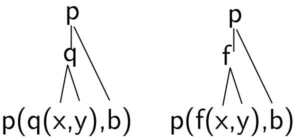


Outline

Distances in ILP

Typical ILP distances measures

- Based on syntax
- Recursively defined over terms $\mathcal{T}$ and predicates $\mathcal{F}$



$$dist_{NC}(\cdot,\cdot) = \frac{1}{4}$$

Nienhuys-Cheng Distance

$$\begin{aligned}
\bigwedge_{t \in \mathcal{T}} \mathrm{dist}_{nc}(t, t) &= 0 \\
\bigwedge_{p/n \in \mathcal{F}} \bigwedge_{q/m \in \mathcal{F}} \mathrm{dist}_{nc}(p(s_1, \ldots, s_n), q(t_1, \ldots, t_m)) &= 1 \\
\bigwedge_{p/n \in \mathcal{F}} \mathrm{dist}_{nc}(p(s_1, \ldots, s_n), p(t_1, \ldots, t_n)) &= \frac{1}{2n} \sum_{i=1}^{n} \mathrm{dist}_{nc}(s_i, t_i).
\end{aligned}$$

Syntax-Based Distances not "Grounded"

- Queries q and p are equivalent with respect to data in E

$$\bigwedge_{e \in E} \big(q(e) \leftrightarrow p(e)\big).$$

- ▶ Syntactic difference tells us little about data!

Syntax-Based Distances not "Grounded"

- Queries q and p are equivalent with respect to data in E

$$\bigwedge_{e \in E} \big(q(e) \leftrightarrow p(e)\big).$$

  - Syntactic difference tells us little about data!

- Two queries q and p are not related with respect to the data

$$\bigwedge_{e \in E} \big((q(e) \rightarrow \neg p(e)) \wedge (p(e) \rightarrow \neg q(e))\big).$$

  - $p \equiv \neg q$? Probably not. Property of the data? Probably.

Define Similarity Using Database

Query-Query Similarity

$$\mathrm{sim}(q_1, q_2) = \left|\left\{e | e \in E \wedge q_1(e) \wedge q_2(e)\right\}\right|$$

- Co-Proven queries are similar

Define Similarity Using Database

Query-Query Similarity

$$\mathrm{sim}(q_1, q_2) = \left|\left\{e | e \in E \wedge q_1(e) \wedge q_2(e)\right\}\right|$$

Query-Interpretation Similarity

$$\mathrm{sim}(q, e) = \begin{cases} 1 & \text{if } q(e) \\ 0 & \text{sonst.} \end{cases}$$

- Co-Proven queries are similar
- Queries are similar to interpretations in which they are true

Define Similarity Using Database

Query-Query Similarity

$$\text{sim}(q_1, q_2) = \Big|\{e | e \in E \wedge q_1(e) \wedge q_2(e)\}\Big|$$

Query-Interpretation Similarity

$$\text{sim}(q, e) = \begin{cases} 1 & \text{if } q(e) \\ 0 & \text{sonst.} \end{cases}$$

- Co-Proven queries are similar
- Queries are similar to interpretations in which they are true
- Can be seen as joint probability when normalized:
  $p_{QQ}(q_1, q_2) = \eta \cdot \text{sim}(q_1, q_2)$ and $p_{QE}(q, e) = \nu \cdot \text{sim}(q, e)$

Define Similarity Using Database

Query-Query Similarity

$$\mathrm{sim}(q_1, q_2) = \left|\left\{e | e \in E \wedge q_1(e) \wedge q_2(e)\right\}\right|$$

Query-Interpretation Similarity

$$\mathrm{sim}(q, e) = \begin{cases} 1 & \text{if } q(e) \\ 0 & \text{sonst.} \end{cases}$$

- Co-Proven queries are similar
- Queries are similar to interpretations in which they are true
- Can be seen as joint probability when normalized: $p_{QQ}(q_1, q_2) = \eta \cdot \mathrm{sim}(q_1, q_2)$ and $p_{QE}(q, e) = \nu \cdot \mathrm{sim}(q, e)$
- Queries with same co-occurrence can be removed

Outline

Outline

CODE

- CODE (Globerson et al. 2007): template for Co-Occurrence Data Embedding algorithms
- We use instance of CODE

CODE for Relational Data and Queries

1. Define $\Phi(\cdot)$, $\Psi(\cdot)$ placing queries, interpretations in 2D space
2. Distance in 2D space reflects co-occurrence probability

$$p_{QQ}(q_1, q_2) \propto \exp(-\left\|\Phi(q_1) - \Phi(q_2)\right\|^2)$$
$$\frac{p_{QE}(q, e)}{p(e)} \propto \exp(-\left\|\Phi(q) - \Psi(e)\right\|^2)$$

3. Maximize log-likelihood of embedding

$$l(\Phi, \Psi) = \sum_{q,e} p_{QE}(q, e) \log p_{QE}(q, e) + \eta \sum_{q_1, q_2} p_{QQ}(q_1, q_2) \log p_{QQ}(q_1, q_2)$$

Outline

Outline

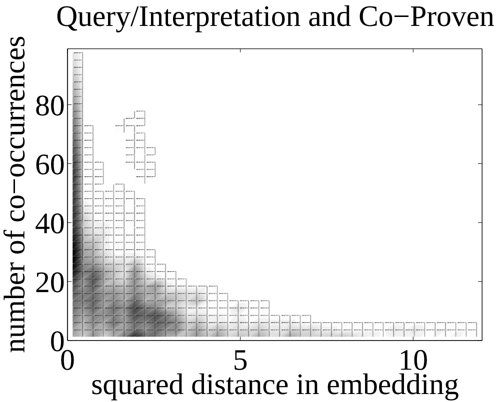
Datasets and Pattern Miners

In molecular databases: Interpretations=molecules; Queries=properties

- Mutagenesis: 188 molecules, ca. 30 atoms per molecule. C-armr finds 16 Mio pattern, 505 semantically different.
- AIDS: Sampled 800 molecules, Molfea finds 3310 linear fragments (e. g.`C-C=C-N-C`).
- Estrogen: 232 chemicals, Molfea finds 843 different linear fragments.

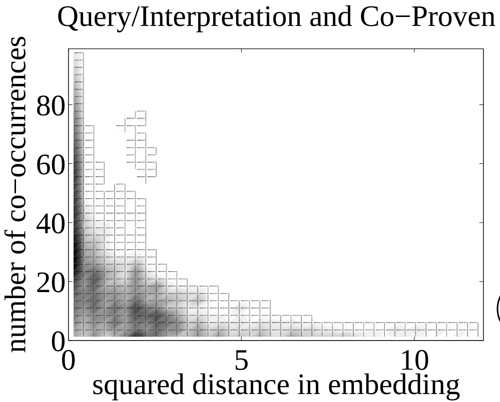
Outline

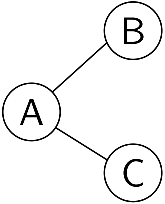
CODE Best Retains Co-Occurrence Statistics



- Cannot be perfect because of intransitivity of co-occurrence

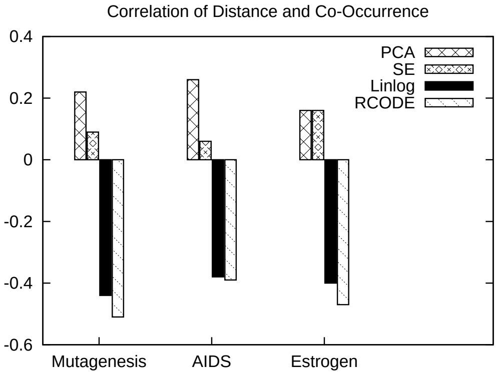
CODE Best Retains Co-Occurrence Statistics





- Cannot be perfect because of intransitivity of co-occurrence

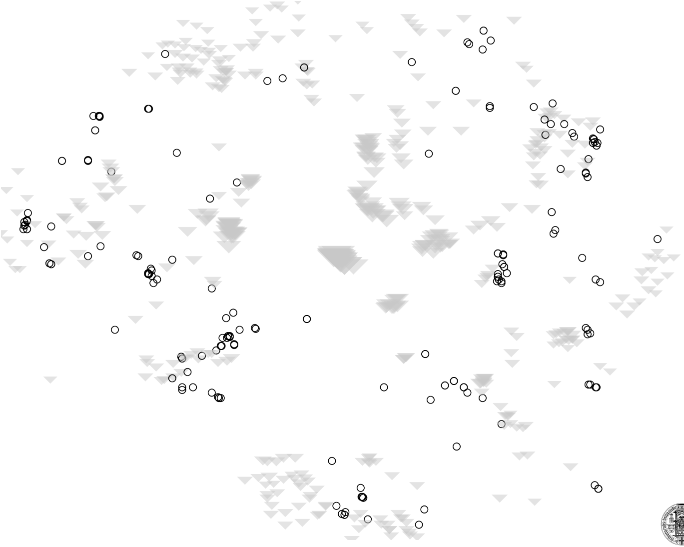
CODE Best Retains Co-Occurrence Statistics



- Cannot be perfect because of intransitivity of co-occurrence

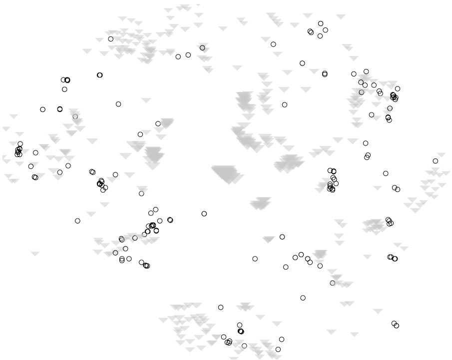
Outline

STARTING POINT



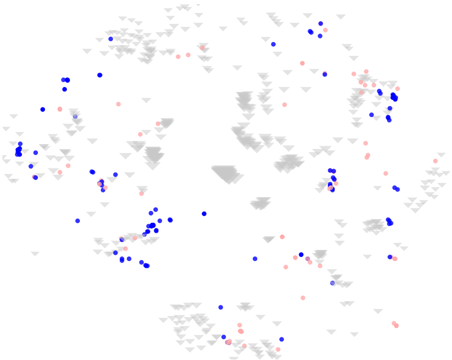
SUPPLYING ADDITIONAL INFORMATION

- Size: Frequency



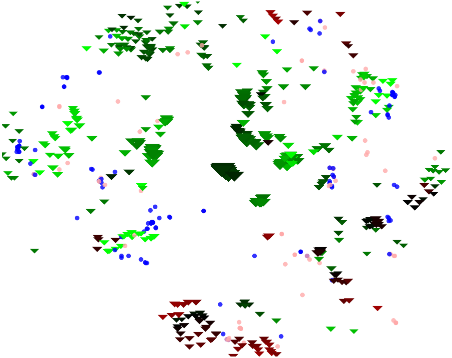
SUPPLYING ADDITIONAL INFORMATION

- Size: Frequency
- Color: interpretation class



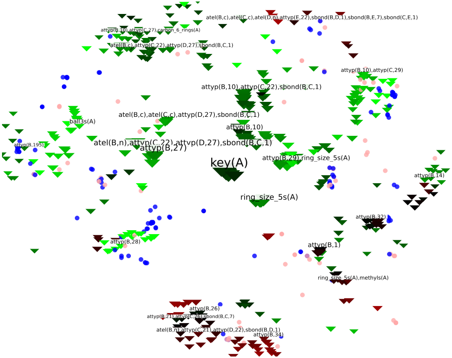
Supplying Additional Information

- Size: Frequency
- Color: interpretation class
- Color: class affinity of queries



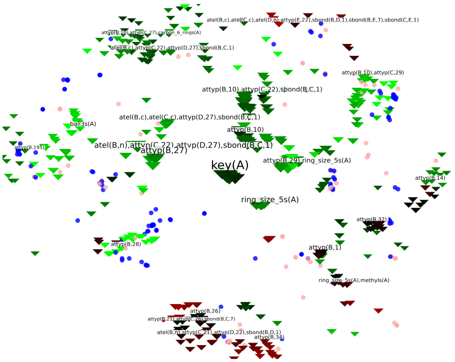
Supplying Additional Information

- Size: Frequency
- Color: interpretation class
- Color: class affinity of queries
- Mark representative queries

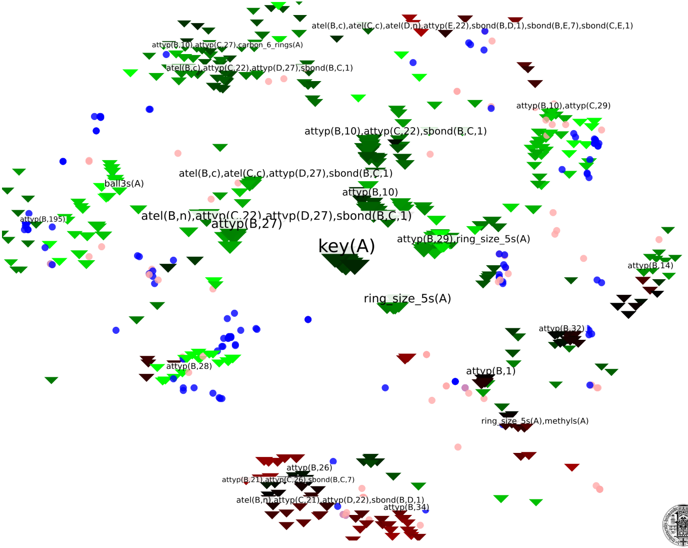


SUPPLYING ADDITIONAL INFORMATION

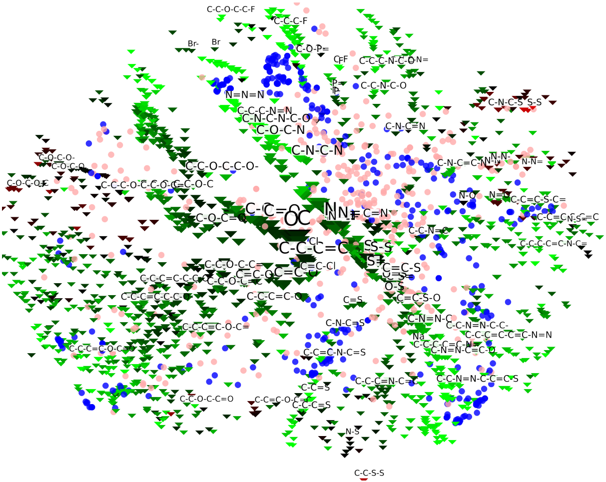
- Size: Frequency
- Color: interpretation class
- Color: class affinity of queries
- Mark representative queries
- Interactivity to disambiguate embedding

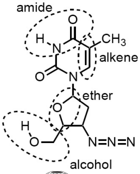


Mutagenesis Dataset



AIDS Dataset





Outline

Conclusion

- First visualization method for relational data and queries
- Interpretations and queries placed in common Euclidean space
- Use semantically grounded distance measure
- In embedding, co-proven queries are close to each other, interpretations close to their queries
- Developed tools to visualize embedding with side-information

Future Work

- Improve interactive program to full-fledged Visual Analytics application
- Use graph miner to crawl databases, use for visualization
- Explore use of technique for other applications with large binary feature vectors (Genetic Algorithms, Bag-of-Words,. . .)

Thanks!

(Elephant pictures from `http://www.wordinfo.info/words/index/info/view_unit/1/?letter=B&spage=3`).