Stability and Other Indices for Concept-based Clustering

Sergei O. Kuznetsov

National Research University Higher School of Economics, Moscow

Workshop “Information, Uncertainty and Imprecision”

Palacky University, Olomouc, June 5, 2012

Outline

Motivation and Goals

- On the one hand concept lattices are a nice tool for semiautomatic generation of taxonomies and ontologies, and just for clustering data
- On the other hand there can be exponentially many concepts as compared to the context size
- Let us keep only few “best” concepts
- But what does “best” mean?

Selection criteria

Selecting concepts wrt.

- intent and extent size constraints [Kuznetsov 1989], [Stumme 2000] (iceberg lattices)
- concept stability [Kuznetsov 1990], [Obiedkov, Roth 2006]
- concept separation [Klimushkin et al. 2010]
- concept probability [Klimushkin et al. 2010] (rediscovered concept probability from [Emillion 2008]

Stability definition

Let $\mathbb{K} = (G, M, I)$ be a formal context and (A, B) be a formal concept of $\mathbb{K}$.

Definition

The intentional stability $\sigma_{in}(A, B)$ of (A, B), or $\sigma_{in}(A)$, is defined as follows:

$$\sigma_{in}(A, B) = \frac{|C \subseteq A \mid C' = B|}{2^{|A|}}$$

Definition

The extentional stability $\sigma_{ex}(A, B)$ of (A, B), or $\sigma_{ex}(B)$, is defined as follows:

$$\sigma_{ex}(A, B) = \frac{|C \subseteq B \mid C'' = B|}{2^{|B|}}$$

Stability applications

Stability is a very useful tool for selecting interesting concepts of the concept lattice. Here are some examples:

- Study of defects in plastic production (S.Kuznetsov, 1990)
- Study of epistemic communities (S.Obiedkov, C. Roth et al. 2006-2008)
- Choosing cure trajectory (N.Jay et al., 2008)
- Filtering noise in contexts (M.Klimushkin et al., 2010)
- Categorizing French verbs (I.Falk et al., 2011)

Restructuring Möbius function: An approach based on concept stability

The numerator of intensional stability $\gamma(A, B) = |C \subseteq A \mid C' = B|$ is the number of all generators of the concept (A, B), so

$$2^{|A|} = \sum_{(C,D) \leq (A,B)} \gamma(C, D)$$

Restructuring Möbius function: An approach based on concept stability

The numerator of intensional stability $\gamma(A, B) = |C \subseteq A \mid C' = B|$ is the number of all generators of the concept (A, B), so

$$2^{|A|} = \sum_{(C,D) \leq (A,B)} \gamma(C, D)$$

$$\gamma(A, B) = \sum_{(C,D) \leq (A,B)} 2^{|C|} \mu((C, D), (A, B)),$$

where $\mu(A, B)$ is the Möbis function of the concept lattice.

Complexity of computing stability

- Given a context (G, M, I) and a concept (A, B), the problems of computing $\sigma_{in}(A, B)$ and $\sigma_{ex}(A, B)$ are #P-complete

Definition: A counting problem is in #P if there is a non-deterministic, polynomial time Turing machine that, for each instance I of the problem, has a number of accepting computations that is exactly equal to the number of distinct solutions for instance I.

Examples of #P-complete problems:

- Given a matrix, output its permanent
- Given a bipartite graph, output the number of its perfect matchings
- Given a CNF, output the number of its satisfying assignments
- Given a graph, output the number of its vertex covers
- Given a context, output the number of its concepts

FPRAS

Many counting problems that are #P-complete can be solved approximately by randomized polynomial algorithms

FPRAS

Many counting problems that are #P-complete can be solved approximately by randomized polynomial algorithms

Definition

Fully Polynomial Randomized Approximation Scheme (FPRAS)

FPRAS

Many counting problems that are #P-complete can be solved approximately by randomized polynomial algorithms

Definition

Fully Polynomial Randomized Approximation Scheme (FPRAS)

- *time complexity is polynomial in* $|INPUT|$ *and* ε^{-1}

FPRAS

Many counting problems that are #P-complete can be solved approximately by randomized polynomial algorithms

Definition

Fully Polynomial Randomized Approximation Scheme (FPRAS)

- *time complexity is polynomial in* $|INPUT|$ *and* ε^{-1}
- $Pr[(1-\varepsilon)\cdot ans \leq N \leq (1+\varepsilon)\cdot ans)] \geq \frac{3}{4}$

FPRAS

Many counting problems that are #P-complete can be solved approximately by randomized polynomial algorithms

Definition

Fully Polynomial Randomized Approximation Scheme (FPRAS)

- *time complexity is polynomial in* $|INPUT|$ *and* ε^{-1}
- $Pr[(1-\varepsilon)\cdot ans \leq N \leq (1+\varepsilon)\cdot ans)] \geq \frac{3}{4}$

FPRAS

Many counting problems that are #P-complete can be solved approximately by randomized polynomial algorithms

Definition

Fully Polynomial Randomized Approximation Scheme (FPRAS)

- *time complexity is polynomial in* $|INPUT|$ *and* ε^{-1}
- $Pr[(1-\varepsilon)\cdot ans \leq N \leq (1+\varepsilon)\cdot ans)] \geq \frac{3}{4}$

Example: Number of truth assignments of a DNF (#DNF)

FPRAS

What about approximations with a fixed constant factor? (the approximation with any factor $1 \pm \varepsilon$ seems to be too strong condition)

FPRAS

What about approximations with a fixed constant factor? (the approximation with any factor $1 \pm \varepsilon$ seems to be too strong condition)

If we have an algorithm for a #P-complete problem with polynomial approximation $(q(|INPUT|) \cdot ans \leq N \leq p(|INPUT|) \cdot ans)$, where *ans* is the exact value being approximated, then there is an FPRAS.

FPRAS

What about approximations with a fixed constant factor? (the approximation with any factor $1 \pm \varepsilon$ seems to be too strong condition)

If we have an algorithm for a #P-complete problem with polynomial approximation ($q(|INPUT|) \cdot ans \leq N \leq p(|INPUT|) \cdot ans$), where *ans* is the exact value being approximated, then there is an FPRAS.

Why randomized?

FPRAS

What about approximations with a fixed constant factor? (the approximation with any factor $1 \pm \varepsilon$ seems to be too strong condition)

If we have an algorithm for a #P-complete problem with polynomial approximation ($q(|INPUT|) \cdot ans \leq N \leq p(|INPUT|) \cdot ans$), where *ans* is the exact value being approximated, then there is an FPRAS.

Why randomized?

For #P-complete problems no deterministic approximate algorithm is known.

Counting independent sets

Given a hypergraph $G = (V, \mathcal{E})$, $\mathcal{E} = \{E_1, \ldots, E_m\}$,
$U \subseteq V$ is called independent set if $E_i \not\subseteq U$, $1 \leq i \leq m$,
$U \subseteq V$ is called coindependent set if $U \not\subseteq E_i$, $1 \leq i \leq m$.

Counting independent set (#IS)
INPUT: A hypergraph G
OUTPUT: The number of independent sets (of all sizes) of G

Counting independent sets

Given a hypergraph $G = (V, \mathcal{E})$, $\mathcal{E} = \{E_1, \ldots, E_m\}$,
$U \subseteq V$ is called independent set if $E_i \not\subseteq U$, $1 \leq i \leq m$,
$U \subseteq V$ is called coindependent set if $U \not\subseteq E_i$, $1 \leq i \leq m$.

Counting independent set (#IS)
INPUT: A hypergraph G
OUTPUT: The number of independent sets (of all sizes) of G

There is no FPRAS for #IS, unless $NP = RP$ (still hard even in the case of graphs)

Counting non-closed sets

Counting non-closed sets (#NC)
INPUT: A formal context $\mathbb{K} = (G, M, I)$.
OUTPUT: The number of sets $B \subseteq M$ that $B'' \neq B$

Counting non-closed sets

Counting non-closed sets (#NC)
INPUT: A formal context $\mathbb{K} = (G, M, I)$.
OUTPUT: The number of sets $B \subseteq M$ that $B'' \neq B$

For any hypergraph G it is easy to construct a context $\mathbb{K}_G$ such that set A is closed iff A is a subset of some hyperedge of G.

Counting non-closed sets

Counting non-closed sets (#NC)
INPUT: A formal context $\mathbb{K} = (G, M, I)$.
OUTPUT: The number of sets $B \subseteq M$ that $B'' \neq B$

For any hypergraph G it is easy to construct a context $\mathbb{K}_G$ such that set A is closed iff A is a subset of some hyperedge of G.

Proposition

There is no FPRAS for #NC unless NP = RP

Approximate stability

Exact computing of concept stability is #P-complete, but can it be computed approximately with FPRAS?

Approximate stability

Exact computing of concept stability is #P-complete, but can it be computed approximately with FPRAS?

There is no FPRAS for stability computation (unless $NP = RP$), since otherwise there would have been an FPRAS for #NC.

Approximate stability

Exact computing of concept stability is #P-complete, but can it be computed approximately with FPRAS?

There is no FPRAS for stability computation (unless $NP = RP$), since otherwise there would have been an FPRAS for #NC.

However we can approximate stability with bounded absolute error using Monte-Carlo approach. By definition, $\sigma(A) = Pr(X'' = A)$, where X is chosen uniformly random from subsets of A.

Monte-Carlo method

GetStability(A, N)

```
answer ← 0
for i ← 1 to N
        do pick random subset X of A
            if X'' = A
                then answer ← answer + 1
answer ← answer / N
return answer
```

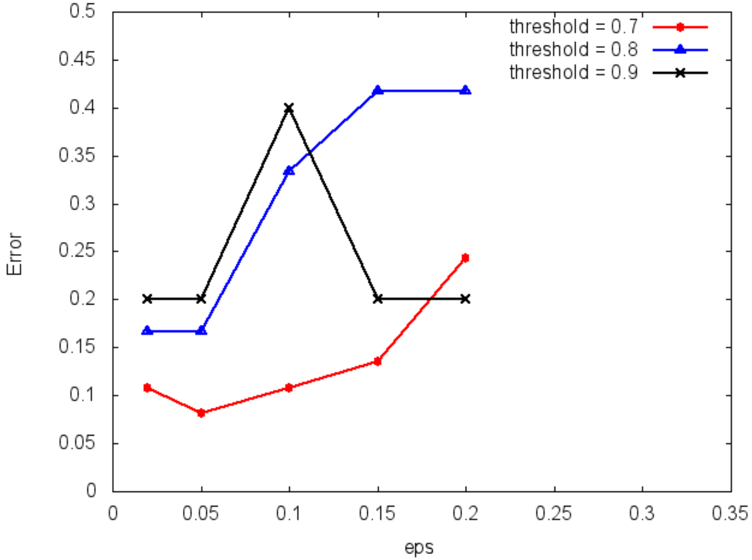
Experimental results on random contexts

The Y-axis (*Error*) gives the relative error

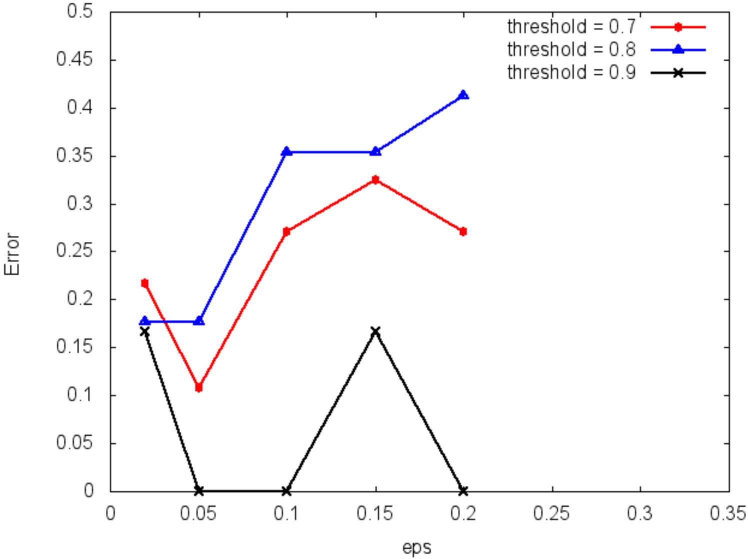
$$|S(\mathbb{K},\tilde{\sigma},\sigma_\theta)\Delta S(\mathbb{K},\sigma,\sigma_\theta)|/|S(\mathbb{K},\sigma,\sigma_\theta)|.$$

$S(\mathbb{K},\sigma,\sigma_\theta)$ denotes the set of all concepts with stability $\sigma \geq \sigma_\theta$; $S(\mathbb{K},\tilde{\sigma},\sigma_\theta)$ denotes the set of all concepts with approximate stability $\tilde{\sigma} \geq \sigma_\theta$, where σ_θ is a parameter (*stability threshold*). For every pair $g \in G$, $m \in M$ of a random context $\mathbb{K} = (G, M, I)$ one has $(g, m) \in I$ with probability d called *context density*.

Experimental results on random contexts



Experimental results on random contexts



Summary for Algorithmic Complexity of Stability

- The problem of computing stability of a concept is #P-complete
- Given a context, no FPRAS for counting non-closed subsets of attributes (objects) is possible unless RP = NP
- An approximate algorithm for computing stability, which can run in reasonable time for approximations with bounded absolute error, was proposed

Concept Separation Index

[M.Klimushkin, S.Obiedkov, C.Roth, ICFCA'2010]

- How much the objects covered by concept (A, B) differ from other objects from $G \setminus A$?
- How much the attributes covered by concept (A, B) differ from other attributes from $M \setminus B$?
- Concept separation index $S(A, B)$ gives a numerical measure to answer these questions

$$S(A, B) = \frac{|A| \cdot |B|}{\sum_{a \in A} |\{a\}'| + \sum_{b \in B} |\{b\}'| - |A| \cdot |B|}$$

Concept Probability Index

[M.Klimushkin, S.Obiedkov, C.Roth, ICFCA'2010], rediscovering the notion from [R.Emillion, 2008]

- Concept probability is the probability of the fact that a concept with the same intent will appear in a random context, attributes being independent.
- The probability p_m that an object has attribute m equals the proportion of objects in the original context that have this attribute
- The probability that a particular object has all attributes from B is $P_B = \prod_{m \in B} pm$.

$$P(B = B'') = \sum_{k=0}^{n} (\binom{n}{k} p_B^k \cdot (1 - p_B)^{n-k} \prod_{m \notin B} (1 - p_m^k))$$