Approximate Computation of Exact Association Rules

Saurabh Bansal[1] Sriram Kailasam[1] Sergei Obiedkov[2]

[1]IIT Mandi, Mandi, India

[2]HSE University, Moscow, Russia

July 1, 2021

Overview

- Computing the canonical basis of a formal context is hard: no total–polynomial time algorithm is known.

Overview

- Computing the canonical basis of a formal context is hard: no total–polynomial time algorithm is known.
  - Some algorithms generate all concept intents as a side product.

Overview

- Computing the canonical basis of a formal context is hard: no total–polynomial time algorithm is known.
  - Some algorithms generate all concept intents as a side product.
  - Other algorithms compute different bases, which can then be reduced to the canonical basis.

Overview

- Computing the canonical basis of a formal context is hard: no total–polynomial time algorithm is known.
  - Some algorithms generate all concept intents as a side product.
  - Other algorithms compute different bases, which can then be reduced to the canonical basis.
- Probably approximately correct computation of the canonical basis has been considered before,

Overview

- Computing the canonical basis of a formal context is hard: no total–polynomial time algorithm is known.
  - Some algorithms generate all concept intents as a side product.
  - Other algorithms compute different bases, which can then be reduced to the canonical basis.
- Probably approximately correct computation of the canonical basis has been considered before,
  - but has never been properly evaluated in terms of efficiency in practice.

Overview

- Computing the canonical basis of a formal context is hard: no total–polynomial time algorithm is known.
  - Some algorithms generate all concept intents as a side product.
  - Other algorithms compute different bases, which can then be reduced to the canonical basis.
- Probably approximately correct computation of the canonical basis has been considered before,
  - but has never been properly evaluated in terms of efficiency in practice.
- We define a notion of frequency-aware approximation and give a total–polynomial time probabilistic algorithm to compute it.

Overview

- Computing the canonical basis of a formal context is hard: no total–polynomial time algorithm is known.
  - Some algorithms generate all concept intents as a side product.
  - Other algorithms compute different bases, which can then be reduced to the canonical basis.
- Probably approximately correct computation of the canonical basis has been considered before,
  - but has never been properly evaluated in terms of efficiency in practice.
- We define a notion of frequency-aware approximation and give a total–polynomial time probabilistic algorithm to compute it.
- We experimentally evaluate the algorithm.

Formal Contexts

Formal context $\mathbb{K} = (G, M, I)$

- a set of objects G
- a set of attributes M
- objects are described with attributes: the binary relation $I \subseteq G \times M$

Formal Contexts

Formal context $\mathbb{K} = (G, M, I)$

- a set of objects G
- a set of attributes M
- objects are described with attributes: the binary relation $I \subseteq G \times M$

Derivation operators

For $A \subseteq G$ and $B \subseteq M$:

- $A' = \{m \in M \mid \forall g \in A\colon (g, m) \in I\}$
- $B' = \{g \in G \mid \forall m \in B\colon (g, m) \in I\}$

$A \mapsto A''$ and $B \mapsto B''$ are closure operators.
$\mathrm{Int}\,\mathbb{K} = \{B'' \mid B \subseteq M\}$ is the set of concept intents of $\mathbb{K}$.

Implications

Implication $A \to B$

$A, B \subseteq M$.

- An attribute subset $X \subseteq M$ is a model of an implication $A \to B$ if $A \not\subseteq X$ or $B \subseteq X$.
- $A \to B$ is valid in context $\mathbb{K}$ if $A' \subseteq B'$.

Valid implications are also called exact association rules.

Implications

- X is a model of an implication set $\mathcal{L}$ ($X \models \mathcal{L}$) if it is a model of every implication in $\mathcal{L}$.
- Mod $\mathcal{L}$ is the set of all models of $\mathcal{L}$.
- Two implication sets are equivalent if they have the same models.

Implications

- X is a model of an implication set $\mathcal{L}$ ($X \models \mathcal{L}$) if it is a model of every implication in $\mathcal{L}$.
- $\operatorname{Mod}\mathcal{L}$ is the set of all models of $\mathcal{L}$.
- Two implication sets are equivalent if they have the same models.

Closure operator $X \mapsto \mathcal{L}(X)$

Maps $X \subseteq M$ to the smallest model of all the implications in $\mathcal{L}$ containing X:

$$\mathcal{L}(X) = \bigcap \{Y \mid X \subseteq Y \subseteq M, \quad Y \models \mathcal{L}\}$$

Canonical Basis

Definition

A set $\mathcal{L}$ of implications over M is an *implication basis* of the context (G, M, I) if it is

- sound: each implication from $\mathcal{L}$ holds in (G, M, I);
- complete: each implication that holds in (G, M, I) follows from $\mathcal{L}$;
- non-redundant: no implication in $\mathcal{L}$ follows from other implications in $\mathcal{L}$.

Pseudo-closed set

A set $P \subseteq M$ is called pseudo-closed if $P \neq P''$ and $Q'' \subset P$ for every pseudo-closed $Q \subset P$. P is also called pseudo-intent.

Canonical Basis

Definition

A set $\mathcal{L}$ of implications over M is an *implication basis* of the context (G, M, I) if it is

- sound: each implication from $\mathcal{L}$ holds in (G, M, I);
- complete: each implication that holds in (G, M, I) follows from $\mathcal{L}$;
- non-redundant: no implication in $\mathcal{L}$ follows from other implications in $\mathcal{L}$.

Pseudo-closed set

A set $P \subseteq M$ is called pseudo-closed if $P \neq P''$ and $Q'' \subset P$ for every pseudo-closed $Q \subset P$. P is also called pseudo-intent.

Canonical basis (Duquenne–Guigues basis)

is the set of all implications of the form $P \to P''$ where P is pseudo-closed.

The canonical basis is minimal in the number of implications among all equivalent implication sets.

Frequent Implications

- The support of $A \subseteq M$ is $|A'|$.
- The relative support of $A \subseteq M$ is $|A'|/|G|$.
- The (relative) support or frequency of $A \to B$ is the (relative) support of $A \cup B$.

Computing the Canonical Basis

- Known exact algorithms that compute the canonical basis $\mathcal{L}$ of $\mathbb{K}$ directly also compute $\operatorname{Int}\mathbb{K}$ as a side product.
- $|\operatorname{Int}\mathbb{K}|$ can be exponentially larger than $\mathcal{L}$.

Computing the Canonical Basis

- Known exact algorithms that compute the canonical basis $\mathcal{L}$ of $\mathbb{K}$ directly also compute $\operatorname{Int}\mathbb{K}$ as a side product.
- $|\operatorname{Int}\mathbb{K}|$ can be exponentially larger than $\mathcal{L}$.

- Probably approximately computation (PAC) of the canonical basis has been considered in (Borchmann *et al.* 2017, 2020).
  - The approach is based on the query-learning algorithm from (Angluin *et al.* 1992).
- We slightly generalise this approach.

Horn Distance

Let

$\mathbb{K} = (G, M, I)$ be a formal context;

$\mathcal{D}$ be a probability distribution over subsets of M;

$\mathcal{L}$ be an implication set over M.

Horn Distance

Let

$\mathbb{K} = (G, M, I)$ be a formal context;

$\mathcal{D}$ be a probability distribution over subsets of M;

$\mathcal{L}$ be an implication set over M.

Definition (Horn $\mathcal{D}$-distance between $\mathcal{L}$ and $\mathbb{K}$)

$$\mathrm{dist}^{\mathcal{D}}(\mathcal{L}, \mathbb{K}) := \Pr_{\mathcal{D}}(A \in \mathrm{Mod}\,\mathcal{L} \triangle \mathrm{Int}\,\mathbb{K})$$

Here, $X \triangle Y$ is the symmetric difference between X and Y.

Horn Distance

Let

$\mathbb{K} = (G, M, I)$ be a formal context;

$\mathcal{D}$ be a probability distribution over subsets of M;

$\mathcal{L}$ be an implication set over M.

Definition (Horn $\mathcal{D}$-distance between $\mathcal{L}$ and $\mathbb{K}$)

$$\operatorname{dist}^{\mathcal{D}}(\mathcal{L}, \mathbb{K}) := \Pr_{\mathcal{D}}(A \in \operatorname{Mod} \mathcal{L} \mathbin{\triangle} \operatorname{Int} \mathbb{K})$$

Here, $X \mathbin{\triangle} Y$ is the symmetric difference between X and Y.

Definition (Strong Horn $\mathcal{D}$-distance between $\mathcal{L}$ and $\mathbb{K}$)

$$\operatorname{dist}^{\mathcal{D}}_{\mathrm{STRONG}}(\mathcal{L}, \mathbb{K}) := \Pr_{\mathcal{D}}(\mathcal{L}(A) \neq A'')$$

Horn Approximation

Let

$\mathbb{K} = (G, M, I)$ be a formal context;

$\mathcal{D}$ be a probability distribution over subsets of M;

$\mathcal{L}$ be an implication set over M.

Definition

$\mathcal{L}$ is an ϵ-Horn $\mathcal{D}$-approximation of $\mathbb{K} = (G, M, I)$ for $0 < \epsilon < 1$ if

$$\text{dist}^{\mathcal{D}}(\mathcal{L}, \mathbb{K}) \leq \epsilon.$$

Strong Horn Approximation

Let

$\mathbb{K} = (G, M, I)$ be a formal context;

$\mathcal{D}$ be a probability distribution over subsets of M;

$\mathcal{L}$ be an implication set over M.

Definition

$\mathcal{L}$ is an ϵ-strong Horn $\mathcal{D}$-approximation of $\mathbb{K} = (G, M, I)$ for $0 < \epsilon < 1$ if

$$\mathrm{dist}^{\mathcal{D}}_{\mathrm{STRONG}}(\mathcal{L}, \mathbb{K}) \leq \epsilon.$$

Strong Horn Approximation

Let

$\mathbb{K} = (G, M, I)$ be a formal context;

$\mathcal{D}$ be a probability distribution over subsets of M;

$\mathcal{L}$ be an implication set over M.

Definition

$\mathcal{L}$ is an ϵ-strong Horn $\mathcal{D}$-approximation of $\mathbb{K} = (G, M, I)$ for $0 < \epsilon < 1$ if

$$\text{dist}^{\mathcal{D}}_{\text{STRONG}}(\mathcal{L}, \mathbb{K}) \leq \epsilon.$$

With $\mathcal{D}$ being the uniform distribution, we get the notions of approximation from (Borchmann *et al.* 2020).

Upper Approximation

$\mathbb{K} = (G, M, I)$ be a formal context;

$\mathcal{D}$ be a probability distribution over subsets of M;

$\mathcal{L}$ be an implication set over M.

Definition

An ϵ- (ϵ-strong) Horn $\mathcal{D}$-approximation $\mathcal{L}$ of $\mathbb{K} = (G, M, I)$ is an upper approximation if all implications of $\mathcal{L}$ are valid in $\mathbb{K}$, i.e., $\operatorname{Int} \mathbb{K} \subseteq \operatorname{Mod} \mathcal{L}$.

Here, we work with upper approximations only.

Probably Approximately Correct Algorithm

Given

- a formal context $\mathbb{K} = (G, M, I)$;
- an oracle $EX_{\mathcal{D}}$ generating subsets of M according to probability distribution $\mathcal{D}$;
- $0 < \epsilon < 1$;
- $0 < \delta < 1$;

Probably Approximately Correct Algorithm

Given

- a formal context $\mathbb{K} = (G, M, I)$;
- an oracle $EX_{\mathcal{D}}$ generating subsets of M according to probability distribution $\mathcal{D}$;
- $0 < \epsilon < 1$;
- $0 < \delta < 1$;

find, with probability $\geq 1 - \delta$,

- an upper ϵ- (ϵ-strong) Horn $\mathcal{D}$-approximation $\mathcal{L}$ of $\mathbb{K}$

Probably Approximately Correct Algorithm

Given

- a formal context $\mathbb{K} = (G, M, I)$;
- an oracle $EX_{\mathcal{D}}$ generating subsets of M according to probability distribution $\mathcal{D}$;
- $0 < \epsilon < 1$;
- $0 < \delta < 1$;

find, with probability $\geq 1 - \delta$,

- an upper ϵ- (ϵ-strong) Horn $\mathcal{D}$-approximation $\mathcal{L}$ of $\mathbb{K}$

in time polynomial in $|G|$, $|M|$, the size of the canonical basis of $\mathbb{K}$, $1/\epsilon$, and $1/\delta$.

Probably Approximately Correct Algorithm

- Based on the query-learning algorithm from (Angluin *et al.* 1992),
  - which is shown in (Arias and Balcázar 2011) to produce the canonical basis.
- First described in (Kautz *et al.* 1995) for the case of uniform distribution.
- Introduced into FCA in (Borchmann *et al.* 2017, 2020).

Probably Approximately Correct Algorithm

- Maintain a set $\mathcal{L}$ of valid implications.

Probably Approximately Correct Algorithm

- Maintain a set $\mathcal{L}$ of valid implications.
- At each iteration, check if $\mathcal{L}$ is an ϵ-approximation of $\mathbb{K}$.

Probably Approximately Correct Algorithm

- Maintain a set $\mathcal{L}$ of valid implications.
- At each iteration, check if $\mathcal{L}$ is an ϵ-approximation of $\mathbb{K}$.
- If not, obtain a counterexample $X \in \operatorname{Mod} \mathcal{L} \setminus \operatorname{Int} \mathbb{K}$

Probably Approximately Correct Algorithm

- Maintain a set $\mathcal{L}$ of valid implications.
- At each iteration, check if $\mathcal{L}$ is an ϵ-approximation of $\mathbb{K}$.
- If not, obtain a counterexample $X \in \operatorname{Mod} \mathcal{L} \setminus \operatorname{Int} \mathbb{K}$ or, in the case of strong approximation, X such that $\mathcal{L}(X) \subsetneq X''$.

Probably Approximately Correct Algorithm

- Maintain a set $\mathcal{L}$ of valid implications.
- At each iteration, check if $\mathcal{L}$ is an ϵ-approximation of $\mathbb{K}$.
- If not, obtain a counterexample $X \in \operatorname{Mod}\mathcal{L} \setminus \operatorname{Int}\mathbb{K}$ or, in the case of strong approximation, X such that $\mathcal{L}(X) \subsetneq X''$.
- Use $\mathcal{L}(X)$ to either refine an implication from $\mathcal{L}$ or add a new implication to $\mathcal{L}$.

Probably Approximately Correct Algorithm

- Maintain a set $\mathcal{L}$ of valid implications.
- At each iteration, check if $\mathcal{L}$ is an ϵ-approximation of $\mathbb{K}$.
- If not, obtain a counterexample $X \in \operatorname{Mod} \mathcal{L} \setminus \operatorname{Int} \mathbb{K}$ or, in the case of strong approximation, X such that $\mathcal{L}(X) \subsetneq X''$.
- Use $\mathcal{L}(X)$ to either refine an implication from $\mathcal{L}$ or add a new implication to $\mathcal{L}$.

Probably Approximately Correct Algorithm

- Maintain a set $\mathcal{L}$ of valid implications.
- At each iteration, check if $\mathcal{L}$ is an ϵ-approximation of $\mathbb{K}$.
- If not, obtain a counterexample $X \in \operatorname{Mod} \mathcal{L} \setminus \operatorname{Int} \mathbb{K}$ or, in the case of strong approximation, X such that $\mathcal{L}(X) \subsetneq X''$.
- Use $\mathcal{L}(X)$ to either refine an implication from $\mathcal{L}$ or add a new implication to $\mathcal{L}$.

- Use $EX_{\mathcal{D}}$ for a number of times to try to generate a counterexample X.

Probably Approximately Correct Algorithm

- Maintain a set $\mathcal{L}$ of valid implications.
- At each iteration, check if $\mathcal{L}$ is an ϵ-approximation of $\mathbb{K}$.
- If not, obtain a counterexample $X \in \operatorname{Mod} \mathcal{L} \setminus \operatorname{Int} \mathbb{K}$ or, in the case of strong approximation, X such that $\mathcal{L}(X) \subsetneq X''$.
- Use $\mathcal{L}(X)$ to either refine an implication from $\mathcal{L}$ or add a new implication to $\mathcal{L}$.

- Use $EX_{\mathcal{D}}$ for a number of times to try to generate a counterexample X.
- At ith iteration,

$$q_i(\epsilon, \delta) = \left\lceil \log_{1-\epsilon} \frac{\delta}{i(i+1)} \right\rceil$$

attempts are sufficient (Yarullin and Obiedkov 2020).

Frequency-Aware Approximation

Definition

An ϵ- (ϵ-strong) Horn $\mathcal{D}$-approximation $\mathcal{L}$ of $\mathbb{K} = (G, M, I)$ is a frequency-aware ϵ- (ϵ-strong) Horn approximation of $\mathbb{K}$ if $\mathcal{D} = \mathcal{D}_f$, where

$$\Pr_{\mathcal{D}_f}(A) = \frac{|A'|}{\sum_{B \subseteq M} |B'|}$$

for $A \subseteq M$.

- Favours frequent implications.
- Completely disregards implications describing incompatibilities between attributes.
- Is much more accurate w.r.t. well-supported implications than approximations based on the uniform distribution.

Sampling Attribute Subsets According to $\mathcal{D}_f$

Boley *et al.* 2011

1. Select $g \in G$ according to

$$\Pr(g) = \frac{2^{|g'|}}{\sum_{h \in G} 2^{|h'|}}.$$

2. Select a subset of g' uniformly at random.

Computing Frequency-Aware Approximations

- Use the algorithm for computing ϵ- (ϵ-strong) Horn $\mathcal{D}$-approximations.
- Simulate $EX_{\mathcal{D}}$ with Boley *et al.*'s algorithm.

- Obtain a total–polynomial time randomised algorithm for computing frequency-aware approximations.

The Quality of Approximation

- Under the uniform distribution, we guarantee, with probability $\geq 1 - \delta$,

$$\frac{|\operatorname{Mod} \mathcal{L}| - |\operatorname{Int} \mathbb{K}|}{2^{|M|}} \leq \epsilon.$$

The Quality of Approximation

- Under the uniform distribution, we guarantee, with probability $\geq 1 - \delta$,

$$\frac{|\operatorname{Mod} \mathcal{L}| - |\operatorname{Int} \mathbb{K}|}{2^{|M|}} \leq \epsilon.$$

- $\operatorname{Mod} \mathcal{L}$ contains at most $\epsilon 2^{|M|}$ extra subsets in addition to those in $\operatorname{Int} \mathbb{K}$.

The Quality of Approximation

- Under the uniform distribution, we guarantee, with probability $\geq 1 - \delta$,

$$\frac{|\,\mathsf{Mod}\,\mathcal{L}| - |\,\mathsf{Int}\,\mathbb{K}|}{2^{|M|}} \leq \epsilon.$$

- $\mathsf{Mod}\,\mathcal{L}$ contains at most $\epsilon 2^{|M|}$ extra subsets in addition to those in $\mathsf{Int}\,\mathbb{K}$.
- Still, $\mathsf{Mod}\,\mathcal{L}$ can be much larger than $\mathsf{Int}\,\mathbb{K}$.

The Quality of Approximation

- Under the uniform distribution, we guarantee, with probability $\geq 1-\delta$,

$$\frac{|\operatorname{Mod}\mathcal{L}| - |\operatorname{Int}\mathbb{K}|}{2^{|M|}} \leq \epsilon.$$

- $\operatorname{Mod}\mathcal{L}$ contains at most $\epsilon 2^{|M|}$ extra subsets in addition to those in $\operatorname{Int}\mathbb{K}$.
- Still, $\operatorname{Mod}\mathcal{L}$ can be much larger than $\operatorname{Int}\mathbb{K}$.

Definition (Quality Factor)

For $A \subseteq M$,

$$QF(\mathcal{L}, \mathbb{K}, A) = \frac{|\operatorname{Int}\mathbb{K} \cap \mathfrak{P}(A)|}{|\operatorname{Mod}\mathcal{L} \cap \mathfrak{P}(A)|}.$$

In the experiments, we measure QF for A consisting of $\alpha|M|$ most frequent attributes of M, where α is $1/4$ for real-world data sets and $1/2$ for artificial data sets.

Experimental Evaluation

- C++ implementation at https://github.com/saurabh18213/Implication-Basis
- Parallelised search for
  - a counterexample through sampling and
  - an implication to be refined

- Intel Xeon E5-2650 v3 @ 2.30GHz
- 20 cores and up to 40 threads

Datasets

Context	Attributes	Objects	Canonical basis	Intents	Density
Census	122	48842	71787	248846	0.08
nom10shuttle	97	43500	810	2931	0.10
Mushroom	119	8124	2323	238710	0.19
Connect	114	7222	86583	50468988	0.38
inter10shuttle	178	43500	936	38199148	0.46
Chess	75	3196	73162	930851337	0.49
Example 1 ($n = 5$)	25	3125	5	28629152	0.80
Example 1 ($n = 6$)	36	46656	6	62523502210	0.83
Example 2 ($n = 10$)	21	30	1024	2038103	0.92
Example 2 ($n = 15$)	31	45	32768	2133134741	0.95

Datasets

Example 1 (Ganter and Obiedkov 2016)

- $M = M_1 \cup \cdots \cup M_n$ — M_is are pairwise disjoint.
- $|M_i| = n$ for all $i \leq n$.
- Object intents g' are all possible attribute combinations with $|g' \cap M_i| = n - 1$ for all $i \leq n$.
- n^n objects with intents of the same size.

Datasets

Example 1 (Ganter and Obiedkov 2016)

- $M = M_1 \cup \cdots \cup M_n$ — M_is are pairwise disjoint.
- $|M_i| = n$ for all $i \leq n$.
- Object intents g' are all possible attribute combinations with $|g' \cap M_i| = n - 1$ for all $i \leq n$.
- n^n objects with intents of the same size.
- The $(2^n - 1)^n + 1$ concept intents are sets that do not contain any of M_i.
- Canonical basis:

$$\{M_i \to M \mid i \leq n\}$$

- n implications for n^2 attributes and n^n objects.

Datasets

Example 2 (Kuznetsov 2004)

<table>
<tr><th></th><th>m_0</th><th>$m_1, \dots, m_n$</th><th>$m_{n+1}, \dots, m_{2n}$</th></tr>
<tr><td>g_1</td><td rowspan="3"></td><td rowspan="3">$\neq$</td><td rowspan="3">$\neq$</td></tr>
<tr><td>$\vdots$</td></tr>
<tr><td>g_n</td></tr>
<tr><td>g_{n+1}</td><td>$\times$</td><td rowspan="5" colspan="2">$\neq$</td></tr>
<tr><td>$\vdots$</td><td>$\vdots$</td></tr>
<tr><td>$\vdots$</td><td>$\vdots$</td></tr>
<tr><td>$\vdots$</td><td>$\vdots$</td></tr>
<tr><td>g_{3n}</td><td>$\times$</td></tr>
</table>

- The canonical basis consists of 2^n implications:

$$\{\{m_{i_1}, \dots, m_{i_n}\} \to \{m_0\} \mid i_j \in \{j, j+n\}\}$$

Default Parameter Values

Context	ϵ	δ
Census	0.1	0.1
nom10shuttle	0.1	0.1
Mushroom	0.1	0.1
Connect	0.1	0.1
inter10shuttle	0.1	0.1
Chess	0.1	0.1
Example 1 ($n = 5$)	0.01	0.1
Example 1 ($n = 6$)	0.01	0.1
Example 2 ($n = 10$)	0.01	0.1
Example 2 ($n = 15$)	0.001	0.1

Comparing Approximations

Uniform: generate subsets of M uniformly at random;

Frequent: generate subsets of M according to $\mathcal{D}_f$;

Both:

- first, generate subsets of M uniformly at random;
- if, at some iteration, all attempts fail, redo them generating subsets according to $\mathcal{D}_f$;
- use $\mathcal{D}_f$ from this point on.

Comparing Approximations

Runtime in seconds

	ϵ-strong Horn approximation			ϵ-Horn approximation		
Data set	Uniform	Frequent	Both	Uniform	Frequent	Both
Census	0.18	1451.64	1184.10	0.16	5.02	0.21
nom10shuttle	0.15	0.73	0.71	0.14	0.43	0.44
Mushroom	0.11	1.89	1.95	0.06	0.16	0.14
Connect	0.14	307.51	307.10	0.07	0.08	0.07
inter10shuttle	0.59	6.77	6.47	0.58	0.60	0.60
Chess	0.07	167.96	169.77	0.04	0.04	0.03

- On real-worlds datasets, Frequent is slower than Uniform.
- Strong approximation takes more time.

Comparing Approximations

The number of implications

	ϵ-strong Horn approximation			ϵ-Horn approximation			Basis
Data set	Uniform	Frequent	Both	Uniform	Frequent	Both	
Census	48	20882	19111	41	1210	71	71787
nom10shuttle	76	201	201	76	137	146	810
Mushroom	95	577	593	7	72	59	2323
Connect	120	10774	10730	7	9	9	86583
inter10shuttle	172	446	430	171	171	171	936
Chess	64	6514	6542	48	48	48	73162

- On real-worlds datasets, Frequent results in more implications than Uniform.
- Strong approximation contains more implications.

Comparing Approximations

The quality factor

	ϵ-strong Horn approximation			ϵ-Horn approximation		
Data set	Uniform	Frequent	Both	Uniform	Frequent	Both
Census	0.0003	0.0184	0.0180	0.0003	0.0014	0.0004
nom10shuttle	0.0004	0.0695	0.0613	0.0004	0.0157	0.0208
Mushroom	0.0004	0.1454	0.1482	0.0001	0.0032	0.0014
Connect	0.9979	0.9979	0.9979	0.0001	0.0016	0.0016
inter10shuttle	0.4900	0.5533	0.5429	0.4900	0.4900	0.4900
Chess	0.6927	1.0000	0.9830	0.6927	0.6927	0.6927

- On real-worlds datasets, Frequent usually results in a higher QF value than Uniform.
- Strong approximation is usually stronger.

Comparing Approximations

Data set	ϵ-strong Horn approximation: Uniform	ϵ-strong Horn approximation: Frequent	ϵ-strong Horn approximation: Both	ϵ-Horn approximation: Uniform	ϵ-Horn approximation: Frequent	ϵ-Horn approximation: Both	Basis
	Runtime in seconds						
Example 1-5	0.03	0.03	0.04	0.03	0.03	0.04	
Example 1-6	0.31	0.27	0.36	0.31	0.29	0.37	
	The number of Implications						
Example 1-5	5	0	5	5	0	5	5
Example 1-6	6	0	6	6	0	6	6
	The quality factor						
Example 1-5	1	0.9692	1	1	0.9692	1	
Example 1-6	1	0.9844	1	1	0.9844	1	

- Frequent is worse than Uniform, since all non-trivial implications have zero support.
- No difference for stronger approximation, since the closures of all non-closed sets are equal to M.

Comparing Approximations

	ϵ-strong Horn approximation			ϵ-Horn approximation			Basis
Data set	Uniform	Frequent	Both	Uniform	Frequent	Both	
	Runtime in seconds						
Example 2-10	0.27	0.17	0.27	0.21	0.19	0.26	
Example 2-15	96.72	74.64	108.77	83.31	75.12	115.81	
	The number of Implications						
Example 2-10	357	269	340	321	262	347	1024
Example 2-15	7993	6813	8375	7612	6970	8424	32768
	The quality factor						
Example 2-10	1	1	1	1	1	1	
Example 2-15	1	1	1	1	1	1	

- Frequent is similar to Uniform, since all non-trivial implications have non-zero support and all implications from the canonical basis have support $n/(2n+1)$.

NB! The quality factor is meaningless here, since any selection of $|M|/2$ most frequent attributes contains at most one subset that is not closed in the context.

Varying ϵ

Time in seconds

Data set	0.3	0.2	0.1	0.05	0.01
Census	0.19	37.63	1184.10	2345.26	2336.88
nom10shuttle	0.44	0.47	0.71	0.82	1.43
Mushroom	0.82	1.27	1.95	2.75	5.03
Connect	308.69	307.54	307.10	306.97	307.44
inter10shuttle	4.41	5.34	6.47	7.91	12.72
Chess	169.23	169.50	169.77	168.04	168.99
Example 1 ($n = 5$)	0.02	0.02	0.03	0.03	0.04
Example 1 ($n = 6$)	0.23	0.23	0.29	0.30	0.36
Example 2 ($n = 10$)	0.002	0.002	0.002	0.01	0.27
Example 2 ($n = 15$)	0.002	0.002	0.002	0.002	0.63

Varying ϵ

The number of implications

Data set	0.3	0.2	0.1	0.05	0.01	Basis
Census	49	2865	19111	26257	26253	71787
nom10shuttle	136	149	201	231	303	810
Mushroom	349	440	593	749	1036	2323
Connect	10790	10746	10730	10735	10759	86583
inter10shuttle	356	383	430	479	582	936
Chess	6563	6572	6542	6537	6578	73162
Example 1 ($n = 5$)	3	4	5	5	5	5
Example 1 ($n = 6$)	1	2	6	6	6	6
Example 2 ($n = 10$)	1	2	4	28	340	1024
Example 2 ($n = 15$)	0	0	0	1	422	32768

Data set	0.3	0.2	0.1	0.05	0.01
Census	0.0004	0.0034	0.0180	0.0208	0.0208
nom10shuttle	0.0090	0.0140	0.0613	0.1017	0.1753
Mushroom	0.0382	0.0692	0.1482	0.2726	0.4504
Connect	0.9979	0.9979	0.9979	0.9979	0.9979
inter10shuttle	0.4956	0.5202	0.5429	0.6451	0.8910
Chess	0.9981	1.0000	0.9830	0.9963	1.0000
Example 1 ($n = 5$)	0.9692	0.9815	1.0000	1.0000	1.0000
Example 1 ($n = 6$)	0.9844	0.9875	0.9969	1.0000	1.0000
Example 2 ($n = 10$)	1.0000	1.0000	1.0000	1.0000	1.0000
Example 2 ($n = 15$)	1.0000	1.0000	1.0000	1.0000	1.0000

Runtime in seconds

ϵ-strong, Both

Data set	1 thread	40 threads	QF	NextClosure	LinCbO
Census	29608.00	1184.10	0.0180	522	177
nom10shuttle	3.34	0.71	0.0613	1.25	0.44
Mushroom	25.92	1.95	0.1482	49	10.8
Connect	6239.75	307.10	0.9979	23 310	19 420
inter10shuttle	42.52	6.47	0.5429	19 223	16 698
Chess	1955.12	169.77	0.9830	325 076	234 309
Example 1-5	0.05	0.04	1.0000	384	65
Example 1-6	0.55	0.36	1.0000	–	–
Example 2-10	0.22	0.27	1.0000	5.94	2.8
Example 2-15	84.97	108.77	1.0000	203 477	29 710

Conclusion

DONE:

- An approximation of the canonical basis biased towards its frequent part.
- A randomised algorithm that computes this approximation with desired probability.
- On dense contexts, the algorithm is (usually) significantly faster than NEXT CLOSURE–based algorithms computing the entire basis, while providing an approximation of decent quality.

TODO:

- Various strategies for parallelising the algorithm.
- Approximations biased towards interestingness measures other than support.