Scalable test data generation from multidimensional models

emina torlak
emina@eecs.berkeley.edu
## a brief intro to multidimensional analysis

<table>
<thead>
<tr>
<th>person</th>
<th>education</th>
<th>height (m)</th>
<th>weight (kg)</th>
<th>BMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>H.S.</td>
<td>1.85</td>
<td>90</td>
<td>26</td>
</tr>
<tr>
<td>2</td>
<td>B.Sc.</td>
<td>1.70</td>
<td>83</td>
<td>29</td>
</tr>
<tr>
<td>3</td>
<td>M.Sc.</td>
<td>1.80</td>
<td>85</td>
<td>26</td>
</tr>
<tr>
<td>4</td>
<td>H.S.</td>
<td>1.64</td>
<td>75</td>
<td>26</td>
</tr>
<tr>
<td>5</td>
<td>B.A.</td>
<td>1.55</td>
<td>60</td>
<td>23</td>
</tr>
<tr>
<td>6</td>
<td>M.Eng.</td>
<td>1.72</td>
<td>65</td>
<td>22</td>
</tr>
<tr>
<td>7</td>
<td>Ph.D.</td>
<td>1.59</td>
<td>61</td>
<td>24</td>
</tr>
</tbody>
</table>
a brief intro to multidimensional analysis

dimensions have structure

<table>
<thead>
<tr>
<th>person</th>
<th>education</th>
<th>height (m)</th>
<th>weight (kg)</th>
<th>BMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>H.S.</td>
<td>1.85</td>
<td>90</td>
<td>26</td>
</tr>
<tr>
<td>2</td>
<td>B.Sc.</td>
<td>1.70</td>
<td>83</td>
<td>29</td>
</tr>
<tr>
<td>3</td>
<td>M.Sc.</td>
<td>1.80</td>
<td>85</td>
<td>26</td>
</tr>
<tr>
<td>4</td>
<td>H.S.</td>
<td>1.64</td>
<td>75</td>
<td>26</td>
</tr>
<tr>
<td>5</td>
<td>B.A.</td>
<td>1.55</td>
<td>60</td>
<td>23</td>
</tr>
<tr>
<td>6</td>
<td>M.Eng.</td>
<td>1.72</td>
<td>65</td>
<td>22</td>
</tr>
<tr>
<td>7</td>
<td>Ph.D.</td>
<td>1.59</td>
<td>61</td>
<td>24</td>
</tr>
</tbody>
</table>
a brief intro to multidimensional analysis

dimensions have structure

each fact associates a point in the person × education space with a set of measures
what are the average BMIs of males and females at different education levels?

<table>
<thead>
<tr>
<th>person</th>
<th>education</th>
<th>height (m)</th>
<th>weight (kg)</th>
<th>BMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>H.S.</td>
<td>1.85</td>
<td>90</td>
<td>26</td>
</tr>
<tr>
<td>2</td>
<td>B.Sc.</td>
<td>1.70</td>
<td>83</td>
<td>29</td>
</tr>
<tr>
<td>3</td>
<td>M.Sc.</td>
<td>1.80</td>
<td>85</td>
<td>26</td>
</tr>
<tr>
<td>4</td>
<td>H.S.</td>
<td>1.64</td>
<td>75</td>
<td>26</td>
</tr>
<tr>
<td>5</td>
<td>B.A.</td>
<td>1.55</td>
<td>60</td>
<td>23</td>
</tr>
<tr>
<td>6</td>
<td>M.Eng.</td>
<td>1.72</td>
<td>65</td>
<td>22</td>
</tr>
<tr>
<td>7</td>
<td>Ph.D.</td>
<td>1.59</td>
<td>61</td>
<td>24</td>
</tr>
</tbody>
</table>
what are the average BMIs of males and females at different education levels?

<table>
<thead>
<tr>
<th>person</th>
<th>education</th>
<th>height (m)</th>
<th>weight (kg)</th>
<th>BMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>H.S.</td>
<td>1.85</td>
<td>90</td>
<td>26</td>
</tr>
<tr>
<td>2</td>
<td>B.Sc.</td>
<td>1.70</td>
<td>83</td>
<td>29</td>
</tr>
<tr>
<td>3</td>
<td>M.Sc.</td>
<td>1.80</td>
<td>85</td>
<td>26</td>
</tr>
<tr>
<td>4</td>
<td>H.S.</td>
<td>1.64</td>
<td>75</td>
<td>26</td>
</tr>
<tr>
<td>5</td>
<td>B.A.</td>
<td>1.55</td>
<td>60</td>
<td>23</td>
</tr>
<tr>
<td>6</td>
<td>M.Eng.</td>
<td>1.72</td>
<td>65</td>
<td>22</td>
</tr>
<tr>
<td>7</td>
<td>Ph.D.</td>
<td>1.59</td>
<td>61</td>
<td>24</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>avg. BMI</th>
<th>male</th>
<th>female</th>
</tr>
</thead>
<tbody>
<tr>
<td>basic</td>
<td>26</td>
<td>26</td>
</tr>
<tr>
<td>college</td>
<td>29</td>
<td>23</td>
</tr>
<tr>
<td>graduate</td>
<td>26</td>
<td>23</td>
</tr>
</tbody>
</table>
operations: what-if analysis

if the average BMIs of all groups were to decrease by 2 points, what would be the effect on the average weight of each group?

<table>
<thead>
<tr>
<th>person</th>
<th>education</th>
<th>height (m)</th>
<th>weight (kg)</th>
<th>BMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>H.S.</td>
<td>1.85</td>
<td>90</td>
<td>26</td>
</tr>
<tr>
<td>2</td>
<td>B.Sc.</td>
<td>1.70</td>
<td>83</td>
<td>29</td>
</tr>
<tr>
<td>3</td>
<td>M.Sc.</td>
<td>1.80</td>
<td>85</td>
<td>26</td>
</tr>
<tr>
<td>4</td>
<td>H.S.</td>
<td>1.64</td>
<td>75</td>
<td>26</td>
</tr>
<tr>
<td>5</td>
<td>B.A.</td>
<td>1.55</td>
<td>60</td>
<td>23</td>
</tr>
<tr>
<td>6</td>
<td>M.Eng.</td>
<td>1.72</td>
<td>65</td>
<td>22</td>
</tr>
<tr>
<td>7</td>
<td>Ph.D.</td>
<td>1.59</td>
<td>61</td>
<td>24</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>level</th>
<th>avg. BMI</th>
<th>male</th>
<th>female</th>
</tr>
</thead>
<tbody>
<tr>
<td>basic</td>
<td>26</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>college</td>
<td>29</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>graduate</td>
<td>26</td>
<td>23</td>
<td></td>
</tr>
</tbody>
</table>
operations: what-if analysis

if the average BMIs of all groups were to decrease by 2 points, what would be the effect on the average weight of each group?

<table>
<thead>
<tr>
<th>person</th>
<th>education</th>
<th>height (m)</th>
<th>weight (kg)</th>
<th>BMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>H.S.</td>
<td>1.85</td>
<td>83</td>
<td>24</td>
</tr>
<tr>
<td>2</td>
<td>B.Sc.</td>
<td>1.70</td>
<td>77</td>
<td>27</td>
</tr>
<tr>
<td>3</td>
<td>M.Sc.</td>
<td>1.80</td>
<td>79</td>
<td>24</td>
</tr>
<tr>
<td>4</td>
<td>H.S.</td>
<td>1.64</td>
<td>70</td>
<td>26</td>
</tr>
<tr>
<td>5</td>
<td>B.A.</td>
<td>1.55</td>
<td>55</td>
<td>23</td>
</tr>
<tr>
<td>6</td>
<td>M.Eng.</td>
<td>1.72</td>
<td>62</td>
<td>21</td>
</tr>
<tr>
<td>7</td>
<td>Ph.D.</td>
<td>1.59</td>
<td>58</td>
<td>23</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>avg. BMI</th>
<th>male</th>
<th>female</th>
</tr>
</thead>
<tbody>
<tr>
<td>basic</td>
<td>24</td>
<td>25</td>
</tr>
<tr>
<td>college</td>
<td>27</td>
<td>21</td>
</tr>
<tr>
<td>graduate</td>
<td>24</td>
<td>21</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>avg. weight</th>
<th>male</th>
<th>female</th>
</tr>
</thead>
<tbody>
<tr>
<td>basic</td>
<td>83</td>
<td>70</td>
</tr>
<tr>
<td>college</td>
<td>77</td>
<td>55</td>
</tr>
<tr>
<td>graduate</td>
<td>79</td>
<td>60</td>
</tr>
</tbody>
</table>
implementation challenges

- Fact tables and dimension instances are huge
- Interactive response times achieved via precomputed views

<table>
<thead>
<tr>
<th>person</th>
<th>education</th>
<th>height (m)</th>
<th>weight (kg)</th>
<th>BMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>H.S.</td>
<td>1.85</td>
<td>90</td>
<td>26</td>
</tr>
<tr>
<td>2</td>
<td>B.Sc.</td>
<td>1.70</td>
<td>83</td>
<td>29</td>
</tr>
<tr>
<td>3</td>
<td>M.Sc.</td>
<td>1.80</td>
<td>85</td>
<td>26</td>
</tr>
<tr>
<td>4</td>
<td>H.S.</td>
<td>1.64</td>
<td>75</td>
<td>26</td>
</tr>
<tr>
<td>5</td>
<td>B.A.</td>
<td>1.55</td>
<td>60</td>
<td>23</td>
</tr>
<tr>
<td>6</td>
<td>M.Eng.</td>
<td>1.72</td>
<td>65</td>
<td>22</td>
</tr>
<tr>
<td>7</td>
<td>Ph.D.</td>
<td>1.59</td>
<td>61</td>
<td>24</td>
</tr>
</tbody>
</table>
implementation challenges

- view selection problem is inapproximable unless $P = NP$
- must test at scale, with representative data, to find the best implementation

<table>
<thead>
<tr>
<th>person</th>
<th>education</th>
<th>height (m)</th>
<th>weight (kg)</th>
<th>BMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>H.S.</td>
<td>1.85</td>
<td>90</td>
<td>26</td>
</tr>
<tr>
<td>2</td>
<td>B.Sc.</td>
<td>1.70</td>
<td>83</td>
<td>29</td>
</tr>
<tr>
<td>3</td>
<td>M.Sc.</td>
<td>1.80</td>
<td>85</td>
<td>26</td>
</tr>
<tr>
<td>4</td>
<td>H.S.</td>
<td>1.64</td>
<td>75</td>
<td>26</td>
</tr>
<tr>
<td>5</td>
<td>B.A.</td>
<td>1.55</td>
<td>60</td>
<td>23</td>
</tr>
<tr>
<td>6</td>
<td>M.Eng.</td>
<td>1.72</td>
<td>65</td>
<td>22</td>
</tr>
<tr>
<td>7</td>
<td>Ph.D.</td>
<td>1.59</td>
<td>61</td>
<td>24</td>
</tr>
</tbody>
</table>
implementation challenges

- view selection problem is inapproximable unless P = NP
- must test at scale, with representative data, to find the best implementation

<table>
<thead>
<tr>
<th>person</th>
<th>education</th>
<th>height (m)</th>
<th>weight (kg)</th>
<th>BMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>H.S.</td>
<td>1.85</td>
<td>90</td>
<td>26</td>
</tr>
<tr>
<td>2</td>
<td>B.Sc.</td>
<td>1.70</td>
<td>83</td>
<td>29</td>
</tr>
<tr>
<td>3</td>
<td>M.Sc.</td>
<td>1.80</td>
<td>85</td>
<td>26</td>
</tr>
<tr>
<td>4</td>
<td>H.S.</td>
<td>1.64</td>
<td>75</td>
<td>26</td>
</tr>
<tr>
<td>5</td>
<td>B.A.</td>
<td>1.55</td>
<td>60</td>
<td>23</td>
</tr>
<tr>
<td>6</td>
<td>M.Eng.</td>
<td>1.72</td>
<td>65</td>
<td>22</td>
</tr>
<tr>
<td>7</td>
<td>Ph.D.</td>
<td>1.59</td>
<td>61</td>
<td>24</td>
</tr>
</tbody>
</table>

where to get representative data?
- live data is unavailable in many settings (industrial & academic)
- existing data generation tools cannot instantiate general multidimensional models
TestBlox: generating test data from models

multidimensional model

- dimension schema (with constraints)
- measure schema (with constraints)

TestBlox

- solver
- parallel instance generator

instance

- dimension instances
- measure tables

report

(un)satisfied constraints

soft constraints

- dimension distributions
- measure distributions
TestBlox: generating test data from models

multidimensional model
- schema modeling language that captures most modeling patterns
- NP-hard but with a practical P-time fragment

TestBlox
- solver for P-time fragment with soft constraints
- parallel instance generation architecture

instance
- dimension instances
- measure tables

report
- (un)satisfied constraints

soft constraints
- simple interface for specifying soft constraints

‣ schema modeling language that captures most modeling patterns
‣ NP-hard but with a practical P-time fragment

‣ simple interface for specifying soft constraints
implicit constraints

- **disjointness**: member sets of different categories are disjoint
- **global strictness**:  
  - rollup relations are functional  
  - every union of compositions of rollup relations is functional: \( b.e \cup b.d.g \cup b.c.f \)
dimension model: schema & constraints

<table>
<thead>
<tr>
<th>implicit constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>disjointness</strong>: member sets of different categories are disjoint</td>
</tr>
<tr>
<td><strong>global strictness</strong>:</td>
</tr>
<tr>
<td>‣ rollup relations are functional</td>
</tr>
<tr>
<td>‣ every union of compositions of rollup relations is functional: $b.e \cup b.d.g \cup b.c.f$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>explicit constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>‣ cardinality</td>
</tr>
<tr>
<td>‣ rollup</td>
</tr>
<tr>
<td>‣ exclusive rollup</td>
</tr>
<tr>
<td>‣ drilldown</td>
</tr>
<tr>
<td>‣ exclusive drilldown</td>
</tr>
</tbody>
</table>

| $| gender | = 2$ |
| $| country | \leq 196$ |

person $\rightarrow (gender \lor city)$
city $\rightarrow (state \odot province \odot country)$

person $\leftarrow gender$
(state $\odot province) \leftarrow country$
dimension model: schema & constraints

- **implicit constraints**
  - **disjointness**: member sets of different categories are disjoint
  - **global strictness**:
    - rollup relations are functional
    - every union of compositions of rollup relations is functional: \( b.e \cup b.d.g \cup b.c.f \)

- **explicit constraints**
  - **cardinality**
  - **rollup**
  - **exclusive rollup**
  - **drilldown**
  - **exclusive drilldown**

- There are exactly 2 genders and at most 196 countries

- person \( \rightarrow (gender \lor city) \)

- city \( \rightarrow (state \odot province \odot country) \)

- person \( \leftarrow \) gender

- (state \( \odot \) province) \( \leftarrow \) country

- | gender | = 2
- | country | \( \leq 196 \)
dimension model: schema & constraints

**implicit constraints**
- **disjointness**: member sets of different categories are disjoint
- **global strictness**:
  - rollup relations are functional
  - every union of compositions of rollup relations is functional: \( b.e \cup b.d.g \cup b.c.f \)

**explicit constraints**
- cardinality
- rollup
- exclusive rollup
- drilldown
- exclusive drilldown

| gender | = 2  
| country | \( \leq 196 \)  

person \( \rightarrow \) (gender \( \vee \) city)  

person \( \leftarrow \) gender  

(city \( \rightarrow \) (state \( \odot \) province \( \odot \) country))  

person \( \leftarrow \) gender  

(state \( \odot \) province) \( \leftarrow \) country  

every person rolls-up to a gender or a city (or both)
dimension model: schema & constraints

**implicit constraints**
- **disjointness**: member sets of different categories are disjoint
- **global strictness**: rollup relations are functional
  - every union of compositions of rollup relations is functional: \( b.e \cup b.d.g \cup b.c.f \)

**explicit constraints**
- **cardinality**
- **rollup**
- **exclusive rollup**
- **drilldown**
- **exclusive drilldown**

| gender | = 2  
| country | ≤ 196  

person ➔ (gender ∨ city)  

city ➔ (state ⊗ province ⊗ country)

person ← gender  

(state ⊗ province) ← country

every city rolls-up to either a state or a province or a country
**dimension model: schema & constraints**

| gender | = 2
| country | ≤ 196

person $\rightarrow$ (gender $\lor$ city)
city $\rightarrow$ (state $\odot$ province $\odot$ country)

person $\leftarrow$ gender
(state $\odot$ province) $\leftarrow$ country

**implicit constraints**
- **disjointness**: member sets of different categories are disjoint
- **global strictness**:
  - rollup relations are functional
  - every union of compositions of rollup relations is functional: $b.e \cup b.d.g \cup b.c.f$

**explicit constraints**
- cardinality
- rollup
- exclusive rollup
- drilldown
- exclusive drilldown

each gender has at least one person rolling-up to it

**dimension model**

- country
- state
- province
- gender
- city
- person

edges:
- f
- g
- e
- c
- d
- a
- b
**dimension model: schema & constraints**

**implicit constraints**
- **disjointness:** member sets of different categories are disjoint
- **global strictness:**
  - rollup relations are functional
  - every union of compositions of rollup relations is functional: \( b.e \cup b.d.g \cup b.c.f \)

**explicit constraints**
- cardinality
- rollup
- exclusive rollup
- drilldown
- exclusive drilldown

| gender | = 2 |
| country | \( \leq 196 \) |

person \( \rightarrow \) (gender \( \lor \) city)

city \( \rightarrow \) (state \( \circ \) province \( \circ \) country)

person \( \leftarrow \) gender

(state \( \circ \) province) \( \leftarrow \) country

every country has either states or provinces rolling-up to it (but not both)
statistical (soft) constraints: scale & shape

```plaintext
person { size = 1000 }

gender ➞ person ➞ city { 
  source = .8
  target = .5
  distribution = normal(30, 4)
}

scale

› the person set should have 1000 members
```
statistical (soft) constraints: scale & shape

- person { size = 1000 }
- person → city {
  - source = .8
  - target = .5
  - distribution = normal(30, 4)
}
- 80% of the persons roll-up to 50% of the cities
- each city has, on average, 30 persons rolling-up to it, with a standard deviation of 4
instantiating dimension models is NP-hard

sample 3-SAT formula

(c1) \( \neg v_1 \lor v_2 \lor v_3 \)
(c2) \( v_1 \lor \neg v_2 \lor \neg v_3 \)
(c3) \( \neg v_1 \lor \neg v_2 \lor \neg v_3 \)
instantiating dimension models is NP-hard

sample 3-SAT formula

(c1) \( \neg v_1 \lor v_2 \lor v_3 \)
(c2) \( v_1 \lor \neg v_2 \lor \neg v_3 \)
(c3) \( \neg v_1 \lor \neg v_2 \lor \neg v_3 \)

\( v_1 = \text{true} \)
\( v_2 = \text{true} \)
\( v_3 = \text{false} \)
polynomial ("regular") fragment

the schema is tiered

if A is an ancestor of D, then
\[ \text{high}(A) \geq \text{low}(D) \]

- high(A) is the upper bound on A implied by cardinality constraints
- low(D) is the lower bound on D implied by cardinality constraints

and weakly exclusive

every exclusive constraint on a set of edges is the only constraint on those edges

| gender | = 2
| country | \( \leq 196 \)

person \( \rightarrow \) (gender \( \lor \) city)
city \( \rightarrow \) (state \( \odot \) province \( \odot \) country)

person \( \leftarrow \) gender
(state \( \odot \) province) \( \leftarrow \) country
polynomial ("regular") fragment

The schema is tiered

If A is an ancestor of D, then
high(A) ≥ low(D)

- high(A) is the upper bound on A implied by cardinality constraints
- low(D) is the lower bound on D implied by cardinality constraints

And weakly exclusive

Every exclusive constraint on a set of edges is the only constraint on those edges

| gender | = 2
| country | ≤ 196

person → (gender ∨ city)
city → (state ⊙ province ⊙ country)
person ← gender
(state ⊙ province) ← country
polynomial ("regular") fragment

the schema is tiered

if A is an ancestor of D, then
\[ \text{high}(A) \geq \text{low}(D) \]

- high(A) is the upper bound on A implied by cardinality constraints
- low(D) is the lower bound on D implied by cardinality constraints

and weakly exclusive

every exclusive constraint on a set of edges is the only constraint on those edges

| gender | = 2
| country | \leq 196

person \rightarrow (\text{gender} \lor \text{city})

\text{city} \rightarrow (\text{state} \odot \text{province} \odot \text{country})

person \leftarrow \text{gender}

\text{(state} \odot \text{province}) \leftarrow \text{country}
TestBlox architecture

- **Dimension Constraints**
  - Dimension Solver
    - Solve a maximal weakly exclusive subset of edge constraints
- **Cardinality Constraints**
  - Cardinality Solver
    - Solve a maximal tiered subset of cardinality constraints
- **Soft Constraints**
  - Instance Generator
TestBlox architecture

dimension constraints

cardinality solver
solve a maximal tiered subset of cardinality constraints

dimension solver
solve a maximal weakly exclusive subset of edge constraints

instance generator
TestBlox architecture

dimension solver
solve a maximal weakly exclusive subset of edge constraints

cardinality solver
solve a maximal tiered subset of cardinality constraints

soft constraints

dimension constraints
cardinality solver in 3 steps

(1) find a maximal tiered subset of cardinality constraints
(2) generate a set of satisfiable inequalities from (1)
(3) solve (2) optimally w.r.t. soft constraints, in polynomial time

| gender | = 2 | person { size = 100 } |
| country | ≤ 196 | city { size = 10 } |
| city | ≤ 100 | state { size = 50 } |
| | | province { size = 30 } |
| | | country { size = 5 } |
cardinality solver (step 1)

(1) find a maximal tiered subset of cardinality constraints
(2) generate a set of satisfiable inequalities from (1)
(3) solve (2) optimally w.r.t. soft constraints, in polynomial time

| gender | = 2 | person { size = 100 } |
| country | ≤ 196 | city { size = 10 } |
| city | ≤ 100 | state { size = 50 } |
|        |     | province { size = 30} |
|        |     | country { size = 5 } |
cardinality solver (step 2)

(1) find a maximal tiered subset of cardinality constraints
(2) generate a set of satisfiable inequalities from (1)
(3) solve (2) optimally w.r.t. soft constraints, in polynomial time

1 ≤ customer ≤ ∞
2 ≤ gender ≤ 2
1 ≤ city ≤ 100
1 ≤ state ≤ ∞
1 ≤ province ≤ ∞
1 ≤ country ≤ 196

person ≥ gender
person ≥ city
city ≥ state
city ≥ province
city ≥ country
state ≥ country
province ≥ country

| gender | = 2
| country | ≤ 196
| city | ≤ 100

person { size = 100 }
city { size = 10 }
state { size = 50 }
province { size = 30 }
country { size = 5 }
cardinality solver (step 3)

(1) find a maximal tiered subset of cardinality constraints
(2) generate a set of satisfiable inequalities from (1)
(3) solve (2) optimally w.r.t. soft constraints, in polynomial time

| gender | = 2  
| country | ≤ 196  
| city | ≤ 100

person { size = 100 }
city { size = 10 }
state { size = 50 }
province { size = 30 }
country { size = 5 }

\[ \sum (v - \text{size}(v))^2 \]
dimension solver in 3 (more) steps

(1) decompose the schema graph into independent subgraphs
(2) generate and solve a set of satisfiable inequalities based on (1)
(3) use (1) and (2) to produce an instance closure for each edge

- person ➔ (gender ∨ city)
- city ➔ (state ⊙ province ⊙ country)
- person ← gender
- (state ⊙ province) ← country

rollup relations are functional.
every union of compositions of rollup relations is functional.
dimension solver (step 1)

(1) decompose the schema graph into independent subgraphs

(2) generate and solve a set of satisfiable inequalities based on (1)

(3) use (1) and (2) to produce an instance closure for each edge

person → (gender ∨ city)
city → (state ⊙ province ⊙ country)
person ← gender
(state ⊙ province) ← country

rollup relations are functional.
every union of compositions of rollup relations is functional.
**dimension solver (step 1)**

1. Decompose the schema graph into independent subgraphs
2. Generate and solve a set of satisfiable inequalities based on (1)
3. Use (1) and (2) to produce an instance closure for each edge

---

- **person** ➞ (gender ∨ city)
- **city** ➞ (state ⊙ province ⊙ country)
- **person** ← gender
- **(state ⊙ province)** ← country

Rollup relations are functional.

Every union of compositions of rollup relations is functional.
dimension solver (step 2)

- **person** $\rightarrow$ (gender $\lor$ city)
- **city** $\rightarrow$ (state $\odot$ province $\odot$ country)
- **person** $\leftarrow$ gender
- **(state $\odot$ province)** $\leftarrow$ country

rollup relations are functional.

every union of compositions of rollup relations is functional.

(1) decompose the schema graph into independent subgraphs
(2) generate and solve a set of satisfiable inequalities based on (1)
(3) use (1) and (2) to produce an instance closure for each edge

\[
\begin{align*}
0 & \leq \text{dom}(a) \leq 100 \\
0 & \leq \text{ran}(a) \leq 2 \\
\text{dom}(a) & \geq \text{ran}(a) \\
\text{dom}(a) & > 0 \lor \text{ran}(a) = 0 \\
& \ldots
\end{align*}
\]
**dimension solver (step 2)**

- **Graph**
  - `country` → `state` (5)
  - `country` → `province` (15)
  - `province` → `city` (20)
  - `city` → `person` (2)
  - `city` → `person` (20)

- **Inequalities**
  - $0 \leq \text{dom}(a) \leq 100$
  - $0 \leq \text{ran}(a) \leq 2$
  - $\text{dom}(a) \geq \text{ran}(a)$
  - $\text{dom}(a) > 0 \lor \text{ran}(a) = 0$
  - $\text{dom}(a) + \text{dom}(b) \geq 100$
  - $\text{dom}(a) \geq 2$

- **Steps**
  1. Decompose the schema graph into independent subgraphs
  2. Generate and solve a set of satisfiable inequalities based on (1)
  3. Use (1) and (2) to produce an instance closure for each edge

**Relations**
- `person` → `(gender ∨ city)`
- `city` → `(state ⊗ province ⊗ country)`
- `person` ← `gender`
- `(state ⊗ province)` ← `country`

Rollup relations are functional. Every union of compositions of rollup relations is functional.
dimension solver (step 2)

person ➞ (gender ∨ city)
city ➞ (state ⊕ province ⊕ country)

rollup relations are functional.
every union of compositions of rollup relations is functional.

100
20
15
2
2
5
20
100

(1) decompose the schema graph into independent subgraphs

(2) generate and solve a set of satisfiable inequalities based on (1)

(3) use (1) and (2) to produce an instance closure for each edge

0 ≤ dom(a) ≤ 100
0 ≤ ran(a) ≤ 2
dom(a) ≥ ran(a)
dom(a) > 0 ∨ ran(a) = 0

…

dom(a) + dom(b) ≥ 100
dom(a) ≥ 2
dom(c) + dom(d) + dom(e) = 20

ran(f) + ran(g) = 5
**dimension solver (step 2)**

1. decompose the schema graph into independent subgraphs
2. generate and solve a set of satisfiable inequalities based on (1)
3. use (1) and (2) to produce an instance closure for each edge

person → (gender ∨ city)
city → (state ∘ province ∘ country)
person ← gender
(state ∘ province) ← country

rollup relations are functional. every union of compositions of rollup relations is functional.

\[
0 \leq \text{dom}(a) \leq 100 \\
0 \leq \text{ran}(a) \leq 2 \\
\text{dom}(a) \geq \text{ran}(a) \\
\text{dom}(a) > 0 \lor \text{ran}(a) = 0 \\
\ldots \\
\text{dom}(a) + \text{dom}(b) \geq 100 \\
\text{dom}(a) \geq 2 \\
\text{dom}(c) + \text{dom}(d) + \text{dom}(e) = 20 \\
\text{ran}(f) + \text{ran}(g) = 5
\]

\[
\sum (\text{dom}(e) - \text{s}(e))^2 + (\text{ran}(e) - \text{t}(e))^2
\]

\[
\text{s}(e) = \text{source}(e) \ast |\text{src}(e)| \\
\text{t}(e) = \text{target}(e) \ast |\text{tgt}(e)|
\]
**dimension solver (step 3)**

1. Decompose the schema graph into independent subgraphs.
2. Generate and solve a set of satisfiable inequalities based on (1).
3. Use (1) and (2) to produce an instance closure for each edge.

---

person → (gender ∨ city)
city → (state ⊕ province ⊕ country)
person ← gender
(state ⊕ province) ← country

Rollup relations are functional.
every union of compositions of rollup relations is functional.
dimension solver (step 3)

(1) decompose the schema graph into independent subgraphs
(2) generate and solve a set of satisfiable inequalities based on (1)
(3) use (1) and (2) to produce an instance closure for each edge

person ➞ (gender ∨ city)
city ➞ (state ⊕ province ⊕ country)
person ➞ gender
(state ⊕ province) ➞ country

rollup relations are functional.
every union of compositions of rollup relations is functional.
**dimension solver (step 3)**

(1) decompose the schema graph into independent subgraphs

(2) generate and solve a set of satisfiable inequalities based on (1)

(3) use (1) and (2) to produce an instance closure for each edge

---

<table>
<thead>
<tr>
<th>c0</th>
<th>c0</th>
<th>c1</th>
<th>c1</th>
<th>c2</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>p4</td>
<td>p5</td>
<td>p6</td>
<td>p7</td>
<td>p8</td>
<td>p9</td>
</tr>
</tbody>
</table>

\[ \delta(b) = 3 \]

person ➞ (gender ∨ city)

city ➞ (state ⊙ province ⊙ country)

person ← gender

(state ⊙ province) ← country

rollup relations are functional.

every union of compositions of rollup relations is functional.
**dimension solver (step 3)**

(1) decompose the schema graph into independent subgraphs

(2) generate and solve a set of satisfiable inequalities based on (1)

(3) use (1) and (2) to produce an instance closure for each edge

---

person ➞ (gender ∨ city)
city ➞ (state ⊙ province ⊙ country)

person ← gender
(state ⊙ province) ← country

rollup relations are functional.
every union of compositions of rollup relations is functional.
(1) decompose the schema graph into independent subgraphs
(2) generate and solve a set of satisfiable inequalities based on (1)
(3) use (1) and (2) to produce an instance closure for each edge

person $\rightarrow$ (gender $\lor$ city)
city $\rightarrow$ (state $\odot$ province $\odot$ country)
person $\leftarrow$ gender
(state $\odot$ province) $\leftarrow$ country

rollup relations are functional.
every union of compositions of rollup relations is functional.
performance on 3 industrial applications

about the benchmarks

- 3 real applications from the retail domain
- randomly generated tiered, weakly exclusive constraints (worst case usage)
- randomly generated soft constraints
  - uniform, normal, exponential, and zipfian distributions
- chosen (as much as possible) to be feasible w.r.t. the model constraints
- .25, .5, .75 and 1GB of test data
scalability

- A1 Instances
  - Number of Threads
  - Generation Time (sec)
  - 0.25 GB
  - 0.5 GB
  - 0.75 GB
  - 1 GB

- A2 Instances
  - Number of Threads
  - Generation Time (sec)
  - 0.0003 GB

- A3 Instances
  - Number of Threads

- 1 GB ≈ 14,000,000 tuples
- 0.0003 GB ≈ 1,700 tuples
  - largest instance that can be generated in 60 secs with SAT
quality of generated data

Quality Indicators for A1

Quality Indicators for A2

Quality Indicators for A3

Number of Schema Edges

Ratio Range

(source, target, mean)
Practical sample data generation for multidimensional analysis applications

- a small DSL for specifying multidimensional constraints with a polynomially solvable fragment
- solver for instantiating multidimensional models, subject to soft requirements on the scale and shape of generated data

Prior work (highlights)

- multidimensional models generally focused on summarizability and expressiveness at the cost of analyzability
- sample data generation work focused on classic relational data models

Future work

- handling more features of other multidimensional models
- providing a richer interface for specifying/solving soft constraints