27 May 2015 GraphSM/TDBKDA 2015 Rome, Italy

On Statistics of Large-Scale RDF Datasets

Iztok Savnik
University of Primorska &
Institute Josef Stefan, Slovenia
iztok.savnik@upr.si

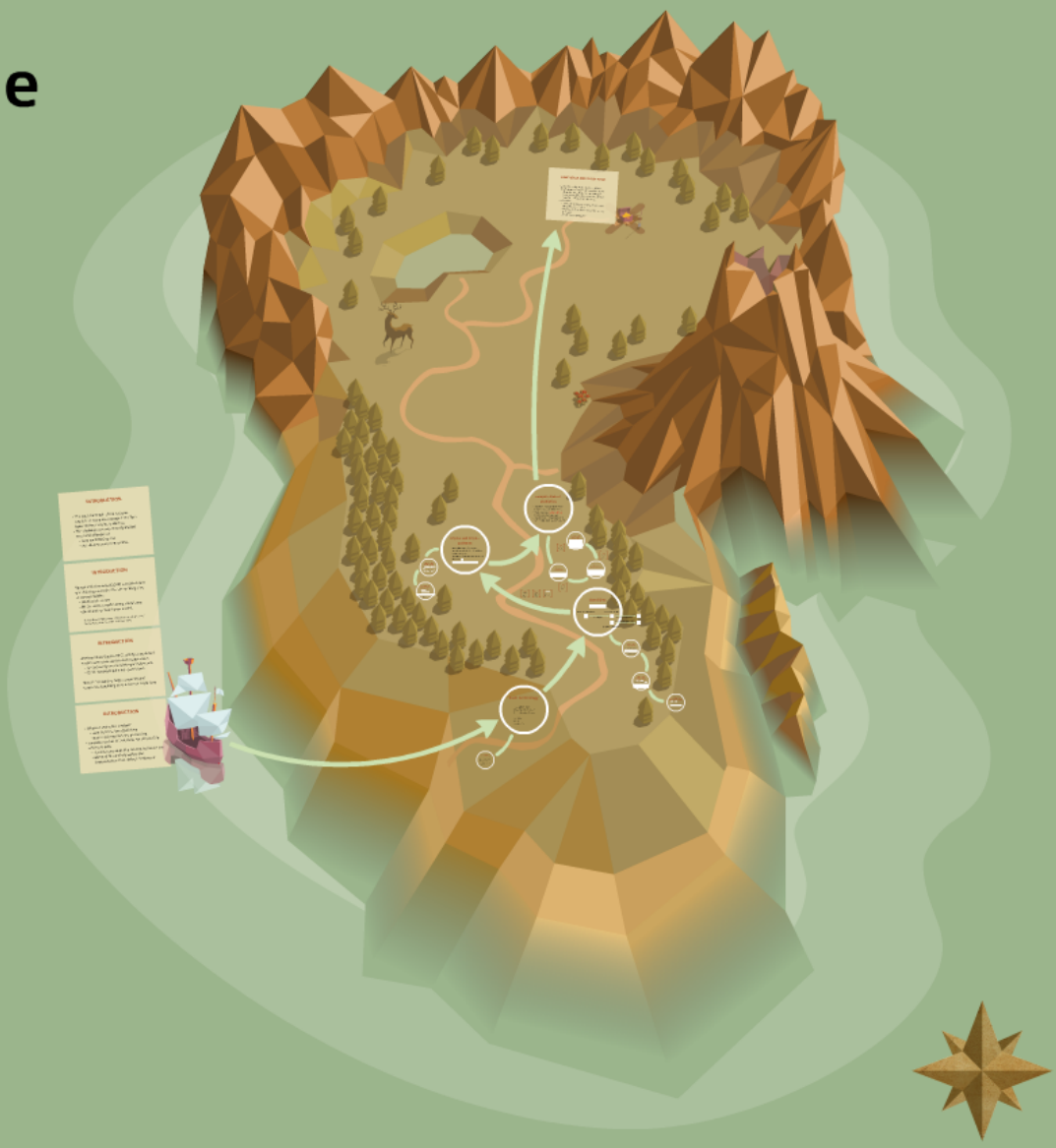
Kiyoshi Nitta
Yahoo JAPAN Research
Tokyo, Japan
knitta@yahoo-corp.jp

27 May 2015 GraphSM/DBKDA 2015 Rome, Italy

On Statistics of Large-Scale RDF Datasets

Iztok Savnik
University of Primorska &
Institute Jozef Stefan, Slovenia
iztok.savnik@upr.si

Kiyoshi Nitta
Yahoo JAPAN Research
Tokyo, Japan
knitta@yahoo-corp.jp



INTRODUCTION

- The need for triple-store systems capable to store and manage from Tera towards Peta triples is obvious.
- The challenge can be primarily divided into subchallenges of
  - data partitioning and
  - distributed query executions.

INTRODUCTION

- Where is statistics needed?
  - data distribution algorithms
  - query optimization and processing
- compute number of instances for all possible schema triples
  - demonstrate what the schema instances are
  - we need to carefully define the interpretation that defines "instances"

basic terminology

I --- set of URI-s

B --- the set of blanks

L --- the set of literals.

$S = I \cup B$

$P = I$

$O = I \cup B \cup L$

identifiers

$$\mathcal{I} = I \cup B$$

individual identifiers

class identifiers

$\mathcal{I}_i$ → rdf:type → $\mathcal{I}_c$ → rdfs:subClassO

$\mathcal{I}_p$ → rdfs:subProperty

property identifiers

triples and triple-patterns

- individual (ground) triples:
  include only individual identifiers
- abstract triples:
  include at least one class identifier
- top class $\top$

$$\forall i : i \in S \cup P \cup O \Rightarrow i \preceq \top$$

2.4:
ring

$= (s_2, p_2, o_2)$
$\times O$.

qual to t_2

$\preceq o_2$.

computation of statistics

- Statistics is computed for each triple t of a given triple-store.
- Triple t can be a schema triple that include the top class T.
- Statistics of t is the size of the set of natural interpretation of t.

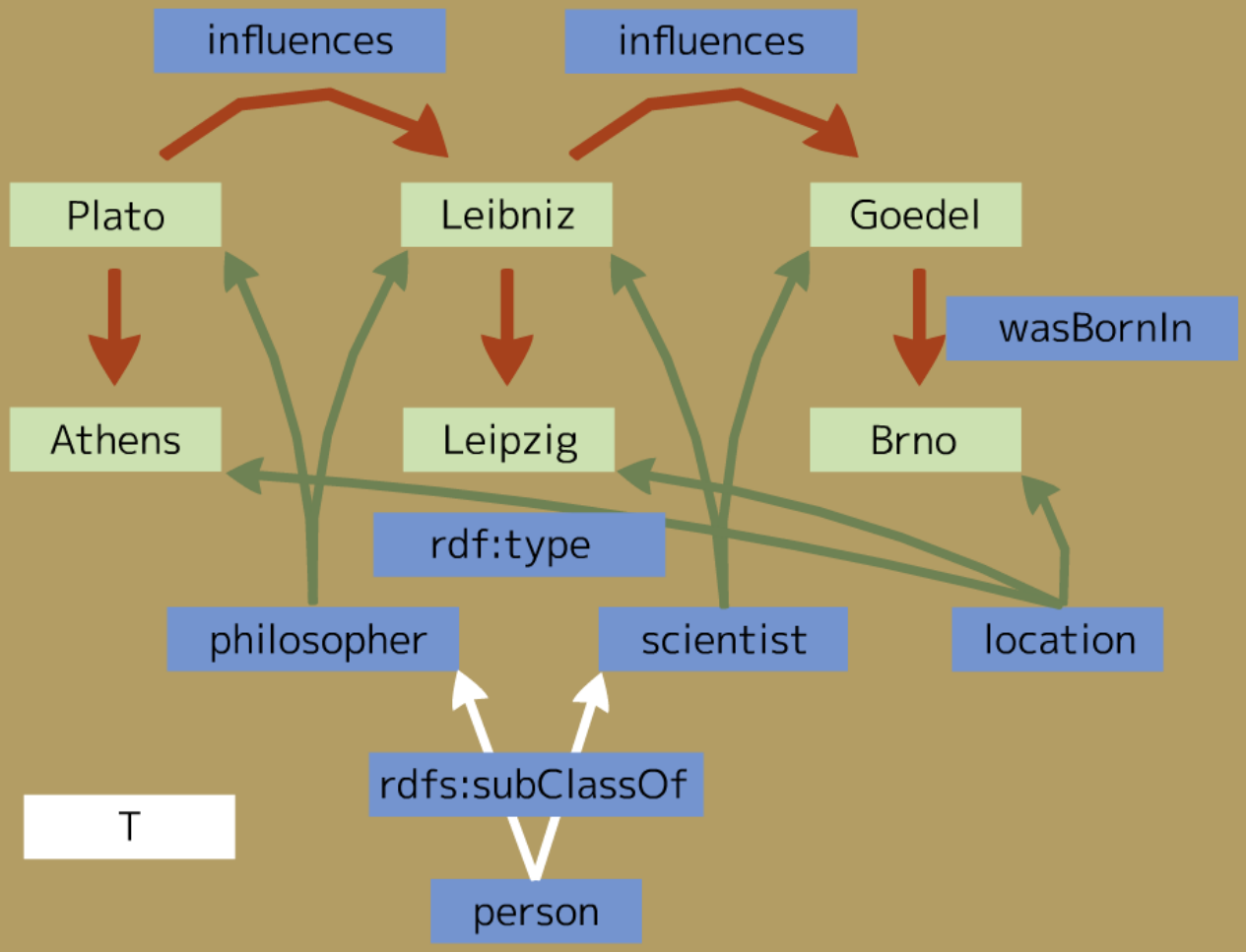
small example

(philosopher, rdfs:subClassOf, person)
(scientist, rdfs:subClassOf, person)

(Plato, rdf:type, philosopher)
(Leibniz, rdf:type, philosopher)
(Leibniz, rdf:type, scientist)
(Goedel, rdf:type, scientist)
(Athens, rdf:type, location)
(Leipzig, rdf:type, location)
(Brno, rdf:type, location)

(Plato, wasBornIn, Athens)
(Plato, influences, Leibniz)
(Leibniz, wasBornIn, Leipzig)
(Leibniz, influences, Goedel)
(Goedel, wasBornIn, Brno)

graph representation



natural interpretation

$$[\![(\text{scientist}, \top, \top)]\!]^*$$

7 triples:

(scientist, rdfs:subClassOf, person)
(Leibniz, rdf:type, philosopher)
(Leibniz, rdf:type, scientist)
(Goedel, rdf:type, scientist)
(Leibniz, wasBornIn, Leipzig)
(Leibniz, influences, Goedel)
(Goedel, wasBornIn, Brno)

function get_types_of

```
get_types_of(i: identifier) -> set_of_identifiers
begin
   if (i is individual identifier) then
      return { c | (i,rdf:type,c) IN  g };
   if (i is class identifier)
      return { i };
end;
```

small example

(Leibniz, wasBornIn, Leipzig)

```
get_types_of(Leipniz) ->
    {philosopher, scientist}
get_types_of(wasBornIn) ->
    {}
get_types_of(Leipzig) ->
    {location}
```

function transitive_closure

```
transitive_closure(a: set_of_identifiers)
      -> set_of_identifiers
begin
   repeat
      b = a;
      for each i IN b do
        extend a with c: (i,rdfs:subClassOf,c) IN g;
        extend a with c: (i,rdfs:subPropertyOf,c) IN g;
   until a == b;
end;
```

small example

(Leibniz, wasBornIn, Leipzig)

```
gs = {philosopher, scientist}
transitive_closure(gs) ->
    {philosopher, scientist, person, T}
get_types_of({}) ->
    {T}
get_types_of({location}) ->
    {location, T}
```

computation of schema triples

```
compute_statistics((s,p,o): triple)
begin
   gs = get_types_of(s);
   gs = transitive_closure(gs);

   gp = get_types_of(p);
   gp = transitive_closure(gp);

   go = get_types_of(o);
   go = transitive_closure(go);

   for each cs IN gs do
     for each cp IN gp do
       for each co IN go do
         increment couter of (cs,cp,co) by 1;
end;
```

small example

(Leibniz, wasBornIn, Leipzig)

t = (Leibniz, wasBornIn, Leipzig)
compute_statistics(t) increases
following schema triples:

(T, T, T)
(philosopher, T, T)
(scientist, T, T)
(person, T, T)

(T, T, location)
(philosopher, T, location)
(scientist, T, location)
(person, T, location)

computation of statistics

- Statistics is computed for each triple t of a given triple-store.
- Triple t can be a schema triple that include the top class T.
- Statistics of t is the size of the set of natural interpretation of t.

computation of schema triples

small example

small example

function transitive_closure

function get_types_of

CONCLUSION AND FUTURE WORK

- preliminary study for developing an efficient partitioning method of entity class-based model
  - algorithm for calculating triple statistics
  - trying to calculate for practical size of data (i.e. size of YAGO2 is 217M triples)
- future work
  - develop algorithms for mapping triple classes and triples to partitions
  - investigate efficiencies of triple partitioning processes
  - benchmark using big3store

27 May 2015 GraphSM/DBKDA 2015 Rome, Italy

On Statistics of Large-Scale RDF Datasets

Iztok Savnik
University of Primorska &
Institute Jozef Stefan, Slovenia
iztok.savnik@upr.si

Kiyoshi Nitta
Yahoo JAPAN Research
Tokyo, Japan
knitta@yahoo-corp.jp


Thank you!