Speeding up RDF aggregate discovery through sampling

Ioana Manolescu, Mirjana Mazuran
CEDAR Team
Inria and LIX (UMR 7161 and Ecole polytechnique), France
{ioana.manolescu, mirjana.mazuran}@inria.fr
Motivations

- RDF graphs can be **large, complex and heterogeneous**; finding out interesting information within them is challenging.

Foodista – Data about food, recipes, ingredients, cooking instruments: 1M triples, 88K distinct subjects
Motivations and Objectives

• One method to guide users is to **automatically find and show interesting aggregates**.

• Dagger [1] pioneered this approach, however it is inefficient due to the need to **evaluate many expensive aggregation queries**.

• We propose **Dagger+**: it builds upon Dagger and **leverages sampling** to speed up the evaluation of potentially interesting aggregates.

State of the art

• Automatic extraction of aggregates is one data exploration techniques.

• Most works assume a fixed relational schema, which is not available for RDF graphs [2][3][4]. Other recent works [5] consider graphs, but (unlike Dagger) assume a pre-computed graph cube with regular structure and suggest interesting portions in it.

• Dagger and Dagger+ start directly from RDF data, and, lacking schema information, automatically derive dimensions and measures that should lead to interesting insights.
Dagger
https://team.inria.fr/cedar/projects/dagger/

• Automatically recommends interesting **aggregation queries** over RDF graphs

“Number of authors of articles by year”
- **Fact**: article
- **Dimension**: year
- **Measure**: author
- **Aggregate**: COUNT(Measure)
- **Variance**: 0.064

“Number of books by publisher”
- **Fact**: book
- **Dimension**: publisher
- **Measure**: bookIdentifier
- **Aggregate**: COUNT(Measure)
- **Variance**: 0.014

Top-3 publishers:
1. Springer
2. Infix Verlag
3. Addison-Wesley
Dagger: concepts

• **Facts**: candidate facts are identified as either (i) all the resources of a given class, or (ii) all the resources having a certain set of properties.

• **Dimensions**: all properties that are frequent and have much fewer distinct values than there are facts. Moreover, \#ofProperty dimensions are derived.

• **Measures**: all properties that are frequent.

• **Aggregates**: depend on the type of the measure (e.g. COUNT for strings, AVG, MIN for numerical values, etc.)

• **Interestingness measure**: variance, skeweness, kurtosis.
Dagger: workflow and performances

- All subjects of type Article from dblp2012 (888.183 distinct articles, 20,132,491 triples)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>AvgTimeToFindAllTypes</td>
<td>2.927 sec</td>
</tr>
<tr>
<td>AvgTimeToGenerateCF</td>
<td>49.195 sec</td>
</tr>
<tr>
<td>AvgTimeToFindDimensions</td>
<td>1649.323 sec (=27 min) → (4.7%)</td>
</tr>
<tr>
<td>AvgTimeToFindMeasures</td>
<td>0.001 sec</td>
</tr>
<tr>
<td>AvgTimeToEvaluate</td>
<td>33431.49 sec (=9.3h !!) → (95%)</td>
</tr>
<tr>
<td>AvgTimeForRun</td>
<td>35140.54 sec</td>
</tr>
</tbody>
</table>

A total of 147 aggregates have been found:
- 3.8 minutes needed to evaluate each
- COUNT DISTINCT are the most challenging ones.
Dagger re-engineering

1. Ported on top of OntoSQL [6], a Java-based platform providing efficient **RDF storage, saturation, and query processing** algorithms.

2. Use of RDFQuotient **summary** [7] to find: (i) all the classes, (ii) for each class, the number of resources in it, and (iii) the properties these resources have. These questions are answered directly from the RDFQuotient reducing dimension enumeration time.

3. **Shared evaluations**: all candidate aggregates that share both dimension and measure are evaluated by a single SQL query.
Sampling strategies: CFSampling

• Introduction of two sampling strategies to trade some accuracy in aggregate selection for running time: CFSampling and Esampling:

• **CFSampling**: the candidate fact is sampled (draw n1 samples of size n2), and candidate dimensions and measures are found for each sample independently. For each of the n1 samples, candidate aggregates are generated, and their interestingness is evaluated on the sample.
Sampling strategies: ESampling

• Introduction of two sampling strategies to trade some accuracy in aggregate selection for running time: CFSampling and Esampling:

• **ESampling**: candidate dimensions and measures are computed on the whole candidate fact as in Dagger. Then, $n_1$ samples of size $n_2$ are drawn from the candidate fact, and the candidate aggregates are evaluated on the samples.
Accuracy of the results

• Each sample produces a list of ranked aggregates. To obtain a single, global Top-K we re-rank all the aggregates:
  • Rank the union of all aggregates across the samples.
  • Rank the intersection of all aggregates across the samples.

• We compute the accuracy of the Top-K as:
  • How many of the cubes found without sampling are in the union?
  • How many of the cubes found without sampling are in the intersection?

• Considering all subjects of type Article from dblp2012 (888.183 distinct articles, 20.132.491 triples)
Accuracy of the results

Accuracy of union

Run time (sec)

Top-5 with 2 samples
Accuracy: 1%

Top-5 with 3 samples
Accuracy: 1%

Top-5 with 4 samples
Accuracy: 1%

Top-5 with 5 samples
Accuracy: 1%

Top-10 with 2 samples
Accuracy: 1%

Top-10 with 3 samples
Accuracy: 1%

Top-10 with 4 samples
Accuracy: 1%

Top-10 with 5 samples
Accuracy: 1%

Top-20 with 2 samples
Accuracy: 1%

Top-20 with 3 samples
Accuracy: 1%

Top-20 with 4 samples
Accuracy: 1%

Top-20 with 5 samples
Accuracy: 1%

X CFSampling
+
ESampling
Future work: Spade
(joint work with Yanlei Diao and Pawel Guzewicz)

• Provide users with top-k most interesting multi-dimensional aggregates.

• Introduce new ways to find candidate fact sets, to enumerate and derive properties.

• Employ lattice-base computation techniques and online aggregation to efficiently compute candidate aggregates.
Thank you!

Bibliography: