PNNL-24435

Pacific Northwest
NATIONAL LABORATORY
Proudly Operated by Battelle Since 1965

HPC Analytics Support: Requirements for Uncertainty Quantification Benchmarks

May 2015

PR Paulson
S Purohit
LR Rodriguez

U.S. DEPARTMENT OF ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062;
ph: (865) 576-8401
fax: (865) 576-5728
email: reports@adonis.osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
email: orders@ntis.gov <http://www.ntis.gov/about/form.aspx>
Online ordering: http://www.ntis.gov

This document was printed on recycled paper.
(8/2010)

PNNL-24435

HPC Analytics Support: Requirements for Uncertainty Quantification Benchmarks

PR Paulson
LR Rodriguez
S Purohit

May 2015

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99352

Summary

This report outlines techniques for extending benchmark generation products so they support uncertainty quantification by benchmarked systems. We describe how uncertainty quantification requirements can be presented to candidate analytical tools supporting SPARQL. We describe benchmark data sets for evaluating uncertainty quantification and an approach for using our benchmark generator to produce data sets.

Acronyms and Abbreviations

RDF	Resource Description Framework
SQL	Structured Query Language
SPARQL	Simple Protocol RDF Query Language
UQ	Uncertainty Quantification

Contents

1.0 Introduction

We have developed a benchmark generation tool that provides large, complex Resource Description Framework (RDF) data sets for the evaluation of analytical tool kits. The benchmark generation tool is suitable for comparing tools on traditional metrics such as precision-recall and scaling behavior. However, for many real-world problems, it is infeasible or unsuitable to provide unambiguous classification labels for data instances. In the cyber-security domain, for example, only one out of every *N* network packets is collected for netflow analysis. In addition, netflows are often collected at only a few of the possible entry points of an enterprise. Both of these situations provide different sampling schemes representing the real-world network. When this is the case, it would be uninformative to state that there is no indication of a particular cyber event; it would be more valuable if some measure of confidence in that analysis result were provided. Where these types of uncertainty exist in a problem domain, analysis tools should provide query results that reflect the uncertainty inherent in the data being analyzed.

The procedures outlined in this report will provide data sets and test queries that allow a test of assertions regarding confidence and the precision of analytic results. We first give a brief overview of the goals and implementation of the benchmark generation tool and then discuss an approach for evaluating uncertainty quantification in future versions of the benchmark generator.

1.1 Benchmark Generation Tool

Our benchmark generation tool produces scalable data sets and target queries with matching query results for a specified target. The benchmark generator accepts a benchmark specification as input and produces a benchmark. The benchmark specification includes a schema for the generated data and statistical constraints on the data. The types of queries and characteristics of generated data sets vary according to the application domain intended for the benchmark. We have initially targeted the domains of cyber-security and social network analysis. The targeted query language is Simple Protocol RDF Query Language (SPARQL). SPARQL is a language similar to Structured Query Language (SQL) but is used against RDF data sets.

1.2 Terminology and Background

Uncertainty Quantification (UQ), in our work, is a method of specifying some metric of confidence in query results. We refer to the benchmark generator as the *generator*. The generator produces *benchmarks*. Benchmarks are used to evaluate *candidate systems* (or *candidates)*. We refer to a data set that can be most informatively interrogated with UQ as a *data set with uncertainty.* A data set that does not require UQ is referred to as a *ground-truth data set*.

2.0 Supporting Uncertainty in Query Evaluation

This section describes how we provide benchmark users with a method to model the uncertainty in the benchmark datasets. A formal description of components of the benchmarks produced by the generator is given, along with a metric for evaluating candidate tools using the generated benchmarks.

2.1 Enabling Uncertain Query Evaluation through Training

Query evaluation of data sets with uncertainty requires more background knowledge than is necessary for a benchmark on a ground-truth data set. We maintain that no meaningful quantification of uncertainty can occur except in reference to a body of knowledge. Two common sources of such domain knowledge are the insight gained through a common mathematical understanding of the domain and knowledge that is implicit in a set of training data sets. We intend to use the second option: the benchmark will provide the ability to generate pairs of data sets, one a ground-truth data set and the other a data set resulting from a transformation of the ground-truth data set that introduces a controlled level of uncertainty. The uncertainty models generated by candidate systems using the training data will be cross-validated against the benchmark data sets.

The choice of using training data sets over the common underlying model is based on the following considerations. In the scenario where one can gain insight into phenomena of interest through underlying models, we can imagine two ways to specify the benchmark:

1. We could provide candidate systems with details of the underlying domain that is being modeled by the uncertainty generator and the statistical assumptions.
2. We could provide a general description of the process that the uncertainty generator models and require systems to develop their own details about the uncertainty such a process introduces.

If the first option is chosen, then there is no need for the candidate systems to provide uncertainty quantification. If the statistical model of the uncertain process is adequate for candidate systems to successfully quantify uncertainty, then it is complete enough to analytically describe the uncertainty of query results. If we instead just describe the domain knowledge on which the uncertainty generator is based, any difference between the uncertainty quantification given by the candidate system and the "correct" UQ given by the benchmark generator is subject to the completeness of the domain models used by the candidate system and the benchmark generator. In many domains, this is a subjective issue that has no pre-determined resolution.

2.2 Formal Description

A *Data set for Uncertainty Quantification* (*DUQ*) consists of two RDF data sets: the benchmark set and the ground-truth set (see Table 2.1 for definitions). We have two suppositions: 1) the ground-truth set is drawn from the set of all valid data sets for the domain of the benchmark (notated ***D***), and 2) the benchmark set is a noisy and lossy sample of the ground-truth set that has been derived from the ground-truth data set by some process. For $D \in DUQ$, we refer to the benchmark data set as D_B and the ground-truth data set as D_E.

Table 2.1. Definitions

Definition	Semantics
$\boldsymbol{D}$	Set of all valid data sets for target domain
$\boldsymbol{DUQ}$	Set of all data sets for UQ for target domain
each $d \in \boldsymbol{DUQ}$ **is a pair** (g_d, b_d) where $g_d \in \boldsymbol{D}$	Every *DUQ* has a ground-truth data set and a benchmark data set. The ground-truth data set is a valid data set for the target domain. The benchmark data set b_d is assumed to have been generated from the ground-truth dataset g by some (possibly noisy, possibly lossy) transform representing some process in the problem domain.
$\boldsymbol{Q}$	Set of all valid SPARQL queries for target domain
$\boldsymbol{QUQ}$	Set of queries for UQs for target domain
each $q \in \boldsymbol{QUQ}$ **is a triple** (q_q, p_q, r_q) where $q_q \in \boldsymbol{Q}$ and p_1 and r_1 are both real numbers in the interval [0,1]	A query for UQ is a SPARQL query together with acceptable lower bonds for precision and recall.
$\boldsymbol{BUQ}$	Set of all benchmarks for UQ for the target domain
each $b \in \boldsymbol{BUQ}$ **is a triple** (Q_b, T_b, E_b) where $Q_b \subseteq \boldsymbol{QUD}, T_b \subseteq \boldsymbol{DUQ}$, **and** $E_b \subseteq \boldsymbol{DUQ}$	A benchmark consists of a set of queries and a set of data sets for training, and a set of data sets for UQ evaluation
$\boldsymbol{V}$	Set of SPARQL variables
QV: $\boldsymbol{Q} \rightarrow$ powerset($\boldsymbol{V}$)	The variables used by a SPARQL query
$\boldsymbol{QR}$	Set of query results for target domain
$\forall r, q, v \mid q \in \boldsymbol{Q} \wedge r \in \boldsymbol{QR}(q) \wedge v \in QV(q) \Rightarrow r : v \rightarrow \emptyset \cup$ set of values in target domain	Messy, but says that a SPARQL query result maps the variables of the query to a set of values.
$\forall q \mid q \in \boldsymbol{Q} \Rightarrow q : \boldsymbol{D} \rightarrow$ powerset($\boldsymbol{QR}(q)$)	A query acts on a data set to return a set of query results for the query
precision : $\boldsymbol{Q} \times \boldsymbol{DUQ} \rightarrow [0,1]$ $\text{precision}(q, d) \mapsto \left(\frac{\lvert q(g_d) \cap q(b_d)\rvert}{\lvert q(b_d)\rvert}\right)$ recall : $\boldsymbol{Q} \times \boldsymbol{DUQ} \rightarrow [0,1]$ $\text{recall}(q, d) \mapsto \left(\frac{\lvert q(b_d) \cap q(g_d)\rvert}{\lvert q(g_d)\rvert}\right)$ F: $\boldsymbol{QUQ} \times \boldsymbol{DUQ} \rightarrow [0,1]$ $\text{F}(q, d) \mapsto 2 \cdot \left(\frac{\text{precision}(q, d) \cdot \text{recall}(q, d)}{\text{precision}(q, d) + \text{recall}(q, d)}\right)$	Precision and recall are defined by comparing the query result against the (unknown) ground-truth data set to the query result against the benchmark data set

Definition	Semantics
$\text{precision}_{UQ}: [0,1] \times \boldsymbol{QUQ} \times \boldsymbol{DUQ} \rightarrow [0,1]$ $\text{precision}_{UQ}(p,q,d) \mapsto \frac{\min(\text{precision}(q,d),p)}{p}$ $\text{recall}_{UQ}: [0,1] \times \boldsymbol{QUQ} \times \boldsymbol{DUQ} \rightarrow [0,1]$ $\text{recall}_{UQ}(r,q,d) \mapsto \frac{\min(\text{recall}(q,d),r)}{r}$ $\text{F}_{UQ}: [0,1] \times [0,1] \times \boldsymbol{QUQ} \times \boldsymbol{DUQ} \rightarrow [0,1]$ $\text{F}_{UQ}(p,r,q,d) \mapsto 2 \cdot \left(\frac{\text{precision}_{UQ}(p,q,d) \cdot \text{recall}_{UQ}(r,q,d)}{\text{precision}_{UQ}(p,q,d) + \text{recall}_{UQ}(r,q,d)}\right)$	If allowable values for precision and recall are specified, then the result is scaled by the allowable value.

A *Query for Uncertainty Quantification* (QUQ) $q \in \boldsymbol{QUQ}$ is a triple (q_q, p_q, r_q) consisting of a SPARQL query q_q appropriate for the domain and a specification of the acceptable values for precision and recall, p_q and r_q respectively. Consider some $d \in \boldsymbol{DUQ}$, where $d = (g_d, b_d)$. Let $q(s)$ represent the result set when the query is executed against some arbitrary data set $s \in \boldsymbol{D}$. For all $d \in \boldsymbol{DUQ}$ and $q \in \boldsymbol{QUQ}$, the candidate system attempts to minimize the expected value of both

$$\text{precision}_{UQ}(p_q, q_q, d) - \text{precision}(q_q, d)$$

and

$$\text{recall}_{UQ}(r_q, q_q, d) - \text{recall}(q_q, d)$$

Here p_q and r_q represent, respectively, the lower limits on precision and recall that can be supported by the benchmark data set. Because precision and recall tend to move in opposite directions, the candidate system can increase overall performance by taking advantage of the imprecision allowed by the parameters for acceptable precision and recall.

2.3 Components of the Benchmark Specification

By using the benchmark specifications supported by the proposed software, vendors of candidate analytic tools can evaluate how their tools perform on an independently designed data set. A benchmark specification for uncertainty quantification consists of a set of target queries, a data set generator, a *benchmark transform*, and a *result evaluator*.

2.3.1 Specification of Queries and Uncertainty Quantification for RDF Data Sets

Neither the RDF nor SPARQL standards provide support for uncertainty quantification. We propose providing UQ requirements to candidate systems outside of the benchmark data sets and queries; each query will have the UQ requirements in addition to a standard SPARQL query. In this way, candidate systems will not be required to process ad hoc versions of RDF or SPARQL. We propose providing the UQ requirements for the benchmark along with each benchmark query. The constraints are determined by specifying the acceptable precision and recall for each benchmark query.

2.3.2 The Data Set Generator

The data set generator is used to generate ground-truth data sets. The result of applying the specifications transform to a ground-truth data set with uncertainty that we refer to as a *benchmark data set*. The data set generator operates in two modes. In *training* mode, it provides both a ground-truth data set and the benchmark data set resulting from applying the transform to the ground-truth data set. In evaluation mode, it produces only the ground-truth data set. The result evaluator accepts the query results for each query in the specification along with the query result sets produced by the candidate system and accumulates an overall score for performance under uncertainty for the candidate tool.

2.3.3 The Benchmark Transform

The benchmark transform accepts a ground-truth data set as input and produces a benchmark data set. The transform is intended to represent some domain-specific real-world process that prevents or distorts the generation of ground-truth data sets in the target domain. Because the real-world process being modeled may have additional parameters and inputs that are independent of the data set being transformed, it cannot be considered a map; the same input may produce different output data sets on different invocations.

2.3.4 The Result Evaluator

The result evaluator accepts the query results generated by the candidate system for each benchmark data set generated in benchmark mode and each benchmark query. It generates an accumulated score for the benchmark system based on the following:

- the results given by the candidate system compared to the results expected on the appropriate ground-truth data set.
- the UQ parameters required for the query specification.
- the number of training data sets consumed by the candidate system.

3.0 Proposed Evaluation of Candidate System

The query generator provides the following:

1. A data set transform capability that provides a noisy transform of a data set to a data set with uncertainty to be used for training and testing.
2. A data set generator that can generate a sequence of data sets for training and a sequence of data sets used for testing.
   a. In training mode, both the source data set and the transformed data set are provided.
   b. For testing mode, only the transformed data set is provided.
3. A set of benchmark mark queries $Q \subseteq \boldsymbol{QUQ}$. Note that each benchmark query provides query along with allowable values for precision and recall.
4. A result evaluator that accepts the candidate system's results for the most recently generated data set in testing mode and produces an accumulated benchmark score.

3.1 Scoring Candidate Systems

The overall score for a candidate on benchmark *B* can be given by calculating the mean of the F-measure (under uncertainty) over the set of benchmarks and scaling by $|T|$, the number of training sets used by the candidate.

$$S(B) = \frac{\sum_{(q,d)\in B} F(p_q, r_q, q_q, d)}{|T|}$$

4.0 Conclusion

We have presented a requirements specification for a general domain-specific benchmark generator that supports uncertainty quantification. UQ is supported through the generation of training data sets that can be used by candidate systems to create UQ models. The proposed design supports the evaluation of standard SPARQL SELECT queries under uncertainty with no additional query or syntax requirements for the SPARQL language.

Implementation of the proposed design requires the development of domain-specific transforms that can generate data sets with some degree of uncertainty from ground-truth data sets. The transforms should be designed to simulate data collection errors inherent in the specified domain.

Pacific Northwest
NATIONAL LABORATORY

Proudly Operated by **Battelle** *Since 1965*

U.S. DEPARTMENT OF
ENERGY

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99352
1-888-375-PNNL (7665)
www.pnnl.gov