



Estimating Characteristic Sets for RDF Dataset Profiles Based on Sampling

Lars Heling^(✉) and Maribel Acosta

Institute AIFB, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
{heling,acosta}@kit.edu

Abstract. RDF dataset profiles provide a formal representation of a dataset’s characteristics (features). These profiles may cover various aspects of the data represented in the dataset as well as statistical descriptors of the data distribution. In this work, we focus on the characteristic sets profile feature summarizing the characteristic sets contained in an RDF graph. As this type of feature provides detailed information on both the structure and semantics of RDF graphs, they can be very beneficial in query optimization. However, in decentralized query processing, computing them is challenging as it is difficult and/or costly to access and process all datasets. To overcome this shortcoming, we propose the concept of a profile feature estimation. We present sampling methods and projection functions to generate estimations which aim to be as similar as possible to the original characteristic sets profile feature. In our evaluation, we investigate the feasibility of the proposed methods on four RDF graphs. Our results show that samples containing 0.5% of the entities in the graph allow for good estimations and may be used by downstream tasks such as query plan optimization in decentralized querying.

1 Introduction

The characteristics of an RDF dataset can be formally represented as a set of features that compose a dataset profile. They support various applications such as entity linking, entity retrieval, distributed search and federated queries [5]. The features in a dataset profile can range from information on licensing, provenance to statistical characteristics of the dataset. Depending on the granularity of the statistics in a profile feature, the computation can be costly and require access to the entire dataset. For instance, characteristic sets are fine-grained statistic that is difficult to compute as it represents the set of predicates associated with each entity in a graph. Yet, several centralized and decentralized query engines rely on fine-grained dataset profiles for finding efficient query plans [7, 11, 13]. For example, *Odyssey* [13] leverages statistics on the characteristic sets of the datasets in the federation to estimate intermediate results when optimizing query plans.

In this work, we focus on the *Characteristic Sets Profile Feature* (CSPF), a statistical feature of RDF graphs that include the characteristic sets, their counts and the multiplicity of their predicates. There are three major reasons why we focus on the CSPF as a representative statistical characterization of RDF graphs. First, it implicitly captures structural features of the underlying graph, such as

the average out-degree, distinct number of subjects, and the set of predicates and their counts. Second, the characteristic sets contain semantic information on the entities represented in the graph and, thus, also implicitly reflect its schema. Lastly, the CSPF provides detailed insights into the predicate co-occurrences and, hence, it is well suited to be used by (decentralized) query engines for cardinality estimations and other downstream tasks. While the CSPFs are very beneficial for applications, their computation can be a challenging task. First, obtaining the entire dataset to compute this feature can be too difficult or costly. For example, in federated querying, data dumps are not always available and datasets can only be partially accessed via SPARQL endpoint or Triple Pattern Fragment servers. Second, the complexity of computing the characteristic sets for n triples is in $\Theta(n \cdot \log(n) + n)$ [11]. This may be an additional restriction for very large and constantly evolving datasets.

To overcome these limitations, we propose an approach that estimates accurate statistical profile features based on characteristic sets and that relies only on a sample of the original dataset. Given an RDF graph, we sample entities and compute their characteristic sets to build the CSPF of the sample. Then, we apply a projection function to extrapolate the feature observed in the sample to estimate the original graph's CSPF. It is important to consider that the estimations for the CSPF are very sensitive to the structure of the graph and the sample. Assume, for example, the following characteristic sets S_1 , S_2 and S_3 from YAGO and the number of associated subjects (*count*):

$$\begin{aligned} S_1 &= \{\text{rdfs:label, skos:prefLabel}\}, \text{count}(S_1) = 783,686, \\ S_2 &= \{\text{rdfs:label, skos:prefLabel, yago:isCitizenOf}\}, \text{count}(S_2) = 7,823, \\ S_3 &= \{\text{rdfs:label, skos:prefLabel, yago:isLocatedIn}\}, \text{count}(S_3) = 188,529. \end{aligned}$$

Even though S_1 differs only by a single predicate from S_2 and S_3 , S_1 occurs over 100 times more often than S_2 , but only about 4 times more often than S_3 . Hence, the main objective of our approach is avoiding misestimations when minor differences in characteristic sets lead to major changes in their *count* values. In summary, our contributions are

- a definition of statistical profile feature estimation and the associated problem,
- a formalization of Characteristic Sets Profile Feature (CSPF),
- an approach for generating profile feature estimations for CSPF, and
- an extensive experimental study examining the effectiveness of our approach on four well-known RDF datasets.

The remainder of this work is organized as follows. We present related work in Sect. 2 and introduce preliminaries in Sect. 3. We provide a formal problem definition in Sect. 4 and present our approach in Sect. 5. We evaluate our approach and discuss the results in Sect. 6. In Sect. 7, we draw our conclusions and point to future work.

2 Related Work

RDF Dataset Profiling. Capturing the characteristics of RDF datasets in *dataset profiles* has been studied in previous works. Ellefi et al. [5] present a taxonomy for dataset features represented in such profiles, which includes the categories general, qualitative, provenance, links, licensing, statistical, and dynamics. Regarding statistical features, different approaches have been proposed. Fernández et al. [6] aim to enable efficient RDF data structures, indexes, and compression techniques. To this end, the authors propose various metrics to characterize RDF datasets incorporating the particularities of RDF graphs. LODStats [3] is statement-stream-based approach that comprises 32 schema-level statistical criteria ranging from out-degree to the number of used classes. The ProLOD++ tool [1] supports profiling, mining and cleansing functionalities for RDF datasets. It enables a browser-based visualizations of domain level, schema level, and data level characteristics. ExpLOD [8] is a tool for generating summaries of RDF datasets combining textual labels and bisimulation contractions. These summaries include statistical information such as the class, predicate, and interlinking usage.

In addition to the existing statistical dataset profile feature covered in the literature, we propose and formalize a novel feature based on characteristic sets capturing both structural and semantic properties of the graph.

RDF Graph Sampling. The concept of sampling data from RDF graphs has been proposed for and applied to different problems. Debattista et al. [4] propose approximating specific quality metrics for large, evolving datasets based on samples. They argue that the exact computation of some quality metrics is too time-consuming and expensive and that an approximation of the quality is usually sufficient. They apply reservoir sampling and use the sampled triples to estimate the dereferenceability of URIs and links to external data providers. Rietveld et al. [16] aim to obtain samples that entail as many of the original answers to typical SPARQL queries. They rewrite the RDF graph to compute the network metrics PageRank, in-degree, and out-degree for the nodes. Based on the metrics, the *top-k* percent of all triples are selected as the sample of the graph. Soulet et al. [17] focus on analytical queries, which are typically too expensive to be executed directly over SPARQL endpoints. They propose separating the computation of such queries by executing them over random samples of the datasets. Due to the properties of the queries, the aggregation values converge with an increasing number of samples.

While in the first work sampling is applied to reduce the computational effort for quality metrics, they do not require the sampling method to capture the semantics of the dataset. The second approach aims to obtain a *relevant* sample which allows answering common queries and not a *representative* sample. Furthermore, the first two approaches require local access to the entire dataset for generating the sample. However, our work, similar to Soulet et al., is motivated by the restrictions that occur especially in decentralized scenarios with large, evolving datasets where it is not feasible to have local access to every dataset.

Different to the work by Soulet et al., we aim to sample the data in such a fashion that a single sample can be used to estimate the statistical profile feature and do not rely on the convergence properties induced by repeated sampling.

Network Sampling. Approaches for sampling large non-RDF graphs have also been proposed. Leskovec et al. [9] provide an overview of methods suitable for obtaining representative samples from large networks, considering three major categories for sampling: by selecting random nodes, by selecting random edges or by exploration. To assess the representativeness of the samples, static graph patterns are used, i.e., the distribution of structural network properties. The agreement for the graph pattern between the original graph and the samples is measured by the Kolmogorov-Smirnov D-statistic. No single best method emerges from their experimental study, but their performance depends on the specific application. Ribeiro et al. [15] focus on directed graphs and propose a directed unbiased random walk (DURW) algorithm. They model directed graphs as undirected graphs such that edges can also be traversed backwards when performing random walks. They incorporate random jumps to nodes with a probability that depends on the out-degree of the node as well as the weights of the edges. Ahmed et al. [2] identify two relevant models of computation when sampling from large networks. The *static model* randomly accesses any location in the graph. The *streaming model* merely allows for accessing edges in a sequential stream of edges. For the two models of computation, they propose methods based on the concept of graph induction and show that they preserve key network statistics of the graph, while achieving low space complexity and linear runtime complexity with respect to the edges in the sample.

In contrast to these methods, our approach aims to generate representative samples that allow for estimating statistic profile features of RDF datasets and therefore, the sampling methods need to be tailored to this task and the particularities of RDF graphs.

3 Preliminaries

The Resource Description Framework (RDF) defines a graph-based data model, where statements are represented as tuples (s, p, o) such that a subject s and an object o are connected nodes via a directed labeled edge by predicate p . The terms of the tuples can be Internationalized Resource Identifiers (IRIs), blank nodes, or literals. Assume the pairwise disjoint sets of IRIs I , blank nodes B , and literals L . A tuple $(s, p, o) \in (I \cup B) \times I \times (I \cup B \cup L)$ is an RDF triple. A set of RDF triples is denominated an RDF graph. The set of subjects in an RDF graph is often referred to as its *entities*.

The characteristics of RDF graphs can be summarized in statistic profiles. In traditional database theory, a statistic profile is a “complex object composed of quantitative descriptors” [10]. The quantitative descriptors cover different data characteristics: (i) central tendency (ii) dispersion, (iii) size, and (iv) frequency distribution. Such statistic profiles are used by query optimizers to devise an

efficient query plan. Similarly, in RDF, statistic profiles are also commonly used by centralized triple stores and federated query engines for query optimization [7, 11, 13]. Typically, the query optimizer uses the statistic profiles to estimate the join cardinalities of subqueries. In the following, we consider statistical profile features and follow the terminology by Ellefi et al. [5], denoting an RDF dataset profile as a formal representation of a set of dataset profile features.

Definition 1 (Profile Feature). *Given a RDF graph G , a profile feature $F(G)$ is defined as a characteristic describing a statistical feature F of the graph G .*

An example statistical profile feature of an RDF graph could be derived from its characteristic sets. The concept of characteristic sets for RDF graphs was presented by Neumann et al. [14] and captures the correlations between join predicates in an RDF graph. The idea of characteristic sets is describing semantically similar entities by grouping them according to the set of predicates the entities share. As a result, such a profile feature incorporates both statistical information on the data distribution as well as semantic information of the entities contained within an RDF graph.

Definition 2 (Characteristic Sets [14]). *The characteristic set of an entity s in an RDF graph G is given by: $S_C(s) := \{p \mid \exists o : (s, p, o) \in G\}$. Furthermore, for a given RDF graph G , the set of characteristic sets is given by $\mathcal{S}_C(G) := \{S_C(s) \mid \exists p, o : (s, p, o) \in G\}$.*

To obtain a statistical profile, the counts for the characteristic sets are computed as well as the multiplicities of the predicates within each characteristic set. These additional statistics are required by centralized triple stores as well as federated query engines to determine exact cardinality estimations for distinct queries as well as computing cardinality estimations for non-distinct queries [7, 11, 13, 14]. Similar to Neumann et al. [14], we define the count of a characteristic set $S = \{p_1, p_2, \dots\}$ in an RDF graph G as

$$\text{count}(S) := |\{s \mid \exists p, o : (s, p, o) \in G \wedge S_C(s) = S\}|. \quad (1)$$

In addition, in this work, we focus on the occurrences of predicates in characteristic sets by considering their mean multiplicity. The mean multiplicity is given by

$$\text{multiplicity}(p_i, S) := \frac{|\{(s, p_i, o) \mid (s, p_i, o) \in G \wedge S_C(s) = S\}|}{\text{count}(S)}. \quad (2)$$

In other words, for a given characteristic set, the multiplicity specifies how often each predicate occurs on average. For example, consider the characteristic set $S_1 = \{\text{rdf:type}, \text{rdfs:label}\}$ with $\text{count}(S_1) = 10$, $\text{multiplicity}(\text{rdfs:label}, S_1) = 1$ $\text{multiplicity}(\text{rdf:type}, S_1) = 2$. This indicates that 10 entities belong to S_1 and each of those entities has *exactly* one `rdfs:label` and *on average* two `rdf:type` predicates.

4 Problem Definition

As outlined in the introduction, it might be too difficult and/or costly to access an entire dataset for computing its profile features. For example, this might be the case for decentralized querying when the datasets may only be partially accessed via SPARQL endpoints or Triple Pattern Fragment servers. To address this problem, we propose the concept of *Profile Feature Estimation* which aims to estimate the original profile feature using limited data of the original dataset. The goal is generating a profile feature estimation which is as similar as possible to the original profile feature while requiring partial data only. More precisely, in this work, we focus on approaches that rely on a sample from the original RDF graph and employ a projection function to estimate the true profile feature. Hence, we define a profile feature estimation as follows.

Definition 3 (Profile Feature Estimation). *Given an RDF graph G , a projection function ϕ , a subgraph $H \subset G$, and the profile feature $F(\cdot)$, a profile feature estimation $\hat{F}(\cdot)$ for G is defined as*

$$\hat{F}(G) := \phi(F(H))$$

Ideally, a profile feature estimation is identical to the true profile feature. However, the similarity of such estimations to the original feature is influenced by the type of feature to be estimated, the subgraph H and the projection function ϕ . For example, given just a small subgraph, the estimation might be less accurate than for a larger subgraph, as it may cover more characteristics of the original graph. Therefore, the problem is finding an estimation based on a subgraph H and a projection function ϕ for the profile feature which maximizes the similarity to the profile feature of the original RDF graph.

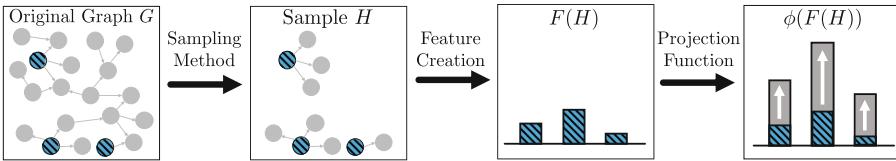


Fig. 1. Overview of the approach to estimate characteristic sets profile features.

Definition 4 (Profile Feature Estimation Problem). *Given an RDF graph G and a profile feature $F(\cdot)$, the problem of profile feature estimation is defined as follows. Determine a profile feature estimation $\hat{F}(\cdot)$, such that $\hat{F}(G) = \phi(F(H))$ and*

$$\max \delta(F(G), \hat{F}(G))$$

with $|H| \ll |G|$ and δ a function assessing the similarity of two statistic profile features.

The method for determining this similarity needs to be defined according to the profile feature. Consider for example a profile feature $F(G)$ counting the literals in a dataset and $\hat{F}(G)$ estimating this value based on a sample. Then the similarity between them may be calculated as the absolute difference between the true count and the estimated value. In network theory, the similarity of a sample is commonly assessed by how well it captures the structural properties of the original graph [2, 9, 15]. However, since the labels of the edges and nodes in an RDF graph hold semantic information on the entities and concepts described in the graph, merely considering structural features may not be sufficient to assess how representative a sample of an RDF graph is. Hence, we propose a more comprehensive profile feature based on the characteristic sets capturing structural and semantic features of the graph's entities, which we present in the following.

5 Characteristic Sets Profile Feature Estimation

In this work, we present a comprehensive profile feature based on characteristic sets that captures both structural and semantic aspects of RDF graphs. This Characteristic sets profile feature (CSPF) can formally be defined as the following.

Definition 5 (Characteristic Sets Profile Feature (CSPF)). *Given a RDF graph G , the characteristic sets profile feature $F(G)$ is a 3-tuple (\mathcal{S}, c, m) with:*

- $\mathcal{S} = \mathcal{S}_C(G)$, the set of characteristic sets in G ,
- $c : \mathcal{S} \rightarrow \mathbb{N}$ a function for count as defined in Eq. 1, and
- $m : I \times \mathcal{S} \rightarrow \mathbb{R}^+$ a function for multiplicity as defined in Eq. 2.

Our approach addressing the profile feature estimation problem for CSPFs is shown in Fig. 1. Given a graph G , we create a sample $H \subset G$ using one of the RDF graph sampling methods presented in Sect. 5.1. Then, we build the CSPF $F(H)$ for the sample H . Finally, we apply one of the projection functions presented in Sect. 5.2, to extrapolate the feature observed in H to estimate those of the original graph as $\phi(F(H))$. We apply a set of similarity measures for characteristic sets defined in Sect. 5.3 to determine the similarity between the original CSPF $F(G)$ and its estimation $\hat{F}(G)$.

5.1 RDF Graph Sampling

The first component of our approach is the sampling method. When designing sampling methods, it is crucial to determine the kind of characteristic that should be captured before the collection of data. In this work, we collect samples to estimate the characteristic sets profile feature. Since each entity is associated with one characteristic set, we define the population as the set of entities in the graph: $E := \{s \mid (s, p, o) \in G\}$. Each observation in the sample corresponds to

one entity. The input of a sampling method is an RDF graph G and a sample size n' . The output of the sampling method is a subgraph H induced by n' entities of G . Let E' be the set of sampled entities with $|E'| = n'$, then $H := \{(s, p, o) \mid (s, p, o) \in G \wedge s \in E'\}$. We present three sampling methods differing in the probabilities of an entity being sampled. Thus, they allow for exploring different parts of the search space of possible characteristic sets during sampling.

Unweighted Sampling. It selects n' entities with equal probability from the population E . Thus, the probability $Pr(e)$ of $e \in E$ being a part of the sample is $Pr(e) = 1/|E|$.

Weighted Sampling. We present a biased sampling method which considers the out-degree of each entity e given by $d(e) := |\{(e, p, o) \mid (e, p, o) \in G\}|$. The weighted sampling method selects n' subjects where the probability of a subject to be chosen is proportional to its out-degree. In this way, entities that appear as subjects of many triples in the graph have a higher probability of being selected. Formally, the probability $Pr(e)$ of $e \in E$ being a part of the sample is given by $Pr(e) = d(e)/|G|$.

Hybrid Sampling. This sampling method combines the previous approaches where $\beta \cdot n'$ entities are selected using the unweighted method and $(1 - \beta) \cdot n'$ entities using the weighted method. Accordingly, the probability $Pr(e)$ of entity e being selected is

$$Pr(e) = \beta \cdot \frac{1}{|E|} + (1 - \beta) \cdot \frac{d(e)}{|G|}, \quad \beta \in [0, 1].$$

The β parameter allows for favoring either the weighted or the unweighted method.

5.2 Profile Feature Projection Functions

Next, the characteristic sets in the sample H are computed to create the corresponding CSPF $F(H)$. This can be done by first sorting the triples in H by subjects and then iterating all subjects determining the characteristic set for each subject. Given a profile feature $F(H) = (\mathcal{S}, c, m)$, the goal of a projection function is to extrapolate the statistical properties observed in sample H to the entire population as $\hat{F}(G) = \phi(F(H))$. In the following, we propose two classes of projection functions for the count values of the characteristic sets in the sample. The multiplicity statistic is not affected by the projection functions as it is a relative measure (the average occurrence of a predicate in a characteristic set) that does not require to be extrapolated. The first class, which we denote *basic projection functions*, only rely on information contained within the sample. The second class of projection functions rely on the information contained in the sample as well as additional high-level information on the dataset. We denote the latter class of functions as *statistics-enhanced projection functions*.

Basic Projection Function. This function simply extrapolates the count values for the given characteristic sets profile feature $F(H)$ based on the relative size of the sample. We define the function ϕ_1 which uses the ratio $r_t := \frac{|G|}{|H|}$ of triples in the sample with respect to the triples in the graph:

$$\phi_1(F(H)) := (\mathcal{S}, r_t \cdot c, m)$$

The assumption of this projection function is that the characteristic sets observed in the sample occur proportionally more often in the original graph. However, it neglects the fact that some characteristics sets might not have been sampled and is affected by potentially skewed distributions of the counts as exemplified in the introduction.

Statistics-Enhanced Projection Functions. The second class of projection functions incorporates additional high-level information about the original graph. In this work, we consider the number of triples per predicate in the original graph as a high-level statistic. The number of triples for predicate p' is given by $t(p') := |\{(s, p', o) \mid (s, p', o) \in G\}|$. We propose the ϕ_2 projection function that applies a true upper bound for the counts:

$$\phi_2(F(H)) := (\mathcal{S}, \dot{c}, m), \text{ with } \dot{c}(S_C) := \min(r_t \cdot c(S_C), \min_{p' \in S_C} t(p'))$$

The idea is that knowing how often a predicate occurs in the original graph allows for limiting the estimated counts for characteristic sets containing that predicate. This reduces the likelihood of overestimating counts without increasing the likelihood of underestimating them. Due to the fact that predicates, especially common ones such as `rdf:type`, may be used in several characteristic sets of the same graph, the aforementioned upper bound may be limited in its effectiveness. This is because it does not consider the number of characteristic sets a given predicate is part of. Therefore, we propose a third projection function ϕ_3 which “distributes” the upper bound for a predicate p' by considering the sum of counts of the characteristic sets, the predicates occurs in:

$$\phi_3(F(H)) := (\mathcal{S}, \ddot{c}, m), \text{ with } \ddot{c}(S_C) := \min \left(r_t \cdot c(S_C), \min_{p' \in S_C} \left(\frac{t(p') \cdot c(S_C)}{\sum_{S'_C \in \mathcal{S} \wedge p' \in S'_C} c(S'_C)} \right) \right)$$

The projection function ϕ_3 is adjusted by multiplying $t(p')$ with the ratio of the count $c(S_C)$ of S_C and the sum of counts for all characteristic sets p' occurs in. In contrast to ϕ_2 , this approach increases the likelihood of underestimating the count of characteristic sets. However, at the same time, it applies a more realistic upper bound by considering all characteristic sets a predicate occurs in and adjusting the upper bound accordingly. Note that further projection functions may be applied. For instance, the size of the characteristic sets or additional statistics about the predicates in the sample could be considered. However, we chose not to include them since they are likely produce projections that are tailored to specific graphs and cannot be generalized to other datasets.

5.3 Similarity Measures for Characteristic Sets

Finally, we define metrics that quantify the similarity between the estimated values and the real values to measure the quality of the profile estimations. Following the profile feature estimation problem defined in Definition 4, the goal is to identify an estimator $\hat{F}(G) = \phi(F(H))$ for the characteristic F that combines a sample H and projection function ϕ which maximizes the similarity δ between the estimated and the original profile feature. The similarity depends on the profile feature and we propose measures tailored to the characteristic sets profile feature (CSPF). Due to the diverse nature of the CSPF, there are multiple criteria to be considered when it comes to defining the similarity $\delta(F(G), \hat{F}(G))$ between the original CSPF $F(G) = (\mathcal{S}, c, m)$ and an estimated CSPF $\hat{F}(G) = (\hat{\mathcal{S}}, \hat{c}, \hat{m})$. In the following, we present a selection of similarity measures which consider both structural as well as statistical aspects captured by the CSPF. These measures take values in $[0, 1]$ and their interpretation is ‘higher is better’.

Structural Similarity Measures. Considering the structural properties, the mean out-degree and the predicate coverage can be considered to assess the similarity between the estimation and the original feature. We compute the *out-degree* similarity as

$$\delta^{od}(F(G), \hat{F}(G)) := 1 - \frac{|d_{mean}(F(G)) - d_{mean}(\hat{F}(G))|}{\max(d_{mean}(\hat{F}(G)), d_{mean}(F(G)))}, \text{ with} \quad (3)$$

$$d_{mean}(F(G)) := \frac{|G|}{\sum_{S_C \in \mathcal{S}} c(S_C)}$$

Note that $d_{mean}(\hat{F}(G))$ is computed analogously using H , $\hat{\mathcal{S}}$, and \hat{c} instead. Next, we can assess the *predicate coverage* similarity by computing the ratio of the number predicates covered in the estimation w.r.t. the number of predicates in the original profile feature as

$$\delta^{pc}(F(G), \hat{F}(G)) := \frac{|\{p \mid p \in S_C \wedge S_C \in \hat{\mathcal{S}}\}|}{|\{p \mid p \in S_C \wedge S_C \in \mathcal{S}\}|}. \quad (4)$$

The quality of the characteristic sets that are covered in the sample can be assessed by the following measures. First, the *absolute set coverage* similarity can be computed as the ratio of characteristic sets in the estimation to those in the original statistic profile:

$$\delta^{ac}(F(G), \hat{F}(G)) := |\hat{\mathcal{S}}| / |\mathcal{S}| \quad (5)$$

This measure, however, does not consider the amount of triples that haven been actually covered by the characteristic sets. The *relative set coverage* similarity of a characteristic set S_C of an RDF graph G reflects the relative amount of triples that S_C induces in G . The relative set coverage similarity δ^{rc} of an estimation

is calculated as the number of triples induced by all characteristic sets in the estimation on the original graph G :

$$\delta^{rc}(F(G), \hat{F}(G)) := \frac{\sum_{S_C \in \hat{\mathcal{S}}} \sum_{p \in S_C} m(p, S_C) \cdot c(S_C)}{|G|}. \quad (6)$$

Note that the characteristic sets in the estimation $\hat{\mathcal{S}}$ are considered while the number of triples they cover, i.e. $\sum_{p \in S_C} m(p, S_C) \cdot c(S_C)$, is w.r.t. the original graph. In this way, δ^{rc} reflects the relevance of the characteristic sets captured in the sample. For example, consider an RDF graph G with two characteristic sets S_1 and S_2 , where S_1 covers 90% and S_2 10% of all triples in G . Now, given an estimation with $\hat{\mathcal{S}} = \{S_1\}$, even though the estimation only capture 50% of the characteristic sets, the importance of S_1 is very high, as it covers 90% of the triples in the original graph.

Table 1. Overview of the similarity measures.

Structural similarity measures				Statistical similarity measures	
Out-degree coverage	Predicate coverage	Absolute set coverage	Relative set coverage	Count similarity	Multiplicity similarity
δ^{od} (3)	δ^{pc} (4)	δ^{ac} (5)	δ^{rc} (6)	$\delta_{S_C}^{count}$ (7)	$\delta_{S_C}^{multiplicity}$ (8)

Statistical Similarity Measures. Next, we focus on similarity measures which consider the *counts* and the *multiplicity* of predicates in the feature estimation. The degree to which counts and the multiplicities can be estimated accurately depends on the characteristic set. There might be characteristic sets for which these estimations may be very accurate, while for others this might not be the case. Hence, to avoid aggregating the similarity values for all characteristic sets to a single value, we define the similarity on the level of characteristic sets. Based on these values, an aggregation, such as mean or the median, may be used to obtain a single similarity value for all sets. For the similarity with respect to the count estimations, we adopt the q-error [12] by computing the maximum of the ratios between true and estimated count. Larger values for the q-error indicate a higher discrepancy between the true value and the estimation, and q-error of 1 indicates that the estimation is correct. Therefore, we use the inverse of the q-error to assess similarity

$$\delta_{S_C}^{count}(F(G), \hat{F}(G)) := \left(\max \left(\frac{c(S_C)}{\hat{c}(S_C)}, \frac{\hat{c}(S_C)}{c(S_C)} \right) \right)^{-1}, \forall S_C \in \hat{\mathcal{S}} \quad (7)$$

Note that the q-error measures the magnitude of the estimation error but does not reveal whether values are over- or underestimated. This property avoids that overestimated values cancel underestimated values out when the similarity values for all characteristic sets in the sample are aggregated. Analogously, we

compute the similarity of the multiplicities based on the q-error. We aggregate the values for all predicates in the characteristic sets using the mean to obtain a single value, as follows

$$\delta_{S_C}^{multiplicity}(F(G), \hat{F}(G)) := \left(\frac{1}{|S_C|} \sum_{p \in S_C} \max \left(\frac{\hat{m}(p, S_C)}{m(p, S_C)}, \frac{m(p, S_C)}{\hat{m}(p, S_C)} \right) \right)^{-1}, \forall S_C \in \hat{\mathcal{S}} \quad (8)$$

Summarizing, a CSPF implicitly and explicitly captures various characteristics of RDF graphs. The quality of estimating such a feature may not be assessed by a single similarity value but requires considering various metrics which are summarized in Table 1.

6 Experimental Evaluation

In this section, we empirically analyze the different components of our proposed approach. The goal of the evaluation is to investigate the following core questions:

- Q1** How do different sampling sizes influence the similarity measures?
- Q2** What is the impact of different sampling methods on the similarity measures?
- Q3** What are the effects of leveraging additional statistics in the projection functions?
- Q4** How do different characteristics of the RDF graph influence the estimation?

Next, we present the setup of our experiments and present and analyze the results of our experiments. Based on our findings, we answer the addressed questions in the conclusions (cf. Sect. 7). The source code and the sample results are available online.¹

Table 2. Characterization of the four RDF graphs studied in the experiments.

RDF graph	# Triples	# Subj.	# Pred.	# Obj.	d_{mean}	d_{std}	$ S $	$\frac{ S }{\# \text{ Subjects}}$	$\frac{ S }{ S }$	AUC
DBLP	88,150,324	5,125,936	27	36,413,780	17.2	9.38	270	0.005%	15%	99.13%
LinkedMDB	5,444,664	688,187	220	1,930,703	7.91	5.9	8516	1.24%	62%	97.40%
Wordnet	5,558,748	647,215	64	2,483,030	8.58	10.26	777	0.12%	37%	98.22%
YAGO	82,233,128	6,429,347	79	50,670,009	12.79	15.82	29309	0.46%	49%	98.76%

Datasets. We selected four well-known RDF graphs from different domains: publications (DBLP), movies (LinkedMDB), linguistics (Wordnet), and cross-domain (YAGO). An overview of their characteristics is shown in Table 2. The graphs differ with respect to their size (number of triples), the number of distinct subjects, predicates, and objects as well as the number of characteristic

¹ https://github.com/Lars-H/hdt_sampler, <https://doi.org/10.5445/IR/1000117614>.

sets $|\mathcal{S}|$. As the number of potential characteristic sets not only depends on the distinct predicates, but it is bound by the number of distinct subjects in the graph, we also provide the ratio $|\mathcal{S}|/\# \text{ Subjects}$ in percent as a measure of the characteristic sets' diversity. Furthermore, we consider *exclusive* characteristic sets defined as $\mathcal{S}^1 := \{S_C \mid \text{count}(S_C) = 1 \wedge S_C \in \mathcal{S}\}$ and provide the ratio of exclusive characteristic sets to all characteristic sets. An exclusive characteristic set only occurs once in the entire graph and as a result, introduces two major difficulties when sampling and projecting the characteristic sets: (i) it is unlikely to sample them as they occur only once, and (ii) when projecting them, it is likely to overestimate their counts. However, because the coverage of exclusive characteristic sets is low, it is potentially less important to correctly project them, as they might be less relevant as other characteristic sets.

For each RDF graph, we indicate the area under the curve (AUC) below the relative cumulative coverage curve (cf. Fig. 2). For the relative cumulative coverage curve, the characteristic sets are ranked and sorted in decreasing order according to the number of triples they cover on the x-axis (cf. Sect. 5.3) and on the y-axis, the cumulative sum of the relative number of triples they cover is indicated. For instance, the curve for DBLP shows that the characteristic set with the highest coverage (i.e., the start of the curve on the left), covers almost 40% of all triples and 20% of the characteristic sets cover almost all triples in the graph (relative cumulative coverage ≈ 0.99). As a result, the shape of the curve indicates how evenly the coverage is distributed across the characteristic sets. A diagonal line indicates all characteristic sets covering the same number of triples. The stronger the curve is dented towards the upper left corner the more unevenly is the coverage of the characteristic sets distributed. This indicates that a few sets cover many triples in the graph. Consequently, a large AUC indicates unevenly distributed coverage.

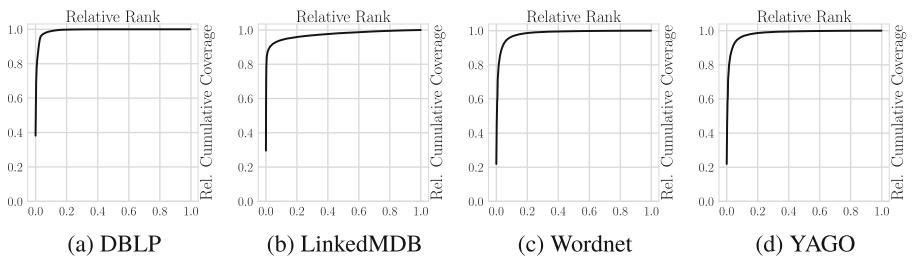


Fig. 2. The cumulative relative coverage curve shows the ratio of triples covered with respect to the characteristic sets ordered by decreasing relative coverage.

Sampling Methods. We study the presented unweighted, weighted and hybrid sampling methods. For the hybrid sampling method we chose $\beta = 0.5$. We study four different sample sizes defined relative to the number of entities $|E|$ with

$n' = \{0.1\% \cdot |E|, 0.5\% \cdot |E|, 1\% \cdot |E|, 5\% \cdot |E|\}$ (Note: $10\% = 1\%$). We generate 30 samples per dataset, sampling method, and sample size resulting in a total of $30 \cdot 4 \cdot 3 \cdot 4 = 1,440$ samples.

6.1 Results: Structural Similarity Measures

Table 3 presents the results for the measures out-degree δ^{od} , predicate coverage δ^{pc} , and absolute set coverage δ^{ac} , and relative set coverage δ^{rc} for the different sampling methods. Included are also the ratios of triples sampled $|H|/|G|$ in permille (%).

Considering sample size (Q1), the results show an improvement on the similarity measures as the sample size increases, with a few exceptions for δ^{od} . In particular, in Wordnet and YAGO, the best similarity values δ^{od} are achieved for the highest relative sample size (5.0%), while for DBLP and LinkedMDB the best performance is achieved with a sample size of 1.0%. The predicate coverage similarity δ^{pc} also improves with increasing sample sizes. For instance, from 220 predicates in LinkedMDB the sampling methods obtain ≈ 66 predicates with the smallest sample size and ≈ 154 with the largest. For all the studied graphs, a similar relation between the absolute (δ^{ac}) and relative set coverage (δ^{rc}) is observed. Even if only a few characteristic sets are sampled (low δ^{ac}), the number of triples in the original graph covered by those sets is very high (high δ^{rc}). For example, in Wordnet, the unweighted sampling (5.0 %) obtains 12% ($\delta^{ac} = 0.12$) of all characteristics sets which cover 95% ($\delta^{rc} = 0.95$) of all triples in the graph.

Table 3. Mean similarity values δ^{od} , δ^{pc} , δ^{ac} , δ^{rc} and mean sampled triples ratio $|H|/|G|$ in permille (%) by sample size and sampling method (h = hybrid, u = unweighted, w = weighted). Best values per RDF graph and similarity measure are indicate in **bold**.

DBLP				LinkedMDB				Wordnet				YAGO									
	$\frac{ H }{ G }$	δ^{od}	δ^{pc}	δ^{ac}	$\frac{ H }{ G }$	δ^{od}	δ^{pc}	δ^{ac}	$\frac{ H }{ G }$	δ^{od}	δ^{pc}	δ^{ac}	$\frac{ H }{ G }$	δ^{od}	δ^{pc}	δ^{ac}	δ^{rc}				
0.1 %	h	0.11	0.84	0.94	0.09	0.98	0.13	0.72	0.32	0.00	0.70	0.17	0.57	0.46	0.02	0.77	0.17	0.59	0.67	0.01	0.32
	u	0.10	0.95	0.93	0.08	0.98	0.10	0.90	0.31	0.00	0.69	0.10	0.86	0.42	0.02	0.71	0.10	0.93	0.60	0.01	0.31
	w	0.13	0.75	0.94	0.09	0.98	0.16	0.61	0.34	0.00	0.70	0.23	0.43	0.50	0.03	0.80	0.25	0.40	0.75	0.01	0.32
0.5 %	h	0.57	0.85	0.98	0.16	0.99	0.64	0.76	0.48	0.01	0.82	0.86	0.57	0.62	0.06	0.90	0.87	0.57	0.88	0.02	0.37
	u	0.50	0.96	0.98	0.16	0.99	0.50	0.94	0.44	0.01	0.82	0.49	0.92	0.56	0.04	0.87	0.50	0.96	0.72	0.02	0.36
	w	0.65	0.75	0.99	0.16	0.99	0.79	0.62	0.46	0.01	0.83	1.22	0.42	0.66	0.07	0.92	1.25	0.40	0.92	0.02	0.37
1.0 %	h	1.15	0.86	1.00	0.21	1.00	1.28	0.78	0.52	0.01	0.85	1.73	0.59	0.70	0.08	0.93	1.74	0.59	0.93	0.03	0.38
	u	1.00	0.98	1.00	0.20	1.00	1.00	0.97	0.51	0.01	0.84	1.01	0.95	0.63	0.06	0.91	1.00	0.97	0.79	0.02	0.38
	w	1.30	0.77	1.00	0.21	1.00	1.57	0.64	0.54	0.02	0.85	2.45	0.42	0.74	0.10	0.94	2.43	0.43	0.95	0.03	0.38
5.0 %	h	5.74	0.85	1.00	0.31	1.00	6.42	0.76	0.72	0.04	0.89	8.48	0.58	0.82	0.17	0.97	8.40	0.59	0.97	0.08	0.40
	u	5.00	0.97	1.00	0.31	1.00	5.00	0.96	0.73	0.03	0.88	5.02	0.96	0.75	0.12	0.95	4.99	0.97	0.88	0.06	0.40
	w	6.49	0.75	1.00	0.31	1.00	7.75	0.63	0.74	0.06	0.90	11.9	0.42	0.84	0.21	0.98	11.7	0.43	0.97	0.09	0.40

Regarding the sampling methods (Q2), the unweighted approach performs best for the out-degree similarity δ^{od} . This relates to the fact that the hybrid and weighted sampling methods select high out-degree entities with a higher probability. To illustrate this, consider Fig. 3 that shows the characteristic sets that are constructed with two different sampling methods (in color) in comparison to the characteristic sets from the original graph (in gray). The weighted sampling methods (Fig. 3a) leads to characteristic sets with higher set size (highlighted in the rectangle), while the unweighted sampling (Fig. 3b) captures average-sized characteristic sets. Furthermore, a higher dispersion of the out-degree distribution (d_{std}/d_{mean}) of the original graph (Q4), leads to a higher similarity for the unweighted sampling method in comparison to the other approaches.

In general, the unweighted sampling method exhibits the lowest predicate coverage similarity (δ^{pc}) in comparison to the other approaches. Combining this observation with Fig. 3, we conclude that the unweighted sampling method fails to obtain those predicates used in characteristic sets with high degrees. The only exception where all methods obtain every predicate is DBLP for sample sizes 1.0% and 5.0%, due a high average out-degree w.r.t. the number of predicates (cf. Table 2) in the original graph.

Considering absolute (δ^{ac}) and relative set coverage (δ^{rc}), the unweighted method performs almost as well in most cases while always sampling the fewest triples ($|H|/|G|$). The relation between absolute and relative set coverage is in accordance with the *AUC* property of the graphs, i.e., most triples are covered by few characteristic sets only.

6.2 Results: Statistical Similarity Measures

Next, we analyze the estimation results for the counts and multiplicity. Instead of presenting the similarity measures δ^{count} and $\delta^{multiplicity}$, we present the q-error as it is more commonly used in the literature. For each sample, mean and median q-error for count and multiplicity estimations across all characteristic sets $S_C \in \hat{\mathcal{S}}$ are computed. Note that mean/median for each sample are computed first to assess the performance on the sample level. We present the average of mean and median q-errors in Table 4 to get an indication of how well the average sample per dataset, size and method performs.

Regarding the graphs (Q4), the best count estimations are observed for DBLP and Wordnet where the best median values are between 1.27 and 1.53 indicating that, for half of the characteristic sets, the counts are misestimated by $\leq 27\%$ and $\leq 53\%$. The difference in the best mean values for Wordnet (6.09) and DBLP (3.55) reflects that in Wordnet there are higher misestimations on average. For YAGO, the best median q-error is 2.12 for the largest sample size and the unweighted method. The corresponding mean (16.0) is almost 8 times higher than the median indicating a strong positive skew of the q-error distribution. For LinkedMDB the best median result 1.49 is achieved with the smallest sample size, however, it needs to be noted that this smaller sample also covers fewer characteristic sets (cf. δ^{ac} in Table 3). Taking the characteristics of the original graphs into consideration, two observation may explain the differences in q-errors: (i) a

Table 4. Mean and median for q-errors of the count estimations for the projection functions ϕ_1 , ϕ_2 , and ϕ_3 as well as for the multiplicity estimation. Best values per column are **bold** and values for the best projection function are highlighted in gray.

DBLP												LinkedMDB													
	ϕ_1			ϕ_2			ϕ_3			multiplicity			ϕ_1			ϕ_2			ϕ_3			multiplicity			
	mean	median	mean	median	mean	median	mean	median	mean	median	mean	median	mean	median	mean	median	mean	median	mean	median	mean	median	mean	median	
0.1 %	h	6.31	1.61	5.11	1.57	4.16	1.49	1.04	1.02	547	2.56	222	2.14	117	2.01	1.04	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.01	
	u	16.4	1.64	10.4	1.59	9.77	1.51	1.04	1.02	262	2.22	117	1.57	79.2	1.49	1.03	1.0	1.01	1.01	1.01	1.01	1.01	1.01	1.01	
	w	25.6	1.62	25.1	1.61	18.4	1.48	1.04	1.02	600	5.07	298	3.84	155	3.03	1.05	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.01	
0.5 %	h	3.9	1.51	3.89	1.49	3.54	1.48	1.04	1.02	207	4.76	173	3.68	87.7	3.07	1.05	1.02	1.02	1.02	1.02	1.02	1.02	1.02	1.02	
	u	4.47	1.38	4.46	1.36	4.12	1.34	1.04	1.02	130	2.44	108	1.84	74.1	1.78	1.04	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.01	
	w	5.36	1.51	5.35	1.5	4.94	1.5	1.06	1.02	217	12.2	185	11.6	76.0	6.9	1.06	1.04	1.04	1.04	1.04	1.04	1.04	1.04	1.04	
1.0 %	h	5.51	1.45	5.51	1.44	5.14	1.43	1.04	1.02	117	6.41	106	6.14	57.1	4.45	1.06	1.04	1.04	1.04	1.04	1.04	1.04	1.04	1.04	
	u	6.85	1.36	6.85	1.35	6.24	1.32	1.04	1.02	95.5	2.81	87.2	2.35	68.6	2.28	1.05	1.02	1.02	1.02	1.02	1.02	1.02	1.02	1.02	
	w	5.89	1.43	5.88	1.43	5.33	1.43	1.05	1.02	128	13.6	116	13.2	49.7	7.08	1.07	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	
5.0 %	h	4.06	1.33	4.06	1.33	3.88	1.32	1.04	1.01	39.9	8.67	39.0	8.64	21.7	5.38	1.07	1.06	1.06	1.06	1.06	1.06	1.06	1.06	1.06	
	u	3.6	1.28	3.6	1.28	3.53	1.27	1.03	1.01	35.2	4.36	34.5	4.09	30.9	3.89	1.06	1.04	1.04	1.04	1.04	1.04	1.04	1.04	1.04	
	w	3.96	1.36	3.96	1.36	3.77	1.38	1.05	1.02	37.9	10.8	37.1	10.8	16.8	5.67	1.07	1.06	1.06	1.06	1.06	1.06	1.06	1.06	1.06	
Wordnet												YAGO													
	ϕ_1			ϕ_2			ϕ_3			multiplicity			ϕ_1			ϕ_2			ϕ_3			multiplicity			
	mean	median	mean	median	mean	median	mean	median	mean	mean	median	mean	mean	median	mean	median	mean	median	mean	median	mean	median	mean	median	
0.1 %	h	43.6	2.31	12.5	2.22	9.04	2.01	1.11	1.11	158	3.07	145	3.04	125	2.91	1.3	1.24	1.24	1.24	1.24	1.24	1.24	1.24	1.24	
	u	74.6	2.42	11.6	2.11	8.98	1.72	1.1	1.08	161	3.05	153	3.03	129	2.77	1.25	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	
	w	53.8	3.34	23.4	2.81	17.8	2.48	1.12	1.11	150	3.67	137	3.65	125	3.49	1.33	1.27	1.27	1.27	1.27	1.27	1.27	1.27	1.27	
0.5 %	h	28.5	2.28	18.9	2.19	14.4	2.04	1.12	1.11	61.2	2.74	58.3	2.74	54.6	2.68	1.29	1.24	1.24	1.24	1.24	1.24	1.24	1.24	1.24	
	u	22.6	1.85	15.1	1.75	12.2	1.57	1.1	1.08	60.0	2.64	59.1	2.64	55.4	2.51	1.24	1.19	1.19	1.19	1.19	1.19	1.19	1.19	1.19	1.19
	w	28.0	2.68	23.5	2.6	18.5	2.37	1.13	1.12	56.3	3.23	52.8	3.22	48.5	3.15	1.32	1.26	1.26	1.26	1.26	1.26	1.26	1.26	1.26	
1.0 %	h	24.4	2.18	18.9	2.17	14.6	2.0	1.12	1.1	38.2	2.51	36.7	2.5	34.6	2.47	1.29	1.24	1.24	1.24	1.24	1.24	1.24	1.24	1.24	
	u	20.3	1.78	15.2	1.71	12.4	1.59	1.1	1.09	41.9	2.45	41.6	2.45	39.3	2.37	1.24	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	
	w	21.2	2.7	17.1	2.68	13.7	2.36	1.13	1.12	37.5	3.12	36.2	3.11	33.1	3.01	1.32	1.27	1.27	1.27	1.27	1.27	1.27	1.27	1.27	
5.0 %	h	10.0	2.14	8.72	2.13	7.26	1.95	1.12	1.1	15.5	2.43	15.5	2.43	14.1	2.39	1.29	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	
	u	7.55	1.6	6.8	1.58	6.09	1.58	1.1	1.08	16.3	2.14	16.3	2.13	16.0	2.12	1.24	1.24	1.2	1.2	1.2	1.2	1.2	1.2	1.2	
	w	9.75	2.52	8.78	2.51	7.07	2.17	1.12	1.11	14.7	3.0	14.7	2.99	13.1	2.89	1.32	1.27	1.27	1.27	1.27	1.27	1.27	1.27	1.27	

higher characteristic set diversity ($|S_C|/\#Subjects$) yields higher q-errors, and (ii) a higher ratio of exclusive characteristic sets yield higher q-errors. Regarding (i): with many possible characteristic sets to be sampled from, it is likely to sample few entities per set. However, sampling several entities per characteristic set allows for better estimating their overall occurrences. Considering (ii): many exclusive characteristic sets increase the likelihood of them being sampled and their counts to be overestimated, as the projection function cannot distinguish them from non-exclusive characteristic sets. Inspecting the projection functions (Q3), the statistic-enhanced functions ϕ_2 and ϕ_3 slightly reduce the mean and median q-errors for the count estimations. In all cases, ϕ_3 yields the best estimations and should be favored over ϕ_2 whenever the additional statistics are

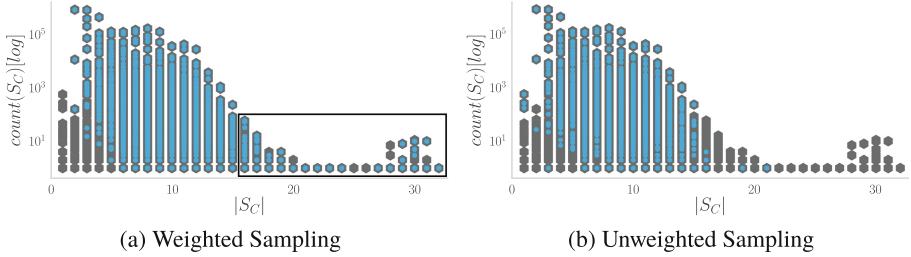


Fig. 3. Example of the sampled characteristic sets for YAGO with respect to the number of their predicates ($|S_C|$) and their count ($\text{count}(S_C)$) on a log-scale. Indicated in gray are all sets of the original dataset and in color those, which are contained in the sample. (Color figure online)

available. Simultaneously, the improvements over the basic projection function ϕ_1 diminish with an increasing sample size indicating that larger samples contain fewer outliers which are corrected by the additional statistics in ϕ_2 and ϕ_3 .

For the multiplicity estimations, the mean and median q-errors are below 1.3 in all cases for all graphs. They are less affected by sampling methods and sample sizes reflecting a uniform predicate usage within the characteristic sets with few outliers. Regarding the sample size (Q1), in most cases, a larger sample provides better results for count and multiplicity estimations while at the same time estimating more characteristic sets from the original graph (cf. δ^{ac} in Table 3). As a result, increasing the sampling size not only improves the overall accuracy but also the number of characteristic sets estimated. Similar to previous observations, the unweighted sampling method (Q2) yields the best results in most cases for count and multiplicity estimations.

7 Conclusions and Future Work

We have introduced the problem of RDF dataset profile feature based on characteristic sets and proposed a solution based on sampling. The presented profile feature estimation approach obtains a sample from the original graph, computes the profile feature for the sample, and uses a projection function to estimate the true profile feature. Different applications can benefit from the resulting feature estimations. For instance, query plan optimization in decentralized querying can benefit from the estimations to find efficient query plans, even when the entire dataset may not accessible to compute the complete statistics. We conducted an empirical study to evaluate the similarities between the estimations and the true profile features. We presented and analyzed the results of our study and to conclude our findings, we answer the questions presented in Sect. 6:

Answer to Q1. Larger sample sizes have two major positive effects: (i) they improve the structural and statistical similarities measures, and (ii) they capture and estimate the statistics for more characteristic sets of the original graph. Regardless, datasets with a high number of characteristic sets can still be challenge. In such cases it may be beneficial to use additional information, such as query logs, to lead the sampling method towards the most relevant characteristic sets.

Answer to Q2. The similarity of the estimated profile features depends on the chosen sampling method. The unweighted sampling method yields the highest similarity values in the majority of cases while requiring the fewest triples to be sampled.

Answer to Q3. Projection functions leveraging additional statistics (i.e., overall counts per predicate) achieve better results for projecting the counts of characteristic sets. The improvements over the zero-knowledge projection function diminish with increasing sample size.

Answer to Q4. The structure of the RDF graph affects the similarity values. Especially count values are misestimated for datasets with a large share of exclusive characteristic sets and a larger diversity of characteristic sets. In such scenarios, larger sample sizes can help improving the estimations.

Our future work will focus on investigating the impact of estimated Characteristic Sets Profile Features on the performance of query plan optimizers.

Acknowledgement. This work is funded by the German BMBF in QUOCA, FKZ 01IS17042.

References

1. Abedjan, Z., Grütsch, T., Jentzsch, A., Naumann, F.: Profiling and mining RDF data with ProLOD++. In: Proceedings of ICDE (2014)
2. Ahmed, N.K., Neville, J., Kompella, R.R.: Network sampling: from static to streaming graphs. TKDD **8**(2), 7:1–7:56 (2013)
3. Auer, S., Demter, J., Martin, M., Lehmann, J.: LODStats - an extensible framework for high-performance dataset analytics. In: Poceedings of EKAW, pp. 353–362 (2012)
4. Debattista, J., Londoño, S., Lange, C., Auer, S.: Quality Assessment of Linked Datasets Using Probabilistic Approximation. In: Gandon, F., Sabou, M., Sack, H., d'Amato, C., Cudré-Mauroux, P., Zimmermann, A. (eds.) ESWC 2015. LNCS, vol. 9088, pp. 221–236. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-18818-8_14
5. Ellefi, M.B., et al.: RDF dataset profiling - a survey of features, methods, vocabularies and applications. Semant. Web **9**(5), 677–705 (2018)
6. Fernández, J.D., Martínez-Prieto, M.A., de la Fuente Redondo, P., Gutiérrez, C.: Characterising RDF data sets. J. Inf. Sci. **44**(2), 203–229 (2018)
7. Gubichev, A., Neumann, T.: Exploiting the query structure for efficient join ordering in SPARQL queries. In: Proceedings of EDBT (2014)
8. Khatchadourian, S., Consens, M.P.: ExpLOD: summary-based exploration of inter-linking and RDF usage in the linked open data cloud. In: Aroyo, L., et al. (eds.) ESWC 2010. LNCS, vol. 6089, pp. 272–287. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13489-0_19

9. Leskovec, J., Faloutsos, C.: Sampling from large graphs. In: Proceedings of ACM SIGKDD, pp. 631–636 (2006)
10. Mannino, M.V., Chu, P., Sager, T.: Statistical profile estimation in database systems. ACM Comput. Surv. **20**(3), 191–221 (1988)
11. Meimaris, M., Papastefanatos, G., Mamoulis, N., Anagnostopoulos, I.: Extended characteristic sets: graph indexing for SPARQL query optimization. In: Proceedings of ICDE (2017)
12. Moerkotte, G., Neumann, T., Steidl, G.: Preventing bad plans by bounding the impact of cardinality estimation errors. PVLDB **2**(1), 982–993 (2009)
13. Montoya, G., Skaf-Molli, H., Hose, K.: The *Odyssey* approach for optimizing federated SPARQL queries. In: d’Amato, C., et al. (eds.) ISWC 2017. LNCS, vol. 10587, pp. 471–489. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68288-4_28
14. Neumann, T., Moerkotte, G.: Characteristic sets: accurate cardinality estimation for rdf queries with multiple joins. In: Proceedings of ICDE (2011)
15. Ribeiro, B.F., Wang, P., Murai, F., Towsley, D.: Sampling directed graphs with random walks. In: Proceedings of the IEEE INFOCOM, pp. 1692–1700 (2012)
16. Rietveld, L., Hoekstra, R., Schlobach, S., Guéret, C.: Structural properties as proxy for semantic relevance in RDF graph sampling. In: Mika, P., et al. (eds.) ISWC 2014. LNCS, vol. 8797, pp. 81–96. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11915-1_6
17. Soulet, A., Suchanek, F.M.: Anytime large-scale analytics of linked open data. In: Ghidini, C., et al. (eds.) ISWC 2019. LNCS, vol. 11778, pp. 576–592. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30793-6_33