Using Provenance for Personalized Quality Ranking of Scientific Datasets

Yogesh Simmhan
University of Southern California, Los Angeles CA 90089
simmhan@usc.edu

Beth Plale
Indiana University, Bloomington IN 47405
plale@cs.indiana.edu

ABSTRACT

The rapid growth of eScience has led to an explosion in the creation and availability of scientific datasets that includes raw instrument data and derived datasets from model simulations. A large number of these datasets are surfacing online in public and private catalogs, often annotated with XML metadata, as part of community efforts to foster open research. With this rapid expansion comes the challenge of filtering and selecting datasets that best match the needs of scientists. We address a key aspect of the scientific data discovery process by ranking search results according to a personalized data quality score based on a declarative quality profile to help scientists select the most suitable data for their applications. Our quality model is resilient to missing metadata using a novel strategy that uses provenance in its absence. Intuitively, our premise is that the quality score for a dataset depends on its provenance – the scientific task and its inputs that created the dataset – and it is possible to define a quality function based on provenance metadata that predicts the same quality score as one evaluated using the user’s quality profile over the complete metadata. Here, we present a model and architecture for data quality scoring, apply machine learning techniques to construct a quality function that uses provenance as proxy for missing metadata, and empirically test the prediction power
of our quality function. Our results show that for some scientific tasks, quality scores based on provenance closely track the quality scores based on complete metadata properties, with error margins between 1 – 29%.

**Keywords**

Scientific data, data quality, information retrieval, provenance, missing metadata, social perception, accessibility, eScience.

1. **INTRODUCTION**

The proliferation of affordable computing resources combined with the pervasive deployment of sensors and large scale instruments capable of observing nature at the macro and microscopic levels has enabled the rapid growth of data driven computational science, also known as *eScience* [1]. This has led to an explosion in the availability of scientific datasets [2], including raw data from instruments like digital survey telescopes [3], carbon-flux sensor towers [4] and next generation sequencing machines [5], and derived data from computational models, such as weather forecasting and particle simulations [6, 7, 8]. These datasets are surfacing online in large numbers in public and private repositories, accessible to users within a scientific community or beyond, in an effort to foster inter-organizational and inter-disciplinary research that can accelerate scientific discovery [9, 4]. These published datasets numbering in the millions are growing by the day, and are archived in perpetuity thanks to cheap Cloud storage and disks [10].

Wide-spread access to datasets across disciplines presents significant opportunities for more effective science: it accelerates scientific impact through interdisciplinary research, democratizes the data access and discovery process, improves scientific accountability
through peer reuse, and makes science less mysterious to the general public and policymakers. Datasets available for open access are often described using structured metadata that allows for its search and provides a context for reuse [11]. XML has emerged as a popular representation for describing these datasets, with semantic ontologies starting to gain traction [12, 13].

Current research on crawling the hidden web and efficient indexing of XML provides the basic building blocks for searching this vast space [14, 15, 16] and scientific catalogs leverage these to search domain specific collections. However, they do not effectively capture the unique needs of an individual researcher or their application, and fail to solve the problem beyond traditional text-based information retrieval techniques. Semantic annotations based on domain ontologies partially address this, but are in limited use and do not cope with missing metadata that is common when the scientific process goes beyond the sandbox.

1.1 Personalized Quality Ranking

As availability of eScience data expands, filtering the search space and ranking the results to best match the needs of the scientist become major concerns. While the scientist may occasionally perform exploratory search for datasets for *ad hoc* visualization, the consumers of these results are often applications and visualization tools that are part of an automated scientific workflow pipeline [6, 17]. As a result, the nature of these applications and workflows end up characterizing the dataset being searched for. Automated discovery of the appropriate datasets for input to these workflows can reduce user involvement and be invaluable when a scientist launches a large-scale parallel parameter sweep of 1000 models simultaneously [18, 19].
In the absence of a universal notion of “good” or “bad” quality scientific data [20, 21, 22], the process of filtering, ranking and selection of datasets must be personalized [23, 24] both for the user and for the particular application that will consume the data. Data quality is relative. One user’s “good quality” data can be another user’s “bad quality” data. For example, when locating radar data to provide the initial inputs to a weather forecast simulation being run at a grid spacing of 2km, datasets with resolutions closest to the required grid spacing are preferred. Instrument input data at a higher resolution will require extra processing to down sample them and also incur higher bandwidth and storage costs, while input data at a lower resolution will lead to a less accurate forecast. So the resolution of the instrument data is not an absolute measure of the dataset’s quality but rather a quality factor which has to be interpreted according to the user application’s requirements.

One major aspect of the data discovery process that we address in this article is ranking the search results according to a personalized data quality score that presents the most appropriate data for the scientists’ application needs.

1.2 Provenance and Missing Metadata

A significant source of the scientific dataset explosion is model-derived datasets – the results or by-products of scientific simulations – that are dwarfing the quantity of raw datasets. This growth is understandable: the same raw radar dataset fed into a parameter sweep ensemble forecast workflow results in a veritable swathe of derived simulation data. While intermediary derived datasets generated during the investigation may not have been retained, the metadata that describes them remains important and is published as part of the investigation [10]. In particular, provenance is a key piece of metadata that
enables reconstruction of derived data and provides the context for reusing it. Provenance describes the derivation history of a dataset, capturing information about the scientific task or service that generated the dataset, and the input datasets and parameters that went into the application execution [25, 26]. Provenance provides an intuitive notion of the quality of data that supports our ranking process. The importance of provenance has led to a community specification for provenance, the Open Provenance Model (OPM) [27], with efforts underway to promote a W3C working group [28] to define a recommendation.

Scientific datasets are commonly described using domain specific metadata schemas, frequently represented in XML or RDF [29, 30]. Given the complex nature of science, these standard schemas tend to be bloated with hundreds of properties used to describe the nuances of a dataset in painstaking detail. This, however, does not necessarily translate to a rich query experience for the end user. Rather, scientists are faced with a bewildering array of metadata properties to specify their data searches. This frequently forces them to adopt a free form text-based keyword search that performs gross filtering but no fine-grained ranking that is sensitive to user needs. The personalized ranking technique we propose addresses this complexity in part.

A more serious consequence of exhaustive metadata schemas is that the data provider is not always able to provide meaningful content for all the properties. Hence, scientific metadata is routinely afflicted with missing elements and none but the most crucial portions of the metadata schema are often present [31, 32]. One common reason is that not all metadata elements are automatically populated. Scientists balk at manually specifying metadata values each time they launch an experiment and this if often a
consequence of the scientific middleware failing to provide the right tools to ease or automate the annotation process. Yet another reason for missing metadata is that the dataset may pass through disconnected data management tools before arriving at the archive location, and information may be lost in the process of export, translation and import.

The importance of provenance to the scientific process gives it a better chance of being captured, retained and shared as compared to other metadata. It is also intuitive to consider provenance as an indicator of the quality of the data. We formalize this notion by using provenance to model a quality function for scientific tasks, which uses the quality score of the input data to map to the quality score for the derived output data. In this process, the provenance-based quality function can provide a personalized quality score even when other metadata about the derived data in missing, and allow effective ranking of datasets.

*The second aspect of the data discovery process that we address in this article is resilient quality scoring and ranking of search results using provenance as a proxy for missing metadata.*

### 1.3 Contributions and Outline

In this paper, we make two specific contributions that address meaningful discovery of scientific datasets:

1. A model and architecture for assigning personalized quality scores for scientific data based on user quality profiles specified over diverse and distributed metadata, in order to allow filtering and ranking of datasets; and
2. A quality function constructed using machine learning techniques, which uses provenance metadata to effectively determine the data quality score in the presence of missing metadata for datasets.

The quality model we propose extends our prior work and allows scientists to define a personalized quality profile for their application using metadata elements relevant to data discovery [33]. Our quality broker framework uses this profile to evaluate a numerical quality score for individual datasets that can be used for uniform and consistent ranking of the datasets.

We handle quality ranking in the presence of missing metadata using a novel strategy based on the premise that for a given user quality profile, the quality score of a derived dataset can be learnt and predicted using its provenance metadata which acts as a proxy for other (missing) metadata specified in the profile. We empirically test and validate this hypothesis to show that for a certain class of scientific tasks, quality scores evaluated exclusively using provenance metadata closely predicts the quality scores based on other metadata elements, with error margins under 29% being observed. This means that the availability of provenance metadata makes the quality scoring and result ranking resilient to other missing metadata.

The rest of the paper is structured as follows. We introduce our quality and information model used for quality scoring in Section 2, and present a formalism for predicting quality score using provenance in Section 3. The architecture for a prototype quality ranking system in described Section 4. We discuss data provenance and its application as a predictor of data quality using machine learning techniques in Section 5, and provide
empirical evidence to test its efficacy in Section 6. We present related work in Section 7 and our conclusions in Section 8.

2. CONCEPTUAL MODEL OF METADATA AND QUALITY

The quality model we propose allows users to define a personal quality profile for a class of scientific applications that is used as the basis for determining the quality scores for datasets. The profile consists of quality constraints or assertions on the values (or existence) of specific metadata properties associated with the dataset, and an associated relative weight and quality sub-score for that constraint being met. The constraints in the quality profile can then be evaluated to an overall quality score for a given dataset that is associated with a set of metadata, and the score used to rank different datasets uniformly. In this section, we describe the metadata and the quality models in detail.

2.1 Metadata Model

The conceptual model of the metadata views it as a set of typed <property, value(s)> tuples. The property is a semantically meaningful term that describes a domain concept that is familiar to a domain user, and possibly backed by a glossary or ontology. The properties are typed so that the values they are assigned in the metadata instance match that type. The conceptual model and our implementation presently support simple numeric, string, enumerated, and Boolean types, and their array variants. The actual metadata representation, for example in XML, is mapped to these conceptual metadata properties and values using extraction and transformation rules described later.

2.2 Classes of Metadata
While data discovery provided by conventional metadata catalogs is primarily based on the intrinsic metadata annotations that are stored within the catalog, in reality, the quality of a dataset for scientific applications depends on additional factors that may be present in other repositories or influenced by external service providers. For example, recommender systems or social networks may provide indications of the quality of a dataset based on how many community members refer to it in their blog post, and data transfer services may provide a particular quality of service for accessing the dataset.

We group our metadata properties into four classes with distinct features, and relevant to scientific data: intrinsic metadata, data provenance, data accessibility, and social perception; these are discussed in detail in our earlier work [33]. While orthogonal to our information or quality model, these provide a practical grounding for our quality specification. To summarize, *intrinsic metadata* describes features of the dataset that are inherent to it and are typically scientific properties such as the spatial resolution of a dataset.

### Table 1. Typical metadata used as quality factors, along with common information providers and attributes.

<table>
<thead>
<tr>
<th>Attribute Class</th>
<th>Information Provider, with Earth Sciences examples</th>
<th>Typical Attributes</th>
<th>Format</th>
<th>Access Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intrinsic Metadata</td>
<td>Metadata catalogs. E.g. LEAD Personal Catalog, THREDDS Catalog</td>
<td>Publisher, Quality Flags, Domain Keywords</td>
<td>XML</td>
<td>Web service, REST, HTTP</td>
</tr>
<tr>
<td>Data Accessibility</td>
<td>Data Storage and Movement Services E.g. Reliable File Transfer (RFT) Service</td>
<td>Data Availability, Bandwidth, Cost</td>
<td>Name-Value, functions, Numeric</td>
<td>Web service</td>
</tr>
<tr>
<td>Social Perception</td>
<td>Guna Quality Broker, Recommender System, Social Networks</td>
<td>Peer quality score, Data usage frequency</td>
<td>Name-Value, Numeric</td>
<td>Web service, REST, HTTP</td>
</tr>
</tbody>
</table>
Data provenance is the derivation history of data, as mentioned before, and multiple generations of ancestral data and deriving processes may be available. Provenance may be part of the local repository, or present in a separate or distributed catalog [46, 47].

Data accessibility refers to properties related to actually accessing the data for use, and includes permissions, intellectual property or resource costs (e.g. for proprietary or Cloud hosted data), and available bandwidth between the source and destination. Data movement service providers typically provide this information [34]. Lastly, social perception incorporates properties related to what peers or the community feels about the datasets, and includes the frequency of use of the data, the number of comments or “likes” on the dataset (to use a Facebook analogy), and whether it was used or generated by members of your professional social network [35]. This information can be sourced from social networks and community recommender systems [36].

Other information classes may exist and our model itself makes no presumption on the nature or number of these classes. A brief description of these classes with examples is given in Table 1. A formalism of the information model is presented in [37].

### 2.3 Quality Model

The quality model allows scientists to define personalized quality requirements for their scientific investigations in the form of a *quality profile*. The quality profile is described using a *quality constraint language* (QCL) that captures the quality predicates over the metadata schema for a dataset, and applies these constraints to evaluate the quality score for a dataset when its metadata instances are provided. The constraints use range and
comparison conditions on metadata properties, and also allow grouping of constraints, assigning them relative weights, defining pre-conditions for their evaluation, and applying complex functional transformations on metadata properties for specific domain needs.

This high-level, declarative, functional, typed language provides a model for specifying the quality profile. At the implementation level, this is mapped to an XML schema that allows the profile to be represented as an XML document. The language or its XML representation is not intended for direct definition by users but rather for translating into from profiles that are specified through a web-form.

2.3.1 Quality Constraints

All metadata properties are potential factors upon which the scientists may base their quality judgment of the data. The profile consists of quality constraints defined over the properties, and associates them with a relative weight and a quality sub-score. Users typically require metadata property values to lie within certain value ranges to meet the requirements of their application. Allowing them to associate a sub-score for each range of values that a property may take gives them fine-grained control over the subjective evaluation of the data quality score.

The model is independent of the scale of the quality score as long as it is an interval measurement that permits counting and ordering. We adopt a scale of $-7$ to $+7$ for our implementation. This choice of scale is arbitrary, but has no statistical bearing on the evaluation (though it may have a psychological impact on users defining profiles) and a scale of -100 to +100 or 1 to 10 is equally valid [55, 56].
For example, a sample quality constraint could depend on the publisher of a particular precipitation data, assigning a high quality sub-score of +7 when “NCSA” or “UniData” are the publishers and a low quality sub-score of -5 if “Acme Exports” is the publisher, and assigning a relative weight of 1 to this constraint. The XML representation of such a constraint is shown in Figure 1 for reference. Users would not see such an XML but would instead be guided through a web-form to select the metadata property and the constraint values, sub-score and weight, and the XML generated automatically.

More formally, constraints ‘R’ are specified over a typed metadata property ‘p’ with conditions ‘c’ or functions ‘f’ that map it to a tuple of relative weight ‘w’ and a quality sub-score ‘q’.

\[ R(p,c|f) \rightarrow <w, q> \]
Constraints conditions can be constructed to compare the property against a single value using standard binary operators such as $<$, $<=$, $=\equiv$, $\neq$, $\geq$, $>$, or to a value range. Comparison is done using lexical or numerical ordering depending on the property type. Substring, keyword and phrase comparisons are additionally possible with string types. Each constraint is associated with a quality score that is assigned to it if the comparison if successful. Multiple comparisons can be performed on a single property using an enhanced *switch-case* construct, with the switch operating on a property, the cases performing the binary or range comparisons, and the result being a score assigned for the constraint.

The model also supports more advanced user defined functions ‘f’ that evaluate a quality sub-score and a weight opaquely. This allows complex, domain specific property transformations or operations to be performed, and provides a generalized solution by allowing extensible, user defined functions to map a property value to a quality sub-score and weight. Some predefined functions include linear or polynomial scaling of numeric properties and counting the number of property values for multi-valued types. Other domain-specific functions may include, for example, calculating the standard deviation over a set of multi-valued properties that describe instrument measurements, with the standard deviation forming an estimate of their stability. Some functions may require the use of multiple property values to evaluate a meaningful quality score, such as, say, a fusion energy scientist applying a Hamiltonian score of momentum $m \times v$ to the metadata properties mass ($m$) and velocity ($v$). While the current model does not support such multi-valued functions, it can be incorporated into the model when necessary without much effort.
2.3.2 Quality Profile

The quality profile consists of a set of quality constraints. These can be grouped into *constraint sets* for easy management of the profile. For example, constraints related to spatio-temporal metadata properties can be grouped together into a set. Or those that are domain specific can be separated from those that are domain independent. The constraint sets also allow users to specify a pre-condition for evaluating the constraint set. While the quality profiles are primarily intended for ranking filtered results, such pre-conditions allow refined application of the constraints to datasets that match select metadata properties. Such preconditions are similar to constraints, with the exception that they evaluate to a Boolean true or false, rather than return a weight and quality sub-score. Also, nested Boolean filters are supported as pre-conditions using *and*, *or*, and *not* operators.

2.3.3 Quality Score Evaluation

A quality score can be evaluated for a dataset given a quality profile and the metadata property values for that dataset. The pseudo code for this evaluation process is given in Figure 2. Properties are first matched against pre-conditions for constraint sets (if any exist) and those that satisfy the condition allowed to proceed to the constraints within the set. The actual scores are assigned and aggregated at the granularity of the constraints. The properties and values that match the conditions of the constraint have quality sub-scores and weights assigned to them. Alternatively, a user defined function may transform the property and value to a weight and sub-score.
// Return quality score q for given metadata
// property values m and quality profile p

FUNCTION EvaluateScore (m, p) -> q

// initialize quality score output
set q = 0

// Initialize sum of constraint set weights
set wss = 0

// iterate thru constraint sets
foreach s in p.constraintSets do

    // Initialize weighted sum of sub-scores
    set qs = 0

    // Initialize sum of constraint weights
    set wcs = 0

    // evaluate precondition function
    if s.PreCondition(m) then

        // iterate thru constraints in set
        foreach c in s.constraints do

            // evaluate the quality sub-score and
            // weight for constraint and metadata
            eval c(m) -> (qc, wc)

            // Update weighted quality score for set
            set qs = qs + wc * qc
            set wcs = wcs + wc

        end foreach

        // Update weighted quality score for set
        set qs = qs / wcs

    end if

    // Update weighted quality score for profile
    set q = q + qs * s.weight
    set wss = wss + s.weight

end foreach

// Update weighted quality score for profile
set q = q / wss

end FUNCTION

Figure 2. Pseudocode for evaluating quality score for a dataset using a quality profile
The weighted sum of the quality sub-scores are added for each quality constraint in the set. Constraint sets themselves may have relative weights with respect to other sets. The overall quality score for the profile is evaluated as a weighted sum of quality sub-scores across all constraint sets.

While quality sub-scores for constraints allow users to discriminate within the value space for a metadata property, relative weights allow them to discriminate across metadata properties. So the scores allow them to specify that a dataset with a grid spacing metadata property set to 5Km is better than one set to 10Km, and that a dataset published by “Oklahoma University” is preferred to one from “National Center for Supercomputing Applications”. The relative weights lets them state that the grid spacing property is more important than the publisher of the dataset.

2.4 Discussion

The typed metadata model we adopt helps with accurate searching and ranking of the datasets. Requiring typed metadata typically does not pose an undue overhead on the data provider. Most scientific datasets already provide well typed, domain specific metadata schemas that are frequently represented using XSD or RDF. Also, while our metadata model seems limited by the lack of hierarchical structure, this is to simplify the model without loss of generality. We have observed that for most parts, the hierarchy present in scientific XML metadata is more for organizing the metadata coherently than for encoding semantics. As we shall describe in the architecture section, the metadata properties present within nested XML elements or attributes can be extracted to fit the <property, value(s)> tuple model we adopt.
The profile can be considered as a set of queries that capture different input data requirements for scientific applications along with associated suggestions for ranking. The two main differences between our profile approach and a traditional keyword search approach are the richness of *query expressibility* and the ease of *management*. While not as powerful as a query language like SQL or XQuery, the quality profile is more intuitive for expressing quality requirements in a declarative manner and aptly captures the functional nature of quality score evaluation. Query constraints based on binary comparisons along with filter preconditions are as powerful, if not more expressive, than keyword searches. In addition, using typed metadata allows more accurate definition of queries. The use of generic or domain-specific external functions gives a great degree of control to the scientists to perform complex lexical or mathematical comparisons and ranking of datasets. The ability to assign relative weights across metadata properties gives a simple, yet powerful way of specifying user needs.

As such, each group of quality constraints is evaluated independently of each other. This allows conflicting constraints to be specified in the profile. While this may not be meaningful or may be unintentional on the user’s part, since the constraints evaluate to a numerical score, the presence of conflicting constraints will just lead to the numerical scores to cancel each other out, and not cause ambiguous evaluation.

The hierarchical grouping of the quality constraints in the profile eases management of profiles for a scientific application. We envisage scientists defining a single profile for each of their application, and using preconditions to group the quality needs for each data type required by that application. All constraint sets are uniquely identified, allow them to be easily reused in other profile descriptions. Indeed, curators and domain data
managers can define common data quality profiles for frequently used scientific applications that other users can derive from. This reduces the overhead for describing the profiles and yet provides users with personalized ranking through their modification to the profile. Reusing parts of the profile also optimizes the evaluation of quality scores through intelligent caching of the scores that is done at different stages of a profile evaluation.

3. USING PROVENANCE TO PREDICT QUALITY SCORE

Scientific tasks can be considered as a quality function that take input data of a certain quality and other input parameters, and produce output data of a certain quality. Metadata describing the input and output data products may capture the quality of the data through their metadata properties, if not explicitly, indirectly in the form of quality scores evaluated over those metadata.
For example, Figure 3(a) shows the quality score evaluated for three data products $A$, $B$ and $C$ using their metadata properties $M_A$, $M_B$, and $M_C$ and the quality profile $P_{\text{User}}$ for a user, as described in the previous section. Now, if the data product $C$ had been derived by a scientific task $T$ using data products $A$ and $B$, and a value parameter $i$ as inputs (Figure 3(b)), it is possible that there exists a quality function $F_T$ for the task $T$ such that the quality score for data product $C$ can be calculated from the quality scores of the input data products and the value parameter(s) to the task (Figure 3(c)). Thus, even in the absence of metadata describing data product $C$, it is possible to evaluate its quality score.

If the quality score for data product $C$ is given by:

$$Q_C = P_{\text{user}}(M_C)$$

And if the data product $C$ was derived from a task $T$: 

Figure 3. Relationship between (a) Quality score evaluation using metadata and user quality profile, (b) Provenance of workflow task execution, and (c) Quality score evaluation using input quality scores and quality function.
Then, it is possible to calculate the quality score for \( C \) using a quality function \( F_T \) as:

\[
Q_C = F_T(P_{user}(M_A), P_{user}(M_B), i)
\]

For a given task \( T \), the quality function \( F_T \) is unique for each of the output datasets in the task signature. For e.g., if task \( T \) generated two datasets \( C \) and \( D \) in the example in Figure 3(b), then there would be a unique \( F_C^C \) and \( F_D^D \), but these would remain the same across all instances of execution of the task \( T \), i.e. for \( T_1(A_1, B_1) \rightarrow (C_1, D_1) \), \( T_2(A_2, B_2) \rightarrow (C_2, D_2) \), and so on.

We refer to this quality score as the provenance-based quality score since the function \( F_T^C \) uses the provenance information of the target dataset \( C \). Through the above process, we have shifted from a quality score evaluated using the metadata properties of the output data to one that is based on the metadata properties of the input data and uses the provenance driven quality function. This is the key idea behind using provenance to predict the quality score for derived data in the presence of missing metadata.

The success of this depends on two things: one, whether the task deriving the output data does indeed operate as a quality function whose output data quality depends on its inputs, and two, if it is possible for us to discover this provenance quality function \( F_T \) that will accurately model the output data quality score using the input metadata properties. The first question will depend on the domain and applications within that domain that are being used as the workflow task. Many computational and statistical applications fall in this category, as do those that are functional in nature. A survey of such applications is outside the scope of this article. However, as described later in Section 5, we address the second question by using machine learning techniques to model such a quality function,
if it exists, as a classifier that is trained using historical provenance and prior quality scores for the input and output data products.

4. QUALITY BROKER ARCHITECTURE

Our prototype quality broker service implementation, Guna, evaluates quality scores for datasets for specific quality profiles. The scores are used to rank the dataset search results. The Guna implementation is modular to allow easy access to metadata properties from diverse online information providers. Wrappers or adapters we refer to as *transmogrifiers* [57] (Figure 4) are used to access specific information sources and map the metadata properties to a normalized XML form that can be processed by the quality profile. These wrappers are implemented as Java libraries implementing a specific interface that can be used by Guna to access specific metadata properties supported by that information provider. The quality profiles for a user is translated to an XQuery, as described later, which operates on the XML metadata returned by the transmogrifiers for each dataset and returns a quality score XML document for the datasets when evaluated by a Saxon XQuery processor engine.

4.1 Metadata Properties

The quality profile is represented in XML and users initially register their quality profile documents with the broker. The profile uses a special URI scheme to refer to metadata properties. This scheme allows the URI to be mapped to an XPath that accesses the particular metadata property in the external information provider. For example, in the FGDC XML metadata schema used in the LEAD meteorology project, the “origin” metadata property can be accessed using the following URI notation.

```
guna:///intrinsic#/lead:LEADResource/data/citation/fgdc:origin
```
The prefix ‘guna://intrinsic’ refers to the intrinsic metadata for that domain, while the underlined suffix is the XPath that accesses the specific metadata property. This scheme also allows additional transformation functions to be included that help it map non-XML metadata sources back to metadata properties used within the quality profile. The transmogrifiers are responsible for performing this mapping.

4.2 XQuery Representation of Profiles

The quality profile evaluation is functional in nature. The profile is evaluated by translating it into an XQuery that operates over XML metadata available for the dataset. This helps avoid inventing a new query evaluation engine while leveraging the richness of the XQuery language and maturity of XQuery engines. Metadata properties referenced
using XPaths encoded within the URI can be natively expressed in the XQuery. Other metadata properties are evaluated by calling the Java transmogrifiers from the XQuery. Quality constraints and filters are mapped to XQuery operations while scores are assigned, scaled, and aggregated using XQuery’s numerical functions.

4.3 Dataset Identification

The broker exposes a web service interface to evaluate the quality score for one or more datasets for a given quality profile. The dataset parameter(s) are passed as one or more GUIDs, and the profile is provided as an inline parameter or using the registered profile identifier. The GUID for the dataset is used to retrieve metadata related to that dataset from multiple information providers using the wrappers. The broker is agnostic to the naming scheme adopted by that particular scientific domain as long as a name resolution service is accessible to the relevant wrappers.

4.4 Quality Score Document

The result of the XQuery evaluation is an XML document that gives the quality score and weight for each quality constraint, and the overall quality score value. Returning such fine-grained scores has two advantages. One, it allows scores at the level of constraint sets to be cached for a dataset using the constraint set and dataset GUID as a compound key. Two, it provides the rationale for the score assignment to the user and allows further client-side filtering or post-processing.

4.5 Quality Score Evaluation Walkthrough

Figure 4 shows the Guna quality broker web service architecture. The numbered circles refer to the execution order when evaluating the quality score for a dataset and user profile. The user first invokes the broker service and passes a dataset GUID and
registered quality profile ID (step 1). The broker retrieves the XML profile from persistent store (step 2) and transforms the profile into an XQuery for that quality score evaluation, embedding the dataset GUID and other dynamic parameters into the query (step 3). The query is sent to a Saxon XQuery engine for evaluation (step 4).

The query contains several calls to external Java wrappers (transmogrifiers) to retrieve specific metadata properties or evaluate domain specific functions used in quality constraints. This allows only the subset of the metadata relevant to the profile evaluation to be retrieved from remote catalogs and passed to the XQuery engine. Saxon invokes these external functions as part of the query execution (step 5), and these functions include accessing intrinsic metadata (step 6), provenance metadata (step 7a), social perception details (step 8a) and quality of data access properties (step 9). Evaluation of some of these functions, such as the provenance-based quality score using the provenance transmogrifier can trigger a recursive callback to the quality broker to get, say, the quality scores of the input dataset used by the provenance-based quality function classifier (step 7b), or it can cause the social perception transmogrifier to invoke the broker to evaluate the average quality score assigned by other peers of this particular user (step 8b). The provenance transmogrifier has access to the Weka machine learning engine and a persistent store of provenance-based quality function classifiers built earlier (step 7a,b). This process is described further in Section 5.

The result of the query is a quality score XML document that contains the overall quality score for the dataset and the constituent quality scores and weights for the constraint sets in the profile. This score document is cached in a quality database (step 10) and returned to the user (step 11). The cached values can be used in the future in case the associated
metadata properties for the dataset have not changed. Caching is possible at the granularity of constraint sets. So this allows only the “dirty” portion of the quality profile to be reevaluated later.

4.6 Application to Earth Sciences Domain

Guna was developed for the Linked Environments for Atmospheric Discovery (LEAD) meteorology project. As part of the implementation, metadata wrappers or transmogrifiers are available for scientific datasets in a cyberinfrastructure environment, and in particular, for the earth sciences domain. The Intrinsic Metadata Transmogrifier is used to access spatio-temporal XML metadata [38, 39] from the myLEAD personal catalog [40] and the LEAD Data Catalog that harvests public NASA and NOAA datasets. Intrinsic metadata properties are identified in the URI using an XPath into the domain XML schema.

The Provenance Transmogrifier is used to retrieve provenance metadata collected for datasets derived from workflows running in the LEAD cyberinfrastructure. The Karma provenance service [41] records and publishes data, process and workflow provenance as XML documents that this wrapper can access. Provenance properties, such as the application deriving the dataset or its input parameters, can be used directly by the scientist in their quality profile. In addition, computed properties, such as the frequency of use of a dataset, are available as functions that post-process provenance metadata.

The Quality of Access Transmogrifier interacts with network monitoring services, such as Network Weather Service [34], and data transfer services like Reliable File Transfer (RFT) [42] to present users with metadata properties describing aspects of data access bandwidth, authorization, availability, economic cost, transfer duration, and reliability.
These are available as external functions with qualified names that users can use in their profile. A Social Perception Transmogrifier provides aggregates of quality scores assigned by other users’ profiles to the dataset. So a user can define a quality constraint based on the average quality score assigned by other users with a certain level of expertise. Future enhancements can incorporate other community perception properties made available through social networking services and recommender systems.

A naming service based on the Replica Location Service (RLS) [43] is available to link globally unique identifiers present within the different forms of metadata to the logical datasets and their physical replicas.

5. PROVENANCE BASED QUALITY SCORE IMPLEMENTATION

Guna implements the provenance-based quality function we propose in this article as part of the Provenance Transmogrifier and makes this predicted quality score available to the quality profile just like any other metadata property. This provenance-based quality score can be used in the profile and returned, say, when metadata properties for a dataset are missing. Despite the simplicity of accessing the provenance-based quality score, several complex operations occur just-in-time in the background when this quality score is accessed during profile evaluation.

5.1 Constructing the Provenance-based Quality Function

The provenance transmogrifier builds a quality function for the workflow task that derives the target dataset. This function is modeled as a classifier using machine learning algorithms that are evaluated using the Weka engine [44]. The classifier quality function takes as input the quality scores of the input datasets and the input value parameters to the task, and returns the quality score of the target output dataset.
For a given task $T$, a quality function $F_T^C$ has to be constructed for each output dataset $C$ in its signature. To build the quality function using machine learning, we need to have a set of samples that can be fed to the learning algorithm. These samples should be previous executions of the task $T$, say $T_1, T_2, \ldots, T_n$ that produced outputs $C_1, C_2, \ldots, C_n$ using inputs $(A_1, B_1, i_1), (A_2, B_2, i_2), \ldots, (A_n, B_n, i_n)$, in Figure 3(b). The samples themselves should not have any missing metadata for input or output dataset. This ensures that the quality scores can be calculated for the input and output datasets that will be used by the learning algorithm to build a classifier to model the quality function. These samples are essentially the provenance records for the different executions of the task $T$.

When the provenance transmogrifier needs to build a Weka classifier initially for a specific output dataset for a task, it first retrieves the process provenance XML documents for all instances of that task from the Karma provenance service. It eliminates provenance records whose input and output datasets have missing metadata, and calculates the quality score for each input and output dataset using the current user’s quality profile. Value input parameters such as strings, floats and Booleans are mapped to a numerical or ordinal value space since the classifier functions only accept numbers of ordinals as input. While numerical and Boolean values can be directly used as cardinal features by Weka, one current limitation is that arbitrary (nominal) strings present in the input need to be mapped to a ordinal form, such as an enum, before they can be used.

The set of samples of the form $([Q_{A1}, Q_{B1}, i_1] \rightarrow Q_{C1}), ([Q_{A2}, Q_{B2}, i_2] \rightarrow Q_{C2}), \ldots, ([Q_{An}, Q_{Bn}, i_n] \rightarrow Q_{Cn})$ are passed to the Weka engine for training a classifier. Here, the input features, show in square brackets ‘[]’, are used to train the output quality score feature.
The $Q$’s represent the calculated quality scores for the input (A,B) and output (C) datasets to the task $T$, while the $i$’s are cardinal or ordinal values present in the input. Each task uses a different set of input features for training depending on the number and type of inputs to that task.

A set of standard machine learning algorithms, such as decision tree, MP3, KStar, and so on, are used by the Weka engine to construct the classifier function $F_T^C$. The classifier built using each machine learning algorithm is tested to determine the one that provides the best predictions. We use the least Root Mean Squared (RMS) error in prediction as a measure of the effectiveness of the prediction, though other statistical and machine learning metrics can be used. This dynamic selection of machine learning algorithm for classification for each task helps to model the quality function accurate for the available set of provenance samples for that task.

Given that a unique classifier has to be built for each task for each user profile (which are liable to change), the time to build the classifier can be a scalability concern. We have observed that Weka takes about 2secs to build a classifier using input 25 samples and 3secs using input 100 samples, assuming all input samples are available. Since constructing the classifier can be done independently for all users and tasks, this step is trivially parallelizable and can be done offline.

5.2 Evaluating the Provenance-based Quality Score

Once constructed, the classifier can be used to predict the quality scores for derived datasets whose metadata is missing. To get the provenance based quality score for a dataset instance $C_x$, the wrapper first retrieves the data provenance for this dataset from the Karma service. It then identifies the task $T$ that generated this dataset and selects the
classifier \( F_T^C \) previously created for this task. Next, it calculates the quality scores \((Q_{A_x}, Q_{B_x})\) for the input dataset instances \((A_x, B_x)\) used by this execution of the task using the user’s quality profile and the metadata properties for these input datasets. The Weka engine is then used to evaluate the provenance quality function \( F_T^C \) by passing it the input dataset quality scores and the input value parameters, and the function returns the predicted quality score \( Q_{C_x} \) for the to the output dataset.

As mentioned before, this quality score can be accesses as part of the quality profile as a metadata property available through the provenance transmogrifier, and either returned directly as the result of the quality profile evaluation of dataset \( C_x \), or used in combination with partially available metadata for \( C_x \).

6. EXPERIMENTAL EVALUATION

We evaluate the effectiveness of the provenance-based quality function classifiers in predicting the quality scores for datasets with missing metadata. For this experiment, we use a scientific workflow that processes and transforms fMRI brain images, that was used as a reference workflow in the first and second Provenance Challenge workshops [45]. We assume that each workflow task has a particular quality behavior, i.e. the quality of the output data it generates depends of the quality of the input datasets. We model this as a specific mathematical function associated with each task that behaves as a blackbox quality function internal to the task. This function operates on the quality scores of input datasets and generates an output dataset with the resulting quality score, for a user quality profile we define for the experiment. Since our assumption that the task behaves like a quality function may not always hold true, we introduce randomness into some of these functions.
Figure 5. Sample workflow with deriving processes that use different output quality functions on input data. In each case, the shaded dotted box gives the mathematical function that the process simulates as the output data’s quality function.

Figure 5 shows the workflow and the math function that represents each task’s quality behavior. These functions range from linear combinations of input data quality scores and parameters (atlas_xpgm), to exponential (resliced_header), and random generators (atlas_xgif).

For these experiments, we run the workflow 1000 times to synthetically generate the provenance and additional metadata for the input and output. Each workflow execution uses an initial set of input datasets with a random quality score between -7 and +7. As the workflow executes, each task generates output datasets with a quality score determined by its internal quality functions. For example, in Figure 5, the metadata for the four inputs
to the AlignWarp task, a, b, c and d, may have random quality scores of +3, +2, –4 and –5 for an execution run. Since AlignWarp’s internal quality function is a linear combination of these inputs, its derived output data, w, would be generated with a quality score of: 

\[ (1.5 \times 3) - (2.5 \times 2) + (3 \times -4) - (2 \times -5) = -2.5. \]

6.1 Resilience towards Missing Metadata

In the first experiment, we predict the quality score for a set of derived datasets that do not have any metadata except for their provenance. We build and use the provenance-based quality function classifier for predicting these scores for 100 datasets, and compare the predicted score against the expected quality score according to the internal quality functions. We define two profiles, one that uses only intrinsic metadata properties to evaluate the quality score while the other uses only the provenance-based quality score. The former uses the results of the internal math functions to determine the real quality
score while the latter uses our classifier models. The absolute and root mean square error between the real and predicted quality scores are measured.

Figure 6 shows the absolute errors in the provenance-based quality score predictions averaged for 100 datasets. The datasets are grouped by the task that derived them along the X axis. The machine learning algorithm used by each of the provenance-based quality function classifiers is also shown above each bar. As can be seen, the algorithm used for the tasks varies according the one that can best model the quality function for the training samples provided.

We see that the absolute values of prediction errors are between 0 and 2, on a quality score scale of -7 to +7. For this scale, the mean absolute error when using a random quality score assignment would be 7. So our absolute error values translate to relative errors of between 1 – 29%. For the majority of the functions, this value is below 7%. The RMS errors observed (not shown) were between 1 – 23%.

6.2 Prediction Accuracy with Number of Samples

In the second experiment, we evaluate the effect of the sample size used in training and building the classifier on the prediction accuracy. We use sample sizes between 10 and 200 provenance records for building the classifiers and measure the absolute prediction errors for trained classifier.

Figure 7 shows the absolute errors on the Y axis for different workflow tasks on the X axis, and for different training set sizes on the Z axis. As can be seen, increasing the number of samples broadly reduces the absolute error for the classifier for each task. We also observed a similar trend for the RMS errors (not shown). This suggests the use of more samples to build the classifiers.
For some tasks, such as resliced_header_1, there is a sharp improvement initially but this plateaus out. For others like align_warp_1, the improvement is steady. Also, it should be noted that for each set of training data, the machine learning algorithm used by the classifier may be different. So atlas_header may use a decision tree for a training set size of 10 provenance records, and switch to a K-Star algorithm for 100 provenance records if it gives a lower RMS error. Our implementation occasionally rebuilds the provenance-based quality function classifier as more provenance records are available for a particular task as it is executed repeatedly over time.

7. RELATED WORK

Information and data quality have been well studied, and various metrics proposed for estimating the quality of data [21, 22, 48, 50]. Some of these concepts have also been used in web searching and ranking [49, 50]. Several of the information sources incorporated in the Guna framework covers those proposed quality metrics. In addition,
we introduce provenance as a quality metric that can be explicitly used by the user in their profile. Our broker framework and profile are extensible, and allow additional metadata sources that denote quality to be added as required by the domain.

Many search engines provide personalized search results. Since web search is typically free text search, the personalization is either learnt from prior searches or explicitly provided as a set of interest categories as part of the users profile [23, 51]. We use the latter model of using profiles. However, since our framework operates on structured metadata that has rich domain information, our quality profiles allow users much more control and flexibility.

The association between provenance and quality has been regularly highlighted [53, 25, 46], and the effect that provenance has on a user’s trust – one of the metrics of quality – in scientific data illustrated [54]. However, there is limited work on how this can be used in a meaningful estimation of data quality. Hartig, et al. [52] proposes to annotate provenance with impact values in order to perform quality assessment and generate an information quality score. Their preliminary work uses the time of creation of data as an impact factor that affects the timeliness quality of data. Our quality framework is more expansive, and incorporates a diverse set of impact factors – what we refer to as the metadata quality factors used in the constraints, including data accessibility, social perceptions and so on. Also, in our approach, the users declaratively specify a quality profile document that is evaluated into a score, rather than annotate provenance with impact factors. This separates the personalized notion of quality for a user, from the metadata recoded and shared by a community. Also, the quality scoring is personalized and we do not prescribe any objective notion of quality such as the timeliness model
proposed. Lastly, [53] proposes alternatives for information quality scoring when provenance is missing. Our model, conversely, uses provenance as a proxy when other metadata are missing.

8. CONCLUSION AND FUTURE WORK

In this article, we describe an architecture for personalized quality scoring of online scientific datasets that allows scientists to rank datasets and perform more effective data discovery. These quality scores evaluated for individual user profiles can be used to automate data selection for workflows by picking those with the higher quality score (like the “I’m feeling lucky” button in Google search), and highlight datasets of interest from the thousands that are published each day.

In particular, we show the use and effectiveness of provenance in compensating for missing metadata as is common when using third party datasets. While the accuracy of our prediction model was demonstrated for several types of tasks in our experiments, it may not always be possible to model a quality function using a classifier. Hence, the users or the quality framework need to decide on its value and assign appropriate weights to this in the profile.

The accuracy of the provenance-based quality score can improve significantly as more provenance samples are available for training classifiers, as expected. However, we also observed that an increase in sample size causes a longer classifier build time and query response time. Hence, the quality framework needs to make intelligent decisions on the right sample size to train upon. Also, the classifiers currently do not consider the impact of error propagation in case we need to recursively build classifiers and generate quality
scores for across multiple levels of missing metadata. We hope to extend our current research to address these issues in future.

The broader use of such a quality framework will depend on several factors, such as the presence of tasks that can be modeled as quality functions and the ease of use for users in adopting quality profiles. However, it highlights the importance of provenance collection and publishing as part of the dataset’s metadata, and translates the broadly held intuition of provenance providing a sense of the data quality into a practical and quantitative model for scoring data quality based on provenance.

REFERENCES


[14] Luciano Barbosa and Juliana Freire. 2007. Combining classifiers to identify online
(WWW '07)

129-138.

[16] Bin He, Mitesh Patel, Zhen Zhang, and Kevin Chen-Chuan Chang. Accessing the

[17] Workflows and e-Science: An overview of workflow system features and
capabilities, Ewa Deelman, Dennis Gannon, Matthew Shields and Ian Taylor,

[18] Workflow-Level Parametric Study Support by MOTEUR and the P-GRADE Portal,
Tristan Glatard, Gergely Sipos, Johan Montagnat, Zoltan Farkas and Peter Kacsuk,

[19] Yolanda Gil, Ewa Deelman, Mark Ellisman, Thomas Fahringer, Geoffrey Fox,
Dennis Gannon, Carole Goble, Miron Livny, Luc Moreau, and Jim Myers. 2007.
Examining the Challenges of Scientific Workflows. Computer 40, 12 (December
2007), 24-32.

Ramanath, and Gerhard Weikum. 2008. NAGA: harvesting, searching and ranking
knowledge. In Proceedings of the 2008 ACM SIGMOD international conference on
Management of data (SIGMOD '08).


[33] Yogesh L. Simmhan, Beth Plale, and Dennis Gannon, Towards a Quality Model for Effective Data Selection in Collaboratories, in International Conference on Data Engineering Workshops (ICDEW) ??? Scientific Workflows and Dataflows (SciFlow)


[35] Yogesh Simmhan and Karthik Gomadam, Social Web-scale Provenance in the Cloud, in International Provenance and Annotation Workshop (IPAW), 2 June 2010


[37] Chapter 3: Quality Model, Yogesh Simmhan, in Provenance framework in support


[45] Luc Moreau, Bertram Ludaescher, Ilkay Altintas, Roger S. Barga, Shawn Bowers, Steven Callahan, George Chin Jr., Ben Clifford, Shirley Cohen, Sarah Cohen-Boulakia, Susan Davidson, Ewa Deelman, Luciano Digiampietri, Ian Foster, Juliana


[55] Do data characteristics change according to the number of scale points used? An experiment using 5-point, 7-point and 10-point scales, J. Dawes, International Journal of Market Research Vol. 50 Issue 1, 2007.
