The F4U System for Understanding the Effects of Data Quality

Daniele Foroni
Huawei ERC
daniele.foroni@huawei.com

Matteo Lissandrini
Aalborg University
matteo@cs.aau.dk

Yannis Velegrakis
University of Trento and
Utrecht University
i.velegrakis@uu.nl

***Abstract*—We demonstrate a system that enables a data-centric approach in understanding data quality. Instead of directly quantifying data quality as traditionally done, it disrupts the quality of the dataset and monitors the deviations in the output of an analytic task at hand. It computes the correlation factor between the disruption and the deviation and uses it as the quality metric. This allows users to understand not only the quality of their dataset but also the effect that present and future quality issues have to the intended analytic tasks. This is a novel data-centric approach aimed at complementing existing solutions. On top of the new information that it provides, and in contrast to existing techniques of data quality, it neither requires knowledge of the clean datasets, nor of the constraints on which the data should comply.**

I. Introduction

To have confidence in any data-driven decision making, there should be trust in the data on which the decision is based. For this trust to materialize, the data has to be of high quality. Unfortunately, real-world datasets have many data quality issues: they are missing values, they are violating constraints, or they provide inaccurate measures. Such errors cost U.S. businesses alone over 3.1 trillion dollars a year [7], and fuel a significant interest in the problem [1], [5]. Many tools have so far been developed trying to quantify the different data quality dimensions in a dataset [1], [10], repair them [4], [3], or develop techniques immune to these issues [6], [9].

It has already been recognized that a dataset is of high quality if it "fits its intended use" [13], yet existing approaches do not consider the task and evaluate the data quality independently of the task [1], [4]. Another limitation is related to how data quality is evaluated. Most of the existing data quality techniques are based on the idea that there is a clean dataset, i.e., the one that models the reality accurately, and they try to measure the data quality by quantifying the difference between the available dataset and the clean one [10]. Usually, the clean dataset is not available or known, as if it were, there would have been no purpose not to use that directly. In these cases, a solution often followed is to test for violations of properties known to hold in the clean dataset, e.g., functional dependencies, and use these as an indication of the quality of the dataset [6].

We demonstrate a novel system for measuring the sensitivity to data quality, called **F4U**, which stands for **F**itness for **U**se, to highlight the fact that the system aims at helping the user understand how good a dataset is for a specific analytic task. We claim that to achieve such a goal it is not enough to provide the user only with information on how much the dataset differs from the perfect (clean) dataset, but also quantify how that difference affects the outcome of the task for which the dataset is to be used. Furthermore, given the fact that many datasets are dynamic, the difference from the clean dataset may change over time. Thus, we believe that it is equally important to provide the user with an idea of the effect on the analytic task results to be expected for other instances of the datasets that may materialize in the future. Empowered by this type of characterization, analysts may prioritize cleaning tasks, or even decide to avoid some of them altogether, reason about the reliability and robustness of their insights, and have a better overall understanding of the actual effect of existing or possible data quality issues on the analytical processes. The way our tool works is that it takes a dataset and an intended task, and it returns: (1) the traditional quality evaluation, as produced by traditional data quality systems; (2) a set of correlation factors that indicate the degree in which the results of the specific task are affected by different levels of variation from the given dataset.

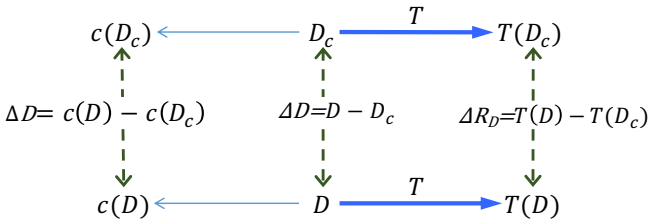

Fig. 1. Data Quality Revisited

Note that the kind of information F4U can provide is valuable even in the case of a clean dataset. In such cases, existing data quality tools will simply indicate that there is no problem, but will not provide any indication on what could happen in the case in which specific quality issues appear. F4U, instead, offers this kind of information to the user, so that the user is aware and possibly prepared to deal with these issues when they appear.

II. Data Quality Extended

Let D_c be a dataset that accurately represents reality (*clean*) and D the dataset that is actually available (*dirty*). Traditional approaches quantify data quality by measuring their difference $\Delta D{=}D{-}D_c$. What should be of interest is how the results of a task applied to these datasets differ. Let T be such a task, and

$T(D)$ denoting the results of the task T applied on a dataset D (and $T(D_c)$ those on D_c, respectively). The effect on the results of the analytic task if the dataset D is used instead of D_c is ΔR_D=$T(D)-T(D_c)$. The right-hand side of Figure 1 graphically depicts this data quality approach.

Traditionally, data quality is quantified as the difference ΔD=$D-D_c$. In practice, of course, as it has already been mentioned, the clean dataset D_c is rarely known. In such cases, some function c that is known to have a 0 value for D_c may be used to quantify the difference. Then, ΔD=$D-D_c$ $c(D)$-$c(D_c)$ = $c(D)$-0 = $c(D)$. Examples of such functions are the set of constraint violations or the number of missing values.

To better understand the quality of a dataset, we believe that it is essential to not only know the ΔD but also how much of ΔR_D is created as a result of that ΔD since at the end it is the results of the analytic task that one cares about. Hence, we expect as a data quality indication to provide not only the ΔD but also a factor that indicates the effect on the ΔR_D. What F4U does is to compute the traditional data quality and in addition to compute that factor and provide an informative presentation of the possible effect.

III. The F4U System

Apart from measuring the quality of a dataset as it is done traditionally, F4U discovers the factor that determines the effect of a data quality issue on the results of a specific task. To do that, F4U modifies on purpose the data quality of the dataset in a controlled and systematic way and measures the effect such a change has on the results of the task at hand.

The change in the dataset data quality is done by introducing into it different kinds and amounts of noise. The triple $\langle$*noise type, noise amount, resulting* $\Delta R_D\rangle$ is referred to as a scenario. Once multiple scenarios have been created, they are grouped by noise type. For each noise type, the system computes a correlation factor between the specific noise type and the expected change in the task results. Then, instead of presenting the user with simply the amount of data quality issue the current dataset has, it additionally presents the computed correlation factor for the task at hand, alongside a graph that indicates the amount of change at different data quality levels. The right-hand side of Figure 3 illustrates how these results are presented to the user.

Note that F4U provides useful information even in the case in which the clean dataset is not available. The effect of some data quality issue (i.e., of artificially introduced noise) is measured as a difference between the task results on the dataset with that noise and the task results on the dataset without it. This means that even if the data quality of the current dataset cannot be computed because the clean dataset is not available, F4U can still produce and provide to the user information on what changes to expect on the results of the analytic task in the case that some quality issue increases its presence in the dataset.

Figure 2 illustrates the different F4U components that implement this approach. Parallelograms represent components, while the cylinders represent data. F4U can apply a series of

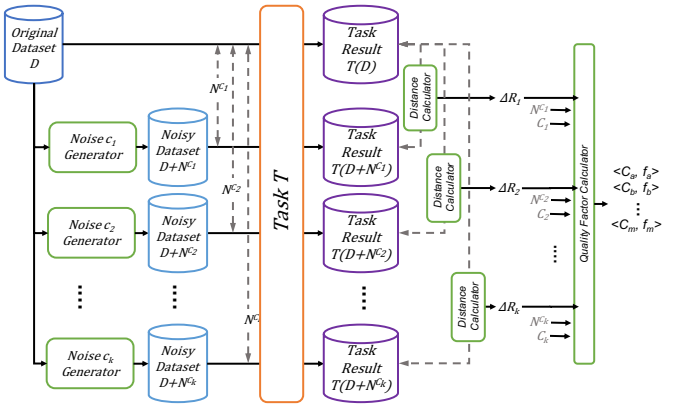

Fig. 2. The F4U System Architecture

noise generators that are responsible for introducing noise in a dataset (Section III-A). Hence, they return a noisy dataset. Each noise generator introduces a specific type of noise and can be parametrized to introduce a specified amount of errors. Another module (the *TaskT*) is responsible for obtaining a dataset and performing the analytic task of interest on it. The outcome of this module will then be compared against the outcome of the initial dataset to identify the variation (ΔR_D) that occurred as a result of the introduced noise. The computation of ΔR_D is a duty of the *Distance Calculator* module (Section III-B). The different noises and the ΔR_D produced are then provided to the *Quality Factor Calculator* that computes the correlation factors for the different noise types (Section III-C).

A. Noise Generators

F4U has a built-in error generator system [2] that comes with a broad spectrum of noise generators covering noise types mentioned in the literature and practical situations (with an approach similar to perturbation analyses over SQL queries [12]). They impact a defined percentage of the dataset and can be classified into three categories: the *universal* generators that perform basic modifications on the source data values without considering types or semantics, the *type-specific* generators that are type-aware and modify the representation of the data, and the *composite* that combines all the different other types. They are the following.

[Universal] *Null* makes parts of the dataset unknown by converting them to *null* values. *MissingInfo* removes parts of the dataset to increase its incompleteness. *Edit* performs insertions, deletions, and substitutions on parts of the values that exist in the dataset. *Permutation* scans the content and reproduces some permutations of consecutive symbols. *Abbreviation* turns some data values from the dataset into abbreviations, by considering the first letter of the selected words. *FD Violation* scans the dataset for records with the same values in the attributes specified in the head of some functional dependency and modifies the values of the attributes specified in the body, to ensure that the dependency is violated.

[Type-specific] *Shuffling* chooses randomly two consecutive words and swaps them. *Acronym* replaces values consisting of

more than one word with the initials of these words. *Synonym* replaces values in the dataset with their synonyms. *Multilingual* substitutes words with their respective counterparts in a different language. *Base Change* changes the base of the numbers, e.g., it turns a decimal number to its hexadecimal representation. *Scale* scales up/down numerical values by multiplying or dividing them by a factor. *Negator* negates a portion of the values by turning them from positive to negative or vice-versa. *Modification* adds or subtracts some fixed value to numbers in the dataset.

[Composite] In real datasets multiple types of noise are present altogether. F4U has this special meta-generator that applies a sequence of the generators mentioned above to a dataset so that the output of the application of one serves as input for the application of next. In this way, the generator produces results with mixed noise type and amount, as in the real datasets.

[Noise pattern] The user has full control over the noise generated in the dataset. F4U supports multiple distribution functions, e.g., uniform distribution, normal distribution, etc., that can be used in the noise generation process. The system allows the user to define portions where the noise has to be introduced or untouchable parts. Furthermore, it is possible to use data profiling tools that identify the patterns of noise that exist in certain real datasets and feed these recognized patterns to F4U in order to generate noise with similar characteristics.

B. Measuring Task Result Effects

To measure the change in the results of an analytic task, there is a need for a distance metric. Unfortunately, there is no universally accepted method for that. For well known analytic and mining tasks, it is possible to exploit metrics that have been explicitly proposed for their evaluation. For instance, the sum of squared errors, the silhouette coefficient, or the Folwkes-Mallows score are metrics used for measuring the degree of success of clustering; the accuracy, the precision, and the F1 score are used for classification; while the RMSE and the RMAE are used for regression. Measuring the change in the values for these measures can serve as a metric for measuring the change that took place between two task outputs. The above metrics are all supported in the F4U framework.

Moreover, for the cases in which no specific distant measure is available for the chosen task, F4U comes with a specific distance metric implementation, based on the difference in the instance values of two datasets. The method considers every task result as a dataset itself. Then, it creates a bipartite mapping between the two results dataset. The algorithm measures the distances of the instances of the two datasets. In this step, any distance metric, e.g., Jaccard distance, can be used to provide a numeric indication of the difference between the two result instances. Establishing the bipartite matching is the most challenging step of the algorithm, mainly due to the size of the task results and the many different alternative matches that have to be considered. F4U has three different methods for computing it. The first is the *greedy* method, with the complexity of $\mathcal{O}(n^2)$ to the size of the task results. This method allows the exploitation of a distributed infrastructure, enabling a faster computation. The second is based on the *Hungarian algorithm*, which provides the best exact matching but is slower than the greedy approach($\mathcal{O}(n^3)$ complexity). F4U applies a third method, an *auction-like algorithm* that is a trade-off between precision and the execution time for finding the matching, even if the complexity is still $\mathcal{O}(n^3)$.

C. Calculating Sensitivity Factors

Knowing the difference in the task results that has been caused by the different amounts and kinds of noise, the Quality Factor Calculator module can compute the sensitivity factors between the noise and the task results, for each type of noise. To compute the correlation, F4U is employing three different algorithms. The first is the linear regression, which can help in predicting the robustness of the task results for future kinds and amount of noise that may occur in the dataset. The second is the polynomial regression that can capture a polynomial relationship between the percentage of noise introduced in the data and the difference that the noise produces in the task results. The last is the Spearman correlation [11] that assesses monotonic relationships even if they are not linear. It indicates how much the two variables are correlated and how much they have a common increasing or decreasing monotonic trend. All the three factors are displayed, to let the user understand, along with the actual plot of the noise, which is the degradation level of the results due to the presence of noise.

IV. The Demonstration

[Goals] The goal of this demonstration is to illustrate that for measuring data quality, it is not enough to only count the amount of noise that is present in a dataset, but also consider the task for which the dataset it is to be used and how much the noise affects its results. The message the demo tries to pass to the participants is that data quality is not a single number but a pair, consisting of the noise plus the correlation factor. Another goal of the demonstration is to illustrate the approach for computing this correlation, highlighting the challenges that this process entails, in particular, the computation of the distance between different result sets and the combination of these distances to generate the correlation factor.

[Audience] The demonstration is intended for every data practitioner that deals with large volumes of data and needs to ensure that the results of any data analysis performed on them are as reliable as possible. The demonstration is also intended for data researchers, since it illustrates that, despite the massive amounts of efforts that the data management community has put on the topic and the significant results that have been achieved, the problem of data quality is still away from being considered solved. Considering data analytic tasks makes data quality a very different problem that simply considering it for query answering. Thus, the demo is intended not only for people working on data quality but for every person that in one way or another deals with real-world data that are subject to incompleteness or inaccuracies.

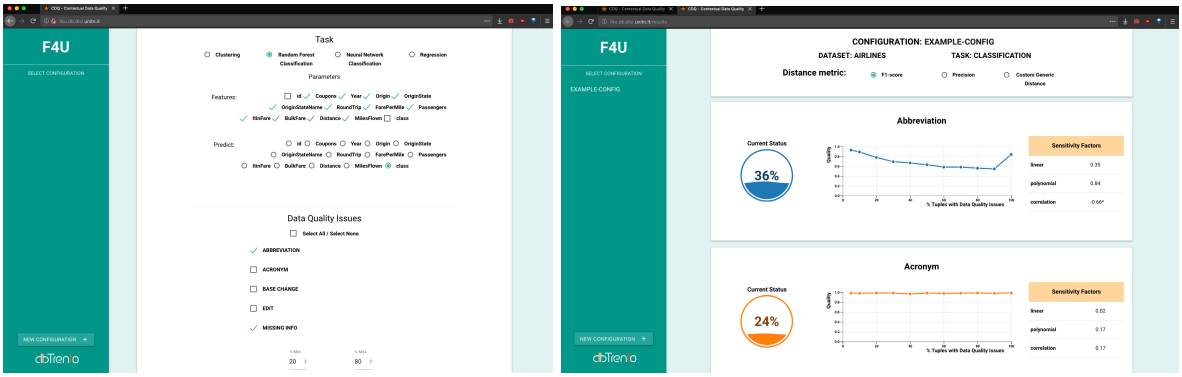

Fig. 3. The F4U Interface. Quality Evaluation Preparation (left) and Quality Evaluation Results (right)

[Scenario] The demo will be performed using three real-world datasets. The first is the ADULT, which contains demographic data about US citizens [8]. We will evaluate the quality of the dataset by predicting the annual income of a person, by classifying with random forest and neural network classification if the person earns more than $ 50K per year or not. The second dataset is the BANK dataset that records information about phone direct marketing campaigns of a Portuguese banking institution. It will be used for predicting with a regression the user account balance through the numerical features of the records. The third dataset is a subset of the AIRLINES dataset, presenting data from the US Department of Transportation about domestic flights. The quality of this dataset will be measured by grouping the flights in 4 clusters.

Note that the size of the datasets has been kept small for the purpose of the demonstration and its time constraints. Datasets of different sizes will be available and will be possible to test those as well, allowing the participant to understand how the system scales. The demo consists of the following steps.

[Dataset Loading] The demo will start by loading the dataset. While the datasets is loaded, the presenter will explain the traditional approach of data quality to allow the audience to appreciate the added value that the current system brings to the table.

[Noise Selection] The user is then selecting different kinds of noise that can be be ingested in the data. It is not only the type of noise that the user can configure, but also many characteristics of that noise.

[Intended Analytic Task Selection] After the selection of the noise types and characteristics, the user will select the analytic task of preference. Different analytic tasks will be available to choose from, among them one that is considered as black box.

[Execution] The system will be fired up. The results of the evaluation will then be presented by illustrating the degrees in which the results of the analytic task results are affected as a function of the variation of the data quality. The materialization of the change of the data quality is by the different noise forms that were generated and injected in the data. These are the coefficients and will be presented in a graphical mode.

[Distance Metric Selection and Result Studying] The users will have the ability to select different metrics that indicate the analytic task variation. Some of these methods will be the standard metrics used in traditional tasks, but also our own dataset distance metric. By changing parameters, and noise generation configurations, the user will be able to get a more spherical understanding of the data quality effects.

Part of the interface is illustrated in Figure 3. The left-hand side is the part that indicates the noise generation configuration, and the right hand-side is the one that presents the results of the coefficients generation. For a more live view of the tool, we refer to the accompanied demo video (which can also be found available online at the project web page https://db.disi.unitn.eu/projects/f4u.html).

We will perform the first demonstration ourselves to give the audience the opportunity to get accustomed with the system, but then we will allow the members of the audience to use and play with the system by modifying different parameters and obtaining a better understanding of the quality of the dataset.

REFERENCES

[1] Z. Abedjan, X. Chu, D. Deng, R. C. Fernandez, I. F. Ilyas, M. Ouzzani, P. Papotti, M. Stonebraker, and N. Tang. Detecting data errors: Where are we and what needs to be done? *PVLDB*, 9(12):993–1004, 2016.

[2] P. C. Arocena, B. Glavic, G. Mecca, R. J. Miller, P. Papotti, and D. Santoro. Messing up with bart: error generation for evaluating data-cleaning algorithms. *VLDB*, 9(2):36–47, 2015.

[3] M. Bergman, T. Milo, S. Novgorodov, and W. C. Tan. Query-oriented data cleaning with oracles. In *SIGMOD*, pages 1199–1214, 2015.

[4] X. Chu, J. Morcos, I. F. Ilyas, M. Ouzzani, P. Papotti, N. Tang, and Y. Ye. KATARA: A data cleaning system powered by knowledge bases and crowdsourcing. In *SIGMOD*, pages 1247–1261, 2015.

[5] W. Fan. Data quality: from theory to practice. *SIGMOD Record*, 2015.

[6] W. Fan and F. Geerts. Capturing missing tuples and missing values. In *PODS*, pages 169–178, 2010.

[7] I. F. Ilyas, X. Chu, et al. Trends in cleaning relational data: Consistency and deduplication. *Foundations and Trends® in Databases*, 5(4):281–393, 2015.

[8] R. Kohavi. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *KDD*, pages 202–207, 1996.

[9] S. Krishnan, J. Wang, E. Wu, M. J. Franklin, and K. Goldberg. Activeclean: Interactive data cleaning for statistical modeling. *VLDB*, 9(12):948–959, 2016.

[10] P. Papotti. Data quality between promises and results. In *ICDE*, page 200, 2016.

[11] C. Spearman. The proof and measurement of association between two things. *The American journal of psychology*, 15(1):72–101, 1904.

[12] B. Walenz and J. Yang. Perturbation analysis of database queries. *Proceedings of the VLDB Endowment*, 9(14):1635–1646, 2016.

[13] R. Y. Wang and D. M. Strong. Beyond accuracy: What data quality means to data consumers. *JMIS*, 12(4):5–33, 1996.