Stable Conditional Metrics for Data Ingestion Validation

Niels Bylois Frank Neven Stijn Vansummeren

UHasselt, Data Science Institute, ACSL

Data-driven decision making permeates all levels of modern enterprises and organisations. At the basis of this decision process lies the continuous collection and ingestion of relevant data from diverse sources. Validating the quality of collected data at ingestion time is crucial. Contemporary data quality validation tools allow specification, either manually or automatically, of so-called *data unit tests*. When a new batch of data is to be ingested, the registered tests are executed to gauge the batch's quality where failing tests highlight data quality problems. The tests themselves entail computing certain metrics on the data batch (e.g., the minimum or maximum value appearing in a numerical column or the number of distinct elements appearing in a column) and checking that these fall within an expected range. Unfortunately, however, the metrics employed to date are *coarse-grained*: they compute data statistics on the entire data batch, or on an entire column thereof. In this talk, we present an approach for *fine-grained* data quality validation aimed to detect errors that pertain to specific entities or groups of entities. We focus on the setting where a data pipeline regularly ingests batches of external data.

We formally introduce *conditional metrics* as a means for detecting fine-grained errors. Intuitively, conditional metrics can compute data quality metrics over specific parts of an ingestion batch (e.g., the number of product reviews in the electronics category), instead of on the entire batch. A *data unit test* is then composed of a conditional metric and a range of acceptable values. Data batches on which the conditional metric returns a value outside of this range are considered to be potentially erroneous.

We propose a method where a sequence $\overline{R}$ of previously ingested batches of data $\overline{R} = R_1, R_2, \ldots, R_n$ is used to automatically derive data unit tests, based on conditional metrics. The quality of a yet-to-be-ingested batch B can then be validated by comparing its conditional metric values with those in the batches of $\overline{R}$. Formally, our method derives a set Φ of data unit tests from $\overline{R}$ such that B is considered to be of acceptable quality if it passes all tests $\phi \in \Phi$. The construction of Φ from $\overline{R}$ requires considering a large space of possible conditional metrics, but not all of these are useful as a data unit test. Indeed, conditional metrics whose values over $\overline{R}$ are more or less constant are much more suited to detect errors on B than those with an arbitrary behavior. We therefore propose a simple notion of *stability* as a means to decide on the set of conditional metrics to promote to unit tests.

For the situations where an ingestion batch violates one or more unit tests, we offer a method that allows us to zoom in on potentially erroneous regions in the batch. Every violating unit test specifies such a region in a natural way: the subrelation of B that it refers to. However, simply flagging *all* the regions of violated unit tests as being potentially erroneous selects too many regions (i.e., it results in a high recall but very low precision). The reason is that violating unit tests may be correlated: a single error in B may cause multiple unit tests (each selecting different regions) to fail. It is therefore important to determine a subset of the violating unit test regions that best describe the errors detected. We propose a two-pronged approach that first determines, from the set of violating unit tests, the principal *entities* with errors, and then, in a second step, returns for each such entity the affected set of columns. In particular, to detect entities, we propose a method that groups unit tests into sets of tests that have a non-zero overlap (in the sense that they select a common tuple). For each group the most relevant entity test is identified based on the average row coverage count, a notion that we define. The group construction method itself is iterative, starting from an initial ranking of possible groups that is also based on row coverage count.

We show that our approach can be implemented on top of existing database systems, with acceptable performance for ingestion batches of medium size and present an extensive experimental validation of our approach on two real world datasets.