Demonstration: MacroBase, A Fast Data Analysis Engine

Peter Bailis, Edward Gan, Kexin Rong, Sahaana Suri
Stanford InfoLab

ABSTRACT

Data volumes are rising at an increasing rate, stressing the limits of human attention. Current techniques for prioritizing user attention in this *fast data* are characterized by either cumbersome, ad-hoc analysis pipelines comprised of a diverse set of analytics tools, or brittle, static rule-based engines. To address this gap, we have developed MacroBase, a fast data analytics engine that acts as a search engine over fast data streams. MacroBase provides a set of highly-optimized, modular operators for streaming feature transformation, classification, and explanation. Users can leverage these optimized operators to construct efficient pipelines tailored for their use case. In this demonstration, SIGMOD attendees will have the opportunity to interactively answer and refine queries using MacroBase and discover the potential benefits of an advanced engine for prioritizing attention in high-volume, real-world data streams.

1. INTRODUCTION

Data volumes are rapidly increasing due to a rise in automated data sources, including sensors, processes, and devices. Today, each of Facebook, Twitter, and LinkedIn record over 12M events *per second* from their production infrastructure. Keeping up with these volumes is challenging: top application operators report utilization of less than 6% of collected data [3], due largely to computational overheads and inability to manually inspect data at scale.

To help prioritize attention in these increasingly abundant high-volume data streams, new data infrastructure is needed. Data infrastructure for executing more complex functionality for prioritizing human attention—i.e., distributed dataflow engines such as Spark Streaming and Storm—is commonplace; however, these engines leave the task of actually specifying and implementing this complex functionality as a rarely-completed exercise for the end user. Instead, application operators currently process these volumes using a combination of ad-hoc, post-hoc analyses (i.e., root cause analysis after a failure) and brittle, static thresholds that are computationally inexpensive but miss important trends and events.

SIGMOD'17, May 14 - 19, 2017, Chicago, IL, USA

ISBN 978-1-4503-4197-4/17/05…$15.00

DOI: http://dx.doi.org/10.1145/3035918.3056446

To capitalize on this opportunity, we are developing MacroBase, an analytics engine designed to prioritize attention in these large-scale, high-volume *fast data* streams [2]. MacroBase is powered by a key observation: to prioritize human attention, we need new analytics operators that both identify points of interest within the stream and aggregate commonalities among them. In the parlance of machine learning, this corresponds to a combination of streaming classification and explanation techniques, at scale. Moreover, by co-designing these operators, we can exploit new optimizations unavailable to each in isolation.

MacroBase serves as a vehicle for concept validation and ongoing research—both of which we highlight in this demonstration. First, the open source MacroBase prototype provides an interface that enables even non-technical domain experts to easily analyze their data. It allows users to connect to structured data sources (e.g., via JDBC) and perform classification and explanation at interactive time-scales via a few mouse clicks. Users in industries including automotive, industrial manufacturing, and mobile application development have utilized this interface (and more advanced dataflow-level interfaces) to discover previously unknown behaviors with their data. Second, feedback from these users drives ongoing research in fast data analytics. MacroBase is the host of a number of ongoing sub-projects spanning fast dimensionality reduction for heterogeneous data, detection of complex, multi-modal events, and time-series presentation and visualization.

In this demonstration, SIGMOD attendees will interact with the MacroBase prototype as it analyzes streaming data from a real-world "smart city." Specifically, MacroBase will process heterogeneous data streams in real-time from urban infrastructure including traffic, parking, weather, and event data to highlight trends including congestion and inefficiency. By exposing attendees to the underlying dataflow operators powering these queries, attendees will observe the benefit of both operator co-design and recent advances in fast data processing first-hand.

2. MACROBASE OVERVIEW

MacroBase performs fast data analysis by executing configurable dataflow pipelines that combine operators for feature transformation—to extract key features from data points in the stream—classification (e.g., outlier detection)—to highlight individual points of interest—and result explanation—to aggregate and summarize key trends across data points [2, 3]. A MacroBase pipeline ingests data from both static and streaming external data sources, then processes it using MacroBase's specialized operators to highlight and contextualize important and unusual behaviors. Users interact with the system to winnow down the massive fast data volumes they face either via high-level graphical interfaces, or by composing operator pipelines directly via scripts.

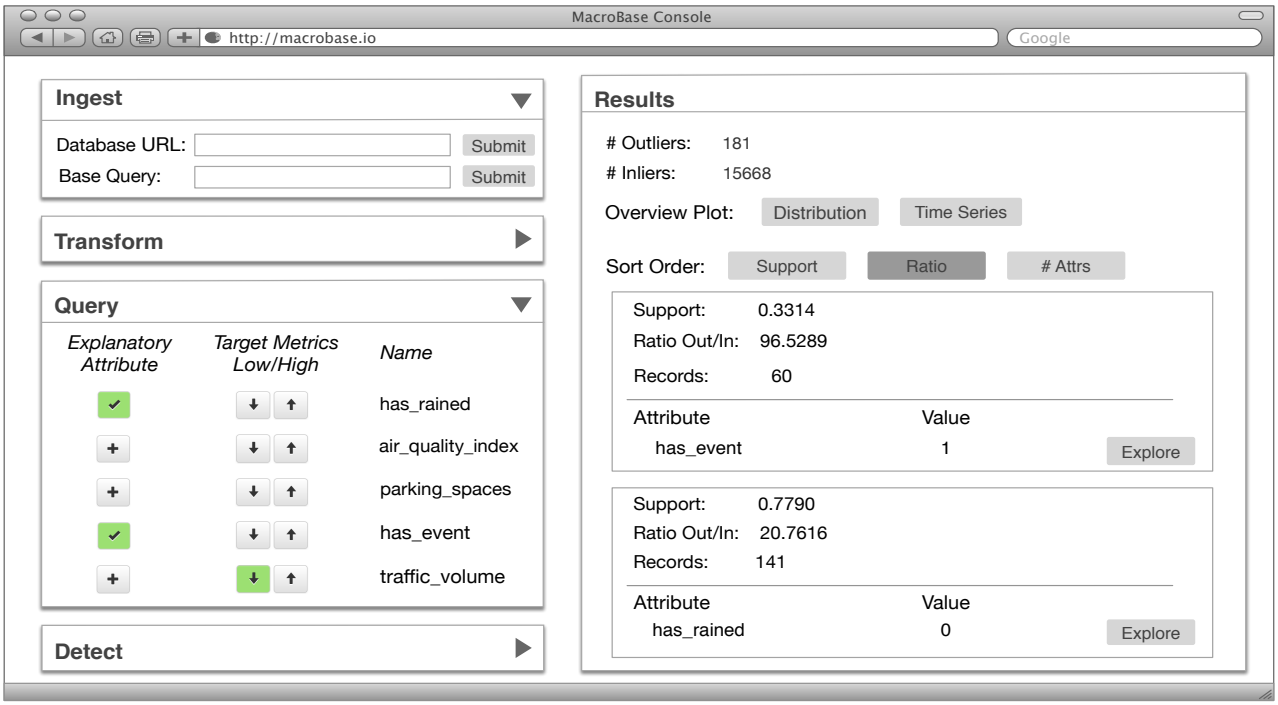

Figure 1: MacroBase's current user interface. (a) By default, users specify target metrics and explanatory attributes as input. (b) MacroBase performs classification and reports combinations of attributes that are disproportionately correlated with abnormal metrics.

Basic Query Model and Concepts. Figure 1 illustrates the primary graphical user interface for interacting with MacroBase. The MacroBase UI is divided into two panes: one for user specified parameters (left, a) and one for MacroBase's result explanations (right, b). Given a dataset (for example, specified via a JDBC connection and a SQL query), users perform manual feature selection by highlighting attributes within the dataset as either key *metrics* or metadata *attributes*. Subsequently, by default, MacroBase performs unsupervised density-based classification and explanation to illustrate attributes that are unusually correlated with abnormal metric behavior (e.g., traffic volume is 60 and 141 times more likely to be unusually bad if there is a special event or it is raining). This UI provides users the chance to specify hints regarding the the type of data and phenomenon they are expecting, and then observe a series of high-level, aggregate explanations of behaviors MacroBase can uncover via its unsupervised classification operators. After examining the summaries provided by the default configuration, users can drill down into each explanation to inspect data at a finer granularity or iterate on their query specification to refine their results.

To enable this workflow, MacroBase executes a pipeline composed of three core operator classes, or stages: feature transformation, classification, and explanation. A MacroBase pipeline processes data according to user-specified metrics, attributes, and any optional input regarding the dataset via feature transformations. By providing a range of optimized, modular operators for each operator class, MacroBase is able to execute efficiently and accurately over different data types and user queries.

Designing modular and high-performance operators for each stage of the MacroBase pipeline is an important element of our ongoing research. We have already determined several core operators for each task: for example, for unsupervised classification (classification without labels), we rely on density-based estimators that find rare points within a population, while for supervised classification (classification with labels), we rely on logistic regression and neural networks. Unlike more generic packages such as sci-kit learn, each of the operators in MacroBase is designed with the transform-classify-summarize pipeline in mind, allowing them to operate more efficiently and deliver results more amenable to human inspection. Designing operators in the context of this pipeline structure is an important feature of MacroBase's design, and makes customization easy for both developers and end-users.

Each operator is designed to run over both offline batch datasets and data streams. Operating over streams typically involves maintaining a reservoir sample of past data to compare against and incrementally updating models as new data arrives, without incurring expensive re-computation costs. However, from an end-user perspective, the stream is mostly transparent to the MacroBase UI: results are presented upon user requests, whether they were obtained from a batch recomputation or an incremental streaming update.

The role of each of these operator types as well as ongoing work highlighted in this demonstration is described in more detail below.

2.1 Feature Transformation

Data drawn from a variety of sources and sensor types is extremely noisy and high-dimensional. Operating over this raw data is often insufficient for performing meaningful analyses over the data stream. MacroBase supports a variety of *feature transformation* operators to enable the construction of more complex features that are most conducive to the task at hand. For instance, to analyze data patterns over time, a Fourier transform operator can extract a signal's periodicity for use in future MacroBase pipeline stages.

Ongoing work: Automated dimensionality reduction. When working with high-dimensional, multi-modal data, it is imperative to select a subset of dimensions (or functions over these dimensions) to analyze. The literature provides a multitude of means of selecting these features via distance-preserving dimensionality reduction. However, existing bounds in the literature are often pessimistic, and many machine-generated data streams are highly "structured," possessing low intrinsic dimensionality. That is, in large-scale, high-volume data streams, there is limited *variance* across data points. Complex systems have a set of well-defined behavioral modes; insofar as we can detect these modes via intelligent sampling, we need not look at all of the data in order to derive a high-quality low-dimensional representation. We have developed new, online, sampling- and gradient-based methods for automatically finding a suitable low dimensional basis for a given stream. Our preliminary results indicate potential performance speedups of three to four orders of magnitude over traditional approaches to this problem (e.g., full Singular Value Decomposition).

2.2 Classification

To identify interesting and/or noteworthy points, MacroBase relies on classification operators. For data that is not labeled a priori, MacroBase performs unsupervised *density estimation* by fitting a distribution to data and then identifying data in the tails. Thus far, much of our work has been on modeling relatively simple distributions in space [2]. However, ongoing work seeks to handle more complex distributions—ideally without compromising throughput.

Ongoing work: Handling complex, multi-modal behaviors. Many interesting behaviors in large-scale sensor streams correspond to low-density regions in multivariate space. For example, we may wish to search for irregular (or "rare") patterns in communications by looking at transmission activity over time. In the observed distributions, there may be multiple regions of high density separated by sharp gaps and no clear center. As a result, we are interested in improving the performance of *non-parametric* techniques flexible enough to capture these distributions. One insight is that we can exploit the end-to-end MacroBase pipeline structure to improve their performance: for example, if we only wish to classify the most "rare" data points in the distribution, we need not actually compute each point's precise density. Our recent results [4] indicate performance improvement of three to four orders of magnitude with negligible cost to classification accuracy.

2.3 Explanation

Once individual data points are identified as interesting, MacroBase attempts to provide high-level, human-interpretable explanations that aggregate and contextualize behaviors. While MacroBase's default explanation operator relies on correlation with metadata attributes [2], modalities such as time-series demand alternative means of presentation.

Ongoing work: Time-series visualization. To understand service quality and system behavior, "Big Data" application operators typically employ dashboards and plots of time-varying data. However, choosing the appropriate plots to help prioritize operators' attention in these reports can be difficult, especially given noisy data. For this purpose, we are interested in adapting low-pass filtering methods to automatically smooth displays and highlight outlying behavior in time-varying signals. The challenge here is to smooth as much as possible without losing the outlying structures. We believe it is possible to automate the hyperparameter selection for common low-pass filters to achieve this goal, and to do so online by leveraging techniques from stream processing (see [5]). Moreover, we can "push down" end-user display parameters such as minimum pixel resolution into the optimization procedure to further improve the efficiency of the hyperparameter search.

3. DEMONSTRATION

At SIGMOD 2017, we will demonstrate MacroBase's ability to prioritize attention in high-volume, heterogeneous, streaming data. MacroBase is already in production use outside our group (e.g., in the automotive, manufacturing, and mobile application industries), due in large part to an open source demo toolbox that users can experiment with on their own that exercises the most basic of MacroBase's functionality. At SIGMOD, we want to recreate the experience and thrill of working with production, high-volume data for attendees while simultaneously showcasing and gathering feedback on features that are under development.

3.1 Demo Scenario: Smart City Traffic

We showcase MacroBase's capabilities by examining activities in a "Smart City", combining two months of traffic, weather, pollution, parking and event data in the city of Aarhus, Denmark [1]—a characteristic example of a fast data workload. Data arrives every five minutes from each of 450 traffic sensors, and hourly from eight parking sensors. Each data point is furthermore associated with dozens of supplementary attributes. We will demonstrate how MacroBase can answer a number of queries in simulated "real time" by combining these heterogeneous streaming data sources. For instance, consider the following scenarios:

- Weather patterns and local events strongly influence driving patterns and congestion levels. In particular, recurrent traffic spikes and jams are often caused by the same underlying factors. Can we uncover such correlations, and use this to inform traffic policy?
- Road and parking lot conditions can vary dramatically across space and time. Some roads and lots can be strikingly underutilized or have unusual usage patterns due to road blocks or maintenance issues. Can we isolate these issues by observing their effect on usage?
- Air quality is becoming a rising health concern as we see increasing amounts of smog over major cities. Can we identify correlations between pollution indicators and traffic volume across the city?

To answer these questions today, end-users must rely on a disparate set of data systems, statistical packages, and summarization engines. Configuring and linking these together into a coherent analytics pipeline is difficult for domain experts who may not also be experts in building systems. Further, off-the-shelf implementations of the relevant components are rarely designed for streaming execution or the data volumes we consider here.

3.2 Demo: Guided Tour

MacroBase is able to provide out-of-the-box results with limited configuration, making data exploration easy to perform. Subsequently, as users employ more complex functionality, they can refine initial results to uncover more nuanced query results. Following this typical usage path, we exhibit how MacroBase prioritizes attention by starting with a basic query using MacroBase's default pipeline, and iteratively refining results via more complex dataflow operators for feature transformation, classification, and explanation. For this walkthrough, we focus on a simple question: can we use indicators including weather, pollution, and events to identify congestion

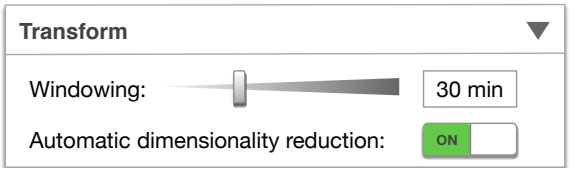

Figure 2: Time Series Transformation. In addition to individual data points, users can also look at time windows and enable automatic dimensionality reduction for speed and quality improvements.

across the city over time?

Base Query. In this demonstration, attendees will interact with a graphical UI similar to the one provided in the current open source release, but augmented with advanced functionality. Users first specify a set of data sources in the Ingest Panel (pre-populated). They can then use the Query panel (Figure 1a) to perform basic feature selection by specifying target *metrics* for classification and *attributes* for summarizing and explaining classification results.

We encourage users to start simple by defining a single metric of interest within the smart city (e.g., measured traffic throughput) and a small number of attributes to explain results with (e.g., local events and weather). Clicking on "Analyze" kicks off the default MacroBase pipeline to begin searching for results.

Initial Feedback. The Results panel presents *summaries* of the anomalies detected via MacroBase's default classifier (Figure 1b). MacroBase displays the number of anomalous data points discovered, and provides ways to contextualize them.

In this example, local events and weather were selected as explanatory attributes. MacroBase has broken down the space of attribute values into classes, say measurements on rainy days during game day, and identified those that have unusually high anomaly rates. These outlying rates are captured by the *risk ratio* of a summary class: the higher the ratio, the more likely an attribute combination coincides with behavior worth exploring.

Time Series Transforms. The previous setup successfully identified interesting data points but failed to account for data patterns over time. Here, we demonstrate how MacroBase utilizes *time series feature transformations* to combat this problem.

Figure 2 depicts an expanded view of the Transform panel in the UI (Figure 1a). First, users determine if they want to group data points by time windows. For instance, a user may wish to detect 30-minute time periods with abnormally high traffic volume. Rather than processing a collection of disjoint scalar values, this windowing parameter directs MacroBase to process vectors representing time series windows. Second, automatic dimensionality reduction can be enabled to both speed up classification runtime and avoid poor classification results resulting from "curse of dimensionality". As previously described, this operator is powered by algorithmic advances in fast, automatic dimensionality reduction.

Anomalies in Context. As we are operating over time windows, it is not enough to summarize only via the risk ratio—contextualization of unusual time windows is indispensable. In Figure 3 we depict MacroBase's solution: not only do we visually compare representative outlier and inlier time windows, but we also highlight them in the context of a much longer, smoothed time series. As described, MacroBase utilizes a new smoothing operator to better present the trends of large and often noisy time series.

Classification over Complex Distributions. Finally, MacroBase's default Gaussian classifier is not sufficient to model the distribution of even dimensionality-reduced time windows. For instance, interesting anomalies may not always be the vectors with the smallest or largest magnitudes, but those with the most distinct shapes. Macrobase's solution is to use a more expensive Kernel Density estimator that works better on multi-modal distributions when one is interested in distance to nearest neighbors. Figure 4 depicts how classifiers can be selected in the UI.

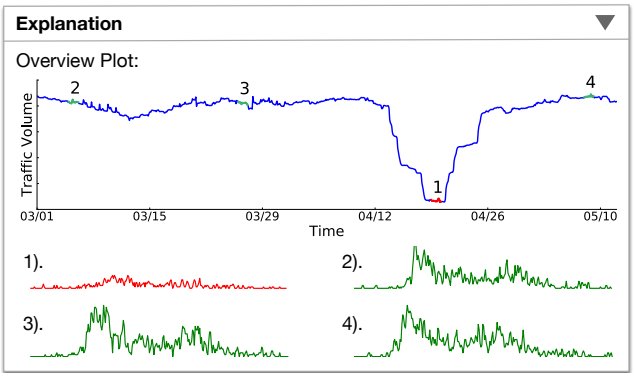

Figure 3: Explaining anomalies in context. We present sample outlier (red) and inlier (green) time windows for visual comparison, as well as windows in the context of the overall trend represented by the smoothed overview target metric plot (of traffic volume).

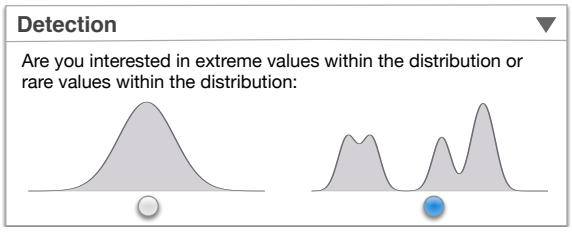

Figure 4: By default, MacroBase uses a Gaussian classifier, but users can also easily configure the system to use the Kernel Density classifier in the GUI, which better models multi-modal distributions.

3.3 Takeaways

Our goal in this demonstration is to exemplify how the co-design of classification and explanation in a unified pipeline is key to processing fast data. MacroBase's default pipeline provides useful out-of-the-box results that can be quickly iterated upon by configuring its operators. In our demo, both generic and nuanced time-series analyses are thus accessible. Looking forward, we wish to develop and additional operators for new domains and hope others will do the same. Fast data is here; are we ready to handle it?

4. REFERENCES

[1] M. I. Ali, F. Gao, and A. Mileo. Citybench: A configurable benchmark to evaluate rsp engines using smart city datasets. In *ISWC*, 2015.

[2] P. Bailis, E. Gan, S. Madden, D. Narayanan, K. Rong, and S. Suri. MacroBase: Prioritizing Attention in Fast Data. In *SIGMOD*, 2017.

[3] P. Bailis, E. Gan, K. Rong, and S. Suri. Prioritizing Attention in Fast Data: Principles and Promise. In *CIDR*, 2017.

[4] E. Gan and P. Bailis. Ic2: Indexed cutoffs for kernel density classification. In *SIGMOD*, 2017.

[5] K. Rong and P. Bailis. ASAP: Automatic Smoothing for Attention Prioritization in Streaming Time Series Visualization. 2017. arXiv:1703.00983.