READ: Rapid data Exploration, Analysis and Discovery

Udayan Khurana∗
University of Maryland
College Park
udyayan@cs.umd.edu

Srinivasan Parthasarathy
IBM Research
Yorktown Heights
spartha@us.ibm.com

Deepak Turaga
IBM Research
Yorktown Heights
turaga@us.ibm.com

ABSTRACT

Exploratory data analysis (EDA) is the process of discovering important characteristics of a dataset or finding data-driven insights in the corresponding domain. EDA is a human intensive process involving data management, analytic flow deployment and model creation, and data visualization and interpretation. It involves extensive use of analyst time, effort, and skill in data processing as well as domain expertise. In this paper, we introduce READ, a mixed initiative system for accelerating exploratory data analysis. The key idea behind READ is to decompose the exploration process into components that can be independently specified and automated. These components can be defined, reused or extended using simple choice points that are expressed using inference rules, planning logic, and reactive user interfaces and visualization. READ uses a formal specification of the analytic process for automated model space enumeration, workflow composition, deployment, and model validation and clustering. READ aims to reduce the time required for exploration and understanding of a dataset from days to minutes.

1. INTRODUCTION

Exploratory data analysis (EDA) is a critical statistical approach for discovering important characteristics of a dataset or finding data-driven insights into the corresponding domain. The core objective of EDA is to identify interesting phenomena in an application domain, and also help an analyst formulate interesting hypotheses that explain these phenomena. EDA involves the use of a variety of statistical and data visualization techniques to discover the core entities in an application domain, their characteristics, key features, interesting behaviors, and relationships. EDA has found wide spread acceptance across multiple domains including semiconductor manufacturing, analysis of communication networks, cyber security and threat analysis, fraud analysis, insurance, log analysis, and econometrics.

A key challenge with the EDA process is its human intensive nature, such as the need for extensive time and effort from highly skilled statisticians and domain experts as illustrated in Figure 1 and exemplified in Figure 2. The analyst starts with a collection of data sets, and deploys a variety of SQL or equivalent map-reduce analytic flows on the data in order to clean and create useful intermediate representations. This step is followed by a model creation step where the analyst typically creates analytic flows, either through an unprecedented level of automation.

During the EDA process, not all the models constructed by the analyst might lead to interesting insights. This implies that multiple iterations are usually required before the discovery of a domain relevant insight. The iterative nature of EDA and its heavy reliance on the analyst skill and effort contributes to the high time to discovery. Further the number of models explored is limited by the human in the loop, which might mean that some non-trivial and potentially interesting analysis paths leading to insightful models and discoveries may never get explored. Another significant challenge in this process is the extent of skills demanded from the analyst. This may include data management, data analysis, query construction and statistical and visualization expertise. Our system, READ, is aimed at these complex challenges in data exploration and discovery through an unprecedented level of automation.

READ is a mixed initiative system for data exploration and stands for Rapid Exploration Analysis and Discovery on Big Data. The key contributions and innovation of READ are described below:

1. READ decouples the description of data sets within the system from the description of analytics and queries deployed on the data. Analytics are represented within READ using a set of choice points: for instance, a histogram query on a data set can be represented using the table ingested in the query, the attributes by which the rows of the table are grouped for aggregation, the statistical function and the attributes on which the function is applied for histogram model generation. We employ automated reasoning for generating a space of potentially interesting models that can be derived from data: this is achieved by reasoning about the various possible ways in which analytic choice points can be instantiated concretely from the data specification.

2. READ uses automated planning technologies for workflow and analytic composition that result in the creation of models. While the analyst still has the option of pruning and filtering the set of models generated from the data, the capabilities described thus far largely frees up the analyst from mundane data management, analytic flow creation, and coding tasks that are involved in model building.

3. Automated model generation in READ yields the capability...
to rapidly generate hundreds or thousands of models via analytic flow deployment over datasets. In order to scale up the interpretation of these models by the analyst, READ uses novel visual and statistical techniques for model exploration. Specifically, READ provides a mechanism to plug in validation metrics, that determine which of the automatically generated models is the most interesting one. For instance, a histogram model can be interesting due to its entropy, skewness, or sparsity. READ also provides a mechanism to plug in similarity or distance metrics in order to find the models that are fairly similar to an interesting model. Clustering such a set of models under one category helps remove redundancy in the space of models. By generalizing the analysis for a representative model from the cluster, the analyst saves time on redundant inspection and interpretation. For instance, if five hundred models out of a pool of two thousand models mark out the same set of entities as anomalies, they could all be clustered together. In this case, the similarity is based on a Jaccard coefficient metric.

**Audience Experience with READ Demo:** We believe the combination of automated model enumeration, workflow composition, and validation and similarity metrics for rapid model interpretation makes for a novel, exciting, and accelerated process of knowledge discovery on massive amounts of data. During this demonstration of READ, users will be able to play with pre-loaded data sets, or plug in their own data sets and descriptions and watch READ auto-compute and enumerate a variety of model building options. Users can prune or filter from this enumeration (this is an optional human input). Once the subset of models to be built is generated, through a simple click of a button, users will also be able to witness automated workflow composition and deployment mechanisms at work leading to mass production of models that have been enumerated in the earlier step. Finally, users will be exposed to the READ model interpretation subprocess where interesting anomalies and trends are easily identified via validation metrics, and model redundancy is eliminated through similarity metrics. Specifically, the amount of time spend by users in driving READ would be proportional to the number of insights (e.g., distinct type of anomalies) they discover within the data, leading to a broad coverage of analysis paths explored and at the same time, fruitless analysis paths being eliminated.

2. RELATED WORK

John Tukey pioneered the use of exploratory data analysis nearly four decades ago [10]. Various packages and languages that support exploratory data analysis have been developed since. This includes, S, S-Plus, R, SPSS, SAS, and Matlab [7, 2]. A recent view of modern data exploration practices is available from Behrens and Yu [5]. SeeDB is a different line of work focused on finding the appropriate visualizations for a database query [9].

Online Analytic Processing (OLAP) is a key technology that supports exploratory data analysis and advanced predictive mining and modeling tasks that could be deployed on the data [8]. OLAP supports a variety of data models and operations including multidimensional conceptual views of data, query performance, intuitive data manipulation interfaces (e.g., slicing and dicing, drilling down), and cross dimensional operations. Another line of research that is related to READ is the use of automated planning algorithms for workflow composition and task execution within the EDA process [3, 4, 11]. READ and these works can mutually benefit each other as their design goals are orthogonal. Specifically, in READ: First, the enumeration and generation of the various analytics and queries that could be deployed on the data are computed by READ automatically (as opposed to providing optimized performance or workflow composition assistance for user initiated analytic tasks); Second, analytic automation and automated model generation with READ results in an explosion in the number of machine generated models and views of the data; READ supports intuitive and rapid exploration of these auto generated models through a novel collection of visual analytics for model validation and similarity clustering, which is not a consideration in the approaches outlined above.

3. READ DESIGN AND IMPLEMENTATION

Data analysis in READ is decomposed into four key subprocesses: 1) Data description, 2) Model space enumeration, 3) Model generation, and 4) Model interpretation. This is illustrated in Figure 3. We now describe the design and implementation of READ components that support these subprocesses.

3.1 Components

**Data Description:** We illustrate READ’s data description methodology in Figure 4. Data description in READ is intended to provide the same look and feel as schema description in a database with a few notable enhancements. There are opportunities within this description to identify specific attributes of a dataset as representing entities within a domain. Entities are the things of importance to an
Pool of models is auto-rank by various validation metrics. Analysts can choose other similar models to make a collective observation. Often, many different models suggest the same. The pool of generated models, potentially in a variety of data sets and automatically enumerate the space of models computable for them. A browsable view of the model space that is automatically enumerated by such inferencing is illustrated in Figure 5.

Algorithm 1 Example of an inference rule for enumerating univariate features from data specification

```plaintext
if D: Data Set with set of Attributes A then
  if a ∈ A and a is an entity then
    if b ∈ A and b ≠ a then
      if sf is a statistical function applicable to a variable of type b then
        Compute feature f := sf(b) from D for per group a
        end if
      end if
    end if
  end if
end if
```

**Model Building:** Once the space of models has been enumerated, queries are automatically composed and deployed using the MARIO automated composer [6]. We omit further details due to lack of space.

**Model Validation:** The pool of generated models, potentially in thousands, need to be interpreted by the analyst. Instead of providing each constructed model to the analyst for interpretation, READ provides a visually interactive mechanism for the analyst to select the top most interesting model as per one (amongst many) validation metrics. Figure 6 shows the selection of the most important box plot amongst approximately 1800 models using the minimum entropy metric. As per the box plot, the median value is zero, there are several non-zero entities, while one has an exceptionally high value. The analyst could choose from several other options such as max skewness, min zero count to name a few.

**Model Clustering:** Often, many different models suggest the same or similar facts. For instance, box plots for counts or perhaps, those using a different statistical function such as distinct count or entropy for the same entity are likely to point the same result (anomaly and more precisely, a primary or secondary DNS server in our cyber-security example). Similar observations can be derived for other entities which exhibit anomalous behavior as well. Hence, for a model that seems to be of interest, the analyst can choose other similar models to make a collective observation.
Interpretation and Annotation: If the analyst finds the observation from a cluster of models interesting, the corresponding entities in the model may be labeled. In our example, the analyst may give a label of “server” or “dns server” to the anomalies. The labels will be stored in a non-sourcable table by default.

6. REFERENCES


---

Figure 6: A novel visual analytics user interface that enabling rapid analyst interaction for interpretation of models. This interaction provides a variety of pluggable validation metrics - each metric ranks and decides on the most interesting model in the collection and a variety of distance metrics, which eliminate model redundancy by identifying other models close to the most interesting model. The user can choose the threshold at which this model cluster is created. The interpretation of this model cluster is represented using a cluster label: in this instance, the user has grouped 612 models out of 1742 models using the similarity threshold and called this cluster DNS Servers, as this group of models provides a way to discriminate between DNS servers and clients.

6. REFERENCES