Data Pipelines for Personalized Exploration of Rated Datasets

Sihem Amer-Yahia*, Anh Tho Le*†, Eric Simon†

*CNRS Univ. Grenoble Alpes, †SAP Paris

Abstract. Rated datasets are characterized by a combination of user demographics such as age and occupation, and user actions such as rating a movie or reviewing a book. Their exploration can greatly benefit end-users in their daily life. As data consumers are being empowered, there is a need for a tool to express end-to-end data pipelines for the personalized exploration of rated datasets. Such a tool must be easy to use as several strategies need to be tested by end-users to find relevant information. In this work, we develop framework based on mining labeled segments of interest to the data consumer. The difficulty is to find segments whose demographics and rating behaviour are both relevant to the data consumer. The variety of ways to express that task fully justifies the need for a productive and effective programming environment to express various data pipelines at a logical level. We examine how to do that and validate our findings with experiments on real rated datasets.

1 Introduction

We are interested in providing a tool for data consumers to explore rated datasets in a personalized fashion. Rated datasets are characterized by a combination of user demographics such as age and occupation, and user actions such as rating a movie or reviewing a book. We aim to provide data consumers with the ability to mine and explore labeled segments such as "young people who like German comedies from the 90's". The variety of ways such segments can be extracted justifies the need for a tool to express end-to-end data pipelines easily. Ease of use is of particular importance here as there is an infinite number of ways to express and find relevant segments. *In this paper, we lay the foundations for a framework to express data pipelines with a particular focus on improving the quality of extracted segments.*

Several frameworks to express pipelines have been proposed for large-scale data analytics. The approaches followed for data pipelines rely on the traditional separation between logical and physical operators. Logical operators capture fundamental operations required for data preparation and mining, whereas physical operators provide alternative implementations of the logical operators. The most prominent systems are SystemML[1] and KeystoneML [7][2] for the development of machine learning pipelines. For instance, in KeystoneML, logical

[1] https://systemml.apache.org/
[2] http://keystone-ml.org/

operators are tailored to the training and application of models whereas optimization techniques perform both per-operator optimization and end-to-end pipeline optimization using a cost-based optimizer that accounts for both computation and communication costs. By contrast, our goal is quality of the data pipeline without compromising response time.

The focus on quality is particularly important in our context. A user wishing to select a restaurant, movie or hotel, will benefit from the opinion of different segments, e.g., those with similar demographics or those with a similar opinion on other items. Indeed, while common demographics matter when inquiring about movies, they matter less for hotels. In practice, a user would benefit from the opinion of a variety of segments. While it is not possible for to examine the opinion of all relevant segments at once, providing the data consumer with the ability to *quickly prototype which segments to explore would be greatly useful.* A data pipeline would then take as input the profile of a data consumer and a rated dataset and return a set of segments whose quality is optimized for the data consumer, using some objective measures.

Several approaches could be used to extract labeled segments from rated datasets. Most of them are expressed as optimization problems that tackle one or multiple quality dimensions [1–5]. We design data pipelines that encapsulate those problems (Section 2). A pipeline could for instance look for the K most uniform segments, in terms of their ratings, and whose coverage of input data exceeds a threshold [2]. Alternatively, it could look for the K most diverse segments with the shortest labels [4]. Data consumers should be able to quickly prototype those pipelines by specifying which subset of the raters' population they want to hear from (e.g., people living in some part of the world, or people who like Indian restaurants) and letting our framework explore different physical implementations of their pipeline (Section 3). As a first step toward designing a full-fledged optimizer, our experiments assess the quality of segments generated by different pipelines for different data consumers (Section 4).

2 Data model

2.1 Rated datasets and labeled segments

A rated dataset $\mathcal{R}$ consists of a set of users with schema $S_{\mathcal{U}}$, items with schema $S_{\mathcal{I}}$ and rating records with schema $S_{\mathcal{R}}$. For example, $S_{\mathcal{U}} = \langle$`uid`, `age`, `gender`, `state`, `city`$\rangle$ and a user instance may be $\langle u1, young, male, NY, NYC\rangle$. Similarly, movies on IMDb[3] can be described with $S_{\mathcal{I}} = \langle$`item_id`, `title`, `genre`, `director`$\rangle$, and the movie *Titanic* as $\langle i2, Titanic, Romance, James\ Cameron\rangle$. The schema of rating records is $S_{\mathcal{R}} = \langle$`uid`, `item_id`, `rating`$\rangle$. The domain of `rating` depends on the dataset, e.g., $\{1, ..., 5\}$ in MovieLens,[4] $\{1, ..., 10\}$ in BookCrossing.[5]

[3] http://www.imdb.com/
[4] https://grouplens.org/datasets/movielens/
[5] http://www2.informatik.uni-freiburg.de/~cziegler/BX/

The record $\langle u1, i2, 5\rangle$, represents *a young male from NYC assigned 5 to the movie Titanic, directed by James Cameron.*

Given a rated dataset $\mathcal{R}$, we could generate a set of labeled segments $\mathcal{S}$ that are *structurally describable* using a conjunction of predicates on user and item attributes, e.g., the label of a segment $s \in \mathcal{S}$ can be {`genre` = Romance, `gender` = male, `state` = NY}. We use $records(s, \mathcal{S}) = \{\langle u, i, r\rangle \in \mathcal{S} \mid u \in s \wedge i \in s\}$ to denote the set of rating records of users on items in s.

Rating Distributions. We define the rating distribution of a segment $s \in \mathcal{S}$ as a probability distribution, $dist(s, \mathcal{S}) = [w_1, \ldots, w_M]$ where the rating scale is $\{1, \ldots, M\}$ and $w_j = \frac{|\{\langle u,i,r\rangle \in records(s,\mathcal{S}) | r=j\}|}{|records(s,\mathcal{S})|}$ is the fraction of ratings with value j in $records(s, \mathcal{S})$.

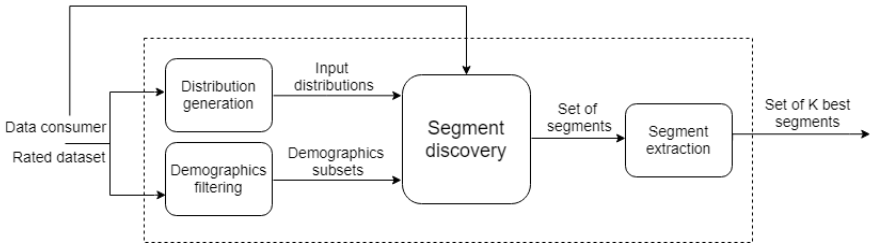

Fig. 1. Example data pipeline

2.2 Data pipelines

A data pipeline $\mathcal{D}$ is formed by a set of logical operators. Each operator o admits a set of segments as input and returns another set of segments. When o operates in a single set of rating records, its input is a single segment containing all those records. Figure 1 shows a pipeline that takes a data consumer profile and a rated dataset and returns a set of K segments relevant to that data consumer. The pipeline has 4 operators. The distribution generation operator takes the rating records of the data consumer and builds a set of segments and their distributions that represent the data consumer. The demographics filtering operator splits an input segment, in this case the input dataset, into demographics subsets, one for each value of the attributes of a data consumer. It is defined as a filtering of the input segment over the attribute value of the data consumer. The segment discovery operator creates a set of segments that are relevant to the data consumer, and the segment extraction chooses the K best segments. Every logical operator must have at least one physical operator associated with it which implements its logic. The presence of multiple physical implementations for each operator make the data pipeline a candidate for optimization.

The distinction from previous work is our focus on the quality of returned segments and the optimization of a logical data pipeline with respect to that quality goal. Quality is expressed as a function of several dimensions. For a set

of segments, quality reflects their coverage of input records and their diversity, i.e., their ability to reflect the opinion of a variety of users. The quality of a single segment can be computed as the length of its description and the relevance of the segment to the data consumer, i.e., how close the demographics or the opinion of users in that segment are to the data consumer.

Algorithm 1 Physical algorithm for segment discovery (`Alg`)

Input: $(\mathcal{R}, \{\rho_1, \ldots, \rho_j, \ldots, \rho_p\}, \theta)$
$parent = \mathcal{R}$
Array children
if $min_{j \in [p]}$`EMD`$(parent, \rho_j) \leq \theta$ **then**
  Add *parent* to *Output*
else
  Attribute `Attr` = findBestAttribute(*parent*)
  children = split(*parent*, `Attr`)
  for $i = 1 \rightarrow$ *No. of children* **do**
    `Alg`$(children[i], \{\rho_1, \ldots, \rho_j, \ldots, \rho_p\}, \theta)$
  end for
end if

Algorithm 2 Physical algorithm for segment discovery (`Alg`)

Input: $(\mathcal{R}, \{\rho_1, \ldots, \rho_j, \ldots, \rho_p\}, \theta)$
$parent = \mathcal{R}$
Array children
if $min_{j \in [p]}$`EMD`$(parent, \rho_j) \leq \theta$ **then**
  Add *parent* to *Output*
else if $min_{j \in [p]}$`EMD`$(parent, \rho_j) > \theta$ **then**
  Attribute `Attr` = findBestAttribute(*parent*)
  children = split(*parent*, `Attr`)
end if
for $i = 1 \rightarrow$ *No. of children* **do**
  if $min_{j \in [p]}$`EMD`$(children[i], \rho_j) \leq \theta$ **then**
    Add *children*[*i*] to *Output*
  else
    `Alg`$(children[i], \{\rho_1, \ldots, \rho_j, \ldots, \rho_p\}, \theta)$
  end if
end for

3 Data pipelines implementation

Each logical operator of a data pipeline can be implemented with different physical algorithms. Algorithm 2 is an example of an implementation of the segment discovery operator. This algorithm was proposed in [1] and relies om generating

a partition decision tree. It takes as input a rating dataset $\mathcal{R}$ and a set of distributions $\{\rho_1, \cdots, \rho_p\}$ that represent a data consumer. The algorithm uses Earth Mover's Distance (`EMD`) for segment comparison [6] and returns segments whose rating distribution is within a threshold θ of the distributions representing the data consumer. Whereas classic decision trees [8] are driven by gain functions like entropy[6] and gini-index,[7] `Alg` uses the *minimum average* `EMD` as its gain function. Suppose splitting a segment s using an attribute $\texttt{Attr}_i$ yields l children $y_1^i \ldots y_l^i$. The gain of $\texttt{Attr}_i$ is defined as the reciprocal of the average `EMD` of its children. More formally:

$$\texttt{Gain}(\texttt{Attr}_i) = \frac{l}{\sum_{j=1}^{l} min_{\rho \in \{\rho_1, \cdots, \rho_p\}} \texttt{EMD}(y_j^i, \rho)}$$

At each node, `Alg` checks if the current segment has $\texttt{EMD} \leq \theta$ to some input distribution (lines 4-5). If the segment's `EMD` distance to the closest input distribution is $> \theta$ (line 6), `Alg` uses our gain function to choose a splitting attribute (line 7), and the segment is split into child segments which are retained (line 8); Finally, retained segments are checked and are either added to the output (line 12) or recursively processed further (line 14). The algorithm finally returns a set of segments that are relevant to the data consumer, i.e., whose rating distributions are within θ of the data consumer's.

There exist other implementations for segment discovery [1–5]. Our goal is to optimize pipelines by comparing the quality of their returned segments.

4 Empirical validation and discussion

4.1 Validation

The purpose of validation is to examine the quality of returned segments for different data pipelines and users and make a case for an optimization framework. We sample the MovieLens dataset and choose rating records for "Drama" movies generated by the 137 random users (out of 6,040 users who rated those movies). Our dataset contains 2,000 rating records for 405 movies. We use the algorithm described in the previous section for segment discovery. For segment extraction, we chppse the top 10 largest segments in size. We run two data pipelines. The first one is an implementation of the pipeline in Figure 1 with Algorithm 2 for segment discovery. The second pipeline splits on both user demographics and movie attributes; and allow a segment which contain at least one of 4 keys ¡age, occupation, gender, location¿. The second implements a variant where no demographics filtering operator is provided and segment discovery splits input rating records on demographics. In the second pipeline, the obtained segments may correspond to users whose demographics are different from the data consumer's.

Table 4.1 reports our results for 3 kinds of consumers and their distributions: the neutral consumer, the polarized consumer, and a random consumer sampled

[6] http://en.wikipedia.org/wiki/Entropy
[7] http://en.wikipedia.org/wiki/Gini_index

from our dataset. We measure the quality of returned segments, i.e., their coverage of input records, their diversity, and the average description length. We also show some example segments. The higher the coverage and diversity, the better. The lower the description length, the better since data consumers prefer to read shorter segment descriptions. Our results show that there is a big difference in segment quality for different pipelines and users and that no pipeline wins on all fronts, thereby justifying to study the automatic optimization of data pipelines.

Data consumers	**Pipeline 1**	**Pipeline 2**
Neutral data consumer: *Young female* *executive from FL* $[0.2, 0.2, 0.2, 0.2, 0.2]$	Coverage: 0.581 Diversity: 0.007 Desc. Length: 1.8 e.g., *Females who rated movies from 2000*	Coverage: 0.533 Diversity: 1 Desc. Length: 2.8 e.g., *Young male artists living from MD*
Polarized data consumer: *Middle-aged male* *engineer from CA* $[0.5, 0, 0, 0, 0.5]$ $[1, 0, 0, 0, 0]$ $[0, 0, 0, 0, 1]$	Coverage: 0.230 Diversity: 0.014 Desc. Length: 1.9 e.g., *Males who rated movies written by Stephen King*	Coverage: 0.016 Diversity: 1 Desc. Length: 2.0 e.g., *Artists who rated movies written by Kenneth Branagh*
Random data consumer: *Young male* *scientist from WI* $[0, 0.5, 0.17, 0.33, 0]$ $[0.33, 0, 33, 0.33, 0, 0]$ $[0, 0.67, 0.33, 0, 0]$	Coverage: 0.691 Diversity: 0.006 Desc. Length: 1.6 e.g., *Young people who rated Steven Soderbergh movies*	Coverage: 0.486 Diversity: 1 Desc. Length: 1.6 e.g., *Male academics from MA*

Table 1. Segment quality for different data consumers and pipelines

4.2 Discussion

Our work opens several directions. The immediate one we are working on is to design an optimizer that switches between different data pipelines to find the most desired combination of coverage, diversity, description length and relevance to the data consumer's rating distributions. We believe that a hybrid approach that switches between automatic decisions and a human-in-the-loop process, is necessary to converge. That is because the final target is a data consumer with an information need in mind. Moreover, similarly to KeystoneML, we would like to study how to automatically optimize execution at both the operator and whole-pipeline levels. Due to our focus on quality, this would translate into defining how to compose pipelines to enable feedback-based optimization. Feedback from a data consumer can translate into a new set of rating distributions and demographics to be used as input in the next iteration. We believe that the ability to integrate that feedback with the automatic computation of segment quality will enable exploratory tasks that go beyond single consumers and serve consumer groups. This opens new directions for multi-feedback optimization.

References

1. S. Amer-Yahia, S. Kleisarchaki, N. K. Kolloju, L. V. S. Lakshmanan, and R. H. Zamar. Exploring rated datasets with rating maps. In *Proceedings of the 26th*

International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017, pages 1411–1419, 2017.
2. M. Das, S. Amer-Yahia, G. Das, and C. Yu. MRI: meaningful interpretations of collaborative ratings. *PVLDB*, 4(11):1063–1074, 2011.
3. M. Das, S. Thirumuruganathan, S. Amer-Yahia, G. Das, and C. Yu. An expressive framework and efficient algorithms for the analysis of collaborative tagging. *VLDB J.*, 23(2):201–226, 2014.
4. B. Omidvar-Tehrani, S. Amer-Yahia, P.-F. Dutot, and D. Trystram. Multi-objective group discovery on the social web. In *ECML/PKDD*, pages 296–312. Springer, 2016.
5. B. Omidvar-Tehrani, S. Amer-Yahia, and A. Termier. Interactive user group analysis. In *CIKM*, pages 403–412. ACM, 2015.
6. Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.
7. E. R. Sparks, S. Venkataraman, T. Kaftan, M. J. Franklin, and B. Recht. Keystoneml: Optimizing pipelines for large-scale advanced analytics. In *33rd IEEE International Conference on Data Engineering, ICDE 2017, San Diego, CA, USA, April 19-22, 2017*, pages 535–546, 2017.
8. P.-N. Tan et al. *Introduction to Data Mining, (First Edition)*. W. W. Norton & Company, 2007.