Exploring Rated Datasets with Rating Maps [*]

Sihem Amer-Yahia[‡], Sofia Kleisarchaki[‡♭], Naresh Kumar Kolloju[††],
Laks V.S. Lakshmanan[†], Ruben H. Zamar[†]
[‡]Univ. Grenoble Alpes, CNRS, LIG, F-38000 Grenoble, France, [♭] Univ Grenoble Alpes, CEA, Leti,
F-38000 Grenoble, France,[†]Univ. of British Columbia, Canada
[‡]firstname.lastname@imag.fr,[††]kollojun@amazon.com,[†]{laks,ruben}@cs.ubc.ca

ABSTRACT

Online rated datasets have become a source for large-scale population studies for analysts and a means for end-users to achieve routine tasks such as finding a book club. Existing systems however only provide limited insights into the opinions of different segments of the rater population. In this paper, we develop a framework for finding and exploring population segments and their opinions. We propose *rating maps*, a collection of *(population segment, rating distribution)* pairs, where a segment, e.g., ⟨*18-29 year old males in CA*⟩ has a rating distribution in the form of a histogram that aggregates its ratings for a set of items (e.g., *movies starring Russel Crowe*). We formalize the problem of building rating maps dynamically given desired input distributions. Our problem raises two challenges: (i) the choice of an appropriate measure for comparing rating distributions, and (ii) the design of efficient algorithms to find segments. We show that the Earth Mover's Distance (EMD) is well-adapted to comparing rating distributions and prove that finding segments whose rating distribution is close to input ones is NP-complete. We propose an efficient algorithm for building *Partition Decision Trees* and heuristics for combining the resulting partitions to further improve their quality. Our experiments on real and synthetic datasets validate the utility of rating maps for both analysts and end-users.

Keywords

Rated Datasets; Rating Distribution Comparison; Earth's Mover Distance; Partition Decision Tree

1. INTRODUCTION

Collaborative rating systems are routinely used by analysts to understand the preferences of different rater populations, and by end-users to make daily choices such as joining a book club or renting a movie. While many recommendation approaches have been proposed [2], very little has been done to contrast and compare the ratings of different segments and enable the exploration of their opinions.

Figure 1a illustrates an example on IMDb [1] for the movie *The Social Network*. Those *pre-computed segments* have similar average ratings and do not carry more information than the overall average. Figure 1b highlights two weaknesses of using averages when exploring different segments. First, an average is less informative than a distribution. While the rating average of *middle-aged raters in Boston* for *American Beauty (AB)* and of all raters for *Blair Witch Project (BWP)* are very close, their rating distributions are significantly different. In the case of *AB*, users are polarized and in the case of *BWP*, ratings are relatively uniform. Second, a common problem with aggregates such as average is that they fall prey to the Simpson's paradox [8]: the average rating for a given population (e.g., 4.3 for *AB* of the entire population) does not necessarily reflect those of its sub-populations (i.e., 3.17 for *AB* of *middle-age raters in Boston*).

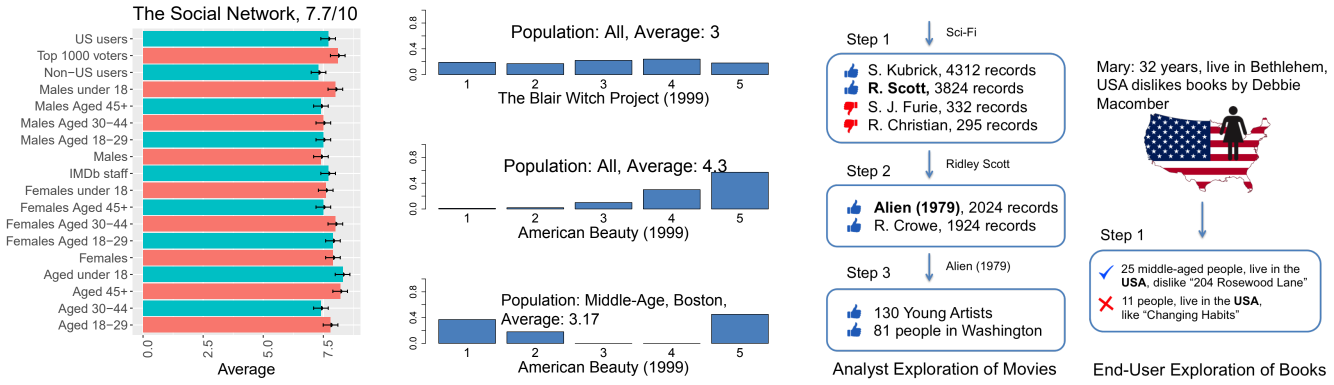
Another limitation when exploring segments on IMDb is that those segments are pre-computed. Figure 1c, on the other hand, shows how population segments are built dynamically, on-demand from rating records. Consider an analyst interested in discovering raters' segments who *like* or *dislike Sci-Fi*s. Figure 1c shows an exploration process where Step 1 reveals four segments: two that like *Sci-Fi* movies by directors *Ridley Scott* and *Stanley Kubrick*, and two that dislike movies by directors *Sidney J. Furie* and *Roger Christian*. Step 2 further explores the *Ridley Scott* segment, revealing other segments that explain the high rating of the director: one for the movie *Alien (1979)*, and another for movies starring *Russel Crowe*. Finally, Step 3 takes a deeper look and reveals that fans of *Alien (1979)* are *young artists* and people living in *Washington*. Similarly, Figure 1c illustrates the explanatory steps of an end-user who is looking for an online book club to discuss author *Debbie Macomber*. The end-user is interested in two segments: readers who agree with her, i.e., *middle-aged* reviewers who do not like the book *204 Rosswood Lane*, and those who disagree with her, i.e., people who love the book *Changing Habits*.

In this paper, we propose *rating maps* (Figure 1c), a collection of disjoint *(population segment, rating distribution)* pairs, and study how to build them dynamically and their utility in the exploration of rated datasets. A number of challenges arise when building rating maps. First, an appropriate measure is needed, able to find subtle differences

[*]This work is supported by the French National Research Agency in the framework of the "Investissements d'avenir" program (ANR-15-IDEX-02)

WWW 2017, April 3–7, 2017, Perth, Australia.
ACM 978-1-4503-4913-0/17/04.
http://dx.doi.org/10.1145/3038912.3052623

[1] *http://www.imdb.com*



(a) (b) (c)

Figure 1: (a) Segments on IMDb (b) Segments' Distributions (c) Segments Exploration with Rating Maps

between the rating distribution of a segment and an input distribution of interest. Second, a scalable algorithm for exploring the huge search space and dynamically building rating maps is imperative. Finally, the segments forming a map must satisfy certain quality criteria: *coverage* of input rating records, *diversity* in segment description to show different facets of the rater population, *size* of each segment (i.e., not too small), and high *proximity* of each segment to an input distribution.

In a nutshell this paper makes the following contributions:

1. We show that several sophisticated distance measures fail to discriminate between distributions. We show that the Earth Mover's Distance (`EMD`) [20] is able to capture subtle differences between two distributions and is appropriate for our problem.

2. Since *short descriptions* are more likely to satisfy segment quality criteria, i.e., coverage, description diversity and size, we formalize building rating maps as an optimization problem that *looks for a partition of a set of rating records such that each segment in the partition has the shortest description possible, and enjoys a rating distribution that has a low EMD with respect to some input distribution.* We show that our problem is NP-complete.

3. We propose a gain function, which selects the attribute (e.g., user or item attribute) from the multi-dimensional search space with the maximum gain when splitting and further exploring a segment.

4. We design `DTAlg` a linear algorithm that leverages the gain function to efficiently prune the exponential search space. `DTAlg` represents population segments using *a decision tree* [21] where nodes split an arbitrary set of rating records along user and item attributes. We also design heuristics tailored to optimize some of the quality criteria (but sacrificing some of their performance). Our heuristics are built upon `DTAlg` producing a *Random Forest (*`RF`*)* [3].

5. We run comprehensive experiments on real and synthetic datasets and demonstrate the effectiveness of rating maps. In particular, we develop scenarios for both analysts and end-users. We confirm the efficiency of `DTAlg` and `RF` heuristics w.r.t to segment quality criteria. We also identify the `RF` heuristic with the best compromise between the quality of generated maps and response time.

This paper is organized as follows. Section 2 presents our data model and formalizes the optimization problem of building rating maps. Section 3 performs a study of various distance measures. In Section 4.2, we discuss `DTAlg`, along with the `RF` heuristics. Our experimental study and findings are given in Section 5. Related work is discussed in Section 6. Section 7 summarizes and concludes the paper.

2. DATA MODEL

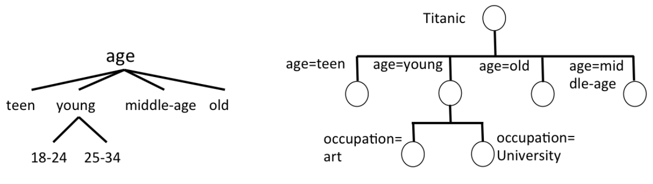
A rated dataset consists of a set of users with schema $S_{\mathcal{U}}$, items with schema $S_{\mathcal{I}}$ and rating records with schema $S_{\mathcal{R}}$. For example, $S_{\mathcal{U}} = \langle$`uid`, `age`, `gender`, `state`, `city`$\rangle$ and a user instance may be $\langle u1, young, male, NY, NYC\rangle$. Similarly, movies on IMDb 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 [18], $\{1, ..., 10\}$ in BookCrossing.[2] As an example, the record $\langle u1, i2, 5\rangle$, essentially says that *a young male from NYC assigned 5 to the romance movie Titanic, directed by James Cameron.* An *instance* consists of relations $\mathcal{U}, \mathcal{I}, \mathcal{R}$.

2.1 Population Segments and Rating Maps

Population Segments. A rated dataset $\mathcal{R}$ is viewed as a set of population segments that are *structurally describable* using a conjunction of predicates on user and item attributes of the form `Attr` $=$ *val.* For a population segment g, we let $g.idesc$ (resp., $g.udesc$) denote the set of item (resp., user) predicates associated with g. We use $g.desc$ to refer to $g.idesc \wedge g.udesc$. E.g., for $g_1.desc = \{$`genre` $= Romance$, `gender` $= male$, `state` $= NY\}$, $g_1.idesc$ refers to the first predicate and $g_1.udesc$ to the remaining ones.

Rating Distributions. The set of all population segments that contributed ratings in a dataset $\mathcal{S} \subseteq \mathcal{R}$ is denoted $G^{\mathcal{S}}$. Given a segment $g \in G^{\mathcal{S}}$, we define $records(g, \mathcal{S}) = \{\langle u, i, r\rangle \in \mathcal{S} \mid u \in g \wedge i \in g\}$ as the set of rating records of all users in g on items in g, in the rated set $\mathcal{S}$. The rating distribution of g in $\mathcal{S}$ is defined as a probability distribution, $dist(g, \mathcal{S}) = [w_1, \dots, w_M]$ where the rating scale is $\{1, \dots, M\}$ and $w_j = \frac{|\{\langle u,i,r\rangle \in records(g,\mathcal{S}) | r=j\}|}{|records(g,\mathcal{S})|}$ is the fraction of ratings with value j in $records(g, \mathcal{S})$. We blur the

[2] *http://www2.informatik.uni-freiburg.de/∼cziegler/BX/*



(a) (b)

Figure 2: (a) Attribute age (b) Categorical Split

distinction between g and $records(g, \mathcal{S})$ and speak of the records in g or the size $|g|$ of g.

Comparing Rating Distributions. We assume a generic function `ratComp` that compares two rating distributions and returns a score to reflect how far apart they are.

Rating Maps. Given a dataset $\mathcal{S}$ and its population segments $G^{\mathcal{S}}$, a rating map associated with $\mathcal{S}$ is a set of pairs $(g, dist(g, \mathcal{S}))$ where $g \in G^{\mathcal{S}}$. A rating map may contain population segments whose distributions are similar (using the function `ratComp`) to *unanimous* distributions $U_1, \ldots, U_M$. Here, U_i denotes the distribution where the mass is concentrated at rating value i: $U_i(j) = 1, j = i$ and $U_i(j) = 0, j \neq i$. For example, $U_1 = [1, 0, 0, 0, 0]$ in a rating scale of 5. Another example is a rating map containing *polarized* distributions $U_{1,M}$ where mass is concentrated on the extreme ratings 1 and M: e.g., $U_{1,M}(1) = U_{1,M}(M) = 0.5$ and $U_{1,M}(j) = 0, j \neq 1, M$.

2.2 Building Rating Maps

We call $[g_1, \ldots, g_\ell]$ a partial partition of $\mathcal{S}$ if g_i's are pairwise disjoint and $\bigcup_i g_i \subseteq \mathcal{S}$. One way to organize a partition is using a *partition decision tree*, defined as follows.

Partition Decision Tree (PDT). Given a rated set $\mathcal{S}$, a *partition decision tree* (PDT) of $\mathcal{S}$ is a rooted tree T such that: (i) the root of T contains the set $\mathcal{S}$ and every node x of T contains a subset of $\mathcal{S}_x \subset \mathcal{S}$ and every edge is labeled with a predicate $\mathtt{Attr} = val$ (ii) for a node x and its children $y_1, \ldots, y_p$, the collection $\{\mathcal{S}_{y_1}, \ldots, \mathcal{S}_{y_p}\}$ forms a disjoint partial partition of $\mathcal{S}_x$; (iii) for parent x and child y with the edge (x, y) labeled by the predicate $\mathtt{Attr} = val$, we have $\mathcal{S}_y = \{t \in \mathcal{S}_x \mid t \text{ satisfies } \mathtt{Attr} = val\}$.

Attribute values can be organized in a hierarchy. Figure 2a shows a partial hierarchy of attribute `age` from MovieLens. In order to generate PDTs, we use *categorical splitting*. Given an attribute $\mathtt{Attr}_i$ and a value v_j, this splitting results in n child segments, one for each value $v_j \in V$ where $V = \{v_1, \ldots, v_n\}$ is the active domain of $\mathtt{Attr}_i$. Figure 2b shows an example of categorical splitting of the rating records of the movie *Titanic* using attribute `age`. This splitting results in four child segments each node containing the rating records of users whose age is labeled by the node. Our PDTs can represent both categorical and numerical attributes. We can bin numerical attributes. In MovieLens and BookCrossing, `age` and `year` are already binned. The height of a PDT is the length of the longest root-to-leaf path. The leaves of a PDT form a disjoint (partial) partition of the rated set $\mathcal{S}$ at the root, corresponding to a segment. Each segment is described by the conjunction of predicates labeling the edges on the path from the root to the leaf.

Building Rating Maps Problem. Building rating maps with short segment descriptions corresponds to finding a PDT of small height. We can therefore state: *Given a rated dataset $\mathcal{S} \subseteq \mathcal{R}$, a rating proximity threshold θ, and a set of input distributions $\{\rho_1, \ldots, \rho_p\}$, find a PDT T of $\mathcal{S}$ with minimum height such that* $\mathtt{ratComp}(dist(g, \mathcal{S}), \rho_j) \leq \theta$ *for some $j \in [1, p]$. A typical value for p is 3, indicating* low, high *and* polarized *input distributions.*

The expression $\mathtt{ratComp}(dist(g, \mathcal{S}), \rho_j) \leq \theta$, finds population segments whose distributions are close to some input distribution. In order to find distributions that are far from input distributions, $\mathtt{ratComp}(dist(g, \mathcal{S}), \rho_j) \geq \theta$ can be used without affecting our algorithms.

3. RATING COMPARISON MEASURES

A key ingredient in the problem we study is the choice of the function `ratComp` that quantifies the proximity between two rating distributions. In this section, we review the Earth Mover's Distance (`EMD`) and argue why it is the best choice among a variety of widely used comparison measures.

EMD. Let $\rho_1 = [1 : p_1, .., i : p_i, .., M : p_M]$, $\rho_2 = [1 : q_1, .., i : q_i, .., M : q_M]$ represent two probability distributions over a discrete domain $D = \{1, 2, 3, \ldots, M\}$. The amount of work required to convert ρ_1 to ρ_2 is defined as: $\min_F \mathtt{Work}(\rho_1, \rho_2, F) = \sum_{i=1}^{M} \sum_{j=1}^{M} d_{ij} f_{ij}$ subject to the constraints: $f_{ij} \geq 0 \quad 1 \leq i, j \leq M$; $\sum_{j=1}^{M} f_{ij} = p_i \quad 1 \leq i \leq M$; and $\sum_{i=1}^{M} f_{ij} = q_j \quad 1 \leq j \leq M$, where f_{ij} is the amount of mass moved from position i to j in the process of converting ρ_1 to ρ_2. $F = [f_{ij}]$ is the matrix representing the flows and d_{ij} is the ground distance from position i to j, which, for simplicity, is defined as the absolute difference in positions, $|i - j|$. A flow F is optimal if the work done in the flow is minimum among all flows that convert ρ_1 to ρ_2. Therefore, the `EMD` is defined as: $\mathtt{EMD}(\rho_1, \rho_2) = \frac{\min_F \mathtt{Work}(\rho_1, \rho_2, F)}{\sum_{i=1}^{M} \sum_{j=1}^{M} f_{ij}}$.

In our setting, the region D over which `EMD` is calculated is always the whole domain of the distribution, so the value of the denominator in the above equation is 1.

EMD vs other measures. Table 1 shows various distance scores between two pairs of distributions.[3] We use
$\rho_1 = [0.9, 0.025, 0.025, 0.025, 0.025]$,
$\rho_2 = [0.025, 0.9, 0.025, 0.025, 0.025]$, and
$\rho_3 = [0.025, 0.025, 0.025, 0.025, 0.9]$,
corresponding to ratings on three books i_1, i_2, i_3. Intuitively, distributions ρ_1 and ρ_2 are more in agreement with each other than ρ_1 and ρ_3: users have similar opinions about books i_1 and i_2 and different opinions about i_1 and i_3. KL-divergence, a well-known proximity measure for probability distributions, defined as $D_{KL}(\rho_1, \rho_2) = \sum_j \rho_1^j log(\frac{\rho_1^j}{\rho_2^j})$, and its symmetric counterpart, JS-divergence, defined as $D_{JS}(\rho_1, \rho_2) = \frac{1}{2}(D_{KL}(\rho_1, \rho_3) + D_{KL}(\rho_2, \rho_3))$, where $\rho_3 = \frac{1}{2}(\rho_1 + \rho_2)$, are two natural choices for us [15]. Or we could interpret rating distributions as vectors and use cosine or Euclidean distance.

Table 1 shows that only Signal Noise Ratio (SNR), Lukaszyk -Karmowski metric, and `EMD` distinguish between the two pairs. Lukaszyk-Karmowski has the undesirable property that the distance between a distribution and itself is not zero. While SNR works for the above example, consider:
$\rho_1 = [0.0125, 0.0125, 0.0125, 0.0125, 0.95]$,
$\rho_2 = [0.0025, 0.0025, 0.0025, 0.0025, 0.99]$, and

[3] *http://en.wikipedia.org/wiki/Statistical_distance*

$\rho_3 = [0.95, 0.0125, 0.0125, 0.0125, 0.0125]$,
$\mathrm{SNR}(\rho_1, \rho_2) = 10.11$, $\mathrm{SNR}(\rho_1, \rho_3) = 6.25$, and $\mathrm{SNR}(\rho_2, \rho_3) = 16.37$. This places ρ_1 closer to ρ_3 than to ρ_2, which is counterintuitive. However, $\mathtt{EMD}(\rho_1, \rho_2) = 0.01$, $\mathtt{EMD}(\rho_1, \rho_3) = 3.75$, and $\mathtt{EMD}(\rho_2, \rho_3) = 3.85$. So EMD finds ρ_1, ρ_2 closer to each other than either of them to ρ_3, with ρ_1 being closer.

Measure	(ρ_1, ρ_2)	(ρ_1, ρ_3)
Cosine	0.058	0.058
KL-Divergence	3.13	3.13
JS-Divergence	0.53	0.53
Euclidean distance	1.24	1.24
Hellinger Distance	0.791	0.791
Total Variation Distance	0.875	0.875
Renyi Entropy Distance (0.5 order)	1.962	1.962
Battacharya Distance	0.981	0.981
Distance correlation	0.2500	0.2500
Signal Noise Ratio	2.0372	4.221
Lukaszyk-Karmowski Metric	1.1625	3.525
EMD	0.875	3.5

Table 1: Distance Measures Comparison

In general, the calculation of EMD between two distributions is done using the Hungarian algorithm and takes time $O(M^3 log\, M)$ where M is the domain size of the distribution [14]. However, in our setting, the distributions are probabilities over the same domain (rating scale), thus it is possible to compute their EMD in linear time [16]. For this purpose, we designed and implemented a linear EMD algorithm, the details of which are omitted for brevity.

4. BUILDING RATING MAPS

We first discuss the inherent complexity of the problem and then present our algorithms.

4.1 Problem Complexity

THEOREM 1. *Given a rated dataset $\mathcal{S} \subseteq \mathcal{R}$, a set of input distributions, and an* EMD *threshold θ, finding a minimum height partition decision tree for $\mathcal{S}$, where each segment's* EMD *is at most θ from some input distribution, is NP-complete.*

Membership in NP-hard. We show that the decision version of our problem is NP-hard by reduction from the classic Minimum Height Decision Tree problem [21]. Given a set I of n m-bit vectors and a number k, the question is whether there is a binary decision tree with height $\leq k$ such that, each of its leaves is a unique bit vector in I and internal nodes are labeled by binary tests on some bit. We construct an instance J from I as follows. J has m binary attributes $\mathtt{Attr}_1, ..., \mathtt{Attr}_m$ and a categorical attribute $\mathtt{Attr}_0$. For each bit vector s_i, J has two records t_i^+ and t_i^-, $i \in [1, n]$, with $t_i^+[j]$ and $t_i^-[j]$ set to the j-th bit of vector s_i, $j \in [1, m]$. Assign $t_i^+[\mathtt{Attr}_0]$ and $t_i^-[\mathtt{Attr}_0]$ to two distinct constants appearing nowhere else. Finally, the rating value for t_i^+ (resp., t_i^-) is 5 (resp., 1). Set the EMD threshold $\theta = 0$ and let the input distributions be $\{U_1, U_5\}$. E.g., if $I = \{011, 010, 100\}$ then J contains the records $(a_1, 0, 1, 1, 5), (b_1, 0, 1, 1, 1), (a_2, 0, 1, 0, 5), (b_2, 0, 1, 0, 1)$, $(a_3, 1, 0, 0, 5), (b_3, 1, 0, 0, 1)$, the last value being the rating.
Claim. *I admits a decision tree of height $\leq k$ iff J admits a partition decision tree of height $\leq k+1$ where each segment at its leaf is describable and exactly matches U_1 or U_5.*
Only If: Given a decision tree T for I, by definition, it contains a unique bit vector $s_i \in I$ at each leaf. If we apply this tree to J, we will get a tree each of whose leaf corresponds to a segment containing exactly the records $\{t_i^+, t_i^-\}$, $i \in [1, m]$. These segments do not match either of U_1, U_5. Applying a split based on $\mathtt{Attr}_0 = a_i$ versus $\mathtt{Attr} = b_i$ divides this segment into two singleton segments $\{t_i^+\}$ and $\{t_i^-\}$ which match U_5 and U_1. The segments are describable. This tree has height one more than that of T.
If: Let T be a partition decision tree of height $\leq k+1$ for J. By definition, each leaf of T contains a unique record of J. Notice that none of the segments at the leaves can contain more than one record with the same rating value, as they are not describable (without disjunction or negation). T must apply the predicates on attribute $\mathtt{Attr}_0$ to separate records t_i^+ and t_i^-. Suppose T applies these tests after all other tests. Then the node at which $\mathtt{Attr}_0 = a_i$ vs. $\mathtt{Attr}_0 = b_i$ is applied must contain exactly the segment $\{t_i^+, t_i^-\}$. By replacing that segment with the corresponding bit vector s_i, we get a decision tree of height $\leq k$ for I. Suppose T applies one or more tests on $\mathtt{Attr}_0$ before other attributes $\mathtt{Attr}_i$, where $i > 0$, we can show that we can "push down" those tests on $\mathtt{Attr}_0$ so they are applied at the parent of leaf nodes, without increasing the tree height.

Membership in NP. Given a height threshold h and a tree T, we can easily check in polynomial time whether T is indeed a partition decision tree of $\mathcal{S}$, each segment has an EMD distance at most δ from some input distribution, and whether the height of T is no more than h.

4.2 Algorithms

We now describe our algorithms for minimizing description length via finding a minimum PDT.

4.2.1 Minimizing Description Length with DTAlg

Algorithm 1 takes as input a rating set $\mathcal{S}$, a set of input distributions $\{\rho_1, \cdots, \rho_p\}$, an EMD threshold θ and divides $\mathcal{S}$, to find segments with minimum descriptions. At each node, DTAlg checks if its segment has EMD $\leq \theta$ to some input distribution (lines 3-4). If the segment's EMD distance to the closest input distribution is $> \theta$ (line 5), DTAlg uses our gain function to choose a splitting attribute (line 6), and the segment is split into child segments which are retained (line 7); Finally, retained segments are checked and are either added to the output (line 11) or recursively processed further (line 13).

Algorithm 1 DTAlg$(\mathcal{S}, \{\rho_1, \ldots, \rho_j, \ldots, \rho_p\}, \theta)$

```
 1: $parent = \mathcal{S}$
 2: $Array\ children$
 3: if $min_{j \in [p]}\mathtt{EMD}(parent, \rho_j) \leq \theta$ then
 4:     Add $parent$ to $Output$
 5: else if $min_{j \in [p]}\mathtt{EMD}(parent, \rho_j) > \theta$ then
 6:     $Attribute\ \mathtt{Attr} = \text{findBestAttribute}(parent)$
 7:     $children = \text{split}(parent, \mathtt{Attr})$
 8: end if
 9: for $i = 1 \to No.\ of\ children$ do
10:     if $min_{j \in [p]}\mathtt{EMD}(children[i], \rho_j) \leq \theta$ then
11:         Add $children[i]$ to $Output$
12:     else
13:         $\mathtt{DTAlg}(children[i], \{\rho_1, \ldots, \rho_j, \ldots, \rho_p\}, \theta)$
14:     end if
15: end for
```

Splitting using a Gain Function.

Whereas classic decision trees [21] are driven by gain functions like entropy[4] and gini-index,[5] we design a gain function that leverages the properties of EMD to discover segments whose distributions are close to input ones.

We use the *minimum average* EMD as our gain function. Suppose splitting a segment g using an attribute $\texttt{Attr}_i$ yields l children $y_1^i \dots y_l^i$. The gain of $\texttt{Attr}_i$ is defined as the reciprocal of the average EMD of its children. If child segments have a zero EMD then the gain is infinity. More formally:

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

An attribute will not be useful for splitting a segment if all the rating records in the segment have the same value for that attribute. For example, if a segment contains rating records of one movie, *Titanic*, none of the movie attributes are useful for splitting the segment. Such attributes are discarded and not considered for further splitting.

4.2.2 *Improving Coverage with Random Forests*

Segments with shorter descriptions are expected to satisfy our quality criteria of coverage, diversity, size and proximity to input opinions. However, splitting $\mathcal{S}$ into segments whose ratings are close to input distributions *does not necessarily guarantee that all records in $\mathcal{S}$ will belong to the resulting segments.* In this section, we focus on obtaining partitions with improved quality. In particular, we devise heuristics that each one is likely to improve at least a quality criterion.

We draw inspiration from the work of Breiman [3] who proposed the approach of *Random Forests*. Given d predictor attributes, DTAlg examines all d of them to pick the best split attribute and split point at each stage. Our RF approach consists of two main Steps. In Step 1, DTAlg is run m times for some parameter m, where each run examines a random subset of $\hat{d}$ predictor attributes at each splitting node, a default value for $\hat{d}$ being $\sqrt{d}$. This generates m PDTs. In Step 2, those m PDTs are combined, in such a way to maximize a quality criterion. In the next section, we examine different RF heuristics for combining the m partitions into a single one. RF heuristics are designed to improve the quality of produced rating maps, but this improvement comes at a (computational) cost (Section 5.2.2).

Combining Partitions.

There are multiple ways of combining the m partitions $p_1, \dots, p_m$ produced in Step 1 of RF. We propose five heuristics that give precedence to different segment quality criteria, namely coverage, diversity, size, and EMD value.

1. RF-Cluster: Each partition p_i intuitively captures a (possibly partial) clustering. For each pair of rating records $r_i, r_j \in \mathcal{S}$, we can compute k_{ij}, the number of partitions to which they both belong. Then, we can define the similarity between r_i and r_j as $sim_{ij} = \frac{k_{ij}}{m}$. Now, we can use any standard clustering algorithm to obtain a clustering of $\mathcal{S}$ with this distance measure (e.g., hierarchical clustering). As the desired number of clusters, we chose the average number of segments in the partitions $p_1, \dots, p_m$. The resulting segments do not necessarily have a natural exact description. We hence adopt a pattern mining approach to solve this issue. Viewing each record as a transaction and each user and item attribute as an "item", we obtain maximal frequent patterns, by setting the support threshold to 90%. Any maximal frequent pattern serves as an approximate description of the segment, with an accuracy of at least 90%. Algorithm RF-Cluster takes $O(mk^2n + mn^3)$ time where m is the number of decision trees built, n the number of rating records and k the number of distinct attribute-value pairs in the dataset.

	MovieLens (+IMDb)	BookCrossing
#Users	6,040	38,511
#Items	3,900	260
#Ratings	1,000,209 (million)	196,842
Rating Scale	1 to 5	1 to 10

Table 2: Summary of Datasets

2. RF-Desc: A second strategy favors a partition containing segments with diverse descriptions. We define the Jaccard distance between segment descriptions as $Jaccard(g_i, g_j) = \frac{|g_i.desc \cap g_j.desc|}{|g_i.desc \cup g_j.desc|}$. RF-Desc starts with an empty output partition and successively adds a segment whose total distance to segments in the output is the highest. The first segment is picked at random. This heuristic takes time $O(mk^2n + (ml)^3)$, where l is the maximum number of segments produced by any of the m runs of DTAlg.

3. RF-Size: This heuristic favors larger segments, which may help with coverage of input rating records. Its time complexity is $O(mk^2n + m^2nl)$.

4. RF-EMD: This heuristic favors segments with the lowest EMD to their closest input distribution. Its time complexity is $O(mk^2n + m^2nl)$.

RF-Cluster and RF-Desc are by far the most expensive heuristics for combining partitions, since the former computes the distance between all pairs of records in S and the latter iterates over all partitions' segments multiple times.

5. EXPERIMENTS

In this section, we evaluate the quality of rating maps over synthetic and real datasets. We study the robustness of our algorithms w.r.t. noisy datasets and demonstrate the quality of rating maps. We then study the scalability of DTAlg and RF approaches. That is followed by exploratory scenarios that show the utility of rating maps.

5.1 Experimental Setup

We use two real datasets, MovieLens (ML) [18] and BookCrossing [6] (BC), summarized in Table 2, and a synthetic one (Section 5.2.1). ML contains user attributes gender, age, occupation and location. We join this data with IMDb (via movie titles) to obtain attributes title, actor, director, writer for each movie. BC provides location, age for users and title, author, year, publisher for books. Some attributes have a hierarchy (e.g, *Country* → *State* → *City* for location). This information is readily available for every user in BC. For ML, we queried Yahoo! Maps[7] to get this information. We manually created hierarchies for attributes age, year and occupation. Other attributes like director, gender, author have trivial hierarchies (i.e., height = 1).

[4] *http://en.wikipedia.org/wiki/Entropy*

[5] *http://en.wikipedia.org/wiki/Gini_index*

[6] *http://www2.informatik.uni-freiburg.de/∼cziegler/BX/*

[7] *https://maps.yahoo.com*

While our framework is general enough to admit any input distribution, we will mostly use 3 intuitive distributions *low, high* and *polarized*, to ease exposition. Particularly, for ML, we use $\{U_{1,2}, U_{4,5}, U_{1,5}\}$ and for BC, $\{U_{1,2,3}, U_{8,9,10}, U_{1,2,9,10}\}$. We use `maxEMD` to denote the maximum value that `EMD` threshold θ can take (4 for ML and 9 for BC). Our threshold is empirically set, starting with a low `EMD` and increasing it if there are not enough output results.

Experiments were conducted on 2 GHz Intel Core i7, 8 GB RAM, MAC OS. Code is written in Java, JDK/JRE 1.6.

5.2 Detailed Evaluation

We use synthetic datasets to stress-test our algorithms over noisy data and show their accuracy (ability to identify the right segments) and rating map quality.

5.2.1 Synthetic Data

We developed a synthetic data generator that provides the flexibility to produce datasets with different distributions and having different percentages of noise. Noise is defined as the mass distributed in different ratings than the ones where the largest mass is assigned. We produce 5 datasets each one consisting of 5,000 rating records and a percentage of noise, ranging from 10% to 50%. Particularly, our ground truth, GT, consists of rating records where each record is associated to a virtual user. Each virtual user has 3 attributes: `age` with values *Teen, Young, Middle, Old*, `occupation` with values *Lawyer, Doctor, Farmer, Sports, Student* and `location` with values *East, Central*, and *West*. For each segment g, a random distribution is assigned given a noise percentage. For example, for a segment described by $\{\texttt{age} = Teen, \texttt{occupation} = Doctor, \texttt{location} = East\}$ and for a 10% noise percentage, a perturbed U_1 distribution, denoted as $\underline{U}_1$, is formed by assigning a large chunk of mass (0.9) to $U_1(1)$ and uniformly distributing the remaining mass (0.1) as noise over other positions $U_1(i), i \neq 1$. The introduction of noise makes it more difficult for our algorithms to identify segments close to input distributions. The `EMD` threshold θ is set as follows: 0.4 for 10% noise, 0.5 for 20%, 0.6 for 30%, 0.7 for 40%, and 0.8 for 50%. We test our algorithms using the input distributions $\{U_1, ..., U_5\}$.

Accuracy Evaluation.

We borrow standard supervised clustering evaluation measures like `Precision`, `Recall` and `F-Measure`,[8] adapted to our context, in order to evaluate the quality of our rating maps.

`Precision` captures the fraction of pure segments in the rating map G, where the purity of a segment g_i is defined as its similarity to its closest segment $g \in GT$. More precisely, it is the product of its description purity (based on Jaccard similarity) and its distribution purity (based on `EMD` distance). A similar remark holds for `Recall` except for the difference in the denominator (i.e., $|GT|$). `F-Measure` is defined as the harmonic mean of `Precision` and `Recall`. The exact formulations are given below.

$\texttt{Precision}(G, GT) = \frac{\sum_{g_i \in G} max_{g \in GT} \texttt{Purity}(g_i, g)}{|G|}$

$\texttt{Purity}(g_i, g) = \texttt{DescPurity}(g_i, g) * \texttt{DistPurity}(g_i, g)$

$\texttt{DescPurity}(g_i, g) = \frac{|g_i.desc \cap g.desc|}{|g_i.desc \cup g.desc|}$

$\texttt{DistPurity}(g_i, g) = \frac{\texttt{maxEMD} - \texttt{EMD}(g_i, g)}{\texttt{maxEMD}}$

Robustness to Noise.

We test the accuracy of all algorithms over the 5 synthetic datasets described earlier. Figure 3 depicts that all algorithms are sensitive to noise and shows a decrease in accuracy as noise increases. However, a proper tuning of θ, ranging from 0.4 to 0.8 results in increasing tolerance to noise. It is worth noting that all algorithms achieve a similar accuracy, which is always higher than 70%. Moreover, `RF-EMD` outperforms all other heuristics, as it is the only heuristic which maximizes, by its design, the distribution purity factor (`DistPurity`). However, it produces lower quality maps on other dimensions (Section 5.2.2).

We also studied the accuracy of our algorithms when noise is not uniformly distributed. E.g., for half of input data, the largest mass is assigned to the highest rating value ($U_{1,5}(5) = mass$) and the remaining mass is assigned as noise to the lowest rating ($U_{1,5}(1) = noise$). We observed no significant difference in accuracy. Thus, our algorithms are not affected by how the noise is distributed over rating values.

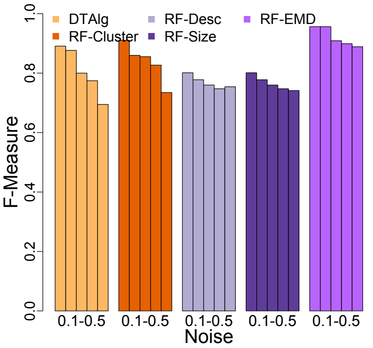


Figure 3: Accuracy of Algorithms vs Noise

Description Length vs Rating Map Quality.

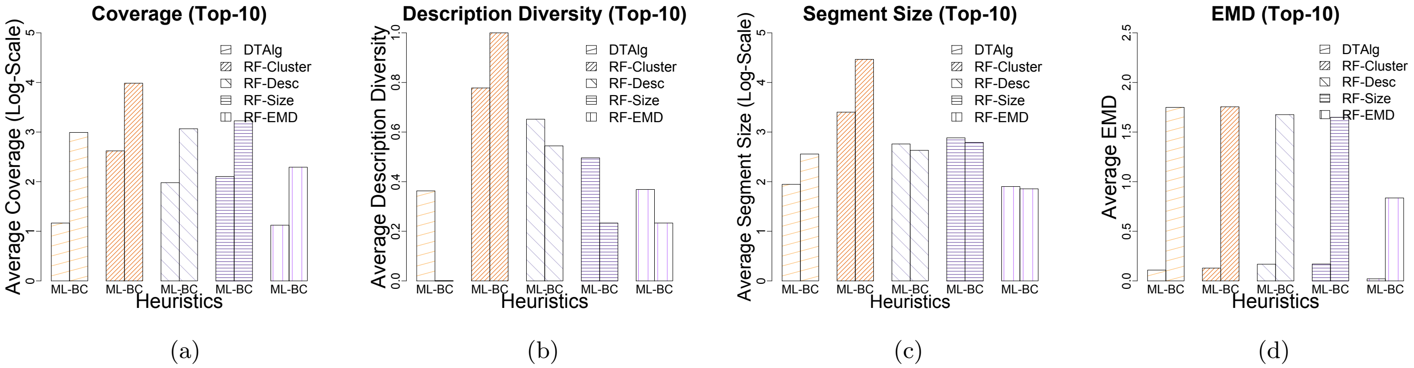
We validate if our optimization objective, i.e., minimizing description length, is a good choice. Coverage indicates the proportion of rating records of a dataset $\mathcal{S}$ included in the resulting rating map G, and is defined as $\texttt{Coverage}(G, \mathcal{S}) = \frac{\sum_{g_i \in G} |records(g_i, \mathcal{S})|}{|\mathcal{S}|}$. The diversity of a rating map G is defined as the average pairwise Jaccard distance between its segments' descriptions, $\texttt{Diversity}(G) = \frac{\sum_{(g_i, g_j) \in G^2, i \neq j} Jaccard(g_i.desc, g_j.desc)}{|\{(g_i, g_j) \in G^2, i \neq j\}|}$.

In order to control the description length, we vary the `EMD` threshold θ and generate trees of different heights. Table 3 summarizes the results of `RF-Cluster` over the first synthetic dataset (3 attributes, 10% noise). The results of the other algorithms are similar and are omitted. We notice that even a slight decrease of description length results in a relatively high increase of our quality criteria (coverage, description diversity, segment size, `EMD`). The value "0" of description length is explained by an empty description with `RF-Cluster` since it is as generated by the pattern mining algorithm of 90% support threshold.

5.2.2 Real Data

We study the quality of rating maps, generated by our heuristics, on real data of ML and BC. We evaluate their

[8] *https://en.wikipedia.org/wiki/Precision_and_recall*



(a)

(b)

(c)

(d)

Figure 4: Evaluation of Rating Map Quality for all RF Heuristics

	$\theta = 0.1, \lvert\mathcal{S}\rvert = 9$	$\theta = 0.5, \lvert\mathcal{S}\rvert = 13$	$\theta = 1, \lvert\mathcal{S}\rvert = 9$
`Description Length`	3	2.77	2.44
`Coverage`	14.34	61.74	96.64
`Description Diversity`	0	0.15	0.31
`Segment Size`	79.67	237.46	536.89
`EMD`	0.04	0.29	0.59

Table 3: Effect of Segment Description Length on Rating Map Quality for `RF-Cluster`

performance over their ability to minimize the optimization objective of description length and improve the various quality criteria. We perform our experiments over the same data sample containing about 1,000 rating records for both ML and BC. We set $m = 20$, $\theta = 0.2$ for ML and $\theta = 2$ for BC.

Quality of Heuristics.

Table 4 shows the average description length of all algorithms. `RF-Cluster` produces population segments with the minimum description length for any dataset. It is worth mentioning that the resulting segments of `RF-Cluster` do not have a natural exact description, but they are assigned a description as generated by the pattern mining algorithm. Thus, setting the support threshold of frequent patterns to 90% can result in segments with an empty description and an average description length lower than 1. Moreover, we observe that all heuristics perform relatively well compared with `DTAlg`, while at the same time they favor other quality criteria (Figure 4).

	`DTAlg`	`RF-Cluster`	`RF-Desc`	`RF-Size`	`RF-EMD`
ML	2.9	0.9	2	2.3	2.9
BC	1	0.25	1.1	1.3	1.7

Table 4: Average Description Length (Top-10)

Figure 4 illustrates the quality of rating maps w.r.t. the various criteria. Each rating map contains the top-10 population segments, as ranked by the different heuristics. For `DTAlg` and `RF-Cluster`, where there is no ranking criterion, we randomly pick 10 segments. Overall, `RF` approaches achieve better results than `DTAlg` for all quality criteria. This confirms the weakness of a single tree to capture the various non-intersecting segments and the benefit of using a forest of trees. We also show that `RF-Cluster` produces rating maps with the highest quality over all criteria. The results of `RF-Size` show that coverage of input records, Figure 4b, is favored by large segments, shown in Figure 4c. `RF-Desc` returns segments with high description diversity, Figure 4b. `RF-EMD` achieves the best average `EMD`, Figure 4d, but it performs poorly on the other quality measures. Finally, all heuristics, except `RF-EMD`, appear to have similar average `EMD` values.

Scalability Evaluation.

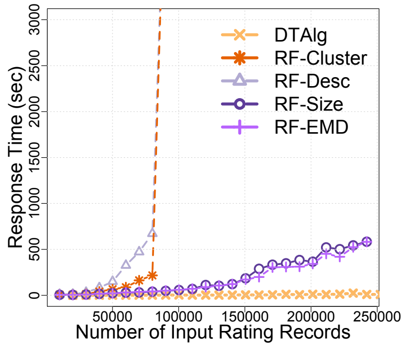


Figure 5: Response Time Vs. Input size on ML

In order to study scalability, we gradually increase the size of input rating records $\mathcal{S}$ using the ML dataset. Our experiments confirm the high scalability of `DTAlg`: it requires around 4 minutes to process 1M records from ML. However, the better quality results of our heuristics arrive with a computational cost. Figure 5 shows the running times of the various heuristics as the size of $\mathcal{S}$ grows. The time taken to create random trees is the same for all `RF` approaches and the difference between them is due to different strategies to combine partitions. Although `RF-Cluster` and `RF-Desc` produce good quality maps, they scale poorly since they iterate over the entire records and generated segments respectively. On the contrary, `RF-Size` and `RF-EMD` scale linearly with the size of $\mathcal{S}$ while producing good quality maps. *We conclude that there is a tradeoff between high quality maps and the time for obtaining them.* `RF-Size` *achieves a good compromise between the quality of rating maps and response time.*

5.3 Exploration Scenarios

In this section, we discuss the utility of rating maps by exploring analysts' and end-users' scenarios. Rating maps are built using RF-Size, which achieves a good compromise between the quality of resulting maps and response time.

5.3.1 Analysts' Scenarios

We set θ, the EMD threshold to 10% of maxEMD i.e., 0.4 for ML and 0.9 for BC. For ML, we tested the quality of rating maps produced by RF-Size with 2 to 32 trees and empirically observed that generating 4 trees resulted in the best average EMD. We hence set that number to 4.

ML Dataset. Our analyst wants to explore the opinion of people for *Drama* movies. A rating map is generated showing that people agree that *Steven Spielberg*, *Tom Hanks* and *Kevin Spacey* directed the best *dramas*. The analyst continues her exploration in order to discover differences between genders for *Drama*s. She discovers that *males* show a preference for directors *Irvin Kershner*, *Quentin Tarantino* and *Mel Gibson*. Particularly, a group of *Californian males* working at a *University* or in *science/technology* show a high preference for *Tarantino*'s *Pulp Fiction (1994)*. Our analyst can hence conclude that *males'* taste in drama depends exclusively on movie attributes (e.g., director), while *females*' varies depending on age and occupation. E.g., *young* women and women in *business administration* love some movies by *Quentin Tarantino*, while women who are *young graduate*s or *artists* love *Braveheart (1995)* by *Mel Gibson*.

In another scenario, the analyst is interested in exploring highly rated romantic movies. The rating map illustrates results of directors *Rob Reiner* and *John Madden*, but also some surprising results; *young* raters liked a romantic movie directed by *Richard Marquand* and *Alfred Hitchcock* is loved for his romantic movies. The analyst decides to further explore these two results. She finds, in a second map, that *young* people who rated *Star Wars: Return of the Jedi*, also classified as romantic, are the ones who love romantic movies by *Richard Marquand*. Furthermore, she discovers a group of female fans of *Alfred Hitchcock*'s *Suspicion (1941)* and *Notorious (1946)*.

BC Dataset. This time, our analyst repeats the same exploratory task for finding *polarized* opinions on books. One prominent observation is that people have polarized opinions on author *J. K. Rowling*. A further exploration of this segment results in a demographics breakdown for that author. The corresponding rating map helps the analyst understand who causes this polarization: people in the USA have polarized opinions on *the first book of Harry Potter*, while *middle-aged* people are polarized on *the fifth book*.

5.3.2 End-Users' Scenarios

ML Dataset. John, a *middle-aged Californian* working in *science/technology*, is interested in finding users like him and users different from him on *adventure* movies. Our input dataset consists of all rating records for *adventure* movies and our input distribution is John's computed from his *192* ratings for *adventure* movies. The EMD threshold is ≤ 0.1 for similar (resp., ≥ 1.2 for dissimilar) in order to impose very close (resp., not-so-close) distributions to John's.

John is active in rating *adventure* movies and he is interested in discovering population segments that share his passion. He decides to find segments with which he shares at least one demographic attribute (e.g., gender, age). He discovers that *young people* with the same occupation and people of the same age from *California* or *Illinois* "agree" with him on *adventure* movies released in the period *1990-1995*. Moreover, John shares the same opinion with people of the same gender working in *universities* and with people of the same gender and occupation on *adventure* movies released in *1990-2000*. Finally, one intriguing finding for John is another reviewer perfectly matching his distribution on *adventure* movies, *a young male artist* living in *Urbana* and who rated *1970-1990* movies, 53 times. On the contrary, John disagrees with a segment of the same age, a set of *self-occupied males* who rated *Star Wars, Episode IV*. Also, he disagrees with people with the same occupation who rated movies directed by *George Lucas* during *1979-1990*, and with *young Californian males* on a movie written by *Leigh Brackett* and *Lawrence Kasdan*.

BC Dataset. Mary, a *32 years old woman* living in *Bethlehem, Pennsylvania, USA*, dislikes books by *Debbie Macomber*. Mary is interested in finding population segments, also located in the *USA*, that have similar or dissimilar opinions as hers. Our input dataset consists of all ratings for books by *Debbie Macomber* and the input distribution is Mary's computed from her *12* ratings on books by *Debbie Macomber*. We set the EMD threshold to ≤ 0.3 for similar users (≥ 3 for dissimilar).

The resulting map illustrates a group of 25 *middle-aged* people in the *USA* with a negative opinion on the book *204 Rosewood Lane* written by *Debbie Macomber*. On the contrary, a small segment of 11 people living in the *USA* "disagree" with Mary as they highly rated the book *Changing Habits*. Mary can get in touch with people in both segments to engage in online debates on the author.

6. RELATED WORK

To the best of our knowledge, this is the first work to deal with exploring rated datasets using rating maps. We review the closest contributions to ours in different areas.

In the database community, Das et al. [6] introduced mining rated datasets with the goal of extracting meaningful demographic patterns that describe users with distributions of the form $U_1, \ldots, U_M$ or with polarized opinions. The proposed problem statements maximize coverage while ours minimizes description length and is shown to produce population segments with high coverage (Section 5.2.1). Our input distributions could have any shape including the ones handled in [6]. Moreover, in [6], the opinion of a population segment is computed as the average of all its rating records. In Section 3, we show why comparing rating averages is less accurate than comparing rating distributions using EMD.

In data mining, subgroup discovery has been concerned with finding data regions where the distribution of a given target variable is substantially different from its distribution in the whole database [12, 11]. Subgroup detection [7] is a related area that is concerned with finding agreeing or disagreeing groups by analyzing their discussions on online forums. Clustering is used to identify such groups based on characterizing each user with a feature vector extracted from discussions. Exceptional model mining [10] is an extension of traditional subgroup discovery where the goal is to find regions of the input space that are substantially different from the entire database. In most cases, a subgroup discovery algorithm performs a top-down traversal of a search lattice. Our approach is more flexible since it can find user groups close to some input distribution and it leverages the additive property of EMD for efficient processing.

Subspace clustering has been used extensively for data exploration [13, 19]. CLIQUE [1] relies on a global notion of density, i.e., the percentage of the overall dataset that falls within a particular subspace. ENCLUS [5] uses information entropy as the clustering objective. CLTree [17] uses a decision-tree approach to identify high-density regions, while Cell-Based Clustering [4] improves scalability with data partitioning. The ability to take into account input distributions to find relevant population segments would require substantial modifications to subspace clustering.

In social media, Choudhury et al. [9] examined opinion biases in the blogosphere, using entropy as an indicator of diversity in opinions. Alternatively, Varlamis et al. [22] proposed clustering accuracy as an indicator of the blogosphere opinion convergence. While our framework is more general, as it handles different input distributions, it does not directly support the online detection of diverging opinions.

7. CONCLUSION

To facilitate online exploration of rated datasets by analysts and end-users, we proposed rating maps consisting of sets of (population segment, rating distribution) pairs with segments that cover a large number of input records, have diverse descriptions and each segment contains many records and is close to one of desired distributions. We formulated the problem of finding rating maps as finding Partition Decision Trees (PDTs) of minimum height and showed that the problem is NP-complete. We proposed a linear time algorithm for finding a basic PDT and heuristics based on random forests for improving their quality. Our extensive experiments show that PDTs with short descriptions (small height) achieve high quality and that among our heuristics, `RF-Size` that selects the largest segments, strikes the best balance between quality of rating maps found and running time. We are currently studying ways to provide more expressivity to analysts in exploring and refining segments interactively.

8. REFERENCES

[1] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data. *Data Mining and Knowledge Discovery*, 11(1):5–33, 2005.

[2] J. Beel, B. Gipp, S. Langer, and C. Breitinger. Research-paper recommender systems: a literature survey. *International Journal on Digital Libraries*, pages 1–34, 2015.

[3] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001. 10.1023/A:1010933404324.

[4] J.-W. Chang and D.-S. Jin. A new cell-based clustering method for large, high-dimensional data in data mining applications. In *Proceedings of the 2002 ACM Symposium on Applied Computing*, SAC '02, pages 503–507, New York, NY, USA, 2002. ACM.

[5] C.-H. Cheng, A. W. Fu, and Y. Zhang. Entropy-based subspace clustering for mining numerical data. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '99, pages 84–93, New York, NY, USA, 1999. ACM.

[6] M. Das, S. Amer-yahia, G. Das, and C. Yu. Mri: Meaningful interpretations of collaborative ratings. *PVLDB*, 4(11):1063–1074, 2011.

[7] P. Dasigi, W. Guo, and M. Diab. Genre independent subgroup detection in online discussion threads: A pilot study of implicit attitude using latent textual semantics. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, pages 65–69, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[8] R. P. David Freedman, Robert Pisani. *Statistics, 4th Edition*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[9] M. De Choudhury, H. Sundaram, A. John, and D. D. Seligmann. Multi-scale characterization of social network dynamics in the blogosphere. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 1515–1516, New York, NY, USA, 2008. ACM.

[10] W. Duivesteijn, A. J. Feelders, and A. Knobbe. Exceptional model mining. *Data Mining and Knowledge Discovery*, 30(1):47–98, 2016.

[11] J. H. Friedman and N. I. Fisher. Bump hunting in high-dimensional data. *Statistics and Computing*, 9(2):123–143, 1999.

[12] W. Klosgen. *Handbook of Data Min. Knowl. Discov, ch. 16.3: Subgroup Discovery*. Oxford Univ., NY, 2002.

[13] H.-P. Kriegel, P. Kröger, and A. Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Trans. Knowl. Discov. Data*, 3(1):1:1–1:58, Mar. 2009.

[14] H. W. Kuhn. The hungarian method for the assignment problem. *NRL Quarterly*, 2:83–97, 1955.

[15] S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86, 03 1951.

[16] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 106–115, April 2007.

[17] B. Liu, Y. Xia, and P. S. Yu. Clustering through decision tree construction. In *Proceedings of the Ninth International Conference on Information and Knowledge Management*, CIKM '00, pages 20–29, New York, NY, USA, 2000. ACM.

[18] *MovieLens*, as of 2003, www.grouplens.org/node/73.

[19] L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data: A review. *SIGKDD Explor. Newsl.*, 6(1):90–105, June 2004.

[20] 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.

[21] P.-N. Tan et al. *Introduction to Data Mining, (First Edition)*. W. W. Norton & Company, 2007.

[22] I. Varlamis, V. Vassalos, and A. Palaios. Monitoring the evolution of interests in the blogosphere. In *Data Engineering Workshop, 2008. ICDEW 2008. IEEE 24th International Conference on*, pages 513–518, April 2008.