

Discovering a Taste for the Unusual Exceptional Models for Preference Mining

Cláudio Rebelo de Sá · Wouter Duivesteijn ·
Paulo Azevedo · Alípio Mário Jorge ·
Carlos Soares · Arno Knobbe

Received: date / Accepted: date

Abstract Exceptional Preferences Mining (EPM) is a crossover between two sub-fields of data mining: local pattern mining and preference learning. EPM can be seen as a local pattern mining task that finds subsets of observations where some preference relations between labels significantly deviate from the norm. It is a variant of Subgroup Discovery, with rankings of labels as the target concept. We employ several quality measures that highlight subgroups featuring exceptional preferences, where the focus of what constitutes ‘exceptional’ varies with the quality measure: two measures look for exceptional overall ranking behavior, one measure indicates whether a particular label stands out from the rest, and a fourth measure highlights subgroups with unusual pairwise label ranking behavior. We explore a few datasets and compare with existing techniques. The results confirm that the new task EPM can deliver interesting knowledge.

Cláudio Rebelo de Sá
LIACS, Universiteit Leiden, the Netherlands and INESC TEC
E-mail: c.f.de.sa@liacs.leidenuniv.nl

Wouter Duivesteijn
Technische Universiteit Eindhoven
E-mail: w.duivesteijn@tue.nl

Paulo Azevedo
HasLab, INESC TEC, Departamento de Informática, Universidade do Minho E-mail:
pja@di.uminho.pt

Alípio Mário Jorge
INESC TEC, Faculdade de Ciências, Universidade do Porto
E-mail: amjorge@fc.up.pt

Carlos Soares
Faculdade de Engenharia, Universidade do Porto
E-mail: csoares@fe.up.pt

Arno Knobbe
LIACS, Universiteit Leiden, the Netherlands
E-mail: a.j.knobbe@liacs.leidenuniv.nl

1 Introduction

Consider a survey where detailed preferences of sushi types have been collected, along with information about the respondents. For each example in the dataset, we have personal details (age, gender, income, etc.) as well as a set of sushi types, ordered by preference [37]. By mapping the demographic attributes and unusual preferences, marketers would be able to target key demographics where specific sushi types have greater potential.

The study of preference data has been approached from a number of perspectives, grouped under the name *Preference Learning* (PL) (e.g. as Label Ranking [18, 11, 48]) Typically, the aim is to build a global predictive model, supported by *preference mining* methods [27], such that the preferences can be predicted for new cases. However, in several areas, such as marketing, there is also great value in identifying subpopulations whose preferences deviate from the norm. If the preference of some sushi type by a certain age group or in a certain region is markedly different from the average population, then the vendor can develop specific strategies for those groups. Finding coherent groups of customers to focus on is an invaluable part of promotion strategies.

In this work, the term *preference* is not strictly interpreted as a literal preference, but instead as an order relation $object_1 \succ object_2$. An order relation can represent several phenomena: a person likes *sushi*₁ more than *sushi*₂ [37]; λ_1 is more likely to occur than λ_2 [33]; *algorithm*₁ is better than *algorithm*₂ [6]. In this context, unusualness is the extent to which some groups show different preferences from average behavior.

Arguably the most generic setting for discovering local, supervised deviations is that of Subgroup Discovery (SD) [40]. The aim of SD is to discover subgroups in the data for which the target shows an unusual distribution, as compared to the overall population [39]. SD is a generic task in the sense that the actual nature of the target variable can be quite diverse. For example, SD approaches have been developed for binary, nominal [1] and numeric target variables [36, 34], as well as multiple targets [22, 46].

We extend the work on Exceptional Preferences Mining (EPM) [16], which focuses on the discovery of meaningful subgroups with exceptional preference patterns. When applying SD to a new context, the main task is to determine what constitutes an *interesting* subgroup. In EPM, different *quality measures* determine the interestingness based on how the preferences in the subgroup, differ from the preferences in the whole data. A set of EPM quality measures reflect different facets of interestingness one might have about the unusualness of a set of preferences.

In this work, we include a more comprehensive experimental setup and propose a new quality measure. We employ EPM on several real-world datasets, using four distinct quality measures. These measures define the type of exception that is identified to either encompass the entire label space or focus on more local peculiarities. In particular, two of them look for overall exceptional preferences; a third measure assesses if one particular label behaves exceptionally; the remaining measure quantifies the exceptional behavior of a single pair of labels.

Finally, to consolidate the previous work on EPM, we compare EPM with a subgroup discovery approach known as Distribution Rules (DR) [35].

We start by introducing Label Ranking in Section 2 and Subgroup Discovery in Section 3. Then, in Section 4 we introduce Exceptional Preferences Mining

and analyze the results obtained in Section 5. Finally, we conclude this paper in Section 6.

2 Label Ranking

In Label Ranking, given an instance x from the instance space \mathbb{X} , the goal is to predict the ranking of the labels $\mathcal{L} = \{\lambda_1, \dots, \lambda_k\}$ associated with x [33]. A ranking can be represented as a *strict total order* over \mathcal{L} , defined on the permutation space Ω .

The Label Ranking task is similar to the classification task, where instead of a class we want to predict a ranking of the labels. As in classification, we do not assume the existence of a deterministic $\mathbb{X} \rightarrow \Omega$ mapping. Instead, every instance is associated with a *probability distribution* over Ω [12]. This means that, for each $x \in \mathbb{X}$, there exists a probability distribution $\mathcal{P}(\cdot|x)$ such that, for every $\pi \in \Omega$, $\mathcal{P}(\pi|x)$ is the probability that π is the ranking associated with x . The goal in Label Ranking is to learn the mapping $\mathbb{X} \rightarrow \Omega$. The training data is defined as D , which is a bag of n records of the form $x = (a_1, \dots, a_m, \pi)$, where $\{a_1, \dots, a_m\}$ is set of values from m independent variables $\mathcal{A}_1, \dots, \mathcal{A}_m$ describing instance x and π is the corresponding target ranking.

Rankings can be represented with total or partial orders and vice-versa.

Total orders A *strict total order* over \mathcal{L} is defined as a binary relation, \succ , on a set \mathcal{L} [9], which is:

1. Irreflexive: $\lambda_a \not\succeq \lambda_a$
2. Transitive: $\lambda_a \succ \lambda_b$ and $\lambda_b \succ \lambda_c$ implies $\lambda_a \succ \lambda_c$
3. Asymmetric: if $\lambda_a \succ \lambda_b$ then $\lambda_b \not\succeq \lambda_a$ ¹
4. Connected: For any λ_a, λ_b in \mathcal{L} , either $\lambda_a \succ \lambda_b$ or $\lambda_b \succ \lambda_a$

A *strict ranking* [48], a *complete ranking* [20], or simply a *ranking* can be represented by a *strict total order* over \mathcal{L} . A strict total order can also be represented as a permutation π of the set $\{1, \dots, k\}$, such that $\pi(a)$ is the position, or *rank*, of λ_a in π . For example, the *strict total order* $\lambda_3 \succ \lambda_1 \succ \lambda_2 \succ \lambda_4$ can be represented as $\pi = (2, 3, 1, 4)$.

However, in real-world ranking data, we do not always have clear and unambiguous preferences, i.e. strict total orders [5]. Hence, sometimes we have to deal with *indifference* [8] and *incomparability* [13]. For illustration purposes, let us consider a survey where a set of n consumers rate k sushi types. If a consumer feels that two sushi types have identical taste, then these can be expressed as indifferent so they are assigned the same rank (i.e. a tie).

To represent ties, we need a more relaxed setting, called *non-strict total orders*, or simply *total orders*, over \mathcal{L} , by replacing the binary strict order relation, \succ , with the binary partial order relation, \succeq where the following properties hold [9]:

1. Reflexive: $\lambda_a \succeq \lambda_a$
2. Transitive: $\lambda_a \succeq \lambda_b$ and $\lambda_b \succeq \lambda_c$ implies $\lambda_a \succeq \lambda_c$
3. Antisymmetric: $\lambda_a \succeq \lambda_b$ and $\lambda_b \succeq \lambda_a$ implies $\lambda_a = \lambda_b$
4. Connected: For any λ_a, λ_b in \mathcal{L} , either $\lambda_a \succeq \lambda_b$, $\lambda_b \succeq \lambda_a$ or $\lambda_b = \lambda_a$

¹ Asymmetry can be derived from 1. and 2. [14].

These non-strict total orders can represent *partial rankings* (rankings with ties) [48]. For example, the *non-strict total order* $\lambda_1 \succ \lambda_2 = \lambda_3 \succ \lambda_4$ can be represented as $\pi = (1, 2, 2, 3)$.

Additionally, real-world data may lack preference data regarding two or more labels, which is known as *incomparability*. Continuing with the sushi survey, if a consumer never tried one or two sushi types, λ_a and λ_b , it leads to incomparability, $\lambda_a \perp \lambda_b$. In other words, the consumer cannot decide whether the sushi types are equivalent or select one as the preferred, because he never tasted at least one of them. In this cases, we can use *partial orders*.

Partial orders Similar to *total orders*, there are *strict* and *non-strict partial orders*. Let us consider the *non-strict partial orders* (which can also be referred to as *partial orders*) where the binary relation, \succeq , over \mathcal{L} is [9]:

1. Reflexive: $\lambda_a \succeq \lambda_a$
2. Transitive: $\lambda_a \succeq \lambda_b$ and $\lambda_b \succeq \lambda_c$ implies $\lambda_a \succeq \lambda_c$
3. Antisymmetric: $\lambda_a \succeq \lambda_a$ and $\lambda_b \succeq \lambda_a$ implies $\lambda_a = \lambda_b$

We can represent partial orders with *subrankings* [31] or *incomplete rankings* [10]. For example, the *partial order* $\lambda_1 \succ \lambda_2 \succ \lambda_4$ can be represented as $\pi = (1, 2, 0, 3)$, where 0 represents $\lambda_1, \lambda_2, \lambda_4 \perp \lambda_3$.

Several learning algorithms proposed for modeling Label Ranking data can be grouped as decomposition-based or direct [17]. *Decomposition methods* divide the problem into several simpler problems (e.g., multiple binary problems). An example is ranking Ranking by Pairwise Comparisons (RPC) [26], which decomposes the LR problem into a set of binary classification problems. A learning method is trained with all examples for which either $\lambda_i \succ \lambda_j$ or $\lambda_j \succ \lambda_i$ is known [26]. The resulting predictions are then combined to predict a total or partial ranking [11]. *Direct methods*, on the other hand, treat the rankings as target objects without any decomposition. Examples of that include decision trees [45, 12], *k*-Nearest Neighbors [6, 12] and the linear utility transformation [29, 19].

3 Subgroup Discovery and Exceptional Model Mining

Subgroup Discovery (SD) [39] is a data mining framework that seeks subsets of the dataset (satisfying certain user-specified constraints) where something exceptional is going on. In SD, we assume a flat-table dataset D , which is a bag of n records of the form $x = (a_1, \dots, a_m, t_1, \dots, t_\ell)$. We call $\{a_1, \dots, a_m\}$ the *descriptors* and $\{t_1, \dots, t_\ell\}$ the *targets*, and we denote the collective domain of the descriptors by \mathcal{A} . We are interested in finding interesting subsets, called *subgroups*, that can be formulated in a *description language* \mathcal{D} . In order to formally define subgroups, we first need to define the following auxiliary concepts.

Definition 1 (Pattern and coverage) Given a description language \mathcal{D} , a *pattern* $p \in \mathcal{D}$ is a function $p : \mathcal{A} \rightarrow \{0, 1\}$. A pattern p covers a record x iff $p(a_1, \dots, a_m) = 1$.

Patterns induce subgroups, and subgroups are associated with patterns, in the following manner.

Definition 2 (Subgroup) A *subgroup* corresponding to a pattern p is the bag of records $S_p \subseteq D$ that p covers:

$$S_p = \{x \in D \mid p(a_1, \dots, a_m) = 1\}$$

The exact choice of the description language is left to the domain expert or analyst. A typical choice is the use of conjunctions of conditions on attributes. Restricting the findings of SD from all subsets to only subgroups that can be defined in such a way, yields results of the following form:

$$\text{Age} \geq 30 \wedge \text{Likes} = \text{Salmon Roe} \quad \text{is unusual}$$

instead of the form:

$$S \subseteq D \quad \Rightarrow \quad \text{interesting.}$$

SD delivers subgroups in a form with which the dataset domain experts are familiar. In other words, the focus of SD lies on delivering *interpretable* results.

Formally, the interestingness of a subgroup can be measured using any characteristics available from its associated pattern. In practice, it depends on the task we are trying to solve. Therefore, we should define one or more *quality measures* to assess the interestingness we want to explore.

Definition 3 (Quality Measure) A *quality measure* is a function $\varphi : D \rightarrow \mathbb{R}$.

In the most common form of pattern mining, *frequent itemset mining* [2], interestingness is measured by the frequency of the pattern. *Subgroup Discovery* [39], on the other hand, measures interestingness in a *supervised* form. One designated target variable t_1 is identified in the dataset, and subgroup interestingness is measured by an unusual distribution of that target. Hence, considering that a survey revealed that the majority of Japanese people like *Fatty tuna* sushi, an interesting subgroup could refer to a group of people for which the majority prefers *Tuna roll*:

$$\text{Age} \geq 30 \wedge \text{Lives in region} = \text{Hokkaido} \quad \Rightarrow \quad \text{Likes} = \text{Tuna roll}$$

If instead of a single target, multiple targets t_1, \dots, t_ℓ are available, and if we are not interested in finding unusual target distributions, but unusual target interactions, we can employ *Exceptional Model Mining* (EMM) [21,23] instead of SD. EMM is instantiated by selecting two things: a model class and a quality measure. Typically, a model class is defined to represent the unusual interaction between multiple targets we are interested in. A specific quality measure that employs concepts from that model class must be defined to express exactly when an interaction is unusual and, therefore, interesting. For example, suppose that there are two target attributes: a person's height (t_1), and the average height of his/her grandparents (t_2). We may be interested in the correlation coefficient between t_1 and t_2 . In this case, we would use EMM with the *correlation model class* [41]. Given a subgroup $S \subseteq D$, we can estimate the correlation between the targets within this subset by the sample correlation coefficient.

For very small subgroups, one easily finds an unusual distribution of the target. Hence, to favor larger subgroups, one defines the quality measure such that it balances the exceptionality of the target distribution with the size of the subgroup.

3.1 Search Strategy

In the EMM process, we explore a large search space, guided by a user-defined quality measure that expresses the type of exceptionality we seek. Typically, *subgroups* are found by a level-wise search through attribute space [21]. However, we consider the exact search strategy to be a parameter of the algorithm.

EMM strives to find descriptions that satisfy certain user-specified constraints. Usually these constraints include lower bounds on the quality of the description and size of the induced subgroup. More constraints may be imposed as the question at hand requires; domain experts may for instance request an upper bound on the complexity of the description.

Most SD algorithms traverse the search space of candidate descriptions in a general-to-specific way: they treat the space as a lattice whose structure is defined by a *refinement operator* $\eta : \mathcal{D} \rightarrow 2^{\mathcal{D}}$. This operator determines how descriptions can be extended into more complex descriptions by atomic additions. Most applications (including ours) assume η to be a *specialization operator*: every description $q \in \mathcal{D}$ that is an element of the set $\eta(p)$, is more specialized than the description p itself. The algorithm results in a ranked list of descriptions (or the corresponding subgroups) that satisfy the user-defined constraints.

In this EMM setting, the *best-first search* strategy is chosen. At each level, the descriptions according to our quality measure φ are sorted, and refined to create the candidate descriptions for the next level. We define constraints on single attributes and define the corresponding subgroups as those records satisfying each one of those constraints. The search is constrained by an upper bound on the complexity of the description (also known as the *search depth*, d) and a lower bound on the support of the corresponding subgroup.

3.1.1 Best-first Search Algorithm in EMM

In Algorithm 1, we outline the pseudo-code of the Best-first search algorithm for EMM. In this code, we assume that there is a subroutine called `SATISFIESALL` that tests whether a candidate description satisfies all conditions in a given set. The `PriorityQueue()` is a queue, with unbounded length, where the elements are stored and sorted with the corresponding quality; One elementary operation, *insert_with_priority*, is for adding an element to the `PriorityQueue`.

The `resultSet` is a `PriorityQueue` maintaining the descriptions ordered by the quality measure. Nothing is ever explicitly removed from the `resultSet`. Hence, the `resultSet` maintains the final result that we seek. When all candidates have been explored or the maximum time is exceeded, the execution ends.

3.2 Distribution Rules

Distribution Rules (DR) is a SD method that analyzes a single target variable. However, rather than a representative value (e.g. the mean), DR identify unusual distributions of the target [35,43]. The approach finds subgroups, expressed as association rules with a statistical distribution on the consequent. A DR may be formally defined as:

$$S \rightarrow t = Dist_t|A$$

Algorithm 1 Best-first Search for Exceptional Model Mining

Input: Dataset \mathcal{D} , QualityMeasure φ , RefinementOperator η , Integer d , Constraints \mathcal{C}

Output: resultSet

```

1 : candidateQueue  $\leftarrow$  new PriorityQueue();
2 : candidateQueue.enqueue({}); ▷ Start with empty description
3 : resultSet  $\leftarrow$  new PriorityQueue();
4 : while (candidateQueue  $\neq$   $\emptyset$ ) do
5 :   seed  $\leftarrow$  candidateQueue.dequeue();
6 :   set  $\leftarrow$   $\eta$ (seed);
7 :   for all (desc  $\in$  set) do
8 :     quality  $\leftarrow$   $\varphi$ (desc);
9 :     if (desc.SATISFIESALL( $\mathcal{C}$ )) then
10 :       resultSet.insert_with_priority(desc, quality);
11 :       candidateQueue.insert_with_priority(desc, quality);
12 :     end if
13 :   end for
14 : end while
15 : return resultSet;

```

where S is a set of conditions corresponding to the antecedent part of a DR (a subgroup), t is a *property of interest* (or target) and $Dist_t|S$ is an empirical distribution of t when S is observed. $Dist_t|S$ is represented by a set of pairs $\langle t_i, freq(t_i) \rangle$, where t_i is one particular value of t found when S is observed and $freq(t_i)$ is the frequency of t_i when the items from S are observed.

4 Exceptional Preferences Mining

Exactly what constitutes an interesting deviation in preferences is governed by the employed quality measure, and the target concept (binary, numeric, preferences, ...). Thus, different measures are required to evaluate different types of targets. SD approaches have been developed for binary, nominal [1] and numeric target variables [34, 36], for targets encompassing multiple attributes [46] and also distributions [35] (Section 3.2). However, none of these approaches is able to capture all the sets of preferences that can be derived from rankings within a SD framework. For that we use, Exceptional Preferences Mining (EPM) [16], which is the search for subgroups with deviating preferences.

In EPM, the target concept at hand consists of a single target t , which would make sense in SD. However, that target object is a ranking of labels, $\pi \in \Omega$, as defined in Section 2. Hence it represents interactions between multiple individual labels, which is more consistent with the EMM scenario.

Some other approaches to mine preferences and ranks can be found in the literature [31, 47]. However, these approaches tackle different problems from the one we address in this paper. In [31], the authors suggest an approach to mine the rankings with association rules that search for *subranking* patterns. Our approach goes beyond this as it relates the ranking patterns with descriptors (otherwise referred to as independent variables). From a different perspective, [47] suggests a *ranked tiling* approach to search for rank patterns, whereas we are interested in the preference relations derived from the ranks.

In the Label Ranking context (Section 2), when the number of labels is large, the search for preference patterns can be hard to analyze and visualize. A real-world example is the Sushi dataset [37], which represents the preferences of 5,000 persons over 10 types of sushi. Even this relatively modest number of sushi types can be ranked in a large number of combinations. This may have a significant effect on the data, as it is shown in this dataset, where more than 98% of the 5,000 rankings present in this dataset are unique. This illustrates why it can be more difficult to directly learn a ranker that associates a reliable complete ranking for any subset in the instance space, \mathbb{X} , when the number of labels is non-trivial.

4.1 Preference Matrix

Before we discuss the approach in detail, it is useful to introduce an alternative representation of rankings that can be useful to look for different categories of exceptionality. Let us define a function, ω , assigning a numeric value to the pairwise comparison of the labels λ_i and λ_j :

$$\omega(\lambda_i, \lambda_j) = \begin{cases} 1 & \text{if } \lambda_i \succ \lambda_j \text{ (}\lambda_i \text{ preferred to } \lambda_j\text{)} \\ -1 & \text{if } \lambda_i \prec \lambda_j \text{ (}\lambda_j \text{ preferred to } \lambda_i\text{)} \\ 0 & \text{if } \lambda_i \sim \lambda_j \text{ (}\lambda_i \text{ indifferent to } \lambda_j\text{)} \\ n/a & \text{if } \lambda_i \perp \lambda_j \text{ (}\lambda_i \text{ incomparable to } \lambda_j\text{)} \end{cases}$$

Note that, by definition, $\omega(\lambda_i, \lambda_j) = -\omega(\lambda_j, \lambda_i)$. We can use ω to represent a ranking π as a *Preference Matrix* (PM), M_π :

$$M_\pi(i, j) = \omega_\pi(\lambda_i, \lambda_j)$$

M_π is, by definition, an antisymmetric matrix with trace equal to zero, $tr(M_\pi) = 0$. PMs can represent partial or incomplete orders but can also be aggregated to represent sets of rankings from an entire dataset D or subgroup S . To aggregate the entries, the *mean* or the *mode* can be used.

The generation of a PM is basically a pairwise decomposition problem. The complexity is $\mathcal{O}(sk^2)$ per subgroup, where s is the size of the subgroup and k the number of labels in the ranking. Even though any number of labels is theoretically permitted in label ranking, in practice the number of labels is usually smaller than 20. Hence, the computational cost of generating PMs should not be a problem.

4.1.1 Preference Matrix of a set of rankings

The PM of a set of rankings from a dataset D with n rankings, M_D , aggregated with the mean is defined by:

$$M_D(i, j) = \frac{1}{n} \sum_{\pi \in D} M_\pi(i, j)$$

where M_π is the PM of the ranking π .

The PM of the example dataset \hat{D} (cf. Table 1) is the following:

$$M_{\hat{D}} = \begin{bmatrix} 0 & 0 & 0 & 0.5 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 \\ -0.5 & 0 & -1 & 0 \end{bmatrix}$$

If entry $M_{\hat{D}}(i, j) = 1$ or $M_{\hat{D}}(i, j) = -1$, then all rankings in \hat{D} agree that $\lambda_i \succ \lambda_j$ or $\lambda_i \prec \lambda_j$, respectively. This means that this representation enables easy detection of strong partial order relations in a set. If row i has all the values very close to 1, then λ_i is systematically preferred to the remaining labels in the corresponding dataset.

Table 1 Example dataset \hat{D} . The first column is the only descriptor. The subsequent four columns represent the preferences among four labels, by providing their ranks. An alternative representation is presented in the rightmost section of the table.

| \mathcal{A}_1 | π | | | | alternative π |
|-----------------|-------------|-------------|-------------|-------------|---|
| | λ_1 | λ_2 | λ_3 | λ_4 | |
| 0.1 | 4 | 3 | 1 | 2 | $\lambda_3 \succ \lambda_4 \succ \lambda_2 \succ \lambda_1$ |
| 0.2 | 3 | 2 | 1 | 4 | $\lambda_3 \succ \lambda_2 \succ \lambda_1 \succ \lambda_4$ |
| 0.3 | 1 | 4 | 2 | 3 | $\lambda_1 \succ \lambda_3 \succ \lambda_4 \succ \lambda_2$ |
| 0.4 | 1 | 3 | 2 | 4 | $\lambda_1 \succ \lambda_3 \succ \lambda_2 \succ \lambda_4$ |

For instance, the records in the illustrative dataset \hat{D} contain distinct total orders (Table 1). But its PM clearly shows that λ_3 is always preferred to λ_2 ($M_{\hat{D}}(3, 2) = 1$). This information, which can be easily obtained from the PM, is harder to read directly from the two columns in Table 1) representing λ_2 and λ_3 : even though, if we analyze carefully, λ_3 is always preferred to λ_2 , this pattern is based on different ranks, namely, $3 > 1$, $2 > 1$, $4 > 2$ and $3 > 2$. Thus, unless one is looking specifically for this pattern, it would be quite hard to find. In real datasets, with more examples and labels, the task would be even harder. Conversely, λ_4 is never preferred to λ_3 , which is represented by $M_{\hat{D}}(4, 3) = -1$. In some cases, the overall trend is not as clear (e.g. λ_1 is preferred to λ_4 but not always) and in other cases, there is no trend at all (e.g. λ_1 and λ_2).

Representing a set of rankings as a PM has another advantage over the traditional permutation representation. On a PM, we can naturally derive a varied set of metrics to search for preference patterns in a set of rankings by characterizing parts of the matrix. For example, it enables simple *labelwise* (by rows/columns of the PM) and *pairwise* (by single entries of the PM) analysis of preferences (see Section 4.3).

From the PM of a subgroup S , one can derive a new ranking π_S . How to do so is a non-trivial question, which has received a lot of attention in several research fields with similar types of matrix [33]. The straightforward way is to sum the rows of the PM and then assign a score to each corresponding label. Higher values correspond to a relatively more preferred label.

On the other hand, PMs can also have limitations in comparison to the traditional permutations representation. Due to the choice of the aggregation metrics, specially in the presence of ties, the PMs can hide the real nature of the rankings. For example, when half of the rankings is $\lambda_1 \succ \lambda_2 \succ \lambda_3 \succ \lambda_4$ and the other half

$\lambda_4 \succ \lambda_3 \succ \lambda_2 \succ \lambda_1$, this results in a PM with all entries equal to zero. Because the same will happen if all rankings are complete ties, there is no way for the method to "see" this obvious difference in the preferences.

In an attempt to mitigate this, subgroups with a PM containing only zeros are not taken into consideration for this work. That is, only subgroups for which we can infer at least one pairwise preference can be considered interesting in this Exceptional Preferences Mining approach.

Finally, to aid in the interpretation of ranking trends within subgroups we use a visual representation of the PMs that is a set of colored tiles (Figure 1). Each tile represents an entry of the PM. The entries of a PM can vary from -1 to 1 . The negative entries of the matrix are represented with red tiles, the positive with green tiles, and 0 is represented in white. The colored tiles fade out as they get closer to 0 .

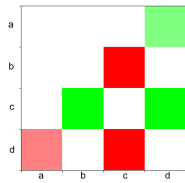


Fig. 1 PM representation of the set of rankings in \hat{D} (cf. Table 1). Dark green tiles represent 1 and dark red tiles represent -1 .

4.2 Characterizing Ranking Exceptionality

In EPM, we want to search for exceptional preference (or ranking) behavior. Because preferences are represented with *rankings*, we can distinguish three categories of exceptionality concerning rankings: *rankingwise*, *labelwise* and *pairwise*.

Measures that fall into the first category, *rankingwise*, will benefit subgroups with exceptional *complete rankings*. This is, if the average ranking of the population is $\lambda_1 \succ \lambda_2 \succ \lambda_3 \succ \lambda_4$, subgroups with an average ranking of $\lambda_4 \succ \lambda_3 \succ \lambda_2 \succ \lambda_1$ will be deemed the most interesting. However, finding a reasonable set of rankingwise exceptional preferences can be challenging in some cases. Considering the example of the Sushi dataset mentioned before, with more than 98% of unique rankings, it will be difficult to observe unusual complete rankings that occur very frequently, due to the low number of ranking repetitions.

Labelwise measures, are less restrictive and focus on subgroups where at least one label is unusually ranked higher (or lower) in comparison to the whole population. The preferences of these subgroups can be represented as incomplete rankings. Considering a population where we observe that $\lambda_1, \lambda_2, \lambda_3 \succ \lambda_4$, therefore, subgroups where $\lambda_4 \succ \lambda_1, \lambda_2, \lambda_3$ will be interesting. Note that, the following list of complete rankings agree with $\lambda_4 \succ \lambda_1, \lambda_2, \lambda_3$: $\lambda_4 \succ \lambda_3 \succ \lambda_2 \succ \lambda_1$, $\lambda_4 \succ \lambda_2 \succ \lambda_3 \succ \lambda_1$, $\lambda_4 \succ \lambda_3 \succ \lambda_1 \succ \lambda_2$, $\lambda_4 \succ \lambda_1 \succ \lambda_2 \succ \lambda_3$ and $\lambda_4 \succ \lambda_1 \succ \lambda_3 \succ \lambda_2$. As an example, if a subgroup ranks *tekka-maki* consistently in the top 3 while the majority in the dataset ranks it in the last 3, this type of measures will find it to be very interesting.

Finally, pairwise measures focus on unusual pairwise preferences. Considering a population where the majority agrees that $\lambda_1 \succ \lambda_4$, any subgroup where most of the subjects agree that $\lambda_4 \succ \lambda_1$ will be considered very interesting. This means that, if a population displays this preference $tamago \succ kappa-maki$, a subgroup where most people prefer $kappa-maki \succ tamago$ will be deemed interesting by these type of measures. Our assumption is that, even though over 98% of the total rankings in the Sushi dataset are unique, there is plenty of information present in these rankings: the partial orders and pairwise comparisons can reveal interesting subgroups.

4.3 Characterizing Exceptional Subgroups

In this section we formally define the quality measures for EPM, which evaluate how exceptional the preferences are in the subgroups. A subgroup can be considered interesting both by the amount of deviation (distance) and by its size (number of records *covered* by the subgroup, as discussed in Section 3) [25]. Since, reasonable quality measures should take both these factors into account, we divide the quality measures into two parts: the *distance* component and the *size* component.

$$QM_S = size_S \cdot distance_S$$

In order to allow direct comparisons between different quality measures, both components are normalized to the interval $[0, 1]$. A common measure for the size in Subgroup Discovery is \sqrt{s} [38], where s is the size of the subgroup. To normalize, we use the square root of the fraction of the dataset covered by S : $size_S = \sqrt{s/n}$.

Before introducing the distance components, let us first define a distance (or difference) matrix L_S , as the distance matrix between two PMs, M_S and M_D :

$$L_S = \frac{1}{2} (M_D - M_S)$$

where $S \subseteq D$ (the division by 2 limits the distance to the interval $[-1, 1]$). We can measure different properties of L_S and represent them with a numeric value. This way we get an indicator of the quality of the distance of preferences for a subgroup. Consider the subgroup $\hat{S}_1 : A_1 \geq 0.3$, which covers the last two cases from our example dataset \hat{D} . Its PM is:

$$M_{\hat{S}_1} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ -1 & 0 & -1 & 0 \\ -1 & 1 & 0 & 1 \\ -1 & 0 & -1 & 0 \end{bmatrix}$$

The first row clearly reveals that λ_1 is always preferred to all other labels in this subgroup. If we compute the distance matrix $L_{\hat{S}_1}$ we get:

$$L_{\hat{S}_1} = \begin{bmatrix} 0 & -0.5 & -0.5 & -0.25 \\ 0.5 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 \\ 0.25 & 0 & 0 & 0 \end{bmatrix}$$

Thus, the distance matrix $L_{\hat{S}_1}$ confirms that the behavior of λ_1 is exceptional in \hat{S}_1 while for the other labels, the behavior is the same as in the original dataset.

4.4 Quality Measures

In this section we introduce the quality measures used in this work. We propose 4 quality measures: 2 rankingwise, 1 labelwise and 1 pairwise (Section 4.2). We describe 3 previously proposed measures [16] and introduce a new one.

As we are interested in subgroups with exceptional preferences, we should be able to measure a preference distance. For that we can use the distance matrix L_S . The distance measures we employ, typically consider a particular subset of the entries of the distance matrix L_S .

4.4.1 Rankingwise measures

Rankingwise quality measures should prefer subgroups whose average rankings are very different to the average ranking of the complete dataset, i.e. maximizing the distance between complete rankings.

Rankingwise Norm If one is searching for subgroups whose average ranking is as close as possible to the inverse ranking of the population, one should use the Rankingwise Norm quality measure, *RWNorm*. Given a set of subgroups with same size, this measure gives the highest score to subgroups whose rankings are the inverse of the population.

In other words, this is done by maximizing all the entries of the distance matrix L_S . Maximizing the distance of preferences is also maximizing the magnitude of L_S . The most fundamental mathematical way to measure the magnitude of a vector or matrix is the *norm*. Hence we can use the *Frobenius norm* of L_S as a distance measure.

$$\text{RWNorm}(S) = \sqrt{s/n} \cdot \|L_S\|_F = \sqrt{s/n} \cdot \sqrt{\sum_{i=1}^k \sum_{j=1}^k L(i, j)^2}$$

As an alternative representation, in some cases, one can use the most frequent values contained in the entries of the distance matrix L_S . That is, one or several modes could be used to represent the preference of a population. Therefore, we define *RWNorm – Mode* as an alternative quality measure where the entries of the PM of the dataset, M_D , and the subgroup, M_S , are aggregated with the mode. In our case, since we can only represent one mode inside a PM, in cases where there are two or more, the median is used.

Rankingwise Covariance Covariance is used in statistics to measure the extent to which two variables change in comparison with each other. In simple terms, a positive value indicates that when one increases, the other also increases. If they behave in opposite directions, the covariance is negative.

As in *RWNorm*, we are interested in subgroups with complete rankings that contradict the preferences in the general population. Hence, we can use covariance to measure the deviations of preferences. The entries of a row in the PM M_S represent how a label relates to the remaining labels in the subgroup S . By abuse of notation, the rows of M_S and M_D can be seen as independent variables, which allows us to measure the covariance between labels. That is, we can compare the

PM values of a label in a subgroup S with the corresponding values of the same label in D using their covariance.

Since our aim is to find opposite preferences in comparison to the population, we are interested in a negative covariance:

$$RWCov(S) = -\sqrt{s/n} \cdot cov(M_D, M_S)$$

In comparison to *RWNorm*, we expect this measure to be more conservative because it requires that most of the entries behave in opposite directions. On the other hand, this measure is better at distinguishing one subgroup whose overall deviation is due to one label deviating strongly and the others not so much, from one where all labels have small deviations.

4.4.2 Labelwise measures

Labelwise measures look for unusual behavior in parts of the rankings. Depending on the application at hand, subgroups might be considered interesting if, at least, parts of their rankings, are in the opposite order of the ranking of the population. For example, a data analyst might be interested in finding subgroups where the preference for a particular sushi type behaves substantially different, when compared to its behavior on the population. That is, the fact that only one label behaves differently, disregarding the interaction between the other labels, can also be interesting [11].

Because rankings have inter-label relations that can be explored [31], there are many ways to tackle this, for example, to use less restrictive measures to look for unusual behaviors of partial rankings.

Labelwise Norm We can measure the preference distance of each label, in a subgroup S , by computing the *norm* of the rows from L_S . This measure considers only the maximum value of the set of rows, hence high values of the measure indicate that, at least, one label behaves differently:

$$LWNorm(S) = \sqrt{s/n} \cdot \max_{i=1, \dots, k} \sum_{j=1}^k L(i, j)$$

Other examples of labelwise measures could be, for example, a variant of this one, but based on the second highest score by label. In that case, it would find subgroups where at least 2 labels are behaving in an unusual way. We could also consider a labelwise covariance, which would focus in the maximum covariance of the each row of L_S .

4.4.3 Pairwise measures

In PL, Pairwise Preferences [33] are often the focus of the analysis, decomposing the preferences into pairs label-vs-label. In EPM, if we are interested in subgroups with at least one pair of labels with distinctive preference behavior we can use pairwise measures.

Pairwise Max We can employ the following pairwise quality measure:

$$PWMax(S) = \sqrt{s/n} \cdot \max_{i,j=1,\dots,k} L(i,j)$$

This quality measure is the least restrictive of this set: a subgroup is interesting if one pair of labels interacts unusually, disregarding all other label interactions.

One alternative pairwise measure could be the pairwise minimum, which would provide the lower bound of *PWMax* for each subgroup.

4.5 Tackling False Discoveries

In SD, one aims to find subsets of the dataset that are interesting in some sense. As such, the space of candidates to be considered for what essentially amounts to a statistical test is vast. Hence, SD suffers from the multiple comparisons problem [32]: when testing a large number of a null hypotheses, by definition, some will incorrectly be rejected. Namely, with a significance level of α , α out of each 100 null hypotheses tested are expected to be incorrectly rejected.

For supervised local pattern mining, to which SD belongs, a swap-randomization-based statistical test procedure has been developed [24]. First, a number of copies of the original dataset is generated, and in each of the copies the target attributes are swap randomized. All other attributes are kept intact. This means that the search space of the mining algorithm and the distribution of the targets remains intact, but the connections between the search space and the target space are broken. The procedure then involves running the algorithm to be tested on each copy of the dataset, and reporting the best subgroup found, according to the selected quality measure. Any subgroup that is found on such a copy of the dataset is interesting only because of random effects. Hence, these are artificially generated false discoveries. The procedure then builds a global model over the artificial false discoveries, the so-called *Distribution of False Discoveries* (DFD). Then, the subgroups found on the original dataset can be assigned a p-value, corresponding to the null hypothesis that a subgroup with this quality is generated by the same process that generated the DFD. Refuting the null hypothesis essentially refutes the hypothesis that the subgroup found is a false discovery.

The DFD validation procedure has only one parameter: the number of dataset copies. This number must be large enough to satisfy certain conditions arising in the global modeling involved in creating the DFD. As noted in [24], typically, 100 copies are enough.

5 Experiments

In this section we start with a description of the experimental setup (Section 5.1), then we present some statistics of the datasets used (Section 5.2). Then we present the results obtained (Section 5.3) and finally we compare our findings with the results of an alternative approach (Section 5.4).

5.1 Implementation and Experimental Setup

We incorporate Exceptional Preferences Mining in the Cortana² software package [44]. This package delivers a generic framework for SD, implements several SD instances, and offers many generic features allowing for different SD approaches. The description language consists of logical conjunctions of conditions on single attributes.

Our experiments use a greedy *best-first search* approach (Algorithm 1). The numeric strategy used for this experiments is an *on the fly* discretization approach of 8 equal-width bins. For every extreme of the bin we use a set of numeric operators such as \geq and \leq .

All the findings we present in this paper have gone through the DFD validation procedure (Section 4.5) with 100 copies, and all have been found significant at a significance level of $\alpha = 1\%$.

All the subgroups presented in this manuscript were found in less than 3 minutes of execution time, on an Intel Core i7 5500U CPU @ 2.40GHz with 16GB of RAM. The DFD validation procedure, for depths bigger than 4 can take more than 30 minutes, depending on the dataset.

5.2 Datasets

To illustrate domain-specific interpretation of the results, we experiment with some real-world datasets (Table 2). The *Algae* dataset,³ is based on the COIL 1999 Competition Data from UCI [42]. This dataset concerns the frequencies of algae populations in different environments. This dataset consists of 340 examples, each representing measurements of a sample of water from different European rivers in different periods. The measurements include concentrations of chemical substances such as nitrogen (in the form of nitrates, nitrites and ammonia), oxygen and chlorine. Also the pH, season, river size and flow velocity are registered. For each sample, we have the preference relations of 7 types of algae which represent the concentrations ordered from larger to smaller concentrations. Those with 0 frequency are placed in last position and equal frequencies are represented with ties. Missing values are set to 0.

The *Sushi* preference dataset [37], is composed of demographic data about 5,000 people and their sushi preferences. Each person sorted a set of 10 different sushi types by preference. The 10 types of sushi, are a) shrimp, b) sea eel, c) tuna, d) squid, e) sea urchin, f) salmon roe, g) egg h) fatty tuna, i) tuna roll and j) cucumber roll.

The *Top7movies* dataset is a subset of the *MovieLens 1M Dataset* [30]⁴. The original dataset has 1 million ratings from 6000 users on 4000 movies. For each user, we have its demographic data, such as gender, age, occupation and zipcode. Using the *zipcode* R package [7], we obtained the city, state, latitude and longitude related to the given zipcodes of the users. We selected the subset of users which have rated all the 7 most rated movies. This means that, in the end we obtained

² <http://datamining.liacs.nl/cortana.html>

³ <http://dx.doi.org/10.17632/spwmg2z7cv.2>

⁴ <https://grouplens.org/datasets/movielens/1m/>

demographic data and a ranking of 7 movies per user. The labels in this dataset represent the following movies:

- a) American Beauty (1999)
- b) Star Wars: Episode IV - A New Hope (1977)
- c) Star Wars: Episode V - The Empire Strikes Back (1980)
- d) Star Wars: Episode VI - Return of the Jedi (1983)
- e) Jurassic Park (1993)
- f) Saving Private Ryan (1998)
- g) Terminator 2: Judgment Day (1991)

Examples which contained rankings with complete ties were removed.

We also study data with socio-economic information from regions of Germany and its electoral results, the datasets *GermanElections2005* and *GermanElections2009*. The 413 records correspond to the administrative districts of Germany, which are described by 39 attributes. Both datasets are parts of data which was extracted from a publicly available database of the German Federal Office of Statistic [4]. A similar study has been presented in [28], but restricted to the city of Cologne.

In terms of independent attributes we have: age and education of the population, economic indicators (e.g. GDP growth, percentage of unemployment), indicators of the labor workforce in different sectors such as production, public service, etc. In terms of the target, we transformed the election results of the five major political parties for the federal elections in 2005 and 2009 into rankings. In this dataset the labels represent:

- a) CDU (conservative)
- b) SPD (center-left)
- c) FDP (liberal)
- d) GREEN (center-left)
- e) LEFT (left-wing)

We also choose to experiment with a Label Ranking dataset from the Data Repository of Paderborn University⁵, since this set of data is well-known in the preference learning community [12]. In particular, we use the *Cpu-small* dataset which was transformed from a regression dataset [12]. The target ranking, with 5 labels, was derived for each example from the order of the values of 5 numerical variables (which are then no longer used as independent variables). In the process, the features were normalized, and its names replaced by A_1, A_2, \dots, A_6 . Therefore, in this case, the reported subgroups cannot be interpreted as in the original dataset domain.

The percentage of unique rankings U_π (Table 2) measures the proportion of distinct rankings in the dataset:

$$U_\pi = \frac{\#\text{distinct rankings}}{n}$$

where n is the size of the data. We also show the expected number of different rankings given n examples, $E(U_\pi)$. This is, if we randomly pick n rankings of a

⁵ <https://www-old.cs.uni-paderborn.de/fachgebiete/intelligente-systeme/software/label-ranking-datasets.html>

Table 2 Dataset details. The column U_π represents the percentage of unique rankings.

| Datasets | #examples | #labels | #attributes | U_π | $E(U_\pi)$ |
|---------------------|-----------|---------|-------------|---------|------------|
| GermanElections2005 | 412 | 5 | 31 | 5% | 28% |
| GermanElections2009 | 412 | 5 | 33 | 7% | 28% |
| Top7movies | 602 | 7 | 7 | 52% | 94% |
| Algae | 316 | 7 | 11 | 72% | 96% |
| Sushi | 5 000 | 10 | 10 | 98% | 99% |
| Cpu-small | 8 192 | 5 | 6 | 1% | 1% |

fixed size k , we should expect $E(U_\pi)$ rankings. By comparison with U_π we can have an idea if there are any biases in the behaviors of the rankings.

Considering the case of the Sushi dataset (Table 2), with an $U_\pi = 98\%$, if we randomly pick 100 instances (i.e. 100 users and its rankings), we will probably have 98 distinct rankings. This means that, it will be extremely unlikely to find more than 3 users with the very same preferences. On the other hand, because the $U_\pi = 98\%$ is close to the $E(U_\pi) = 99\%$, we should also not expect very strong biases in the ranking behaviors. For these reasons, we expect that it will be harder to find complete ranking patterns in this dataset.

Looking into the $E(U_\pi)$ of the two german elections datasets, their U_π is considerably less than its expected value. This seems to indicate that, not all rankings have equal probability in this election scenario. However, because we know that in elections it is very unusual that all parties have equal chances of being in all positions, across different regions, it makes sense.

5.3 Results

In this section we show some of the most interesting results obtained with the different quality measures.

5.3.1 Study on the behavior and biases of the Quality Measures

With each of the introduced quality measures, one can find subgroups featuring exceptional ranking behavior. The exceptionality is measured in (sometimes subtly) different ways for the different quality measures; which quality measure one uses depends on what type of exceptional ranking one is looking for. In this section, we briefly explore the differences in focus between the quality measures, to enable the user to make an informed choice.

In order to explore the relations of the bias between quality measures, we generated 10,000 random subgroups and their scores were measured by all quality measures. The generation was performed by randomly combining descriptions until the maximum depth is reached. The search depth was fixed as 3, to allow some diversity of combinations. The final result is presented in Figure 2, where the blue dots represent the significant subgroups according to the DFD test [24]. In Figure 2, the first row highlights the significant subgroups of $RWNorm$ and the vertical axis represents its score. The horizontal axis represents the scores of each quality measure, in the following order: $RWNorm$, $RWNorm-Mode$, $RWCov$, $LWNorm$ and

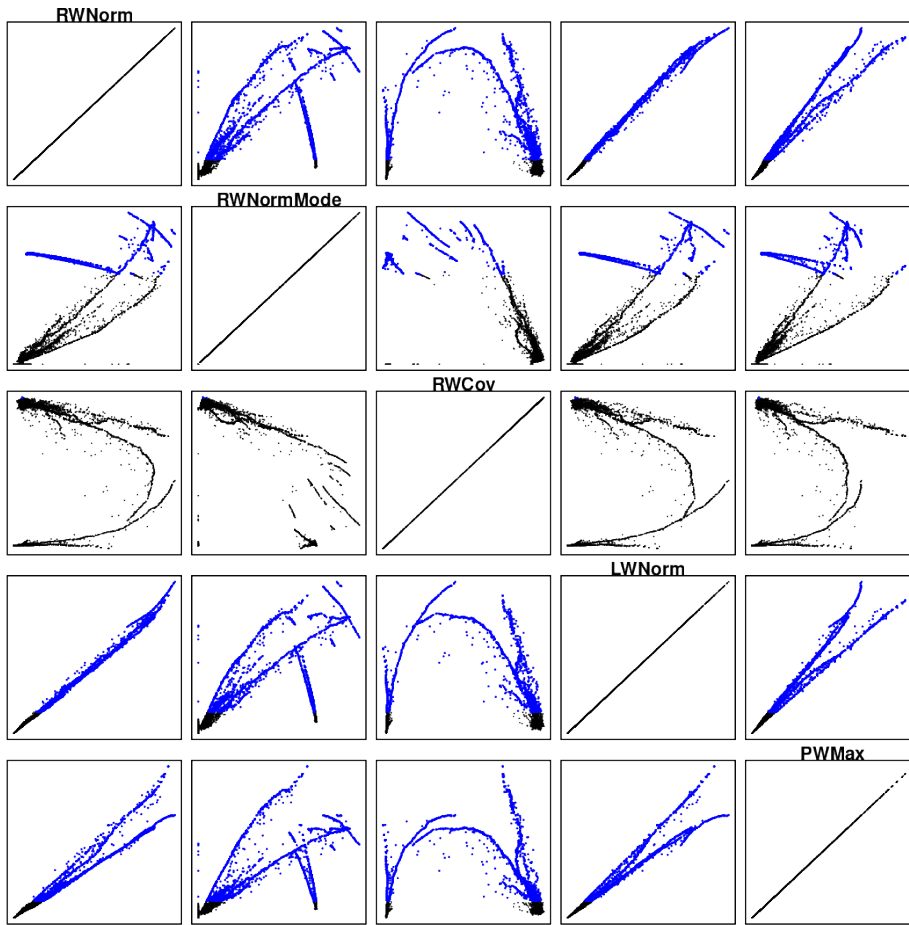


Fig. 2 Comparison of the scores of the quality measures on random subgroups obtained on the Cpu-Small dataset. The blue dots represent significant subgroups according to the DFD test [24].

PWMax. The second row highlights the significant subgroups of *RWNorm-Mode*, and so on.

As expected, some quality measures have a different but congruent bias. We can observe that 3 measures have a very similar bias, *RWNorm*, *LWNorm* and *PWMax*. This is somewhat expected, since they basically have the same measure, but applied in different parts of the distance matrix L_S .

The *RWNorm-Mode* shows a distinct behavior from the latter group. This measure is based on a different distance matrix L_S , obtained from the difference between the modes of the population M_D and the modes of the subgroups M_S . Its behavior can be explained with a simple example. For simplicity, let us consider only one entry of L_S , if one assumes that 51% of the subjects of a population agree that $\lambda_a \succ \lambda_b$, then, a reasonable sized subgroup where 51% agree that $\lambda_b \succ \lambda_a$ and the remaining 49% agree that $\lambda_a \succ \lambda_b$, will have a very high score with this measure. In fact, in this subgroup, only 2% less of the subjects prefer $\lambda_a \succ \lambda_b$, if

compared to the population. For the measures $RWNorm$, $LWNorm$ and $PWMax$, subgroups of this type will not be very interesting, unless that difference is bigger. This explains the behavior of the line on the top-left, observed on the second row of Figure 2, where $RWNorm-Mode$ compares to $RWNorm$, $LWNorm$ and $PWMax$. The rest of the behavior seems to be in line with the other measures.

Finally, $RWCov$, seems to have the most different bias. That is because it is not based in the distance matrix L_S . It measures the negative correlation directly between the population M_D and the subgroups M_S . Therefore, with this quality measure, we will find subgroups that do not necessarily maximize *preference distance*, but instead feature unusual preference *behavior* in a more abstract sense.

Figure 2 presents the significant subgroups per quality measure in blue. Despite that it seems there are a lot of significant subgroups, most of them have a near zero score. As a matter of fact, at most 30% of the random subgroups were considered significant. The total number of significant subgroups found, out of the 10,000 random subgroups, Table 3.

Table 3 Total number of significant subgroups found per dataset, with depth 3, using a random combinations of descriptions.

| Datasets | #Subgroups |
|-------------|------------|
| RWNorm | 2,977 |
| RWNorm-Mode | 1,128 |
| RWCov | 3 |
| LWNorm | 2,639 |
| PWMax | 2,746 |

We should note that, despite this random generation of the subgroups, the subgroups presented in Figure 2 are not totally random. Because the generation was performed by *randomly combining* descriptions, these "random subgroups" are bounded by the numeric strategy we chose and the nominal descriptors. For example, when $Sex = Male$ with $City = Tokyo$ are combined, even though the combination is random the two descriptions $Sex = Male$ and $City = Tokyo$ are not random subpopulations of the dataset. This is why it is possible to observe such a high number of interesting subgroups Table 3.

Now, let us focus on the number of subgroups obtained per measure, in terms of the given datasets in Table 4. Using a *best-first* search to find subgroups, we compare the number of subgroups obtained, per quality measure per dataset. For simplicity, we use a search depth of 1. $RWCov$ is, by far, the measure that identifies the least number of subgroups throughout measures and datasets. This seems to indicate that this measure is very restrictive, as expected (Section 4.4).

5.3.2 German Elections

With the *GermanElections2005* dataset, using the $PWMax$ with a search depth of 1, we found 62 significant subgroups. The best subgroup, $Region = East$, indicates that the party with label e in comparison to the party with label c has a very different behavior from the majority. In fact, while on 75% of the districts in Germany the FDP party (label c) was more voted than the LEFT party (label e),

Table 4 Total number of significant subgroups found per dataset, with depth 1, using the different quality measures.

| Datasets | RWNorm | RWNorm-Mode | RWCov | LWNorm | PWMax |
|---------------------|--------|-------------|-------|--------|-------|
| GermanElections2005 | 59 | 19 | 0 | 59 | 62 |
| GermanElections2009 | 55 | 18 | 1 | 53 | 59 |
| Top7movies | 2 | 0 | 0 | 2 | 2 |
| Algae | 22 | 5 | 1 | 22 | 21 |
| Sushi | 25 | 5 | 0 | 18 | 20 |
| Cpu-small | 12 | 10 | 6 | 12 | 12 |

on the 2005 elections, all the 87 districts from East Germany voted more on the LEFT party than on the FDP party. This shows a great example of an extreme inversion of preferences.

The second best subgroup obtained, compares the center-left GREEN party (label d) with the left-wing LEFT party (label e). The GREEN party had more votes than the LEFT party on 72% of the districts in Germany. On the other hand, on 88% of the districts where the average income is less or equal than 16,979, the LEFT party was more voted than the GREEN party.

To compare with the German elections of 2009, we used the *GermanElections2009* dataset with the same settings and found 57 significant subgroups. As in the 2005 elections, the best subgroup shows that 100% of the districts in east Germany gave more votes to the LEFT party than on the GREEN party, in comparison to only 27% in the whole Germany. The second best subgroup, as in the 2005 case, compares the center-left GREEN party (label d) with the left-wing LEFT party (label e). However, in this case, 94% of the districts, where the average income is less or equal than 16,979, the LEFT party was in advantage in comparison to the GREEN party. Comparing to the 88% of 2005, we realize that, in 2009, 6 p.p. more districts, where the average income was $\leq 16,979$, increased the votes in the LEFT party, in comparison to the GREEN party.

Continuing with the *GermanElections2009* and using the *LWNorm* with a search depth of 2, we found 2965 significant subgroups. The most relevant is expressed with a simple condition $\text{Region} = \text{East}$. This subgroup is interesting because it shows that, in most regions of East Germany, the *LEFT* party is often one of the top voted parties. In Figure 3 we can clearly see the distribution of the ranks. We observed that, the LEFT party was either first or second in the elections of 2009 in 97% of the districts in East Germany. Moreover, it was 3rd place in 3% of them. Other subgroups encountered show a very similar behavior in terms of the label that represents the LEFT party, like:

- Children Population $\leq 14.8\% \wedge$ Income $\leq 16,634$
- Children Population $\leq 14.8\% \wedge$ Unemployment $\geq 8.4\%$

On the other hand, we also found subgroups where the LEFT party is often the least voted party. Some examples are:

- Income $\geq 18,442$
- Income $\geq 17,791 \wedge$ Youth unemployment $\leq 8.5\%$

In Figure 4 we can visualize the distribution of Income ≥ 18442 .

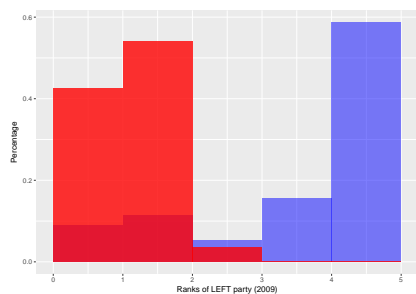


Fig. 3 Histograms representing the relative position of the LEFT party obtained in the 2009 elections of districts in Germany. In red, the subgroup Region = East and in blue the distribution for all districts.

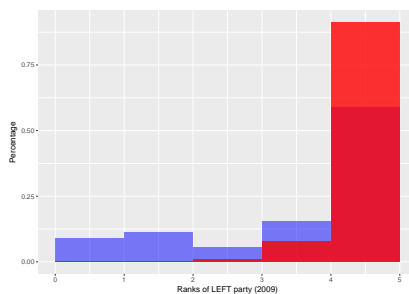


Fig. 4 Histograms representing the relative position of the LEFT party obtained in the 2009 elections of districts in Germany. In red, the subgroup Income ≥ 18442 and in blue the distribution for all districts.

Finally, in Figure 5 we can visualize the PM of subgroups which are described by the name of the state. This visualization clearly shows some nuances in the voting behavior on the different states of Germany.



Fig. 5 PM representation of some subgroups described by the feature *State* in comparison to the base matrix (All districts). The subgroups are sorted by relevance (first row, first column: most relevant; second row, second column: least relevant)

From a different perspective, if we look at the average rankings of each PM from Figure 5 we obtain:

- *CDU* \succ ***LEFT*** \succ *SPD* \succ *FDP* \succ *GREEN* (Thuringia)
- ***LEFT*** \succ ***SPD*** \succ *CDU* \succ *FDP* \succ *GREEN* (Brandenburg)
- ***LEFT*** \succ *CDU* \succ *SPD* \succ *FDP* \succ *GREEN* (Saxony-Anhalt)
- *CDU* \succ ***LEFT*** \succ *SPD* \succ *FDP* \succ *GREEN* (Saxony)
- *CDU* \succ *SPD* \succ *FDP* \succ ***GREEN*** \succ *LEFT* (Bavaria)
- *CDU* \succ *SPD* \succ *FDP* \succ *LEFT* \succ *GREEN* (All states)

We highlight (in bold) the parties which got a better relative position in the corresponding state, in comparison to the overall average ranking. As one can

conclude from most of the rankings in this list, at least one party (one label), seems to have its position changed relatively to the others. This clearly shows that the method is working as expected.

This analysis, also shows the potential of EPM as a tool to study election data. By looking at different levels of granularity of the preferences, EPM does not necessarily focus on the winners, but rather on major preference shifts. Also, considering the elections application, different ranking aggregation metrics can be used to comply with the Condorcet method [15].

5.3.3 Top7Movies

With the *LWNorm* quality measure, we found 2 significant subgroups for a search depth of 2. The members of the first subgroup, people older than 34 years old living below a latitude of 32.9, seem to dislike the most voted movie *American Beauty*, more than usual (Figure 6). This subgroup, includes people from different states, such as Arizona, California, Florida, Georgia, Louisiana, New Mexico, Texas and even Hawaii. An interesting conclusion we can draw, is that, this group voted in *Star Wars: Episode IV - A New Hope* and *Saving Private Ryan* with high scores. On

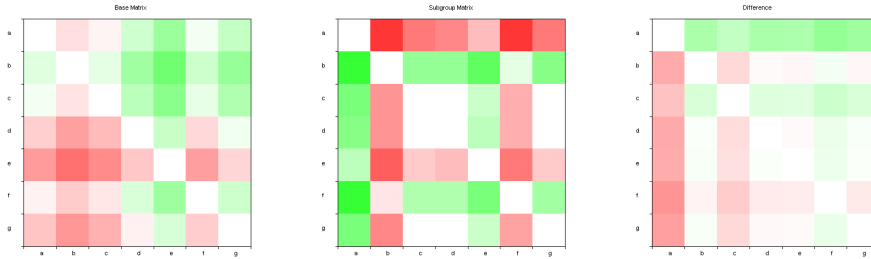


Fig. 6 PM representation of the dataset Top7Movies (base matrix), the subgroup $Age \geq 35 \wedge Latitude \leq 32.9$ (subgroup matrix) and the difference (difference matrix).

the other hand they seem to dislike *American Beauty* and *Jurassic Park*. In fact, the average ranking of this subgroup is $b \succ f \succ c \succ d \succ g \succ a \succ e$ and the average ranking of the whole population is $b \succ c \succ a \succ f \succ d \succ g \succ e$.

5.3.4 Algae

With the Algae dataset, we obtain results about the concentrations of algae with the *RWNorm* measure. Results seem to indicate that during *Spring*, the species of algae *a*, *b* and *c* are much more common in rivers than the others species. This can be easily concluded by studying the PM representation of the subgroup (Figure 7). On the other hand, we also see an interesting behavior during the *Autumn* season.

With the *LWNorm* measure, we find a bit more than 400 subgroups with maximum depth 2, the best of which is presented in Figure 8. In the subgroup, the label *a* is strongly preferred over all others, while the image is much more nuanced over the whole dataset. If we ignore the label *a*, the PMs for both the overall dataset

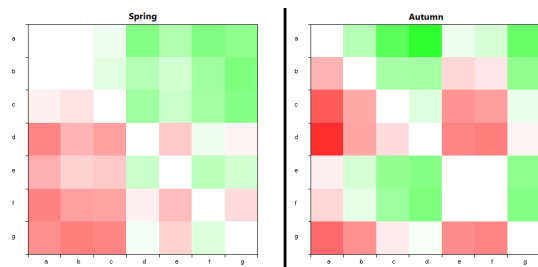


Fig. 7 PM representation of the subgroups $Season = Spring$ (left subgroup matrix) and $Season = Autumn$ (right subgroup matrix) from the Algae dataset.

and the subgroup are rather bland, and their difference is not very pronounced. But for this one particular label a , the behavior on the subgroup is extremely clear-cut, and the $LWNorm$ quality measure picks up on that effect.

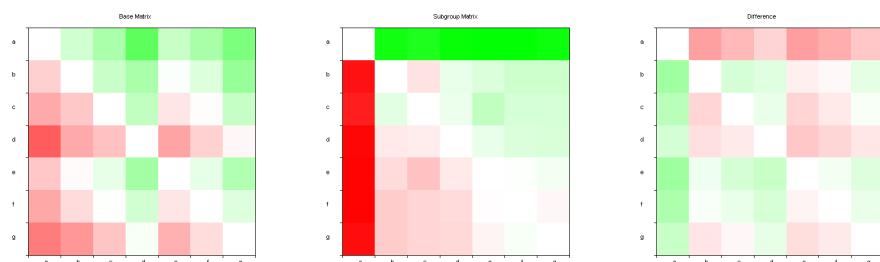


Fig. 8 PM representation of the dataset Algae (base matrix) and the subgroup $V10 \leq 59 \wedge V6 \leq 11.87$ (subgroup matrix), with difference matrix on the right.

Using a depth of 3 with the same measure, we found around 5,400 subgroups. We show the best one in Figure 9. One interesting aspect of this subgroup is that it shows an opposite behavior, in comparison to the one in Figure 8, in terms of the label a (as it is clear from the difference matrix).

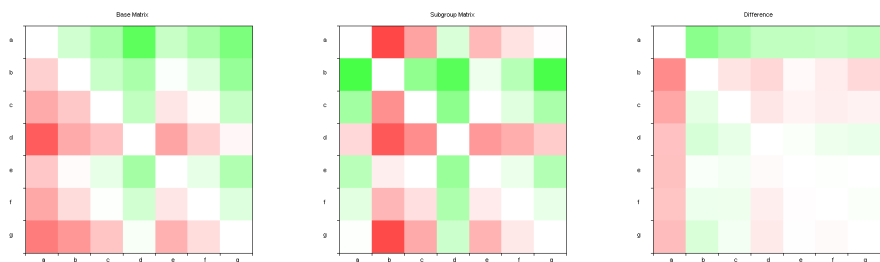


Fig. 9 PM representation of the dataset Algae (base matrix) and the subgroup $V10 \geq 137.78 \wedge V6 \geq 14.32 \wedge V9 \geq 60.83$ (subgroup matrix), with difference matrix on the right.

The visual representations of the PM clearly reveal the effect of the $LWNorm$ quality measure in this dataset. We can also observe from the description of the

subgroups obtained, that the variables $V10$ and $V6$ are highly correlated with the presence of the algae a .

5.3.5 Sushi

Considering the high percentage of unique rankings in the sushi dataset (Table 2) we do not expect to find strong patterns in the whole PM, therefore, we focus on labelwise ranking patterns.

With the $LWNorm$ measure, we find 149 subgroups on the Sushi dataset. We present the best subgroup using this measure in Figure 10. The subgroup (Males over 30 years) shows a preference for Sea Urchin, since the majority of men rank this sushi type in the top 4. By contrast, in the whole population, more than half rate it between 5th to 10th, and every fifth person rate it in the last place.

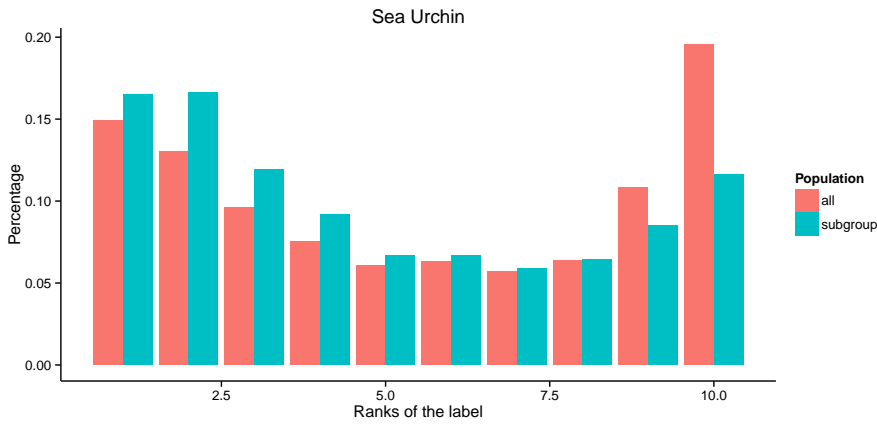


Fig. 10 Percentage of ranks for Sea Urchin (Sushi dataset) for all individuals in comparison to the subgroup (males older than 30 years).

5.3.6 Cpu-small

On the Cpu-small dataset, we used the $RWCov$ quality measure. Experiments with a maximum depth of 4, found 275 significant subgroups. In Figure 11 we can visualize the PM of the most relevant subgroup found. The PM of this subgroup, of size 62, shows deviations in all the entries of the matrix, which is a good indicator that this measure is working as expected.

In terms of the rankings, the average ranking of the whole dataset is (2, 4, 3, 1, 5), and the average ranking in this subgroup is (3, 1, 5, 4, 2). The Kendall τ correlation of these two rankings is -0.4 , which confirms the unusualness of the subgroup.

We could also observe that, despite having obtained 275 significant subgroups, there were many subgroups whose PM was very similar and showing the same unusual behavior. This could also be observed in terms of the ranking derived from their PM.

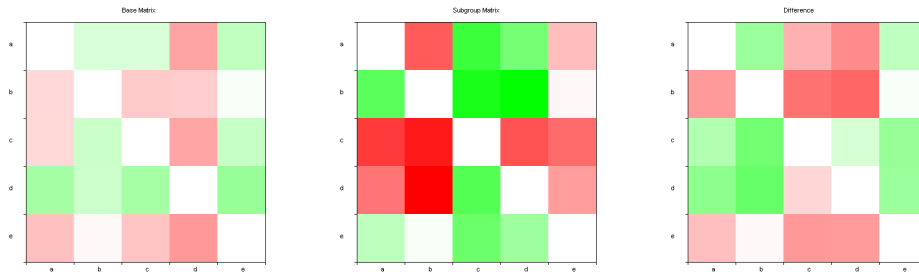


Fig. 11 PM representation of the dataset Cpu-small (base matrix), the subgroup $A5 \geq 0.710 \wedge A6 \geq 2.143 \wedge A3 \leq 0.755$ (Subgroup Matrix) and the difference (difference matrix).

5.3.7 Comparison of different aggregation metrics

As mentioned in Section 4.1, different metrics can be used in the aggregation of PM. To test how this choice can affect the model, we analyzed some results where PMs are aggregated with the mode (instead of the mean), however, for the sake of space, we only present one dataset and one quality measure, *RWNorm-Mode*.

Using the mode as the aggregation, *RWNorm-Mode* quality measure, we found 131 significant subgroups of depth 2 on the Cpu-small dataset. As a point of comparison, we obtained 155 significant subgroups, with the same settings, using the *RWNorm* quality measure (aggregation with the mean). Despite the similar number of subgroups found, the two groups of subgroups are quite distinct. This is somehow expected from the previous analysis of the quality measures in Section 5.3.1.

A striking difference is that the rankings of the subgroups from *RWNorm-Mode* are consistently different from the ones obtained with *RWNorm*. However, despite being different, the average rankings of the subgroups have a similar correlation (in terms of the Kendall τ) to the average ranking of the population.⁶ In other words, the subgroups are at a similar "preference distance" from the population. This seems to indicate that *RWNorm-Mode* can be a complementary measure with *RWNorm*.

The behavior described above, is also observed on the remaining datasets presented in Table 2. For the sake of space, let us consider the best subgroup, according to *RWNorm-Mode*, depicted in Figure 12. This subgroup is described by: $A4 \geq -0.22354$ In Figure 12 we can observe that the difference matrix of the best subgroup has very faint colored tiles, which means that the PM is not very different from the PM of the whole dataset. On the other hand, these small differences are quite spread along the difference matrix, which, when summed up, makes it interesting too.

From a different perspective, in Figure 13 we compare the distributions of the correlation between the average ranking of the dataset and each one of the rankings that are part of the best subgroup. We measure this correlation in terms of the Kendall τ correlation coefficient. As seen in Figure 13, the distributions are similar. This behavior was also observed in other subgroups and other datasets. Therefore,

⁶ We note that two distinct rankings can have the same Kendall τ correlation with a third ranking.

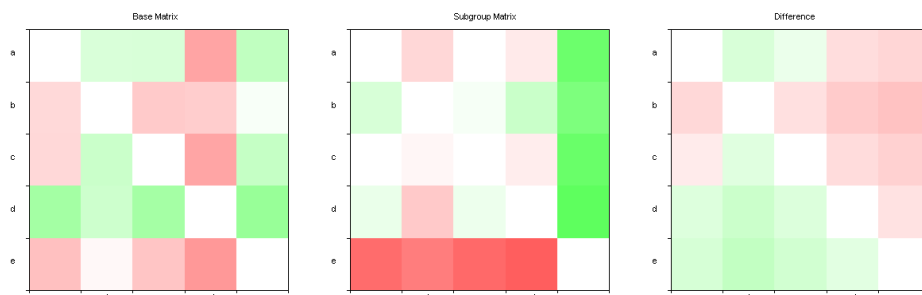


Fig. 12 Representation of the PMs, aggregated with the mode, of the dataset Cpu-small (base matrix), the subgroup $A4 \geq -0.22354$ (subgroup matrix) and the difference (difference matrix).

this confirms what we observed above, that *RWNorm-Mode* and *RWNorm* find different subgroups but with similar "preference distances".

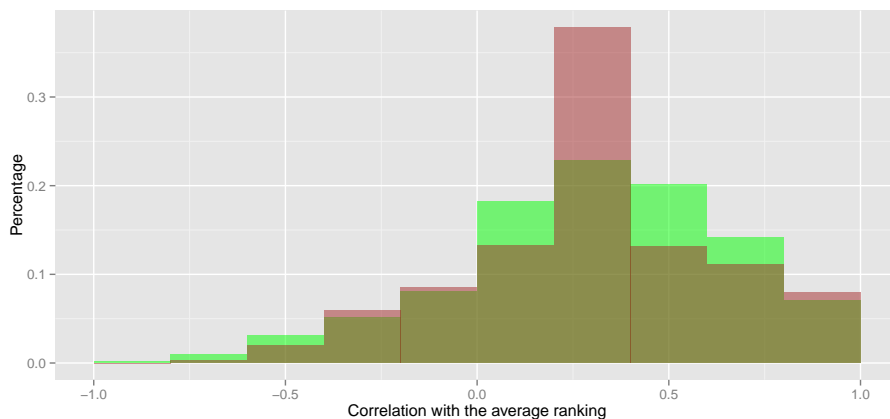


Fig. 13 Distributions of the correlation between the average ranking and each ranking belonging to the best subgroup found with *RWNorm-Mode* (green) and *RWNorm* (brown).

Aggregating a PM with the mode can yield either 1, 0 or -1 in contrast to the mean where any value in the interval $[-1, 1]$ is possible. Therefore, the mean can measure exceptionality on subgroups with the same mode as the dataset (e.g. label *a* in Figure 8). On the other hand, the mode can detect subgroups where the majority of the pairs behave differently. Therefore, depending on the task, the best choice of the aggregation metric for the quality measures can change. However, we believe that the best way is to complement the use of *RWNorm-Mode* with *RWNorm* and vice versa.

5.4 Comparison with Distribution Rules

In this section, we compare subgroups found with our algorithm (using CORTANA) with subgroups from a different approach, Distribution Rules (DR) (using CAREN [3] software⁷). As mentioned before (Section 3.2), Distribution Rules are a SD method that looks for unusual target distributions [35, 43]. CORTANA and CAREN can be used for mining other structures of data. For simplicity, in this work we refer to CORTANA and CAREN as the tools with our preference learning approaches.

DR use a numeric target to construct the distributions. Since we have rankings as targets, we propose a simple way to represent individual rankings as numeric values. For each example we compute the similarity score between its ranking and the *average ranking* (consensus ranking [6]) of the dataset. Given that, the similarity measure that we use is the Kendall τ , the new target can have values in the range $[-1, 1]$.

We show in Table 5 how the example dataset \hat{D} would look like under this transformation. Considering that the average ranking of the rankings in \hat{D} is: $(2, 3, 1, 4)$, for the second example in \hat{D} , we do: $\tau((2, 3, 1, 4), (3, 2, 1, 4)) = 0.66$.

Table 5 Example dataset \hat{D} with the proposed alternative representation in the rightmost column of the table.

| \mathcal{A}_1 | π | | | | similarity to average ranking |
|-----------------|-------------|-------------|-------------|-------------|----------------------------------|
| | λ_1 | λ_2 | λ_3 | λ_4 | |
| 0.1 | 4 | 3 | 1 | 2 | 0 |
| 0.2 | 3 | 2 | 1 | 4 | 0.66 |
| 0.3 | 1 | 4 | 2 | 3 | 0.33 |
| 0.4 | 1 | 3 | 2 | 4 | 0.66 |

For a fair comparison between the two methods, we discretized the numeric attributes beforehand with an *equal width* discretization of 8 bins. We handle the discretized numerical attributes as a nominal, not ordinal, scale. In terms of the property of interest (target), this numerical variable does not have to be previously discretized, because the method works with raw distributions [43].

In terms of the experimental setup, we will use the same maximum search depth for both methods. In CORTANA, we take the *RWNorm* quality measure. For each subgroup, we perform a Kolmogorov-Smirnov statistical test to compare the target distribution of the subgroup with the target distribution of the whole population. Subgroups which are deemed interesting, are the ones whose distributions differ significantly from the distribution of the whole population.

We will use the term subgroup and distribution rules interchangeably to refer to distribution rules. However, when there is the need to differentiate from subgroups found with CORTANA and CAREN, we will use the terms subgroups and distribution rules, respectively.

⁷ <http://www4.di.uminho.pt/~pja/class/caren.html>

5.4.1 German Elections

With the *GermanElections2009* dataset, we found 1,597 significant distribution rules using CAREN and 1,073 subgroups with CORTANA for a search depth of 2. The most interesting distribution rules are not only in line with the subgroups found, in this experiment, but also with the ones previously discussed in Section 5.3.2. For the sake of simplicity, we only show the top five subgroups obtained by both approaches in Table 6. It is clear from Table 6 that the subgroups found

Table 6 Comparison of subgroups found by CAREN and CORTANA

| CAREN | CORTANA |
|---|---|
| Region = East | Region = East |
| Region = East \wedge Type = Rural | Region = East \wedge Reg.Web.Dom. = a |
| Region = East \wedge Reg.Web.Dom. = a | Income = a \wedge Region = East |
| Income = a \wedge Region = East | Region = East \wedge Type = Rural |
| Income = a | Income = a |

by CAREN are very similar from the subgroups of CORTANA, despite their very distinct approaches.

The distribution of the most interesting subgroup, Region = East, is represented in Figure 14. We can observe that, the majority of the rankings in the whole dataset have a similarity of 0.8 with the average ranking. On the other hand, the rankings of this subgroup, have at most a similarity of 0.7.

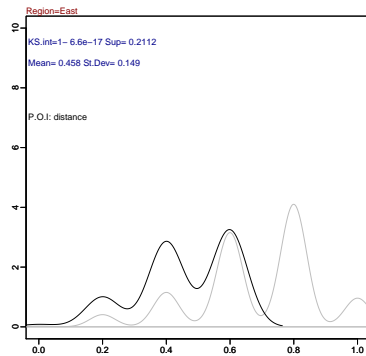


Fig. 14 Graphical representation of the distributions of the target of the subgroup Region = East (in bold) in comparison to the whole target distribution in *GermanElections2009*.

5.4.2 Top7Movies

In this section, we analyze a set of DR found with the *Top7Movies* dataset and compare to the subgroups obtained with CORTANA. We found 7 significant DR with CAREN and a search depth of 2. In Figure 15 we can see the description and the distributions of the DR found on the *Top7Movies* dataset.

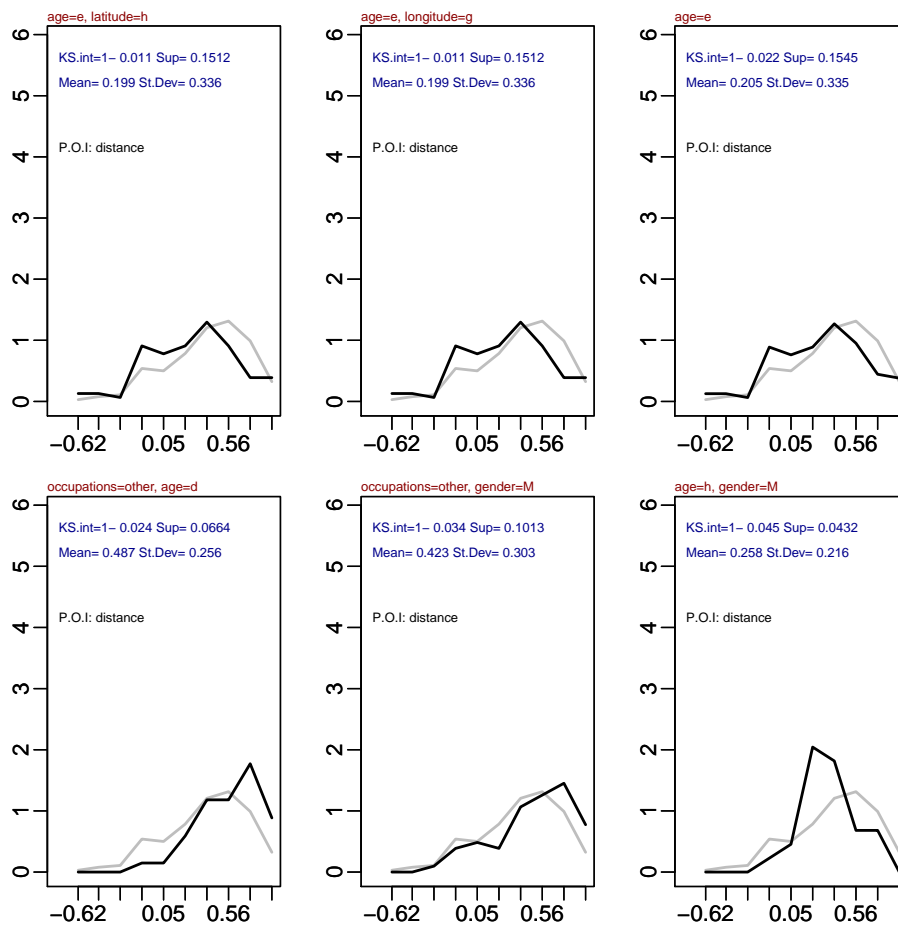


Fig. 15 Graphical representation of the distributions rules found in *Top7Movies* dataset.

With CORTANA, we found 7 significant subgroups with a search depth of 2. From this set, 3 subgroups are the same (but in a different order), as we can see from Table 7.

Table 7 Comparison of subgroups found by CAREN and CORTANA

| CAREN | CORTANA |
|--|------------------------------------|
| Age = 35-44 \wedge Latitude = h | Age = 35-44 \wedge Gender = Male |
| Age = 35-44 \wedge longitude = g | Age = 35-44 |
| Age = 35-44 | Age = 35-44 \wedge Latitude = h |
| Occupations = Other \wedge Age = 25-34 | Age = 35-44 \wedge Longitude = g |
| Occupations = Other \wedge Gender = Male | Age = 18-24 |
| Age = 50+ \wedge Gender = Male | Age = 18-24 \wedge Latitude = h |
| Occupations = Other | Age = 18-24 \wedge Longitude = g |

We note that, in the Label Ranking context, despite the similarities between the subgroups found both by CAREN and CORTANA, the interpretation of the rankings is richer with a PM than with a distribution. PM are better for spotting slight nuances in the preference patterns, for example, when a particular label is under- or overappreciated. Moreover, if we want to search for partial ranking patterns such as labels or simply label-vs-label, it is simpler to visualize and handle it with a PM. This means that, EPM, due to its representation of rankings, has a bigger margin for the creation of new quality measures.

6 Conclusions

In this work, we empirically show how Exceptional Preferences Mining (EPM) can be used in problems where the target concept can be represented as a ranking of a fixed set of labels. The results are a set of subgroups, that can be described in terms of a conjunction of few conditions on some attributes, where the label preferences are exceptional in some sense. The presented subgroups form clear coherent parts of the search space, which means that EPM finds deviating preferences that are actionable for domain experts, since their description in terms of attributes should be familiar to them.

All subgroups whose PM deviates significantly from the Preference Matrix (PM) for the whole dataset are considered to be interesting. We used four quality measures for EPM that instantiate this concept of ‘interesting’ to different levels, Rankingwise, Labelwise and Pairwise. The *RWNorm*, *RWNorm-Mode* and *RWCov* quality measures consider a subgroup interesting if the full set of preference relations is substantially displaced. The *LWNorm* quality measure highlights subgroups where any one label interacts exceptionally with the other labels, agnostic of how those other labels interact with each other. The *PWMax* quality measure finds a subgroup interesting if any one pair of labels display exceptional preference relations. Hence, by choosing the appropriate quality measure, EPM delivers subgroups featuring preference relations that are exceptional at your preferred scope.

To show the potential of the approach, we provided experiments on several datasets. The experiments with the *RWNorm* quality measure on the Algae dataset revealed several interesting conditions that can affect the populations of the different species of algae from rivers. The experiments with the *LWNorm* quality measure on the Sushi dataset illustrate the relative merit of this quality measure: it focuses on subgroups where one particular label is exceptionally under- or overappreciated. The subgroup presented has a penchant for Sea Urchin (cf. Figure 10). The *PWMax* measure shows its potential on the *German2005elections* dataset by identifying several subgroup with strong exceptional preferences with respect to the different parties. The experiments with the *RWCov* quality measure on the Cpu-small dataset (e.g. Figure 11) reveal a subgroup with quite unusual preference behavior. Finally, the *RWNorm-Mode* was compared to the *RWNorm* measure, in different experiments, and we could observe that it revealed some interesting subgroups too. Moreover, we concluded that *RWNorm-Mode* and *RWNorm* can be complementary measures to study exceptional preference patterns.

As we argued in Section 3, one of the main benefits of a local pattern mining method such as EPM is that it delivers interpretable results. That means that the

resulting subgroups are ideally suited to instigate real-world policies and actions. For this reason, we studied several real-world datasets.

We also compared the results found with EPM with an alternative approach, the Distribution Rules (DR). Despite their very different setting, the subgroups found by this method were very similar to the ones found with CORTANA. In our opinion, this simple comparison empirically shows that our suggested quality measures for EPM are finding relevant patterns. In terms of interpretation, PM are better than distribution rules to detect slight nuances in the preference patterns, for example, when a particular label is under- or overappreciated. In some cases, information which is not easy to obtain with the usual representations of rankings, is clearly revealed through the PM visualization (see Section 5.3.2).

From this study, we also understand some limitations of our approach. We observed that, in some cases, despite having obtained many significant subgroups, most of them are specializations of simpler subgroups with very similar average rankings, if not equal. This means that, many different subgroups are finding the same ranking behaviors.

EPM also has the disadvantage to be time consuming. A large number of labels combined with a still reasonably high search depth makes the statistical tests very time consuming.

As future work we would like to study alternative ways to represent and look for patterns in rankings, for example for rankings with a large number of labels as well as for partial orders. Finally, we would also like to study how pruning techniques such as *minimum improvement* can be used to filter out subgroups, that are specializations of simpler subgroups, but have very similar PMs.

Acknowledgments

This research has received funding from the ECSEL Joint Undertaking, the framework programme for research and innovation Horizon 2020 (2014-2020) under grant agreement number 662189-MANTIS-2014-1.

References

1. T. Abudawood and P. A. Flach. Evaluation measures for multi-class subgroup discovery. In *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2009, Bled, Slovenia, September 7-11, 2009, Proceedings, Part I*, pages 35–50, 2009.
2. R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo. Fast discovery of association rules. In *Advances in Knowledge Discovery and Data Mining*, pages 307–328. AAAI/MIT Press, 1996.
3. P. J. Azevedo and A. M. Jorge. Ensembles of jittered association rule classifiers. *Data Min. Knowl. Discov.*, 21(1):91–129, 2010.
4. M. Boley, M. Mampaey, B. Kang, P. Tokmakov, and S. Wrobel. One click mining: interactive local pattern discovery through implicit preference and performance learning. In *Proceedings of the ACM SIGKDD Workshop on Interactive Data Exploration and Analytics, IDEA@KDD 2013, Chicago, Illinois, USA, August 11, 2013*, pages 27–35, 2013.
5. F. Brandenburg, A. Gleißner, and A. Hofmeier. Comparing and aggregating partial orders with kendall tau distances. *Discrete Math., Alg. and Appl.*, 5(2), 2013.
6. P. Brazdil, C. Soares, and J. P. da Costa. Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results. *Machine Learning*, 50(3):251–277, 2003.
7. J. Breen. *zipcode: U.S. ZIP Code database for geocoding*, 2012. R package version 1.0.

8. K. Brinker and E. Hüllermeier. Label ranking in case-based reasoning. In *Case-Based Reasoning Research and Development, 7th International Conference on Case-Based Reasoning, ICCBR 2007, Belfast, Northern Ireland, UK, August 13-16, 2007, Proceedings*, pages 77–91, 2007.
9. V. Chankong and Y. Haimes. *Multiobjective Decision Making: Theory and Methodology*. Dover Books on Engineering. Dover Publications, 2008.
10. W. Cheng, K. Dembczynski, and E. Hüllermeier. Label ranking methods based on the plackett-luce model. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 215–222, 2010.
11. W. Cheng, S. Henzgen, and E. Hüllermeier. Labelwise versus pairwise decomposition in label ranking. In *LWA 2013. Lernen, Wissen & Adaptivität, Workshop Proceedings Bamberg, 7.-9. October 2013*, pages 129–136, 2013.
12. W. Cheng, J. C. Huhn, and E. Hüllermeier. Decision tree and instance-based learning for label ranking. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, pages 161–168, 2009.
13. W. Cheng, M. Rademaker, B. D. Baets, and E. Hüllermeier. Predicting partial orders: Ranking with abstention. In *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part I*, pages 215–230, 2010.
14. J. Chomicki. Preference formulas in relational queries. *ACM Trans. Database Syst.*, 28(4):427–466, 2003.
15. M. de Condorcet. Éssai sur l’application l’analyse à la probabilité des dés décisions rendues à la pluralité des voix (trans. essay on the application of mathematics to the theory of decision-making), 1785.
16. C. R. de Sá, W. Duivesteijn, C. Soares, and A. Knobbe. Exceptional preferences mining. In *Discovery Science*, page 1–16, 2016.
17. C. R. de Sá, C. Soares, A. M. Jorge, P. J. Azevedo, and A. Knobbe. Preference rules. *submitted to Information Fusion Journal*, 2017.
18. C. R. de Sá, C. Soares, and A. J. Knobbe. Entropy-based discretization methods for ranking data. *Inf. Sci.*, 329:921–936, 2016.
19. O. Dekel, C. D. Manning, and Y. Singer. Log-linear models for label ranking. In *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada]*, pages 497–504, 2003.
20. K. Dembczynski, W. Kotlowski, R. Slowinski, and M. Szelag. Learning of rule ensembles for multiple attribute ranking problems. In *Preference Learning.*, pages 217–247. Springer, 2010.
21. W. Duivesteijn. *Exceptional Model Mining*. PhD thesis, Leiden University, 2013.
22. W. Duivesteijn, A. Feelders, and A. J. Knobbe. Different slopes for different folks: mining for exceptional regression models with cook’s distance. In *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’12, Beijing, China, August 12-16, 2012*, pages 868–876, 2012.
23. W. Duivesteijn, A. Feelders, and A. J. Knobbe. Exceptional model mining - supervised descriptive local pattern mining with complex target concepts. *Data Min. Knowl. Discov.*, 30(1):47–98, 2016.
24. W. Duivesteijn and A. J. Knobbe. Exploiting false discoveries - statistical validation of patterns and quality measures in subgroup discovery. In *11th IEEE International Conference on Data Mining, ICDM 2011, Vancouver, BC, Canada, December 11-14, 2011*, pages 151–160, 2011.
25. V. Dzyuba and M. van Leeuwen. Interactive discovery of interesting subgroup sets. In *Advances in Intelligent Data Analysis XII - 12th International Symposium, IDA 2013, London, UK, October 17-19, 2013. Proceedings*, pages 150–161, 2013.
26. J. Fürnkranz and E. Hüllermeier. Pairwise preference learning and ranking. In *Machine Learning: ECML 2003, 14th European Conference on Machine Learning, Cavtat-Dubrovnik, Croatia, September 22-26, 2003, Proceedings*, pages 145–156, 2003.
27. J. Fürnkranz and E. Hüllermeier, editors. *Preference Learning*. Springer, 2010.
28. H. Grosskreutz, M. Boley, and M. Krause-Traudes. Subgroup discovery for election analysis: A case study in descriptive data mining. In *Discovery Science - 13th International Conference, DS 2010, Canberra, Australia, October 6-8, 2010. Proceedings*, pages 57–71, 2010.

29. S. Har-Peled, D. Roth, and D. Zimak. Constraint classification: A new approach to multi-class classification. In *Algorithmic Learning Theory, 13th International Conference, ALT 2002, Lübeck, Germany, November 24-26, 2002, Proceedings*, pages 365–379, 2002.
30. F. M. Harper and J. A. Konstan. The movielens datasets: History and context. *TüS*, 5(4):19:1–19:19, 2016.
31. S. Henzgen and E. Hüllermeier. Mining rank data. In *Discovery Science - 17th International Conference, DS 2014, Bled, Slovenia, October 8-10, 2014. Proceedings*, pages 123–134, 2014.
32. Y. Hochberg and A. Tamhane. *Multiple comparison procedures*. Wiley series in probability and mathematical statistics: Applied probability and statistics. Wiley, 1987.
33. E. Hüllermeier, J. Fürnkranz, W. Cheng, and K. Brinker. Label ranking by learning pairwise preferences. *Artif. Intell.*, 172(16-17):1897–1916, 2008.
34. N. Jin, P. A. Flach, T. Wilcox, R. Sellman, J. Thumim, and A. J. Knobbe. Subgroup discovery in smart electricity meter data. *IEEE Trans. Industrial Informatics*, 10(2):1327–1336, 2014.
35. A. M. Jorge, P. J. Azevedo, and F. Pereira. Distribution rules with numeric attributes of interest. In *Knowledge Discovery in Databases: PKDD 2006, 10th European Conference on Principles and Practice of Knowledge Discovery in Databases, Berlin, Germany, September 18-22, 2006, Proceedings*, pages 247–258, 2006.
36. A. M. Jorge, F. Pereira, and P. J. Azevedo. Visual interactive subgroup discovery with numerical properties of interest. In *Discovery Science, 9th International Conference, DS 2006, Barcelona, Spain, October 7-10, 2006, Proceedings*, pages 301–305, 2006.
37. T. Kamishima. Nantonac collaborative filtering: recommendation based on order responses. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 24 - 27, 2003*, pages 583–588, 2003.
38. W. Klösgen. Explora: A multipattern and multistrategy discovery assistant. In *Advances in Knowledge Discovery and Data Mining*, pages 249–271. American Association for Artificial Intelligence, 1996.
39. W. Klösgen and J. M. Zytkow, editors. *Handbook of Data Mining and Knowledge Discovery*. Oxford University Press, New York, NY, USA, 2002.
40. N. Lavrac, B. Kavsek, P. A. Flach, and L. Todorovski. Subgroup discovery with CN2-SD. *Journal of Machine Learning Research*, 5:153–188, 2004.
41. D. Leman, A. Feelders, and A. J. Knobbe. Exceptional model mining. In *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML/PKDD 2008, Antwerp, Belgium, September 15-19, 2008, Proceedings, Part II*, pages 1–16, 2008.
42. M. Lichman. UCI machine learning repository, 2013.
43. J. P. Lucas, A. M. Jorge, F. Pereira, A. M. Pernas, and A. A. Machado. A tool for interactive subgroup discovery using distribution rules. In *Progress in Artificial Intelligence, 13th Portuguese Conference on Artificial Intelligence, EPIA 2007, Workshops: GAIW, AIASTS, ALEA, AMITA, BAOSW, BI, CMBSB, IROBOT, MASTA, STCS, and TEMA, Guimarães, Portugal, December 3-7, 2007, Proceedings*, pages 426–436, 2007.
44. M. Meeng and A. Knobbe. Flexible enrichment with cortana—software demo. In *Proceedings of BeneLearn*, pages 117–119, 2011.
45. L. Todorovski, H. Blockeel, and S. Dzeroski. Ranking with predictive clustering trees. In *Machine Learning: ECML 2002, 13th European Conference on Machine Learning, Helsinki, Finland, August 19-23, 2002, Proceedings*, pages 444–455, 2002.
46. L. Umek and B. Zupan. Subgroup discovery in data sets with multi-dimensional responses. *Intell. Data Anal.*, 15(4):533–549, 2011.
47. T. L. Van, M. van Leeuwen, S. Nijssen, A. C. Fierro, K. Marchal, and L. D. Raedt. Ranked tiling. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part II*, pages 98–113, 2014.
48. S. Vembu and T. Gärtner. Label ranking algorithms: A survey. In *Preference Learning.*, pages 45–64. Springer, 2010.