Verbal Characterization of Probabilistic Clusters using Minimal Discriminative Propositions

Yoshitaka Kameya, Satoru Nakamura, Tatsuya Iwasaki* and Taisuke Sato

Graduate School of Information Science and Engineering, Tokyo Institute of Technology
2-12-1 Ookayama, Meguro-ku, Tokyo 152-8552, Japan

Abstract. In a knowledge discovery process, interpretation and evaluation of the mined results are indispensable in practice. In the case of data clustering, however, it is often difficult to see in what aspect each cluster has been formed. This paper proposes a method for automatic and objective characterization or "verbalization" of the clusters obtained by mixture models, in which we collect conjunctions of propositions (attribute-value pairs) that help us interpret or evaluate the clusters. The proposed method provides us with a new, in-depth and consistent tool for cluster interpretation/evaluation, and works for various types of datasets including continuous attributes and missing values. Experimental results with a couple of standard datasets exhibit the utility of the proposed method, and the importance of the feedbacks from the interpretation/evaluation step.

1 Introduction

In a knowledge discovery process, interpretation and evaluation of the mined results are indispensable in practice. In the case of data clustering [1], however, it is often difficult to see in what aspect each cluster has been formed, only from a list of the instances in the cluster. Visualization is a natural way for understanding things, and particularly in text clustering, Hotho et al. applied formal concept analysis with Hasse diagrams to visualize the similarity and dissimilarity among the obtained clusters [2]. On the other hand, since there would generally be a physical limitation or a high implementational cost in visualization, we would rather like to "verbalize" the clusters, i.e. we associate an intuitive descriptive label (or a set of such labels) with each cluster. Additionally it seems desirable that the labels are chosen objectively and automatically from the clusters. So far, there have been only a few labeling methods, e.g. LabelSOM [3], Mei et al.'s automatic labeling for topic models [4] and others [5,6]. CLIQUE [7] also has a similar motivation to ours in that it performs hyper-rectangular clustering and at the same time produces comprehensible descriptions of the obtained clusters.

In this paper, we propose a new labeling method that associates conjunctions of propositions (attribute-value pairs), called propositional labels, with the clusters obtained by mixture models. For example, consider a cluster $C$ which contains several creatures such as dolphins, mink, platypus and seals. Then, letting "milk" and "aquatic" be the boolean attributes of the creatures, $(\text{milk}=\text{True} \land \text{aquatic}=\text{True})$ would be a suitable propositional label for the cluster $C$, if none of the creatures in the other clusters has these properties together. Finally we easily find that $C$ is a cluster of aquatic mammals. To find these propositional labels objectively and automatically, we conduct an Apriori-style breadth-first search for minimal propositional labels that discriminate the cluster of interest from the others. Due to these features, as we will see later, the proposed method can provide us with a new, in-depth and consistent tool for cluster interpretation/evaluation. It is also notable that, unlike the previous attempts, the proposed method is fully applicable to various types of datasets including continuous attributes and missing values. Another novel contribution of this paper is to show empirically the importance of the feedbacks from the interpretation/evaluation step in achieving a reasonable clustering result.

The rest of this paper is structured as follows. In Section 2 we describe the details of the proposed method. Section 3 then reports the experimental results with a couple of standard datasets. Finally, we mention the related work in Section 4 and conclude the paper in Section 5.

2 Proposed method

2.1 Preliminaries

Before starting, let us introduce some terminology and notation. Suppose that we have a dataset $\mathcal{D}$ of $N$ instances which are described by $m$ discrete attributes $A_1, A_2, \ldots, A_m$. Then, we simply refer to each instance by

* Currently working at NTT Data Corporation
adaptation of those relevance scores to our labeling problem. We also write \( \mathcal{V}(A_j) \) as the set of possible values of \( A_j \) (i.e. \( a_j \in \mathcal{V}(A_j), 1 \leq j \leq m \)). We now introduce a propositional label (or a label, for short) \( "X_1 = x_1" \wedge \cdots \wedge "X_n = x_n" \) such that \( \{X_1, X_2, \ldots, X_n\} \subseteq \{A_1, A_2, \ldots, A_m\} \), \( X_i \) and \( X_j \) are distinct \( (i \neq j) \), and \( x_i \in \mathcal{V}(X_i) \). In a probabilistic context, \( p("X_1 = x_1" \wedge \cdots \wedge "X_n = x_n") = p(X_1 = x_1, \ldots, X_n = x_n) \) holds. Also, \( p(Z = z, \ldots) \) for a random discrete variable \( Z \) and its value \( z \) is generally abbreviated as \( p(z, \ldots) \) if the context is clear.

Furthermore, we add some notational conventions. First, without loss of generality, we assume that the attribute values are not overlapped among attributes (i.e. \( \mathcal{V}(A_j) \cap \mathcal{V}(A_j') = \emptyset \) for \( j \neq j' \)). Then, a propositional label \( "X_1 = x_1" \wedge \cdots \wedge "X_n = x_n" \) is unambiguously simplified as \( x = (x_1 \wedge \cdots \wedge x_n) \) or \( x = (x_1, \ldots, x_n) \). Here we have \( |x| = n \), where \( |x| \) denotes the number of conjuncts in \( x \), and is called the length of \( x \). An instance \( a = (a_1, \ldots, a_m) \) is also regarded as a propositional label \( "A_1 = a_1" \wedge \cdots \wedge "A_m = a_m" \). In this paper, for notational brevity, we use a conjunctive form and a vector form for propositional labels interchangeably depending on the context. Besides, to simplify the algorithm descriptions presented later, in a propositional label \( "X_1 = x_1" \wedge \cdots \wedge "X_n = x_n" \), we will always enumerate \( X_1, X_2, \ldots \) so that the order of enumeration preserves the original one \( A_1, A_2, \ldots \), i.e. for (\( j_1, j_2, \ldots j_n \) such that \( X_i \) corresponds to \( A_{j_i} \) (\( 1 \leq i \leq n \), \( 1 \leq j_i \leq m \)), \( j_i < j_i' \) holds when \( i < i' \).

Here consider a propositional label \( x = (x_1, \ldots, x_n) \). Then, a label \( x' = (x'_1, \ldots, x'_m) \) is called a subconjunction of \( x \) if \( \{x'_1, \ldots, x'_m\} \subseteq \{x_1, \ldots, x_n\} \), and we denote this by \( x' \subseteq x \). If \( x' \subseteq x \) but \( x' \neq x \), we write \( x' \subset x \). For an instance \( a \) and a propositional label \( x \), we say “\( a \) satisfies \( x \)” if \( x \subseteq a \). For a boolean attribute \( A_j \), we may abbreviate “\( A_j = \text{True} \)” and “\( A_j = \text{False} \)” as “\( A_j = T \)” and “\( A_j = F \)”, respectively.

2.2 Overview

In this paper, we consider probabilistic clustering based on a simple mixture model called a naive Bayes model. A naive Bayes model has a latent variable \( \mathcal{C} \) taking on the identifiers \( \{1, 2, \ldots, K\} \) of \( K \) clusters, and represents a simple joint distribution: \( p(C = k, A_1 = a_1, \ldots, A_m = a_m) = p(C = k) \prod_{j=1}^{m} p(A_j = a_j | C = k) \), or equivalently \( p(k, a) = p(k) \prod_j p(a_j | k) \). Here the probabilities \( p(k) \) and \( p(a_j | k) \) are treated as the model parameters. Given a dataset \( D \) of instances and the number \( K \) of clusters, we do:

1. Estimate the parameters in a model \( p(k, a) \) from \( D \).
2. Assign the most probable class \( k^*(a) = \arg\max_{1 \leq k \leq K} p(k | a) \) to each instance \( a \) based on the estimated parameters. The \( k \)-th cluster \( C_k \) is then formed as a set of instances \( a \) such that \( k^*(a) = k \).
3. Find propositional labels \( x \) that characterize well each cluster \( C_k \).

In the first two steps, we perform clustering, and the third step is called labeling. As is well-known, the first step is realized by the EM (expectation-maximization) algorithm \[8\]. From the second step, clustering can be casted as an unsupervised classification task, and we call \( p(k | a) \) the class membership probability of an instance \( a \).

In the last step, it is unspecified what are the propositional labels that characterize the clusters, and how to obtain them. The next two sections, Sections 2.3 and 2.4, address these issues, respectively.

2.3 Characteristic propositional labels

Relevance scores: To choose suitable propositional labels \( x = (x_1 \wedge \cdots \wedge x_n) \) or \( x = (x_1, \ldots, x_n) \), of a cluster \( C_k \) objectively and automatically, we introduce a scoring function that measures how relevant \( x \) and \( C_k \) are. Previously, several relevance scores have been proposed in various statistical/data-mining tasks. The followings are an adaptation of those relevance scores to our labeling problem:

- Growth rate: \( GR_k(x) = p(x | k)/p(x | \neg k) \), where \( \neg k \) indicates that the instance under consideration belongs to a class other than \( k \). This score is mainly used in emerging pattern mining \[9\] and explicitly states that the instances satisfying \( x \) are likely to occur in the cluster \( C_k \) and unlikely to occur in the clusters other than \( C_k \). \( GR_k(x) \) ranges from 0 (when \( p(x | k) = 0 \)) to \( \infty \) (when \( p(x | k) > 0 \) and \( p(x | \neg k) = 0 \)).
- Membership probabilities: \( p(k | x) \). PRIM, a rule-based method for bump hunting, tries to find \( x \) such that \( p(k | x) \geq r \), where \( r \) is some threshold, under a separate-and-conquer strategy \[10\]. It is crucial to see that, for a fixed \( k \), \( p(k | x) = p(k|x)/p(x) \propto p(x | k)/p(x) \) holds. In class association rule (CAR) mining \[11\], \( p(k | x) \) is called the confidence of a rule \( x \Rightarrow C_k \).

\[1\] As discussed in Section 4, we can also use the K-means algorithm for clustering.
- Pointwise mutual information: $\text{PMI}_k(x) = \log p(k, x) - \log(p(k)p(x))$. PMI has been used in text analysis [12]. This score is rewritten as $\log p(x | k) - \log p(x)$, which is adopted by a well-known probabilistic clustering tool AutoClass [13] for post-analysis (named “attribute influence values”), in a limited case with $|x| = 1$. The non-logarithmic version $p(k, x)/(p(k)p(x))$ is called the lift of a class association rule $x \Rightarrow C_k$ [14].

- Leverage: $\text{Leverage}_k(x) = p(k | x) - p(k)p(x)$. This score is often used for finding interesting association rules [15]. $\text{Leverage}_k(x)$ is equivalent to the weighted relative accuracy (WRACc), a score used in subgroup discovery, and can be rewritten as $p(x)(p(k | x) - p(k))$ or $p(k)(p(\neg k | x) - p(x | \neg k))$ [16]. A related score $|p(x | k) - p(x | \neg k)|$, often called support difference, is used in contrast set mining [17].

- TF-IDF: $\text{TF-IDF}_i(x) = p(x | k) \log(1/p(x))$. This is a popular measure in information retrieval [18], and is a product of term frequency (TF) and inverse document frequency (IDF). TF of a term $t$ in a document $d$ is the relative frequency of $t$ occurring in $d$, and IDF of $t$ is the logarithm of the inverse of the relative frequency that a document containing $t$ occurs in the whole document set. Then, assuming that a term occurs at most once in a document, the TF-IDF of a term $t$ in a document $d$ is given as $p(t | d) \log(1/p(t))$. Since TF-IDF is known to give a reasonably high score to $t$ that characterizes $d$, TF-IDF$_i(x)$ above can be used by analogy where $t$ corresponds to $x$, and $d$ corresponds to $k$.

- Precision/Recall: Precision and recall are also popular measures in information retrieval. In our context, $p(k | x)$ and $p(x | k)$ can be regarded as precision and recall of label $x$ for the $k$-th cluster [14]. Also in COBWEB [19], a well-known conceptual clustering method, $p(k | x)$ and $p(x | k)$ are respectively used as metrics for inter-class dissimilarity and intra-class similarity. To balance the opposite behavior of precision and recall, in information retrieval, we often use their harmonic mean $2p(k | x)p(x | k)/(p(k | x) + p(x | k))$ and call it the $F$-score. Lamirel et al. proposed the use of the $F$-score for automatic labeling of clustering results [6]. Similarly, the product of precision and recall $p(k | x)p(x | k)$, which substantially works as the geometric mean of $p(k | x)$ and $p(x | k)$, is used by Popescul and Ungar [5].

Other relevance scores are discussed in comprehensive surveys by Kralj Novak et al. [16] and by Geng et al. [14]. It is easy to show that $p(k | x_1) \leq p(k | x_2)$ iff $\text{GR}_k(x_1) \leq \text{GR}_k(x_2)$ [4] and $p(k | x_1) \leq p(k | x_2)$ iff $\text{PMI}_k(x_1) \leq \text{PMI}_k(x_2)$. Consequently, for a particular cluster $C_k$, the first three scores give the same ranking over the propositional labels. Hereafter we call $p(x | k)$ the local support, and $p(x)$ the global support. The relevance scores above commonly rely on the local support with a penalty regarding the global support. This contrastive use of the global support and the local support is also found in the category utility adopted in COBWEB [19].

In this paper, we choose $p(k | x)$ as the relevance score for two reasons on intuitiveness for the end users. First, we can of course interpret $p(k | x)$ as discriminative probabilities, by which we classify an instance satisfying $x$. As mentioned in Section 2.2, clustering is performed based on the membership probabilities $p(k | a)$, which are a special case of $p(k | x)$. The second reason is more practical: $p(k | x)$ is inherently normalized (i.e. $0 \leq p(k | x) \leq 1$). From this nature, we can use a threshold $r$, which just ranges over $(0, 1]$ and is commonly applied to all clusters, to filter out $x$ such that $p(k | x) < r$.

Minimality: Let us consider two propositional labels $x_1$ and $x_2$ that fulfill some requirement (e.g. $p(k | x_1) \geq r$ and $p(k | x_2) \geq r$ for some threshold $r$), and also suppose that $x_1 \subseteq x_2$ holds. In such a case, we favor $x_1$ over $x_2$, because the longer one may have some redundant information which hinders us from understanding the cluster.

In other words, we would like to have only minimal labels. In the literature on emerging pattern mining, such minimal patterns are called essential emerging patterns [20], and Ji et al. proposed an efficient mining algorithm named ConSGapMiner for minimal distinguishing sequences [21].

Model-based computation of relevance scores: We have introduced several relevance scores which are based on probabilities. In most of the previous work, these probabilities are directly estimated from a given dataset $D$ of instances. For example, membership probabilities are estimated as $\hat{p}(k | x) = \{a \in C_k | x \subseteq a \} / \{|a \in D | x \subseteq a\}$. In our method, on the other hand, relevance scores are computed from the model parameters via the joint distribution (Section 2.2). This model-based approach has a couple of advantages. First, as seen later, we can efficiently compute the scores, exploiting the conditional independence in the model, without scanning the whole dataset $D$. In many cases, the space for the model parameters is much smaller than the dataset. The second advantage is that the model parameters are well-abstracted data as long as the model fits to $D$, and there would be less chance to be affected by noise. Finally, there is a positive side-effect that we need not care about missing values in $D$ since we only use the parameters estimated by the EM algorithm.

\[ \text{GR}_k(x) = (p(k)/p(\neg k))^{-1}(p(k | x)/p(\neg k | x)) \propto p(k | x)/(1 - p(k | x)). \]
Selecting characteristic propositional labels: Now based on the discussions above, we define characteristic propositional labels, which characterize well the obtained clusters. A propositional label $x$ of the cluster $C_k$ is characteristic iff:

1. $p(k \mid x) \geq r$,
2. $p(x) \geq s_{global}$,
3. $p(x \mid k) \geq s_{local}$, and
4. There is no $x' \subset x$ that satisfies 1~3 above,

where $r$, $s_{global}$ and $s_{local}$ are user-specified thresholds, and the probabilities $p(k \mid x)$, $p(x)$ and $p(x \mid k)$ are computed via the joint distribution. Conditions 1~4 are called the relevance condition, the global support condition, the local support condition, and the minimality condition, respectively.

While most of the existing CAR mining algorithms run based on the guide from the threshold for $p(x \mid k)$, we treat the first and the fourth conditions as the primary filters. The remaining conditions are introduced to remedy the problem that we often obtain unintuitive characteristic labels with very low global/local support, and also to reduce the burden in the exhaustive search for characteristic labels, which will be described in the next section. So currently we do not consider to put a tight restriction on global/local support (e.g. $s_{local} = 1/(|D|/K) = K/|D|$, which implies that each of equally-sized clusters should contain at least one instance).

2.4 Exhaustive search for characteristic propositional labels

All possible propositional labels form a version space \cite{22}, and on this structure, we conduct an Apriori-style breadth-first search for the entire set of characteristic labels for each cluster. There are two major styles for such an exhaustive search: depth-first and breadth-first. We take a breadth-first style because, as seen later, it is easier to check the minimality of characteristic labels in a breadth-first style\footnote{ConSGapMiner mentioned above works in a depth-first fashion, and needs to introduce an extra data structure (a prefix tree) to reduce the time for the post-check on minimality.} and because we do not necessarily need very long characteristic labels that are difficult to read.

The Find procedure (Algorithm 1) is the main routine of the search algorithm for characteristic labels, which calls the GenCandidate function (Algorithm 2). The basic flow is similar to Apriori (GenCandidate is our version of the apriori-gen function in \cite{23}), but is different in that we make probability computation while generating candidates. In addition, since this probability computation requires normalization for each membership probability $p(k \mid x)$, the most part of the algorithm should work in parallel for clusters. It is also crucial to note that the global/local support of $x$ ($p(x)$ and $p(x \mid k)$) are anti-monotonic w.r.t. the inclusion relation (i.e. $p(x \mid k) \geq p(x' \mid k)$ if $x \subseteq x'$), but in general our relevance score is not. Instead, like ConSGapMiner, we make pruning based on the minimality of characteristic labels.

In the Find procedure, for each $C_k$, $S_n[k]$ indicates a set of propositional labels of length $n$ that satisfy the global/local support condition, and $R_n[k]$ indicates a set of labels in $S_n[k]$ that additionally satisfy the relevance condition. $R_n[k]$ are the characteristic labels of length $n$ which we wish to have, and we do not extend the labels in $R_n[k]$. $W_n[k] = S_n[k] \setminus R_n[k]$ are therefore the labels to be worked on next.

The candidate labels of length $(n + 1)$ are generated from the GenCandidate function, in which the labels of length $n$ in $W_n[k]$ are combined effectively. In Line 5 of GenCandidate, like the “prune” step of Apriori, “SubCon($x_{ext}$) \subseteq W_n[k]” filters out the over-generated candidate labels using anti-monotonicity of global/local support and minimality at the same time. SubCon($x$) is a function that returns a set of $x$’s subconjunctions of length $|x| - 1$\footnote{More specifically, for $x = (x_1, \ldots, x_{n-1}, x_n)$, SubCon($x$) = \{(x_2, x_3, \ldots, x_{n-1}, x_n), (x_1, x_3, \ldots, x_{n-1}, x_n), \ldots, (x_1, x_2, \ldots, x_{n-2}, x_n), (x_1, x_2, \ldots, x_{n-2}, x_{n-1})\}.} and using the property that $W_n[k] = S_n[k] \setminus R_n[k]$, the filtering condition requires that each of the immediate subconjunctions of $x_{ext}$ should be in $S_n[k]$ (due to anti-monotonicity), but should not be in $R_n[k]$ (due to minimality). This way of filtering, together with the breadth-first strategy, enables us to perform effective pruning by only checking the labels in $W_n[k]$\footnote{$W_n[k]$ is constructed from the labels in $W_{n-1}[k]$, and hence is guaranteed not to include any labels $x$ such that $x' \subseteq x$, $x' \in R_{n'}[k]$ and $1 \leq n' \leq n$.}. Then, for each candidate label that has passed the filter, we compute the probabilities $p(x \mid k)$, $p(x)$ and $p(k \mid x)$ (Lines 11~13). The point here is that we take the union of $C_{n+1}[k]$’s in advance (Line 11) to avoid a redundant computation, and reuse the previously computed values for $p(x_{prev} \mid k)$, exploiting the conditional independence in the naive Bayes model.

To speed-up further the search algorithm in the case with many attributes, in the Find procedure, we optionally introduce a greedy pruning, similarly to a commercial data-mining tool named Magnum Opus \cite{15}. To be more
concrete, we delete \( a_j \) such that \( p(k \mid a_j) < p(k) \) from \( S_1[k] \) after Line 2. In addition, \( x = (x_1, \ldots, x_n, x_{n+1}) \in S_{n+1}[k] \) such that \( p(k \mid x) < p(k \mid x') \), where \( x' = (x_1, \ldots, x_n) \), are considered as unpromising, and deleted from \( S_{n+1}[k] \) after Line 11. This greedy pruning is unsafe, i.e. we may miss some characteristic labels actually satisfying the conditions in Section 2.3 but it would bring high efficiency in many practical cases.

### 2.5 Handling continuous attributes

Until now, we have assumed that all attributes are discrete. To handle continuous attributes and discrete attributes consistently in terms of membership probabilities, we also “propositionalize” each continuous attribute. To be more specific, as is often done in mixture modeling, we consider that each continuous attribute follows a univariate Gaussian distribution, in which two types of parameters, the mean \( \mu_{jk} \) and the variance \( \sigma^2_{jk} \) are introduced for the \( j \)-th continuous attribute \( A_j \) and the \( k \)-th cluster \( C_k \). These parameters are also estimated by the EM algorithm. We further assume that we are given a set \( Q = \{q_1, q_2, \ldots, q_{|Q|}\} \) of different probabilities, where \( 0 < q_h < 1 \) for \( 1 \leq h \leq |Q| \), and the indices are given so that \( q_h < q_{h'} \) if \( h < h' \). For instance, we may have \( Q = \{0.1, 0.2, \ldots, 0.9\} \). Then, using a cumulative distribution function \( F_{jk} \) with the mean \( \mu_{jk} \) and the variance \( \sigma^2_{jk} \) for each \( A_j \) and \( C_k \), we introduce \( \alpha_h^{(jk)} < \beta_h^{(jk)} \) as a conjunct in a propositional label, where \( \alpha_h^{(jk)} = \mu_{jk} - \sigma_{jk} \) and \( \beta_h^{(jk)} = \mu_{jk} + \sigma_{jk} \) such that \( F_{jk}(\alpha_h^{(jk)}) - F_{jk}(\beta_h^{(jk)}) = q_h \). It can be seen here that \( \alpha_h^{(jk)} \) and \( \beta_h^{(jk)} \) are symmetric w.r.t. the mean \( \mu_{jk} \). Hereafter we omit the superscript \((j, k)\) unless they are needed.

Now consider \( x = (x_0 \land \alpha_h, x_0 \land \beta_h) \) and \( x' = (x_0 \land \alpha_h', x_0 \land \beta_h') \) where \( h < h' \). Then, we define \( x' < x \) and we have \( p(x) < p(x') \). With this new inclusion relation, in the search algorithm, an additional minimality check is made for the last conjunct corresponding to a continuous attribute, just after \( R_n[k] \) being computed. One may see that \( \alpha_h \)’s and \( \beta_h \)’s above are model-based quantile values\(^6\) and choosing an appropriate \( \alpha_h < A_j \leq \beta_h \) leads to an automatic adjustment of \((\alpha_h, \beta_h)\), which resembles the ‘peeling’ operation in PRIM, a rule-based bump hunting method\(^7\).

### 3 Experiments

In the experiments, we used four datasets: the zoo dataset, the iris dataset, the 20 newsgroup dataset and the flags dataset\(^8\). For the first three datasets, we gave the correct number \( K \) of clusters to the clustering algorithm, considering ideal situations. We then compare the obtained characteristic labels and the original (human-annotated)
creatures in “reptiles” and “amphibians” are merged into cluster number of clusters by characteristic labels together with a Bayesian score for model selection. For simplicity, classes. On the other hand, since the flags dataset does not contain the class information, we explore a plausible

Algorithm 2 GenCandidate(W₁[1],...,Wₖ[K])

1: for all k = 1, 2, ..., K do
2: Cₙ₊₁[k] := ∅
3: for all x = (x₁,...,xₖ₋₁,xₖ) ∈ Wₖ[k] and x’ = (x₁,...,xₖ₋₁,x’ₖ) ∈ Wₖ[k] such that ∀j: xₖ,j’ ∉ V(A₀) do
4: xext := (x₁,...,xₖ₋₁,xₖ)
5: if SubCon(xext) ⊆ Wₖ[k] then
6: Cₙ₊₁[k] := Cₙ₊₁[k] ∪ {xext}
7: end if
8: end for
9: end for
10: Dₙ₊₁ := ∪ₖ Cₙ₊₁[k]
11: for all x = (x₁,...,xₖ₋₁,xₖ,xₖ₊₁) ∈ Dₙ₊₁ do
12: xprev := (x₁,...,xₖ)
13: p(x | k) := p(xprev | k)p(xₖ₊₁ | k) for k = 1,...,K
14: p(x) := ∑ₖ p(k)p(x | k)
15: end for
16: p(k | x) := p(k)p(x | k)/p(x) for k = 1,...,K and x ∈ Cₙ₊₁[k]
17: return (Cₙ₊₁[1],...,Cₙ₊₁[K])

classes. On the other hand, since the flags dataset does not contain the class information, we explore a plausible number of clusters by characteristic labels together with a Bayesian score for model selection. For simplicity, throughout the experiments, we set a small value (1/|D|) to the threshold slocal for the global support p(x), so that the influence from slocal is negligible. In addition, we tried 1,000 re-initializations in the EM algorithm not to get trapped into unwanted local optima.

3.1 Zoo dataset

The zoo dataset describes the classification of 101 species of creatures with 17 attributes. The species are originally categorized into seven classes. Table 1(top-left) shows the confusion matrix of the clustering result. We can see from this matrix that the creatures in the class “mammals” are split into two clusters C₁ and C₂, whereas the creatures in “reptiles” and “amphibians” are merged into cluster C₃. Besides, the remaining tables in Table 1 show the obtained characteristic labels for C₁, C₂ and C₃, where C₃ corresponds to the original class “birds.” The labels in the tables are ordered firstly according to the length of x (i.e. syntactic generality), secondly according to the magnitude of p(x | k) (i.e. statistical generality), and thirdly according to the magnitude of p(k | x) as the thresholds for p(k | x) and p(x | k), respectively, where D is the dataset and K = 3 is the number of clusters.

Since the original classes are unknown in real situations, we interpret the clusters C₁, C₂ and C₃, only from the obtained characteristic labels. For example, all creatures in C₁ have feathers, so we can guess that C₁ corresponds to birds. Also there are several plausible labels for C₁ which support our guess. Interestingly, on the other hand, the obtained labels indicate that the (wrongly) split classes C₁ and C₂ correspond to terrestrial and aquatic mammals, respectively. So one may conclude that these split clusters are still meaningful. In the past, to evaluate the quality of the obtained clusters, there has been no way but to numerically check the closeness between the obtained clusters and the human-annotated classes, using some matching criteria, such as purity, normalized mutual information and the (adjusted) Rand index. Contrastingly, as seen above, the characteristic labels provide us with a new and in-depth way for cluster evaluation. Similar interpretations are possible for the other clusters, whose characteristic labels are shown in Table 2.

3.2 Iris dataset

As a typical continuous dataset, we picked up the iris dataset, in which there are four attributes: petal width, petal length, sepal width and sepal length. Each of 150 cases in the dataset originally belongs to one of three classes: Setosa, Versicolour and Virginica. The confusion matrix and the obtained labels are shown in Table 3. We used

---

9 We also observed that intuitive labels tend to be highly ranked according to the harmonic mean of p(k | x) and p(x | k).
Table 1. The confusion matrix, and the characteristic labels for the clusters C₁, C₂ and C₃ in the zoo dataset.

| original classes | clusters C₁ | C₂ | C₃ | C₄ | C₅ | C₆ | C₇ |
|------------------|----------|----|----|----|----|----|----|
| mammals          | 35       | 6  | 0  | 0  | 0  | 0  | 0  |
| birds            | 0        | 0  | 20 | 0  | 0  | 0  | 0  |
| fishes           | 0        | 0  | 0  | 13 | 0  | 0  | 0  |
| amphibians       | 0        | 0  | 0  | 4  | 0  | 0  | 0  |
| reptiles         | 0        | 0  | 0  | 0  | 5  | 0  | 0  |
| insects          | 0        | 0  | 0  | 0  | 0  | 8  | 0  |
| others           | 0        | 0  | 0  | 0  | 1  | 2  | 7  |

The 20 newsgroups dataset is originally a collection of approximately 20,000 articles from 20 different newsgroups. A preprocessed dataset available from [http://people.csail.mit.edu/jrennie/20Newsgroups/](http://people.csail.mit.edu/jrennie/20Newsgroups/) is used, and the articles from three newsgroups: comp.sys.ibm.pc.hardware, rec.sport.hockey and soc.religion.christian. We made further preprocessing: stemming by the Porter’s algorithm [18], removing infrequent words (≤ 200 occurrences), removing short articles (≤ 10 words) and removing the attributes taking only one value. The dataset was finally converted into 2,799 bag-of-words boolean vectors whose dimension is 2,016. In labeling by the proposed method, we did not use the conjuncts of the form “w = False” (or “w = F”).
which means the absence of word w in the article. The thresholds r and s\textsubscript{local} were respectively configured as 0.9 and 10 × K/|D|\textsuperscript{10}. Furthermore, we applied the greedy pruning described at the last of Section 2.4.

The results are shown in Table \textsuperscript{30}. From the obtained characteristic labels for C\textsubscript{1}, it is seen that the article containing words such as “hockey” (“hockey”; the suffix should have been replaced by the stemmer) and “nhl” (“NHL”; the National Hockey League) are likely to belong to C\textsubscript{1}. There are also the names of a hockey team and its home city (i.e. Pittsburgh Penguins). So we can guess from this information that C\textsubscript{1} is a cluster of articles related to hockey. Similarly, it is easy to see that C\textsubscript{2} is a cluster of articles related to computer hardware\textsuperscript{11} from the words such as “mb” (“megabytes” or “motherboard”), “disk” and “motherboard.” C\textsubscript{3} would be understood as a cluster that contains the articles talking about religious matters. Although there are many attributes in this dataset, our search algorithm is feasible\textsuperscript{11} thanks to the pruning based on the minimality and the optimized setting described above.

### 3.4 Flags dataset

The flags dataset contains the details of 194 national flags, originally described by 30 attributes. In this experiment, we focused on the clusters of national flags grouped on their visual aspects, and hence non-visual attributes (landmass, zone, area, population, language and religion) were removed in advance. As is written above, since

\textsuperscript{10} s\textsubscript{local} was configured as 10 × K/|D| because the 20 newsgroup dataset is 10 times (or more) larger than the zoo and the iris dataset.

\textsuperscript{11} As shown in the confusion matrix in Table \textsuperscript{4}, C\textsubscript{2} contains the articles from soc.religion.christian, but the characteristic labels related to religion did not appear. This would be because the articles from soc.religion.christian mainly use non-technical terms, which are less likely to form characteristic labels.

\textsuperscript{12} It took 404 seconds on a PC with Core i7 2.66GHz to get all characteristic labels for all clusters. Currently the search algorithm is implemented in the Ruby script language.

---

Table 2. The obtained characteristic labels for clusters C\textsubscript{4}, C\textsubscript{5}, C\textsubscript{6} and C\textsubscript{7}.

| labels for C\textsubscript{4} | $p(k|x)$ | $p(x|k)$ |
|-------------------------------|----------|----------|
| milk=F ∧ fin=T               | 1.000    | 1.000    |
| breathes=F ∧ tail=T          | 0.948    | 1.000    |
| eggs=T ∧ fin=T               | 0.951    | 1.000    |
| toothed=T ∧ breathes=F       | 0.941    | 1.000    |
| backbone=T ∧ breathes=F      | 0.935    | 1.000    |
| hair=F ∧ fin=T               | 1.000    | 1.000    |
| fin=T ∧ csize=F              | 0.906    | 1.000    |
|                             |          |          |

| labels for C\textsubscript{5} | $p(k|x)$ | $p(x|k)$ |
|-------------------------------|----------|----------|
| venomous=T ∧ legs=4           | 0.943    | 0.239    |
| eggs=F ∧ milk=F               | 1.000    | 0.199    |
| milk=F ∧ toothed=T ∧ fin=F    | 1.000    | 0.799    |
| hair=F ∧ toothed=T ∧ fin=F    | 0.935    | 0.799    |
| milk=F ∧ toothed=T ∧ breathes=T| 1.000    | 0.719    |
| milk=F ∧ breathes=T ∧ legs=4  | 1.000    | 0.539    |
| feathers=F ∧ milk=F ∧ backbone=T ∧ fin=F | 1.000    | 0.899 |
| hair=F ∧ feathers=F ∧ backbone=T ∧ fin=F | 0.931 | 0.899 |
|                             |          |          |

| labels for C\textsubscript{6} | $p(k|x)$ | $p(x|k)$ |
|-------------------------------|----------|----------|
| backbone=F ∧ breathes=T       | 0.916    | 1.000    |
| predator=F ∧ backbone=F      | 0.978    | 0.899    |
| breathes=T ∧ legs=6           | 1.000    | 0.800    |
| aquatic=F ∧ legs=6            | 0.965    | 0.800    |
| predator=F ∧ legs=6           | 1.000    | 0.720    |
| airborne=T ∧ backbone=F       | 1.000    | 0.600    |
| feathers=F ∧ eggs=T ∧ airborne=T ∧ toothed=F | 1.000 | 0.600 |
|                             |          |          |
the class information is not given in this dataset, we first estimated the number of clusters as \( \hat{K} \) by the Cheeseman-Stutz score \([13]\), a Bayesian model selection criterion adopted in AutoClass, and then starting from \( \hat{K} \), we explored a plausible number of clusters by observing the characteristic labels. Another point in this dataset is that discrete attributes and continuous attributes are mixed. That is, all of eight integer attributes (e.g. the number of circles in Stutz score \([13]\), a Bayesian model selection criterion adopted in AutoClass, and then starting from \( \hat{K} \), we explored a plausible number of clusters by interacting with characteristic labels, with a help from model selection techniques. Also we conducted the greedy pruning.

Fig. 2 shows the curve of the Cheeseman-Stutz score with various numbers of clusters, and we have \( \hat{K} = 5 \) as a peak of this curve. We further continued to compute characteristic labels with the number \( K \) of clusters being around \( \hat{K} \), and found that readable characteristic labels are obtained when \( K = 6 \). Table 3 presents these labels.\(^1\)

The shortest characteristic label for the cluster \( C_1 \) says that the national flags in \( C_1 \) (and none in the other clusters) have one saltire (diagonal cross). A typical example of such flags is the Union Jack, and actually many flags in \( C_1 \) have one quartered section (i.e. #quarters=1) for the Union Jack. Similarly, the clusters \( C_2 \) and \( C_3 \) contain the flags with vertical bars and with circles, respectively. The label (#saltires=0 \& #quarters=1) for \( C_6 \) distinguishes \( C_1 \) and \( C_6 \), and similarly the labels (#crosses=1 \& #saltires=0) and (#crosses=1 \& #quarters=0) for \( C_4 \) jointly work for distinguishing \( C_4 \) from \( C_1 \) and \( C_6 \), where #crosses indicates the number of upright crosses. Indeed, \( C_6 \) contains the flag of the United States, and \( C_4 \) contains the flags of several Scandinavian countries (note that the Union Jack also contains upright crosses). From the labels for \( C_5 \), one may see that \( C_5 \) is a cluster of miscellaneous flags. On the other hand, when the number \( K \) of clusters is set at \( \hat{K} = 5 \), the clusters \( C_2 \) and \( C_3 \) are merged into one cluster, whose characteristic labels are not so intuitive as in Table 3. These results imply that a plausible number of clusters can be determined by interactively consulting characteristic labels, with a help from model selection techniques, and clearly exemplify how the feedbacks from the interpretation/evaluation step contribute in knowledge discovery.

### 4 Related work

As mentioned above, there have been only a few labeling approaches. LabelSOM \([3]\) is a labeling method for self-organizing maps, and Mei et al.’s automatic labeling method for unigram topic models \([4]\) uses a heuristic score based on pointwise mutual information. As described in Section 2, different relevance measures are used

---

\(^1\) Since each continuous attribute \( A_j \) is originally an integer attribute, a proposition "\( \alpha < A_j \leq \beta \)" (assume here that \( \alpha \) and \( \beta \) are not integers, for simplicity) was translated back into "\( A_j = [\alpha], [\alpha] + 1, \ldots, [\beta] \)" in Table 3. Non-minimal labels produced by this translation were then removed.
by Popescul and Ungar [5] and by Lamirel et al. [6] for automatic labeling of document clusters. In these labeling methods, the length of possible labels seems to be limited in advance, and thus no pruning mechanism, like the one described in Section 3.4 is given.

CLIQUE [7] is a novel hyper-rectangular clustering method that additionally gives comprehensible descriptions of the obtained clusters. The description of each cluster is a DNF formula of the ranges of continuous attributes such as \((30 \leq \text{age} < 50) \land (4 \leq \text{salary} < 8)) \lor ((40 \leq \text{age} < 60) \land (2 \leq \text{salary} < 6)).\) Although CLIQUE has a similar motivation to ours, it is mainly designed for the dataset with continuous attributes. According to the original description (“Remarks” in Section 2.2 of [7]), if we use discrete attributes, all instances in a cluster must take the same value for each discrete attribute in a selected subspace. In the proposed method, contrastingly, we do not have such a restriction, and as seen in Section 3.4 we can make use of advanced statistical techniques such as ones for model selection in the clustering step. The latter point also contrasts the proposed method with conceptual clustering methods such as COBWEB [19].

In the research on expert systems, it has been a problem to explain the expert system’s conclusion to human users. Wolverton [25] proposed the use of satisficing conclusion-substantiating (SCS) explanations to explain an expert system’s conclusion. Given a system’s conclusion \(c\) and a threshold \(\rho\), the SCS explanation \(e\) is the shortest sequence of facts such that \(p(c \mid e) > \rho\) (or if no such sequence of facts, \(e = \arg\max_{e'} p(c \mid e')\)). Our search algorithm would contribute in efficient finding of SCS explanations.

Traditional rule induction methods such as C4.5 and RIPPER [24] can also be applied to find comprehensible cluster descriptions. However, Hotho et al. reported that these methods tend to produce too many rules to manage for human [1]. One possible reason is that C4.5 and RIPPER have a representational limitation that the premises in the obtained rules are always exclusive and need to be understood fragmentarily. In the proposed method, on the other hand, each characteristic label is independently interpretable. Another possibility is that C4.5 and RIPPER tried to find the exact boundaries among clusters, by their design. In labeling, however, we do not always
have to find such exact boundaries. Furthermore, traditional rule induction methods often suffer from a so-called rare-class problem [27] when we have imbalanced or many clusters (if there are many clusters, each cluster is relatively rare). For example, small groups of instances (small disjuncts) in a rare class are often missed. Actually, in the zoo dataset, C4.5/RIPPER only generated the rules for the cluster C_3 ("birds"); “feathers=True” \(\Rightarrow C_3\), and “feathers=False” \(\Rightarrow \neg C_3\), and the rest of the antecedent patterns we found (Table 1 bottom-right) were ignored. This is presumably because most of the instances have been covered by the simple rules above in the rule construction process of C4.5/RIPPER. It is reported that a classifier based on emerging patterns works well for the rare-class problem [28].

Recently it is proposed in [16] to unify three similar data mining tasks, contrast set mining, emerging pattern mining and subgroup discovery, under the name of supervised descriptive rule discovery. Our labeling method can be seen as a model-based approach in this framework, which focuses on interpretation/evaluation of probabilistic clusters. In a broader context, for knowledge discovery under an unsupervised setting, a sequential run of clustering and discriminative labeling would be a promising alternative to frequent pattern mining. Besides, also recently, Zimmermann and De Raedt introduced a general data mining task called cluster-grouping [29], and a branch-and-bound algorithm, named CG, for this task. CG efficiently finds characteristic patterns (labels, in our case) following a guide from a convex relevance score such as \(\chi^2\), information gain (used in ID3), WRAcc (Section 2.3) and category utility (used in COBWEB). Although this algorithm is powerful, it could not be directly applied to our labeling problem, since the membership probability \(p(k \mid x)\) seems not convex.

![Cheeseman-Stutz scores with various numbers of clusters](image)

**Fig. 2.** The Cheeseman-Stutz scores with various numbers of clusters.

| labels for \(C_1\) | \(p(k \mid x)\) | \(p(x \mid k)\) |
|---------------------|-----------------|-----------------|
| \#saltires=1        | 1.000           | 0.900           |
| topleft=white \& \#quarters=1 | 0.817 | 0.622 |
| stripes=0,1,2 \& \#quarters=1 | 0.827 | 0.540 |
| botright=blue \& \#quarters=1 | 0.819 | 0.505 |
| green=T \& \#crosses=1 | 0.906 | 0.467 |
| gold=T \& \#crosses=1 | 0.763 | 0.467 |
| mainhue=blue \& \#quarters=1 | 0.810 | 0.467 |
| \#crosses=1 \& \#quarters=1 | 0.751 | 0.420 |

| labels for \(C_2\) | \(p(k \mid x)\) | \(p(x \mid k)\) |
|---------------------|-----------------|-----------------|
| \#bars=1,2,3,4 | 0.782 | 0.800 |

| labels for \(C_3\) | \(p(k \mid x)\) | \(p(x \mid k)\) |
|---------------------|-----------------|-----------------|
| \#circles=1,2 \& \#crosses=0 | 0.781 | 0.540 |
| \#circles=1,2 \& \#quarters=0 | 0.781 | 0.540 |
| black=T \& \#crosses=1 | 0.766 | 0.225 |
| blue=F \& \#circles=1 | 0.765 | 0.200 |
| botright=green \& \#circles=1,2 | 0.781 | 0.181 |
| topleft=orange \& \#saltires=0 | 0.999 | 0.135 |
| topleft=orange \& \#crosses=0 | 0.970 | 0.135 |
| mainhue=orange \& \#crosses=0 | 0.970 | 0.135 |

| labels for \(C_4\) | \(p(k \mid x)\) | \(p(x \mid k)\) |
|---------------------|-----------------|-----------------|
| \#crosses=1 \& \#saltires=0 | 0.810 | 0.81003 |
| \#crosses=1 \& \#quarters=0 | 0.829 | 0.81002 |
| \#crosses=1 \& \#sunstars=0 | 0.751 | 0.720 |
| \#circles=0 \& \#crosses=1 | 0.768 | 0.640 |
| green=F \& \#crosses=1 | 0.757 | 0.500 |
| \#color=2,3 \& \#crosses=1 | 0.759 | 0.490 |
| gold=F \& \#crosses=1 | 0.754 | 0.356 |

| labels for \(C_5\) | \(p(k \mid x)\) | \(p(x \mid k)\) |
|---------------------|-----------------|-----------------|
| \#bars=0 | 0.803 | 0.900 |
| \#circles=0 | 0.752 | 0.900 |
| \#crosses=0 | 0.755 | 0.600 |
| \#quarters=0 | 0.752 | 0.400 |
| triangle=T | 0.889 | 0.240 |
| botright=black | 0.888 | 0.080 |
| mainhue=black | 0.799 | 0.040 |

| labels for \(C_6\) | \(p(k \mid x)\) | \(p(x \mid k)\) |
|---------------------|-----------------|-----------------|
| \#saltires=0 \& \#quarters=1 | 0.960 | 0.360 |
| topleft=blue \& \#quarters=1 | 0.875 | 0.320 |

**Table 5.** The characteristic labels for \(C_1, \ldots, C_6\) in the flags dataset.
In the context of probabilistic modeling, the proposed method with mixture models could be extended for evidence-based sensitivity analysis (e.g. [30]) or explanatory analysis (e.g. [31]) of Bayesian networks, in which the membership probability \( p(k \mid x) \) is generalized as \( p(q \mid e) \), where \( q \) is an instantiation of a query variable and \( e \) is an instantiation of (a part of) evidence variables, and thus we search for a minimal combination \( e \) of evidences which is highly influential to the observation \( q \). To the best of our knowledge, the most recent and closest work is Yuan et al.’s general framework for most relevant explanation (MRE) [32,33]. Their MRE framework adopts a relevance score called generalized Bayes factor (GBF), defined as \( \text{GBF}_k(x) = p(k \mid x)/p(k \mid \sim x) \) in our labeling problem. The MRE framework looks attractive, but seems unfit to our case for a couple of reasons. First, for the \( k \)-th cluster, a ranking over the propositional labels \( x \) by \( \text{GBF}_k(x) \) is different from the one used in the clustering step (i.e. by \( p(k \mid x) \)). Second, \( \text{GBF}_k(x) = \frac{1-p(x \mid k)}{1-p(x \mid k)} \cdot \frac{p(x \mid k)}{p(x)} \) can be numerically unstable when \( p(x \mid k) \approx 1 \). For instance, we cannot order the labels \( x \) such that \( p(x \mid k) = 1 \), which in fact appear in one of our experiments (i.e. Table [3]). Third, the MRE framework only provides an MCMC-based approximate method or an exact (exhaustive) method without safe pruning (like the one based on global/local support and minimality in the proposed method) for finding relevant \( x \). Lastly, the MRE papers do not describe how to handle continuous attributes.

Handling continuous attributes is an important issue in CAR (class association rule) mining. For example, Washio et al. [34] proposed a CAR mining method that discretizes the continuous space on the fly with hyper-rectangular clustering. The difference from our labeling method is that we are given probabilistic clusters from beginning and thus we effectively limit propositions to the ones of the form “\( \alpha < A_j \leq \beta \)”, where \( \alpha \) and \( \beta \) are symmetric w.r.t. the mean in the cluster. Besides, as in usual CAR mining, Washio et al.’s method searches for the antecedent patterns \( x \) based on the local support \( p(x \mid k) \).

Section [3] described that the EM algorithm is adopted for clustering. We can also use the \( K \)-means algorithm instead, since \( K \)-means can be seen as an instance of a parameter estimation framework often called Viterbi training [14] tailored for a Gaussian mixture model with equal class probabilities and a common covariance matrix of the form \( \sigma^2 I \) [35]. Once the model parameters have been estimated, our labeling method is applicable as written in this paper. Similarly to the case in Section [3], when combined with \( K \)-means, the Euclidean distance from the centroid (the mean in the cluster) is translated into a cumulative probability under a Gaussian distribution.

5 Conclusion and future work

In this paper, we proposed a new labeling method that associates propositional labels (conjunctions of attribute-value pairs) with the clusters obtained by mixture models, to help us interpret or evaluate the clusters. As shown in the experimental results, the proposed method finds a set of intuitive descriptive labels that characterize well or “verbalize” the clusters. The proposed method is fully applicable to various datasets including continuous attributes and missing values, and can be a new, in-depth and consistent tool for cluster interpretation/evaluation. Besides, the experimental results also show that the feedbacks from the interpretation/evaluation step can play an important role for achieving a reasonable clustering result. In future work, we would like to extend the proposed method to use disjunctive formulas or a richer representation. For example, we may merge two similar characteristic labels (milk=T \( \land \) legs=4) and (hair=T \( \land \) legs=4) into ((milk=T \( \lor \) hair=T) \( \land \) legs=4) to gain a higher local support. In a purely logical sense, our labeling algorithm can be formulated under the setting of inductive logic programming (ILP) with a simple refinement operator. As in ILP, the use of background knowledge such as taxonomy seems helpful for having more comprehensible descriptions.

Acknowledgments.

The authors would like to thank Toshihiro Kamishima for his helpful comments on related work. This work is supported in part by Grant-in-Aid for Scientific Research (No. 20240016) from Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

1. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Computing Surveys 31(3) (1999) 264–323
2. Hotho, A., Staab, S., Stumme, G.: Explaining text clustering results using semantic structures. In: Proc. of the 7th European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD-03). (2003)

14 Viterbi training has also been called hard EM, Viterbi EM, classification EM, sparse EM, and so on.
3. Rauber, A.: LabelSOM: on the labeling of self-organizing maps. In: Proc. of the 1999 Int'l Joint Conf. on Neural Networks (IJCNN-99). (1999) 3527–3532
4. Mei, Q., Shen, X., Zhai, C.: Automatic labeling of multinomial topic models. In: Proc. of the 13th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD-07). (2007) 490–499
5. Popescul, A., Ungar, L.H.: Automatic labeling of document clusters. Unpublished manuscript available from http://www.cis.upenn.edu/~popescul/ (2000)
6. Lamirel, J.C., Ta, A.P., Attik, M.: Novel labeling strategies for hierarchical representation of multidimensional data analysis results. In: Proc. of the 26th IASTED Int'l Conf. on Artificial Intelligence and Applications (AIA-08). (2008) 169–174
7. Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic subspace clustering of high dimensional data for data mining applications. In: Proc. of the 1998 ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD-98). (1998) 94–105
8. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. of the Royal Statistical Society B39 (1977) 1–38
9. Dong, G., Li, J.: Efficient mining of emerging patterns: discovering trends and differences. In: Proc. of the 5th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD-99). (1999) 43–52
10. Friedman, J.H., Fisher, N.I.: Bump hunting in high-dimensional data. Statistics and Computing 9 (1999) 123–143
11. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In: Proc. of the 4th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD-98). (1998) 80–86
12. Church, K.W., Hanks, P.: Word association norms, mutual information, and lexicography. In: Proc. of the 27th Annual Meeting on Association for Computational Linguistics (ACL-89). (1989) 76–83
13. Cheeseman, P., Stutz, J.: Bayesian classification (AutoClass): theory and results. In: Advances in Knowledge Discovery and Data Mining. The MIT Press (1995)
14. Geng, L., Hamilton, H.J.: Interestingness measures for data mining: a survey. ACM Computing Surveys 38(3) (2006) 1–32
15. Webb, G.I., Butler, S., Newlands, D.: On detecting differences between groups. In: Proc. of the 9th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD-03). (2003) 256–265
16. Kralj Novak, P., Lavraˇ c, N., Webb, G.I.: Supervised descriptive rule discovery: a unifying survey of contrast set, emerging pattern and subgroup mining. J. of Machine Learning Research 10 (2009) 377–403
17. Bay, S.D., Pazzani, M.J.: Detecting group differences: mining contrast sets. Data Mining and Knowledge Discovery 5 (2001) 213–246
18. Manning, C., Raghavan, P., Shütze, H.: Introduction to Information Retrieval. Cambridge Univ. Press (2008)
19. Fisher, D.: Knowledge acquisition via incremental conceptual clustering. Machine Learning 2 (1987) 139–172
20. Fan, H., Ramamohanarao, K.: A Bayesian approach to use emerging patterns for classification. In: Proc. of the 14th Australasian Database Conf. (ADC-03). (2003) 39–48
21. Ji, X., Bailey, J., Dong, G.: Mining minimal distinguishing subsequence patterns with gap constraints. Knowledge and Information Systems 11(3) (2007) 259–286
22. Mitchell, T.: Machine Learning. McGraw-Hill (1997)
23. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proc. of the 20th Conf. on Very Large Data Bases (VLDB-94). (1994) 487–499
24. Meilă, M.: Comparing clusterings — an information based distance. J. of Multivariate Analysis 98 (2007) 873–895
25. Wolverton, M.: Presenting significant information in expert systems explanation. In: Proc. of the 7th Portuguese Conf. on Artificial Intelligence. (1995) 435–438
26. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. 2nd edn. Morgan Kaufmann (2005)
27. Weiss, G.M.: Mining with rarity: a unifying framework. ACM SIGKDD Explorations 6(1) (2004) 7–19
28. Ramamohanarao, K., Bailey, J., Fan, H.: Efficient mining of contrast patterns and their applications to classification. In: Proc. of the 3rd Int'l Conf. on Intelligent Sensing and Information Processing (ICISIP-05). (2005) 39–47
29. Zimmermann, A., De Raedt, L.: Cluster grouping: from subgroup discovery to clustering. Machine Learning 77 (2009) 125–159
30. Jensen, F.V.: An Introduction to Bayesian Networks. UCL Press (1996)
31. Chajewska, U., Halpern, J.Y.: Defining explanation in probabilistic systems. In: Proc. of the 13th Conf. on Uncertainty in Artificial Intelligence (UAI-97). (1997) 62–71
32. Yuan, C., Lu, T.C.: A general framework for generating multivariate explanations in Bayesian networks. In: Proc. of the 23rd AAAI Conf. on Artificial Intelligence (AAAI-08). (2008) 1119–1124
33. Yuan, C., Liu, X., Lu, T.C., Lim, H.: Most relevant explanation: properties, algorithms, and evaluations. In: Proc. of the 25th Conf. on Uncertainty in Artificial Intelligence (UAI-09). (2009) 631–638
34. Washio, T., Nakanishi, K., Motoda, H.: A classification method based on subspace clustering and association rules. New Generation Computing 25 (2007) 235–245
35. Celeux, G., Govaert, G.: A classification EM algorithm for clustering and two stochastic versions. Computational Statistics & Data Analysis 14 (1992) 315–332