Exploiting clustering algorithms in a multiple-level fashion: A comparative study in the medical care scenario

Tania Cerquitelli, Silvia Chiusano, and Xin Xiao

Control and Computer Engineering Department, Politecnico di Torino, Corso Duca degli Abruzzi, 24 – 10129 Torino, Italy.
name.surname@polito.it

Abstract. Clustering real-world data is a challenging task, since many real-data collections are characterized by an inherent sparseness and variable distribution. An appealing domain that generates such data collections is the medical care scenario where collected data include a large cardinality of patient records and a variety of medical treatments usually adopted for a given disease pathology.
This paper proposes a two-phase data mining methodology to iteratively analyze different dataset portions and locally identify groups of objects with common properties. Discovered cohesive clusters are then analyzed using sequential patterns to characterize temporal relationships among data features. To support an automatic classification of a new data objects within one of the discovered groups, a classification model is created starting from the computed cluster set. A mobile application has been also designed and developed to visualize and update data under analysis as well as categorizing new unlabeled records.
A comparative study has been conducted on real datasets in the medical care scenario using diverse clustering algorithms. Results were compared in terms of cluster quality, execution time, classification performance and discovered sequential patterns. The experimental evaluation showed the effectiveness of MLC to discover interesting knowledge items and to easily exploit them through a mobile application. Results have been also discussed from a medical perspective.

keywords: Cluster analysis, data with a variable distribution, diabetic patient treatments, multiple-level method, comparison.

1 Introduction

Cluster analysis is an exploratory technique which aims at grouping a data object collection into subsets (clusters) based on object properties, without the support of additional a priori knowledge [22]. Nevertheless clustering is a widely studied data mining problem, clustering real-world data collections may impose new challenges. Real datasets are usually characterized by an *inherent sparseness*

and *variable distribution*, since they are generated by a large variety of events, and *high data dimensionality* because features used to model real objects and human actions may have very large domains. The variability in data distribution grows with data volume, thus increasing the complexity of mining such data. For example, health care data collections can have large volume due to the large cardinality of patient records. Because of the variety of medical treatments usually adopted for the different degrees of severity of a given pathology, patient data collections are also usually characterized by high dimensionality, variable data distribution and inherent sparseness. However, at present, most clustering algorithms perform better with uniform data distribution, while their performance as well as the quality of the extracted knowledge tend to decrease in non-uniform collections.

Aimed at addressing the above issues, this paper presents a *M*ultiple-*L*evel *C*lustering (MLC) framework which comprises two data mining phases. First MLC exploits clustering algorithms in a multiple-level fashion to iteratively focus on different dataset portions and *locally* identify groups of correlated objects. Cohesive and well-separated clusters with diverse data distributions are discovered. Then, the cluster content is concisely described in terms of data features most frequently appearing in the cluster and sequential patterns capturing temporal correlations among data features. Moreover, for supporting the automatic categorization of a new data object into one of the discovered cluster, a classification model is created starting from the cluster set. To allow ubiquitous real-time classification of new data, a two-tier architecture based on a mobile (Android) application has been designed and developed.

Before to apply the clustering analysis, in the MLC framework data are represented in the Vector Space Model (VSM) [29] using the TF-IDF method [22] with the aim of highlighting the relevance of specific data characteristics. In this study, five different multiple-level clustering algorithms have been integrated into MLC, based on K-means (i.e., bisecting and refined K-means [32]), K-medoids (i.e., bisecting and refined K-medoids [18]), and DBSCAN methods (i.e., multiple-level DBSCAN [2]). Clustering results have been then analyzed and compared using some well-established quality indices, as SSE, Silhouette and overall similarity, and Rand Index [22]. Maximal sequential patterns [34] have been selected to concisely describe temporal correlations among data features appearing in each cluster. Decision trees [22] have been used to build the classification model, since they have been shown to provide accurate models in various application domains.

The MLC framework has been validated on three real datasets in the medical care scenario, i.e., underwent examinations by patients, drug prescriptions to patients, Twitter messages on healthcare job information. We considered as a reference case study the former dataset including the examination log data of (anonymized) patients with overt diabetes. Diabetic patients may suffer by various disease complications as eye problems, neuropathy, kidney and cardiovascular diseases. Patients affected by disease complications (or at risk of them) should be tested with more specific examinations in addition to routine tests to

monitor its status (or reveal the pathology). The considered data collection is characterized by an inherently sparse distribution due to the variety of possible examinations, covering both routine tests and more specific examinations for different degrees of severity in diabetes.

The experimental evaluation showed that the multiple-level clustering strategy can effectively partition the initial data collection into cohesive groups, that can be then locally analyzed. Specifically, in the considered use case, interesting clusters containing patients with a similar examination history (with standard or more specific examinations) can be discovered. It also pointed out that, nevertheless both the multiple-level DBSCAN and the refined k-means algorithms generate cluster sets with good quality and agreement, from a medical perspective the multiple-level DBSCAN algorithm appears as the more suitable approach for patient analysis in the considered case study. Maximal sequential patterns characterizing cluster content highlight how examinations are interleaved and distributed over time. The classification performance showed the goodness of the constructed model and its efficiency in classifying new unlabeled data through a mobile application.

This paper is organized as follows. Section 2 describes previous work using clustering techniques in the medical care scenario. Section 3 presents the MLC framework and how the selected algorithms have been tailored to MLC. Section 4 reports the experimental study on real datasets, while Section 5 compares algorithm performance and analyses the results from a medical perspective. Section 6 draws the future developments of the proposed approach.

2 Related work

Clustering algorithms find application in a wide range of different domains, including sensor network data [1], biological data [4], and network traffic data [8]. Clustering algorithms have been also widely used to analyse medical data [9]. Many studies addressed the identification of correlated groups of patients affected by different diseases. For example, [31] reviewed the cluster methods used to diagnose heart valve diseases. In [35], clustering techniques were used to diagnose breast cancer based on tutor features, by recognising hidden patterns of benign and malignant tumors. Authors in [20] exploited the K-means algorithm to cluster a collection of patient records aimed at identifying relevant features of patients subjected to heart attack.

Some research efforts have been devoted to exploiting clustering techniques on data related to diabetic patients [9]. Different issues have been addressed as food analysis [23], gait patterns [30], discovering relationships among diabetes and risk factors [7], analyses of various imputation techniques [25], and discovering similar medical treatments [2]. [25] focuses on diabetes datasets using the K-means algorithm aimed at analysing various imputation techniques. Different from [25], in this work we aim at identifying groups of patients with similar examination histories to provide a preliminar patient categorization into a set

of predefined classes. Thus, we detailed each cluster with sequential patterns to discover how examinations are interlived and distributed over time.

The idea of exploiting a clustering algorithm in a multiple-level fashion was first introduced in [2] and used in [5] to analyze twitter messages. A first study towards a combined distance measure for clustering medical records has been presented in [6]. A parallel effort devoted to clustering documents proved that bisecting K-means was preferable to other clustering methods as standard K-means and hierarchical approaches [32].

The MLC data analysis framework presented in this study enhances the methodology proposed in [2] by providing a more general approach which (i) integrates different clustering algorithms, (ii) uses more indices to evaluate cluster quality, (iii) characterizes temporal aspects of interlived examinations through sequential patterns, (iv) exploits cluster set enriched with domain semantics to train a classification model, and (v) allows ubiquitous classification on new unlabeled examination histories through a mobile application. MLC does not exploit the distance measure proposed in [6] because information on patient profiles (i.e., patient age and gender) are not available on the real data collection discussed as a reference case study. Among the different categories of clustering algorithms, i.e., prototype (e.g., K-means [16], K-medoids [19]), density (e.g., DBSCAN [10]), model (e.g., EM [14]), and hierarchical based methods [22], in this study we focused on the two popular categories of prototype and density based methods for the development of the MLC framework. Furthermore, we integrated in MLC the maximal sequential pattern miner [12] to characterize cluster content and identify how patient examinations are interlived and distributed over time. To ease the exploitation of cluster results, the decision tree (an in [6]), has been integrated in MLC to train a classification model. The latter is then exploited in an Android application to allow ubiquitous patient classification to new unlabeled examination histories.

The wide diffusion of mobile technologies and the increasing capabilities of mobile computing devices caused an increased interest in designing, implementing and testing innovative applications running on mobile devices to provide a wide range of useful services. Im the medical care scenario, some efforts[17, 21, 24] have been devoted on this appealing research. In [17], a distributed end-to-end pervasive healthcare system utilizing neural network computations for diagnosing diabetes was developed in small mobile devices. [21] developed a new mobile-based approach to automatically detect seizures, using k-means as unsupervised classification technique. [24] have presented Generalized Discriminant Analysis and Least Square Support Vector Machine models to diagnose the diabetes disease. Also in this study, we integrated in MLC a two-tier architecture to allows ubiquitous patient classification through a mobile application. The proposed solution allows to efficiently and effectively exploiting knowledge items discovered through MLC cluster analysis to different user profiles (e.g., medical staff, patients). Thus, the proposed mobile application allows ubiquitous patient classification on new unlabeled examination histories.

3 Proposed method

The main components of the MLC framework are depicted in Figure 1. The considered data collection is first prepared for the subsequent analysis phase. The multiple-level clustering strategy is then exploited to discover cohesive groups into data collections with variable data distribution. Clusters are then locally analyzed through sequential patterns to characterize the temporal aspects of data (data distribution over time). The cluster set evaluated with the support of a domain expert is exploited to build a classification model for subsequent classifications of new data objects into one of the discovered groups. To allow ubiquitous classification on new data, real-time analysis is executed on mobile devices through an ad-hoc Android application exploiting the classification model. In this study, we considered as a case study the medical care scenario.

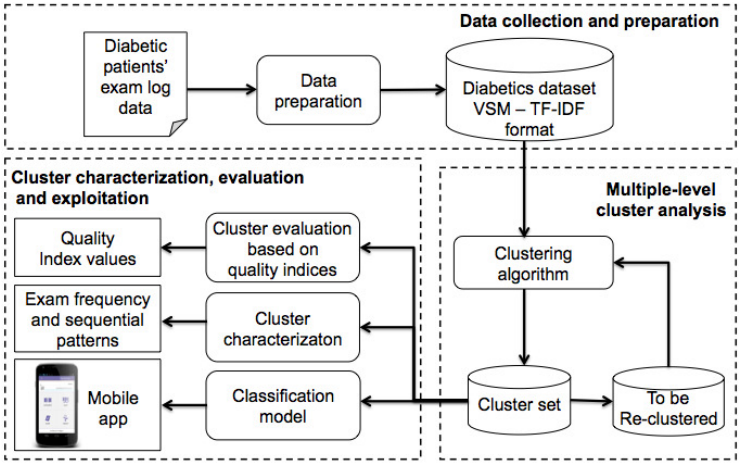

Fig. 1. The MLC **framework**

3.1 Data representation

In the considered collection of patient records, each record corresponds to a medical examination done by a patient in a given date. For instance, Table 1 shows a toy example dataset listing the medical examinations undergone by two patients p_1 and p_2 in year 2014. A more formal definition of a collection of patient records is given in Definition 31.

Definition 31 Collection of patient records. *A collection of patient records $\mathcal{D}$ is a set of records, such that $\Sigma = \{e_1, \ldots, e_k\}$ is the set of examinations in $\mathcal{D}$ and $\Theta = \{p_1, \ldots, p_n\}$ is the set of patients in $\mathcal{D}$. Each record r_k in $\mathcal{D}$ models an examination $e_j \in \Sigma$ done by a patient $p_i \in \Theta$ in a given date.*

Table 1. Example of a collection of patient records

PatientID	Examination	Date	PatientID	Examination	Date
p_1	Glucose level	2014-02-10	p_2	Urine test	2014-12-01
p_2	Fundus oculi	2014-01-06	p_2	Triglycerides	2014-11-30
p_2	Urine test	2014-02-28	p_2	Urine test	2013-04-16
p_1	Fundus oculi	2014-03-10	p_1	Urine test	2014-09-06
p_2	Urine test	2014-04-11	p_2	Triglycerides	2014-08-01
p_1	Glucose level	2014-04-15	p_2	Urine test	2014-07-25
p_2	Electrocardiogram	2014-06-16	p_1	Fundus oculi	2014-07-10
p_1	Glucose level	2014-06-21	p_1	Urine test	2014-11-23

Table 2. VSM representation for dataset in Table 1

PatientID	Glucose level	Fundus oculi	Electrocardiogram	Urine test	Triglycerides
p_1	3	2	0	2	0
p_2	0	1	1	5	2

Table 3. VSM representation using the TF-IDF weighting score for dataset in Table 1

PatientID	Glucose level	Fundus oculi	Electrocardiogram	Urine test	Triglycerides
p_1	0.347	0	0	0	0
p_2	0	0	0.077	0	0.154

To enable the mining process and discover valuable knowledge, in the MLC framework the collection of patient records is tailored to the Vector Space Model (VSM) representation [29] and the Term Frequency (TF) - Inverse Document Frequency (IDF) scheme [22] has been adopted to weight the examination frequency. In this study, we neglect the information on when an examination has been done because we focus on the frequency of performed examinations. The VSM representation has been applied in previous works [29] to represent text documents, while the TF-IDF scheme has been used to weight the relevance of words appearing in the document.

In the VSM representation, each patient p_i is a vector in the examination space. This vector represents the *patient examination history*. The vector cell (p_i, e_j) corresponds to examination e_j done by patient p_i. Cell (p_i, e_j) is a weight describing the relevance of examination e_j for patient p_i. A more formal definition of the patient examination history follows.

Definition 32 Patient examination history. *Let $\mathcal{D}$ be a collection of patient records, $\Sigma = \{e_1, \ldots, e_k\}$ the set of examinations in $\mathcal{D}$ and $\Theta = \{p_1, \ldots, p_n\}$ the set of patients in $\mathcal{D}$. Each patient p_i in $\mathcal{D}$ is represented by a weighted examination frequency vector v_{p_i} of $|\Sigma|$ cells. Each cell $v_{p_i}[j]$ of vector v_{p_i} reports the weighted frequency w_{p_i,e_j} of examination e_j, $e_j \in \Sigma$, for patient p_i, $p_i \in \Theta$. Thus, $v_{p_i} = [w_{p_i,e_1}, \ldots, w_{p_i,e_{|\Sigma|}}]$.*

Table 2 reports a base VSM representation for the example dataset in Table 1. Table 2 has one row for each patient in Table 1, and a number of columns

equal to the number of different examinations in Table 1. Each cell (p_i, e_j) in Table 2 reports the weight of examination e_j for patient p_i. In this base VSM representation the weight is simply given by the number of times examination e_j was repeated by patient p_i. However, a patient data representation as in Table 2 may not properly characterize the patient condition. In fact, it may give more relevance to standard routine tests, which usually appear with higher frequency, than to more specific tests, which often appear with lower frequency. The adoption of the TF-IDF scheme allows highlighting the relevance of specific examinations for a given patient condition. The TF-IDF value increases proportionally to the number of times an examination has been done by the patient, but it is offset by the frequency of the examination in the examination dataset, which helps to control the fact that some examinations are generally more common than others. The definitions of TF and IDF are given below.

Definition 33 Term Frequency (TF) and Inverse Document Frequency (IDF). *Let $\mathcal{D}$ be a collection of patient records, $\Sigma = \{e_1, \ldots, e_k\}$ the set of examinations in $\mathcal{D}$, and $\Theta = \{p_1, \ldots, p_n\}$ the set of patients in $\mathcal{D}$.*

1. *For each pair (p_i,e_j) in $\mathcal{D}$, the Term Frequency TF_{p_i,e_j} is the relative frequency of examination e_j for patient p_i. It is computed as $f_{p_i,e_j} / \sum_{1 \leq k \leq |\Sigma|} f_{p_i,e_k}$, where f_{p_i,e_j} is the number of times patient p_i underwent examination e_j and $\sum_{1 \leq k \leq |\Sigma|} f_{p_i,e_k}$ is the total number of examinations done by p_i.*
2. *The Inverse Document Frequency IDF_{e_j} for examination e_j is the frequency of e_j in $\mathcal{D}$. It is computed as $Log[|\Theta|/|p_k \in \Theta : f_{p_k,e_j} \neq 0|]$ where $|\Theta|$ is the number of patients in $\mathcal{D}$ and $|p_k \in \Theta : f_{p_k,e_j} \neq 0|$ is the number of patients in $\mathcal{D}$ who underwent (at least once) examination e_j.*

Mathematically, the base of the log function for IDF computation in Definition 33 does not matter and constitutes a constant multiplicative factor towards the overall result.

The TF-IDF weight w_{p_i,e_j} for the pair (p_i, e_j) is high when examination e_j appears with high frequency in patient p_i and low frequency in patients in the collection $\mathcal{D}$. When examination e_j appears in more patients, the ratio inside the IDF's log function approaches 1, and the IDF_{e_j} value and TF-IDF weight w_{p_i,e_j} become close to 0. Hence, the approach tends to filter out common examinations. A more formal definition of TF-IDF weight follows.

Definition 34 TF-IDF weight. *For each pair (p_i,e_j) in $\mathcal{D}$, the TF-IDF weight w_{p_i,e_j} is computed as $w_{p_i,e_j} = TF_{p_i,e_j} * IDF_{e_j}$, where TF_{p_i,e_j} is the Term Frequency and IDF_{e_j} is the Inverse Document Frequency.*

Table 3 reports the VSM representation using the TF-IDF scheme for the example dataset in Table 1. The TF-IDF weights for examinations Fondus oculi and Urine Test are equal to 0 since they are performed by both patients. Instead, TF-IDF weights are different than zero for the other examinations, which are performed by only one of the two patients.

3.2 Data clustering using a multiple-level strategy

The MLC framework applies clustering algorithms in a multiple-level fashion to progressively focus on different dataset portions and locally compute clusters. The pseudocode of the multiple-level clustering strategy is in Algorithm 1. It performs multiple runs over the considered data collection. Initially, the whole dataset is analysed. Then, at each subsequent iteration, the clustering algorithm is applied on a selected portion of the dataset, and clusters are locally identified on it. Clustering algorithm parameters can be properly set at each iteration according to the local data distribution of the considered dataset portion. Clusters computed at each iteration contribute to the final cluster set. The approach is iterated until the target objective is achieved, as the minimum threshold value of a given quality index or the maximum allowed number of clusters in the final cluster set.

Data: Initialize $\mathcal{D}$ with the whole initial data object collection
repeat
  if *first iteration* **then**
    select $\mathcal{D}$ as target dataset;
  else
    select a portion of $\mathcal{D}$ as target dataset;
  end
  apply basic clustering algorithm on the target dataset;
  update the final cluster set;
  evaluate the quality of the final cluster set;
until *target objective is verified*;

Algorithm 1: Multiple-level clustering strategy

Clustering algorithms currently integrated in MLC are described in Section 3.2. Data objects in the analysed data collection corresponds to patients in our application scenario. For patient clustering, patient examination histories are compared using the cosine distance measure (see Section 3.2).

Multiple-level clustering algorithms Clustering algorithms integrated in the MLC framework are described in the following. Their main characteristics are summarized in Table 4, by highlighting the improvement with respect to the corresponding (not multiple-level) standard algorithms. Based on this evaluation, they appear as good candidates for the analysis considered in this study. Objects in the analyzed data collection correspond to patients in our application scenario.

Bisecting K-means [32] applies the standard K-means algorithm in a multiple-level fashion. K-means [16] discovers K clusters modeled by their representatives, named *centroids*, given by the mean value of the objects in the clusters. Initially, K objects of the dataset are randomly chosen as centroids. Then, each object

is assigned to the cluster whose centroid is the nearest to that object. Finally, centroids are relocated by computing the mean of the objects within each cluster. The process iterates until centroids do not change or some objective functions are achieved.

Nevertheless K-means is a widely used clustering method, it is biased to spherical clusters and it is sensitive to the initial choice of centroids. Aimed at overcoming this second limitation, the bisecting K-means algorithm adopts a multiple-level clustering approach based on a bisecting strategy. Instead of looking for all representative centroids (and corresponding clusters) at the same time, it iteratively focuses on a dataset portion and locally identifies centroids (and their clusters). More in detail, two clusters are initially generated using the standard K-means algorithm. Then, at each subsequent iteration level, a cluster is selected among those generated up to the current step. The selected cluster is split into two subclusters using K-means. K-1 level iterations are needed for discovering the desired K clusters. Different criteria can be exploited to choose the cluster to split: (i) The cluster size (i.e., the number of objects in the cluster), (ii) the cluster SSE (Sum of Squared Errors), which measures the squared total distances among cluster objects and cluster centroid, and (iii) a criterion based on both cluster size and SSE. In this study, the cluster with the largest SSE value is split.

Bisecting K-medoids [18] relies on the standard K-medoid algorithm (PAM) [19] for implementing a multiple-level clustering technique similar to bisecting K-means. K-medoid works similarly to K-means, but clusters are in this case represented by an object (*medoid*) instead of a mean point (centroid). As for bisecting K-means, bisecting K-medoids is less susceptible to the initialization problems than standard K-medoids. K-medoids methods were also investigated in this study, since they can be less sensitive to outliers than K-means methods.

Refined K-means and refined K-medoids[32]. Both bisecting strategies described above use the standard (K-means and K-medoids) clustering algorithms to bisect individual clusters. It follows that the final cluster set does not represent a local minimum with respect to the total SSE value over the whole cluster set. To deal with this problem, the cluster set generated by bisecting K-means and bisecting K-medoids can be refined as follows. The centroids (resp. medoids) in the computed cluster set are used as the initial centroids (resp. medoids) for the standard K-means (resp. K-medoids) algorithm.

Multiple-Level DBSCAN [2] progressively applies the standard DBSCAN [10] algorithm on different (disjoint) dataset portions. DBSCAN separates dense regions (with a similar density) from a sparse one in the dataset, driven by the user-specified parameters *Eps* and *MinPts*. A dense region in the data space is a n-dimensional sphere with radius *Eps* and containing at least *MinPts* objects. Objects are classified as being (i) in the interior of a dense region (a core point), (ii) on the edge of a dense region (a border point), or (iii) in a sparsely occupied

region (an outlier point). A cluster contains any two core points close within a distance *Eps*, and any border point close within a distance *Eps* to at least one core point in the cluster. Outlier points are filtered out and they are unclustered.

Standard DBSCAN can discover clusters with different sizes and shapes, but it is weak in recognizing clusters with variant density. The multiple-level DBSCAN algorithm allows overcoming this limitation, by decomposing the clustering process into subsequent steps. The whole original dataset is clustered at the first level. Then, at each subsequent level, objects labeled as outliers in the previous level are re-clustered using the standard DBSCAN. With the multiple-level approach, parameters *Eps* and *MinPts* can be set at each level by adapting the definition of dense region to the local data density. Furthermore, the number of unclustered outlier points progressively reduces at each iteration level. Consequently, the multiple-level DBSCAN algorithm can finally provide a more homogenous but also richer cluster set, because it includes a larger portion of the original dataset. The number of iteration levels can be tuned based on the final number of unclustered objects and the number of computed clusters.

Table 4. Comparison of multiple-level clustering algorithms

	Bisecting and Refined K-means	**Bisecting and Refined K-medoids**	**Multiple-level DBSCAN**
Initialization problem	Reduced	Reduced	No
Sensitivity to outliers	Reduced	Reduced	No
Unclustered data objects	No	No	Reduced
Need of convex shape	Yes	Yes	No
Parameter specification	K	K	Eps, MinPts Num. of iterations
Num. of iterations	K-1	K-1	To be specified
Dealing with variable data distribution	Improved	Improved	Improved

Comparing patient examination histories For all clustering algorithms described above, the weighted examination frequency vectors representing the patient examination histories are compared using the cosine distance measure [22]. In our reference case study, let p_i and p_j be two arbitrary patients in the collection $\mathcal{D}$. Let v_{p_i} and v_{p_j} be the corresponding weighted examination frequency vectors. The cosine distance between patients p_i and p_j is computed as

$$dist(p_i, p_j) = \arccos\left(cos(v_{p_i}, v_{p_j})\right) \tag{1}$$

where the cosine similarity between patients p_i and p_j is computed as

$$cos(v_{p_i}, v_{p_j}) = \frac{v_{p_i} \bullet v_{p_j}}{\|v_{p_i}\| \, \|v_{p_j}\|} = \frac{\sum_{1 \le k \le |\Sigma|} v_{p_i}[k] v_{p_j}[k]}{\sqrt{\sum_{1 \le k \le |\Sigma|} v_{p_i}[k]^2} \sqrt{\sum_{1 \le k \le |\Sigma|} v_{p_j}[k]^2}}. \tag{2}$$

The cosine distance in Equation 1 verifies the triangle inequality. The cosine similarity is in the range [0,1]. $cos(v_{p_i}, v_{p_j})$ equal to 1 describes the exact similarity of examination histories for patients p_i and p_j, while $cos(v_{p_i}, v_{p_j})$ equal to 0 points out that patients have complementary histories (i.e., the sets of their examinations are disjoint).

3.3 Cluster evaluation

For the (internal) validation of clustering results, MLC adopts the quality indices typically used for the considered algorithms. The Total SSE index [22] is used for K-means and K-medoids methods, while the Silhouette coefficient [28] for the multiple-level DBSCAN approach. Similar to [32], the overall similarity measure is used to compare cluster sets computed by different algorithms. Finally, the Rand Index [26] has been used to evaluate the agreement between different clustering results.

The **Sum of Squared Error (SSE)** is used to evaluate the cluster cohesion for center-based clusters, as clusters generated using K-means and K-medoids methods [22]. For an arbitrary patient, its error is computed as the squared distance between the patient and the centroid (resp. medoid) in the cluster including the patient. The SSE for a cluster C_i is computed as

$$SSE(C_i) = \sum_{p_j \in C_i} dist(c_i, p_j)^2 \tag{3}$$

where $dist(c_i, p_j)$ is the distance between the centroid (resp. medoid) c_i of cluster C_i and a patient p_j in C_i. The cosine distance metric in Equation 1 has been used for distance evaluation. The smaller the SSE, the better the quality of the cluster. The *Total SSE* on a set of K clusters is computed by summing up the SSE values of the K clusters.

The **Silhouette** index measures both intra-cluster cohesion and inter-cluster separation to evaluate the appropriateness of the assignment of a data object to a cluster rather than to another one [28]. The silhouette value for a given patient p_i in a cluster C is computed as

$$s(p_i) = \frac{b(p_i) - a(p_i)}{max\{a(p_i), b(p_i)\}}, s(p_i) \in [-1, 1], \tag{4}$$

where $a(p_i)$ is the average distance of patient p_i from all other patients in cluster C, and $b(p_i)$ is the smallest of average distances from its neighbour clusters. The silhouette value for cluster C is the average silhouette value on all patients in C. Silhouette values in the range [0.51,0.70] and [0.71,1] show that a reasonable and a strong cluster structure has been found [19]. Lower silhouette values progressively indicate clusters with a weak structure until a no substantial structure. The cosine distance metric in Equation 1 has been used for silhouette evaluation.

The **Overall Similarity** index evaluates the cluster quality. In this study, it has been adopted for comparing the cluster sets from the algorithms integrated into the MLC framework. Specifically, it is used to measure the cluster cohesiveness based on the pairwise cosine similarity of patients in a cluster. For each cluster C, the overall similarity is computed as

$$Overall\_Similarity(C) = \frac{1}{|C|^2} \sum_{\substack{v_{p_i} \in C \\ v_{p_j} \in C}} cos(v_{p_i}, v_{p_j}) \tag{5}$$

where $|C|$ is the cluster size, $cos(v_{p_i}, v_{p_j})$ is the cosine similarity between two patients p_i and p_j in C represented by their weighted examination frequency vectors v_{p_i} and v_{p_j}. The overall similarity on a set of K clusters is computed as the weighted similarity of the clusters

$$Overall\ Similarity = \sum_{i=1}^{K} \frac{|C_i|}{N} Overall\_Similarity(C_i) \tag{6}$$

where N is the total number of patients in the cluster set.

The **Rand Index** computes the number of pairwise agreements between two partitions of a set [26]. It is exploited to measure the similarity between the cluster sets obtained by two different clustering techniques. In our case study, let O be a set of N patients, and X and Y two different partitions of set O to be compared. The Rand Index R is computed as

$$R = \frac{a+b}{\binom{N}{2}} \tag{7}$$

where a denotes the number of pairs of patients in O which are in the same cluster both in X and Y, and b denotes the number of pairs of patients in O which do not belong to the same cluster neither in X nor in Y . Therefore, the term $a + b$ is the number of pair wise agreements of X and Y, while $\binom{N}{2}$ is the number of different pairs of elements which can be extracted from O. The Rand Index ranges from 0 to 1, where 0 indicates that the two partitions do not agree for any patient pair, and 1 that the two partitions are equivalent.

3.4 Cluster content characterization

In the MLC framework, the content of each computed cluster is concisely described as follows. (i) The *most representative examinations* occurring in their patient histories. (ii) The *temporal relationship among examinations* underwent by patients, i.e., which examinations frequently precede or follow other examinations. This information provides a more detailed characterizion of patient histories because the distribution of patient examinations over time is analysed. The two analyses can support a first categorization of the cluster content into a category of patients (possibly) affected by a given diabetes pathology. In fact,

the different pathologies usually require monitoring the patient through some specific examinations.

To support temporal data analysis, the patient data collection contained in each cluster is represented as a *sequence database* [22]. Then, within each cluster, the *sequential patterns of medical examination sets* underwent by patients in subsequent days are analyzed.

Definition 35 Sequence database. *Let $\mathcal{D}$ be a collection of patient records. Let $C \subseteq \mathcal{D}$ be a cluster on $\mathcal{D}$ containing a subset of patient records. The sequence database $\mathcal{D}_S$ defined on C is a collection of sequences p_i:S, where p_i is the patient identifier and S=$< s_1 \ \ldots \ s_n >$ is the temporal list of sets s_t of examinations e_j done by p_i.*

When examinations are done within a short time frame, their temporal order may not be relevant being due to scheduling reasons rather than prescription constraint. For example, in the considered case study of diabetic patients, routine checks through blood tests are usually performed on the same day. Thus, in our data representation, each element s_t in a sequence p_i:S represents the set of examinations done patient p_i on the same day.

The number of elements s_t in a sequence S is the *sequence length*. It corresponds to the number of different days in which the patient has performed at least one examination. The sequence length provides the information on how frequently the patient conditions have been monitored. For example, sequence $p_i : S =<(e_1)(e_2,e_3)(e_4)(e_1)>$ has length equal to 4 because patient p_i performed examinations on 4 different days. The sequence element (e_2 e_3) includes examinations e_2 and e_3 done on the same day.

A sequence S is said to *contain* a sequence S' if S' is a *subsequence* of S, i.e., S' contains a subset of the elements in S and preserves their order. S is called supersequence of S'. For example, sequence $S' =<(e_3)(e_1)>$ is a subsequence of sequence $S =<(e_1)(e_2,e_3)(e_4)(e_1)>$. The *support* (or frequency) on the sequence database $\mathcal{D}_S$ of a sequence S' is the percentage of sequences in $\mathcal{D}_S$ that contain S'. A sequence S' is a *sequential pattern* if its support is above a user-specified minimum support threshold.

Mining the complete set of sequential patterns in all discovered clusters may often provide a too large solution set, making difficult for end-users the comprehension of the results. To overcome this limitation, compact representations of the sequential pattern set have been proposed (as closed sequential patterns and maximal sequential patterns). Among them, maximal sequential patterns [34] have been adopted in this study. The set of maximal sequential patterns is representative since it can be used to recover all sequential patterns, and the exact frequency of these latter can also be computed with a single database pass. Besides, the set of maximal sequential patterns is generally a small subset of the set of (closed) sequential patterns. A sequential pattern S' is said to be a *maximal sequential pattern* if there is no other sequential pattern S'' so that S'' is a superpattern of S' [34]

3.5 Cluster content exploitation

Clusters computed as described in Section 3.2 and characterized as reported in Section 3.4 can be analyzed with the support of a domain expert to describe their content from a medical perspective and assign a representative class label to each of them. Then, to automatically categorize a new patient into one cluster based on his/her examination history, a classification model can be created starting from the discovered cluster set. The possibility of automatically categorize patient histories using the classification model has been made accessible to end-users through a mobile application (app).

Patient classification model Classification is the task of learning a classification model that maps each data object to one of the predefined class labels [22]. A classification model is typically used to assign the class label for a new unlabeled data object. Among various classification methods, decision tree classifiers have been used in this study to characterize the results of the clustering process. *Decision trees* are powerful classification methods that have been widely used in many different application domains. Besides, they provide a readable classification model that can also serve to explain what features characterize objects in each class.

The decision tree is grown in a recursive fashion by iteratively partitioning the training records into successively purer subsets. In the tree structure, each node specifies a test on an attribute, and each branch descending from that node corresponds to one of the possible values for that attribute. Each leaf node represents class labels associated with the instances having, as attribute values, the values appearing in the path reaching the leaf node. Once the decision tree has been created, a new data object is classified by navigating the tree from the root to a leaf node, according to the outcome of the tests along the path.

For the patient representation considered in this work, each node represents one examination undergone by the patient, while each branch descending on a node represents a possible value, or a range of values, for the TF-IDF weight associated with each examination. Decision trees have been previously applied in text mining to classify documents weighted through the TF-IDF scheme [13]. In the analysis, the *Gini index* impurity-based criterion has been considered to split the record set for growing the tree. The Gini index [22] measures how often a randomly chosen instance from the set would be incorrectly labeled if it were randomly labeled according to the distribution of labels in the subset. To evaluate the quality of constructed classification model, we have adopted the three usually metrics, i.e., accuracy, precision and recall [22] (see Section 5.4).

Mobile application This section describes the main functionalities of the proposed mobile applications, while some example screenshots are reported in Figure 2.

New patients can *register* to the application by inserting reference information as their fiscal code and birthdate (Figure 2(a)). The application allows

both *visualizing and updating the list of examinations* underwent by the patient (Figures 2(b) and 2(c)). Any new underwent examination can be inserted by specifying the examination name and code together with the date and time the patient underwent the examination. To enhance usability, the autocomplete feature is used for entering the examination name and code. Moreover, only one of the two fields must be specified, while the other is automatically filled by the application. For the diabetes dataset considered in this study, examination codes have been defined based on the ICD 9-CM (International Classification of Diseases, 9th revision, Clinical Modification) [15].

Through the application, the patient can be *classified* into one of a set of predefined categories based on his/her examination history and a precomputed classification model (Figure 2(d)). Moreover, the application allows *collecting feedbacks and suggestions* from the domain expert on the proposed categorization. Specifically, he/she can either confirm this categorization or suggest an alternative one, by also specifying his/her degree of expertise in the provided feedbacks (Figure 2(d)).

In the medical domain, possible end-users of the application are mainly medical staff and patients to some extent. The application can support the medical staff in the patient evaluation by automatically proposing the patient classification into one out of a set of predefined categories. This automatic categorization can be a valuable support since usually the classification model is computed considering large data collections, and tuned to guarantee an accurate classification. Medical staff still preserves the possibility of proposing an alternative classification based on his/her degree of expertise. The application can also support patients in a self-evaluation of their condition.

The proposed architecture includes mobile devices (e.g., tablet, smartphone) running the application and a server storing the collection of patient examination histories used for creating the classification model. A web server provides functionalities to query this repository and read/insert new data from the application.

To minimize data exchange between the server and the mobile devices at any new classification request, once generated on the server the classification model is downloaded on the mobile devices running the application. Consequently, any new classification request is locally processed by accessing the copy of the classification model stored on the mobile device. On the other hand, this local copy can be periodically updated by downloading the new version of the model generated on the server. More in detail, the classification model based on decision trees is stored on a text file as a list of if-then-else rules.

To allow enriching the central data repository available on the server, new data collected on the mobile device can be transmitted to the central server using the application. New data includes newly registered patients, updated examination histories and feedbacks provided by the domain expert. This enriched data collection can be later used for recomputing the classification model, aimed at increasing the classification accuracy.

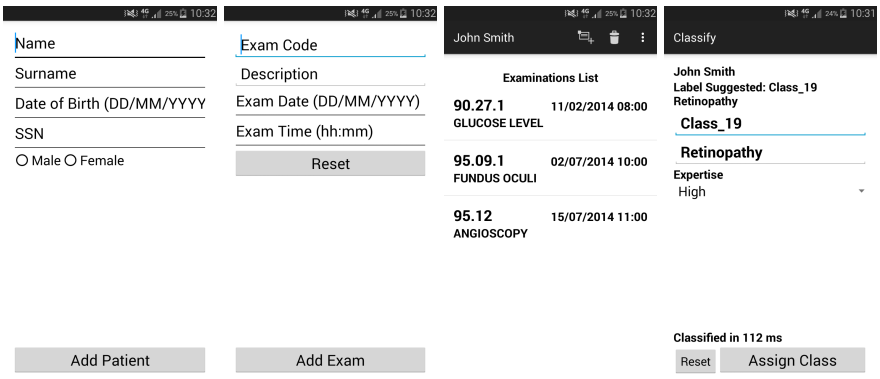

Fig. 2. Mobile app: (a) patient registration, (b) insertion of a new examination done by the patient, (c) visualize patient examination history, (d) patient classification

4 Experimental results

This section presents the results of the experiments with the MLC framework regarding (i) *quality evaluation* for the computed cluster sets, (ii) *execution time* for cluster set computation, and (iii) impact of *data dimensionality*, given by the number of different examinations used to describe patient histories, on the quality of the cluster sets. The MLC methodology has been validated on a real collection of examination log data for diabetic patients.

4.1 Dataset

As a reference case study we considered a real dataset of (anonymized) diabetic patients collected by an Italian Hospital. It contains the examination log data of a set of 6,380 patients with overt diabetes, covering the time period of one year. Both male and female patients in a wide age range are included. The domain of the examinations includes 159 different examination types. Table 5 lists the most frequent examinations including routine examinations as well as more specific diagnostic tests for diabetes complications with varying degrees of severity. Complications due to diabetes can affect for example the cardiovascular system, eyes, and liver. The diagnostic and therapeutic procedures are defined using the ICD 9-CM (International Classification of Diseases, 9th revision, Clinical Modification) [15].

4.2 Evaluation setup and parameter configuration

The MLC framework has been implemented as follows. To perform the multiple-level cluster analysis, the DBSCAN, K-means and K-medoids algorithms avail-

Table 5. Most frequent examinations for each category in the diabetes dataset

Category	*Examination*	*Freq.(%)*	*Category*	*Examination*	*Freq.(%)*
Routine	Glucose level	85	Liver	Alanine aminotransferase enzyme (ALT)	30
	Venous blood	79		Aspartate aminotransferase enzyme (AST)	30
	Capillary blood	75		Gamma GT	15
	Urine test	75		Bilirubin	2
	Glycated hemoglobin	46		Upper abdominal ultrasound	2
	Complete blood count	18	Kidney	Culture urine	25
Cardiovascular	Cholesterol	36		Uric acid	23
	Triglycerides	36		Microscopic urine analysis	23
	HDL Cholesterol	35		Microalbuminuria	21
	Electrocardiogram	23		Creatinine	20
Eye	Fundus oculi	27		Creatinine clearance	16
	Retinal photocoagulation	2	Carotid	ECO doppler carotid	3
	Eye examination	2	Limb	ECO doppler limb	3
	Angioscopy	2		Vibration sense thresholds	1

able in the RapidMiner toolkit have been used, and they have been applied in a multiple-level fashion. RapidMiner is an open-source platform including a number of data mining algorithms [27]. For a more accurate evaluation of the multiple-level strategy, also the standard (not multiple-level) K-means, K-medoids, and DBSCAN algorithms have been considered for performance comparison.

We developed in Java programming language the procedures for transforming the patient examination log data into the corresponding VSM representation using the TF-IDF weighting score, and for cluster evaluation through the SSE, silhouette, and overall similarity measures. Procedures for cluster evaluation have been implemented as a RapidMiner plugin. The procedure for Rand Index computation has been developed in Python programming language.

For K-means and K-medoids methods, experiments have been run by varying the K parameter, corresponding to the number of clusters in the final cluster set. For bisecting algorithms, this set is computed with K-1 iteration levels of the bisecting approach. For refined algorithms, the refinement process has been run for each final cluster set provided by bisecting algorithms. The usual approach has been adopted to address the problem of centroids and medoids initialization for bisecting algorithms, and for standard K-means and K-medoids when considered for performance comparison. Multiple runs, each with set of randomly chosen initial centroids (resp. medoids) have been performed, and then the cluster set with the minimum SSE has been selected. Specifically, RapidMiner parameters maximum number of random initialisations and maximum number of iterations for each initialisation have been set to 50 and 300, respectively, for K-means methods. The same parameters have been set to 10 and 100 (default values in RapidMiner) for K-medoids methods because of their relevant execution time on the considered use case (see Section 4.4).

For the multiple-level DBSCAN, in setting the number of iterations, and the Eps and $MinPts$ values at each iteration level, we aimed at avoiding clusters with few patients, to discover representative examination sets, and at limiting the number of outlier patients, to take into account the contribution of various

examination histories. Clusters should show good cohesion and separation (i.e., silhouette values greater than 0.5). Different Eps and $MinPts$ values have been selected at each iteration level due to the different data distribution of the dataset portion locally analyzed. This portion tends to be progressively sparser because it includes subsets of patients with more and more specific examinations (see Section 5). Consequently, at each subsequent iteration level, smaller $MinPts$ values are progressively selected to define a dense area region. The Eps value has been then locally tuned by trading-off the quality of the cluster set and the number of outlier patients.

Maximal sequential patterns have been extracted using the VMSP algorithm [12] available at [11]. This algorithm has been adopted because it uses a data vertical representation for a depth-first exploration of the search space, that has been shown to be effective in various domains.

To create the classification model, the decision trees algorithm available in the RapidMiner toolkit have been used. The mobile application has been developed on the Android environment version 4.4.

Experiments were performed on a 2.66-GHz Intel(R) Core(TM)2 Quad PC with 8 GBytes of main memory, running linux (kernel 3.2.0).

4.3 Cluster quality evaluation

The quality for the computed cluster sets has been evaluated based on the SSE (for K-means and K-medoids methods), Silhouette (for DBSCAN methods), and overall similarity (for all methods) measures.

Evaluation of K-means methods For all K-means methods, the total SSE measure progressively decreases, and the overall similarity measure progressively increases, when growing the value of K and thus the number of clusters (see Figure 3). The bisecting K-means algorithm always provides the worst results for both measures, i.e., the cluster sets with the highest total SSE and the lowest overall similarity values. Nevertheless, the refined K-means algorithm always provides better results than bisecting K-means, showing that the use in a subsequent clustering phase of the "centroids" computed with the bisecting K-means algorithm can improve the quality of the final cluster set.

Compared to standard K-means, the refined K-means algorithm provides better results when increasing K (about K$>$ 30, i.e., more than 30 clusters). It is worse than standard K-means when a lower value of K is considered ($5 \leq K \leq 15$, i.e., between 5 and 15 clusters). It follows that the final cluster set can benefit from a multiple-level clustering strategy when the number of iteration levels, and thus the final number of clusters, increases. The K parameter can be selected based on the desired number of clusters and the expected quality of the cluster set.

As a reference example, Table 6 reports the main characteristics of the solution with 32 clusters, in terms of number of patients, different examinations, SSE and overall similarity for each cluster.

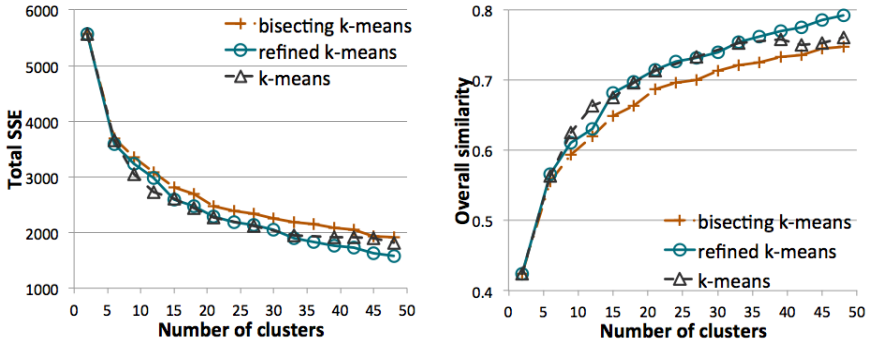

Fig. 3. K-means methods: quality of the cluster set when varying the number of clusters

Table 6. Detailed clustering results for refined K-means

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
Number of patients	96	172	169	97	124	239	233	206	13	88	38	376
Number of examinations	42	39	25	18	52	51	44	40	31	34	38	60
SSE	67.6	112	39.9	8.72	43.3	105	88.7	65.5	5.17	22.4	14.7	134
Overall similarity	0.51	0.50	0.80	0.92	0.70	0.63	0.67	0.72	0.70	0.79	0.67	0.69

	C_{13}	C_{14}	C_{15}	C_{16}	C_{17}	C_{18}	C_{19}	C_{20}	C_{21}	C_{22}	C_{23}	C_{24}
Number of patients	18	78	402	231	351	50	47	26	201	182	226	146
Number of examinations	33	30	56	44	67	28	34	51	37	41	46	54
SSE	7.43	38.3	137	100	149	24.2	17.6	15.7	98.7	48.9	76.3	113
Overall similarity	0.66	0.61	0.70	0.63	0.64	0.60	0.69	0.54	0.59	0.77	0.71	0.45

	C_{25}	C_{26}	C_{27}	C_{28}	C_{29}	C_{30}	C_{31}	C_{32}
Number of patients	74	1,126	509	61	169	170	257	205
Number of examinations	39	35	35	40	20	24	28	30
SSE	58.7	55.3	57	34.2	22.5	65.5	43.9	55.8
Overall similarity	0.43	0.96	0.90	0.57	0.88	0.76	0.85	0.76

Whole cluster set	
Total SSE	1,926.01
Overall similarity	0.75

Evaluation of K-medoids methods The experimental results reported in Figure 4 show that K-medoids methods exhibit a similar behavior to K-means ones. The bisecting K-medoids algorithm always provides the worst results in terms of overall similarity and total SSE values. The refined K-medoids algorithm always improves bisecting K-medoids and provides comparable results to standard K-medoids.

K-medoids methods showed a very high computational cost which limited their applicability in the MLC framework (see Section 4.4). Due to this cost, solution sets with a larger number of clusters have not been generated.

Evaluation of DBSCAN methods As reported in Table 7, when iterating the multiple-level DBSCAN approach for four levels, 32 clusters are computed in total showing good overall similarity and silhouette values (greater than 0.5). These

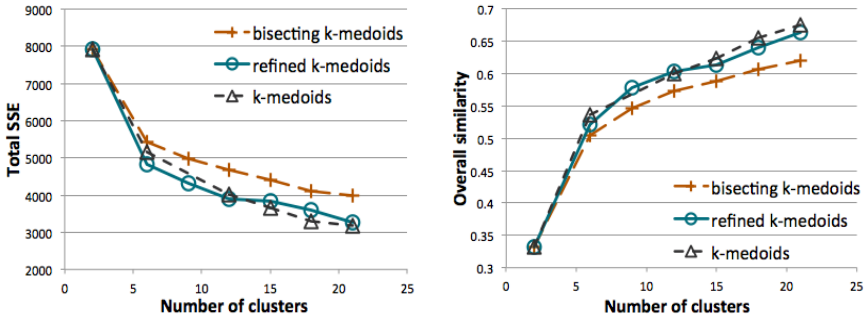

Fig. 4. K-medoids methods: quality of the cluster set when varying the number of clusters

clusters globally includes 3,510 patients (about 55% of the diabetes dataset). Most patients belong to clusters computed at the first level, while a comparable number of patients is included in clusters computed at the next levels. After four iterations, 2,870 patients are labeled as outliers and remain unclustered. Note that these patients can be additionally clustered by iterating the approach for more levels.

Clustering about 55% of the patients using the standard DBSCAN algorithm generates a lower quality cluster set than when using the multiple-level DBSCAN approach. To deepen into the analysis of this point, Figure 5 plots the silhouette and overall similarity values, and number of outlier patients, when the whole patient collection is analyzed using the standard DBSCAN. With parameters Eps=0.36 and $MinPts$=30, a cluster set is generated including almost the same number of patients than the cluster set from the multiple-level DBSCAN approach, but with a significantly lower quality. The overall similarity value is 0.73 and the silhouette is 0.4 (i.e., lower than 0.5), while these values are 0.85 and 0.55, respectively, for the multiple-level DBSCAN when iterated for four levels (see Table 7). It follows that, also for the DBSCAN method, the final cluster set can benefit of the multiple-level strategy.

Clusters computed with four level iterations are described in Table 8 in terms of their number of patients, different examinations and overall similarity value, while silhouette plot is reported in Figure 6. Clusters mainly show a rather prominent silhouette. Few patients have negative silhouette values in clusters computed at the first level, i.e., 198 patients out of 1,764 in cluster C_{1_1}, 2 patients out of 223 in C_{2_1} and 7 patients out of 294 in C_{4_1}. At the fourth level, cluster C_{3_4} shows a less prominent silhouette, but the average silhouette value is almost 0.5.

Table 7. Clustering results for multiple-level DBSCAN

	1st level	2nd level	3rd level	4th level
(MinPts, Eps)	(30, 0.3)	(30, 0.5)	(20, 0.5)	(10, 0.35)
Number of clusters	11	5	4	12
Number of patients	2,872	260	104	274
Silhouette	0.54	0.61	0.66	0.6
Overall similarity	0.85	0.86	0.89	0.94

Whole cluster set	
Number of clusters	32
Number of clustered patients	3,510
Number of outliers	2,870
Silhouette	0.55
Overall similarity	0.86

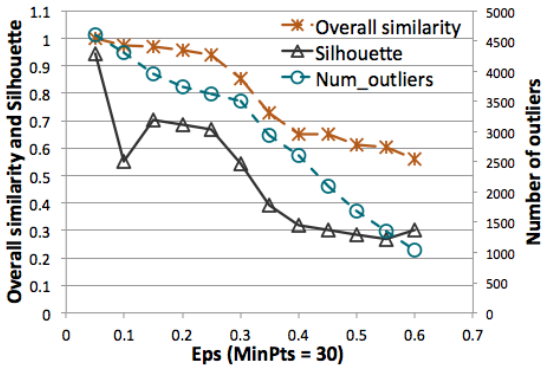

Fig. 5. DBSCAN algorithm: quality of the cluster set and number of outlier patients when varying the *Eps* value ($MinPts$=30)

4.4 Execution time

For the multiple-level DBSCAN algorithm, the total run time for computing a solution with 32 clusters is 13min 40s. The first, second, third and fourth iteration level require 3min 34s, 3min 8s, 3min, and 2min 58s, respectively. The time tends to progressively reduce at each level because a smaller dataset portion is progressively analysed.

The run time for bisecting and refined K-means algorithms for computing a solution with 32 clusters is (slightly) lower than for the multiple-level DBSCAN approach. Bisecting k-means requires 10 min, while refined K-means requires 7s in addition for the refinement of centroids (i.e., to run K-means after having initialized centroids). The time for K-means is about 2 minutes.

The run time is significantly higher for bisecting K-medoids, making the approach not suitable for datasets with many examinations as the one considered in this study. The time is approximately 38 hours for generating a set of 20 clusters, while refined K-medoids requires 34 min in addition for the refinement of medoids. The time for K-medoids is about 5 hours and a half.

Table 8. Detailed clustering results for multiple-level DBSCAN

	First-level										
	C_{1_1}	C_{2_1}	C_{3_1}	C_{4_1}	C_{5_1}	C_{6_1}	C_{7_1}	C_{8_1}	C_{9_1}	C_{10_1}	C_{11_1}
Number of patients	1,764	223	140	294	144	110	42	43	35	36	41
Number of examinations	10	6	8	7	6	2	7	8	9	19	2
Overall similarity	0.82	0.87	0.94	0.88	0.92	1.00	0.96	0.94	0.97	0.94	1.00

	Second-level					Third-level			
	C_{1_2}	C_{2_2}	C_{3_2}	C_{4_2}	C_{5_2}	C_{1_3}	C_{2_3}	C_{3_3}	C_{4_3}
Number of patients	75	73	49	30	33	32	29	21	22
Number of examinations	35	27	15	16	8	22	19	14	15
Overall similarity	0.84	0.85	0.91	0.89	0.86	0.9	0.83	0.92	0.91

	Fourth-level											
	C_{1_4}	C_{2_4}	C_{3_4}	C_{4_4}	C_{5_4}	C_{6_4}	C_{7_4}	C_{8_4}	C_{9_4}	C_{10_4}	C_{11_4}	C_{12_4}
Number of patients	19	19	100	12	14	14	24	30	10	10	12	10
Number of examinations	7	3	20	9	8	19	12	12	9	16	4	19
Overall similarity	0.93	1	0.91	0.98	0.95	0.94	0.94	0.92	0.94	0.94	0.99	0.95

Whole cluster set	
Overall similarity	0.86

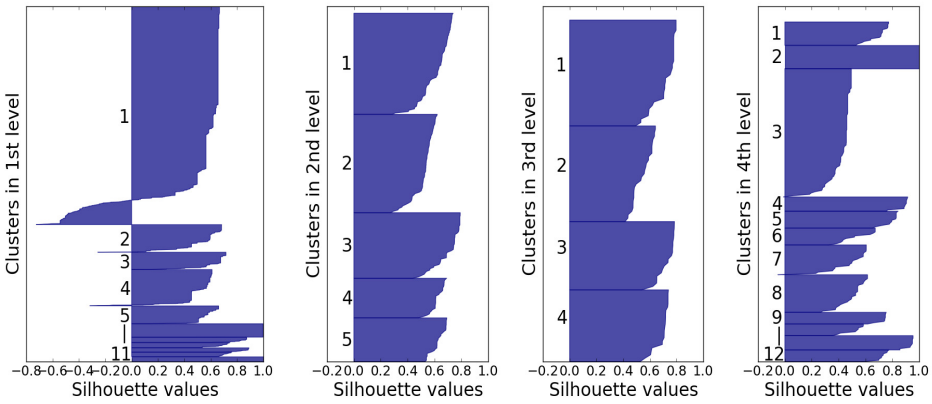

Fig. 6. Silhouette plot for multiple-level DBSCAN

4.5 Impact of data dimensionality on cluster sets

In the patient data representation considered in this study, the data dimensionality is given by the set of examinations describing the patient examination history. When the cardinality of this set increases, a larger set of facets characterizes patient care plans. Besides routine tests, also more specific examinations are considered, which are progressively undergone by a reduced number of patients. Consequently, the patient distribution tends to become increasingly sparser, and the computation of cohesive clusters becomes more complex.

To evaluate how data dimensionality impacts on the quality of the cluster set, in addition to the whole diabetes dataset (with 159 examinations), two other configurations of this dataset have been considered, including about 60%

and 40% of the most frequent examinations (i.e., 60 and 30 examinations, respectively). The three datasets contain the same number of patients, showing that patient histories include various examinations, possibly repeated a different number of times by each patient. The multiple-level DBSCAN and the refined K-means algorithms have been considered as reference example methods for this analysis.

For refined K-means, given a number of clusters, the overall similarity value decreases, and the total SSE increases, as the number of examinations (and thus the dataset sparsity) increases (see Figure 7). Consequently, when the number of examinations increases, a larger number of clusters should be generated to discover cohesive groups of patients. For example, the overall similarity value gradually tends to 0.8 when considering 20 clusters for dataset with 30 examinations and 40 clusters for datasets with 60 and 159 examinations.

The multiple-level DBSCAN has been iterated for four levels for all three datasets, aimed at generating cluster sets with comparable good quality in terms of overall similarity and silhouette values. As the number of examinations increases (and thus the dataset sparsity), the final number of patients labeled as outliers, and thus unclustered, decreases. After four iterations, the final number of outliers is 2,573, 2,678 and 2,870 for datasets with 30, 60, and 159 examinations, respectively (see Tables 7 and 9). It follows that when the dataset sparsity increases, more iterations are needed to cluster a larger subset of patients but preserving the quality of the cluster set.

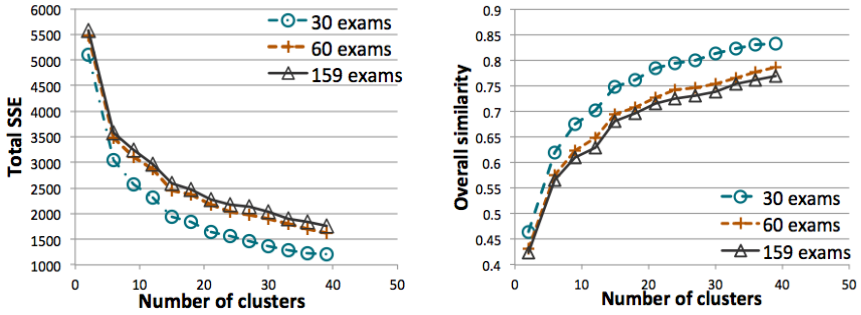

Fig. 7. Refined K-means on the three datasets: quality of the cluster set when varying the number of clusters

4.6 Evaluation on additional datasets

The multiple-level clustering strategy has been further evaluated using two additional real datasets in the medical area. DBSCAN and K-means methods have

Table 9. Clustering results for multiple-level DBSCAN on datasets with 30 and 60 examinations

	30 examinations				60 examinations			
	1st level	2nd level	3rd level	4th level	1st level	2nd level	3rd level	4th level
(MinPts, Eps)	(50, 0.3)	(20, 0.45)	(10, 0.4)	(15, 0.25)	(30, 0.3)	(30, 0.55)	(15, 0.25)	(10, 0.6)
Number of clusters	6	12	7	10	11	6	10	14
Number of patients	2,837	617	147	206	2,891	358	186	267
Silhouette	0.56	0.60	0.72	0.64	0.54	0.54	0.70	0.6
Overall similarity	0.84	0.89	0.90	0.98	0.85	0.83	0.99	0.65
Whole cluster set								
Number of clusters	35				41			
Number of clustered patients	3,807				3,702			
Number of outliers	2,573				2,678			
Silhouette	0.57				0.55			
Overall similarity	0.86				0.84			

been considered as reference algorithms for this analysis. The former dataset contains the list of *drugs* prescribed to 3,500 (anonymized) diabetic patients, with 103 distinct drugs encoded at the fourth level of the standard pharmaceutical coding system adopted by the Anatomical Therapeutic Chemical (ATC) Classification System [3]. The latter dataset contains 6484 *tweets* on healthcare job information posted in the New York area from July 1st to July 22nd in 2014. Tweets have been retrieved from twitter.com via Twitter's Streamming Application Programming Interfaces (APIs) using a java crawler. Tweets were written in english and they have been preprocessed by removing stop words and other elements as numbers and usernames. Both datasets have been tailored to the VSM representation using the TF-IDF scheme to weight the relevance of drugs/words.

The experimental evaluation on the two additional datasets show similar performance to the reference case study on patient examinations. K-means and refined K-means algorithms provide (almost) comparable performance (in terms of SSE and overall similarity), and they both outperform bisecting K-means. Multiple-level DBSCAN tends to produce the most cohesive clusters. In the following the performance of refined K-means and multiple-level DBSCAN are compared considering the solution with the same number of clusters.

On the drug dataset, after four level iterations, the multiple-level DBSCAN generates 34 clusters globally including about 48.83% of the original dataset (1709 patients) and with average silhouette 0.62. The overall similarity is 0.86 and on the average clusters contain 31 different drugs. When 34 clusters are computed with refined K-means, the overall similarity is 0.62 and on average a larger number of different drugs (55) are included in clusters.

The tweet dataset is characterized by a higher sparsity due to the large number of words and the limited length of tweet messages (140 characters). The multiple-level DBSCAN iterated for four levels generates a 53 cluster set containing 34.46% of the collection (2300 tweets). The average silhouette is 0.59. The overall similarity and the average number of different words in clusters are

0.8 and 16 for multiple-level DBSCAN and 0.44 and 40 for refined K-means, respectively (considering 53 clusters for both approaches).

The execution time is about 66 min and 136 min for multiple-level DBSCAN on drugs and tweets dataset respectively, and 133 sec and 16 min 37 sec for refined K-means.

5 Discussion

Here we discuss the clustering results discovered through the MLC framework. The discussion addresses the performance comparison for clustering methods, the comparison from a medical perspective for discovered cluster sets, and the cluster characterization in terms of association rules.

5.1 Performance comparison

Concerning *K-means methods*, *refined K-means* in particular benefits of the multiple-level strategy. The quality of the final cluster set is at least comparable to the cluster quality of standard and bisecting K-means algorithms, but it outperforms them when the approach is iterated for more levels. Also the *multiple-level DBSCAN* algorithm pointed out the improvement in adopting a multiple-level strategy with respect to the standard DBSCAN in the considered case study. On the contrary, *K-medoids methods* do not seem suitable to be used in a multiple-level fashion in our case study, because they provide cluster sets with lower quality. For example, for the solution with 21 clusters, the overall similarity is 0.67 and total SSE is 3,200 for K-medoids methods (see Figure 4), while these measures are 0.71 and 2,275 for K-means methods (see Figure 3). In addition, the high computational time of K-medoids methods limits the possibility of iterating them for more levels, thus progressively improving cluster quality.

Based on the discussion above, we focused our attention on comparing the refined K-means and the multiple-level DBSCAN algorithms. Let us consider, as a reference example, the solutions with 32 clusters generated by the two algorithms on the whole dataset with 159 examinations. The following considerations hold. (i) Both *cluster sets exhibit good quality* in terms of overall similarity, even if this value is higher for multiple-level DBSCAN (0.86, see Table 7) than for refined K-means (0.75, see Figure 3). (ii) In both cases, the clustering process requires a *comparable and acceptable execution time*, slightly lower for refined K-means (about 10min) than for multiple-level DBSCAN (about 13min). Thus, (iii) in both cases the multiple-level strategy can be potentially *iterated for more levels* by further increasing the quality of the final cluster set. Specifically, the unclustered outlier patients can be progressively reduced for multiple-level DBSCAN, while clusters can be split into more cohesive subclusters for refined K-means.

To deepen into the comparison of the two algorithms, the agreement between the two cluster sets is evaluated using the Rand Index. While refined K-means clusters the whole dataset, the multiple-level DBSCAN clusters a subset, since

outlier patients are grouped into a separate cluster. The following two options are considered to guarantee the same number of patients in the compared cluster sets. The separate cluster of outlier patients is (a) excluded from, or (b) it is included in, the final cluster set generated by the multiple-level DBSCAN algorithm. In case (a), the outlier patients are also removed from clusters computed by the refined K-means algorithm. The Rand Index value shows a good agreement between the two clustering results, higher in option (a) (Rand Index = 0.83) than in option (b) (Rand Index = 0.73). It follows that the two cluster sets mainly differ on the patients labeled as outliers. While they are isolated by multiple-level DBSCAN, they are clustered together with other patients by refined K-means.

5.2 Comparison from a medical perspective

Discovered cluster sets are also analysed from a medical perspective. Following the discussion on performance comparison in Section 5.1, we focused on the multiple-level DBSCAN and the refined k-means algorithms, and we analysed and compared the solutions with 32 clusters computed on the whole dataset with 159 examinations.

Nevertheless the two algorithms generate cluster sets with good quality and agreement, from a medical perspective the *multiple-level DBSCAN* appears as the *more suitable approach* for patient analysis. The refined K-means algorithm is less effective in partitioning the initial data collection into subsets with different data distributions, i.e., including patients with (significantly) different examination histories. Instead, the multiple-level BSCAN algorithm isolates these outlier patients, and separately analyzes them in a subsequent clustering phase. Since refined K-means computes a cluster set including *all* the patients in the original dataset, these outlier patients are always assigned to some clusters, thus increasing the variety of examinations in each cluster.

More in detail, unlike refined K-means, the multiple-level DBSCAN approach computed clusters including, on average, a limited number of different examinations. These clusters contain from 2 to 35 different examinations and about 12 on average (see Table 8), while clusters from refined K-means include from 18 to 67 different examinations and about 38 on average (see Table 6). In addition, clusters from refined K-means mostly contain patients with diversified examination histories, including both routine and more specialized examinations to test different diabetes complications. Instead, in clusters from multiple-level DBSCAN, the number of examinations tend to increase with the iteration levels, thus progressively including more specialized examinations.

For both methods, the content of some example clusters, in terms of the most frequent examinations in the cluster, is reported in Table 10. For the multiple-level DBSCAN, first-level clusters contain patients who mostly performed standard routine tests to monitor diabetes conditions (cluster C_{2_1}). Second-level clusters contain patients tested with an increasing number of specific examinations, showing that patients can be affected by a particular disease complication or by more disease complications (e.g., on cardiovascular and eye system in

cluster C_{5_2}). Examinations become progressively more numerous and specific in third- and fourth-level clusters, indicating patients that can have diabetes complications of increasing severity (clusters C_{1_3} and C_{12_4}). Instead, in clusters from refined K-means, examinations cover most categories. Thus, patients with different disease complications can be included in the same cluster (clusters C_2, C_5, C_{11} and C_{21}).

Table 10. Multiple-level DBSCAN and refined K-means: most frequent examinations in some example clusters (examination frequencies are in %)

		Multiple-level DBSCAN				Refined K-means			
Category	*Examination*	1st level	2nd level	3rd level	4th level	C_2	C_5	C_{11}	C_{21}
		C_{2_1}	C_{5_2}	C_{1_3}	C_{12_4}				
Routine	Glucose level	78	100	75	100	68	94	63	90
	Capillary blood	72	97	72	100	58	69	61	57
	Urine test	72	100	72	100	60	68	61	55
	Venous blood	96	91	69	70	56	98	68	96
	Glycated Hemoglobin	100	76	16	10	24	90	40	79
	Complete Blood Count	-	-	-	-	5	73	16	100
Cardiovascular	Cholesterol	-	-	13	10	10	85	37	70
	Triglycerides	-	-	13	1	11	84	37	69
	HDL Cholesterol	-	-	13	10	10	84	37	67
	Electrocardiogram	-	79	25	-	20	25	26	15
Eye	Fundus oculi	-	100	-	20	26	34	45	20
	Retinal photocoagulation	-	-	-	-	-	1	3	-
	Eye examination	-	-	-	-	1	7	8	1
	Angioscopy	-	-	100	-	-	2	8	-
Liver	ALT	-	-	-	10	9	95	26	50
	AST	-	-	-	10	10	97	29	49
	Gamma GT	-	-	-	10	5	83	18	10
	Bilirubin	-	-	-	-	-	95	-	-
	Upper abdominal ultrasound	-	-	-	-	1	6	3	2
Kidney	Culture urine	-	-	-	-	7	52	37	20
	Uric acid	-	-	-	10	6	65	21	33
	Microscopic urine analysis	-	-	-	10	4	69	13	50
	Microalbuminuria	-	-	-	-	6	44	26	11
	Creatinine	-	-	-	-	4	61	13	29
	Creatinine clearance	-	-	-	10	6	29	18	11
Carotid	ECO doppler carotid	-	-	-	-	67	4	11	2
Limb	ECO doppler limb	-	-	-	10	53	2	16	2
	Vibration sense thresholds	-	-	-	100	-	2	-	2

Being clusters computed using the multiple-level DBSCAN algorithm rather homogeneous in their patient examination histories, clinical domain experts can inspect the cluster content from a medical perspective to support various analysis as for example those reported below. (a) Discover, for each cluster, the examinations actually prescribed to diabetic patients included in the cluster. (b) Check the coherence between the underwent examinations in each cluster and the existing medical guidelines for diabetes disease [15]. (c) Provide feedbacks to health care organizations to improve the application of the existing medical guidelines, but also to enrich these guidelines or assess new ones.

5.3 Cluster characterization based on sequence pattern analysis

To analyse the temporal order of examinations when testing patients, the cluster content has been described using sequence patterns. The average length of sequences describing patient histories increases from each subsequent level of clustering. The sequence length represents the number of different days in which patients had at least one examination. It corresponds to the frequency used to monitor patients within the time period of one year covered in the considered dataset. The average sequence length is lower for patients in clusters at the first level (about 2.4), including patients mainly monitored through periodic routine tests. Instead, it increases in the next levels being patients tested with more specific examinations to check diabetes complications in addition to routine tests (about 4.3 in the second level and 3.6 in the third and fourth level).

As an example of the type of information that can be mined, some maximal sequential patterns are reported in Table 11 for the multiple-level DBSCAN clusters in Table 10. Sequences S_1 and S_2 in the first-level cluster C_{2_1} mainly show the periodic repetitions of the routine examinations used to monitor patient conditions. Routine blood examinations are usually performed on the same day, possibly together with urine test. In next level clusters, sequences include routine examinations interleaved with more specific examinations to test diabetes at different degrees of severity. Sequences tend to be progressively characterized by lower support values, being patient histories more diversified.

In the second level cluster C_{5_2}, sequence S_3 show that the eye examination fundus oculi is followed by repetition of routine tests. In third level cluster C_{1_3}, the angioscopy examination preceded and/or follows routine tests (sequence S_5) and/or more specific examinations to monitor possible cardiovascular complications and cholesterol concentration (sequence S_6). The angioscopy examination is an eye examination allowing a deeper analysis of patient eye condition (than other examination as fundus oculi) and it is typically underwent by patients with possible retinopathy.

5.4 Patient classification results

The clustering results were also evaluated by a domain expert to describe the cluster content from a medical perspective and assign a class label to each cluster. For example, considering clusters in Table 10, patients in cluster C_{5_2} may be affected by retinopathy, while patients in cluster C_{2_1} are (probably) not affected by diabetes complications. Then, a classification model based on decision trees was built starting from the cluster set computed with the multiple-level DBSCAN approach when iterated for four levels summarized in Table 7 and detailed in Table 8. To preserve the characteristics of the discovered clusters, where patients with similar examination histories have been grouped together, each cluster has been labeled with a different class label.

The 7-fold cross validation method has been adopted for evaluating the classification model based on accuracy, precision and recall measures. The *accuracy*,

Table 11. Example of maximal sequential patterns for some clusters from multiple-level DBSCAN

Cluster	Maximal sequential patterns	Sup.(%)
C_{2_1}	S_1: < (Venous blood,Glycated Hemoglobin)(Glucose level,Capillary blood, Urine test,Venous blood)(Glucose level,Capillary blood,Urine test,Venous blood) >	14.8
	S_2: < (Venous blood,Glycated Hemoglobin)(Glucose level,Venous blood, Glycated Hemoglobin)(Glucose level,Venous blood) >	5.38
C_{5_2}	S_3: < (Fundus oculi)(Glucose level,Capillary blood,Urine test,Venous blood, Glycated Hemoglobin)(Glucose level,Capillary blood,Urine test,Venous blood)>	18.18
C_{1_3}	S_4: <(Angioscopy)(Triglycerides,Cholesterol,Glycated Hemoglobin,HDL Cholesterol) (Glucose level,Capillary blood,Urine test,Venous blood) >	6.25
	S_5: <(Glucose level,Capillary blood,Urine test,Venous blood)(Electrocardiogram) (Angioscopy) >	9.38
C_{12_4}	S_6: <(Capillary blood,Urine test,Glucose level) (Capillary blood,Vibration sense thresholds,Glucose level,Urine test) (Capillary blood,Urine test,Glucose level)(Glucose level,Urine test,Capillary blood) (Venous blood,Glucose level,Capillary blood,Urine test)>	30

measuring the overall quality of the classifier, is the ratio of the number of correctly classified patients over the total number of given patients. Precision and recall analyse the performance of the classifier with respect to a given class c. *Precision* is the number of patients correctly classified in c divided by the number of patients classified in c. *Recall* is the number of patients correctly classified in c divided by the number of patients labeled with c in the collection. The experimental result showed the goodness of the constructed model. The accuracy value is about 97.3%. The average recall value is around 88%, except for clusters C_{6_4} (78.6%), C_{1_3} (71.9%) and C_{2_4} (47.4%), and the precision value has an average of 91%, apart from clusters C_{2_4} (69.2%) and C_{10_4} (72.7%). The values are all very high, which guarantees the quality of the classification model.

The final decision tree contains 146 nodes, 74 paths with average length 9.53, and leaf nodes with quite good degree of purity. The creation time was about 30 sec on a 2.66-GHz Intel(R) Core(TM)2 Quad PC with 8 GBytes of main memory, running linux (kernel 3.2.0). For locally accessing the classification model on the mobile device, the decision tree was transformed into the corresponding textual representation as an ordered list of if-the-else rules. This text file has size 16 Kbytes. The classification time of a new patient is about 140 milliseconds using as mobile device a Samsung Galaxy A5 smartphone running Android 4.4.4 based on 1.2 GHz Quad Core Qualcomm Snapdragon 410 Cortex-A53 processor with 2 GB RAM.

6 Conclusion

This paper presented a two-phase data mining methodology to effectively analyze real data collection with variable data distribution. Discovering useful knowledge from such collections is a complex task due to the inherent data sparseness. The proposed multiple-level clustering strategy can perform a valuable preprocessing

step for partitioning the data collection into cohesive groups, that are then locally analyzed. The mobile application allows a real-time classification of new data objects into one of a the above groups, also collecting domain-expert feedbacks on the proposed classifications.

In our reference case study, we focus on one facet of the patient treatment given by the underwent examinations. Groups of patients with similar examination histories, and thus possibly similar degree of disease severity, are discovered. Cluster content analyses, as frequent examinations and their temporal distribution, can provide useful information about how patient conditions are actually monitored. It can be used to check coherence with medical guidelines or revels when patients tend to be over/under monitored, and it can be also later enriched with other facets of the patient treatment. The mobile application providing the automatic patient classification into one category, represents a valuable support for medical staff that can also propose an alternative classification.

Some limitations of the MLC framework as proposed in this study can represent interesting future research directions to enhance MLC. In the multiple-level clustering strategy, the number of iteration levels should be currently experimentally tuned by trading-off the computed and the expected quality of the cluster set. Preliminary *studies on the data distribution* to identify the diverse degree of data sparseness can support in selecting this parameter. Nevertheless the MLC ability in discovering cohesive clusters, high data sparseness can enforce, at some iteration levels, clusters with a limited size. To cope with this issue, *data taxonomies* can be locally exploited for reducing data sparseness by climbing the abstraction level used for data representation. Moreover, alternative patterns as for example *weighted sequential patterns* [33] can be adopted to characterize the cluster content.

Additional research directions concern devising proper strategies to exploit classifications suggested by domain-expert through the mobile application. They can be used to modify label assignment in the initial data collection aimed at *increasing the quality of the knowledge base* used to build the classification model. From the technological perspective, to deal with huge data collections, a future activity can address the deployment of the proposed framework in a *cloud-based platform*, as Apache Mahout or Spark.

Bibliography

[1] Abbasi, A. A., & Younis, M. (2007). A survey on clustering algorithms for wireless sensor networks. *Comput. Commun.*, *30*, 2826–2841.

[2] Antonelli, D., Baralis, E., Bruno, G., Cerquitelli, T., Chiusano, S., & Mahoto, N. A. (2013). Analysis of diabetic patients through their examination history. *Expert Syst. Appl.*, *40*, 4672–4678.

[3] ATC (2013). Norwegian-institute-of-public-health: Atc/ddd index 2013. available: http://www.whocc.no/atc_ddd_index/ . last access on november 2013.

[4] Au, W., Chan, K. C. C., Wong, A. K. C., & Wang, Y. (2007). Correction to "attribute clustering for grouping, selection, and classification of gene expression data". *IEEE/ACM Trans. Comput. Biology Bioinform.*, *4*, 157.

[5] Baralis, E., Cerquitelli, T., Chiusano, S., Grimaudo, L., & Xiao, X. (2013). Analysis of twitter data using a multiple-level clustering strategy. In *MEDI* (pp. 13–24).

[6] Bruno, G., Cerquitelli, T., Chiusano, S., & Xiao, X. (2014). A clustering-based approach to analyse examinations for diabetic patients. In *2014 IEEE International Conference on Healthcare Informatics, ICHI 2014, Verona, Italy, September 15-17, 2014* (pp. 45–50).

[7] Chaturvedi, K. (2003). Geographic concentrations of diabetes prevalence clusters in texas and their relationship to age and obesity. *http://www.ucgis.org/summer03/studentpapers/kshitijchaturvedi.pdf. Retrieved*, *9*, 2010.

[8] Eriksson, B., Barford, P., & Nowak, R. D. (2008). Network discovery from passive measurements. In *Proceedings of the ACM SIGCOMM 2008 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, Seattle, WA, USA, August 17-22, 2008* (pp. 291–302).

[9] Esfandiari, N., Babavalian, M. R., Moghadam, A.-M. E., & Tabar, V. K. (2014). Knowledge discovery in medicine: Current issue and future trend. *Expert Systems with Applications*, *35*, 4434–4463.

[10] Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Knowledge Discovery and Data Mining (KDD)* (pp. 226–231).

[11] Fournier-Viger, P., Gomariz, A., Gueniche, T., Soltani, A., Wu, C.-W., & Tseng, V. S. (2014). Spmf: a java open-source pattern mining library. *The Journal of Machine Learning Research*, *15*, 3389–3393.

[12] Fournier-Viger, P., Wu, C.-W., Gomariz, A., & Tseng, V. S. (2014). Vmsp: Efficient vertical mining of maximal sequential patterns. In *Advances in Artificial Intelligence* (pp. 83–94). Springer.

[13] Friedl, M. A., & Brodley, C. E. (1997). Decision tree classification of land cover from remotely sensed data. *Remote sensing of environment*, .

[14] G. McLachlan and T. Krishnan (1997). *The EM algorithm and extensions*. Wiley series in probability and statistics. John Wiley and Sons.

[15] ICD-9-CM, I. (2011). International Classification of Diseases, 9th revision, Clinical Modification. Available: http://icd9cm.chrisendres.com. Last access on March 2011, .

[16] Juang, B.-H., & Rabiner, L. (1990). The segmental k-means algorithm for estimating parameters of hidden markov models. *IEEE Transactions on Acoustics, Speech and Signal Processing*, *38*, 1639–1641.

[17] Karan, O., Bayraktar, C., Gümüşkaya, H., & Karlk, B. (2012). Diagnosing diabetes using neural networks on small mobile devices. *Expert Systems with Applications*, *39*, 54 – 60.

[18] Kashef, R., & Kamel, M. S. (2008). Efficient bisecting k-medoids and its application in gene expression analysis. In *Image Analysis and Recognition* (pp. 423–434). Springer.

[19] Kaufman, L. and Rousseeuw, P. J. (1990). *Finding groups in data: An introduction to cluster analysis*. Wiley.

[20] Khaing, H. W. (March 2011). Data mining based fragmentation and prediction of medical data. In *Int. Conf. Computer Research and Development (ICCRD)* (pp. 480–485).

[21] Menshawy, M. E., Benharref, A., & Serhani, M. (2015). An automatic mobile-health based approach for {EEG} epileptic seizures detection. *Expert Systems with Applications*, *42*, 7157 – 7174.

[22] Pang-Ning T. and Steinbach M. and Kumar V. (2006). *Introduction to Data Mining*. Addison-Wesley.

[23] Phanich, M., Pholkul, P., & Phimoltares, S. (2010). Food recommendation system using clustering analysis for diabetic patients. In *IEEE International Conference on Information Science and Applications (ICISA)* (pp. 1–8).

[24] Polat, K., Güneş, S., & Arslan, A. (2008). A cascade learning system for classification of diabetes disease: Generalized discriminant analysis and least square support vector machine. *Expert Systems with Applications*, *34*, 482 – 487.

[25] Purwar, A., & Singh, S. K. (2015). Hybrid prediction model with missing value imputation for medical data. *Expert Systems with Applications*, *42*, 5621–5631.

[26] Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, *66*, 846–850.

[27] Rapid Miner Project, R. M. (2013). The Rapid Miner Project for Machine Learning. Available: http://rapid-i.com/ Last access on Febraury 2014, .

[28] Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Computational and Applied Mathematics*, (pp. 53–65).

[29] Salton G. (1971). *The SMART retrieval system: experiments in automatic document processing*. Prentice-Hall.

[30] Sawacha, Z., Guarneri, G., Avogaro, A., & Cobelli, C. (2010). A new classification of diabetic gait pattern based on cluster analysis of biomechanical data. *Journal of Diabetes Science and Technology*, *4*, 1127–38.

[31] Sengur, A., & Turkoglu, I. (2008). A hybrid method based on artificial immune system and fuzzy k-nn algorithm for diagnosis of heart valve diseases. *Expert Systems with Applications*, *35*, 1011–1020.

[32] Steinbach, M., Karypis, G., & Kumar, V. (2000). A comparison of document clustering techniques. In *KDD Workshop on Text Mining*.

[33] Yun, U. (2008). A new framework for detecting weighted sequential patterns in large sequence databases. *Know.-Based Syst.*, *21*, 110–122.

[34] Zaki, M. J. (2001). Spade: An efficient algorithm for mining frequent sequences. *Mach. Learn.*, *42*, 31–60.

[35] Zheng, B., Yoon, S. W., & Lam, S. S. (2014). Breast cancer diagnosis based on feature extraction using a hybrid of k-means and support vector machine algorithms. *Expert Systems with Applications*, *41*, 1476–1482.