Under consideration for publication in Knowledge and Information Systems

An adaptive version of *k*-medoids to deal with the uncertainty in clustering heterogeneous data using an intermediary fusion approach

Aalaa Mojahed[1,2] and Beatriz de la Iglesia[1]

[1] University of East Anglia, Norwich Research Park, Norwich, Norfolk, UK; [2]Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia

Abstract. This paper introduces H*k*-medoids, a modified version of the standard *k*-medoids algorithm. The modification extends the algorithm for the problem of clustering complex heterogeneous objects that are described by a diversity of data types, e.g. text, images, structured data and time series. We first proposed an intermediary fusion approach to calculate fused similarities between objects, SMF, taking into account the similarities between the component elements of the objects using appropriate similarity measures. The fused approach entails uncertainty for incomplete objects or for objects which have diverging distances according to the different component. Our implementation of H*k*-medoids proposed here works with the fused distances and deals with the uncertainty in the fusion process. We experimentally evaluate the potential of our proposed algorithm using five datasets with different combinations of data types that define the objects. Our results show the feasibility of the our algorithm and also they show a performance enhancement when comparing to the application of the original SMF approach in combination with a standard *k*-medoids that does not take uncertainty into account. In addition, from a theoretical point of view, our proposed algorithm has lower computation complexity than the popular PAM implementation.

Keywords: Heterogeneous data, *k*-medoids, uncertainty, data fusion, clustering, SMF.

1. Introduction

Big data is defined in the context of velocity, volume and variety (Laney, 2001). Variety is often associated with heterogeneous data, i.e. data that represents

Received Oct 07, 2015
Revised Jan 02, 2016
Accepted Feb 28, 2016

some complexity, for example where objects are defined by several data types such as structured data, images, free text, and time series. In such scenarios, each data type may describe distinct perspective of the objects. For example, a patient's condition may be defined by its structure data such as demographics, treatment records etc., some time series to represent the results of blood samples over time, some images from radiography examinations and some textual reports recording the doctor's observations. To apply cluster analysis to this kind of data in a meaningful way, say to understand which patients experience similar disease progression over time, it may be necessary to include all of the descriptors of the patient's condition, i.e. all the data types. This is an area under addressed in current data mining research.

In previous work (Mojahed and De La Iglesia, 2014) we have proposed an intermediary fusion approach called SMF. The idea is to represent a heterogeneous object as a collection of its component elements. For each element, e.g. a text descriptor or an image, similarities to the same element in other objects are calculated independently. This produces a number of similarity or distance matrices, DMs, one per element, which are then fused to produce an individual similarity matrix for objects. Hence we produce a matrix of fused distances, FM, between heterogeneous objects that can be used to produce a configuration using a standard clustering algorithm. In this context, SMF also computes two uncertainty expressions: UFM and DFM. UFM reflects the uncertainty arising from assessing incomplete objects and DFM expresses the uncertainty in the final FM arising from the degree of disagreement between DMs.

For the clustering analysis, we can use any clustering algorithms that take as input a distance matrix, for example the standard k-medoids (Kaufman and Rousseeuw, 1987) which is one of the most popular techniques for clustering. Several versions of this algorithm have been proposed in the literature. For example: PAM (Partitioning Around Medoids) (Kaufman and Rousseeuw, 1987), CLARA (Clustering LARge Applications) (Kaufman. and Rousseeuw, 1990) and CLARANS (Clustering Large Applications based upon RANdomized Search) (Ng and Han, 1994). However, its application to our heterogeneous data with its inherent uncertainty may require some adaptation. Thus, in this paper we present Hk-medoids, an adapted version of k-medoids that is better suited to dealing with the type of heterogeneous objects we present and that incorporates the uncertainty of fusing the element distances into the algorithm. The main contributions of the paper are: a new Hk-medoids algorithm for clustering heterogeneous data that uses uncertainty in the fusion process to produce better clustering configurations; a comparison of intermediate data fusion approaches for clustering heterogeneous data; the compilation of a number of heterogeneous datasets that are made available for other researchers in this area; extensive experimental results on the clustering of heterogeneous data including multiple validation measures and statistical significance tests as well as empirical evidence of the efficiency of our algorithm in comparison with other clustering algorithms.

The rest of the paper is structured as follows: Section 2 presents a brief discussion of the related research. In Section 3 we give a definition of our problem and in Section 4 we summarise the SMF approach. This is followed by a description of the new Hk-medoids in Section 5 and its computation complexity in Section 6. Descriptions of the experimented data sets and the experimental set up are given in Sections 7 and 8. Then, Section 9 evaluates the performance of Hk-medoids, followed by Section 10 that concludes the paper and presents our future research intentions.

2. RELATED WORK

In the community of data mining and machine learning, clustering homogeneous data has been studied a great deal; comparatively, clustering of heterogeneous data is not a well-developed area of research (Gao et al., 2006). Few researchers have ventured into this field, as the basic assumption is that only homogeneous data objects can be successfully clustered; nothing substantial has been achieved yet. Two recent surveys have appeared on mining multimedia data (i.e. data containing mixed data types) (Manjunath et al., 2010; Akeem et al., 2012). They discuss various data mining approaches and techniques, including clustering. However, as survey papers, detailed procedures are not provided; instead, they focus only on defining the problem including the nature of this challenging data.

Clustering two data types simultaneously, documents and terms, is tackled in two similar studies: Dhillon (2001) and Zha et al. (2001). In both studies, researchers clustered documents and terms as vertices in a bipartite graph with the edges of the graph indicating their co-occurrence, using edge weights to indicate the frequency of this co-occurrence. There was a restriction in these papers: each word cluster was associated with a document cluster. The underlying assumption here was that words that typically appear together should be associated with same/similar concept which means similar documents. Considering this assumption as a limitation, Dhillon et al. (2003) worked on the same problem but they did not impose such a restriction in their study.

In addition to simultaneously clustering data types as above, a reinforcement approach was suggested by other researchers (Wang et al., 2003). The idea is to cluster multiple data types separately with inter-type links used to iteratively project the clustering results from one type onto another. The researchers applied their scheme on multi-type interrelated web objects, and they noted that their experimental results proved the effectiveness of this approach. Significant improvements in clustering accuracy were delivered compared to the result obtained by a standard "flat" clustering scheme. Their idea might have been inspired by a former study conducted by Zeng et al. (2002), which attempted to develop a unified framework for clustering heterogeneous web objects. Both studies represent relationships between objects as additional attributes of data that are used in the clustering. Thus, so far much of the work in this area relates to the clustering of multi-class interrelated objects, that is, objects defined by multiple data types and belonging to different classes that are connected to one another.

On the other hand, fusion approaches (Boström et al., 2007; Acar et al., 2013) are often used to deal with this mix of data as they can combine diverse data sources even when they differ in terms of representation. Generally speaking, fusion approaches focus on the analysis of multiple matrices and formulate data fusion as a collective factorisation of matrices. For example, Long et al. (2006) proposed a spectral clustering algorithm that uses the collective factorisation of related matrices to cluster multi-type interrelated objects. The algorithm discovers the hidden structures of multi-class/multi-type objects based on both feature information and relation information. Ma et al. (2008) also used fusion in the context of a collaborative filtering problem. They propose a new algorithm that fuses a user's social network graph with a user-item rating matrix using factor analysis based on probabilistic matrix factorisation in order to find more accurate recommendations. Some recent work on data fusion (Acar et al., 2013) has sought to understand when data fusion is useful and when the analysis of in-

dividual data sources may be more advantageous. Data fusion approaches have become popular for heterogeneous data as they handle the process of integration of multiple data and knowledge from the same real-world object into a consistent, accurate, and useful representation. In practice, data fusion has been evolving for a long time in multi-sensor research (Hall and Llinas, 1997; Khaleghi et al., 2013) and other areas such as robotics and machine learning (Abidi and Gonzalez, 1992; Faouzi et al., 2011). However, there has been little interaction with data mining research until recently (Dasarathy, 2003).

According to the stage at which the fusion procedure takes place in the modelling process, data fusion approaches are classified into three categories (Maragos et al., 2008; Pavlidis et al., 2002; Greene and Cunningham, 2009): early integration, late integration and intermediate integration. In early integration, data from different modalities are concatenated to form a single dataset. According to Žitnik and Zupan (2014), this fusion method is theoretically the most powerful approach but it neglects the modular structure of the data and relies on procedures for feature construction. Intermediate integration is the newest method. It retains the structure of the data and concatenates different modalities at the level of a predictive model. In other words, it addresses multiplicity and merges the data through the inference of a joint model. The negative aspect of intermediate integration is the requirement to develop a new inference algorithm for every given model type. However, according to some researchers (Žitnik and Zupan, 2014; van Vliet et al., 2012; Pavlidis et al., 2002; Lanckriet et al., 2004a) the intermediate data fusion approach is very accurate for prediction problems and may be very promising for clustering. In late integration, each data modality gives rise to a distinct model and models are fused using different weightings.

Though many studies (e.g. (Lanckriet et al., 2004b; Bie et al., 2007; Shi et al., 2010)) have examined data fusion in classification there is less work in the clustering domain. However, work on intermediate fusion for data clustering was conducted by Yu et al. (2009) and found to be promising. Yu et al. formulated data fusion by fusing similarity matrices and reported better results than using a clustering ensemble approach. On the other hand, Greene and Cunningham (2009) presented an approach to clustering with late integration using matrix factorisation. Others have also derived clustering using various ensemble methods (Dimitriadou et al., 2002; Strehl and Ghosh, 2003; Wang et al., 2003; Gao et al., 2006) to arrive at a consensus clustering.

Despite some researchers working on related studies as presented above, they either have a different definition of data heterogeneity (e.g. relaying on the interlinking of data types), work on other data mining tasks (e.g. collaborative filtering), or their approaches are not fully explained. Thus, a comparison against other state-of-art intermediary fusion approaches on the same problem was not possible. Instead, we provide a comprehensive comparison including experiments of our two proposed intermediate fusion techniques as well as on applying cluster analysis separately to individual data types. Our future work includes comparisons with late fusion techniques.

In our previous work, we have proposed an intermediary fusion approach, SMF, similar to that of Yu et al. (2009) as we fuse dis/similarity matrices. We calculated individual dis/similarity measures for each element that defines the object and fused them to find the FM. Clustering was then obtained using a standard clustering algorithm on the FM. However, we needed to incorporate the uncertainty that arises in the fusion mechanism. For this, we now present

notation	description
H	a heterogeneous dataset
O_i	the i^{th} object $\in H$
N	the total number of objects $\in H$
$\mathcal{E}^j_{O_i}$	the j^{th} element of the i^{th} object
M	the total number of elements of O_i
UFM	matrix expressing the degree of uncertainty from missing elements
DM	matrix expressing the standard deviation of similarity values in the DMs
FM	a fusion matrix reporting fused distances
CV	a certainty vector
SD	a structured data element
TS	a time-series element
TE	a free text element
IE	an image element

Table 1. Notation used

a modified k-medoids algorithm that can take advantage of it. We evaluate our algorithm on a number of datasets that we have compiled for this purpose and which include time series, text and image elements. We investigate whether our approach can help us to discover the most relevant or informative elements in terms of clustering performance, and also whether the FM can perform as well as the best elements. We leave the comparison to late fusion as our next step.

3. PROBLEM STATEMENT

Important notation used in this paper from this point is summarised in Table 1.

A definition of our problem has been given in (Mojahed and De La Iglesia, 2014; Mojahed et al., 2015) but we reproduce it here to aid the reader in following the discussion. The formal definition of a heterogeneous dataset, H, is a set of objects such that $H = \{O_1, O_2, \ldots, O_i, \ldots, O_N\}$, where N is the total number of objects in H and O_i is the i^{th} object in H. Each object, O_i, is defined by a unique *Object Identifier*, $O_i.ID$. We use the dot notation to access the identifier and other component parts of an object. In our heterogeneous dataset objects are also defined by a number of components or elements $O_i = \{\mathcal{E}^1_{O_i}, \ldots, \mathcal{E}^j_{O_i}, \ldots, \mathcal{E}^M_{O_i}\}$, where M represents the total number of elements and $\mathcal{E}^j_{O_i}$ represents the data relating to $\mathcal{E}^j$ for O_i. Each full element, $\mathcal{E}^j$, for $1 \leq j \leq M$, may be considered as representing and storing a different data type. Hence, we can view H from two different perspectives: as a set of objects containing data for each element or as a set of elements containing data for each object. Either representation will allow us to extract the required information. We begin by considering a number of data types:

SD A heterogeneous dataset may contain a (generally only one) SD element, $\mathcal{E}^{SD}$. In this case, there is a set of attributes $\mathcal{E}^{SD} = \{A^1, A^2, \ldots, A^p\}$ defined over p domains with the expectation that every object, O_i, contains a set of values for some or all of the attributes in $\mathcal{E}^{SD}$. Hence, $\mathcal{E}^{SD}$ is a $N \times p$ matrix in which the columns represent the different attributes in $\mathcal{E}^{SD}$ and the rows represent the values of each object, O_i, for the set of attributes in $\mathcal{E}^{SD}$. The domains for

SD are those considered in relational databases, e.g. primitive domains, date or partial date domains, time domain, etc.

TS The heterogeneous dataset may also contain one or more time-series elements: $\mathcal{E}^{TS1}, \ldots, \mathcal{E}^{TSg}, \ldots, \mathcal{E}^{TSq}$. A TS is a temporally ordered set of r values which are typically collected in successive (possibly fixed) intervals of time: $\mathcal{E}^{TSg} = \{(t_1, v_1), \ldots, (t_l, v_l), \ldots, (t_r, v_r)\}$ such that v_1 is the first recorded value at time t_1 , v_l is the l^{th} recorded value at time t_l, etc., $\forall l, v_l \in \Re$. Any TS element, $\mathcal{E}^{TSg}$, can be represented as a vector of r time/value pairs. Note, however, that r is not fixed, and thus the length of the same time-series element can vary among different objects.

TE A heterogeneous object may be described using one or more distinct text elements. A text element is referred to an unstructured or a semi-structured segment of text forming a document and modeled as a vector of t values that belongs to the term-frequency-matrix, *TFM*. A term is a word(s) or set of words or a phrase (a word in our case) that exists in a document and is extracted using one of the string matching algorithms. *TFM* is a mathematical d x t matrix that represents the frequency of a list of t terms in a set of d documents. Rows correspond to documents and columns correspond to terms. Term frequency-inverse document frequency (tf-idf) (Huang, 2008) is a weighting scheme that was used to determine the value of each entry in *TFM*. This scheme uses a statistic weighting factor that reflects how important a word is to a particular document that belongs to a set of documents. Note that, in the case of having more than one *TE* for the same object, they might be viewed as distinct elements or they could be merged and viewed as one element when that makes sense.

IE A heterogeneous object may be described by one or more m×n 24-bit RGB images, sometimes known as a true color images. An RGB image is stored as a 3-dimensional matrix which is m×n×3 such as $IMG = \{img_{1,1,1}, img_{1,1,2}, img_{1,1,3}, img_{1,2,1}, img_{1,2,2}, \ldots, img_{1,n,3}, \ldots, img_{2,1,1}, \ldots, img_{m,n,3}\}$. The first two dimensions of the matrix, m and n, are the image dimensions, that is, m×n is the number of pixels. The third dimension of the matrix, 3, is used to define red, green, and blue color components for each individual pixel. The color of each pixel is determined by the combination of the three color intensities. For a particular pixel, color intensities are stored in each of the three color planes at the pixel's position as a number between 0 and 1. The color components for a black pixel are 0, 0, and 0 for the red, green and blue plane, while a pixel whose color components are 1, 1, and 1 is displayed as white. The three color components for each pixel are stored along the third dimension of the RGB matrix. For example, the red, green, and blue color components of the pixel (6,15) are stored in the following position of the RGB matrix: (6,15,1), (6,15,2), and (6,15,3), respectively. In a 24-bit RGB images, every color plane is 8 bits which produces up to 16 million different colors, 2^{24} combinations.

As a general comment, this definition of an object is extensible and allows for the introduction of further data types such as video, sounds, etc. Moreover, it can be concluded from the above definition that any object $O_i \in H$ might contain more than one element drawn from the same data category. In other words, a particular object O_i may be composed of a number of *SDs* and/or *TSs* and/or images. Moreover, incomplete objects are permitted, where one or more of their elements are absent.

For a heterogeneous dataset, *H*, comprising N objects as defined above, then

the target is to cluster the N objects into k groups where $k \leq N$. Normally, to achieve the clustering goal, the number of clusters has to be $k \ll N$. The partition of H into k clusters is denoted as $\hat{C} = \{C_1, C_2, \ldots, C_k\}$ where each C_i is formed by grouping similar heterogeneous objects based on similarity measures.

4. SIMILARITY MATRIX FUSION

Our proposed SMF approach (Mojahed and De La Iglesia, 2014) requires us to calculate the FM before we consider the clustering algorithm. This involves calculating the individual DMs for each element and then using a weighted approach to produce the FM. Uncertainty expressions are also calculated at this time, including UFM, DFM and the certainty vector, CV. We provide in this section a summary of those calculations for clarity.

We begin by computing distance matrices for each given data element, $\mathcal{E}^z$, associated with a particular data type:

$$DM^{\mathcal{E}^z}_{O_i,O_j} = dist(O_i.\mathcal{E}^z, O_j.\mathcal{E}^z), \tag{1}$$

where O_i and O_j are two heterogeneous objects and in each case *dist* represents an appropriate distance measure for the given data type. This might need some knowledge about data manipulation, in particular distance measures, for different data types. However, there are widely-known methods that can work effectively for each data type and those are the ones we apply and recommend here. An in-depth study of distance measures is outside the scope of this paper.

We then scale $DM^{\mathcal{E}^z}_{O_i,O_j}$ as follows:

$$DM^{\mathcal{E}^z}_{O_i,O_j} = \frac{dist(O_i.\mathcal{E}^z, O_j.\mathcal{E}^z) - min\left\{DM^{\mathcal{E}^z}\right\}}{max\left\{DM^{\mathcal{E}^z}\right\} - min\left\{DM^{\mathcal{E}^z}\right\}}, \tag{2}$$

To generate FM, assuming all weights equal, i.e. $\forall z, w^z{=}1$:

$$FM_{O_i,O_j} = \frac{\sum_{z=1}^{M} w^z \times dist(O_i.\mathcal{E}^z, O_j.\mathcal{E}^z)}{\sum_{z=1}^{M} w^z} \tag{3}$$

Uncertainty expressions, UFM and DFM, are also calculated for each pair of objects and can be considered as companion matrices for the FM that express the degree of uncertainty in the fused calculations. UFM computes the proportion of missing similarity values in the DMs associated with the elements, while DFM, calculates the standard deviation of similarity values in the DMs associated with the elements:

$$UFM_{O_i,O_j} = \frac{1}{M}\sum_{z=1}^{M}\begin{cases}1, & DM^{\mathcal{E}^z}_{O_i,O_j} \neq null\\ 0, & otherwise\end{cases} \tag{4}$$

$$DFM_{O_i,O_j} = \left(\frac{1}{M}\sum_{z=1}^{M}(DM^{\mathcal{E}^z}_{O_i,O_j} - \overline{DM_{O_i,O_j}})^2\right)^{\frac{1}{2}}, \tag{5}$$

where,

$$\overline{DM_{O_i,O_j}} = \frac{1}{M}\sum_{z=1}^{M} DM_{O_i,O_j}^{\mathcal{E}^z}$$

We then define certainty criteria by setting threshold(s) for one or both of the UFM and DFM expressions, for example, $UFM \geq \phi_1$ and/or $DFM \geq \phi_2$. Accordingly, we can determine pairs of objects for which FM calculations are uncertain, given defined thresholds:

$$Certain(O_i, O_j) = 1 \quad \forall O_i, O_j \mid UFM_{O_i,O_j} \geq \phi_1 \; and/or \; DFM_{O_i,O_j} \geq \phi_2 \quad (6)$$

For a given object, O_i, the certainty is defined in relation to all the other objects:

$$Certainty(O_i) = \sum_{1 \leq j \leq N} Certain(O_i, O_j) \quad (7)$$

We can then produce a certainty vector, CV, such that:
$CV = \{CV_{O_1}, CV_{O_2}, \ldots, CV_{O_N}\}$ where

$$CV_{O_i} = \begin{cases} 0, & Certainty(O_i) \geq \frac{N}{2} \\ 1, & otherwise \end{cases}$$

In other words, CV is a N binary vector indicating which of the N objects have uncertain fused calculations according to the UFM and/or DFM thresholds, ϕ_1 and ϕ_2. CV_{O_i} is created for O_i by analysing the uncertainty calculations that are defined for O_i in relation to all the other objects. When the number of objects that hold uncertain calculations with O_i is greater than half of the total number of objects in the dataset, CV considers it as an object with uncertain calculations and vice versa.

5. THE PROPOSED H*K*-MEDOIDS

Similar to the standard k-medoids, the proposed Hk-medoids makes multiple iterative passes through the dataset and allows object membership to change based on distance from medoids. It seeks to minimize the total variance of the clusters, i.e., the sum of the distances from each object to its assigned cluster medoid. In both algorithms, we need to update the objects assignments and the medoids allocations.

For the update stage, some k-medoids implementations work in a similar way to k-means, that is, they have two update phases iteratively applied over all k clusters. The literature often describes the two update phases as batch update and PAM-like online update. For example, the implementation that we have used in this paper called 'small' employs a variant of the Lloyd's iterations based on the work of Park and Jun (2009). During the batch update, each iteration consists of reassigning objects to their closest medoid, all at once, followed by recalculation of cluster medoids. During the PAM-like online update, for each cluster, a small subset of data objects that are normally the furthest from and nearest to the medoid are chosen. For each chosen data object, the algorithm reassigns the clustering of the whole dataset and checks if doing so will reduce

the sum of distances. This approach is similar to what PAM does, however, the swap considerations are limited to the points near the medoids and far from the medoids. The operation of both update phases tends to improve the quality of solutions generated. Individually, the online update seems to produce better solution than those found by the full batch update (Liang and Klein, 2009).

Thus, in Hk-medoids, we exploit the difference between batch and PAM-like online update phases; however, we use a different subset selection condition. We restrict the PAM-like swap step to certain objects only, then reallocate all the objects to the new medoids. The rationale for this is that the certain objects play a bigger role in establishing the clustering solution initially, while the uncertain objects are discounted. However, the second phase allows the clustering to be influenced by the uncertain objects as well, hopefully producing a good clustering solution for all objects. The pseudo code of Hk-medoids is presented below in Figure 1. Our proposed algorithm therefore takes account of the uncertainty inherent in the fusion process to drive the clustering process.

6. THE COMPUTATIONAL COMPLEXITY OF THE PROPOSED HK-MEDOIDS

A time consuming part of any standard k-medoids implementation is the calculation of the distances between objects. However, our algorithm takes the pairwise fused distance matrix as an input, thus this becomes a preliminary step. It uses $\mathbf{O}(M \times N^2)$ steps to calculate FM, where M is the number of elements and N is the number of heterogeneous objects. To compare the efficiency of our proposed algorithm to the most popular k-medoids implementation, PAM, we can discuss their computational complexity. We are interested in comparing our work to PAM because our algorithm has a main iterative step that works similarly to PAM. Also, we have analysed the complexity of 'small' for the same reason. The complexity of PAM is $\mathbf{O}(k(N-k)^2)$, where k is number of clusters. However, other k-means like implementations, e.g. 'small', are $\mathbf{O}(kN)$. By analysing the pseudo code of Hk-medoids in Figure 1 we can observe that the iterative parts of the algorithm are in step 3 (similar to 'small') and step 4 (similar to PAM). The computational complexity of step 3 is $\mathbf{O}(k(N-n))$ where n is the number of uncertain objects, while the complexity of Step 4 is $\mathbf{O}(k(N-n-k)^2)$. Thus, the cost of step 3 is less than the cost of 'small' and the cost of step 4 is less than the cost of PAM given large n. That is, the differences become more noticeable when we use specific uncertainty thresholds that control the number of certain/uncertain objects. In other words, if we come to a point where $n = 0$ or n is a very small number, so that most objects are certain, the cost of step 3 will be equivalent to the cost of 'small' and step 4 will not be executed at all, hence the behaviour of our algorithm will approximate that of 'small'. On the other hand, with a reasonable number of uncertain objects n (as in our experimental Section 8), Hk-medoids will be more efficient in term of execution time compared to the standard PAM as the number of swaps in step 4 will be n and not N. Thus, we overcome a main drawback of PAM which works inefficiently for large datasets due to its swap complexity. In summary, Hk-medoids consists of two different iterative steps, but it is still less expensive than PAM +'small'. This is true even in worse scenario, i.e. when $n = N$.

Input: **FM**: $N \times N$ pairwise distance fusion matrix for N objects, $O_1, O_2, \ldots, O_N$
k: number of clusters
CV:certainty vector for N objects, $CV = \{CV_{O_i}\}_{i=1}^{N}$
Output: a set of ***k*** clusters' medoids, $\mathfrak{M} = \{\mathfrak{M}_j\}_{j=1}^{k}$
label assignments $\forall O_i$, $L = \{L_{O_i}\}_{i=1}^{N}$
Method:

1: Choose k initial objects as medoids, $\mathfrak{M}_1, \mathfrak{M}_2, \ldots, \mathfrak{M}_k$ randomly
2: Assign the remaining $N-k$ objects to the closest medoids using the FM:
foreach $O_i \in$ the remaining $N-k$ objects
$L_{O_i} \leftarrow \arg\min_j \mathbf{FM}(O_i, \mathfrak{M}_j), j \in \{1 \ldots k\}$
end
3: Begin the batch-updating phase using certain objects only:
repeat
%% calculate medoids using certain objects
foreach $\mathfrak{M}_p \in \mathfrak{M}$ **do**
$x \leftarrow \arg\min_{1 \le j \le N} \sum_{i=1}^{N} \mathbf{FM}(O_i, O_j), \forall \quad certainO_i, certainO_j \in \mathfrak{M}_p$, i.e. $CV_{O_i} = 1$, $CV_{O_j} = 1$
if $(O_x \neq \mathfrak{M}_p)$ **then**
$\mathfrak{M}_p = O_x$
%% assign certain objects to the nearest medoids
foreach O_i, $i \in \{1 \ldots N\}$
if $CV_{O_i} = 1$ **then**
$\mathscr{L}_{O_i} \leftarrow \arg\min_j \mathbf{FM}(O_i, \mathfrak{M}_j), j \in \{1 \ldots k\}$
if $(\mathscr{L}_{O_i} \neq L_{O_i})$ **then**
$L_{O_i} \leftarrow \mathscr{L}_{O_i}$
end
end
end
until none of the L_{O_i} change
4: Begin the PAM-like online-updating phase to deal with uncertain objects:
%% assign uncertain objects to the nearest medoids
foreach O_i, $i \in \{1 \ldots N\}$
if $CV_{O_i} = 0$ **then**
$\mathscr{L}_{O_i} \leftarrow \arg\min_j \mathbf{FM}(O_i, \mathfrak{M}_j), j \in \{1 \ldots k\}$
if $(\mathscr{L}_{O_i} \neq L_{O_i})$ **then**
$L_{O_i} \leftarrow \mathscr{L}_{O_i}$
end
end
end
%% operate PAM-like swap step using uncertain objects only
do
foreach $\mathfrak{M}_p \in \mathfrak{M}$ **do**
$x \leftarrow \arg\min_{1 \le j \le N} \sum_{i=1}^{N} FM(O_i, O_j), \forall \quad O_i, uncertainO_j \in \mathfrak{M}_p$, i.e. $CV_{O_i} = 0 || 1, CV_{O_j} = 0$
if $(O_x \neq \mathfrak{M}_p)$ **then**
$\mathfrak{M}_p \leftarrow O_x$
end
end
%% if any $\mathfrak{M}_j$ change, assign all objects to the nearest medoids
foreach O_i, $i \in \{1 \ldots N\}$
$\mathscr{L}_{O_i} \leftarrow \arg\min_j \mathbf{FM}(O_i, \mathfrak{M}_j), j \in \{1 \ldots k\}$
if $(\mathscr{L}_{O_i} \neq L_{O_i})$ **then**
$L_{O_i} \leftarrow \mathscr{L}_{O_i}$
end
end
until none of the $\mathfrak{M}_j$ change
5: **return** $\mathfrak{M}$ and L

Fig. 1. Hk-medoids clustering algorithm

7. THE EXPERIMENTAL DATA SETS

This section gives descriptions of the heterogeneous datasets that we have compiled for these experiments. As unfortunately, there are no readily available large datasets that we could find containing data heterogeneity as we define it, we have started compiling our own collection. It is not easy to construct these datasets

dataset	no. of objects	no. of elements	type of elements	no. of groupings	no. of FMs
Cancer	1,598	24	1 SD, 23 TSs	4	1
Plants	100	3	1 SD, 1 TE, 1 IE	1	4
Journals	135	3	1 SD, 2 TSs	3	1
Papers	300	3	1 SD, 1 TS, 1 TE	1	2
Celebrities	100	3	1 SD, 2 TSs	1	1

Table 2. Main characteristics of our heterogeneous datasets

as it is a semi manual process. Hence, although the number of objects we have gathered is limited in our datasets, they are complex as they are composed of several different elements. Moreover, the number of objects in the cancer dataset is large compared to the other datasets and the data comes from a real world problem. Note also that it was not possible to gain access to the datasets that were examined by other researchers who studied similar problems.

The datasets we have compiled are publicly available at (Mojahed, 2015). They comprise different mixtures of elements, e.g. multiple TSs and SD, text and SD, etc. We start by proposing five heterogeneous datasets: the prostate cancer dataset, the plants dataset, the papers dataset, the journals dataset and the celebrities dataset. Table 2 summarises the main characteristics of these datasets and we follow with some additional descriptions.

The cancer dataset (Bettencourt-Silva et al., 2011) was the one that originally motivated our work and was donated to us. It contains data for a total of 1,904 patients diagnosed with prostate cancer at the Norwich and Norfolk University Hospital (NNUH), UK. Each patient's journey from diagnosis to end of study period is represented by a number of attributes. The structured data that describe each patient includes: demographics data (e.g. age at diagnosis, death indicator), disease state at diagnosis (e.g. Gleason score, tumor staging) and the types of treatments that the patient received (e.g. Hormone Therapy, radiology, surgery). In addition, 23 different blood test results (e.g. Vitamin D, MCV, Urea) are recorded over time and represented as 23 distinct TSs. After the data preparation stage, we ended up with 1,598 patients that had 100% complete SD elements. There are different natural groupings that can be drawn from the data, for example by risk score at diagnosis or by mortality outcome at the end of the study period (Mojahed et al., 2015). The natural grouping systems for patients were suggested by the data donors. They are as follows:

– **NICE system for risk stratification**
  There are a number of approaches used to classify risk groups for prostate cancer patients. A widely used system is a composite risk score. It uses three data variables: Prostate-Specific Antigen (PSA), Gleason Grade, and Clinical Stage (Tumour Stage). Risk assessment is conducted at the time of diagnosis or as soon as possible thereafter. This stratification reflects the clinicians' belief that patients with the same risk have a similar clinical outcome and may follow a similar trajectory through the disease pathway. The National Institute for Health and Care Excellence (NICE) (NICE, 2014) provides the following guidance, presented in Table 3 for the risk stratification of men with localised prostate cancer.
  Our dataset requires some adaptation to apply this guidance, and advice on

level of risk	**PSA** ng/ml		**Gleason score**		**clinical stage**
Low risk	<10	**and**	≤6	**and**	T1-T2a
Medium risk	10 -20	**or**	7	**or**	T2b
High risk	>20	**or**	8-10	**or**	≥T2c

Table 3. NICE risk group classification system for localised prostate cancer (NICE, 2014)

this was obtained from the data creators. PSA is recorded as a TS. What we have done is consider the value at diagnosis, and if there is nothing recorded at $time = 0$, then the closest value before any type of treatments. Gleason score is divided into two values; primary and secondary, thus we use the sum of both scores. The clinical stage is reported using numbers. We considered the following: clinical stage <2 as low, clinical stage = 2 as medium and clinical stage > 2 as high risk.

– **Gleason score risk classification system**
Another well-known risk classification can be obtained by using Gleason grade alone to classify patients diagnosed with prostate cancer. Gleason grade shows the level of differentiation of the cancer cells under the microscope. High differentiation is associated with worst prognosis which indicates more aggressive tumors (Chan et al., 2000). Gleason grade is computed as a sum of two or sometimes three scores: primary, secondary and tertiary (if applicable). Primary is the most commonly seen level of differentiation under the microscope, secondary is second most common and so on. The level of differentiation for these three scores is given from 1 to 5 and then summed together. The totals of Gleason scores in our dataset are all > 5 as all the cases are definite cancer patients. We have defined two ways of groupings patients according to their Gleason score:Gleason-score-1 and Gleason-score-2. The first way of grouping, Gleason-score-1, has 3 groups: low, medium and high risk. Gleason-score-2, classifies patients into 4 groups: low, medium-1, medium-2 and high risk. The difference between the two groupings is in the medium risk group. In Gleason-score-2 the medium group is divided into two subgroups depending on the exact values of the primary and secondary scores and not only their sum.
– **Mortality grouping**
This labeling procedure classifies patients according to the outcome at the end of the study period, rather than looking at the potential risk of patients at diagnosis. For this grouping we used death indicators after conducting some changes on the values of the corresponding attribute in the data preparation stage (for details see Mojahed et al. (2015)).

The plants dataset was derived from the website of the Royal Horticultural Society (RHS) (Society, 2014). We constructed the dataset by choosing 100 plant objects belonging to 3 distinct groups: 42 kinds of fruits, 22 different roses and 36 types of grass. Each plant has a description in the form of SD and another in the form of free text, TE, in addition to an image representation, IE. The structured data element includes data for 8 attributes, e.g. the plant's height, rate of growth, color, flowering period etc. The text element is a general free text description about the plant. The image element is a picture of the plant in Joint Photographic Experts Group, JPEG, image format. Hence each element contributes a complementary description of the objects. In the preparation stage

for TE, the tf-idf weighting scheme was used to construct a 100×1189 term matrix. The 1189 list of terms was created after removing punctuations, discarding duplications, eliminating stop words and applying a stemming algorithm. In addition, if we apply a basic frequency based term selection method to remove rare terms the list is cut down to 631 terms.

The journals dataset was obtained from the Journal Citation Reports (JCR) in the ISI Web of Knowledge website (Reuters, 2015a). We have developed the dataset by choosing 135 journals from two related fields of research: computer science and information systems. Each journal has a description in the form of SD and another in the form of two distinct TSs. The structured data element includes data for 11 attributes, e.g. number of citations, number of issues published by the journals per year, language of scripts, number of articles, etc. The two time-series elements report the annual number of citations for a 10 year period from 2004 to 2013. One TS element defines the number of citations to articles published in the journal, TStoJ, and the other TS reports the number of citations from articles published in the journals, TSfromJ. We have defined 3 grouping systems for our 135 journals. All the grouping systems use citation data to assess and track the influence of a journal in relation to other journals. They are as follows:

- **The Impact Factor score (IF)**
  The journal impact factor is calculated by dividing the number of citations in the JCR year by the total number of articles published in the two previous years. An Impact Factor of 1.0 means that, on average, the articles published one or two year ago have been cited one time. An Impact Factor of 2.5 means that, on average, the articles published one or two year ago have been cited two and a half times. The citing works may be articles published in the same journal. However, most citing works are from different journals, proceedings, or books indexed by Web of Science. The journals in our dataset are divided into 5 categories, presented in Table 4.
- **The Eigenfactor Score (ES)**
  This score is based on the number of times articles from the journal published in the past five years have been cited in the JCR year, but it also considers which journals have contributed these citations so that highly cited journals will influence the network more than lesser cited journals. References from one article in a journal to another article from the same journal are removed, so that Eigenfactor Scores are not influenced by journal self-citation. Our objects are divided into 3 categories, presented in Table 4.
- **The Article Influence score (AI)**
  This score determines the average influence of a journal's articles over the first five years after publication. It is calculated by dividing a journal's Eigenfactor Score by the number of articles in the journal, normalized as a fraction of all articles in all publications. This measure is roughly analogous to the 5-Year Journal Impact Factor in that it is a ratio of a journal's citation influence to the size of the journal's article contribution over a period of five years. The mean Article Influence Score is 1.00. A score greater than 1.00 indicates that each article in the journal has above-average influence. A score less than 1.00 indicates that each article in the journal has below-average influence. The journals in our dataset are divided into 3 categories, presented in Table 4.

We also constructed a dataset, the papers dataset, containing research papers published in year 2002. The dataset is obtained from the Web of Science (Reuters,

grouping	group definition	number of objects
IF	IF≤0.5	28
	0.5<IF≤1.0	36
	1.0<IF≤1.5	29
	1.5<IF≤2.0	22
	IF>2.0	20
ES	ES≤0.0025	84
	0.0025<ES≤0.0	30
	ES>0.0	21
AI	AI≤0.4	62
	0.4<AI≤0.8	41
	AI>0.8	32

Table 4. The grouping systems for the journals dataset with the definitions of clusters and the number of objects that belong to each cluster

2015b), Thomson Scientific, by selecting 300 papers from 3 different research fields. These were: computing sciences, business and healthcare services, for each field we chose 100 papers. Each research paper has a description in the form of SD and another in the form of a TS, in addition to a TE element. SD includes data for 7 attributes, e.g. number of pages, total number of citations, number of authors, month of publication, etc. The time-series, TS, is supplementary data for the paper's citations spanning 16 years. It reports the annual number of citations to the paper per year, from 2000 to year 2015. The text element, TE, is basically the paper's abstract. There are 5 papers that have some missing values within their SD element. In the data preparation stage, the TE element was processed according to standard text mining operations, similarly to how we dealt with the text element of the plant dataset. As a result, the terms list includes 4,351 words and 1,080 words after removing rare terms.

The celebrities dataset was obtained from multiple web sources: Forbes (Inc., 2015), Wikipedia (Wikipedia, 2015) and Google Trends (Google, 2015). We have developed the dataset by collecting data about the 100 celebrities that we have in our list. They are divided into 3 groups of professions: actors/actresses (30), musicians (24) and other celebrity personalities including athletes, directors, producers and authors (46). Each celebrity has a description in the form of SD and another in the form of two distinct TSs that report the weekly normalized number of searches about the celebrity. Structured data includes data for 12 attributes, e.g. age, gender, number of awards, the year of activation, etc. The two time-series elements, TSs, report the weekly normalized number of searches of the celebrity that have been performed from the first week in January 3013 to the first week in January 2015. One TS element defines the interest of people in the UK through web searches, TSweb, and the other TS reports their interest using Youtube searches, TSUtube.

8. EXPERIMENTAL SET UP

In order to compute DMs for SD element in all the experimented datasets, we chose the Standardized Euclidean distance, which requires computing the standard deviation vector. With regards to TE elements, we chose the most common

% of uncertain objects	**4%**	**8%**	**15%**	**30%**	**46%**	**73%**
margins limits	objects≤**5%**	**5%**<objects≤**10%**	**10%**<objects≤**20%**	**20%**<objects≤**35%**	**35%**<objects≤**50%**	**50%**<objects≤**75%**
SMF	0.2900	0.3321	0.4107	0.4246	0.3233	0.2100
Hk-medoids	0.3833	0.4148	0.7200	0.7233	0.5800	0.3633

Table 5. Certainty thresholds sensitivity example: performance on the paper dataset is measured by Jaccard coefficients. The first raw indicates the actual percentages and the second raw represents the thresholds margins range.

measure of similarity in text mining, the Cosine calculation (Salton and McGill, 1987) as this measure is widely used and reported to be effective with text, for example in information retrieval applications (Baeza-Yates and Ribeiro-Neto, 1999) and in clustering (Larsen and Aone, 1999). For TSs, we use Dynamic Time Warping (DTW), first introduced into the data mining community in 1996 (Berndt and Clifford, 1996). DTW can cope with TSs of different lengths. Its ability to do this was tested by many researchers (e.g., (Ratanamahatana and Keogh, 2005)). However, our calculated distances are normalized through the sum of the lengths of the TSs that we are comparing. For IE, we use the GIST (Oliva and Torralba, 2001) descriptor as it is easy to compute, provides a compact representation of the images and it is not prone to segmentation errors. Also, it has recently shown good performance in different image tasks (e.g., image retrieval (Li et al., 2008) and image completion (Hays and Efros, 2007)).

By choosing the above similarity calculations, we were able to obtain individual DMs as the first step of SMF. Afterwards, we combine the individual DM as proposed in section 4 to calculate the FMs; then we calculate UFMs, DFMs and CVs. Using all these calculations, we first apply a standard k-medoids algorithm to the FMs.

As a second step, we want to compare the clustering results obtained with standard k-medoids on all objects to k-medoids using uncertainty filters so that certain objects dominate the experiment. For this, we applied the k-medoids only to the objects that are considered as certain using specified thresholds for UFMs and DFMs, and then we assigned uncertain objects to the closest generated medoids.

We want to set the thresholds in a way that considers a reasonable number of objects as uncertain, thus, we neither assess a very big nor a very small proportion of objects as uncertain. Our parameter experimentations lead to thresholds associated with between 10% and 35% of objects being considered as uncertain because between those margins we saw little effect on performance. However, when going outside those margins, clustering performance deteriorates. We illustrate the sensitivity of this parameter using the paper dataset in Table 5. It compares jaccard coefficients for both SMF and Hk-medoids when we set different thresholds for UFM and DFM. We can see that thresholds leading to less than 10%, and greater than 35% of objects being considered as uncertain gave worse results for Hk-medoids and the SMF approach. Note that the same conclusion is also obtained for the other datasets.

Note that, uncertainty filters for the plants dataset and the celebrities dataset included DFM only because all objects are complete so it is only necessary to deal with uncertainty arising from the disagreement between DMs.

Finally, we implement our proposed Hk-medoids algorithm using all the re-

quired pre-calculated matrices and specified settings. We assess and compare all the obtained clustering configurations.

With regards to clustering, the five heterogeneous datasets we have compiled have one or more natural grouping system(s). Thus, we can benefit from the ground truth labels when evaluating clustering performance. However, we are interested in the ability of the FM to identify the correct clusters, as opposed to details of the individual grouping systems. Hence, instead of the grouping's name, we used here numbers to identify the different systems, e.g. grouping1, grouping2, etc. To evaluate the clusterings obtained by each approach we calculate 3 different external validation measures: Jaccard coefficient (Jaccard, 1908), rand statistic (Rand, 1958) and Dice's index (Dice, 1945). Finally, we demonstrate the significance of Hk-medoids performance using statistical testing. We apply a z-test to establish if the differences in performance between Hk-medoids and the best individual DM and between Hk-medoids and SMF are statistically significant. We compare the difference in performances using the Jaccard calculations as a representative of the external validation coefficients. Note that as the nature of k-medoids implies that we may get different results with different initialisations, we applied each algorithm 50 times. Each run was executed with random initialization. In the next section we report the best result for each experiment out of 50 runs, that is both for k-medoids and Hk-medoids.

9. RESULTS AND DISCUSSION

Before applying our Hk-medoids algorithm, we apply SMF to produce a single matrix to represent dis/similarities between heterogeneous objects as well as the matrices for uncertainty. For the cancer dataset, SMF produced 24 DMs that reflect the distances for each element separately in addition to FM which fuses all the 24 elements with equal weights. Uncertainty related to FM was calculated in UFM and DFM. Thresholds were set as UFM=0.4 and DFM=0.3, as a result we considered 175 patients as uncertain objects which is about 10.95% of the total number of objects. This dataset can be characterised by 4 different natural groping systems according to either diagnostic information or outcome information, hence we assess the clustering results obtained against those 4 groupings.

For the plants dataset, 5 DMs were generated by SMF. They corresponded to the SD, TE, TE element discounting rare terms (TENoRare), IE and the IE element represented with reduced colours (IEReduced). We can therefore fuse different combinations of those, producing 4 FMs. All fused distances were calculated using equal weights:

– FM fuses DM^{SD}, DM^{TE} and DM^{IE};

– FM-NoRare fuses DM^{SD}, $DM^{TENoRare}$ and DM^{IE};

– FM-NoRare-Reduced fuses DM^{SD}, $DM^{TENoRare}$ and $DM^{IEReduced}$;

– FM-Reduced fuses DM^{SD}, DM^{TE} and $DM^{IEReduced}$.

Only a DFMs filter was used because there were no incomplete objects. The value of the filter was 0.3, and that lead to the inclusion of 14, 24, 25 and 20 plants respectively for FM, FM-NoRare, FM-NoRare-Reduced and FM-Reduced. The

number of uncertain objects according to this filter was in all cases $> 10\%$ and $\leq 25\%$ of the total number of plants objects.

For the journals dataset we produced 3 DMs that reflect the distances for each element separately in addition to one FM which fuses all the 3 elements with equal weights. UFM and DFM thresholds were set up as $UFM = 0.33$ and $DFM = 0.1$. By applying those filters, we considered 41 journals as uncertain or just around 30% of the 135 journals that we have.

For the papers dataset, SMF produced 4 DMs for SD, TS, TE and TE element without rare terms respectively. In addition, 2 FMs were produced using equal weights:

– FM fuses DM^{SD}, DM^{TS} and DM^{TE};

– FM-NoRare fuses DM^{SD}, DM^{TS} and $DM^{TENoRare}$.

Uncertainty thresholds were set up as $UFM = 0.33$ and $DFMs = 0.4$. Using those filters, 99 papers were considered in both FM and FM-NoRare analysis as uncertain objects or $< 33\%$ of the total number of objects.

For the celebrities dataset, SMF generated 3 DMs that reflect the distances for each element separately and FM-1 which fuses all the 3 elements with equal weights. DFM was also computed. UFM was not calculated as there were no incomplete objects in the analysis. The threshold was set up as $DFM = 0.2$. As a result of applying this filter, we dealt with 23% of our objects as uncertain data.

After calculating all the required distance matrices and setting uncertainty filters, we applied the clustering algorithm. Clustering results are summarised in table 6. For the cancer and journals dataset, there are more than one natural grouping system for the objects and those are represented in the table using grouping1, grouping2, ..., etc. For the other 3 datasets, we consider only one possible grouping system. However, in the plants and papers datasets, we generated multiple fusion matrices to examined all different possible combinations of the individual DMs as described in Section 9 and they are presented as rows. The performance of the SMF approach is reported in the first two columns and that of Hk-medoids is given in the last column in the table. The first column shows the results of using SMF in conjunction with the standard k-medoids algorithm applied on all objects. The second column shows the results of applying SMF and k-medoids but this time using the uncertainty filters to apply k-medoids only to the objects that are considered as certain. Uncertain objects are then assigned to the resulting clustering. In the final column, we show the results of applying our proposed Hk-medoids algorithm using all objects. The numbers represent the value of the Jaccard coefficient in each case, as a representative measure for external clustering validity. Although, we only present Jaccard coefficients for space reasons, the same conclusion was reached by the other external validity coefficients: Rand and Dice's index. We used * next to the performance of Hk-medoids to signify statistical difference with the performance of the standard SMF approach without uncertainty filters.

The results in table 6 suggest that the performance decreases when we allow only certain objects to establish the initial clustering (SMF with uncertainty filters) compared to the results obtained using the full FM. However, the Hk-medoids approach has produced better clustering performance in all cases. To validate this important conclusion, we have tested if the differences between per-

grouping system/ fused matrices	**SMF** without uncertainty filters	**SMF** with uncertainty filters	**Hk-medoids**
The cancer dataset			
grouping1	0.5382	0.4132	0.7021*
grouping2	0.5651	0.3440	0.6358*
grouping3	0.4061	0.3292	0.4431*
grouping4	0.4781	0.3990	0.5307*
The plants dataset			
FM	0.6900	0.4651	0.7200
FM-NoRare	0.7300	0.4468	0.8500*
FM-NoRare-Reduced	0.8500	0.5761	0.8600
FM-Reduced	0.6900	0.4375	0.8300*
The journals dataset			
grouping1	0.3556	0.3085	0.4222
grouping2	0.7926	0.4468	0.8222
grouping3	0.5111	0.4362	0.5926*
The papers dataset			
FM	0.6833	0.4246	0.7233
FM-NoRare	0.6833	0.5238	0.7333
The celebrities dataset			
FM	0.5400	0.4286	0.6200

Table 6. A comparison between the performance of SMF and Hk-medoids clustering for the prostate cancer, plants, journals, papers and celebrities dataset. Jaccard coefficients are calculated using ground truth labels.

formances are significant. All p values that compare the performance of SMF with uncertainty filters and Hk-medoids are < 0.05 which indicates significant difference between Jaccard values. With regards to Hk-medoids and SMF without uncertainty filters, the p statistics report them as significant for the cancer data (< 0.00001, 0.02305, 0.017172 and 0.00147 for grouping1, grouping2, grouping3 and grouping4 respectively) and also for two of the FMs of the plants dataset, FM-NoRare (0.018626) and FM-Reduced (0.010225). In addition there is significant improvement in performance when we used grouping3 classification (0.024477) for the journals dataset. In general, these statistics prove that the Hk-medoids approach produces significantly better or comparable result to the standard SMF approach.

From Table 7 to Table 11 we present more detailed results for each dataset. In these tables, we highlighted in bold the best results for each validation measure and grouping. We used * next to the performances of Hk-medoids in order to highlight statistical difference in relation to the performance of the standard SMF approach without uncertainty filters. We use Jaccard calculations only as representative to test for statistical significance. A + indicates statistical difference between the value of Jaccard coefficients for the Hk-medoids algorithm when compared with the individual DMs.

Table 7 shows the value of external validity measures for the cancer dataset. It compares the clustering obtained using SMF and Hk-medoids to the one ob-

grouping system	**SD**			**best TS**			**SMF**			**Hk-medoids**		
	Jaccard	Rand	Dice	Jaccard	Rand	Dice	Jaccard	Rand	Dice	Jaccard	Rand	Dice
grouping1	0.3335	0.5037	0.4002	0.5119	0.5195	0.5059	0.5382+	0.5072	0.5184	**0.7021*+**	**0.5807**	**0.58407**
grouping2	0.4230	0.4860	0.4583	0.5569	0.5065	0.5269	0.5651+	0.51751	0.5306	**0.6358*+**	**0.5480**	**0.5598**
grouping3	0.2829	**0.5985**	0.3613	0.3767	0.5975	0.4297	0.4061+	0.5828	0.4482	**0.4431*+**	0.5502	**0.4698**
grouping4	0.3191	**0.5412**	0.3896	0.3899	0.5263	0.4381	0.4781+	0.5209	0.4888	**0.5307*+**	0.4878	**0.5172**

Table 7. Cancer dataset: A comparison of external clustering validity measures for clustering obtained using SMF, Hk-medoids, the SD element alone and the best TS element in the four natural grouping systems.

fusion matrix	**best DM**			**SMF**			**Hk-medoids**		
	Jaccard	Rand	Dice	Jaccard	Rand	Dice	Jaccard	Rand	Dice
FM	0.6600	**0.6778**	0.5690	0.6900	0.6469	0.5798	**0.7200**	**0.6778**	**0.5902**
FM-NoRare	0.7600	0.7251	0.6032	0.7300	0.7101	0.5935	**0.8500***	**0.8103**	**0.6296**
FM-NoRare-Reduced	0.7600	0.7251	0.6032	0.8500	0.6296	**0.6324**	**0.8600+**	**0.8319**	0.6323
FM-Reduced	0.6600	0.6778	0.5690	0.6900	0.6477	0.5798	**0.8300*+**	**0.7826**	**0.6241**

Table 8. Plants dataset: A comparison of external clustering validity measures for clustering obtained using SMF, Hk-medoids and the SD element alone showing as rows the different FMs.

tained by the SD element alone as well as the best individual TS in all the four grouping systems. From the table, we can see that Jaccard and Dice's are always in agreement and put the performance of SMF and specially Hk-medoids ahead. Rand agreed on their judgment in grouping1 and grouping2 but not in the other two groupings. With regards to the significant testing, all p values that compare the performance of SMF and Hk-medoids to the SD element and to the best TS using Jaccard index are < 0.05 which indicates significant differences (indicated by + in the table). Hence in terms of using individual elements to cluster versus using the SMF approach, for the cancer dataset the proposed Hk-medoids outperforms using the SD alone, despite the groupings being derived from information contained in the SD, and also it outperforms using the best TS.

For the plants dataset, table 8 compares the performances of SMF and Hk-medoids to the one obtained by the best individual DMs for all the four different FMs. Numbers in the table show that Jaccard, Dice's and Rand almost entirely agree on their judgment putting Hk-medoids ahead of the others. All three external validation techniques agree that Hk-medoids outperforms the best individual DM in all four cases. The significance test between Jaccard index of Hk-medoids and the best individual DM, represented by + in the table, shows that the difference is significant for FM-NoRare-Reduced and FM-Reduced.

For the journals dataset, table 9 shows the performances of SMF and Hk-medoids as well as TStoJ, the best element, for all the 3 groupings. Jaccard and Dice's conclude the same outcome and put Hk-medoids ahead. Rand agrees on their judgment only for grouping1. The statistical tests indicate that the difference is significant only in the case of grouping2 when comparing the Jaccard index for Hk-medoids and TStoJ. This is represented in the table using + symbol.

For the papers dataset, table 10 shows the performance of SMF and Hk-medoids against the best individual element. All validity measures put the performance of Hk-medoids ahead of the others for this dataset. The statistical tests, however, do not show significant improvements. For this, we calculated p

grouping system	best DM			SMF			Hk-medoids		
	Jaccard	Rand	Dice	Jaccard	Rand	Dice	Jaccard	Rand	Dice
grouping1	0.3407	0.6977	0.4053	0.3556	0.6816	0.4156	**0.4222**	**0.7216**	**0.4578**
grouping2	0.7333	0.6986	0.5946	0.7926	**0.8292**	0.6132	**0.8222+**	0.7703	**0.6218**
grouping3	0.5481	**0.6646**	0.5230	0.5111	0.4779	0.5161	**0.5926***	0.6524	**0.5424**

Table 9. Journals dataset: A comparison of external clustering validity measures for clustering obtained using SMF, Hk-medoids and the best individual element in all the three grouping systems.

fusion matrix	best DM			SMF			Hk-medoids		
	Jaccard	Rand	Dice	Jaccard	Rand	Dice	Jaccard	Rand	Dice
FM	0.6700	0.7313	0.5726	0.6833	0.7265	0.5775	**0.7233**	**0.7526**	**0.5902**
FM-NoRare	0.6700	0.7313	0.5726	0.6833	0.7265	0.5775	**0.7333**	**0.7603**	**0.5946**

Table 10. Papers dataset: A comparison of external clustering validity measures for clustering obtained using SMF, Hk-medoids and the best individual element when using the different fusion matrices.

values using Jaccard coefficients to test the differences in performance between Hk-medoids and the best DM and also between SMF and the best individual performer DM.

For the celebrities dataset, shown in table 11, again Hk-medoids is the best performer for all validity indexes, but not with a significant difference according to z-test assessment. Furthermore SMF performed slightly better than the best individual matrix, TSWeb.

With regards to the time cost of Hk-medoids, we said that our Hk-medoids is, theoretically, faster than the standard PAM implementation of the k-medoids. To back this with empirical evidence, we compared the elapsed time needed to produce the results by both algorithms for all the previous experiments over the five datasets. The specifications of the processor we used to run our implementations are: Intel(R) Core(TM) i5-3337U CPU, 1.8 GHz, 64-bit windows 8.1 operating system with 6 GB installed RAM. Table 12 compares the actual running time measured in seconds for all the 14 experiments. To demonstrate how Hk-medoids copes with the number of objects in datasets compared to PAM, a summarised graph of the running times is shown in Figure 2. The figure shows the average time needed to execute both algorithms on each of the datasets. Note that the graph orders the datasets according to the number of objects in an ascending order: plants, celebrities, journals, papers and cancer dataset. Table 12

fusion matrix	best DM			SMF			Hk-medoids		
	Jaccard	Rand	Dice	Jaccard	Rand	Dice	Jaccard	Rand	Dice
FM	0.5300	0.5469	0.5146	0.5400	0.5921	0.5192	**0.6200**	**0.6374**	**0.5536**

Table 11. Celebrities dataset: A comparison of external clustering validity measures for clustering obtained using SMF, Hk-medoids and the best individual element.

grouping system/ fused matrices	**Hk-medoids**	**PAM**
The cancer dataset		
grouping1	0.090856	2.696481
grouping2	0.091641	2.697643
grouping3	0.094189	2.781021
grouping4	0.092765	2.764705
The plants dataset		
FM	0.006175	0.011016
FM-NoRare	0.00603	0.011506
FM-NoRare-Reduced	0.005926	0.011456
FM-Reduced	0.00635	0.012425
The journals dataset		
grouping1	0.006851	0.018278
grouping2	0.007501	0.01645
grouping3	0.006707	0.01404
The papers dataset		
FM	0.008335	0.033052
FM-NoRare	0.008053	0.03301
The celebrities dataset		
FM	0.006441	0.013363

Table 12. The execution time measured in seconds of Hk-medoids and PAM implementation of the standard k-medoids for all the experiments.

and Figure 2 are empirical evidence of our claim about the time complexity of our algorithm, discussed in Section 6. The difference in the running time between the two algorithms is substantial when the number of objects changes from the minimum in the plants/celebrities datasets (100) to the maximum in the cancer dataset (1589). Hence, for real world datasets such as the cancer dataset our approach holds some promise.

10. CONCLUSIONS

In this paper we present a new algorithm, Hk-medoids, to cluster heterogeneous objects defined by numerous data types. Our algorithm makes use of uncertainty inherent in the fusion process to provide better clustering solutions and to improve on running time. Experimental results show promising outcomes both in terms of clustering quality obtained and running times.

In previous work, we have handled the challenge of applying clustering analysis to heterogeneous data by first computing a fused distance matrix that takes into account the distance values for each data type. We also proposed calculations that express the related uncertainty for both missing elements and also diverging assessment by different elements. However, the uncertainty calculations were not used in a meaningful way and did not provide any improvements to the basic fusion. Our previous approach, SMF, used a traditional clustering algorithm that could take a distance matrix as input, even when the original data matrix was

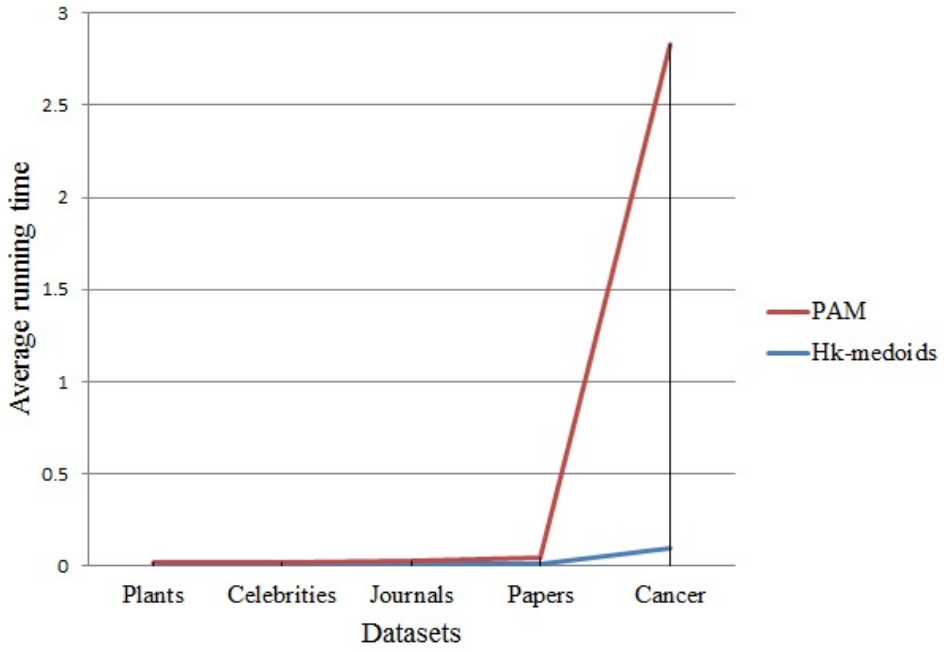

Fig. 2. The average execution time measured in seconds of Hk-medoids and PAM implementation of the standard k-medoids calculated for the heterogeneous datasets ordered in ascending number or objects.

not available, i.e. k-medoids. This could be suitable for many real-world applications, for example, when the data is private and should not be disclosed but the distance between objects could be published without compromising the original data.

Although, several versions of k-medoids were proposed and experimented with in the literature, they are not able to handle data heterogeneity as we have defined it nor the related uncertainty that arises in similarity calculations. Thus, with a view towards an integrated analysis of heterogeneous data, we introduce Hk-medoids, an adapted version of the standard 'small' k-medoids implementation that can address the aforementioned problems. This version takes as input the distance matrix and related uncertainty calculations from the SMF fusion approach but then uses those more effectively within the algorithm to produce a more reliable and accurate clustering configuration. The focus on certain objects for some parts of the algorithm also help to improve its efficiency.

We present five datasets that are compiled for our experimentation and which are made available to other researchers. In those datasets, objects are represented by standard data, text, time series and images in various combinations. Though some of those datasets are limited in size, they are complex and provide a contribution to other researchers working on this field.

Experimental results on those datasets compare the performance of the SMF approach, our initial attempt to use uncertainty within it by filtering uncertain objects, and our new Hk-medoids algorithm that integrates uncertainty into the clustering process. They also compare to the best performance obtainable by ap-

plying clustering to individual data elements. The results show the effectiveness of the proposed Hk-medoids algorithm. In all cases the new algorithm performs better in terms of computation time when compared to a PAM implementation, and this is particularly noticeable for the larger cancer dataset. In addition, as assessed by external clustering validation indexes it also performs equally well or statistically significantly better (as measured by a z-test) than the SMF approach and than clustering according to the best individual element. Since in practice it may not be possible to identify the best performing element in advance, using Hk-medoids may be more beneficial than it appears for clustering heterogeneous data. Another important feature of our implementation is that we adapted k-medoids, known as less sensitive to outliers compared to other popular clustering techniques. Moreover, our proposed algorithm deals with uncertainty that arises from the disagreement between DMs, calculated as DFM, which helps to tackle the noise in the data. All this increases the credibility of our proposal.

Clustering heterogeneous data is a rapidly growing area of research. We intend to expand on our research by conducting comparative studies with late fusion approaches applying ensemble methods. In late fusion the clustering analysis is performed separately on each data type and then at a later stage we arrive to the final results by fusing the different clusters.

We have published our datasets as well as our matlab implementation so other researchers can reproduce and compare to our results.

Acknowledgements. We acknowledge support from grant number ES/L011859/1, from The Business and Local Government Data Research Centre, funded by the Economic and Social Research Council to provide economic, scientific and social researchers and business analysts with secure data services.

References

Abidi, M. A. and Gonzalez, R. C. (1992). *Data Fusion in Robotics and Machine Intelligence*. Academic Press Professional, Inc., San Diego, CA, USA.

Acar, E., Rasmussen, M. A., Savorani, F., Naes, T., and Bro, R. (2013). Understanding data fusion within the framework of coupled matrix and tensor factorizations. *Chemometrics and Intelligent Laboratory Systems*, 129(0):53 – 63. Multiway and Multiset Methods.

Akeem, O. A., Ogunyinka, T. K., and Abimbola, B. L. (2012). A framework for multimedia data mining in information technology environment. *International Journal of Computer Science and Information Security (IJCSIS)*, 10(5):69–77.

Baeza-Yates, R. A. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Berndt, D. J. and Clifford, J. (1996). Finding patterns in time series: A dynamic programming approach. In Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R., editors, *Advances in Knowledge Discovery and Data Mining*, pages 229–248. American Association for Artificial Intelligence, Menlo Park, CA, USA.

Bettencourt-Silva, J., Iglesia, B., Donell, S., and Rayward-Smith, V. (2011). On creating a patient-centric database from multiple hospital information systems in a national health service secondary care setting. *Methods of Information in Medicine*, pages 6730–6737.

Bie, T. D., Tranchevent, L.-C., van Oeffelen, L. M. M., and Moreau, Y. (2007). Kernel-based data fusion for gene prioritization. In *ISMB/ECCB (Supplement of Bioinformatics)*, pages 125–132.

Boström, H., Andler, S. F., Brohede, M., Johansson, R., Karlsson, A., van Laere, J., Niklasson, L., Nilsson, M., Persson, A., and Ziemke, T. (2007). On the definition of information fusion as a field of research. Technical report, Institutionen för kommunikation och information.

Chan, T. Y., Partin, A. W., Walsh, P. C., and Epstein, J. I. (2000). Prognostic significance

of gleason score 3+4 versus gleason score 4+3 tumor at radical prostatectomy. *Urology*, 56(5):823 – 827.

Dasarathy, B. V. (2003). Information fusion, data mining, and knowledge discovery. *Information Fusion*, 4(1).

Dhillon, I. S. (2001). Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pages 269–274, New York, NY, USA. ACM.

Dhillon, I. S., Mallela, S., and Modha, D. (2003). Information-theoretic c-oclustering. *In Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 89–98.

Dice, L. R. (1945). Measures of the amount of ecologic association between species. *Ecology*, 26:297–302.

Dimitriadou, E., Weingessel, A., and Hornik, K. (2002). A combination scheme for fuzzy clustering. In Pal, N. and Sugeno, M., editors, *Advances in Soft Computing (AFSS 2002)*, volume 2275 of *Lecture Notes in Computer Science*, pages 332–338. Springer Berlin Heidelberg.

Faouzi, N.-E. E., Leung, H., and Kurian, A. (2011). Data fusion in intelligent transportation systems: Progress and challenges a survey. *Information Fusion*, 12(1):4 – 10. Special Issue on Intelligent Transportation Systems.

Gao, B., Liu, T., Zheng, X., Cheng, Q., and Ma, W. (2006). Consistent bipartite graph copartitioning for star structured high-order heterogeneous data co-clustering. *In Proceedings of the 6th IEEE International Conference on Data Mining (ICDM)*, pages 1–31.

Google (2015). Explore trends. `http://www.google.com/trends/?hl=en-GB`. Accessed: 2015-04-24.

Greene, P. and Cunningham, P. (2009). A matrix factorization approach for integrating multiple data views. *In Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part I*, pages 423–438.

Hall, D. and Llinas, J. (1997). An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23.

Hays, J. and Efros, A. A. (2007). Scene completion using millions of photographs. In *ACM SIGGRAPH 2007 Papers*, SIGGRAPH '07, New York, NY, USA. ACM.

Huang, A. (2008). Similarity Measures for Text Document Clustering. In Holland, J., Nicholas, A., and Brignoli, D., editors, *New Zealand Computer Science Research Student Conference*, pages 49–56.

Inc., F. (2015). The world's most powerful celebrities. `http://www.forbes.com/`. Accessed: 2015-04-24.

Jaccard, S. (1908). Nouvelles researches sur la distribution florale. *Bull. Soc. Vaud. Sci. Nat*, 44:223–270.

Kaufman, L. and Rousseeuw, P. J. (1987). Clustering by means of medoids. In Dodge, Y., editor, *Statistical Data Analysis Based on the L1 - Norm and Related Methods*, pages 405–416. Springer Berlin Heidelberg, North-Holland.

Kaufman., L. and Rousseeuw, P. J. (1990). *Finding Groups In Data. An Introduction To Cluster Analysis*. Wiley-Interscience, New York.

Khaleghi, B., Khamis, A., Karray, F. O., and Razavi, S. N. (2013). Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, 14(1):28 – 44.

Lanckriet, G. R. G., Bie, T. D., Cristianini, N., Jordan, M. I., and Noble, W. S. (2004a). A statistical framework for genomic data fusion. *Bioinformatics*, 20(16):2626–2635.

Lanckriet, G. R. G., Cristianini, N., Bartlett, P., Ghaoui, L. E., and Jordan, M. I. (2004b). Learning the kernel matrix with semidefinite programming. *J. Mach. Learn. Res.*, 5:27–72.

Laney, D. (2001). 3D data management: Controlling data volume, velocity, and variety. Technical report, META Group.

Larsen, B. and Aone, C. (1999). Fast and effective text mining using linear-time document clustering. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '99, pages 16–22, New York, NY, USA. ACM.

Li, X., Wu, C., Zach, C., Lazebnik, S., and Frahm, J.-M. (2008). Modeling and recognition of landmark image collections using iconic scene graphs. In *Proceedings of the 10th European*

Conference on Computer Vision: Part I, ECCV '08, pages 427–440, Berlin, Heidelberg. Springer-Verlag.

Liang, P. and Klein, D. (2009). Online EM for unsupervised models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 611–619, Stroudsburg, PA, USA. Association for Computational Linguistics.

Long, B., Zhang, Z., Wu, X., and Yu, P. S. (2006). Spectral clustering for multi-type relational data. In *ICML*, pages 585–592.

Ma, H., Yang, H., Lyu, M. R., and King, I. (2008). Sorec: Social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 931–940, New York, NY, USA. ACM.

Manjunath, T. N., Hegadi, R. S., and Ravikumar, G. K. (2010). A survey on multimedia data mining and its relevance today. *International Journal Of Computer Science And Network Security (IJCSNS)*, 10(11):165–170.

Maragos, P., Gros, P., Katsamanis, A., and Papandreou, G. (2008). Cross-modal integration for performance improving in multimedia: A review. In Maragos, P., Potamianos, A., and Gros, P., editors, *Multimodal Processing and Interaction*, volume 33 of *Multimedia Systems and Applications*, pages 1–46. Springer US.

Mojahed, A. (2015). Heterogeneous data: data mining solutions. `http://amojahed.wix.com/heterogeneous-data`. Accessed: 2015-08-30.

Mojahed, A., Bettencourt-Silva, J., Wang, W., and de la Iglesia, B. (2015). Applying clustering analysis to heterogeneous data using similarity matrix fusion (smf). In Perner, P., editor, *Machine Learning and Data Mining in Pattern Recognition*, volume 9166 of *Lecture Notes in Computer Science*, pages 251–265. Springer International Publishing.

Mojahed, A. and De La Iglesia, B. (2014). A fusion approach to computing distance for heterogeneous data. In *Proceedings of the sixth International Conference on Knowledge Discover and Information Retrieval (KDIR 2014)*, pages 269–276, Rome, Italy. SCITEPRESS.

Ng, R. and Han, J. (1994). Efficient and effective clustering methods for spatial data mining. *In Proceedings of the 20th Conference on VLDB*, pages 144–155.

NICE (2014). Prostate cancer: diagnosis and treatment. *NICE clinical guideline 175*, pages 1–48.

Oliva, A. and Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. J. Comput. Vision*, 42(3):145–175.

Park, H.-S. and Jun, C.-H. (2009). A simple and fast algorithm for k-medoids clustering. *Expert Syst. Appl.*, 36(2):3336–3341.

Pavlidis, P., Cai, J., Weston, J., and Noble, W. S. (2002). Learning gene functional classifications from multiple data types. *Journal of Computational Biology*, 9(2):401–411.

Rand, W. M. (1958). "objective criteria foe the evaluation of clustering methods". *Journal of the American Statistical Association*, 66(336):846–850.

Ratanamahatana, C. A. and Keogh, E. (2005). Three myths about dynamic time warping data mining. *Proceedings of SIAM International Conference on Data Mining (SDM05)*, pages 506–510.

Reuters, T. (2015a). ISI Web of Knowledge: Journal Citation Reports. `http://wokinfo.com/products_tools/analytical/jcr/`. Accessed: 2015-04-14.

Reuters, T. (2015b). Web of Science. `http://apps.webofknowledge.com/WOS_GeneralSearch_input.do?product=WOS&SID=P1JvWUMqY5wYpc8EIER&search_mode=GeneralSearch`. Accessed: 2015-04-14.

Salton, G. and McGill, M. J. (1987). *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, NY, USA.

Shi, Y., Falck, T., Daemen, A., Tranchevent, L.-C., Suykens, J. A. K., De Moor, B., and Moreau, Y. (2010). L2-norm multiple kernel learning and itsapplication to biomedical data fusion. *BMC Bioinformatics*, 11:309 – 332.

Society, T. R. H. (2014). Plants. URL: https://www.rhs.org.uk/.

Strehl, A. and Ghosh, J. (2003). Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, 3:583–617.

Žitnik, M. and Zupan, B. (2014). Matrix factorization-based data fusion for gene function prediction in baker's yeast and slime mold. *Systems Biomedicine*, 2:1–7.

van Vliet, M. H., Horlings, H. M., van de Vijver, M. J., Reinders, M. J. T., and Wessels, L. F. A. (2012). Integration of clinical and gene expression data has a synergetic effect on predicting breast cancer outcome. *PLoS ONE*, 7(7):e40358.

Wang, J., Zeng, H., Chen, Z., Lu, H., Tao, L., and Ma, W. (2003). "recom: Reinforcement clustering of multi-type interrelated data objects". *In Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 274–281.

Wikipedia (2015). Wikipedia: The free encycipedia. `https://en.wikipedia.org/wiki/Main_Page`. Accessed: 2015-04-24.

Yu, S., Moor, B., and Moreau, Y. (2009). Clustering by heterogeneous data fusion: framework and applications. In *NIPS workshop*.

Zeng, H., Chen, Z., and Ma, W. (2002). "a unified framework for clustering heterogeneous web objects". *In Proceedings of the 3rd International Conference on Web Information Systems Engineering (WISE)*, pages 161–172.

Zha, H., Ding, C., and Gu, M. (2001). "bipartite graph partitioning and data clustering". *In Proceedings of the 10th International Conference on Information and Knowledge Management*, pages 25–32.

Author Biographies

Aalaa Mojahed obtained her her BSc in Computer Science at the Faculty of Sciences, King Abdulaziz University (KAU), Jeddah, Saudi Arabia with First Class Honors in 2004 and then 3 years later she received a MSc degree in Advanced Computing Sciences in the School of Computing Sciences at the University of East Anglia (UEA), Norwich, UK. Since 2012, she has worked for the faculty of Computing Sciences, KAU as a lecturer and besides she joined the machine learning group at UEA and started her PhD research in April 2013 in the field of data mining. She has been involved in a number o projects. Her main research interests include data mining, multimedia data, database and algorithms design and analysis.

Beatriz de la Iglesia is currently a Senior Lecturer in the School of Computing Sciences at the University of East Anglia. She obtained her PhD in Computing Sciences in 2001. Since then she has worked on data mining research with particular experience in health care data analysis. She has worked, among other themes, on the analysis of primary care datasets for cardiovascular disease risk evaluation; on text mining of gastroenterology procedural reports to identify key success indicators and on linking data in the secondary care setting in order to create patient-centric databases suitable for clinical research. She has experience of developing new data mining algorithms using optimisation techniques and has over 40 peer reviewed publications.

Correspondence and offprint requests to: Aalaa Mojahed, King Abdulaziz University, Faculty of Computing and Information Technology, Jeddah, Saudi Arabia Email: amojahed@kau.edu.sa