# TOWARDS A CONTEXT AWARE MINING OF USER INTERESTS FOR CONSUMPTION OF MULTIMEDIA DOCUMENTS

*M. Wallace, G. Stamou*

Image, Video and Multimedia Laboratory
Department of Electrical and Computer Engineering
National Technical University of Athens
Heroon Polytechniou 9, 157 73 Zographou, Greece
wallace@image.ntua.gr, gstam@softlab.ece.ntua.gr

**ABSTRACT**

As the annotation of multimedia documents uses multiple descriptors, it is possible to define multiple, semantically meaningful, similarity (or dissimilarity) relations among them. Therefore, for cases such as the mining of user interests for consumption of multimedia documents, based on usage history, where the clustering of documents is necessary, it is important to develop context aware clustering algorithms that are able to handle this type of information. In this paper we explain the relation between context, user interest and the multiple relations; furthermore, we present a clustering algorithm that is able to mine user interests from multi-relational data sets.

## 1. INTRODUCTION

As huge amounts of multimedia documents are nowadays available to all users, it is very important for an *information retrieval system* (IRS) to be able to quickly and correctly identify which documents would satisfy each user's needs. In order to achieve this the IRS needs to study the users' actions over a period of time and mine their likes and dislikes, i.e. the *user profiles*.

User profiles have been widely used in IRSs of textual documents [1]. The emerging MPEG-7 standard also defines descriptors meant to store usage history and user preferences, i.e. user profiles [4]. The issue of the automatic creation of user profiles is still open, as it involves the extraction of conceptual information about humans, in a fully automated manner. Still, it is possible to reduce its difficulty by defining the set of simpler problems/steps that comprise it. In the following we do so for the case of multimedia documents.

A first step is to A) identify the simple, or compound (i.e. groups of), features that make a multimedia document of interest to a user. Examples of such features (in the case of films) are the director, the cast, the type of film (action, comedy etc) and so on. Then, we need to B) define semantic measures of document similarity (or dissimilarity), based on these features (e.g. how similar are two films, as far as the cast is concerned). Next, we need to C) find a semantically meaningful way to partition documents that are of interest to the user, into groups that correspond to his likes/interests. Finally, we need to D) find the minimum parameters necessary to fully describe a user's interest; then extract these parameters from each group of documents formed in step C.

In this paper we tackle the problem of step C. The definition of multiple document similarity (or dissimilarity) measures/relations calls for the use of new mining techniques that are able to handle multi-relational input and produce more meaningful output. In the field of *data mining*, data sets with multiple relations have been studied [3]. These studies consider relations as *contexts* and show that the results of the data mining procedure are indeed context sensitive, i.e. the choice of the correct context is an important one. Unfortunately, although various ways to benefit from the use of the context have been explored, the problem of automated extraction of the context is still open.

This is an important issue for user profiling. The same user may be interested in various documents for different reasons, which implies that it is not possible to use a common, pre-selected context for all cases. The user profiling process needs to be able to automatically identify both the reason a user is interested in each document (context), as well as the corresponding groups of documents.

In this paper we propose a novel *context aware* hierarchical clustering algorithm. Specifically, given a set of elements (such as documents), among which a variety of dissimilarity measures is defined, it produces groups of elements that resemble each other (have small dissimilarities from each other), as far as one or more of the given relations are concerned. The way in which elements in each group are similar to each other, i.e. the context, is also provided as output of the algorithm.

In section 2 we present the general structure of hierarchical algorithms and define a new measure for the *compactness* (i.e. 'goodness') of a cluster. In section 3 we give the definition of context and describe the way to mine the context that relates two clusters. We also propose a way to use this context in order to estimate the similarity among the clusters. As will become obvious in section 2, the definition of this context aware similarity measure is enough for the definition of a context aware clustering algorithm. In section 4 we justify our approach's relevance to multimedia and discuss on its complexity. Finally, in sections 5 and 6, we present experimental results from the application of the proposed algorithm as well as our concluding remarks.

## 2. AGGLOMERATIVE CLUSTERING AND CLUSTER COMPACTNESS

Most clustering methods belong to either of two general methods, partitioning and hierarchical. Partitioning methods create a crisp or fuzzy clustering of a given data set, but require the number of clusters as input. Unfortunately, this information is not known a priori when trying to partition a set of documents, as is the case in user profiling. Miyamoto, for example, clearly states that the "clustering of documents is not dealt with by nonhierarchical methods" [6].

Hierarchical methods are divided in agglomerative and divisive. Of those, the first are the most widely studied and applied. Their general structure is as follows.

1. Turn each available item into a singleton i.e. into a cluster of its own.
2. For each pair of clusters calculate a *compatibility indicator* (CI). The CI is also referred to as cluster similarity, or dissimilarity, measure.
3. Merge the pair of clusters that produced the best CI.
4. Continue at step two, unless termination criteria are met. The trivial (and most popular) termination criterion is that a single cluster should remain.

This process produces a sequence of clusterings, whose cardinality (count of clusters) ranges from $n$ to one, where $n$ is the cardinality of the set of available items. The two key points are the definition of a suitable compatibility indicator, which we address in this paper, and the identification of the optimal terminating step. Various CIs and termination criteria can be found in the literature [8].

The CI for two clusters can be considered as a measure of the 'goodness' of the cluster that would be created by merging them. Therefore, in order to define the CI in a semantically meaningful way, we first need to define a semantically meaningful measure of the 'goodness' of a cluster. We propose the following metric:

$$g(C,r)=\frac{\sqrt[k]{G(C,r)}}{|C|(|C|-1)}=\frac{\sqrt[k]{\sum_{x,y\in C}(r(x,y))^k}}{|C|(|C|-1)}$$

where $r$ is the dissimilarity measure used and $|C|$ is the cardinality of the set/cluster $C$. This can be considered as a generalized variance metric for spaces where the mean value cannot be defined. It is easy to see that when $k=1$, $g(C,r)$ is the mean dissimilarity between elements of cluster $C$, while as $k\to\infty$, $g(C,r)$ approaches the diameter of the cluster $\max_{x,y\in C} r(x,y)$. In general, for different values of $k$, $g(C,r)$ provides an estimation of how compact cluster $C$ is. Therefore, we name the quantity $1/g(C,r)$ *k-compactness* and, depending on the value of $k$, we refer to it as *1-compactnes*, *2-compactness*, ¥*-compactness* etc.

This measure is meaningful enough to be used as a CI for a hierarchical clustering of a set on which a single relation $r$ is defined. Still, as hierarchical clustering algorithms suffer from high complexity [7], it is necessary to define a CI that preserves the descriptive power of $g(C,r)$, while having a smaller complexity. The following is such a measure.

$$f_1(C_1,C_2,r)=\frac{\sqrt[k]{G(C_1\cup C_2,r)-G(C_1,r)-G(C_2,r)}}{|C_1||C_2|}$$

$f_1(C_1,C_2,r)$ provides an estimation of the overall 'deterioration of compactness' that would result from merging clusters $C_1$ and $C_2$. Since, in order to decide which clusters to merge, the order of the CIs will be considered (and not their ratios), we can equivalently use the following, simpler measure.

$$f(C_1,C_2,r)=\frac{G(C_1\cup C_2,r)-G(C_1,r)-G(C_2,r)}{|C_1||C_2|}$$

This can be calculated using the formula

$$f(C_1,C_2,r)=\frac{\sum_{x\in C_1,y\in C_2}(r(x,y))^k}{|C_1||C_2|}$$

The calculation of $f(C_1,C_2,r)$ has a complexity of $O(|C_1||C_2|)$. It is a great improvement over the complexity of the calculation of $g(C_1\cup C_2,r)$, which is $O((|C_1|+|C_2|)^2)$.

## 3. INTERESTS AND CONTEXTS

In order to facilitate the explanation of the meaning of context in the handling of multimedia documents, let us first restrict to the case of films. It is obvious that a user may be interested in a set of films for a variety of reasons. For example, he may be interested in one set of films (Set I) because his favorite actors appear in them and in another one (Set II) because they are musicals. Therefore, in order to identify the user's interests, it is not enough to partition his favorite films in groups; we also need to find the reason for the existence of each group.

Let us suppose that one of the relations defined on the set of films is a dissimilarity measure related to the actors that participate in each film. It makes sense to assume that the documents in Set I have small dissimilarities, as far as this relation is concerned, while there is no reason to suppose the same for the films in Set II. This observation allows us to use the measures defined in the previous section in order to mine the context, as explained in the following.

The CI defined in the previous section is based on a dissimilarity relation $r$ defined on the set of items to be clustered. If $m$ different relations $r_1,r_2,...,r_m$ are defined, then, for every pair of clusters, $m$ different CIs $f(C_1,C_2,r_1), f(C_1,C_2,r_2),..., f(C_1,C_2,r_m)$ may be defined. Each relation can be considered as a context and each CI as an indication of how similar the two candidate clusters are, as far as the corresponding context is concerned. A simple, but meaningful, approach is to select as correct context the relation that produces the best CI.

This choice of context assumes that the user's interest is always described perfectly by exactly one of the available relations. Of course, such an assumption is quite constraining, as more than one relations might be necessary in order to describe an interest. For example, a user might be interested in a group of films because they 1) describe love stories and 2) have been shot in Greece. In order to be able to handle such cases efficiently we need to describe contexts in a more versatile way (a general definition of *formal context* can be found in [2]).

Let $e^{\mathrm{T}}\in R^m$ be the vector

$e^{\mathrm{T}} = [1,1,...1]^{\mathrm{T}}$

Then, we can define the space of contexts $R_{\mathrm{int}}^{m}$ as

$$R_{\mathrm{int}}^{m} = \left\{ x \in R_m : e^{\mathrm{T}} x = 1 \wedge x(i) > 0 \right\}$$

where $x(i)$ is the $i$-th element of vector $x$. In other words, we define a context $x$ as a fuzzy set on the space of relations; the scalar cardinality of this fuzzy set is one and the membership degree with which the $i$-th relation participates in the context is denoted as $x(i)$ [5].

Given a context $x$, the CI for two clusters $C_1$, $C_2$ can be defined as

$$f(C_1, C_2, x) = x_l \cdot F(C_1, C_2) \qquad (1)$$

where

$$F(C_1, C_2) = [(f(C_1, C_2, r_1)), ..., (f(C_1, C_2, r_m))]^{\mathrm{T}}$$

$$x_l = [x(1)^l, ..., x(m)^l]^{\mathrm{T}}$$

and $l$ is a real parameter.

The pair $\{C_1 \cup C_2, x\}$ corresponds to a user's interest; it contains not only the set of documents $C_1 \cup C_2$ that describe it, but also the reasons $x$ that relate these documents to the user's interest (i.e. the context).

Obviously, for every pair of clusters $C_1$, $C_2$, the corresponding context $x$ is the one that produces the best (minimum) CI. Therefore, the mining of the context, as well as the calculation of the CI, becomes an optimization problem. If $l = 1$, then the solution is trivial and the result is a crisp context (only one relation participates); as $l$ increases the results become fuzzier.

When $l > 1$, if none of the elements of $F(C_1, C_2)$ is equal to zero, it is easy to prove that the solution is given in the following equations.

$$x(m) = \frac{1}{\sum_{i=1}^{m} \left( \frac{F(C_1, C_2, r_m)}{F(C_1, C_2, r_i)} \right)^{\frac{1}{l-1}}}$$

$$x(i) = x(m) \left( \frac{F(C_1, C_2, r_m)}{F(C_1, C_2, r_i)} \right)^{\frac{1}{l-1}}, \quad i = 1,2,...,(m-1)$$

If exactly one element of $F(C_1, C_2)$ is equal to zero, then only the corresponding relation participates in the definition the context. In this case $x$ lies on one of the axes of $R^m$. If more than one elements of $F(C_1, C_2)$ are equal to zero, then the solution is not unique; any relation $r_i$ for which $F(C_1, C_2, r_i) = 0$ may participate in the context, to any degree. Therefore, it only makes sense to have them all participate equally.

Concluding this section, we summarize that using the above formulas to calculate $x$, and then applying (1), we can calculate a context aware CI. Therefore, we can perform a context aware hierarchical clustering applying the general schema presented in section 2.

## 4. RELEVANCE TO MULTIMEDIA AND COMPLEXITY

The proposed algorithm appears to be a generic data mining technique with no specialized multimedia features. Still, it is ideal for the problem of mining user preferences for consumption of multimedia documents; in this field, the problem of Step A, as it is described in section 1, is trivial, as most archives of multimedia documents already store an extended set of (meta) data for each document. Most of this data is labeled, i.e. assumes values from a restricted finite set. For example, there is finite number of directors. Therefore, the problem of Step B may also be easily solved by defining similarity or dissimilarity relations amongst such labels. For example, if we define a dissimilarity relation among directors, we automatically have the dissimilarity of any two films, based on their director.

Another aspect of the algorithm that appears to be weak is its complexity. As all hierarchical clustering algorithms, it has a rather high complexity, which makes it inappropriate for use in cases where large amounts of data need to be processed real time. Still, the problem of user profiling, as described in section 1, is not one that has to be solved real time. Moreover, the complexity of the algorithm is found mainly in the calculation of the CI for each relation, rather than the consideration of the context, which is done with linear complexity. Therefore, the use of a simpler CI, such as minimum or total linkage (i.e. selection of $k \to -\infty$ or $k \to +\infty$) can further enhance our approach.

## 5. EXPERIMENTAL RESULTS

The proposed algorithm was simulated in a Java environment and applied to a data set of 14 films named A, B, C,…,N. Three dissimilarity relations were defined on this set: dissimilarity based on the cast (Relation A), dissimilarity based on the director (Relation B) and dissimilarity based on the type of film (Relation C). The parameter values used were $k = l = 2$.

The dissimilarity relations defined on the data set are presented in Table 2. With gray shading we indicate the clusters that exist in the data. Table 1 presents the algorithm's output after step 11.

| Step 11 | Context | | |
|---|---|---|---|
| Cluster | Relation A | Relation B | Relation C |
| ABCDE | 0.6034 | 0.2384 | 0.1583 |
| FGHIJ | 0.5237 | 0.4312 | 0.04525 |
| KLMN | 0.0320 | 0.0320 | 0.9360 |

**Table 1.** The algorithm's output after step 11

We can see that in the 11th step the algorithm has successfully identified the three clusters in the data. Furthermore, it has correctly indicated that relation C, i.e. the type of the film, dominates the context for cluster 'KLMN', while both relation A and relation B, i.e. both the cast and the director, have an important influence in the context for cluster 'FGHIJ'.

The execution of the algorithm with this data set was fast enough for consideration of usage in cases where real time processing is needed. Still, as a specific user profile may be quite larger, this should be avoided.

| Relation A (cast), Relation B (director), Relation C (type of film) | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | A | B | C | D | E | F | G | H | I | J | K | L | M | N |
| A | 0,0,0 | 0,3,6 | 1,2,5 | 3,3,6 | 1,4,4 | 3,3,3 | 7,7,7 | 3,3,3 | 7,7,7 | 3,3,3 | 7,7,7 | 3,3,3 | 7,7,7 | 8,3,8 |
| B | 0,3,6 | 0,0,0 | 2,4,8 | 0,2,7 | 3,5,2 | 7,7,7 | 6,6,6 | 7,7,7 | 6,6,6 | 7,7,7 | 6,6,6 | 7,7,7 | 6,6,6 | 7,7,7 |
| C | 1,2,5 | 2,4,8 | 0,0,0 | 1,4,3 | 1,3,3 | 5,5,5 | 3,3,3 | 5,5,5 | 3,3,3 | 5,5,5 | 3,3,3 | 5,5,5 | 8,8,8 | 5,5,5 |
| D | 3,3,6 | 0,2,7 | 1,4,3 | 0,0,0 | 4,2,2 | 7,7,7 | 5,5,5 | 7,7,7 | 5,5,5 | 7,7,7 | 5,5,5 | 7,7,7 | 5,5,5 | 7,7,7 |
| E | 1,4,4 | 3,5,2 | 1,3,3 | 4,2,2 | 0,0,0 | 5,5,5 | 7,7,7 | 5,5,5 | 7,7,7 | 5,5,5 | 7,7,7 | 5,5,5 | 7,7,7 | 5,5,5 |
| F | 3,3,3 | 7,7,7 | 5,5,5 | 7,7,7 | 5,5,5 | 0,0,0 | 1,2,6 | 1,1,6 | 1,0,5 | 1,1,4 | 3,3,3 | 7,7,7 | 3,3,3 | 7,7,7 |
| G | 7,7,7 | 6,6,6 | 3,3,3 | 5,5,5 | 7,7,7 | 1,2,6 | 0,0,0 | 1,1,4 | 1,3,3 | 0,0,5 | 5,5,5 | 3,3,3 | 5,5,5 | 8,8,8 |
| H | 3,3,3 | 7,7,7 | 5,5,5 | 7,7,7 | 5,5,5 | 1,1,6 | 1,1,4 | 0,0,0 | 0,0,5 | 3,1,6 | 2,2,2 | 5,5,5 | 2,2,2 | 5,5,5 |
| I | 7,7,7 | 6,6,6 | 3,3,3 | 5,5,5 | 7,7,7 | 1,0,5 | 1,3,3 | 0,0,5 | 0,0,0 | 1,1,7 | 3,3,3 | 7,7,7 | 3,3,3 | 7,7,7 |
| J | 3,3,3 | 7,7,7 | 5,5,5 | 7,7,7 | 5,5,5 | 1,1,4 | 0,0,5 | 3,1,6 | 1,1,7 | 0,0,0 | 7,7,7 | 3,3,3 | 7,7,7 | 8,8,8 |
| K | 7,7,7 | 6,6,6 | 3,3,3 | 5,5,5 | 7,7,7 | 3,3,3 | 5,5,5 | 2,2,2 | 3,3,3 | 7,7,7 | 0,0,0 | 6,6,1 | 7,7,0 | 4,4,1 |
| L | 3,3,3 | 7,7,7 | 5,5,5 | 7,7,7 | 5,5,5 | 7,7,7 | 3,3,3 | 5,5,5 | 7,7,7 | 3,3,3 | 6,6,1 | 0,0,0 | 4,4,1 | 5,5,1 |
| M | 7,7,7 | 6,6,6 | 8,8,8 | 5,5,5 | 7,7,7 | 3,3,3 | 5,5,5 | 2,2,2 | 3,3,3 | 7,7,7 | 7,7,0 | 4,4,1 | 0,0,0 | 7,7,1 |
| N | 8,3,8 | 7,7,7 | 5,5,5 | 7,7,7 | 5,5,5 | 7,7,7 | 8,8,8 | 5,5,5 | 7,7,7 | 8,8,8 | 4,4,1 | 5,5,1 | 7,7,1 | 0,0,0 |

**Table 2.** The dataset used for the simulation

The algorithm's efficiency and overall performance could not be compared to those of other approaches, as no other context mining clustering techniques have been met in the literature by the authors.

## 6.CONCUSIONS AND FUTURE WORK

In this paper we have presented a context aware hierarchical clustering algorithm. It is suitable for applications in which the context is an important factor and the number of clusters is not known a priori; an example of such applications is user profiling and, more specifically, the mining of user interests.

This algorithm may be further enhanced by investigating, for example, ways to further reduce its complexity, without ruining the semantic content of its output.. Other related research areas include the definition of meaningful similarity and dissimilarity relations / measures between multimedia documents, the efficient use of user interests for the enhancement of an IRS's performance and more. All these research topics can contribute to the creation of a new IRS, able to significantly aid the user during his searches, with the use of his profile.

The development of such an IRS is one of the goals of the EU FAETHON IST project, in which the authors participate.

## 7. ACKNOWLEDGMENTS

This work was partially funded by the EC IST-1999-20502 Project.

The authors would also like to express their gratitude to the anonymous reviewers for their helpful comments.

## 8. REFERENCES

[1] P. M. Chen and F. C. Kuo, "An information retrieval system based on a user profile," *The Journal of Systems and Software*, vol. 54, pp. 3-8, 2000.

[2] B. Ganter and R. Wille, *Formal Concept Analysis*, Springer, 1999.

[3] K. Hirota and W. Pedrycz, "Fuzzy Computing for Data Mining," *Proceedings of the IEEE*, vol. 87 (9), pp. 1575-1600, 1999.

[4] ISO/IEC JTC1/SC29/WG11 N4032, "Introduction to MPEG-7" Singapore, March 2001.

[5] G. J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic, Theory and Applications*, Prentice Hall, New Jersey, 1995.

[6] S. Miyamoto, *Fuzzy Sets in Information Retrieval and Cluster Analysis*, Kluwer Academic Publishers, Dordrecht / Boston / London, 1990.

[7] Y. El-Sonbaty and M. A. Ismail, "On-line hierarchical clustering" *Pattern Recognition Letters*, vol. 19, pp. 1285-1291, 1998.

[8] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, 1998.