DIMACS Technical Report 2007-13
September 2007

ACM Classification Can Be Used for Representing Research Organizations

by

Boris Mirkin[1],
School of Computer Science
Birkbeck University of London
London, UK WC1E 7HX

Susana Nascimento[2] and Luis Moniz Pereira
Computer Science Department and Centre for Artificial Intelligence (CENTRIA)
FCT, Universidade Nova de Lisboa
Caparica, Portugal

[1]Several visits to DIMACS 2002–2005 contributed to this work; a visit to DIMACS in August 2007 helped to finalize the report.
[2]Supported by the grant PTDC/EIA/69988/2006 from the Portuguese Science & Technology Foundation

---

ABSTRACT

We present a method for representation a Computer Science Research organization by using the ACM Computing Subjects classification tree. The representation comprises head subjects of the upper level as well as their gaps and offshoots found by parsimoniously mapping main subject clusters, extracted from the data on similarity ACM research topics according to the working in the organization, onto the ACM classification. A robust method for possibly overlapping clustering is described. A real-world example of the representation is provided.

Contents

1 Introduction: ACM Computing Classification System as a domain ontology fit for representing

ACM Computing Classification System (ACMC) is a conceptual three level classification of the Computer Science subject area built to reflect the vast and changing world of computer oriented writing. This classification was first published in 1991 and then thoroughly revised in 1998 and it is being revised since [1]. It comprises eleven major partitions (first-level subjects):

A. General Literature

B. Hardware

C. Computer Systems Organization

D. Software

E. Data

F. Theory of Computation

G. Mathematics of Computing

H. Information Systems

I. Computing Methodologies

J. Computer Applications

K. Computing Milieux

These are subdivided into 81 second-level topics. For example, item I. Computing Methodologies consists of eight subjects:

I.0 GENERAL

I.1 SYMBOLIC AND ALGEBRAIC MANIPULATION

I.2 ARTIFICIAL INTELLIGENCE

I.3 COMPUTER GRAPHICS

I.4 IMAGE PROCESSING AND COMPUTER VISION

I.5 PATTERN RECOGNITION

I.6 SIMULATION AND MODELING (G.3)

I.7 DOCUMENT AND TEXT PROCESSING (H.4, H.5)

which are further subdivided into third-layer topics as, for instance, I.5 PATTERN RECOGNITION which consists of seven topics:

I.5.0 General

I.5.1 Models

I.5.2 Design Methodology

I.5.3 Clustering

- Algorithms
- Similarity measures

I.5.4 Applications

I.5.5 Implementation (C.3)

I.5.m Miscellaneous

These are further subdivided in unlabeled subtopics such as those two shown for topic I.5.3 Clustering.

As can be seen from the examples above, there are a number of collateral links between topics both on the second and the third layers - they are in the parentheses in the ends of some topics.

The ACMC is used, mainly, as a device for annotation and search for publications in collections such as that on the ACM portal [1], that is, for the library and bibliographic applications. Meanwhile, as an adequate domain ontology, it can and should be used for other means as well. The ACMC tree has been applied as a gold standard for ontologies derived by web mining systems such as the CORDER engine [2]. The ACMC tree is used in determining the semantic similarity in information retrieval [3] and an e-Learning applications [4] as well as for matching software practitioners' needs and software researchers' activities [5]. Here we propose its use for representing research organizations in such a way that the organization's research topics are parsimoniously mapped onto the ACMC topology.

The art of representation of various items on an ontology is of interest in many areas such as text analysis, web mining, bioinformatics and genomics. In web mining representations are extracted from domain ontologies: the ontologies are used to automatically characterize usage profiles by describing user's interests and preferences for web personalisation [6]. There are also recommender systems for on-line academic research papers [7], which extract user profiles based on an ontology of research topics. In bioinformatics several clustering techniques

have been successfully applied in the analysis of gene expression profiles and gene function prediction incorporating gene ontology information into clustering algorithms [8].

However, this line of thinking has never been applied to representing research organizations. The very idea of representing research organizations may seem rather odd because conventionally it is only the accumulated body of results that does matter in the sciences, and these always have been and still are provided by individual efforts. The assumption of individual research efforts implicitly underlies systems for reviewing and comparing different research departments in countries such as the United Kingdom in which scientific organizations are subject to regular comprehensive review and evaluation practices. The evaluation is based on the analysis of individual researchers' achievements, leaving the portrayal of a general picture to subjective declarations by the departments [9]. Such an evaluation provides for the assessment of relative strengths among different departments, which is good for addressing issues of funding. There exists yet another aspect, that of the integral portrayal rather than comparative analysis of the developments. This aspect is important for decisions regarding long-term or wide-range issues of scientific development such as national planning or addressing the so-called 'South–North divide' between developed and underdeveloped countries. The latter would require comparing between integral systems of scientific scope and capabilities of scientific organizations and university departments in both the South and North (see, for instance, The United Nations Millennium Project task force web-site [10]).

Potentially, the ACMC representations can be used for such actions as:

i Overview of scientific subjects being developed in an organization.

ii Positioning the organization over ACMC.

iii Overview of scientific disciplines being developed in organizations over a country or other territorial unit, with a quantitative assessment of controversial subjects, for example, those in which the level of activity is not sufficient or the level of activities by far excesses the level of results.

iv Assessing the scientific issues in which the character of activities in organizations does not fit well onto the classification; these can be potentially the growth points or other breakthrough developments.

v Planning research restructuring and investment.

2 Ontology representation of a subject cluster

In our approach, we consider a research organization as consisting of one or several groupings of people working together on scientific problems of their interest. A grouping may be not necessarily formal nor needs it to consist of more than a single researcher. Each of the groupings can be represented by a set of ACMC topics of the third or second level - those pertaining to the problems of interest to the grouping; we refer to this set as a subject

cluster. Mapping a subject cluster to the ACMC may lead to different portrayals of that on the ACMC tree whose root corresponds to the entire field of Computer Science. A cluster can fit quite well into the classification or not (see Figure 1), depending on how much its topics are dispersed among the tree nodes.

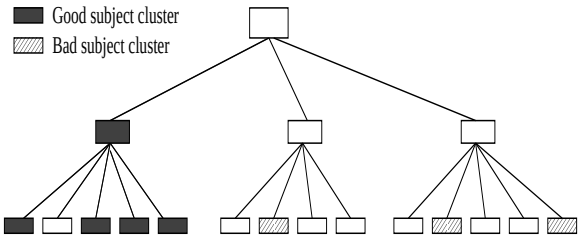

Figure 1: Two clusters of second-layer topics, presented with checked and diagonal lined boxes, respectively. The check box cluster fits all within one first -level category (with one gap only), whereas the diagonal line box cluster is dispersed among two categories on the right. The former fits the classification well; the latter does not fit at all.

The best possible fit would be when all topics in the subject cluster fall within a parental node in such a way that all the siblings are covered and no gap occurs. The parental tree node, in this case, can be considered as the head subject of the cluster. A few gaps, that is, head subject children topics not included in the cluster, although diminish the fit, still leave the head subject unchanged. A larger misfit occurs when a cluster is dispersed among two or more head subjects. One more type of misfit may emerge when almost all cluster topics fall within the same head subject node but one or two of the topics offshoot to other parts of the classification tree (see Figure 2).

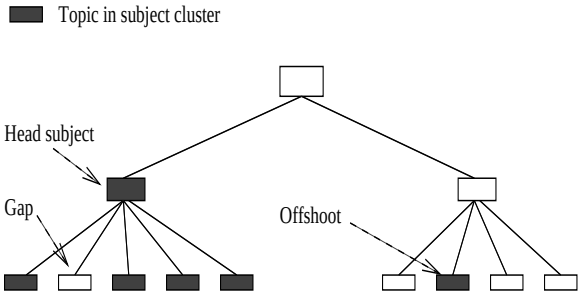

Figure 2: Three types of features of mapping of a subject cluster to the ontology.

Such offshoots, when persist at subject clusters in different organizations, may show some tendencies in the development of the science, that the classification tree has not taken into account yet. The total count of head subjects, gaps and offshoots, each type weighted accordingly, can be used for scoring the extent of the fit between a research grouping and the classification tree as illustrated on Figure 3. The greater the score, the worse the fit.

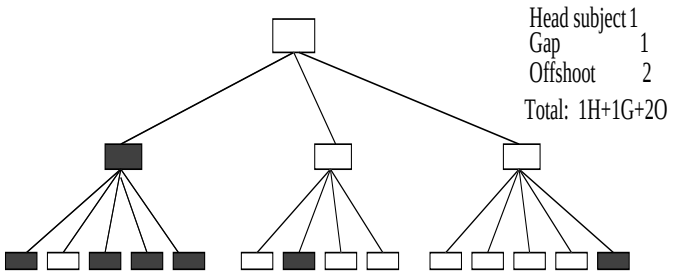

Figure 3: Scoring a mapping of subject cluster to the ontology.

When the topics under consideration relate to deeper levels of classification, such as the third layer of ACMC, the scoring may allow some tradeoff between different possibilities for selecting the head subjects. Such a case is presented on Figure 4. The subject cluster consists of third-layer topics presented by checked boxes. The cluster can be considered as pertaining to two head subjects as on (A) or, just one, the upper category on (B), with the "cost" of three more gap nodes added, and one offshoot subtracted. Depending on the relative weighting of gaps, offshoots and multiple head subjects, either mapping can minimize the total misfit. In fact, the gaps and offshoots are determined by the head subjects specified.

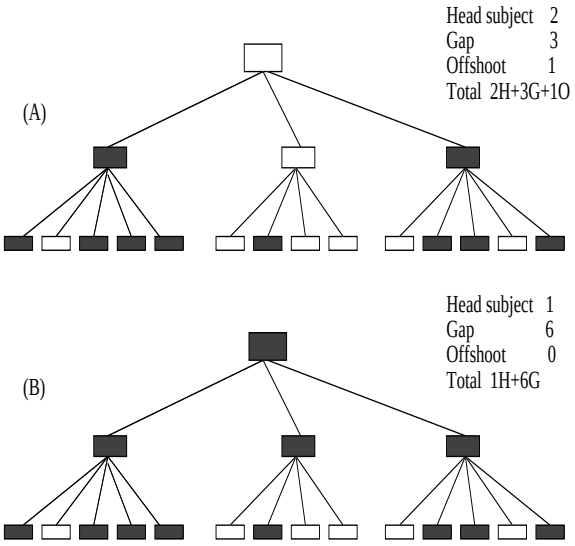

Figure 4: Tradeoff between different mappings of the same subject cluster: mapping (B) is better than (A) if gaps are much cheaper than additional head subjects.

Altogether, the set of subject clusters, their head subjects, offshoots and gaps constitutes what can be referred to as a profile of the organization in consideration. Such a representation can be easily accessed and expressed as an aggregate. It can be further elaborated by highlighting representation subjects in which the organization members have been especially

successful (i.e., publication in best journals, award or other recognition) or distinguished by another feature (i.e., industrial product or inclusion to a teaching program).

The problem of parsimoniously mapping, that is, minimizing the total weight, for a cluster was considered in [11] for a special application in genomics with a different weight function.

Building a parsimonious mapping of a subject cluster can be achieved by recursively building a parsimonious scenario for each node of the ACMC tree based on parsimonious scenarios for its children. For the sake of simplicity, let us tackle a simplified problem in which offshoots are not considered, that is, weight of an offshoot O is zero. Getting a head subject will be referred to as a "head gain". At each node of the tree, sets of gap and head gain events are to be determined and iteratively raised to the parents under each of two different assumptions that specify the situation "above the parent"..

One assumption is that the head subject has been inherited at the parental node from its own parent, and the second assumption is that it has not been inherited but gained in the node only. It is necessary to distinguish these two cases since, clearly, it is only meaningful to consider the loss of a head subject at a node if it was inherited at that node; similarly, it is only meaningful to consider the gain of a head if it was not inherited. These assumptions can be considered as parallel to those in the 2-state Markov chain probabilistic modeling: each corresponding to a different state of the chain: head subject inherited from above or not. Consider the parent-children system as shown in Figure 5, with each node assigned with sets of gap and head gain events under the above two inheritance assumptions.

Let us denote the total number of events under the inheritance and non-inheritance assumptions by e_i and e_n, respectively, where head gains are weighted by the head penalty h and gaps by the gap penalty g formerly denoted by H and G, respectively (either can be taken to be unity; see discussion in [11]). A mapping result at a given node is defined by a pair of sets (H, G), representing the tree nodes at which events of head gains and gaps, respectively, have occurred in the subtree rooted at the node. We use (Hi, Gi) and (Hn, Gn) to denote mapping results under the inheritance and non-inheritance assumptions, respectively. Our results for the case of a binary tree [11], can be extended to arbitrary taxonomies to derive the parental inconsistency score from those of its children, in a parsimonious scenario, under the inheritance or non-inheritance assumption, respectively. At a leaf node the four sets Hi, Gi, Hn and Gn are empty, except that Hn $=\{a\}$ if topic cluster a is present in the given leaf or Gi $=\{a\}$ if a is not present. The algorithm then will compute parsimonious scenarios for parental nodes according to the topology of the tree, proceeding from the leaves to the root in the manner described in [11]. Specifically, in a parsimonious scenario, the total number of events, weighted by h and g, can be derived from those of its children (indicated by subscripts 1, 2 and 3 for the case of three children on Figure 5) as $e_i = \min(e_{n1} + e_{n2} + e_{n3} + g,\ e_{i1} + e_{i2} + e_{i3})$ or $e_n = \min(e_{i1} + e_{i2} + e_{i3} + h,\ e_{n1} + e_{n2} + e_{n3})$, under the inheritance or non-inheritance assumption, respectively; the proof given in [11] for the binary tree case can be easily extended to an arbitrary rooted tree.

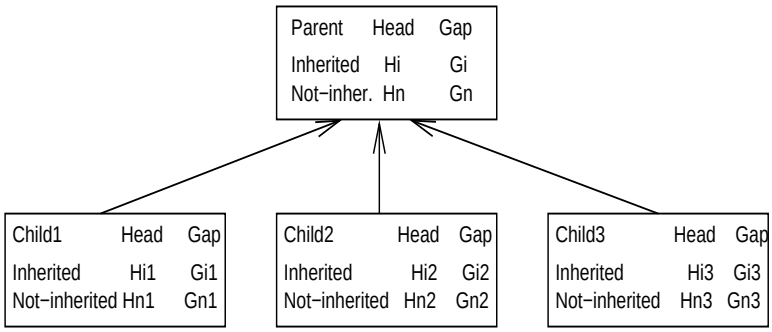

Figure 5: Events in a parent-children system according to a parsimonious scenario.

3 Building a subject cluster

3.1 Similarity clustering: A review

We consider that members of each research organization are engaged in a number of research projects that are relevant to some of ACMC topics. This should give rise to a measure of similarity between the topics. Specifically, an index of similarity between two topics should be proportional to the number and importance of the projects that are relevant to both of the topics. The degrees of relevance should be used as weighting coefficients to the projects. Therefore, we assume that a survey of the projects of members of the organization can be conducted in such a way that a set of ACMC topics that are worked on in the organization can be discerned, along with a matrix of similarity indices between the topics. In this way, the issue of determining of the subject clusters can be explicated as the well-known problem of finding clusters, potentially overlapping, over a similarity matrix.

Similarity clustering emerged rather early – in graph theory, probably before the discipline of clustering itself. A graph may be thought of as a structural expression of similarity data, its nodes corresponding to entities with edges joining similar nodes. Cluster related graph-theoretic concepts include: (a) *connected component* (a maximal subset of nodes in which every pair of nodes is connected by a path), (b) *bicomponent* (a maximal subset of nodes in which each pair of nodes belongs to a cycle), and (c) *clique* (a subset of nodes in which each pair of nodes is connected by an edge).

Other, less straightforward, early clustering concepts include the B-coefficient method for clustering variables using their correlation matrix [12] and the Wrozlaw taxonomy [13]. These are precursors to the ADDI and ADDI-S methods [14], described later, and the single linkage method [15], respectively. Two more recent graph-theoretic concepts are also relevant: *maximum density subgraph* [16] and *min-multi-cut* in a weighted graph [17].

The density $g(S)$ of a subgraph on $S \subset I$, where I is the set of all vertices is the ratio of the number of edges in S to the number of elements of S. For an edge weighted graph with weights specified by the matrix $A = (a_{ij})$, the density $g(S)$ is equal to the *Raleigh quotient* $s^T As/s^T s$, where $s = (s_i)$ is the membership vector of S, viz. $s_i = 1$ if $i \in S$ and

$s_i = 0$ otherwise. A subgraph of maximum density represents a cluster. After removing such a cluster from the graph, a maximum density subgraph of the remaining graph can be found. This may be repeated until no "significant" clusters remain. To our knowledge, this method has never been applied to real-world problems, probably because it involves rather intensive computations. We consider that the maximum density subgraph problem is of interest because it is a reasonable relaxation of the maximum clique problem and, also, it fits well into the data recovery clustering approach [19]. The maximum value of the Raleigh quotient of a symmetric matrix over any real vector s is equal to the maximum eigenvalue and is attained at an eigenvector corresponding to this eigenvalue. This gives rise to the so-called *spectral clustering*, a method of clustering based on first finding a maximum eigenvector s^* and then defining the spectral cluster by $s_i = 1$ if $s_i^* > t$ and $s_i = 0$ otherwise, for some threshold t. This method may have computational advantages when A is sparse. Unfortunately, it does not necessarily lead to an optimal cluster, but empirically it produces good clusters in most cases.

The concept of min-multi-cut is an extension of the max-flow min-cut concept in capacitated networks, and essentially seeks a partition of nodes into classes having minimum summary similarities between classes or, equivalently, maximum summary similarities within classes. When similarities are non-negative, this criterion may often lead to a highly unbalanced partition with one huge class and a number of singleton classes. This line of research has led to using the *normalized cut*, proposed in [21], as a meaningful clustering criterion. The normalized cut criterion assumes that the set of all vertices I should be split into two parts, S and $\bar{S}$, so that the normalized cut

$$nc(S) = a(S, \bar{S})/a(S, I) + a(S, \bar{S})/a(\bar{S}, I)$$

is minimized. Here $a(S, T)$ denotes the summary similarity between subsets S and T. The criterion $nc(S)$ can be expressed as a Raleigh quotient for a generalized eigenvalue problem [21], so the spectral clustering approach may be applied to minimizing the normalized cut too.

3.2 Additive cluster model and iterative extraction

In the framework of the data recovery approach to clustering, a model for overlapping clustering can be formulated as follows.

Let I be the set of entities, such as ACMC topics, under consideration and $A = (a_{ij})$ a similarity matrix over $i, j \in I$. Any subset $S \subseteq I$ can be one-to-one associated with a "square-like" relation defined by a binary matrix $r = (r_{ij})$ in such a way that $r_{ij} = 1$ if both i and j belong to S and $r_{ij} = 0$, otherwise. In other words, $r_{ij} = s_i s_j$ where $s = (s_i)$ is the membership vector of S so that for any $i \in I$, s_i is 1 or 0 depending on whether $i \in S$ or not. The universal cluster $S^0 = I$ corresponds to the universal relation r^0 whose all entries are unities.

Each cluster $S \subseteq I$ can be assigned with a number λ, expressing the intensity of the cluster, which is interpreted as the intensity of similarities between its elements. Assume

that similarities in A are generated by a set of "additive clusters" $S^k \subseteq I$ along with their intensities λ_k, $k = 0, 1, ..., K$, in such a way that each a_{ij} approximates the sum of the intensities of those clusters that contain both i and j so that:

$$a_{ij} = \sum_{k=1}^{K} \lambda_k s_i^k s_j^k + \lambda_0 + e_{ij}, \tag{1}$$

where $s^k = (s_i^k)$ are the membership vectors of the unknown clusters S^k, $k = 1, 2, ..., K$, and e_{ij} are the residuals to be minimized. The intensity λ_0 of the universal cluster I can be considered as a similarity scale shift because the equation in (1) can be equivalently changed to that without the universal cluster but with shifted similarities $a'_{ij} = a_{ij} - \lambda_0$.

In this additive clustering model, introduced in [18], the intensities λ_k, $k = 1, 2, ..., K$, and the shift λ_0 also have to be optimally determined. In the more general formulation of the "categorical factor analysis" [14], these values may be user specified.

We note that the role of the parameter λ_0 in (1) is threefold: it can be considered as

1. an intercept of the bilinear data model, similar to that in the linear regression or
2. the intensity of the universal cluster I or
3. a 'soft' similarity threshold in the sense that it is the shifted similarity matrix a'_{ij}, rather than the original A, is used to determine the clusters S^k, $k = 1, 2, ..., K$. This role is of a special interest when λ_0 is user specified.

A computationally viable strategy for fitting model (1) is of iterative cluster extraction [14]. According to this strategy, clusters and their intensities are found one by one starting from the universal cluster $S_0 = I$. To find a cluster, model (1) is restricted to $K = 1$ and shifted a'_{ij} for a_{ij}. That means that criterion

$$L^2(S, \lambda) = \sum_{i,j \in I} (a'_{ij} - \lambda s_i s_j)^2 \tag{2}$$

is to be minimized with respect to unknown λ and/or S. After a solution at k-th step is found, similarities are updated by subtracting the cluster similarities taken into account $a'_{ij} \leftarrow a'_{ij} - \lambda s_i s_j$, k is increased by one, and the process reiterates. The optimal λ can be proven to be equal to the average within-S similarity, because of the quadratic nature of the criterion. The initial step involves the universal cluster $S_0 = I$ and its optimal intensity, which is the average value of all the initial similarities a_{ij}.

This procedure does not necessarily lead to the optimal fitting of model (1), but it allows for a useful decomposition of the data scatter, which is analogous to that due to the so-called spectral decomposition of matrices over their eigen vectors in linear algebra. Specifically,

$$(A, A) = \sum_{k=0}^{K} [s^{kT} A^k s^k / s^{kT} s^k]^2 + (E, E) \tag{3}$$

In this formula, the inner products (A, A) and (E, E) denote the sums of the squares of the elements of the matrices, considering A and E as vectors; these are conventionally expressed as the traces (sums of diagonal elements) of the products $A^T A$ and $E^T E$, respectively. The residual simiularity matrix on k-th step is denoted by A^k. The optimal λ_k is equal to $\bar{a}_k$, the average of the residual similarities a^k_{ij} for $i, j \in S^k$.

Similarity data A may not be symmetric. However, it is not difficult to prove that if A is not symmetric, it can be equivalently changed for symmetric $\tilde{A} = (A + A^T)/2$ [14]. For the sake of simplicity, in this section, we assume that the diagonal entries a_{ii} are all zero.

3.2.1 Pre-specified intensity

We first consider the case in which the intensity λ of the cluster to be found is pre-specified. Noting that $s_i^2 = s_i$ for any 0/1 variable s_i, criterion (2) can be expressed as

$$L^2(S) = \sum_{i,j \in I} (a_{ij} - \lambda s_i s_j)^2 = \sum_{i,j \in I} a_{ij}^2 - 2\lambda \sum_{i,j \in I} (a_{ij} - \lambda/2) s_i s_j \tag{4}$$

Since $\sum_{i,j} a_{ij}^2$ is constant, for $\lambda > 0$, minimizing (4) is equivalent to maximizing the summary within-cluster similarity after subtracting the threshold value $\pi = \lambda/2$:

$$f(S, \pi) = \sum_{i,j \in I} (a_{ij} - \pi) s_i s_j = \sum_{i,j \in S} (a_{ij} - \pi). \tag{5}$$

This criterion implies that, for an entity i to be added to or removed from the S under consideration, the difference between the value of (5) for the resulting set and its value for S, $f(S \pm i, \pi) - f(S, \pi)$, is equal to $\pm 2 f(i, S, \pi)$ where

$$f(i, S, \pi) = \sum_{j \in S} (a_{ij} - \pi) = \sum_{j \in S} a_{ij} - \pi |S|.$$

This gives rise to a local search algorithm for maximizing (5): start with $S = \{i^*, j^*\}$ such that $a_{i^* j^*}$ is maximum element in A, provided that $a_{i^* j^*} > \pi$. An element $i \notin S$ may be added to S if $f(i, S, \pi) > 0$; similarly, an element $i \in S$ may be removed from S if $f(i, S, \pi) < 0$. The greedy procedure ADDI [14] iteratively finds an $i \notin S$ maximizing $+f(i, S, \pi)$ and an $i \in S$ maximizing $-f(i, S, \pi)$, and takes the i giving the larger value. The iterations stop when this larger value is negative. The resulting S is returned along with its contribution to the data scatter, $4\pi \sum_{i \in S} f(i, S, \pi)$. The following version of ADDI reducing the dependence on the initial S proved successful in experiments. The computations here start from the singleton $S = \{i\}$, for each $i \in I$, so that $|I|$ ADDI based results are generated; of these, that cluster S is selected that contributes most to the data scatter, i.e., that minimizes the square error $L^2(S)$ (4). In fact, the set of resulting clusters should be of interest on its own since many of them coincide or almost coincide and the structure of not coinciding clusters represents an overlapping structure of the similarity data.

The heuristic algorithm CAST [20], popular in bioinformatics, is in fact a version of the ADDI algorithm, because it uses the same iterative process of adding or removing an entity

by utilizing criterion $\sum_{j \in S} a_{ij} > \pi|S|$, for the case of adding, with the $\sum_{j \in S} a_{ij}$ referred to as the affinity of i to S – which is equivalent to criterion $f(i, S, \pi) > 0$.

Another property of the criterion is that $f(i, S, \pi) > 0$ if and only if the average similarity between a given $i \in I$ and the elements of S is greater than π, which means that the final cluster S produced by ADDI/CAST is rather tight: the average similarities between $i \in I$ and S is at least π if $i \in S$ and no greater than π if $i \notin S$ [14].

Intuitively, changing the threshold π should lead to corresponding changes in the optimal S. Indeed, it has been proven that the greater π is, the smaller S will be [14].

3.2.2 Optimal intensity

When λ in (4) is not fixed but chosen to further minimize the criterion, it is not difficult to prove that

$$L^2(S) = (A, A) - [s^T A s / s^T s]^2, \tag{6}$$

in line with the decomposition (3), with $K = 1$ and $L^2(S) = (E, E)$. The proof is based on the fact that the optimal λ is the average similarity $a(S)$ within S, i.e.,

$$\lambda = a(S) = s^T A s / [s^T s]^2, \tag{7}$$

since $s^T s = |S|$.

The decomposition (6) implies that an optimal cluster S must maximize the criterion

$$g^2(S) = [s^T A s / s^T s]^2 = a^2(S)|S|^2 \tag{8}$$

According to (8), the maximum of $g^2(S)$ may correspond to either positive or negative value of $a(S)$. The latter case may emerge when the similarity shift λ_0 is large and corresponds to S being the so-called *anti-cluster*. In this paper, we do not consider this case, but focus on maximizing (8) for positive $a(S)$ only. This is equivalent to maximizing the Raleigh quotient,

$$g(S) = s^T A s / s^T s = a(S)|S| \tag{9}$$

It should be pointed out that this criterion not only emerges in the data recovery framework but it also fits into some other frameworks such as (i) maximum density subgraphs and (ii) spectral clustering.

To maximize $g(S)$, one may utilize the ADDI-S algorithm [14], which is the same as the algorithm ADDI/CAST, described above, except that the threshold π is recalculated after each step as $\pi = a(S)/2$, corresponding to the optimal λ in (7).

A property of the resulting cluster S, similar to that for the constant threshold case, holds: the average similarity between i and S is at least half the within-cluster average similarity $a(S)/2$ if $i \in S$, and at most $a(S)/2$ if $i \notin S$.

To obtain a set of (not necessarily disjoint) clusters within the framework of the additive clustering model, one can repeatedly extract a cluster S using ADDI-S and then replacing A by the residual matrix $A - a(S)ss^T$.

ADDI-S utilizes no ad hoc parameters, so the process of iterative extraction of clusters can be halted by using either or all of the following criteria:

i A pre-specified number of clusters is reached.

ii A pre-specified proportion of the data scatter (A, A) taken into account by the found clusters according to (3), such as 60%, is reached.

iii Next cluster takes into account less than a pre-specified proportion of the data scatter, such as 3% or 5%.

4 An example of implementation

Let us describe how this approach can be implemented by using the data from a survey at the Department of Computer Science, Faculty of Science & Technology, New University of Lisboa (DI-FCT-UNL).

For simplicity, we use only data of the second level of ACM, each having a code V.v where V=A,B,...,K, and v =1,..,mK, with mK being the number of second level topics. Each member of the department supplied three ACM subjects most relevant to their current research. These comprise altogether 26 of the 59 topics at the second level in ACMC. (We omit two subjects of the second level, General and Miscellaneous, occurred in every first-level division as they do not contribute to the representation.)

We define similarity between two ACM subjects, V.v and W.w, as the number of members of the department that work on both of them. In principle, the measure can be further elaborated by taking into account various structural aspects of the department's structure such as formally defined working groups, grant projects, etc. This measure serves as the base for finding subject clusters. It is important to notice that clusters are not necessarily disjoint; they may overlap.

With the algorithm ADDI-S applied to the 26x26 similarity matrix, we get the following 6 clusters (with the halt at reaching the threshold of 4% contribution to the data scatter):

1. Cl1 (contribution 27.08%, intensity 2.17), 4 items: D3, F1, F3, F4;
2. Cl2 (contribution 17.34%, intensity 0.52), 12 items: C2, D1, D2, D3, D4, F3, F4, H2, H3, H5, I2, I6;
3. Cl3 (contribution 5.13%, intensity 1.33), 3 items: C1, C2, C3;
4. Cl4 (contribution 4.42%, intensity 0.36) , 9 items: F4, G1, H2, I2, I3, I4, I5, I6, I7;
5. Cl5(contribution 4.03%, intensity 0.65), 5 items: E1, F2, H2, H3, H4;

6. Cl6 (contribution 4.00%, intensity 0.64), 5 items: C4, D1, D2, D4, K6.

The next 7th cluster's contribution is just 2.5%. These clusters mapped to the ACMC are presented on Figure 5, in which only those first-level categories that overlap them are shown.

One can see the following:

- The department covers, with a few gaps and offshoots, six head subjects shown on the Figure using pentagns filled in by different patterns;

- The most contributing cluster, with the head subject F. Theory of computation, comprises a very tight group of a few second level topics;

- The next contributing cluster has not one but two head subjects, D and H, and offshoots to every other head subject in the department, which shows that this cluster currently is the structure underlying the unity of the department;

- Moreover, the two head subjects of this cluster come on top of two other subject clusters, each pertaining to just one of the head subjects, D. Software or H. Information Systems. This means that the two-headed cluster signifies a new direction in Computer Sciences, combining D and H into a single new direction, which seems a feature of the current developments indeed; this should eventually get reflected in an update of the ACM classification (by raising D.2 Software Engineering to the level 1?);

- There are only three offshoots outside the department's head subjects: E1. Data structures from H. Information Systems, G1. Numerical Analysis from I. Computing Methodologies, and K6. Management of Computing and Information Systems from D. Software. All three seem natural and should be reflected in the list of collateral links between different parts of the classification tree.

5 Conclusion

We have shown that ACMC can be used as an ontology structure for representing CS research activities. Altogether, to apply this approach to a Computer Science organization such as a University department, one needs to perform the following steps:

- surveying the members of ACMC subjects they are working on; this can be supplemented with indication of the degree of success achieved (good publication, award, etc.);
- deriving similarity between ACMC topics resulting from the survey and clustering them. (A similarity measure involving ACMC-tree inheritance is under development.);

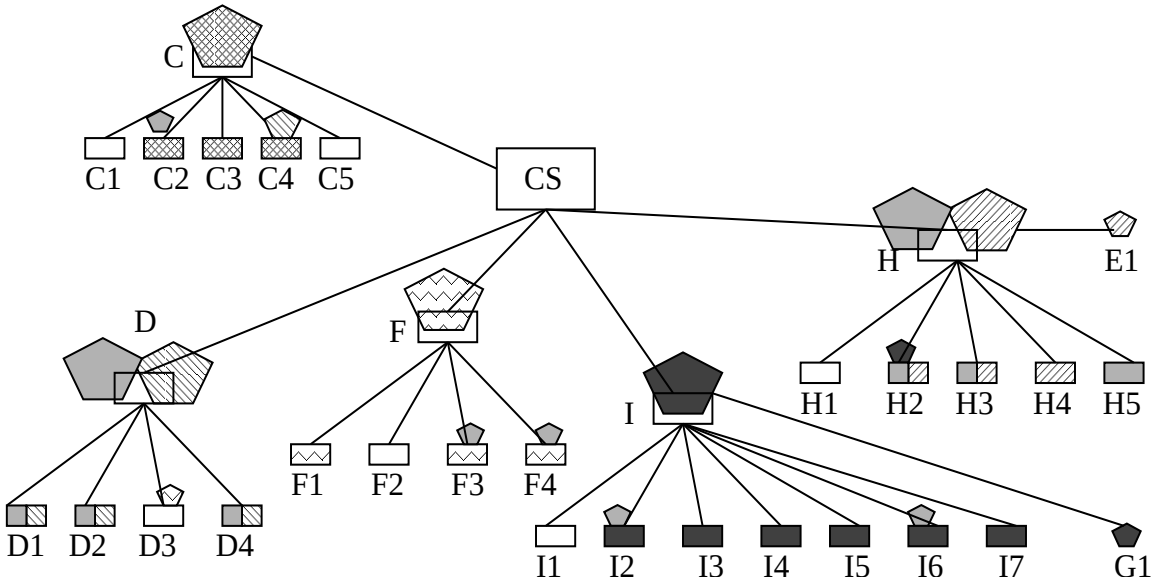

Figure 6: Six subject clusters in the DI FCT UNL represented over the ACMC ontology. Head subjects are shown with differently patterned pentagons. Topic boxes shared by different clusters are split-patterned.

- mapping clusters to the ACMC, which should be done in a parsimonious way by minimizing the weighted sum of counts of head subjects, gaps and offshoots;
- aggregating results from different clusters and, potentially, different organizations;
- interpretation of the results.

In principle, the approach can be extended to other areas of science or engineering, provided that these areas have been systematized into comprehensive ontologies or taxonomies. Potentially, this approach could lead to a useful instrument of visually feasible comprehensive representation of developments in any field of human activities prearranged as a hierarchy of relevant topics.

References

[1] *The ACM Computing Classification System* (1998), url= http://www.acm.org/class/1998/ccs98.html.

[2] C. Thorne, J. Zhu, V. Uren (2005), Extracting domain ontologies with CORDER, *Tech. Report kmi-05-14*, Open University, 1-15.

[3] S. Miralaei, A. Ghorbani (2005), Category-based similarity algorithm for semantic similarity in multi-agent information sharing systems, *IEEE/WIC/ACM Int. Conf. on Intelligent Agent Technology*, 242-245.

[4] L. Yang, M. Ball, V. Bhavsar, H. Boley (2005), Weighted partonomy-taxonomy trees with local similarity measures for semantic buyer-seller match-making, *Journal of Business and Technology*. Atlantic Academic Press, 1 (1) 42-52.

[5] M. Feather, T. Menzies, J. Connelly (2003), Matching software practitioner needs to researcher activities, *Proc. of the 10th Asia-Pacific Software Engineering Conference (APSEC'03)*, IEEE, p. 6.

[6] S.M. Weiss, N. Indurkhya, T. Zhang, F.J. Damerau (2005), *Text Mining: Predictive Methods for Analyzing Unstructured Information*, Springer Verlag, 237 p.

[7] S. Middleton, N. Shadbolt, D. Roure (2004), Ontological user representing in recommender systems, *ACM Trans. on Inform. Systems*, 22(1), 54-88.

[8] J. Liu, W. Wang, J. Yang (2004), Gene ontology friendly biclustering of expression profiles, *Proc. of the IEEE Computational Systems Bioinformatics Conference*, IEEE, 436-447.

[9] *RAE2008: Research Assessment Exercise* (2007), url= http://www.rae.ac.uk/.

[10] *The United Nations Millennium Project Task Force*, url= http:// www.cid.harward.edu/cidtech.

[11] B. Mirkin, T. Fenner, M. Galperin and E. Koonin (2003) Algorithms for computing parsimonious evolutionary scenarios for genome evolution, the last universal common ancestor and dominance of horizontal gene transfer in the evolution of prokaryotes, *BMC Evolutionary Biology* 2003, 3:2

[12] K.J. Holzinger and H.H. Harman (1941) *Factor Analysis*, University of Chicago Press, Chicago.

[13] K. Florek, J. Lukaszewicz, H. Perkal, H. Steinhaus and S. Zubrzycki (1951) Sur la liason et la division des points d'un ensemble fini, *Colloquium Mathematicum, 2*, 282-285.

[14] B. Mirkin (1987), Additive clustering and qualitative factor analysis methods for similarity matrices, *Journal of Classification*, 4, 7-31.

[15] J.A. Hartigan (1975) *Clustering Algorithms*, New York: J. Wiley & Sons.

[16] G. Gallo, M.D. Grigoriadis and R.E. Tarjan (1989) A fast parametric maximum flow algorithm and applications, *SIAM Journal on Computing, 18*, 30-55.

[17] N. Garg, V. V. Vazirani and M. Yannakakis (1996) Approximate Max-Flow Min-(Multi) Cut theorems and their applications, *SIAM Journal on Computing, 25*, n.2, 235-251.

[18] R.N. Shepard and P. Arabie (1979) Additive clustering: representation of similarities as combinations of overlapping properties, *Psychological Review*, 86, 87-123.

[19] B. Mirkin (2005), *Clustering for Data Mining: A Data Recovery Approach*, Chapman & Hall /CRC Press, 276 p.

[20] A. Ben-Dor, R. Shamir and Z. Yakhini (1999) Clustering gene expression patterns, *Journal of Computational Biology*, **6**, 281-297.

[21] J. Shi and J. Malik (2000) Normalized cuts and image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*, n. 8, 888-905.