Clustering via centroids a bag of qualitative values and measuring its inconsistency

Adolfo Guzman-Arenas and *Alma-Delia Cuevas*
a.guzman@acm.org, almadeliacuevas@gmail.com

Centro de Investigación en Computación (IPN) and *Escuela Superior de Cómputo (IPN), México*

ABSTRACT. It is well understood how to compute the average of a set of numeric values; thus, handling inconsistent measurements is possible. Recently, using *confusion*, we showed a way to compute the "average," consensus or centroid of a bag of assertions (made by *observers*) about a non-numeric property, such as John's pet. The values of those assertions lie in a hierarchy. Intuitively, such consensus minimizes the discomfort of all observers (of the pet) when they know which of the animals of the bag was selected as the consensus pet. The *inconsistency* of the bag is such total discomfort divided by the bag's size. It is a number that tells how far apart the values of the bag are. It should be emphasized that an asserted value obtained by an observer (such as *Schnauzer* in "the pet was a Schnauzer") represents not only itself, but all the values from it up to the root of the hierarchy: Schnauzer, dog, mammal, animal, living creature.

A bag of dissimilar assertions will have a large inconsistency, which could diminish if the problem at hand allows *several centroids* to be selected. John could have two pets, and the inconsistency of these two "consensus values" with all observations will be much better (much smaller): one part of the observers will feel little discomfort with one of the centroids; the remaining part will feel little discomfort with the second centroid. This chapter finds the set of centroids of a bag of qualitative values that minimizes the inconsistency of the bag; that is, the total discomfort of all members of the bag will be smallest. These centroids define clusters, that is, subsets of the bag.

All observers are equally credible, so differences in their findings arise from perception errors, and from the limited accuracy of their individual findings.

Keywords: qualitative values; hierarchies; inconsistency; centroid; clustering; confusion; knowledge representation; consensus

ACM Computing Classification: H.3.3 Information search and retrieval; **I.2.4** Knowledge representation formalisms and methods; **I.2.7** Natural language processing; **I.5.3** Clustering.

1. Previous work and problem statement

Our work is in the general area of extracting useful properties (such as "centers" and "clusters") from a set of non-numeric values.

1.1 Problem Statement

Assume several measurements are performed on the same property (for instance, the length of a table). One measurer took a quick look and asserted "3m." Another person with the help of a meter said "3.13m". A lady with a micrometer reported "3.1427m." The problem of finding the average of a set of quantitative values (to be called "Problem 0") can be solved simply by computing the average (μ=3.09m, the average length) as well as the dispersion of these measurements

(σ, the variance), perhaps disregarding some outliers. For quantitative measurements we know how to take into account contradicting facts, and we do not regard them necessarily as inconsistent. We just assume that the observers' gauges have different precisions or accuracies.

It could also be that observers have a propensity to lie, and in this case we apply the Theory of Evidence (Dempster-Schafer [3, 12]). Or we could use Fuzzy Logic, selecting some sets as possible answers and assigning a degree of membership to each measurement for each set.

Problem 1 statement (informal). Similar to Problem 0, we want to solve the problem of finding the "average," most plausible value, or centroid of several non-numeric or symbolic values.[1] This is "Problem 1," solved elsewhere [6] and briefly exposed in §1.4. There we find that the centroid is the value that minimizes the *total confusion* in the bag, a number that tells us how "comfortable" the elements of the bag with the chosen centroid are. Nevertheless, according to the problem at hand, it may be possible to have more than one "average." A bag, thus, may have several centroids, each of them representing or being "the center" of a cluster.[2] Several symbolic values in a bag could be better represented (in the sense of a smaller *total confusion* for the bag) by more than one centroid. Thus, we would like to cluster a bag of values into several centroids. This is "Problem 2" and its formal statement and solution (in Section 2) is the subject of this chapter.

Section 1.4 tells how to find one centroid of a bag of values. Sections 2.1 to 2.3 tell how to find the (several) centroids of a bag. –Section 2.4 tells how to find *one more centroid* of a bag. Section 2.6 tells the "best" number of centroids that represent a bag of values.

1.2 Related work

Many clustering methods that are used for objects represented by numbers (points in an *n*-dimensional space) can also be applied to symbolic values, if a numeric similarity function sim(*x*, *y*) is defined for each pair of those values. See, for instance, [13]. The great majority of them assume that *sim* is a distance function. A recent work [11] classifies these distances using information theory.

Yin *et al* [14] provide a manner to find the most likely "truth" among a set of qualitative information obtained from "information providers" in the Web. The information is an assertion about a qualitative value, a "fact" as found in the Web. This work resorts to the "trustworthiness" of each informant (resembling Dempster-Schafer), as well as a measure of the similarity among two of these non-numeric values (resembling our *confusion*, as defined in §1.3).

A recent paper [6] finds the centroid or most likely value of a bag of qualitative values, such as {Afghanistan; Beirut; Iraq; Kabul; Middle East; Afghanistan; Syria}. The answer is not necessarily the most popular value or mode (Afghanistan), or the lowest common ancestor (Middle East). The answer is not based on the probability that informants lie (like in the theory of Dempster-Schafer [3, 12]), nor it contains fuzzy values. The answer assumes that all informants are equally credible, and the discrepancy of their findings arises from the way or method used when

[1] Qualitative attributes (such as religion or color of hair) are also called symbolic or non-numeric properties, features, facets, attributes, or linguistic variables. The values these attribute attain (such as Muslim or brown) are called qualitative values, non numeric values, or linguistic constants.

[2] Each centroid has a cluster (a subset of the bag) associated with it: those elements that are most comfortable (lowest total confusion) with that centroid.

obtaining their observations. This work relies on the confusion *conf* (§1.3) of using a value *r* instead of the real or intended value *s*.

As an example, let us assume we want to find the ethnicity of Emille. We pose this question to her friends. One of them, from the sound of the name, assumes she is French. Other friend, knowing the Roman origin of the name, tells us that she is Italian. Another acquaintance tells us "she is white," still another says "she is European." Assume that the values are {French, Roman, Italian, White, European, American}. Given that information, one of these values is her most likely ethnicity. If "Italian" is selected, reporter 3 ("Italian") is happy (shows no discomfort, since our selection agrees with his report). Reporter 2 ("Roman") is somewhat displaced, since she reported a girl from Rome, not just from Italy. Reporter 1 ("French") is somewhat displeased, but reporter 6 ("American") is even more displeased with our selection. If we select "American" as Emille's ethnicity, only reporter 6 is at comfort, while the others show different degrees of dissatisfaction. If we could measure these discomforts, we could select as her most likely ethnicity (consensus value) *the value that minimizes the sum of disagreements* or discomforts for all the observers when they learn of the value chosen as the consensus value.

It should be emphasized that an asserted value obtained by an observer (such as *Doberman* in "John's pet was a Doberman") represents not only itself, but all the values from it up to the root of the hierarchy: Doberman, dog, mammal, vertebrate, and animal. This is because the observer, having all these values to select when reporting his observation, reports the most precise value.

The "discomfort" or disagreement when value *r* is reported instead of the "true" value *s* (as found by the observer) is called the *confusion* in using *r* instead of *s* [9]. To measure this, it is necessary to give all observers the same *context*, that is, the same set of possible qualitative answers as well as how these are related by specificity or generality. This set is called a *hierarchy* (Figure 1); it is a tree where each node is a qualitative value or, if it is a set, then its immediate descendants form a *partition* of it.

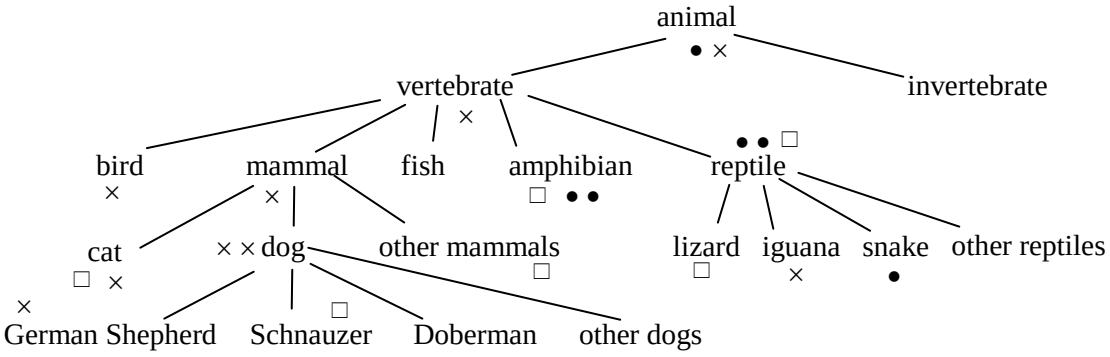

Figure 1. A hierarchy of symbolic values is a tree where every node is either a symbolic value or, if it is a set, then its immediate descendants form a partition. Hierarchies make possible to compute the confusion conf(*r*, *s*) that results when value *r* is used instead of *s*, the true or intended value. The confusion (§1.3) is the number of *descending* links in the path from *r* to *s*, divided by the height of the hierarchy. For instance, conf(dog, Doberman) = 1/4, conf(Doberman, dog) = 0, conf (Doberman, German Shepherd) = 1/4, conf (Doberman, iguana) = 2/4, conf(iguana, Doberman) = 3/4. Observe that conf ∈ [0, 1]. Refer to Section 1.3. The values marked ×, □ and • are used in examples 3, 5 and 6 of sections 1.3 and 1.4

Using hierarchies, next section (§1.3) tells how to compute the confusion among two qualitative values, while section 1.4 explains how to find the consensus or most likely value of a bag of qualitative values.

Sections 1.3 and 1.4 report some of our previous work, necessary to understand this article. Our contributions appear in section 2, where several centroids of a bag of qualitative values are found, both in an exact manner (§2.2) and in an approximate algorithm (§2.5). §2.6 tells us how many centroids (clusters) are good to have. Clusters are derived for each centroid found. Also, the *inconsistency* of each cluster (degree of fitting of the centroids to the cluster) is also computed. Results and comparisons against similar methods should wait for a comprehensive test of experiments. Conclusions and discussion are in section 3.

1.3 Comparing values. The confusion between two qualitative values

The *confusion* between qualitative values is dealt elsewhere [4, 5, 9, 10]; this section is placed here for completeness. How close are two numeric values v_1 and v_2? The answer is $|v_2 - v_1|$. How close are two symbolic values such as *cat* and *dog*? The answer comes in a variety of similarity measures and distances. Hierarchies (introduced in Figure 1) allow us to define the confusion conf(*r, s*) between two symbolic values. The function conf will open the way to evaluate in Section 1.4 the inconsistency of a bag of symbolic observations. We assume that the observers of a given fact (such as *the killer is…*) share a set of common vocabulary, best arranged in a hierarchy. A hierarchy can be regarded as the "common terminology"[3] for the observers of a bag: their *context.* Observers reporting in other bag may share a different context, that is, another hierarchy.

What is the capital of Germany? *Berlin* is the right answer; *Frankfurt* is a close miss, *Madrid* a fair error, and *sausage* a gross error. What is closer to a *cat*, a *dog* or an *orange*? Can we measure these errors and similarities? Can we retrieve objects in a database that are close to a desired item? Yes, because qualitative variables take symbolic values such as *cat, orange, California,* which can be organized in a hierarchy *H*, a mathematical construct among these values. Over *H*, we can define the function *confusion* resulting when using a symbolic value instead of another.

Definition. For $r, s \in H$, the **absolute confusion** in using *r* instead of *s*, is

$$\text{CONF}(r, r) = \text{CONF}(r, \text{any ascendant of } r) = 0;$$
$$\text{CONF}(r, s) = 1 + \text{CONF}(r, \text{father\_of}(s)).$$

To measure CONF, count the descending links from *r* (the replacing value) to *s* (the intended or real value) in *H*. CONF is not a distance, nor an ultradistance.

We can normalize CONF by dividing it into *h*, the height of *H* (the number of links from the root of *H* to the farthest element of *H*), yielding the following

Definition. The **confusion** in using *r* instead of *s* is

$$\text{conf}(r, s) = \text{CONF}(r, s)/h.$$

Notice that $0 \le \text{conf}(r, s) \le 1$. It is not symmetric: $\text{conf}(r, s) \ne \text{conf}(s, r)$, in general. The function conf is not a distance, but it obeys the triangle inequality [6]. Also, $\text{conf}(r, s) = 0$ does not mean necessarily that $r=s$.

The relation of a hierarchy with an ontology is:

[3] If the symbolic values become full *concepts,* it is best to use an *ontology* instead of a *hierarchy* to place them. [2].

(a) A hierarchy can be regarded as a simplified ontology, where only the relations "subset" and "member" are used.
(b) In an ontology, a concept may have several ancestors; in a hierarchy, only one.

Example 1. For the hierarchy of Figure 1, CONF(cat, mammal) =0; if I ask for a mammal and I am given a cat instead, I am happy, and CONF=0. But CONF(mammal, cat) =1; if I ask for a cat and I get a mammal, I am somewhat unhappy, and CONF=1. For the same reason, CONF(vertebrate, cat) =2. Being given a vertebrate when I ask for a cat makes me unhappier than when I was handed a mammal.

Example 2. In the hierarchy of Figure 1, conf(cat, dog) =1/4; conf(cat, Schnauzer) =1/2.

Example 3. The confusion among values marked with □ in Figure 1 is given in Table 1.

Table 1. The confusion conf (*r*, *s*) of using value *r* instead of the intended value *s* is found at the intersection of row *r* and column *s*. For instance, conf(lizard, reptile) = 0, while conf(reptile, lizard) = ¼, and conf(reptile, Schnauzer) = ¾ [If I want a Schnauzer and they give me a reptile, my confusion is large = ¾, close to 1, the highest]. The last column (explained in §2.2) gives the total confusion provoked in the bag {Schnauzer, cat, other mammals, amphibian, reptile, lizard} by the corresponding row. For instance**,** the total confusion that "cat" provokes in the bag is ½ + 0 + ¼ + ¼ + ¼ + 1/2 =1.75.

	Schnauzer	**cat**	**Other mammals**	**Amphibian**	**reptile**	**lizard**	**Total confusion**
Schnauzer	0	¼	¼	¼	¼	½	1.5
cat	½	0	¼	¼	¼	½	1.75
other mammals	½	¼	0	¼	¼	½	1.75
amphibian	¾	½	½	0	¼	½	2.5
reptile	¾	½	½	¼	0	¼	2.25
lizard	¾	½	½	¼	0	0	2

Remark. Since symbolic values lie in a hierarchy, it is not possible for a value to have two immediate ascendants, to have more than one path from it towards the root. That is, *rabbit* may not be both a mammal and a bird.

The type of hierarchy of Figure 1 is the most common type, and it is called a *normal* hierarchy, as opposed to ordered [(B) in Figure 3] or percentage hierarchies [9].

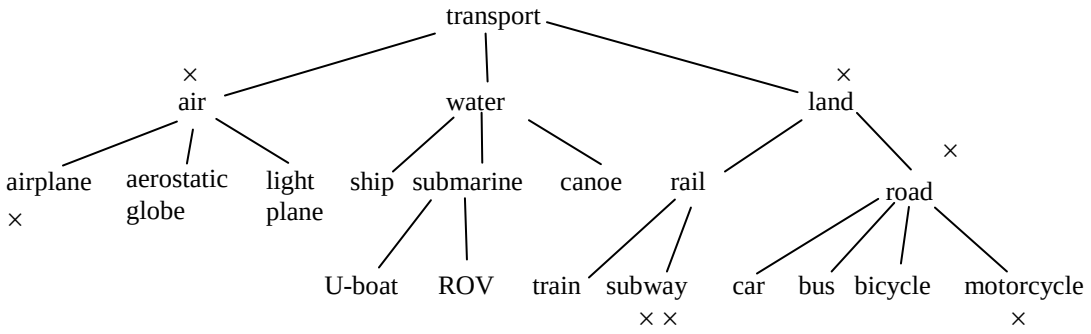

Figure 2. A hierarchy of types of transportation. The observations of bag_3 (see Example 4) are shown with ×. conf(airplane, transport) =0; conf(transport, airplane) =2/3; conf(U-boat, ROV) =conf(U-boat, canoe) =1/3; conf(U-boat, motorcycle)=1

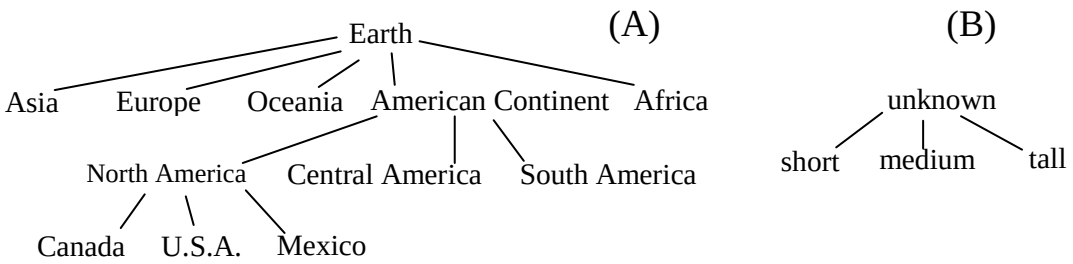

Figure 3. Hierarchies for places to live (A), a normal hierarchy, and height of persons (B), an ordered hierarchy. conf(Mexico, North America) =0; conf(North America, Mexico) =1/3; conf(Mexico, South America) =2/3; conf(South America, Mexico)=1. Ordered hierarchies have height 1 always, and all the values hanging from the root are totally ordered. In [5] we find that, for an ordered hierarchy with a root and *n* children, the confusion between children *i* and *j* is $|i - j| / (n - 1)$. Thus, conf(short, medium) =conf(medium, tall)=1/2; conf(short, tall) =1; conf(short, unknown) = 0; conf(unknown, short) = conf(unknown, medium) =1

1.4 The consensus and inconsistency of a bag of qualitative values

The consensus or *centroid* of a bag is dealt in [6, 8]; this section is placed here for completeness.

The setting is that several observers report (qualitative) values about a given property of an object they were asked to observe. Two observers may report the same value, thus a bag (and not a set) is used to collect them. These values are usually different, but our observers are not liars (the Dempster-Schafer's theory of evidence does not apply). Their reported values are crisp (no fuzzy values are reported –no fuzzy logic is needed). The explanation for not all reporting the same value is that the way they observed (assessed, "measured" or gauged) the property was different –their methods of observation had different precision; accuracy varies. Granularities were not the same.

Problem 1. Given a bag of n observations reporting non-numeric values, how can we measure its inconsistency? What is the value in the bag that minimizes this inconsistency? We shall call r^* this value and σ the inconsistency that r^* produces. Notice that *inconsistency* is a property of a bag of values, not of a single value.

Restrictions to the solution to Problem 1:

(A) All the reported values are about the same *fact* or property. One observer can not report about the identity of the killer, while another observer tells us about the weather in London.

(B) The fact or feature that the observers are gauging, has a single value. There is only one killer.[4] The weather in London (for a particular date and corner of the city) is unique.

(C) All reporters use the same *context* expressed in a vocabulary arranged in a hierarchy –the same hierarchy for all observations in a bag. It contains all possible answers. It is clear that for observers with other conceptions about the animals of Figure 1 and their differences, the consensus r^* will differ. Thus, r^* and σ are a function of the bag and the hierarchy.

Intuitively, r^* is the value *in the bag*[5] most likely to be true, given the available information, and taking into account observation errors. One of the values of the bag must be the most plausi-

[4] Problem 1 assumes just one centroid is possible. This chapter solves Problem 2 in Section 2, where the situation is such that several centroids are possible.

ble value, the consensus, its "centroid". Since all observers are equally believable, we could find the confusion of any given observer with respect to a selected value *r* --a kind of "discomfort" measured by conf(*r*, *s*) when value *r* is preferred or selected (as the centroid), instead of the value *s* reported by him. Adding these confusions for all observers, we find the total confusion (total "discomfort") that such value *r* produced (if it were selected as the "consensus") in all *k* observers. There must be a value *r** that produces the lowest total confusion. Such *r** is the consensus or centroid of the bag. The inconsistency of the bag, called σ, is such minimum divided by the number of elements of the bag. Thus, we have

Solution. The *centroid* or *consensus* *r** of a bag *B* of *n* observations reporting qualitative values $\{s_1, s_2, \ldots s_n\}$ is the value $r_j \in B$ that minimizes

$$\sum_{i=1}^{n} \text{conf}(r_j, s_i) \qquad \text{for } j = 1,..., n$$

For each r_j, this sum is the total confusion that r_j produces among all elements of the bag.
The *inconsistency* σ of *B* is the minimum that such *r** produces, divided by *n*:

$$\sigma = (1/n) \min_j \sum_{i=1}^{n} \text{conf}(r_j, s_i) = (1/n) \sum_{i=1}^{n} \text{conf}(r^*, s_i)$$

Example 4. Assume bag_3 is {air, airplane, land, road, subway, subway, motorcycle}. Bag_3 is marked with × in Figure 2. Since the bag contains the set of observations made by different observers, some of their findings may coincide. In this example, two observers found "subway". For bag_3, the total confusion for air is (0+1+1+2+3+3+3)/3 =4.333; for airplane, is (0+0+1+2+3+3+3)/3 =4; for land, it is 3.333; for road is 2.666; for subway is 2; for motorcycle is 2.333. Thus, its consensus is subway, and its inconsistency is 2/7. *Example 5.* For bag_1 = {animal, vertebrate, bird, mammal, cat, dog, dog, iguana, German Shepherd} (marked with × in Figure 1), the consensus or centroid *r** is German Shepherd, and the inconsistency of bag_1 is [(0+0+1+0+1+0+0+2+0)/4]/9 =1/9. *Example 6.* For bag_2 = {animal, amphibian, amphibian, reptile, reptile, snake} marked with • in Figure 1, *r** = snake, σ = 1/12. *Example 7.* For observations in Figure 1 with □, *r** = Schnauzer, σ = (6/4)/6 = ¼.

Remarks. (More at [6])

I. *r** and σ are properties of the bag, and depend on the context of use –represented by the hierarchy employed. The role of the hierarchy in the solution to Problem 1 is to provide a *common vocabulary* for all observations. See restriction 1.4.(C).
II. The inconsistency $\sigma \in [0, 1)$. In fact, for a bag *B* of size *n*, $0 \le \sigma \le (n\text{-}1)/n$.
III. There may be more than one value *r** that minimizes the total confusion.
IV. To compute the inconsistency of a bag, we resort to finding *r** first. In other words, the inconsistency of a bag is the average total discomfort (average total confusion) produced

[5] No values other than those in the bag are available for consensus; it will be a surprise to *all* observers to find out that the killer was Andrew, if *noboby* reported Andrew in his/her findings.

by *r**. This is the lowest discomfort attainable; any other element different from *r** will give a larger or equal total confusion (by definition of *r**).

V. The consensus *r** is not inevitably the most popular value (the mode), which is dog in Example 5 for the elements marked with (×), while *r** =German Shepherd.

VI. The *lowest common ancestor* (vertebrate in example 7) produces a total confusion larger or at best equal than the total confusion produced by *r** = Schnauzer. It is "too general" for many of the observers.

VII. Given the consensus *r** of B, there is no $r' \in B$ such that *r*' is a descendant of *r**. This implies that, if an element *r* of the bag has a descendant *r*' in the bag, then that *r* can not be the consensus.

Notice that we have found a way of adding (and averaging) apples and oranges, and a quantity (σ) to quantify out how disperse or divergent a bag of symbolic values is.

1.5 Is the centroid the Condorcet winner?

The Condorcet winner of an election is the candidate who, when compared with every other candidate, is preferred by more voters. Informally, the Condorcet winner is the person who would win a two-candidate election against each of the other candidates. A Condorcet winner will not always exist in a given set of votes.

If we consider the selection of the centroid of a bag as an election held by the members of such bag, then its centroid is the Condorcet winner. This is so because *r** has the lowest total confusion (the criteria to select the winner), thus beating (or tying) any other candidate.

A voting system satisfies the Condorcet criterion if it chooses the Condorcet winner when one exists. Our method for "electing" the centroid (using total confusion) satisfies the Condorcet criterion. In fact, the total confusion induces a total ordering among the qualitative values of the bag. They can be ordered $s_1 \le s_2 \le ... \le s_k$, where s_1 is the centroid (or Condorcet winner), and s_k is the Condorcet loser -- a candidate who can be defeated (or tied) in a head-to-head competition against every other candidate.

2. When a bag can have several centroids

According to the problem at hand, it may be meaningful or desirable to find more than one "consensus" for a bag of qualitative values. For a bag reporting intelligence findings about the place where Osama Bin Laden may be hiding, it makes sense to find the two or three most likely places, and go and search for him there. Thus, an algorithm that computes more than one centroid is desirable, especially if the inconsistency provoked by a single centroid (§1.4) is high.

In the solution to Problem 1 (§1.4), the centroid *r** produces the lowest total confusion. The inconsistency is that lowest total confusion divided by *n*, the size of the bag.

Intuitively, to find more than one centroid, we could ask every element *e* of the bag to select *the best* candidate for such *e*, and we will select two or three of these "best" candidates to be the centroids. Unfortunately, this approach will not work –every *e* will select *itself,* since conf(*e*, *e*) =0, and we will end up with a large number of "best" candidates.

If we want to select two centroids, we could present sequentially every pair of candidates (c_i, c_j) to each element *e* of the bag. For each pair, *e* will select c_i as its preferred centroid if conf(c_i, *e*) < conf(c_j, *e*), select c_j if conf(c_i, *e*) > conf(c_j, *e*), or select one of c_i, c_j arbitrarily if conf(c_i, *e*)

= conf(c_j, e). The smaller of these confusions will accumulate so as to obtain the total confusion provoked by the pair of candidates in all elements of the bag. Having done that with all possible pairs of candidates, the pair with the smallest total confusion will be the winner –they are the two centroids. Each of these two centroids induces a cluster (a subset of the bag), formed by those elements that selected that centroid as their best candidate (its "core constituency"); there may be a few elements of the bags whose best candidate was a tie between the two winning centroids; those elements could be considered outside the two clusters, or could be included in either.

2.1 Finding the *k* centroids of a bag

A bag may have not just one or two centroids, but *k* of them. Thus, we state formally

Problem 2. Given a bag of n values, what are its k centroids? What is the inconsistency of such bag?

Solution.

To find the two (the pair of) *centroids* (Algorithm 1 of §2.3 formalizes this)*:*

The total confusion (tc) provoked by two candidates c_i, c_j in a bag $\{c_1, c_2, \ldots, c_n\}$ is

$tc(c_i,c_j) = \min(conf(c_i,c_1),conf(c_j,c_1)) + \min(conf(c_i,c_2),conf(c_j,c_2)) + \ldots + \min(conf(c_i,c_n), conf(c_j,c_n))$

and the pair of centroids r_1*, r_2* is the pair c_i, c_j that minimizes the above tc. The inconsistency of the bag is such smallest tc divided by *n*. The bag is broken in two clusters: one of them holds those elements that preferred r_1* to r_2*; the other cluster holds those elements that preferred r_2* to r_1*. The elements *e* having conf(r_1*, *e*) = conf(r_2*, *e*) could be left as a residue, or could assigned to either cluster.

Remark. There may be more than one pair of centroids; for instance, two pairs (r_1*, r_2*) and (r_3*, r_4*) may provoke the same lowest total confusion in the bag.

To find the three (the trio of) *centroids:*

The total confusion provoked by three candidates c_i, c_j, c_m in a bag $\{c_1, c_2, \ldots, c_n\}$ is

$tc(c_i,c_j,c_m) = \min(conf(c_i,c_1), conf(c_j,c_1), conf(c_m,c_1)) + \min(conf(c_i,c_2), conf(c_j,c_2), conf(c_m,c_2)) + \ldots + \min(conf(c_i,c_n), conf(c_j,c_n), conf(c_m,c_n))$

and the trio of centroids r_1*, r_2*, r_3* is the trio c_i, c_j, c_m that minimizes the above tc. The inconsistency of the bag is such smallest tc divided by *n*. The three clusters are formed by the elements that preferred r_1*, r_2* or r_3*, respectively. An "undecided" element c_i showed the same lowest confusion for more than one centroid: either $conf(r_1*,c_i)=conf(r_2*,c_i)$, or $conf(r_1*,c_i)=conf(r_3*,c_i)$ or $conf(r_2*,c_i)=conf(r_3*, c_i)$ or $conf(r_1*,c_i)=conf(r_2*,c_i)=conf(r_3*,c_i)$. That c_i can be assigned to either of its preferred clusters, or it can be left in a residue bag.

To find the k centroids:

The total confusion provoked by *k* candidates c_i, c_j,… c_k in a bag $\{c_1, c_2, \ldots, c_n\}$ of *n* elements is

$tc(c_i,c_j, \ldots c_k) = \min(conf(c_i,c_1), conf(c_j,c_1), \ldots conf(c_k,c_1)) + \min(conf(c_i,c_2), conf(c_j,c_2), \ldots conf(c_k,c_2)) + \ldots + \min(conf(c_i,c_n), conf(c_j,c_n), \ldots conf(c_k,c_n))$

and the solution r_1*, r_2*,… r_k* is the *k*-tuple c_i, c_j,…c_k that minimizes the above tc. The inconsistency of the bag is such smallest tc divided by *n*. The *k* clusters are formed by the elements that preferred r_1*, r_2*,… or r_k*, respectively. "Undecided" elements are handled as before. There may be more than one *k*-tuples of centroids that minimize tc.

As the number of allowed centroids increases, the inconsistency of the bag decreases, since each element of the bag can find better candidates (lower confusion) when there are more of them.

Remarks.

1. The above solution is slow –it is an exhaustive search. The number of distinct sets of *k* candidates is $n!/[(n\text{-}k)!k!]$ and the amount of *conf*'s needed to compute tc for each of these sets of candidates is *nk*, so that the total number of *conf*'s needed is $nkn!/[(n\text{-}k)!k!]$; thus, the complexity of the solution is $\mathcal{O}(n)\times\mathcal{O}(n^n)/\mathcal{O}(1) = \mathcal{O}(n^n)$, since *k* is constant.
2. Up to this point, we still do not know how to set *k*, the appropriate number of centroids to have. A low number of centroids is desirable, but it is also desirable to have a small inconsistency.
3. We see that the inconsistency of a bag is a monotonic function, starting at some value $\sigma \geq 0$ for 1 centroid, and reaching 0 for *n* centroids. In fact, if there are repeated elements in the bag, it reaches 0 for less than *n* centroids. See Figure 4.
4. We could try to minimize a linear combination of the number *k* of centroids and the inconsistency σ caused by them, such as $\alpha k + (1\text{-}\alpha)\sigma$ for $0 \leq \alpha \leq 1$. Small values of α will favor many centroids, while a large α will favor few. Empiricism will dictate which α is "best" or "appropriate." We follow instead another (empirical) solution: to search for a large drop in the inconsistency (§2.6) as we add more centroids; that drop sets *k*.

2.2 Finding the *k* centroids of a bag of qualitative values –exact solution

We explain the algorithm with an example, and then formalize it in *Algorithm 1* of §2.3 for *k*=2. Let us compute the centroids of bag {Schnauzer, cat, other mammals, amphibian, reptile, lizard} which is marked with □ in Figure 1.

One centroid. If we were going to select one centroid for the bag {Schnauzer, cat, other mammals, amphibian, reptile, lizard}, it will be Schnauzer, with an inconsistency σ =1.5/6 =0.25 (See Table 1). That is, to find one centroid of a bag we apply the formula of §1.4.

Two centroids. For selecting two centroids, we have the following choices: (Schnauzer, cat); (Schnauzer, other mammals), (Schnauzer, amphibian), (Schnauzer, lizard)[6], (cat, other mammals), (cat, amphibian), (cat, lizard), (other mammals, amphibian), (other mammals, lizard), and (amphibian, lizard). We compute the total confusions caused by each of these pairs. As an example, let us compute the confusion caused by the pair (cat, lizard). Considering the rows of Table 1 corresponding to cat and lizard, we construct Table 2. Each element (each "voter") of the bag selects one of these two candidates –the best for it, the candidate provoking in the candidate the smaller confusion. For instance, voter Schnauzer selects candidate cat, voter cat selects candidate cat, voter "other mammals" selects candidate cat, voter am-

[6] Due to remark VII of §1.4, the value reptile can never be a centroid.

phibian selects either, voter reptile selects lizard… The results are shown in the last row of Table 2. Thus, the pair (cat, lizard) provokes a total confusion of 1.

Computing the total confusion for all pairs, the values are: (Schnauzer, cat) =1.25; (Schnauzer, other mammals) =1.25, (Schnauzer, amphibian) =1.25, (Schnauzer, lizard) =0.75, (cat, other mammals) =1.5, (cat, amphibian) =1.5, (cat, lizard) =1, (other mammals, amphibian) =1.5, (other mammals, lizard) =1, and (amphibian, lizard) =1.75. Thus, the best pair is (Schnauzer, lizard) with an inconsistency of 0.75/6 =0.125. §2.3 gives a detailed description of this algorithm.

Table 2. To compute the total confusion caused by the pair (cat, lizard) in each element (each "voter") of the bag, we select for each column the smallest confusion (results shown in last row) and add these values. For instance, for column "Schnauzer" we select the smallest of ½ and ¾, and place that value (1/2) in the third row of that column. The result of adding the values of the third row is ½ + 0 + ¼ + ¼ + 0 + 0 =1 (which is placed in the last row, last column). This table is part of Table 1, hence, a practical algorithm will use Table 1 and ignore or select appropriate rows of it.

	Schnau	cat	Other mamma	amphib	reptile	lizard	Total confusion
cat	½	0	¼	¼	¼	½	
lizard	¾	½	½	¼	0	0	
Smaller value	½	0	¼	¼	0	0	1

Three centroids. Computing the total confusion for all trios, the values are: (Schnauzer, cat, other mammals) =1; (Schnauzer, cat, amphibian) =1; (Schnauzer, cat, lizard) =0.5; (Schnauzer, other mammals, amphibian) =1; (Schnauzer, other mammals, lizard) =0.5; (Schnauzer, amphibian, lizard) =0.5; (cat, other mammals, amphibian) =1.25; (cat, other mammals, lizard) =0.75; (cat, amphibian, lizard) =1; (other mammals, amphibian, lizard) =0.75. Thus, we have a triple tie, the winners of the best three centroids are (Schnauzer, cat, lizard), (Schnauzer, other mammals, lizard), and (Schnauzer, amphibian, lizard), each with an inconsistency of 0.5/6 = 0.083.

Four centroids. Computing the total confusion for all quartets, these are: (Schnauzer, cat, other mammals, amphibian) =0.75; (Schnauzer, cat, other mammals, lizard) =0.25; (Schnauzer, cat, amphibian, lizard) =0.25; (Schnauzer, other mammals, amphibian, lizard) =0.25; and (cat, other mammals, amphibian, lizard) =0.5. There are three winners, the quartets (Schnauzer, cat, other mammals, lizard), (Schnauzer, cat, amphibian, lizard), and (Schnauzer, other mammals, amphibian, lizard), with an inconsistency of 0.25/6 =0.041.

Five centroids. Finally, we see that the quintet (Schnauzer, cat, other mammals, amphibian, lizard) has total confusion =0. That is the solution if we insist in five centroids. See Figure 4.

Notice that, once the pair wise confusions among the values (Table 1) are found, there are no more tedious calculations to make. For finding the best pair of centroids, it is enough to select every two rows of Table 1, finding the max of each column, and add these values. Table 2 seeks to explain this procedure. Notice that Table 2 is a subset of Table 1. Similarly, for finding the best trio of centroids, it is enough to select every three rows of Table 1, find the max of each column, and add these values. And so on.

The above algorithm is exact (it will select the best pair, the best trio…) but it will be too slow for large bags. We present in §2.5 an approximate algorithm that is much faster.

2.2.1 Repeated centroids are unnecessary

If a bag has repeated elements, could it be that in its *k* centroids exist repeated elements, too? For instance, let us consider bag {Schnauzer, Schnauzer, cat, other mammals, amphibian, reptile, lizard}. Its two centroids are (Schnauzer, lizard). Could it be that its three centroids are (Schnauzer, Schnauzer, lizard)? That can not be, for the following reason. Replacing one of the Schnauzers by any other value in the bag not already in the set of centroids (say, by cat) will lower (or at least will not increase) the total confusion provoked by the new trio (Schnauzer, cat, lizard), since elements that selected the eliminated Schnauzer can still vote *for the other Schnauzer* and not increase the total confusion. And the new inserted candidate, cat, will now vote for itself with confusion 0, which may lower the total confusion, since 0 is not larger that conf(cat, Schnauzer) or conf(cat, lizard). Thus, replacing a duplicated candidate by some other candidate will not increase the total confusion, and may lower it. Likewise, expunging the duplicate candidate from the set of candidates will lower the number of candidates, but will not lower the total confusion provoked by them.

2.3 Finding the two centroids of a bag –exact solution

The algorithm outlined in §2.2 for finding the two centroids of a bag –that is, for clustering the bag in two clusters, is described here in detail.

We find the total confusion produced in the bag by each pair (c_i, c_j) of candidates, and select the pair with the lowest total confusion. If there is a tie (more than one pair achieves the lowest total confusion) we use some breaking rules, later explained. Let us call "voters" the elements of the bag (a single voter will be denoted by *v*), since they will select or "vote" for one of the two candidates c_i or c_j. The algorithm is the following.

Algorithm 1. Finding the two centroids of bag B of n>1 elements (qualitative values).
Input: *B*, a bag with at least two values.
Output: Two elements of *B*, its two centroids.

1. For every pair *p* of candidates (c_i, c_j) such that i=1,… *n*, *j*>*i*, $c_i \neq c_j$, compute:
   toti(*p*) = total confusion caused by voters *v* for which conf(c_i, *v*) < conf(c_j, *v*)
   = $\sum$ conf(c_i, *v*) where the sum is over all $v \in B$ such that conf(c_i, *v*) < conf(c_j, *v*));
   totj(*p*) = total confusion caused by voters *v* for which conf(c_i, *v*) > conf(c_j, *v*));
   totd(*p*) = total confusion caused by voters *v* for which conf(c_i *v*) = conf(c_j, *v*));
   tc(*p*) = toti(*p*) +totj(*p*) +totd(*p*) /* tc(*p*) is the total confusion caused in *B* by pair *p*, as defined in §2.1 */
2.A. Find those *p*'s which minimize tc(*p*). Usually there is one; there may be more than one.
   If *p* is unique, return that pair as the answer. Otherwise,
   2.B. Eliminating one *p*. If there is more than one *p*, then take a couple of these pairs, in order to eliminate one of them. Say the chosen pairs are $p_1 = (c_1, c_2)$ and $p_2 = (c_3, c_4)$.
      i. It could be that one of the candidates in one pair appears in both pairs. Without loss of generality, we can assume that $c_1 = c_3$. Then, that candidate will surely be one of the two centroids of the surviving pair. The other surviving centroid, c_2 or c_4, will be that with the lowest total confusion toti or totj. (Notice that tc is not used here.) If there is a tie, select either as the second survivor.

ii. If none of the candidates of one pair appears in the other pair, select as survivor the pair containing the candidate with the lowest total confusion. In a tie, select either pair.

2.C. Eliminate the other pair. Go to 2.B unless one pair remains.

2.D. Return the surviving pair as the answer. *End of Algorithm 1.*

Notice that step 2.A finds the pair(s) *p* that minimizes the total confusion, tc(*p*). If several *p* are attained, only then toti(*p*) and totj(*p*) are used (steps 2.B and C) to discriminate among them.

Example 8. For bag {Schnauzer, cat, other mammals, amphibian, reptile, lizard}, Step 1 of Algorithm 1 produces:

For candidates (Schnnauzer, cat): toti(Schnauzer) =0;[7] totj(cat) =0;[8] totd =1.25;[9] tc(Schnauzer, cat) =1.25;

For candidates (Schnnauzer, other mammals): toti(Schnauzer) =0;[10] totj(other mammals) =0;[10] totd =1.25; [10] tc(Schnauzer, other mammals) =1.25;

For candidates (Schnnauzer, amphibian): toti(Schnauzer) =0.5; totj(amphibian) =0; totd =0.75; tc(Schnauzer, amphibian) =1.25;

For candidates (Schnnauzer, lizard): toti(Schnauzer) =0.5; totj(lizard) =0; totd =0.25; tc(Schnauzer, lizard) =0.75;

For candidates (cat, other mammals): toti(cat) =0; totj(other mammals) =0; totd =1.5; tc(cat, other mammals) =1.5;

For candidates (cat. amphibian): toti(cat) =0.75; totj(amphibian) =0; totd =0.75; tc(cat, amphibian) =1.5;

For candidates (cat, lizard): toti(cat) =0.75; totj(lizard) =0; totd =0.25; tc(cat, lizard) =1 (see also Table 2);

For candidates (other mammals, amphibian): toti(other mammals) =0.75; totj(amphibian) =0; totd =0.75; tc(other mammals, amphibian) =1.5;

For candidates (other mammals, lizard): toti(other mammals) =0.75; totj(lizard) =0; totd =0.25; tc(other mammals, lizard) =1, and

For candidates (amphibian, lizard): toti(amphibian) =0; totj(lizard) =0; totd =1.75; tc(amphibian, lizard) =1.75.

In step 2.A, we find that the lowest total confusion (tc) is 0.75, and it is reached only by (Schnauzer, lizard). Therefore, the accumulators toti( ) and totj( ) were not used to disambiguate any tie among best pairs.

Thus, the two centroids are (Schnauzer, lizard) with an inconsistency of 0.75/6 =0.125. The clusters (according to Table 1) are {Schnauzer, cat, other mammals}, {lizard, amphibian, reptile}. The only indifferent value was amphibian, who selected equally Schnauzer or lizard, since conf(Schnauzer, amphibian) =conf(lizard, amphibian) = ¼.

2.3.1 Is the pair of centroids a Condorcet winner?

The Condorcet winner, when it exists, is a single individual –not two of them. But we could imagine a country which always elects two co-presidents, so that voters manifest their preference about couples of candidates –each candidate appears in every possible pair. In that country, competition at elections is among pairs of candidates, not among candidates. In that case, our method of "electing" the best pair is a Condorcet method (Cf. §1.5), and the victorious pair (r_1*, r_2*) is the Condorcet winner, since it beats (or draws) all other pairs. The cluster "around" r_1* [formed by those elements *e* where conf(r_1*, *e*) < conf(r_2*, *e*)] is the constituency or supporters of r_1*; the cluster "around" r_2* is the constituency of r_2*; the elements *e* for which conf(r_1*, *e*) = conf(r_2*, *e*) are the undecided voters (swing voters, floating voters –their vote can go to r_1* or to r_2), they

[7] Only value Schnauzer voted for candidate Schnauzer.

[8] Only value cat voted for candidate cat.

[9] Values "other mammals," amphibian, reptile and lizard voted indifferently (with the same confusion) for Schnauzer or cat.

[10] Only value Schnauzer voted for candidate Schnauzer; only value "other mammals" voted for candidate "other mammals;" values cat, amphibian, reptile and lizard voted indifferently for Schnauzer and for "other mammals."

can cluster around r_1*, around r_2*, or be split into the two clusters. These swing voters swung only between r_1* and r_2*; any other candidate was less preferred.

2.4 Find one more centroid of a bag that already has *k*

If by some means we already found *k* centroids or consensus for a bag, and still we want to find *k*+1 of them, this section provides an algorithm that does this in an approximate manner. This algorithm avoids the exhaustive search of §2.2, but the *k*+1 "centroids" it delivers may not be the true centroids.

The idea is to replace the worst of the *k* centroids, by two new found. This replacement is straightforward, if we accumulate for each centroid the total confusion caused by it (Step D below). The algorithm follows.

Algorithm 2. Find one more centroid of bag B with k centroids (c_1, c_2, ..., c_k)
Input: *B*, a bag of size at least *k*+1; *k*, the number of centroids**;** and the *k* centroids (c_1, c_2, ..., c_k)
Output: The *k*+1 centroids c_1', c_2'…, c_k', c_{k+1}'.

A. If *k*=0, apply formula of §1.4 to find one centroid, and return that as output.

B. If *k*=1, apply Algorithm 1 of §2.3 to find two centroids; return that as output.

C. Otherwise, find which of the centroids c_1, c_2…, c_k is the worst – has the largest total confusion. To achieve that, let each value *v* of the bag select the centroid which provokes in that *v* the lowest confusion; that is, the centroid c_j that minimizes conf(c_j, *v*) is selected by *v*.

a. Thus, for each *v* in the bag,

1. If precisely one centroid c_j is selected by *v* (as that minimizing conf(c_j, *v*)), then accumulate the confusion conf(c_j, *v*) in a counter totconf(*j*) for that *j*th centroid.

   Also, keep the list (*v*, c_j) meaning "value *v* voted for c_j."

2. If several centroids c_i, c_j,… c_m are selected (that is, conf(c_i,*v*)= conf(c_j,*v*)= … = conf(c_m,*v*) --several centroids provoked the same minimum in the confusion of using that centroid instead of *v*), accumulate the confusion conf(c_i,*v*) in a counter totconfindif –here we accumulate the confusion provoked in *v* if *v* is an *indifferent* voter.

   Also, keep the candidates selected by *v* in the list (*v*, c_i, c_j, c_m) meaning "value *v* voted indifferently for c_i or c_j or … or c_m".

b. After all elements of the bag have made their selection in step a, find which of these centroids has the largest (worst) total confusion totconf(). In this, totconfindif is not used. Call *w* this worst centroid. If there is a tie, select any.

c. Discard that worst centroid *w* found in step b, but first find which voters voted for it, including the voters which voted for *w* indifferently (that is, voters who voted for *w* and for somebody else). Let *B*' = those voters. Remark: *B*' ⊂ *B*.

D. At this point, *k*-1 centroids remain (one was discarded). Using Algorithm 1, find the best two centroids for bag *B*'. Return those two best centroids together with the remaining *k*-1 centroids as the desired *k*+1 centroids of bag *B*. *End of Algorithm 2*.

Remarks.

I. In step C, we compute again the total confusion totconf(j) that centroid c_j provokes. We do not use totj(p) of Algorithm 1, because totj(p) was computed when only two centroids were contending (for the vote of v), whereas in Algorithm 2 we have k centroids contending.

II. In step C, all k centroids are eligible for removal.

III. In step D, the two centroids for bag B' should belong, of course, to B'.[11]

IV. The centroid discarded in step C.c could be selected again when step D returns two additional centroids; that is, the discarded centroid could reappear and be one of these two.

V. Algorithm 2 converges quickly because at each step, the best k-1 centroids are kept, and in the new election only the voters (B') who voted for the discarded centroid are going to select two by voting again –a voting population smaller than B.

Example 9. Let us ask Algorithm 2 to find one more centroid of bag {Schnauzer, cat, other mammals, amphibian, reptile, lizard} for k = 2 and centroids (Schnauzer, lizard) as already found in Example 8. Inputs for Algorithm 2 are: B ={Schnauzer, cat, other mammals, amphibian, reptile, lizard}, n=2, centroids =(Schnauzer, lizard).

Step **C** computes, for every v in B, conf(Schnauzer, v), conf(lizard, v) and accumulates conf(Schnauzer, v) into totconf(Schnauzer) if conf(Schnauzer, v) < conf(lizard, v), into totconf(lizard) if conf(Schnauzer, v)>conf(lizard, v), or into totconfindif if conf(Schnauzer, v) =conf(lizard, v). These accumulations are shown in Table 3, which was built from left to right. The last column shows the final contents of accumulators totconf(Schnauzer) =1/2, totconf(lizard) =0, totconfindif =1/4 when step C.a finishes.

Now step C.b selects the worst of the two centroids: Schnauzer. Step C.c discards Schnauzer and finds who voted for Schnauzer (including indifferent voters). They are {Schnauzer, cat, other mammals, amphibian}. *They will be given another chance to vote.* Thus, B' is set to {Schnauzer, cat, other mammals, amphibian}.

Table 3. Step C.a of example 9. Columns show voters. They vote for Schnauzer or for lizard. The first two rows show their confusions. The selection of each voter (the smallest of the two confusions) accumulates in either totconf(Schnauzer) or totconf(lizard). If both confusions are the same, it gets accumulated into totconfindif. For instance, first voter (column Schnauzer) selects to vote for Schnauzer, so it adds 0 to the accumulator totconf(Schnauzer), shown in the same Schnauzer column. Second voter (cat) selects to vote for Schnauzer, so adds ¼ to totconf(Schnauzer). Other mammals, our third voter, votes for Schnauzer, so ¼ is added to totconf(Schnaunzer), which now is ½. Fourth voter (column amphibian) is an indifferent voter, so adds its confusion (1/4) to totconfindif. Reptile votes for lizard and adds 0 to totconf(lizard); the same happens to lizard. In the last column we find the final value of accumulators totconf(Schnauzer)=1/2, totconf(lizard)=0 and totconfindif=1/4: the final values when the vote of lizard, the last voter, is taken.

	Schnauzer	**cat**	**Other mammals**	**amphibian**	**reptile**	**lizard**
Schnauzer	0	¼	¼	¼	¼	½
lizard	3/4	½	½	¼	0	0
totconf(Schnauzer)	0	¼	½	½	½	½
totconf(lizard)	0	0	0	0	0	0
totconfindif	0	0	0	¼	¼	¼

[11] By definition of the centroid of a bag, in step D only members of B' participate in the election of the two centroids of B'.

Step D calls Algorithm 1 to find two centroids for *B*'. They are (Schnauzer, cat). Then, (lizard, Schnauzer, cat) is returned as the best three centroids of bag {Schnauzer, cat, other mammals, amphibian, reptile, lizard}. Note that the centroid Schnauzer, discarded in step C.b, reappeared again in step D. See Figure 4. This solution agrees with the exact solution found in §2.2. The three clusters are {lizard, reptile, amphibian}, {Schnauzer}, {cat, other mammals}. The indifferent values (see Table 1) were "other mammals" (selecting either Schnauzer or cat as its preferred centroid)[12] and amphibian (selecting as its preferred centroid Schnauzer, cat or lizard).

2.5 Finding (approximately) the centroids of a bag of values –two by two

Up to this point, we can find *k* centroids of a bag *B*. But, who sets the value *k*? How many centroids is it "best" to have? Let us call κ this "best" *k*.

We now give an algorithm for finding the κ centroids of a bag of qualitative values. Rigorously, they should be called "quasi-centroids," since they are not guaranteed to be the *real* centroids. The idea is to find one centroid for *B* (using §1.4), then two centroids (using §2.3), then the three centroids (using §2.4), and so on, at each time using §2.4 to find one more centroid, but to stop as soon as the condition of §2.6 is met: a sharp drop in the inconsistency. Notice that this algorithm (Algorithm 3) finds κ as well as the κ centroids of *B*.

Algorithm 3. Find the (approximate) centroids of a bag of qualitative values.

1. Find the centroid of B, with the help of §1.4. One centroid.
2. Find the two centroids of B, with the help of Algorithm 1 of §2.3.
3. Apply the test of §2.6 to see whether the inconsistency of the larger number of centroids is significantly inferior to the inconsistency of the smaller number of centroids.
    a. If that is the case, return the larger number of centroids as the answer. Stop.
    b. Otherwise, find the worst centroid and replace it with two new ones, using Algorithm 2 of §2.4.
4. Go to step 3. *End of Algorithm 3.*

2.6 How to select the appropriate number κ of centroids

In clustering or unsupervised pattern classification, there is often no good way to select an appropriate number of clusters, since frequently it is desired to minimize some similarity function inside each cluster, and simultaneously not to have too many clusters. That is our case. In these situations, an empirical choice is made. One of them is already mentioned in §2.1: minimize a linear combination of *k* (the number of centroids) and σ (the inconsistency). In this section we provide another empirical selection: stop adding centroids when a sharp decrease in inconsistency happens. Other selection criteria could be used.

The inconsistency of a bag decreases monotonically (Figure 4) as the number of centroids grows. We look for sharp descents in the inconsistency (say, a drop of at least 20%), and we select the number of centroids that contains the first sharp descent. In other words, adding more centroids will give only slight decreases in inconsistency.

In Figure 4, when we go from 1 centroid to 2, the inconsistency drops from 0.25 to 0.125, or a drop of 50%. Thus, we have found the first sharp drop and we keep these two centroids. There-

[12] conf(Schnauzer, other mammals) = conf(cat, other mammals) =¼, as per Table 1.

fore, we say that bag {Green lizard, German Shepherd, other dogs, Green lizard, cat, dog} has two centroids and an inconsistency of 0.125, and $\kappa = 2$.

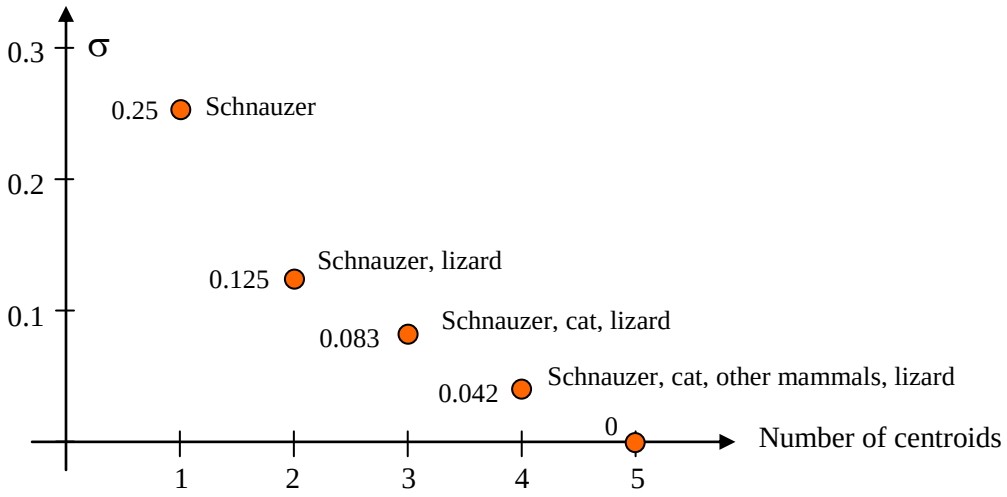

Figure 4. The inconsistency of bag {Schnauzer, cat, other mammals, amphibian, reptile, lizard} as a function of the number of centroids it is allowed to have. The centroid Schnauzer gives an inconsistency of 0.25. If the bag is allowed to have two centroids, these are (Schnauzer, lizard) and σ drops to 0.125. The three centroids are (Schnauzer, cat, lizard) with σ =0.083. Another trio of centroids is (Schnauzer, other mammals, lizard); still another trio is (Schnauzer, amphibian, lizard). The four centroids of the bag are (Schnauzer, cat, other mammals, lizard) with σ =0.042. There are other quartets of centroids (see text). Finally, the quintet (Schnauzer, cat, other mammals, amphibian, lizard) yields the lowest value, 0, for the bag's inconsistency

2.7 Experimental examples

This section presents two experiments, aimed at the exploitation of Internet search using the semantics provided by centroids of bags. They aim at improving the accuracy of searches, by using a taxonomy and inconsistency, so that the retrieved documents are more relevant to the user, given the words used for searching. For these examples we use the hierarchy of Figure 5. Since our metrics and tools have been recently invented, more realistic examples shall wait for a subsequent paper.

Software ✖ ♦ ♦
- "Programming Techniques" ✖✖♦
  - "Automatic Programming"
  - "Concurrent programming"
  - "Sequential Programming"
  - "Object-oriented programming" ✖
  - "Logic Programming" ✖ ♦
  - "Visual Programming"
- "Software Engineering" ✖ ♦♦
  - Requirements OR Specifications
  - "Design tools and techniques" ✖
  - "Coding Tools and Techniques" ✖
  - "Software Verification" OR "Program Verification"
  - Testing tools ✖
  - "Programming Environments"
  - Metrics
  - "Software Architectures"

Programming Languages ♦♦
"Language Constructs and features" OR polymorphism OR procedures OR frameworks
"Language processors" OR "code generators" OR debuggers
Operating Systems ♦♦
"Process management"
"Storage management"
"File system management" ♦
Reliability
Security OR protection OR "massive attack"
Performance
Miscellaneous
"Software Psychology"

Figure 5. A hierarchy of computer-related values. The values belonging to bag B of Experiment 1 are marked with ✖; those belonging to bag C of Experiment 2 are marked with ♦

Experiment 1 (Example 10)**.** Providing a rather large set of words to search for Web pages often renders null results. For instance, when Google searches with the bag B = {Software, "Programming Techniques", "Programming Techniques", "Object-oriented Programming", "Logic Programming", "Software Engineering", "Design Tools and Techniques", "Coding Tools and Techniques", "Testing tools"}, marked with ✖ in Figure 5, the answer has only 51 elements, with only 6 relevant answers (most are computer taxonomies). Will the centroids of this bag provide a better search result? How will these results will compare with "software", the "lowest common meaning" of the bag (the most precise value in the hierarchy [Figure 5] from which all the values in the bag are descendants)?

A. Searching with bag B provides 51 answers, of which only 6 are relevant (most are courses, computer classifications and syllabi). See Figure 6.

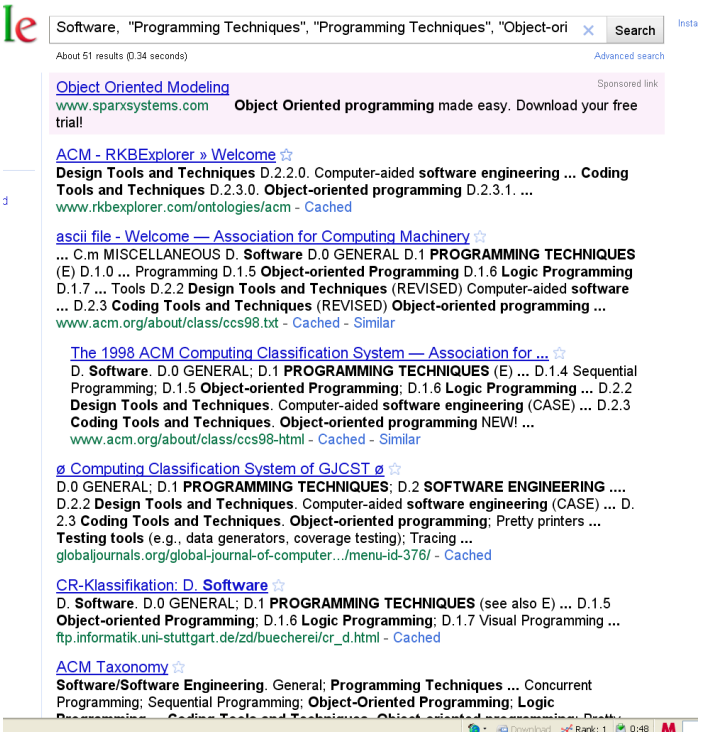

Figure 6. Experiment 1 searches with bag B = { Software, "Programming Techniques", "Programming Techniques", "Object-oriented Programming", "Logic Programming", "Software Engineering", "Design Tools and Techniques", "Coding Tools and Techniques", "Testing tools"} and obtains 51 answers, of which most are irrelevant (computer classification systems, courses and syllabi)

B. To find the centroids of the bag, we use §2.5 and §2.6. Two centroids are found: "object-oriented programming" and "testing tools". This agrees with the exact solution (if §2.3 were used). Searching with them gives 26,100 answers, of which about 82% are relevant (value manually estimated, Figure 7).

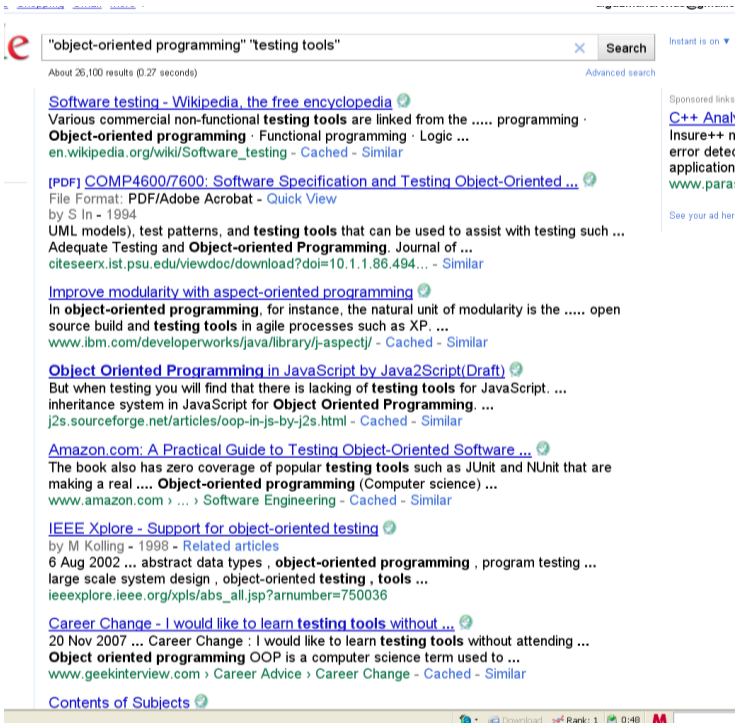

Figure 7. For Experiment 1, searching with the centroids "object-oriented programming" and "testing tools" of bag B = {Software. "Programming Techniques". "Programming Techniques". "Object-oriented Programming". "Logic Programming". "Software Engineering". "Design Tools and Techniques". "Coding Tools and Techniques". "Testing tools"} renders about 26,100 answers, of which about 82% are relevant (value estimated manually)

C. Bag B could have been considered as having just one centroid (not the two found). The method to find one centroid of a bag is given in §1.4. The centroid thus obtained is "object-oriented programming", with which we obtain about 2.3×10^6 answers, too many to be useful.
D. The lowest common ancestor of the bag is "software" (See Figure 5). Searching with "software" provides about 6×10^9 answers, far too many to be useful.
E. The results of A-D are compared in Table 4.

Table 4. Results of Experiment 1 (bag B or ✖)

	A. searching with all the keywords	**B. Searching with the two centroids**	**C. searching with the centroid (one)**	**D. Searching with the lowest common ancestor**
Documents obtained	51	26,100	2.3×10^6	6×10^9
precision	9%	82%	?	?

Experiment 2 (Example 11)**.** Different searches are performed trying to select a "good" set of replacements for many key words. Now, the complete bag of keywords is C = (Software, Software, "Programming Techniques", "Logic Programming", "Software Engineering", "Software Engineering", "Programming Languages", "Programming Languages", "Operating Systems", "Operating Systems", "File system management"), marked with ♦ in Figure 5.

A. Searching with the whole bag C renders 200 answers, of which about 13% are relevant (count manually made), most are courses and syllabus. See Figure 8.

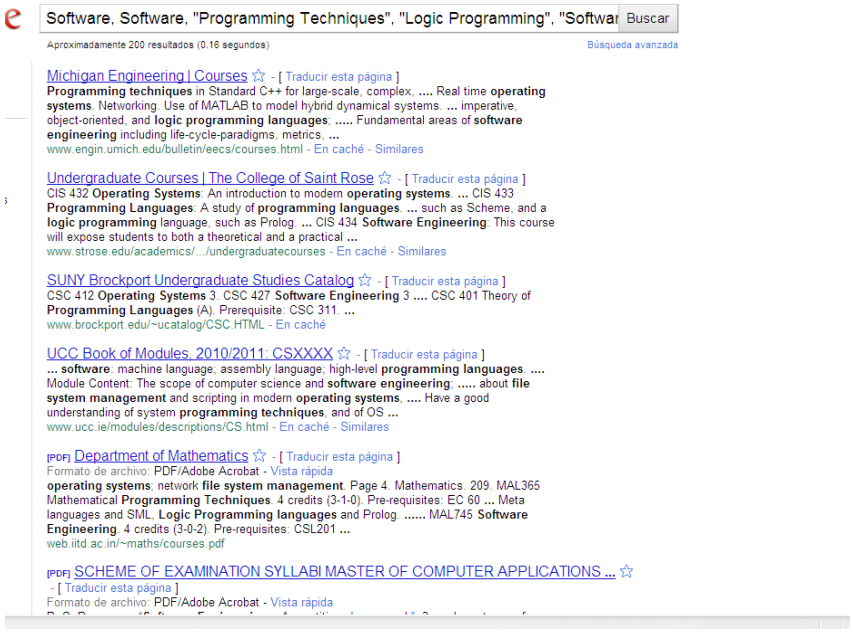

Figure 8. Search with the whole bag C produces 200 answers, with a relevance of about 13%

B. Using §2.5 and §2.6, we find the centroids of bag C to be two: "File System management" and "Logic Programming". Searching only with them renders 315 results (Figure 9), of which about 40% are relevant.

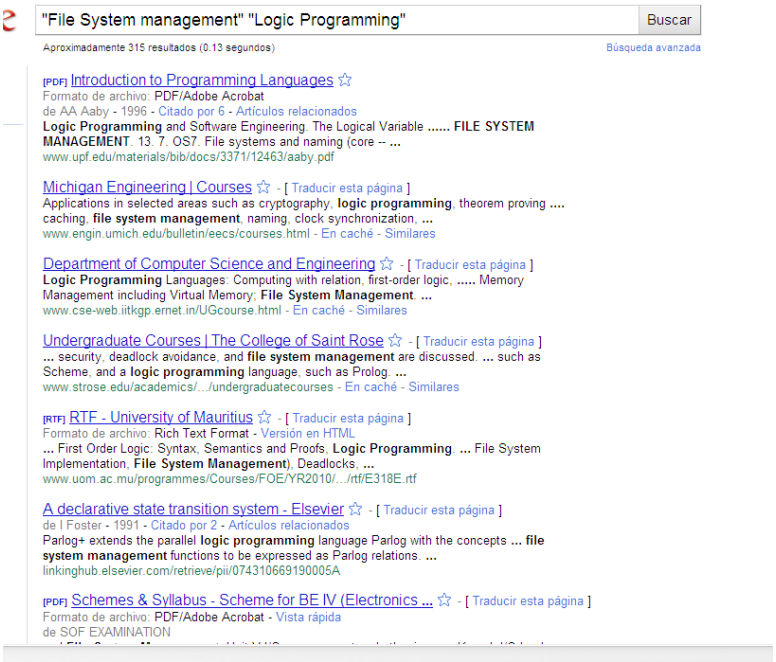

Figure 9. 315 results are obtained when searching with the two centroids "File System management" and "Logic Programming" of bag C of Example 11. These centroids were found by the method in §2.5, and the limit 2 was obtained using §2.6. Precision in this example is about 40% (manually computed)

C. If only one centroid is desired, it is "Logic Programming" (found by the method described in §1.4). Searching with it gives 1,010,000 results (Figure 10).
D. The lowest common ancestor (lowest common meaning) of the bag is "software" (See Figure 5). Searching with "software" provides about 6×10^9 answers, too many to be useful.
E. The results of A-D for Experiment 2 are compared and shown in Table 5.

Table 5. Results of Eperiment 2 (bag C or ♦)

	A. searching with all the keywords	**B. Searching with the two centroids**	**C. searching with the centroid (one)**	**D. Searching with the lowest common ancestor**
Documents obtained	200	315	1,010,000	6×10^9
precision	13%	40%	?	?

As we see, the results of these restricted experiments seem to indicate that search will improve with the use of the centroids of the search terms. They are not conclusive; more of them need to be performed before the real value of this recently discovered method (or a suitable adaptation of *inconsistency*) for accurate search using "augmented keywords" or "centroids" can be assessed.

3. Conclusions and discussion

Conclusions. Our paper [6], summarized in §1.4, provides a way, using conf, to obtain the centroid (r^*) and inconsistency (σ) of a bag of symbolic values reported (by several observers) for the same observed property of the same object**.** r^* and σ are crisp, not fuzzy, values. In addition, σ is a number, not a qualitative assessment. The observers are equally credible, so that their dissimilar observations are due to the difference in their methods (or instruments) of observation. When methods are crude, the observed values have "limited precision" (are located close to the root of the hierarchy). Other observers could obtain more detailed measurements, positioned deeper in the hierarchy. The *inconsistency* of the bag measures how far apart the testimonies (the values) in the bag are; it is a number between 0 and 1, not just 0 or 1.

Work reported in this chapter solves the same problem when it is possible for the bag to have *several* centroids. Each of the centroids induces a *cluster* of the bag**:** those elements for which that centroid minimizes the confusion conf(centroid, element). In this way, we can cluster qualitative values in such a way to produce a small inconsistency. The method also shows *indifferent values*, which could belong to more than one cluster.

Chapter 2 describes (§2.3) an exact method for computing the two centroids of a bag. It also describes (§2.2) an exact method for computing the *k* centroids of a bag, and an approximate, but faster, method (§2.6) to do the same.

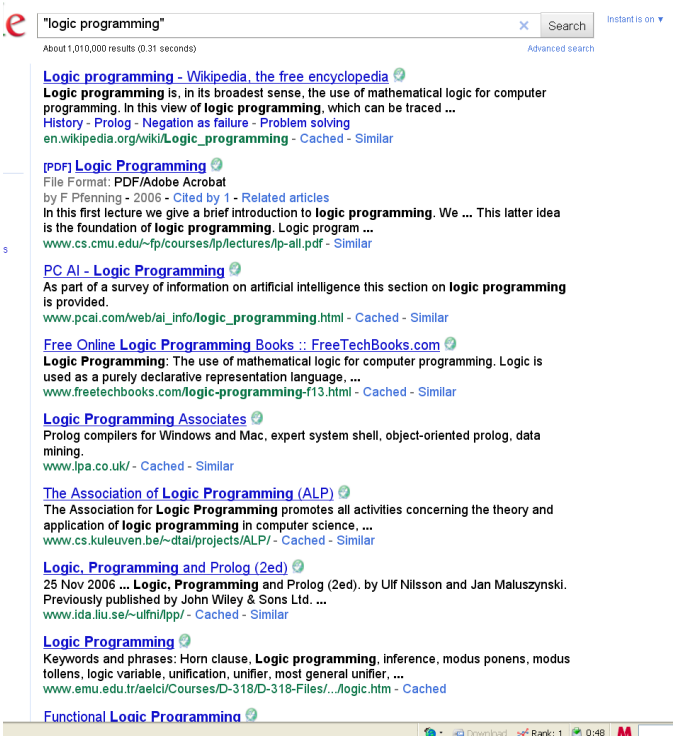

Figure 10. For Experiment 2, the figure shows the results when the centroid "Logic Programming" of bag C is used

3.1 Discussion

Numeric values have a meaning "of their own," so it is easy to agree in the difference between, say, 7 and 13. Opposite to that, symbolic values (such as Mexico City or airplane) must have a *context* against which their closeness or difference can be gauged. This context is provided by hierarchies, over which the function conf is defined. The function conf(*r*, *s*) measures the confusion when symbolic value *r* is used instead of the real, or intended, value *s*. The function conf always has as a parameter a hierarchy of possible or likely symbolic values. This hierarchy can be regarded as a simplified ontology, where only the relations "subset" and "member" are used. In principle, all the relations in an ontology could be used by a (modified) function conf, but this issue remains unexplored.

The similarity function conf is not a distance. Many other similarity functions exist (for instance, the Jacquard coefficient), and which is better depends on its purpose, on its intended use. conf is useful when qualitative values have different precisions, so that some values are refinements or "extensions" of another values (Doberman and Schnauzer are "refinements" of dog), and there are levels or tiers of values having the same "granularity" (or "precision"). [Therefore, they can best be arranged in a *hierarchy*].

When several numeric measurements are performed over the same property of the same object, and a bag of different values is obtained, how can we measure the inconsistency of that bag?[13] What is the most likely value for the property? Ordinary Logic tells us that there is no such

[13] I mean a bag of assertions, such as {the length is 7.2m; the length is 7.29m; the length is 6.85m; the length is 7m}.

value, and the inconsistency of the bag is 1 (false), since the measurements are incompatible. Dempster-Schafer theory [3, 12] uses the likelihood of different measurers telling the truth (their credibility), to compute the most likely value. Fuzzy logic can also be used. Other researchers [1, 7] count how many predicates are violated by the bag of observations, and that count is the inconsistency of the bag. Still others try to remove assertions from an inconsistent set; the number of assertions removed until the remaining set becomes consistent tells how inconsistent the original set was. For most of us, the most likely value for that property, given a bag of measurements, is just the average or centroid of these values, and the inconsistency of the bag is just the variance σ of the observations.

Bags of values referring to a single-valued variable (such as "*x* is the father of Emille") can have just one centroid (a bag of opinions about who is the father of Emille), whereas other bags can have more than one centroid (a bag of opinions about who are the friends of Emille). When the problem allows the existence of several centroids, the exposed methods relying in confusion and total confusion easily express the preference of voters (elements of the bags) about candidates (possible centroids), so that the task of finding a reasonable number of centroids attains relevance. We solve it reasonably by a heuristic method: seeking a sharp drop in the inconsistency as the number of allowed centroids grows. The solution is quantitative and crisp.

It should be emphasized that an asserted value obtained by an observer (such as *Doberman* in "John's pet was a Doberman") represents not only itself, but all the values from it up to the root of the hierarchy: Doberman, dog, mammal, vertebrate, and animal. This is because the observer, having all these values to select when reporting his observation, reports the most precise value.

Acknowledgments. Work was in part supported by CONACYT grant 43377 and by SNI. Comments by the reviewers increased the quality of this work.

References

1. Byrne, E., & Hunter, A. (2005) Evaluating violations of expectations to find exceptional information, *Data and Knowledge Engineering*, **54**(2):97-120.
2. Cuevas, A and Guzman-Arenas, A. (2008) A language and algorithm for automatic merging of ontologies. Chapter of the book *Handbook of Ontologies for Business Interaction*, Peter Rittgen, ed. IGI Global (formerly Idea Group Inc.), USA. 381-404
3. Dempster, Arthur P. (1968) A generalization of Bayesian inference, *Journal of the Royal Statistical Society*, Series B, Vol. **30**, pp. 205-247.
4. Guzman-Arenas, A., and Levachkine, S. (2004) Hierarchies Measuring Qualitative Variables. *Lecture Notes in Computer Science* LNCS **2945**, Springer-Verlag. 262-274. http://www.divshare.com/download/6271736-09c
5. Guzman-Arenas, A., and Levachkine, S. (2004) Graduated errors in approximate queries using hierarchies and ordered sets. *Lecture Notes in Artificial Intelligence* LNAI **2972**, Springer-Verlag. 139-148. ISSN 0302-9743
6. Adolfo Guzman-Arenas, Adriana Jimenez. (2010) Obtaining the consensus and inconsistency among a set of assertions on a qualitative attribute. *Journal Expert Systems with Applications* **37**, 158-164.
7. Hunter, A., and Konieczny, S. Measuring inconsistency through minimal inconsistent sets. *Proceedings of the* 11th *International Conference on Knowledge Representation* (KR'08), AAAI Press

8. Jimenez, A. (in press) Characterization and measurement of logical properties of qualitative values organized in hierarchies. Ph. D. Thesis. CIC-IPN. In Spanish.
9. Levachkine, S, Guzman-Arenas, A. and de Gyves, V.P. (2005) The semantics of confusion in hierarchies: from theory to practice. In *Contributions to ICCS 05 13th International Conference on Conceptual Structures: common semantics for sharing knowledge*. Kassel, Germany. 94-107 http://www.divshare.com/download/6257877-9a7
10. Levachkine, S., and Guzman-Arenas, A. (2007) Hierarchy as a new data type for qualitative variables. *Journal Expert Systems with Applications* **32**, 3.
11. Luo, P., Xiong, H., Zhan, G., Wu, J., and Shi, Z. (2009) Information-theoretic distances for cluster validation: generalization and normalization. *IEEE Trans. on Knowledge and Data Engineering,* **21,** 9, 1249-1262.
12. Shafer, Glenn. (1979) A Mathematical Theory of Evidence. Princeton University Press.
13. Ruiz-Shulclóper, J., Guzman-Arenas, A. and Martinez-Trinidad, F. (1999) Logical combinatorial approach to Pattern Recognition: supervised classification. Editorial Politécnica. (In Spanish)
14. Xiaoxin Yin, Jiawei Han, Philip S. Fu. (2008) Truth Discovery with multiple conflicting information providers on the Web. *IEEE Trans. KDE* **20**, 6, 796-808.