Disordered Databases and Ordered Explanations

T. J. Parsons
British Aerospace, Hatfield

5th July 1988

Keywords: Unsupervised Machine Learning, conceptual clustering, entropy, contextual information.

Abstract

'Learning begins with organized knowledge which grows and becomes better organized' - Charnick and McDermont.

The paper adopts an 'information theoretic' approach to an area of Machine Learning that is known as unsupervised, conceptual clustering. The approach is developed in the 'Vision Domain', databases of upto 1000 entries being analysed containing information derived from digitised colour images.

Typically, the field of supervised Machine Learning attempts to develop programs which learn from example and counter-example, failure or even instruction. Unsupervised Machine Learning algorithms adopt a different approach. The programs simply try to discover patterns, trends and hierachies of relationships which, although not explicitly stated, are never the less present within a database.

In terms of the Vision problem, this corresponds to searching for significant classes or clusterings implicit in the attributes and relationships of image regions or features.

The paper develops further the Knowledge representation structure and transformation rules presented at AVC87 [2]. A Hyperdigraph structure for encoding relational knowledge is used, these being suitable for treatment from an information theoretic viewpoint. By allowing non-disjoint classes to exist in this structure, it is shown that the transformation rules reduce the database to a minimal state of complexity. The uniqueness of this state allows classes of features, formed by the clustering process, to be ranked in importance by the increase in complexity within the hyperdigraph structure that their destruction would invoke.

As regards implementation, the clustering process may be regarded as a "many-pattern / many-object" matching problem and the efficiency of such algorithms is discussed.

Results based on the analysis of real images are presented.

1 Introduction

Any data structure encodes information by the discrete set of states in which it can exist. For extremely large structures, such as might be found in a database of many thousand entries, these states are analogous to the *microstates* of physical systems within the context of statistical physics.

The gross behaviour (*macrostates*) of such systems is deducible from an examination of these microstates. Concepts such as disorder, entropy and information content arise naturally and are of fundamental importance in the study of physics.

The paper uses these fundamental concepts to analyse the state of such large scale structures using an information theoretic approach. Conceptual clustering, understood to be the replacement of relationships between facts by relationships between classes of facts, is interpreted as a 'structuring' or reduction in complexity of the knowledge encoding structure.

The concept of a 'minimally disordered database' is shown to be the consequence of freely allowing non-disjoint classes. Additionally it is shown how this concept enables a contextually dependent measure of 'importance' to be derived for clusterings within the database, the context being the knowledge encoding structure in which the cluster resides.

A mechanism for quantifying the belief in

any structure to be imposed upon the database (an hypothesis) is discussed.

Experimental results based on the reduction of a database containing some 1000 facts, gleaned from a region based analysis of colour images, are presented. One consequence of using large amounts of real data is the formation of classes which have a large intersection set. The tendency to complete these sets (so that they become identical) is presented as a sensible automatic query mechanism to supply missing information to the database.

Significantly, the approach outlined in this paper avoids the mechanism of Probabilistic Reasoning and its associated problems [1]. Also, the derived measures are a function of the state of the context of the database in which they are found. The derived measures will change as the database evolves with time. Additionally, an item might have a completely different value associated with it when viewed in the context of another database.

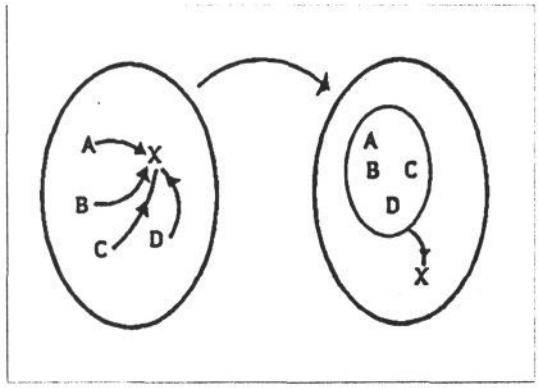

Figure 1: Complexity change on forming a class w.r.t. X

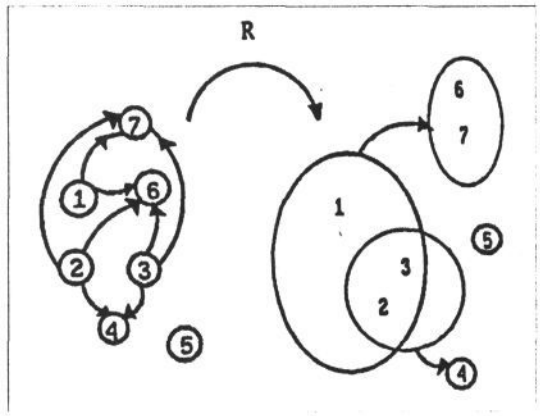

Figure 2: Unstructured database changing on creating classes α, β, γ

2 The Representation of Knowledge

Following Berge [3] we introduce the term "Hyper-digraph" to describe a directed graph structure in which nodes of the graph may themselves be expanded as Hyper-digraph structures.

If nodes are interpreted as items within a database and arcs as relationships between these items, then hyperdigraphs permit the encoding of both attribute and relational descriptions at many levels of abstraction to exist simultaneously within the database. This is an extremely desirable feature when forming conceptual cluestering algorithms [5].

Hyper-digraphs form the basic knowledge representation structure used in this paper for a number of reasons. Firstly, it is the simplest structure that is general enough to encode information at many levels of abstraction, and this simplicity is useful when adopting an information theoretic approach by attempting to quantify the information content of the structure. Secondly, schemata and conceptual graphs may be considered as specialized instances of a Hyper-digraph structure; thus any body of knowledge encoded in these ways is equally capable of being represented as a Hyperdigraph structure. This is illustrated later in the paper when Winston's conceptual graph of an arch is treated as a labelled digraph structure.

Two particular restrictions on the concept of an unlabelled digraph will be of interest. A digraph may contain both loop arcs and parallel arcs, however if it contains no parallel arcs it is termed a 'restricted digraph' and if both loop arcs and parallel arcs are absent, then it is termed a 'simple digraph'.

It is assumed that the number of nodes is finite and not zero, and that the number of arcs in the digraph structure is finite but may be zero.

3 Terms and Definitions

The concept of a *cluster* or *class* is introduced.

1. A given subset of the above database forms a cluster or class if each member of the class behaves identically with respect to a second subset of the database. This second subset may be the same, partially the same or completely different, in terms of members, from the first class.

   The act of replacing individual relationships between members of a class and the database by a relationship between that class and the database constitutes the act of "class formation" and this process is illustrated in Fig. 1. Class formation is often refered to as "clustering". Clustering, as an operation upon the database, leads to the formation of non-disjoint classes.

2. An Assertion, Rule or Hypothesis that is to be added or removed from the database is termed an *item*.

3. Members of a 'class' may or may not behave identically with respect to each other, but those which do may be said to form a subclass of that class.

The terms 'class formation' and 'clustering' are similar. However we will tend to use the term 'cluster' to imply class formation as a result of shared spatial properties. Algebraically it may be represented by the transformation:

$$R(A,X) + R(B,X) + ... \rightarrow R(ABC..,X) \quad (1)$$

Where $R(A,X)$ indicates a relationship R between the two nodes or classes A and X. The symbol '+' should be read as 'and'. The formation of a class α within a database may be considered to be the result of an operation on that database. If that operator is $\hat{R}$ acting on the database D then formally we write:

$$\hat{R}_\alpha D \rightarrow D_\alpha \quad (2)$$

By the definition of a class, introduced above, it is easily shown that the creation of a certain class does not affect the ability of each member of that class to be considered as members of another class, that is, the formed classes are not necessarily disjoint. Hence the formation of a set of classes in a database, whether they overlap or not, results in the formation of a simplified database which is independent of the order in which classes are created, i.e., if $C(D)$ is the complexity of a database, then :

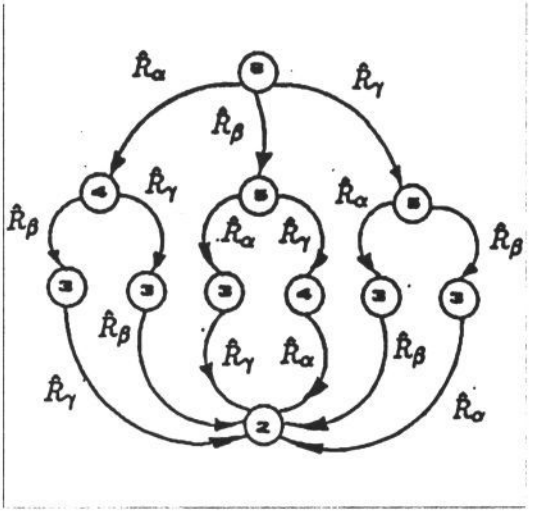

Figure 3: Tracing Arc Complexity changes on creating α, β, γ

$$C(D_{\alpha\beta\gamma}) = C(D_{\beta\gamma\alpha}) = C(D_{\gamma\beta\alpha})...etc \quad (3)$$

Fig. 2, presents a small example database and illustrates the reduction in complexity incurred under the transformation $\hat{R}_{\alpha\beta\gamma}$ and Fig. 3 traces the changes in the number of arcs in the database on the formation of these classes to illustrate equation 3.

In its initial state, the structure has eight arcs. Regardless of the sequence of class formation, i.e. which ever path is chosen in Fig. 3, the final state has just 2 arcs.

4 The Database Monitor Program (DMP)

Equation 3 has extremely important implications. Because, on repeated class formation within the database, each step induces a reduction in the complexity of the structure (or at least it can never give an increase) and because the formation of any given class does not inhibit the formation of any subsequent class or modify the members of that subsequent class, then we can guarantee to always find a unique final state beyond which there is no further change under the operation of class formation. No local minima exist within the search space, hence the final state of the database is independent both of the order of class formation and the order of the database.

This final state is termed the *minimal complexity state* and, as will become clear later, it may be formally defined as that form of the database containing the minimum number of microstates consistent with preserving the information content within it.

The DMP algorithm is based upon the concept of repeated class formation and guarantees that any labelled hyper-digraph structure will be reduced to a minimal state using the arguments developed in the previous section.

Rules 4 to 8 summarize the set of transformations that the DMP algorithm induces on a labelled hyper-digraph structure.

$$R(A,B) + R(a,b) \rightarrow R(A,B) \quad (4)$$
$$R(A,B) + R(A,C) \rightarrow R(A,BC) \quad (5)$$
$$R(C,A) + R(B,A) \rightarrow R(BC,A) \quad (6)$$
$$R_1(A,B) + R_2(A,B) \rightarrow R_1R_2(A,B) \quad (7)$$
$$U(A) + U(B) \rightarrow U(AB) \quad (8)$$

$U(A)$ indicates a unary relationship on A.

Rule 4 illustrates the removal of redundant information from within the database, If $a \subset A$ and $b \subset B$ then the relationship $R(a,b)$ may be deleted without any loss of information. The information is preserved in the relationship on the super-set.

Rules 5 and Rule 6 illustrate the fundamental process of clustering. If two classes have the same relationship w.r.t. a set, then they may be considered as forming a class BC *by virtue of that relationship.*

Rule 7 illustrates the process of relationship concatenation. If class A has two different relationships R_1 and R_2 with respect to "B", then the two relationships may be concatenated and the number of entries in the database reduced by one, again without any information loss.

Rule 8 illustrates a clustering process based on class attributes. If A has a unary relationship with respect to itself (an attribute) and B has a similar attribute, then AB form a class by virtue of this common attribute, again without any information loss. An example might be the formation of the class of "green" regions within an image.

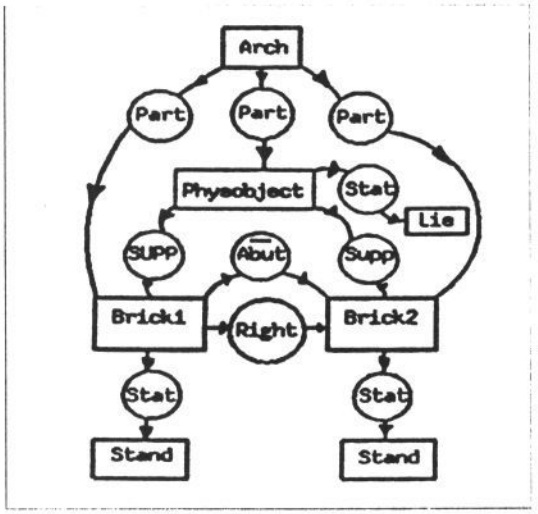

Figure 4: Bipartite Conceptual Graph of an Arch

5 Winston's Arch

Figs 4 and 5 illustrate the operation of the DMP algorithm in a less abstract manner. Fig 5 is a bipartite graph representation of Winston's conceptual graph of an arch whereby the essence of the concept arch is deemed to be the structure of a 'physical object being supported by two bricks one of which is right of the other but not touching'. The bipartite graph may be considered to be just a labelled digraph and fig 5 represents the results on passing this structure through the DMP.

Firstly, note that the program suggests that both bricks (B1 and B2) may be considered as a class by virtue of the fact that they support a physical object (P.O.). Secondly, note that both the bricks and the physical objects themselves may be considered as a hyper-node by virtue of the fact that they form part of the arch. Finally note that the initially separate relationships 'not-abut' and 'right' have been concatenated into one relationship using transformation 7. This form of the database is minimal. No further reduction is possible without information loss.

It is the aim of the paper to show that classes formed using the ideas presented so far in the paper may, on analysis of the Hyperdigraph structure, be assigned measures which reflect the "Importance" of a class in the context of the rest of the database. In the next section the idea of "Importance" is quantified.

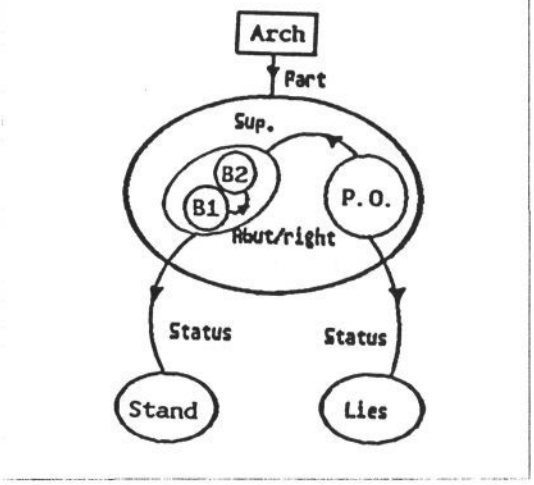

Figure 5: Conceptual Graph of an Arch – minimal form

The technique can be used to show that, on a normalised scale between 0 and 1, the redundancy of information in Winston's original conceptual graph of an Arch is 0.3

The DMP is central to the ideas presented in this paper because it is intended that the importance of an item will be calculated, with respect to a database, by noting the change in entropy of that database on introduction of a new item. Before this can be done however, it is necessary to ensure that the database is in a unique minimal state.

6 Derivation of Measures

The following analysis yields an expression for the state of entropy of a database modelled by the digraph structure. Consider a database capable of existing in a large number of states and changing its internal structure under the addition or removal of items (Assertions, Rules or Hypotheses). The question is posed 'How does the entropy of the database change under such transformations? '.

Classical Physics defines the entropy of a system in terms of the number of Microstates in which it can exist, that is:

$$S = K(\ln \Omega) \tag{9}$$

where (Ω) is the number of Microstates.

The task remains to develop a formula yielding the number of Microstates in which a digraph of N nodes and m arcs may exist.

Given N nodes there are $N(N-1)$ possible places to insert an arc and N places to insert a loop. Hence the number of 'slots' P in which an arc may be placed is:

$$P = N^2 \tag{10}$$

Some thought shows that the number of distinct ways of placing m arcs into P slots (the number of microstates) is:

$$\Omega(m) = \frac{P!}{m!(P-m)!} \tag{11}$$

and hence the expression for entropy becomes

$$S = K \ln \left[\frac{P!}{(m!(P-m)!)} \right] \tag{12}$$

When adding or removing items to the database, the number of arcs and nodes is changed and a new value of the entropy measure results:

$$S_2 = K \ln \left[\frac{P_2!}{(m_2!(P_2-m_2)!)} \right] \tag{13}$$

where m_2 and P_2 are related to the number of arcs and nodes in the digraph after the addition of an item.

If P_1 and m_1 pertain to the state of the database before the addition of these items, then the associated change in entropy is:

$$\delta S = K \ln \left[\frac{P_1!m_2!(P_2-m_2)!}{(P_2!m_1!(P_1-m_1)!)} \right] \tag{14}$$

Sterling's approximation may be used to analyse the behaviour of equation 12 for large databases.

If both P and m are large then:

$$\ln(n!) \approx n\ln(n) - n \tag{15}$$

where n is either P or m. The expression for entropy becomes:

$$S \approx K \left[P\ln(P) - m\ln(m) - (P-m)\ln(P-m) \right] \tag{16}$$

Note, by substitution in equation 16, that the function is symmetric in that if either $m \to P$ or $m \to 0$ then:

$$S \to 0 \tag{17}$$

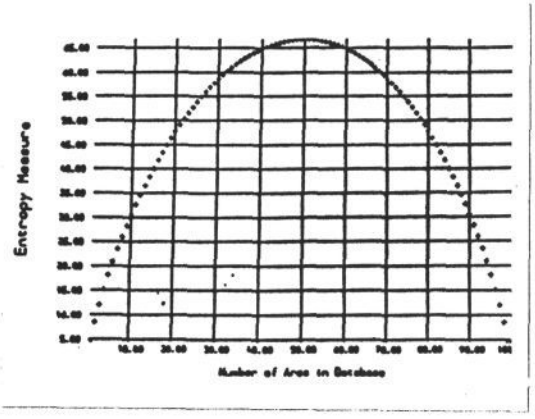

Figure 6: Diagram showing how entropy changes with the number of arcs in the database

and differentiating entropy with respect to the number of arcs yields:

$$\frac{\partial S}{\partial m} \approx -K\left[\ln(m)\ln(P-m)\right] \qquad (18)$$

The function thus behaves as illustrated in Fig. 6. The underlying symmetry of the function reflects the essential duality of graph structures, in that the presence or absence of an arc may represent information.

The function illustrated in Fig. 6 warrants further discussion. At first inspection the symmetry about point $m = P/2$ is worrying because it implies an ambiguity in our measure of entropy.

The number of arcs in our database is related to the number of nodes. The minimum number of arcs the database can have is N whilst the maximum number it may have is N^2 (N is the number of nodes). Thus the point of symmetry $N^2/2$ lies in between this possible number of arcs for all N > 2 and one might hence indeed expect the entropy measure to be ambiguous. However, when one more arc is added to the minimum number of arcs $m_{min} = N$, the DMP, as discussed in section 4, will recognize the existence of a class with two members and will simplify the knowledge base accordingly. The overall effect of the DMP is to ensure that the entropy measure returned lies in that monotonically increasing region of the curve below $N^2/2$ and labelled "A" in Fig. 6 An unambiguous measure of the state of entropy for the knowledge base is thus assured.

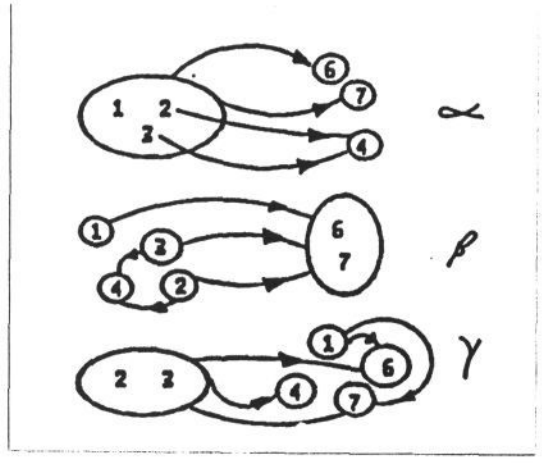

Figure 7: Class concepts and associated Importance values

7 'Importance' as a Measure

Once a class has been formed within a minimal database by the action of the DMP algorithm, it's 'importance' may be calculated by considering the displacement from this minimal state that its 'destruction' would induce upon that database. This is the measure given by equation 19

$$\text{Importance} = 1 - \frac{\ln(\Omega_{min})}{\ln\Omega} \qquad (19)$$

However, when adding an item to the database, an estimate of it's 'importance' may be obtained by calculating the change in complexity that occurs compared with the maximum change that could be possible at that point (that is, reduction to a two node, single arc relation). As each item is added to the database it becomes necessary to reform the minimal state in order to calculate measures of importance for the new items and to re-calculate the 'importances' associated with any formed classes. The ability to achieve this incrementally is important.

Fig. 7 illustrates the difference between the two measures.

Three classes α, β, γ have again been formed by the DMP and class α might be considered important with respect to the database because of its size.

In this example the estimates of importance turn out to be 0.77, 0.31 and 0.19 for classes

α, β and γ respectively and class α is indeed more important than classes β and γ.

However, once all the classes are formed and the database exists in its minimal state, equation 19 can be used to calculate the importance values 0.56, 0.38, 0.19 for α, β and γ respectively. The importance of class α has been changed by the context of class β within the database, which overlaps with class α considerably.

8 Measures for labelled Structures

The analysis performed in the previous section may be extended to include labelled digraphs by noting that such a structure may be effectively decomposed into a set of singly labelled digraphs. If there are L labels then:

$$S = K \ln \left(\sum_{i=1}^{L} \frac{P!}{m_i!(P - m_i)!} \right) \qquad (20)$$

where m_i is the number of arcs of a given label.

The above analysis assumes that the existence of a label L_1 between two nodes does not preclude the existence of a label L_2 between those same two nodes, i.e., the labels are assumed to be independent. In the system described here, which deals only with certain information, contradictory information already within the database is ignored and information already within the database, which is contradicted by new information, is updated.

9 Implementation and Efficiency

The reduction transformations 4 to 8 may be considered to be a set of rules matching upon specific entries within the database. Viewed in this light, the DMP algorithm is an implementation of the "Many-Pattern / Many-Object" matching problem well known in the field of Expert Systems. The time complexity of such a system, for our application, is roughly $O(N^2)$, where N is the number of entries within the database. Each fact neeeds to be paired with each other fact within the database to determine which rules are now eligible to fire.

One pass through the Database, matching against the transformations given in section 4, is termed a 'firing of the Rule Set' . In general the database will require n fireings of the Rule Set before the minimal complexity state is reached.

Rete [4] provided an optimal improvement to this algorithm by making use of structures which preserved match information for each clause in a rule. This algorithm essentially removes the necessity to re-compute clause matches which have already been calculated elsewhere. After an initial building phase, rules eligible to fire are determined from an examination of the match structures and this is a much faster process than a complete database search. The structures are updated as new items are added to the database and such an algorithm renders Expert Systems with many thousands of facts and many hundreds of rules achievable.

For this reason time was taken to code a Version of Rete to Implement the DMP algorithm. However, because each of the clauses in the rules given in section 4 may potentially match against most of the database, such structures were found to approach the size of the database itself. The overheads associated with compiling, maintaining and updating these structures were such that, in practice, little increase in speed was actually observed.

The overriding requirement for such a large system was the need to avoid an exhaustive search each time a new item was added to the database. This was achieved by employing an "indexed database" algorithm which restricted the search to an area of the database where the fact of interest was known to exist. The results presented in the next section employed this type of algorithm.

It is evident that a direct coding of the theory proposed earlier in the paper would not lead to a feasible algorithm. The requirement that sets are allowed to be non-disjoint implies that all intermediate class structures built by the transformations discussed in section 4 are preserved so that they may take part in the formation of other classes within the database. In practice it is observed that many of the intermediate classes become involved with clusterings such that the resultant classes are identical. At this point the size of the database reduces as redundant information is removed,

however this reduction does not occur until the database has grown considerably.

Databases with an initial number of entries of the order of several hundred facts expand rapidly and soon approach the limits of working memory. Processing time soon becomes unacceptable.

Fortunately, it is unecessary to keep all clusterings in working memory throughout the operation of the DMP. It is possible to recognise when a class will no longer be involved with any further transformations and to remove the structure to disc at this point. This renders an analysis of Large Scale Databases (in the order of thousands of entries) achievable.

The DMP fuses tuples of classes to form the hyper-classes which are of interest to this paper. As soon as sets are eligible to fuse, they do so and after n fireings of the Rule Set a whole spectrum of classes is produced with orders ranging from 2^n down to $n+1$, (assuming that the classes are always eligible to fuse).

It can be shown that no given relationship $R(x, y)$ will fuse with a relationship involving sets of higher order because if it can, it will have already fused with one of the tuple components of that higher order set. Conversely, after n fireings of the Rule Set, classes with order $\leq n$ may be ignored because these sets can now never be involved with future fireings. It is these structures that are removed to disc, keeping the contents of working memory acceptably small.

10 Results

Fig 8 illustrates one of the large database of images used within the Alvey MMI007 consortium. This Colour image was segmented into 198 regions and a database of some 1000 entries was compiled using a variety of Image processing routines. Each of these entries stored some fact about the region, either an attribute such as colour, brightness and shape, or a relationship, such as it's spatial positioning with respect to other regions within the image. Fifteen different relationships were represented in total.

This database was subject to the transformations discussed earlier in the paper and the resultant classes ranked in terms of the "Importance" value quantified in section 6.

Of the class structures produced by the DMP, only four were assigned values significantly greater that all others within the database and, furthermore, these results were found to be identical, *regardless of the initial ordering of facts.*

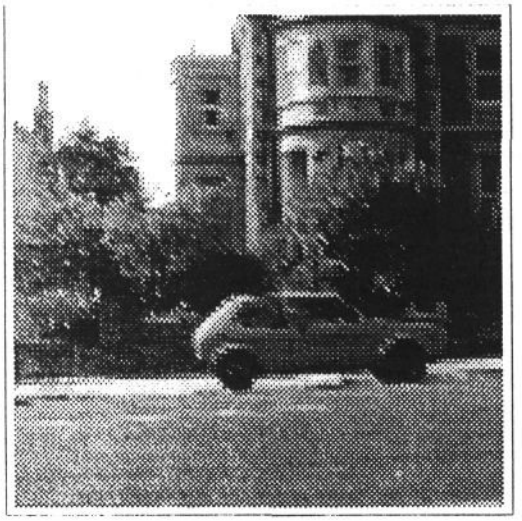

Figure 8: Example Database Image

Independence of results on the initial ordering of the database is evidently an essential requirment if any significance is to be assigned to the emergent class structures. Therefore, to test this result more thoroughly, the DMP was run seventeen different times on different versions of the database, each version being randomized by interchanging facts ranging from several hundred times to many thousands of times. Each time the results from the DMP were identical.

Considering the structural relational digraph corresponding to Fig 8, the most important class was deemed to be the long vertical structure associated with the side of the house (to be found some way above the rear window of the car in the foreground). This structure was interesting, not because it was 'long' and 'thin' but because it was 'adjacent' and 'leftof' an unusually large number of other regions. The segmented region corresponding to 'sky' was deemed to be interesting because it was found to be adjacent to an unusually large number of other regions, as was the boundary of the bushes immediately in front of the house. Also of interest was the group of regions corresponding to the isolated house window, basically because they were all surrounded by the same larger 'wall' region.

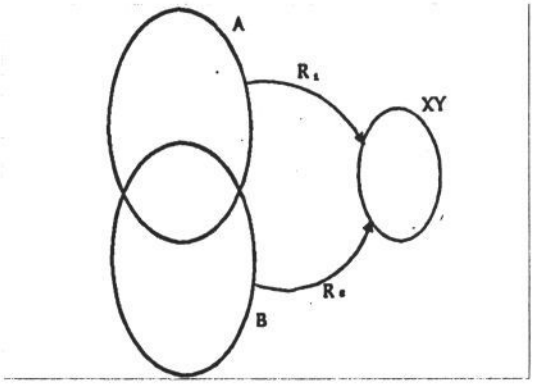

Figure 9: An example of Overlapping Classes

Unfortunately, the car in the foreground is not considered to be important in this example but a look at the segmentation map reveals why. This region of the image is badly under-segmented, the 'car windows' merging with adjacent 'bushes' and the 'car wheels' merging with the adjacent 'road structure'. This under-segmentation denies any significance that might be placed on structural relationships in this area of the image and illustrates the fact that, whilst the algorithm effectively builds up structures from over-segmented regions, it is unable to use under-segmented regions.

11 Missing Information

A further analysis of the database considered so far and databases corresponding to other colour images reveals that, after application of the DMP, some large classes emerge which are very similar, bar from a few members.

Consider Fig 9 which illustrates two classes with such a large intersection set. Because a large part of both sets behaves in the same way with respect to XY, it is indeed possible that the only reason that they emerged as seperate sets within the database was because of missing information. To complete this information we need to ask if the following two relationships are true?

$$R_2(A - A \cap B, XY)$$

$$R_1(B - A \cap B, XY)$$

In general, both class XY may be treated as two overlapping sets and the relationships themselves must be considered to be 'relationship sets'

The generalised questions, are hence.

$$(R_1 - R_1 \cap R_2)((B - A \cap B), X)$$

$$(R_2 - R_1 \cap R_2)((A - A \cap B), Y)$$

In the current implementation, the database is scaned and questions of the above form are asked of the user when there is sufficient intersection between the two sets. The user answers these questions by looking both at the database and the coloured image. Initial results suggest that the questions asked via this mechanism are often found to be pertainent and it is suggested that this 'tendency to complete sets' mechanism may provide a basis for a 'top-down' automatic interrogation system.

12 Belief in a Hypothesis

So far we have attempted to quantify the importance of a class by virtue of the simplification it induces within the database. A Hypothesis may be considered as a potential class, it is simply the suggestion that a given subset of nodes within a database may be considered as forming a class and acting as a unit with respect to some other node class within the database.

Thus a hypothesis may have a dramatic effect on the complexity of a database by virtue of the simplification that such a formation would incur. A number of both arcs and nodes would be 'hidden' from the database within the suggested class.

For clarity, a particular example is given. Consider the situation of a relational database describing a complex scene containing a car. The identification of a large sub-section of the relational graph containing the structural description of a car may be be considered as a class concept named "car". The consideration of the single class "car" as opposed to the complexity it may hide induces a large simplification in our database and hence the hypothesis would be given a large belief value. Expressed in a slightly different way, the class concept "car" explains a large portion of our relational database and hence has a large value of belief associated with it.

A measure of belief in the hypothesis to be imposed upon the database is:

$$Belief = \frac{\delta S}{S_{min}} \qquad (21)$$

Where δS is the reduction in entropy as defined earlier in the paper. However in this case 'Belief' is quantified by examining the *further* reduction to an already minimal database that the Hypothesis would incur.

Realistically, any hypothesis imposed upon a database would predict the occurrence of relationships which are either not seen or are directly contradictory to relationships already within that database. In the first case the hypothesis may be used to supplement the database with information directly pertinent to that hypothesis. Expressed differently the system is postulating "If this hypothesis is true then these facts are predicted to be true ... let's see if they are". This mode of operation is entitled the 'verification' cycle. In the second case, the shift in entropy due to the number of contradictory relationships (i) remaining after verification may be used to form a measure of disbelief. Expressed differently, the measure of disbelief in a hypothesis is related to the complexity measure of the hyperdigraph structure required to complete the partial structure within the database, after the verifictaion process.

$$Disbelief = \frac{\delta S'}{S_{min}} \qquad (22)$$

However, the imposition of a model upon the database is not really in accordance with the philosophy underlying the work descibed in this paper. Rather than *impose* structure upon information, we have defined a set of transformations which make explicit, implicit class hierachies within that information and defined a new method of quantifying their importance.

The system currently under consideration builds class structures in the manner of the DMP algorithm but uses work described in the previous two sections to build up measures of belief and disbelief in the emergent structures together with the verification mechanism for consolidating belief in that structure. This has advantage of combining a 'data driven' and 'goal driven' mechanism however the goals are suggested by the data itself, rather than imposed by an external source.

13 Conclusions

A conceptual clustering algorithm (called the DMP) capable of formulating relational descriptions at many levels of abstraction has been presented.

A method of quantifying this simplification has been derived and used to assess the Importance of the class with respect to the rest of the database. Simply, the more important the conceptual class, the more it orders the database.

Results based on the analysis of databases containing data from real images has been analysed. Results indicate that, when thousands of items are considered, the structures that are formed and designated as 'important' correspond to those classes that a human observed might call "significant", having an underlying physical cause. These results are independent of the any ordering of the database. They pose the possibility that Image Analysis might be achieved by massive applications (most probably parallel) of a set of simple transformations rather that one application of a complex algorithm.

A verification mechanism, based upon the tendency to complete sets, is suggested as the basis of a top-down directed automatic mechanism for supplying missing information to the database. A combined 'data-driven' and 'goal-driven' algorithm, whereby the goal is suggested by the data is seen as the future direction for this work.

The work is developed in the context of a database containing only certain information, as modelled by un-weighted structures, and it is evident that this work may be extended by a consideration of uncertain information as modelled by weighted relation hyper-digraph structures.

14 Acknowledgements

The work was conducted within the MMI 007 Alvey consortium. I would like to thank Dave Nuttall for many valuable conversations and much enthusiasm. I would also like to thank SRC Bristol for the provision of image derived databases.

References

[1] Bundy A. Incidence calculus: a mechanism for probabilistic reasoning. *Department of Artificial Intelligence, Edinburgh University*, Research paper no. 216 Edinburgh:, 1984.

[2] AVC87. *Proceedings of the Third Alvey Vision Conference. University of Cambridge*, The Alvey Vision Club Committee, September 1987.

[3] Claude Berge. *Graphs and Hyper-Graphs.* North-Holland Mathematical Library, University of Paris, 1979.

[4] Forgy L. Charles. *Rete: A Fast Algorithm for the Many Pattern/Many Object Match Problem. Artificial Intelligence 19 (1982) 17-37*, North-Holland, 1982.

[5] Langley P. Machine learning and concept formation. *Machine Learning volume2, number 2*, September 1987.