Robotics Research
Technical Report

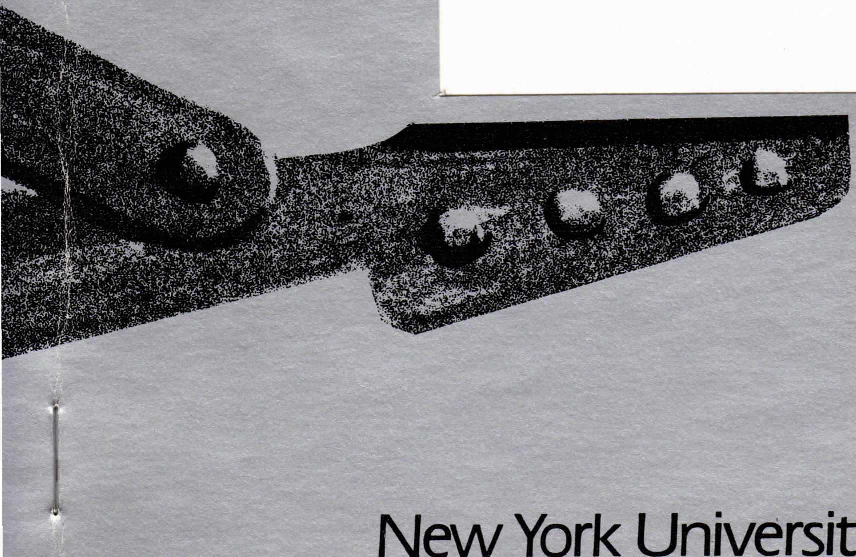
A Brief Survey of Knowledge Aggregation Methods

by

Michael S. Landy †

Robert A. Hummel ‡

Technical Report No. 177
Robotics Report No. 51
September 1985



New York University
Courant Institute of Mathematical Sciences

Computer Science Division
251 Mercer Street New York, N.Y. 10012

A Brief Survey of Knowledge Aggregation Methods

by

Michael S. Landy †
Psychology Department, New York University

Robert A. Hummel ‡
Courant Institute of Mathematical Sciences

Technical Report No.
Robotics Report No.
September, 1985

†Psychology Department, NYU
6 Washington Place, Room 961
New York, NY 10003

‡New York University
251 Mercer Street
New York, NY 10012

The work described in this paper was supported in part by a grant from the Office of Naval Research, Grant N00014-85-K-0077, and by National Science Foundation Grant DCR-8403300. We wish to acknowledge the helpful comments of Jean-Claude Falmagne.

Appears in: Proceedings of the International Conference on Pattern Recognition, Paris, France, Oct., 1986, pp. 248-252

A Brief Survey of Knowledge Aggregation Methods

Michael Landy
Robert Hummel

Abstract

This paper discusses several iterative knowledge aggregation methods. Such methods are used to choose one of a finite set of labels about each of a set of objects. First, a stimulus is analyzed locally at each object, yielding an initial state which assigns a weight of the evidence from that analysis to each of the labels. The methods continue as a sequence of trials. On each trial two events occur. First, new evidence is gathered either externally or using internal compatibility constraints on the current state. Second, the current state and new evidence are combined, resulting in a new state, which becomes the current state for the next trial. This method iterates until sufficient confidence in a single label at each object is achieved. In this paper, several such methods are reviewed and compared in terms of the form of the state space, the type of evidence which can be represented, and the efficiency and convergence properties of the methods as a whole.

1. Introduction

In this paper, methods are discussed which gradually accumulate evidence concerning a finite collection of objects. This is a particularly difficult problem which has arisen in areas as diverse as computational models of vision, expert systems, neural modeling, and statistical decision theory. Several methods have been defined in the recent past to contend with this problem. A survey of some of these methods is given here, set in a common framework and notation in order to permit easy comparisons.

Each method is an iterative procedure for knowledge aggregation, consisting of a sequence of *trials*. On each trial, two events take place: 1) the establishment of the current *state* of the procedure, and 2) the gathering of *evidence*. These two events are combined using an *update rule*, resulting in a new state for the next trial. The current state consists of a momentary estimate of the relative aggregated evidence for the identity of each of a finite collection of *objects*. Evidence is gathered either from internal constraints on the state, or from outside the process (from the ''environment''). The sequence of trials continues until either the evidence is exhausted or a stopping criterion is reached, hence the name: iterative knowledge aggregation. The entire procedure outline is shown in Fig. 1.

A brief example is illustrated in Fig. 2. This portrays an application of relaxation labeling (Rosenfeld, Hummel, and Zucker, 1976) to a constraint satisfaction problem in computational vision. A scene is given with four regions which have already been segmented. The relaxation procedure is intended to infer the real-world object associated with each region. There are four *objects* to be labeled (in the terminology of relaxation labeling), or four separate inferences to be made. For each object, there is a finite set of possible inferences or *labels* from which to choose. In our toy example, these labels indicate whether the given segment is a house (H), grass (G), trees (T), or sky (S).

The state of the procedure is given by specifying the current weights of each label at each object. In relaxation labeling, at any given object these weights essentially define a probability distribution across the set of potential labels (i.e. they are non-negative and sum to one). The

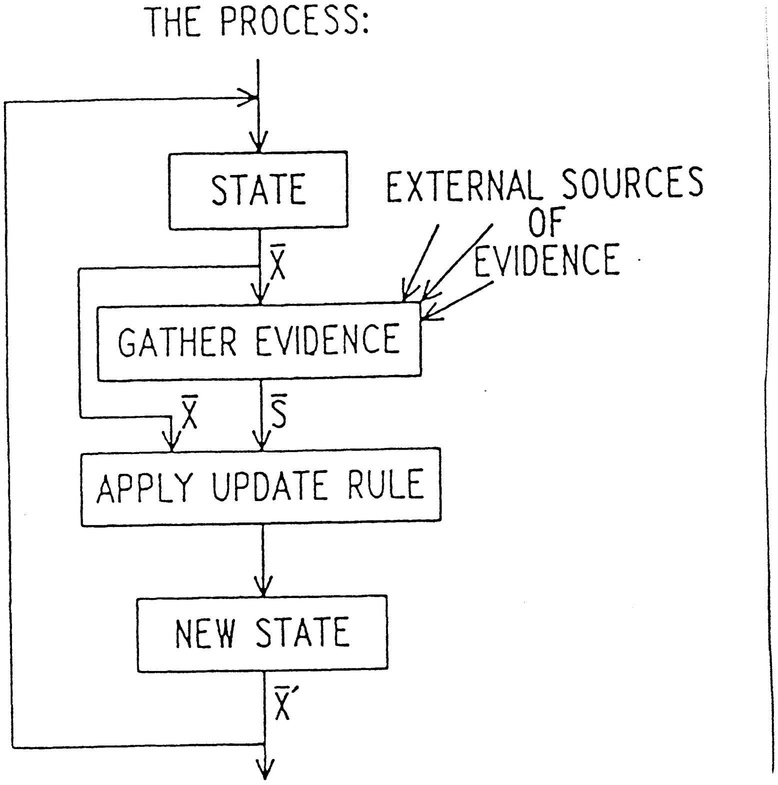
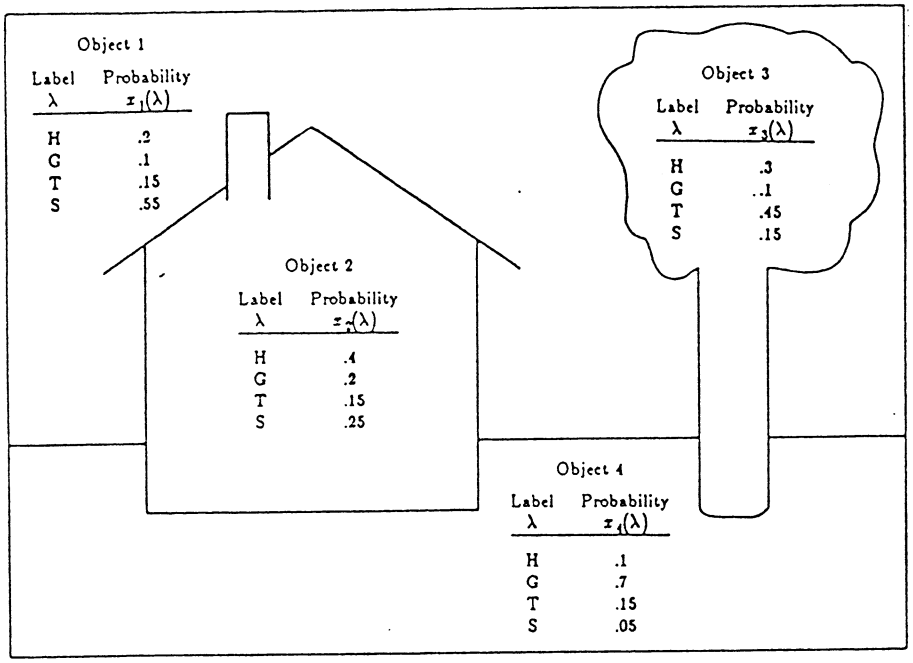


Figure 1. General outline of iterative knowledge aggregation methods.



Object 1

Label λ	Probability $x_1(\lambda)$
H	.2
G	.1
T	.15
S	.55

Object 2

Label λ	Probability $x_2(\lambda)$
H	.4
G	.2
T	.15
S	.25

Object 3

Label λ	Probability $x_3(\lambda)$
H	.3
G	..1
T	.45
S	.15

Object 4

Label λ	Probability $x_4(\lambda)$
H	.1
G	.7
T	.15
S	.05

Figure 2. Illustration of a labeling problem. There are four *objects*, corresponding to the four image segments. Each object can have one of four possible *labels*. In this example, the possible labels are *H*ouse, *G*rass, *T*rees, or *S*ky. The state of the procedure is defined by a probability distribution over the possible labels at each object. In the figure, at the current time we are 55% sure that Object 1 is the Sky. For any iterative knowledge aggregation method, the intent is that the probabilities converge to 1 for the correct labels at each object.

initial state is given to the procedure, and presumably is determined by image properties such as color, shape, etc.

The new evidence is computed using a set of constraints between labels at various objects. For example, in Fig. 2 it might be the case that a label of S for region 1 supports a label of G for region 4 because region 1 is above 4 and sky tends to appear above grass in natural images (again, this is only intended as a toy example). Similar constraints are computed between all neighboring objects and labels and result in a number for each label at each object indicating the degree to which other labels and objects support (or are consistent with) this particular label at this object. This set of support values is the current evidence.

Finally, the current state and evidence are combined, resulting in a new state. This new state is again a probability distribution over the set of labels at each object. After a number of iterations of the procedure, the intent is that the process will converge to a state with only one label at each object having a high probability.

2. Basic concepts

We consider a *stimulus*, consisting in a collection of n unknown *objects*, numbered $1,2,\ldots,i,\ldots,n$. For each object i, there is a finite set Λ_i of *labels*. It is assumed that over all applications of this method, Λ_i will always contain one, and only one label which describes object i correctly. Without loss of generality, we assume that for all objects i, Λ_i consists of m

possible labels, numbered $1,2,\ldots,m$. A particular label in Λ_i will be denoted λ_i. On a given trial, the current information concerning object i consists of a function x_i: $\Lambda_i \rightarrow \mathbf{R}$, which gives the relative weights of previous evidence for the labels in set Λ_i. The state of the procedure is a vector $\bar{x}=(x_1,\ldots,x_n)$ of such functions, one for each object, and summarizes the weight of all previous evidence for each label at each object.

On each trial, the new evidence is summarized as a *support vector* $\bar{s}=(s_1,\ldots,s_n)$. For each object i, s_i is a function s_i: $\Lambda_i \rightarrow \mathbf{R}$. $s_i(\lambda_i)$ denotes the weight of the new evidence for label λ_i at object i. $\bar{s}$ is similar to $\bar{x}$ in form, although different aggregation methods put different constraints on these two vectors. For example, in relaxation labeling the function $\lambda_i \rightarrow x_i(\lambda_i)$ is a probability distribution over Λ_i. Support values are not necessarily constrained in this manner. Indeed, in some relaxation labeling methods a support value $s_i(\lambda_i)$ can be negative, indicating that the weight of the new evidence is *against* label λ_i at object i.

At the end of each trial, the current state and evidence are combined in order to achieve a new state. Since the procedures used vary widely, at this point we simply indicate the update rule as $F(\bar{x},\bar{s})=\bar{x}'$. The new state $\bar{x}'$ becomes the current state for the subsequent trial of the procedure.

In the next section, four methods will be outlined which iteratively combine evidence in order to achieve a labeling. The methods include relaxation labeling (Rosenfeld et al., 1976; Hummel and Zucker, 1983), stochastic relaxation (Geman and Geman, 1984; Ackley, Hinton, and Sejnowski, 1985), stochastic assessment of knowledge (Falmagne and Doignon, 1985), and theory of evidence (Dempster, 1967 and 1968; Shafer, 1976). In the subsequent section, these methods will be compared with regards to state space, evidence gathering, and update rules.

3. Methods

3.1. Relaxation labeling processes

Relaxation labeling processes (RLP), which were introduced above, are used in constraint satisfaction problems. The initial state is generally all that survives of the external evidence given to the procedure. Subsequent evidence gathering is all internal to the procedure. It involves a measure of the internal consistency of the object labelings. Groups of objects and labels which are given credence initially and are mutually consistent should survive the process, whereas inconsistent labelings are given less and less weight.

RLP is generally intended to operate in a situation in which there is a large number of objects to be labeled. For example, in computational models of vision, an image will be described as a large number of image regions (e.g. pixels), each of which acts as an object to be labeled (is it an edge? is it foreground or background? etc.). Consistency constraints act between neighboring objects, where the neighborhood structure is defined by the problem at hand. In image computations, the neighbors of a given image region will generally be nearby regions.

Given a set of objects, the neighborhood is defined by a neighbor relation N. If iNj, then j is a neighbor of i, and labels at object j constrain the choice of labels at object i. Given two objects such that iNj, a set of compatibility coefficients is defined for the labels at the two objects, c_{ij}: $\Lambda_i \times \Lambda_j \rightarrow \mathbf{R}$. The value $c_{ij}(\lambda_i,\lambda_j)$ is a number which indicates the degree to which label λ_j at object j is compatible with (and thereby supports) label λ_i at object i. Given a large $c_{ij}(\lambda_i,\lambda_j)$ and a high previous weight for label λ_j at object j (i.e. $x_j(\lambda_j)$ is large), this combination is treated as evidence for label λ_i at object i. Since x_i is a probability distribution over Λ_i, $c_{ij}(\lambda_i,\lambda_j)$ might be derived as a correlation coefficient between the two labels at the two objects (Rosenfeld, Hummel, and Zucker, 1976), or using conditional probabilities (Rosenfeld et al., 1976; Faugeras and Berthod, 1981).

To effect this treatment of compatibility, we define the support vector

$$s_i(\lambda_i) = \sum_j \sum_{\lambda \in \Lambda_j} c_{ij}(\lambda_i,\lambda)x_j(\lambda). \tag{1}$$

The support for each object is thus a linear sum of the compatibility coefficients of neighboring objects' labels weighted by the current probability value for those labels (recall that in RLP the state values at a given object may be viewed as a probability distribution).

There have been a number of update rules suggested for RLP, and this has been one of the main points of contention. The original paper by Rosenfeld et al. (1976) suggests:

$$x_i'(\lambda_i) = \frac{x_i(\lambda_i)[1 + s_i(\lambda_i)]}{\sum\limits_{\lambda \in \Lambda_i} x_i(\lambda)[1 + s_i(\lambda)]} \tag{2}$$

A very similar expression has been used for synaptic modification in neural models of learning (e.g. Landy, 1981). The numerator of this expression causes labels with large support s to increase relative to those with smaller s, as we have indicated. The denominator ensures that the values of the state at a given object still sum to one after the update (see Fig. 3-a). The intent, as described in the example above, is that the state values at a given object be nonnegative and sum to one. Reviewing the above update rule, we see that the first of these conditions will *not* be assured if any of the support values $s_i(\lambda_i) < -1$. This clearly imposes a (perhaps undesirable) condition on the compatibility coefficients. In Rosenfeld et al. (1976), the $c_{ij}(\lambda_i,\lambda_j)$ coefficients are constrained so that $s_i(\lambda_i)$ always falls in the range $[-1,1]$.

A second approach, depicted in Fig. 3-b, is taken by Peleg (1980). In this rule, the compatibility coefficients are all positive. Support is redefined as

$$s_i(\lambda_i) = \prod_j \left[\sum_{\lambda \in \Lambda_j} c_{ij}(\lambda_i,\lambda)x_j(\lambda) \right]. \tag{3}$$

The update rule is then

$$x_i'(\lambda_i) = \frac{x_i(\lambda_i)\, s_i(\lambda_i)}{\sum\limits_{\lambda \in \Lambda_i} x_i(\lambda)\, s_i(\lambda)}. \tag{4}$$

Providing the c_{ij}'s are all nonnegative, this rule circumvents the problem of negative state values. If the compatibility coefficients are defined as

$$c_{ij}(\lambda_i,\lambda_j) = \frac{\mathrm{Prob}(O_i = \lambda_i \mid O_j = \lambda_j)}{\mathrm{Prob}(O_i = \lambda_i)}, \tag{5}$$

where O_i is the random variable giving the correct label for object i in a randomly chosen labeling problem to which the method might be applied, Peleg (1980) has shown that the support formula and update rule given in Eqns. 3 and 4 is a Bayesian estimation procedure.

Finally, Hummel and Zucker (1983), returning to the original support function definition (Eqn. 1), deal with the problem of maintaining state values which are probability distributions at each object by enforcing that condition with the update rule. Briefly, at a given object, the state may be regarded as a point in an m-dimensional simplex over the label weights at that object. The support of those labels may then be regarded as a vector based at that point indicating the relative increase or decrease of each state element that is desired (see Fig. 3-c). Since this vector may point off of the simplex, Hummel and Zucker suggest an update along the projection of that vector onto the simplex. They prove that this is equivalent to taking an update:

$$x_i'(\lambda_i) = x_i(\lambda_i) + \varepsilon u_i(\lambda_i) \tag{6}$$

where $\bar{u}$ is a vector which maximizes $\bar{u} \cdot \bar{s}$ subject to the constraints that $\bar{u} \le 1$ and there exists a $\bar{v}$ in the simplex such that $\bar{u} = \bar{v} - \bar{x}$. In Mohammed, Hummel, and Zucker (1983) an algorithm is provided which computes this vector.

The advantage of the Hummel and Zucker update rule is mathematical tractability. Despite its unobvious form, viewing this in the vocabulary of dynamical systems (Bhatia and Szego, 1967), they were able to prove a local convergence property for this method. This kind of formal result had been sorely lacking for the other methods.

3.2. Stochastic relaxation

Recently there has been a spate of papers concerning a method known as stochastic relaxation (e.g. Geman and Geman, 1985; Ackley, Hinton, and Sejnowski, 1985). This method uses a subset of the state space used by RLP, where only *unambiguous labelings* are allowed, those for which one label at each object has a weight of one, and all others are zero (i.e. a label is *chosen* at each object). This choice of a state space consisting of the corners of the simplex is shared with the predecessor of RLP, discrete relaxation (Haralick and Shapiro, 1979). The support vector computation is identical to that used by Rosenfeld et al. (1976) for RLP described above, although the equation simplifies since only a single label at a given object has a nonzero (in fact, unity) weight.

Where stochastic relaxation methods differ from RLP is in the update rule. Given that only a single label is chosen for each object, the update rule needs to specify how transitions to other labels occur. In stochastic relaxation methods, the label transitions occur as a Markov process, and are thus chosen stochastically. The transition probabilities are determined by the current

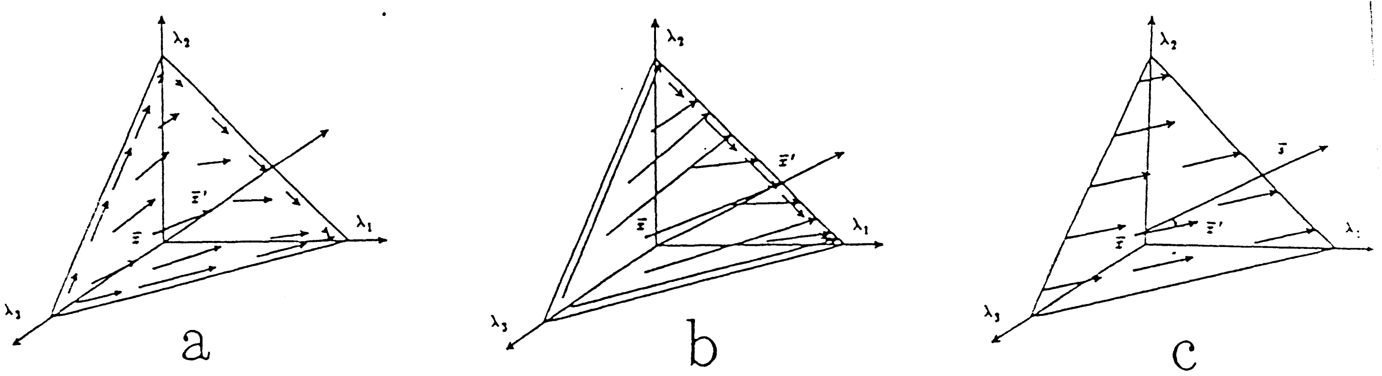


Figure 3. Three update rules for relaxation labeling. There is only one object illustrated, and three labels. The state space consists of the single simplex shown. In each case, the state change is shown given a support vector of (2,1,0). The state change denoted with filled circles begins with $\bar{x} = (.25,.25,.5)$, and shows the new state $\bar{x}'$. The other vectors show the state change given the same support vector and varying the initial state. a) The original rule from Rosenfeld, Hummel, and Zucker (1976). The vector from the origin is $(x_1(1+s_1), x_2(1+s_2), x_3(1+s_3))$, which is renormalized to the simplex to yield $\bar{x}'$. b) The rule suggested by Peleg (1980). The vector from the origin is $(x_1 s_1, x_2 s_2, x_3 s_3)$, which is renormalized to the simplex to yield $\bar{x}'$. c) The projection operator of Mohammed, Hummel, and Zucker (1983). The support vector $\bar{s}$, based at the current state $\bar{x}$ is projected onto the simplex, and a step of fixed size in that direction is taken (here, of length about .28), making sure that such a step remains within the simplex boundaries.

state and support vectors.

The particular stochastic method described by Geman and Geman (1985) reflects the theory of Ising spin glass models in physics. It has been applied to cases in which there is only one object, so we shall drop the subscript denoting the object in the following. Given a support function s, the probability of a transition to label λ is given by

$$P(\lambda) = \frac{e^{s_i(\lambda)/T}}{\sum_{\lambda' \in \Lambda} e^{s_i(\lambda')/T}}. \tag{7}$$

The parameter T may be regarded as the temperature at which the process takes place. With a high value of T, the support values are virtually ignored, and the process percolates randomly among the various labels. As T decreases, the process at each step is increasingly likely to choose the label with the greatest support among those at each object. By running the process over a large number of iterations with T decreasing sufficiently slowly over those trials (called *simulated annealing*), the method can be proven to globally minimize an energy functional

$$E = -\sum_{\lambda<\lambda'} c(\lambda,\lambda')\,x(\lambda)\,x(\lambda') + \sum_{\lambda} \theta_\lambda\, x(\lambda), \tag{8}$$

with a probability approaching one as the trial number increases. The energy indicates the degree of inconsistency in the state. The θ_λ are bias terms which can be included in the support function.

Stochastic relaxation is mathematically tractable, and can be proven to converge to a globally consistent state. On the other hand, at present there are problems with the method. First, the convergence results are only available if the compatibility coefficients are symmetric (i.e $c_{ij}(\lambda_i,\lambda_j) = c_{ji}(\lambda_j,\lambda_i)$). Second, the convergence results are based on a particular schedule for the decrease in T which makes the method intolerably slow, and we have some evidence that the method is not at all robust over accelerations in the process.

3.3. Stochastic knowledge assessment

A very different area of research, that of computer-assisted instruction (CAI), has led to a strikingly similar method to RLP. In the research of Falmagne and his colleagues (Falmagne and Doignon, 1985), the problem is to efficiently assess the extent of a student's knowledge in a restricted learning domain. The intent is to quickly refine an estimation of the student's knowledge by a few well-placed quiz questions. The results of this assessment are then to be used to guide the subsequent teaching process. The method is made efficient and robust by carefully choosing the question to ask based on the already gained estimation of the student's knowledge, and by carefully updating that estimation given the student's response. Although intended as an aid to CAI systems, the assessment method certainly has applications to other areas in which "expert systems" are applied, such as medical diagnostic systems (Gorry, Kassirer, Essig, and Schwartz, 1973.).

There are a number of ways of describing stochastic knowledge assessment as an iterative knowledge aggregation method. The following is intended to highlight the similarities between it and RLP. The state space is identical to that of RLP. There is one object, the student being assessed, and therefore the subscript denoting the object will be dropped. The set of potential labels for that student is a pre-specified set of possible states of knowledge. The definition of a state of knowledge is based on the idea that a given educational area can be broken down into a finite set Q of basic notions, each of which can be associated with a *question* (or equivalence class of questions) which tests that notion. A single state of knowledge λ of a student consists of the set of questions that the student is capable of correctly answering ($\lambda \subseteq Q$ or, alternatively, $\lambda \in 2^Q$). The set of potential states (the label set) is a distinguished set $\Lambda \subseteq 2^Q$. The only constraint placed on Λ is that it be closed under union. This is based on the intuition that if you had two students in states λ_1 and λ_2, it would be possible if they talked to each other long enough for one of them to attain state $\lambda_1 \cup \lambda_2$.

The stochastic knowledge assessment procedure operates within the state space defined, as in RLP, as the set of probability distributions across the labels Λ at each object (here, the student). Evidence is gathered by choosing a question, asking it of the student, and getting the student's response. The method used to choose the questions is intended to gain the most information possible at each trial. This part of the procedure is beyond the scope of this paper (see Falmagne and Doignon, 1985). Given the question-answer pair, an update rule is used to generate a new state.

In order to compute the support, first the response to the question $q \in Q$ is graded:

$$r(q) = \begin{cases} 1 \text{ if the response was correct} \\ 0 \text{ if the response was incorrect} \end{cases} \tag{9}$$

A correct response supports all states which predict a correct response. Thus, $r(q) = 1$ supports all states in the set

$$\Lambda_q = \left\{ \lambda \in \Lambda \,|\, q \in \lambda \right\}. \tag{10}$$

Similarly, an incorrect response $r(q) = 0$ supports all states in the set

$$\Lambda_{\bar{q}} = \left\{ \lambda \in \Lambda \,|\, q \notin \lambda \right\}. \tag{11}$$

All of this results in a support vector

$$\bar{s}(\lambda) = r(q)\, \iota_\lambda(q) + (1 - r(q))\,(1 - \iota_\lambda(q)), \tag{12}$$

where ι is the indicator function

$$\iota_\lambda(q) = \begin{cases} 1 \text{ if } q \in \lambda \\ 0 \text{ if } q \notin \lambda \end{cases} \tag{13}$$

In Falmagne and Doignon (1985) two update rules are discussed, which are described here in simplified form. The first rule is referred to as the *affine assessment procedure.* Given the previous state $\bar{x}$, the question that was asked q, and the response r, the new state $\bar{x}'$ is

$$\bar{x}' = (1 - \theta)\bar{x} + \theta\, \bar{g}(q, r, \bar{x}) \tag{14}$$

where

$$\bar{g}(q,r,\bar{x})(\lambda) = \frac{\bar{s}(\lambda)\,\bar{x}(\lambda)}{\sum_{\lambda' \in \Lambda} \bar{s}(\lambda')\,\bar{x}(\lambda')} \tag{15}$$

and $0 \le \theta \le 1$.

As illustrated in Fig. 4-a, this rule may be regarded as computing a convex combination between two points on the simplex -- the current state $\bar{x}$ and the goal state $\bar{g}$. The goal state is that state which treats the current evidence as infallibly counter to labels which the evidence refutes, and otherwise uninformative between those that it supports. In the context of knowledge assessment this is reasonable since the support gained from one student response only works against those labels inconsistent with the response, but in no way differentiates between labels consistent with it. The goal state retains the relative strengths of the labels consistent with the student response, setting all other labels to zero (and renormalizing to keep within the simplex). The convex assessment rule changes the state along the line segment from $\bar{x}$ to $\bar{g}$. Parameter θ determines how large a step along that segment is taken.

The second update rule is termed a *multiplicative assessment procedure.* In this procedure, the support vector is

$$\bar{s}(\lambda) = \begin{cases} 1 & \text{if } \iota_\lambda(q) \neq r \\ \theta & \text{if } \iota_\lambda(q) = r \end{cases} , \tag{16}$$

where $\theta > 1$. Given this new support vector, the subsequent state is

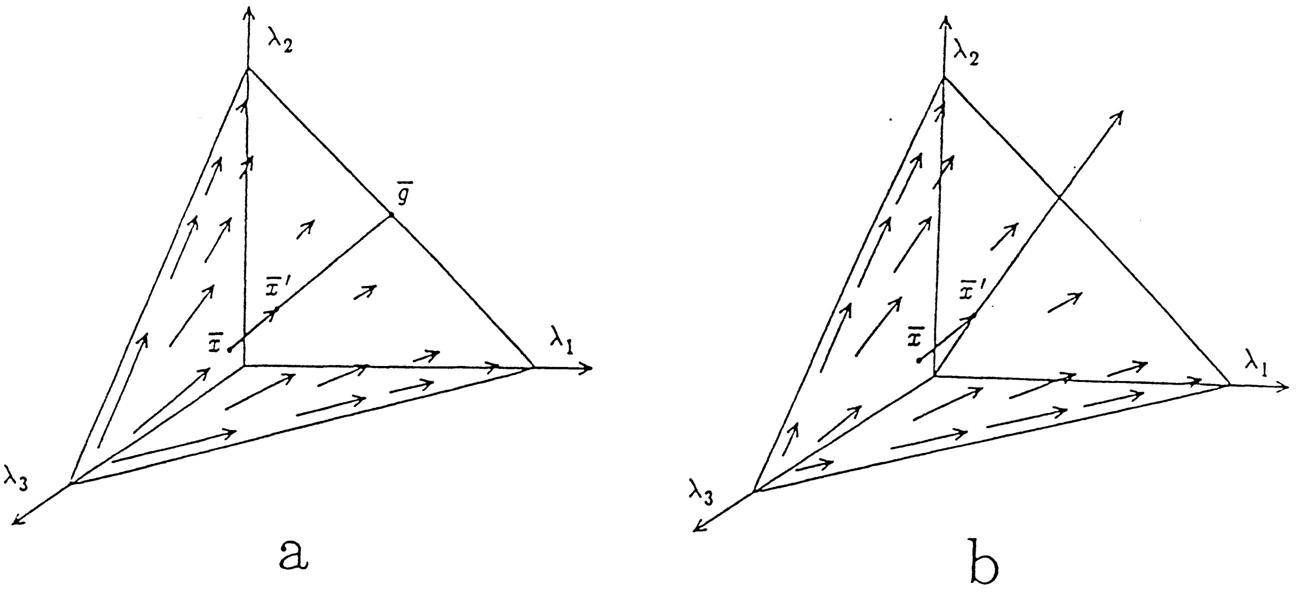


Figure 4. Two update rules for the stochastic assessment of knowledge. There is only one object, the student, and three labels. The state space consists of the single simplex shown. In each case, the state change is shown given a support vector based on a student response which supported the knowledge states represented by labels λ_1 and λ_2, but not λ_3. The state change denoted with filled circles begins with $\bar{x} = (.25,.25,.5)$, and shows the new state $\bar{x}'$. The other vectors show the state change given the same support vector and varying the initial state. a) The convex assessment procedure. Given the state $\bar{x}$ and support $\bar{s} = (1,1,0)$, a goal state $\bar{g}$ is defined which has all labels set to zero which aren't supported, and maintains the relative standing of the other labels. The update is found by taking a step along the line segment from $\bar{x}$ to $\bar{g}$ (30% of the way, as depicted here). b) The multiplicative assessment procedure, which is identical to the relaxation labeling rule suggested by Peleg (1980). The support vector for this rule (with parameter $\theta = 1$ here), is $\bar{s} = (2,2,1)$. The vector from the origin is $(x_1 s_1, x_2 s_2, x_3 s_3)$, which is renormalized to the simplex to yield $\bar{x}'$.

$$\bar{x}'(\lambda) = \frac{\bar{s}(\lambda)\bar{x}(\lambda)}{\sum\limits_{\lambda' \in \Lambda} \bar{s}(\lambda')\bar{x}(\lambda')}. \tag{17}$$

This rule is identical to the Peleg rule (Peleg, 1980) for relaxation labeling, although the derivation of the support vector is of necessity a bit different. The method is illustrated in Fig. 4-b.

3.4. Theory of evidence

The theory of evidence, as first described by Dempster (1967, 1968), and later expounded by Shafer (1976), is a method for combining evidence from disparate sources. It has also been suggested as a relaxation labeling method by Faugeras (1982). It has gained some popularity recently as a tool for expert systems (c.f. Gordon and Shortliffe, 1984a and 1984b, Barnett, 1981; Friedman, 1981, Gouvernet, et al., 1980; Strat, 1984). The most attractive feature of this theory is that it makes it possible to handle evidence for or against subsets of the set of possible inferences in addition to evidence relevant to single inferences.

In the terms of knowledge aggregation, there is again only a single object to be labeled. Given a mutually exclusive and exhaustive set of possible inferences, I, to be made about that object, theory of evidence uses as its label set $\Lambda = 2^I - \{\emptyset\}$. It is this greatly expanded label set that allows theory of evidence to describe evidence impinging on a set of possible inferences in addition to single inferences. The state space is the space of all probability distributions over Λ.

The interpretation of this state space is as follows. If all of the probability is on the set containing a single inference i, then that indicates 100% probability that i is true. If all of the probability is on the set I, this indicates that we have no knowledge whatsoever to discern between the possible inferences. If the probability is split equally among several singleton subsets of I (say, $\{i\}$, $\{j\}$, and $\{k\}$), this indicates that there is equally strong but conflicting evidence for these three inferences. Finally, if all of the probability is concentrated on the set $\{i,j,k\}$, this indicates that we are 100% certain that one of the three inferences are true, but have no evidence to discern these three possibilities. This ability of the theory to distinguish between cases of no knowledge (the first example) and conflicting knowledge (the third example) is another attractive feature of the theory.

A single state x, as described thus far, corresponds to the *mass function* in the theory of evidence. Note that this function in the theory of evidence is used to form a second function, the belief function

$$Bel(A) = \sum_{B \subseteq A} x(B). \tag{18}$$

This states that the belief we have that the correct inference lies in set A is the sum of the masses of evidence that have been gathered for any subsets of A.

The theory of evidence has a number of interesting features, including discussions of refinements and coarsenings of the inference set (Shafer, 1976), properties of belief functions, and so on. Here, our only interest is to describe the combination of a previous mass function (or state, in our terminology), and new evidence (a support function, which is simply a new mass function over the same set I). This is the update rule in theory of evidence. Given two mass functions, a state x and a support function s over the same inference set I, one first defines m': $2^I \rightarrow \mathbb{R}$,

$$m'(A) = \sum_{\substack{B,C \\ B \cap C = A}} x(B) \cdot s(C). \tag{19}$$

Finally, the new mass function or state becomes

$$x'(A) = \frac{m'(A)}{1 - m'(\emptyset)} \quad \text{for } A \neq \emptyset \tag{20}$$

The first step says that evidence for two overlapping sets supports the intersection of those two sets to the degree that one trusts both sources of evidence (thus treating sources of evidence as independent). The second step is a renormalization required to keep mass off of the empty set (since one inference must be true). Note that this combination can not work with two totally incompatible sources of evidence (with mass entirely concentrated on mutually exclusive sets), since such a situation requires a division by zero in Equation (20).

The theory of evidence is yet another method which has a state space which is a simplex. Unfortunately, this does not make its update rule easy to describe in geometric terms as we have for the previous rules. First of all, the dimensionality of the space is not the number of possible inferences $|I|$, but rather $2^{|I|}$, a much larger figure. Second, the update rule takes strong evidence for one subset A (i.e. closeness to the corner of the simplex representing A) and strong support for subset B (a support vector pointing in the direction of corner B), and yields as an updated state a distribution with a moderately large weight for the set $A \cap B$, which is yet another corner entirely.

There is an alternative viewpoint for the theory of evidence, which we will briefly mention in the next Section, and is more fully described in a forthcoming note (Hummel and Landy, 1985). In this viewpoint, the state space is the set of unambiguous labelings on $2^{|I|}$, and the state is the current set of statistics on a stochastic variable which takes on values in the state space. Under this viewpoint, the state space is simpler, in that it is the space of unambiguous labelings rather than the full simplex, but the current state, however, is much more complex, and its measurement requires that a stochastic system be observed through many transitions. Also in this viewpoint, the evidence is gathered externally, and is presented in the form of another state in the same state space. These two states are combined, essentially by constructing a new stochastic function whose unambiguous labeling at any given instant is given by the intersection of the two subsets indicated at that instant by the two stochastic functions being combined.

4. Points of comparison

4.1. State space

The composition of the state space in a knowledge aggregation method determines the types of evidence that can be handled by the method. Consider, for the moment, a case in which there are only three possible labels, *i*, *j*, and *k*. Considering the possible state of evidence regarding these three labels, there are several cases of interest:

1) There is no evidence pro or con any label.
2) There is strong evidence for *i*, *j*, and *k*, despite the fact that they are mutually exclusive.
3) There is strong evidence against *i*, *j*, and *k*.
4) There is strong evidence for *i*, and no evidence one way or the other concerning *j* and *k*.
5) There is strong evidence against *i*, and no evidence one way or the other concerning *j* and *k*.
6) There is a strong piece of evidence for labels *i* and *j* which does not differentiate between them.

As Falmagne has pointed out (Falmagne, 1983), using a probability measure across the three labels can not differentiate between the first three cases, since each would lead to giving each label an equal weight of 1/3. It may be suggested that this is not a defect in the sense that in any reasonable method one would not react any differently in the three cases. But, this is not at all clear. In the first case, one might withhold one's "belief" while actively seeking new information regarding the three possibilities, while in the second, one may commit one's belief right away, split equally. Also, given a new piece of evidence, it is not clear that the update rule should achieve the same next state starting with each of the first 3 cases above, given new evidence impinging on, say, *i*.

Thus a state space which is restricted to a simplex in $\mathbb{R}^m$, where $m = |\Lambda|$, can not differentiate between the first three cases. A reasonable solution is to increase the dimensionality of the state space. In RLP, this is often dealt with by adding a "don't know" or null label, but this results in further ad hoc choices for compatibilities. One can allow a superadditive probability measure, thus using as the state space at a given object the hypercube $[0,1]^m$. But, given the possibility of evidence *for* or *against* a given label (cases 4 and 5 above), one can use instead the hypercube $[-1,1]^m$, with evidence pro a label giving a positive value, and evidence against a negative one. This is exactly the state space used in Anderson's associative memory model (Anderson, 1972, 1973, and 1977), which is another iterative knowledge aggregation method. Finally, excluding the possibility of perfect evidence (values of 1 or -1), the resulting open hypercube may be mapped into $\mathbb{R}^m$, treating the state of the evidence as simply a vector in $\mathbb{R}^m$. This corresponds to the "weight of the evidence" in theory of evidence (Shafer, 1976). (The weight of evidence for a label λ, which is a subset of I, is

$$w(\lambda) = -\log(1-x(\lambda)),$$

where $x(\lambda)$ is the mass on λ.

Case 6 above requires an even larger state space. If we are to permit evidence to refer not only to single labels but also sets of labels, then the evidence summary will have to be over the power set of the label set. This is exactly the step taken by the theory of evidence (Shafer, 1976). Evidence against a subset A may be equally well viewed as evidence for $\bar{A}$, so the evidence weights become non-negative quantities, and the state space is either $\mathbb{R}^{2^m,+}$ — the positive quadrant of $\mathbb{R}^{2^m}$ — (for evidence weights) or $[0,1]^{2^m}$ (for mass functions as described in Section 2.4, where the mass $s(\lambda) = 1 - e^{-w(\lambda)}$, for evidence weight $w(\lambda)$).

4.2. Evidence

The computation of evidence is not the most critical area for comparing the methods discussed above. Relaxation labeling and stochastic relaxation are methods intended to gather evidence using compatibility constraints which are internal to the state. On the other end of the spectrum, the theory of evidence is intended to combine externally gained new evidence with the current state, which was based on aggregating previous evidence. The stochastic knowledge assessment procedure falls somewhere in between. It uses the current state to guide the acquisition of new external evidence by using the current state to compute the most effective question to ask of the student.

It is certainly possible to consider other combinations of evidence computation and update rule. For relaxation labeling, this sort of extension has been considered before. Three possible update rules have been described above. Also, the computation of evidence need not be restricted to a linear sum of pairwise compatibilities. Other authors have suggested nonlinear functionals for the support function and also triadic and possibly higher-order compatibility functions (Davis and Rosenfeld, 1981; Haralick et al., 1978; Haralick and Shapiro, 1979 and 1980; Hummel and Zucker, 1983; Faugeras and Berthod, 1981). The convergence result for stochastic knowledge assessment procedures (Falmagne and Doignon, 1985) applies to a far more general class of update rules than the two described in Section 2.3.

Relaxation labeling can in a sense be faulted for relying utterly on compatibility evidence after the initial state. In an image analysis application, image data are analyzed resulting in the initial state of the RLP process. In subsequent trials of the RLP process, the image data are never referred to again except to the extent that the data, as codified in the initial state, survive the RLP iterations. Given this, it is possible for an RLP process to converge on a state having nothing whatsoever in common with the image data, because the image data were severely inconsistent. In fact, a linear update rule considered by Rosenfeld et al. (1976) always converged to the same state regardless of the choice of initial state. This lead to the nonlinear update rule of Eqn. 2, which at least *potentially* is affected by the choice of initial state.

A perfectly reasonable approach to this problem would be to allow the image data to continue to impact the support function computation, by computing support as a combination of internal (state compatibility) and external (image data) evidence. In stochastic relaxation, maximization of a posteriori likelihoods leads explicitly to such a formulation (Geman and Geman, 1984). Similarly, evidence in the theory of evidence could certainly be extended to include internal compatibility information among multiple objects, if that were relevant to the particular task.

4.3. Update rule

The choice of update rule in any iterative knowledge aggregation method is affected by two factors, the form of the evidence and state spaces, and considerations of efficiency. The first consideration is meant to reflect common sense considerations of how the state should be changed given particular types of evidence. In all of the methods described, as one might expect, evidence for a particular label causes its state value to rise, and evidence against it causes its state value to fall. But, in the case of the theory of evidence, the combination rule can lead to some paradoxical behavior. Consider a simple example. Assume there are only three possible inferences *i*, *j*, and *k*. Furthermore, assume that the previous evidence satisfies $x(\{i\}) = .9$ and $x(\{j\}) = .1$, and the new evidence satisfies $s(\{k\}) = .9$ and $s(\{j\}) = .1$. The update rule will result in $x'(\{j\}) = 1$. Thus, conflicting strong evidence can cause labels with low probability values to achieve very strong support because of the renormalization in Eqn. 20. Whether or not such behavior is desired can guide the choice of an update rule.

In order to justify the form of a particular update rule, in several cases authors have tried to recast the methods in familiar Bayesian terms. As we have already mentioned, the Peleg rule (1980) for RLP was defined based on Bayesian arguments, where the compatibilities are based upon conditional probabilities.

Several authors have recast the theory of evidence in Bayesian terms. Ignoring the renormalization part of the rule, a particularly simple version can be accomplished by defining a random vector with jointly distributed random variables as components (Landy and Hummel, 1985). The entire update rule including the renormalization can also be described using a set of jointly distributed random variables (Falmagne, 1983). Shafer (1981) also provides a model which generates the combination rule in a situation where one is delivered a message about which inference is true but which is made using a randomly chosen code.

The efficiency of an update rule is also a consideration. There are two forms of efficiency that are relevant: the conditions required for convergence and the number of iteration steps required to get reasonably close, and whether the method is computationally efficient and feasible. Convergence results have been proven for various RLP update rules (Rosenfeld et al., 1976; Faugeras and Berthod, 1981; Hummel and Zucker, 1983), stochastic relaxation (Geman and Geman, 1984), and for a very general class of stochastic knowledge assessment procedures (Falmagne and Doignon, 1985). In the case of stochastic relaxation, the convergence result requires symmetric compatibility coefficients. For stochastic assessment of knowledge, the convergence result places certain conditions upon both the rule used for question choice and the update rule.

The number of iteration steps required to approach convergence is an important consideration. To the extent that these methods might be considered as models of human information processing (e.g. Ackley et al., 1985), this is an important consideration. Given that the human nervous system is capable of reaction times as fast as, say, 300-500 ms., and produces these responses with neurons which fire on the order of once ever 10 ms., there is not much time for computation. Despite the massive parallelism available with relaxation methods, if they require a large number of iterations, they are unsuitable for describing human behavior. Several authors have pointed out that stochastic relaxation is extremely slow (e.g. Ballard, 1985), and recent research suggests that any attempt to speed up the method destroys its convergence properties. On the other hand, RLP appears to be fairly robust across a variety of possible update rules, and the number of iterations required need not be large.

The computational efficiency considerations arise in particular with implementations of the theory of evidence. Given that theory of evidence requires a state space of size $2^{|\Omega|}$, the time and space requirements can be extreme. This has led several authors to consider restricted versions of the theory (Barnett, 1981; Gordon and Shortliffe, 1984b), but it is by no means clear that these alterations in the theory will exhibit appropriate behavior.

5. Conclusions

In this paper, several methods for the aggregation of knowledge have been reviewed. Each method has its own particular benefits and problems, related to the type of evidence and state that may be represented, the way in which new evidence is treated, and the efficiency. It is hoped that by providing a context in which these methods can be reasonably compared, researchers may be in a position to use this information to carefully choose the appropriate method for a given application.

References

Ackley, D. H., Hinton, G. E., and Sejnowski, T. J., A learning algorithm for Boltzmann machines, *Cognitive Science* **9**, 147-169, 1985.

Anderson, J. A., A simple neural network generating an interactive memory, *Mathematical Biosciences* **14**, 197-220, 1972.

Anderson, J. A., A theory for the recognition of items from short memorized lists, *Psychological Review* **80**, 417-438, 1973.

Anderson, J. A., Distinctive features, categorical perception, and probability learning: Some applications of a neural model, *Psychological Review* **84**, 413-451, 1977.

Ballard, D. H., Cortical connections and parallel processing: Structure and function, *The Behavioral and Brain Sciences*, in press, 1985.

Barnett, J. A., Computational methods for a mathematical theory of evidence, In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, Vancouver, B.C., 868-875, 1981.

Bhatia, N. P., and Szego, G. P., Dynamical Systems: Stability Theory and Applications (Lecture Notes in Mathematics, Vol. 35), New York: Springer-Verlag, 1967.

Davis, L. S., and Rosenfeld, A., Cooperating processes for low-level vision: A survey, *Artificial Intelligence* **17**, 245-263, 1981.

Dempster, A. P., Upper and lower probabilities induced by a multivalued mapping, *Annals of Mathematical Statistics* **38**, 325-339, 1967.

Dempster, A. P., A generalization of Bayesian inference, *Journal of the Royal Statistical Society, Series B* **30**, 205-247, 1968.

Falmagne, J.-C., A random utility model for a belief function, *Synthese* **57**, 35-48, 1983.

Falmagne, J.-C., and Doignon, J.-P., A class of stochastic procedures for the assessment of knowledge, manuscript, 1985.

Faugeras, O. D., Relaxation labeling and evidence gathering, Proceedings of the 6th International Conference on Pattern Recognition, Munich, Germany, Oct. 19-22, 1982, 405-412, Silver Springs, Maryland: IEEE Computer Society, 1982.

Faugeras, D. D., and Berthod, M., Improving consistency and reducing ambiguity in stochastic labeling: An optimization approach, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-3**, 412-424, 1981.

Friedman, L., Extended plausible inference, In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, Vancouver, B.C., 487-495, 1981.

Garvey, T. D., Lowrance, J. D., and Fischler, M. A., An inference technique for integrating knowledge from disparate sources, *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, Vancouver, B.C., 319-325, 1981.

Geman, S., and Geman, D., Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-6**, 721-741, 1984.

Gordon, J., and Shortliffe, E. H., The Dempster-Shafer theory of evidence, In Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project (B. G. Buchanan and E. H. Shortliffe, Eds.), Reading, Massachusetts: Addison-Wesley, 1984a.

Gordon, J., and Shortliffe, E. H., A method for managing evidential reasoning in a hierarchical hypothesis space, Stanford Computer Science Department Technical Report STAN-CS-84-1023, 1984b.

Gorry, G. A., Kassirer, J. P., Essig, A., and Schwartz, W. B., Decision analysis as the basis for