Decision Making in Context

ROBERT M. HARALICK, SENIOR MEMBER, IEEE

Abstract—**From a Bayesian decision theoretic framework, we show that the reason why the usual statistical approaches do not take context into account is because of the assumptions made on the joint prior probability function and because of the simplistic loss function chosen. We illustrate how the constraints sometimes employed by artificial intelligence researchers constitute a different kind of assumption on the joint prior probability function. We discuss a couple of loss functions which do take context into account and when combined with the joint prior probability constraint create a decision problem requiring a combinatorial state space search. We also give a theory for how probabilistic relaxation works from a Bayesian point of view.**

Index Terms—**Artificial intelligence, context, decision theory, pattern recognition, probabilistic relaxation, Viterbi.**

I. Introduction

The difference between the information that people use to solve recognition problems and the information that computers use to solve pattern recognition problems is clear: people make excellent use of the global organizational structure of the problem while the typical computer pattern recognition paradigms tend to work almost everything possible from only the immediate local structure. The difference is context.

It is the recognition of this difference—that more is to be gained by discovering suboptimal ways of handling context than by discovering optimal ways of handling local structure—which caused the artificial intelligence researchers to break with the paradigms enthusiastically endorsed by the pattern recognition researchers one decade ago. The artificial intelligence researchers had correctly assumed that even if they had the optimal techniques, parametric or nonparametric, as long as these techniques only paid attention to local structure, there was no hope in solving the difficult problem.

This paper reviews a few ways in which context has been handled in the literature and it introduces additional possibilities. It shows how state space search can be involved in an evaluation of an expected loss. It illustrates under what assumptions the cooperative processing and probabilistic relaxation algorithms can precisely minimize an expected loss. Although the paper is not comprehensive, does not bridge all the artificial intelligence recognition algorithms, and does not discuss the place of production rule systems, it is hoped that the essence of the description will encourage more researchers to solve more problems using context in a decision theoretic framework.

A. Pattern Recognition and Artificial Intelligence

There are many problems in pattern recognition and artificial intelligence which involve decision making. A pattern recognition algorithm often makes an evaluation of a conditional probability function and decides, typically observed sample by observed sample, that class or label having smallest expected loss for each observed sample. An artificial intelligence algorithm often postulates some kind of relational structure, production rule structure, or constraint structure among the observed samples and executes a search in an explosively large state space to determine that description most consistent with the measurements and the constraints. Thus, the emphasis in the pattern recognition research has centered around density functions and decision boundaries in a Euclidean space, while the emphasis in the artificial intelligence research has centered around understanding the constraints inherent in the reality we are trying to make decisions about and in the use of heuristics to speed up the combinatorial search required in making use of these constraints.

There are pattern recognition researchers who have used context, beginning with the dictionary methods of Bledsoe and Browning [2] and the neighborly dependence of Chow [4]. Toussaint [27] gave a survey of context techniques. Recently, there have been pattern recognition researchers such as Stockman *et al.* [24] and Kanal [13] who are using and advocating state space search in solving pattern recognition problems, and artificial intelligence researchers such as Feldman and Yakimovsky [7] and Lowerre and Reddy [15] who are using and advocating the use of decision theory in solving image understanding and speech understanding artificial intelligence problems.

B. Outline of Paper

Section II describes the general decision making problem as a labeling problem. There are units which are measured and these units must be assigned labels. In digital signal processing, the units are the instances of time at which a sample is observed. In image processing, the unit is the row column coordinates of a pixel position. The joint assignment of labels made by the recognition process depends upon the measurements of all the units and must minimize some specified expected loss.

Section III reviews many of the pattern recognition approaches in terms of the kinds of loss functions used and the conditional independence assumptions which make the computation of the labels minimizing the expected loss easy to compute. The salient characteristic of the usual approaches is that the loss functions have a decomposition as a simple sum or product of partial losses taken unit by unit, and the prior joint probability of unit labels have a correspondingly simple decomposition. With the simple decomposition classically employed there is nothing inherent in the form of the loss function or joint prior that makes related units have related labels. Hence, context is not taken into account.

Manuscript received April 5, 1982; revised June 4, 1982.

The author is with the Departments of Electrical Engineering and Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061.

Section IV reviews some of the artificial intelligence approaches from the point of view of decision theory. First, we discuss how the legal possibilities and constraints can be expressed by the equal probability of ignorance assumption as applied to the joint prior probability; either a joint labeling of all the units is legal and makes sense, or it does not make sense. All those labelings which are legal have the same nonzero prior probability. All those which are not legal have zero prior probability. Then expressions are given for the loss minimizing labels using this kind of joint prior and the usual loss functions. Solving for the minimizing labels here is clearly a combinational search problem.

Second, we discuss one form for the constraint label set specifying all the legal labelings. We give a representation for the set of legal labelings as a set of overlapping pieces or segments, each piece being a group of related units. Then we discuss a corresponding natural decomposition of the loss function which keeps related units and labels together, penalizing for sequences of labels that are not meaningful for sequences of related units.

Third, we discuss a simple class of loss functions which makes it possible to solve the difficult combinatorial problem by dividing it into small groups of related units and then easily combining the solutions of the segments into an entire approximate or suboptimal global solution.

Finally, Section V discusses the popular probabilistic relaxation or cooperative processing model used by a variety of researchers in computer vision and artificial intelligence. Here we illustrate that under a set of general conditional independence assumptions, not as strong as the ones usually discussed in textbooks, probabilistic relaxation is in fact just an algorithm for the assignment of labels which can use the entire context and minimize expected loss using the total error loss function.

II. Statement of Problem

Let the universe be divided up into recognizable and measurable pieces which we call units. Let U be such a set of units. Each unit in U can be characterized by 1) its relationships with other units, 2) an n-tuple of measurements determined by some local measuring process, and 3) the appropriate category interpretation for the unit. We call each category interpretation a label. We denote by L the set of possible category interpretations for a unit.

The Bayesian framework for decision making poses the following problem: given the measurement n-tuple made on each unit, and given the prior world knowledge Q which specifies allowable category interpretations for each group of related units, determine any functional assignment $f, f: U \rightarrow L$ which assigns an interpretation to each unit, satisfying that f has the least expected loss for a specified loss function.

It is important to recognize that this statement of the problem hides the reality that although it is the observed units which are measured and each assigned a label, the meaningful entities are objects which are collections of related units of the same type. Furthermore, these collections of related units are not easily specified ahead of time. In image processing the problem of determining who these groups of related units are is called segmentation. The point of view taken in this paper is that the collections of related units are discovered as a byproduct of the labeling process. After labeling, we just need to group together those connected units with the same label.

The decision making problem we have just described involves context because the assignment of each unit depends, in general, on the measurements of all the units. Usual approaches typically make complete independence assumptions or Markov assumptions which considerably reduce or simplify the general dependence of expected loss on all unit measurements. The typical loss functions also do not emphasize dependencies among related units. We will discuss such approaches in Section III and then suggest in Section IV one way to model more complex basic unit dependencies. We begin our discussion by describing the general Bayesian model in greater detail.

A. The Bayesian Model

Corresponding to a unit i, there is its assigned interpretation z_i, and its measurement n-tuple x_i. Given the n-tuple for each unit and the world model Q, describing the unit dependencies we would like to assign labels $z_1, \cdots, z_M$ to units $1, \cdots, M$, respectively, which minimize the expected loss

$$\sum_{(t_1, \cdots, t_M)} L(z_1, \cdots, z_M, t_1, \cdots, t_M) \cdot P(t_1, \cdots, t_M | x_1, \cdots, x_M, Q) \quad (1)$$

where $P(t_1, \cdots, t_M | x_1, \cdots, X_M, Q)$ is the probability that the assigned labels are the true labels given 1) the information $(x_1, \cdots, x_M)$ we have measured about the units and 2) the prior information Q we have about unit dependencies and where $L(z_1, \cdots, z_M, t_1, \cdots, t_M)$ is the loss incurred for the assignment of interpretations $z_1, \cdots, z_M$ to units $1, \cdots, M$ when the true interpretations are $t_1, \cdots, t_M$. Such an optimal discision rule is called a Bayes rule.

III. Pattern Recognition

In pattern recognition [9], [17], we often make the following assumptions about the world and unit measurement process. The first assumption states that the description process is local. When the unit i is being examined, no characteristics from any other unit but unit i affect the description obtained from unit i. Hence,

$$P(x_1, \cdots, x_M | t_1, \cdots, t_M, Q) = \prod_{i=1}^{M} P(x_i | t_1, \cdots, t_M, Q). \quad (2)$$

The second assumption states that the n-tuple measurement of unit i depends only upon the true interpretation associated with unit i and does not depend upon any relationships unit i may have with other units or upon the interpretation associated with any other unit. Hence,

$$P(x_i | t_1, \cdots, t_M, Q) = P(x_i | t_i), \qquad i = 1, \cdots, M. \quad (3)$$

Under these assumptions, the optimal decision rule determines interpretations $z_1, \cdots, z_M$ for units $1, \cdots, M$ which minimize

$$\sum_{(t_1, \cdots, t_M)} L(z_1, \cdots, z_M, t_1, \cdots, t_M) \cdot \prod_{m=1}^{M} P(x_m | t_m) P(t_1, \cdots, t_M) / P(x_1, \cdots, x_M). \tag{4}$$

The assumption often made in pattern recognition without context is that the units themselves are independent. There are no unit dependencies.

$$P(t_1, \cdots, t_M | Q) = \prod_{i=1}^{M} P(t_i). \tag{5}$$

Here, the true interpretation of any one unit does not constrain the true interpretation of any other unit.

A. The Loss Function

For each possible M-tuple of true interpretations and for each possible M-tuple of assigned interpretations, the loss function evaluates the consequences, summarizing them in simple terms of economic loss. There are a variety of reasonable definitions for the loss function. Perhaps the most common one is to have no loss for a correct joint assignment and unit loss for any incorrect joint assignment. Here, correct assignment means that each of the M assigned interpretations are correct. Thus, there is no distinction in loss between an incorrect assignment in which only one unit is incorrectly assigned or an incorrect assignment in which all units are incorrectly assigned. Such a loss function is defined by

$$L(z_1, \cdots, z_M, t_1, \cdots, t_M) = \begin{cases} 0, & \text{when } z_m = t_m, m = 1, \cdots, M \\ 1, & \text{otherwise.} \end{cases} \tag{6}$$

Under assumption (5) and in the case that the loss function is defined by (6), the Bayes decision procedure is to give interpretation z_i to unit i where label z_i maximizes $p(x_i | z_i) p(z_i)$ as stated in the following equation:

$$P(x_i | z_i) P(z_i) \geqslant P(x_i | z) P(z) \qquad \text{for all } \; z \in L.$$

To see this, first suppose

$$L(z_1, \cdots, z_M, t_1, \cdots, t_M) = 1 - \prod_{m=1}^{M} G(z_m, t_m). \tag{7}$$

Then,

$$\begin{aligned} &\min_{z_1, \cdots, z_M} \sum_{t_1, \cdots, t_M} L(z_1, \cdots, z_M, t_1, \cdots, t_M) \cdot \prod_{m=1}^{M} P(x_m | t_m) P(t_m) \\ &= P(x_1, \cdots, x_M) - \max_{z_1, \cdots, z_M} \prod_{m=1}^{M} \left[\sum_{t_m} G(z_m, t_m) \cdot P(x_m | t_m) P(t_m) \right] \\ &= P(x_1, \cdots, x_M) - \prod_{m=1}^{M} \max_{z_m} \left[\sum_{t_m} G(z_m, t_m) \cdot P(x_m | t_m) P(t_m) \right]. \end{aligned}$$

The product and maximization can be interchanged because the quantity in square brackets is only a function of z_m.

Now note that the loss function of (6) can be expressed by (7) by defining

$$G(z_m, t_m) = \begin{cases} 0 & \text{if } \; z_m \neq t_m \\ 1 & \text{otherwise.} \end{cases} \tag{8}$$

From this (7) follows directly.

Another kind of the loss function can be given by

$$L(z_1, \cdots, z_M, t_1, \cdots, t_m) = \sum_{m=1}^{M} L(z_m, t_m). \tag{9}$$

In this case

$$\begin{aligned} &\min_{z_1, \cdots, z_M} \sum_{t_1, \cdots, t_M} \sum_{m=1}^{M} L(z_m, t_m) \prod_{n=1}^{M} P(x_n | t_n) P(t_n) \\ &= \min_{z_1, \cdots, z_M} \sum_{m=1}^{M} \sum_{t_m} L(z_m, t_m) P(x_m | t_m) \cdot P(t_m) \prod_{\substack{k=1 \\ k \neq m}}^{M} P(x_k) \\ &= \sum_{m=1}^{M} \left[\min_{z_m} \sum_{t_m} L(z_m, t_m) P(x_m | t_m) P(t_m) \right] \cdot \prod_{\substack{k=1 \\ k \neq m}}^{M} P(x_k). \end{aligned}$$

The interchange of minimization and summation is allowed because the terms in the summation depend only upon m and z_m. In this form it is clear that the minimization can occur term by term independently where the square brackets designate the term.

When $L(z_m, t_m)$ is given by

$$L(z_m, t_m) = \begin{cases} 0 & \text{if } \; z_m = t_m \\ 1 & \text{otherwise} \end{cases} \tag{10}$$

the loss function defined by (9) is called the total error loss function. It has loss equal to the number of incorrect assignments. In this case also, the decision rule given by (7) results.

When

$$L(z_m, t_m) = (z_m - t_m)^2$$

the individual unit loss being equal to the square of the difference between the true and assigned interpretations, the loss function defined by (9) is called the least-squares loss function. The Bayes decision rule can be found by taking derivatives, setting them to zero, and solving. The Bayes decision rule gives interpretation z_m to unit m, where

$$z_m = \sum_t t P(x_m | t) P(t) \; \sum_t P(x_m | t) P(t). \tag{11}$$

Of course, when the set of possible labels which are assigned to units has no natural arithmetic properties, the least-squares loss function makes no sense and cannot be used.

Unfortunately, the joint prior independence assumption of (5) is clearly inappropriate in pattern recognition problems which have a rich context. The next section discusses the next more complex assumption, the Markov dependence assumption, an assumption often used in signal processing.

B. The Markov Dependence Assumption

A weaker assumption than unit independence is the Markov assumption which is used when the units are linearly ordered and the true interpretation of any unit given the true interpretations of all the previous units depends only upon the interpretation of the immediately preceeding unit in the order. That is,

$$P(t_i \mid t_1, \cdots, t_{i-1}, Q) = P(t_i \mid t_{i-1}). \tag{12}$$

Then using the identity

$$\begin{aligned} P(t_1, \cdots, t_M \mid Q) &= P(t_M \mid t_1, \cdots, t_{M-1}, Q) \\ &\quad \cdot P(t_{M-1} \mid t_1, \cdots, t_{M-2}, Q), \cdots, P(t_2 \mid t_1) P(t_1) \end{aligned} \tag{13}$$

we obtain

$$P(t_1, \cdots, t_M \mid Q) = \prod_{i=1}^{M} P(t_i \mid t_{i-1}). \tag{14}$$

Hence, minimizing the expected loss (4) is equivalent to minimizing

$$\sum_{t_1, \cdots, t_M} L(z_1, \cdots, z_M, t_1, \cdots, t_M) \cdot \prod_{m=1}^{M} P(x_m \mid t_m) P(t_m \mid t_{m-1}).$$

With the loss function defined by (6), the best decision procedure chooses interpretations $z_1, \cdots, z_M$ which satisfy the maximality condition

$$\prod_{i=1}^{M} P(x_i \mid z_i) P(z_i \mid z_{i-1}) \geqslant \prod_{i=1}^{M} P(x_i \mid z'_i) P(z'_i \mid z'_{i-1}) \quad \text{for all } (z'_1, \cdots, z'_M) \in L^M. \tag{15}$$

The choice of $z_1, \cdots, z_M$ satisfying this maximality condition cannot be independently done unit by unit. It is a dynamic programming problem [3], [8] and requires on the order of $|L|^2 M$ operations where $|L|$, the size of the set L, is the number of possible values for each interpretation. In the signal processing contexts, the optimizing algorithm is called the Viterbi algorithm and it is described in Section III-B-1.

When the loss function $L(z_1, \cdots, z_M, t_1, \cdots, t_M)$ has the form of a sum of partial losses as in (9), the minimal expected loss is, with the proportionality constant $1/P(x_1, \cdots, x_M)$, equal to

$$\min_{z_1, \cdots, z_M} \sum_{m=1}^{M} \left[\sum_{t_1, \cdots, t_M} L(z_m, t_m) \cdot P(x_1, \cdots, x_M, t_1, \cdots, t_M \mid Q) \right]. \tag{16}$$

Because the term in square brackets is a function only of z_m once the measurements $x_1, \cdots, x_M$ are known, the order of minimization and summation may be interchanged. Expression (16) is equivalent to

$$\sum_{m=1}^{M} \min_{z_m} \sum_{t_1, \cdots, t_M} L(z_m, t_m) \cdot P(x_1, \cdots, x_M, t_1, \cdots, t_M \mid Q). \tag{17}$$

From this expression (17) it is clear that the interpretation for each unit m can be chosen independently as that value m which minimizes

$$\sum_{t_1, \cdots, t_M} L(z_m, t_m) P(x_1, \cdots, x_M, t_1, \cdots, t_M \mid Q). \tag{18}$$

With the Markov assumption of (12) and when $L(z_m, t_m)$ is given by (10), thereby making $L(z_1, \cdots, z_M, t_1, \cdots, t_M)$ count the total number of incorrect assignments, minimizing (18) reduces to determining the interpretation z_m which maximizes $P(z_m, x_1, \cdots, x_M \mid Q)$.

With the Markov assumption (12) and when $L(z_m, t_m)$ is the squared difference between z_m and t_m, thereby making $L(z_1, \cdots, z_M, t_1, \cdots, t_M)$ be the total squared error, the minimizing z_m to (18) is obtained by

$$z_m = \frac{\sum_{t_m} t_m P(t_m, x_1, \cdots, x_M \mid Q)}{\sum_{t_m} P(t_m, x_1, \cdots, x_M \mid Q)}. \tag{19}$$

In either the case of the total incorrect assignment loss function or the total squared error loss function the probabilities

$$P(t_m, x_1, \cdots, x_M \mid Q), \quad m = 1, \cdots, M$$

must be determined. Under the Markov assumption (12) and the conditional independence assumptions (2) and (3)

$$P(t_1, \cdots, t_M, x_1, \cdots, x_M \mid Q) = \prod_{m=1}^{M} P(x_m \mid t_m) P(t_m \mid t_{m-1}). \tag{20}$$

Hence, the required probabilities can be obtained by taking the appropriate sums of products. In Section III-B-2, we describe the BAMPS algorithm for computing all the $M|L|$ probabilities $P(z_m, x_1, \cdots, x_M \mid Q)$ in the order of $M|L|^2$ operations.

1) The Viterbi Algorithm: The Viterbi algorithm [8] determines the Bayes labeling $z_1, \cdots, z_M$ under loss function (6) and Markov assumption (12) by determining the maximizing $z_1, \cdots, z_M$ of

$$\sum_{i=1}^{M} [\log P(x_i \mid z_i) + \log P(z_i \mid z_{i-1})].$$

The algorithm is most easily understood by drawing a graph having M columns of nodes, each column having $|L|$ nodes. The jth node in column i corresponds to the assignment of the jth interpretation value v_j to unit i (i.e., $z_i = v_j$). The weight in this node is $\log P(x_i \mid v_j)$. There are arcs between the nodes in each pair of adjacent columns and the graph looks like a trellis, as illustrated in Fig. 1. The weight of the arc connecting the jth node in column i with the kth node in column $i + 1$ is $\log P(v_k \mid v_j)$. The best joint interpretation $z_1, \cdots, z_M$ is

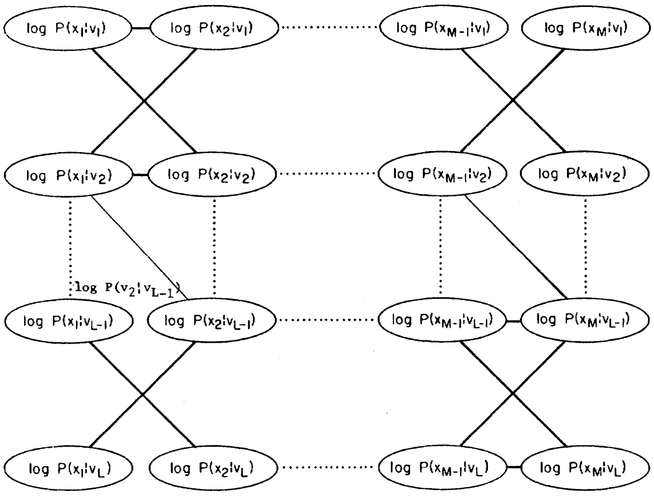


Fig. 1. Illustrated the trellis data structure that the Viterbi algorithm works on.

found by selecting one node in each column, the path from column one to column M determined by these selections being those giving the highest weight path.

The Viterbi algorithm making this highest weight path selection proceeds by recognizing that whatever the optimal path is, it must pass through one node in each column. Thus, if a calculation could be made in which every node in column i had the weight of the best path from some node in column one to it, then the calculation of the weight $w(i+1, v)$ of the best path from some node in column one to node v in column $i+1$ is simple. It is given by

$$w(i+1, v) = \log P(x_{i+1}|v) + \max_{v'} [w(i, v') + \log P(v|v')] \tag{21}$$

where $w(i, v')$ is the weight of the best path up to column i node v, $\log P(v|v')$ is the weight of the arc from node v' column i to node v column $i+1$, and $\log P(x_{i+1}|v)$ is the weight of node v column $i+1$.

To determine the best path, we need only look through the path weights $w(M, v)$ in the last column M. One of these has a maximal value. To have reached this maximal value, the path had to pass through some node in column $M-1$. If we save the node which maximizes each computation (21), z_{M-1} is easily determined by a table lookup. In a like manner the best z_{M-2} is determined and so on.

There have been two suggestions in the literature for a modified Viterbi algorithm which reduces the computational complexity with little effect on optimality. Shinghal *et al.* [23] suggest that instead of considering all possible labels at each column in the trellis, only consider those K labels for which $P(x|z)\,P(z)$ is highest. Erman *et al.* [6] indicate that in the HARPY speech understanding system instead of considering all possible labels at each column in the trellis, only those few labels which are part of the best paths through the given column need be considered. The modified search technique is called beam searching and is described by Rubin and Reddy [22].

2) The BAMPS Algorithm: The BAMPS algorithm determines the Bayes labeling $z_1, \cdots, z_M$ under the loss function (9) where the partial losses are given by (10) and under the Markov assumption for $P(x_1, \cdots, x_M, t_1, \cdots, t_M|Q)$ given by (20). It does this by computing the probabilities $P(z_m, x_1, \cdots, x_M|Q)$ which can be expressed as a sum of products due to the Markov assumption.

Define

$$a(n-1, z_n) = \sum_{t_1, \cdots, t_{n-1}} \left[\prod_{i=1}^{n-1} P(x_i|t_i) \cdot P(t_i|t_{i-1})\, P(x_n|z_n)\, P(z_n|t_{n-1}) \right] \tag{22}$$

and

$$b(n, z_n) = \sum_{t_{n+1}, \cdots, t_M} P(x_{n+1}|t_{n+1})\, P(t_{n+1}|z_n) \cdot \prod_{i=n+2}^{M} P(x_i|t_i)\, P(t_i|t_{i-1}). \tag{23}$$

It is clear from (20) and the definition of $a(n-1, z_n)$ and $b(n, z_n)$ that

$$a(n-1, z_n)\, b(n, z_n) = P(z_n, x_1, \cdots, x_M|Q). \tag{24}$$

The BAMPS algorithm, named by Lehan [14], stands for Bayes and Markov processing system. It is an order $M|L|^2$ algorithm for the computation of the $M|L|$ probabilities $P(z_n, x_1, \cdots, x_M|Q)$ by computing $a(n-1, z_n)$ and $b(n, z_n)$ as stated in (25) and (26) and then multiplying them together as in (24)

$$a(n-1, z_n) = \sum_{t_{n-1}} a(n-2, t_{n-1})\, P(x_n|z_n)\, P(z_n|t_{n-1}) \tag{25}$$

where $a(0, z_1) = P(x_1|z_1)\, P(z_1)$ for each value z_1 and

$$b(n, z_n) = \sum_{t_{n+1}} P(x_{n+1}|t_{n+1})\, P(t_{n+1}|z_n)\, b(n+1, t_{n+1}) \tag{26}$$

where $b(M, z_M) = 1$ for each value z_M.

Equation (25) is similar to the iterative procedure of Raviv [19] who shows how to make a decision on the nth unit using all the past measurements $x_1, \cdots, x_{n-1}$ and the current measurement x_n. Equation (26) is basically (25) with indexing starting from the end rather than the beginning. Equation (24) shows that a label can be assigned to the nth unit on the basis of all measurements: the past units $x_1, \cdots, x_{n-1}$, the current one x_n, and the future ones $x_{n+1}, \cdots, x_M$. The technique given by (24) is, therefore, more powerful then the technique described by Raviv. An approach related to the BAMPS algorithm can be found in Askar and Derin [1].

IV. Artificial Intelligence

For most problems with context, the Markov assumption (12) is too weak and the squared error loss functions are usually meaningless. Problems with context require a more sensitive way of handling the prior probability $P(z_1, \cdots, z_M|Q)$. To this end, artificial intelligence researchers such as Duda and

Hart [5] have implicitly assumed that the global interpretation $(z_1, \cdots, z_M)$ for units $(1, \cdots, M)$ is either allowable or not allowable and all allowable global interpretations have equal probability. This is an equal probability of ignorance assumption. But it is one that applies to the entire context and not to each unit individually. Artificial intelligence researchers call this kind of constraint the constraint which embodies the problem semantics. Thus, if the problem semantics are given by A, where $A \subseteq L^M$ is the set of allowable global interpretations, we have

$$P(z_1, \cdots, z_M | Q) = \begin{cases} \frac{1}{\#A} & \text{if } (z_1, \cdots, z_M) \in A \\ 0 & \text{if } (z_1, \cdots, z_M) \notin A. \end{cases} \tag{27}$$

Under our context equal probability of ignorance assumption of (27), the Bayes decision rule for loss function (6) determines interpretations $z_1, \cdots, z_M$ for units $1, \cdots, M$ which maximize

$$\begin{aligned} P(z_1, \cdots, z_M, x_1, \cdots, x_M | Q) &= P(z_1, \cdots, z_M | Q) \prod_{i=1}^{M} P(x_i | z_i) \\ &= \begin{cases} \frac{1}{\#A} \prod_{i=1}^{M} P(x_i | z_i) & \text{if } (z_1, \cdots, z_M) \in A \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{28}$$

The brute force algorithm to solve this optimization problem must then go sequentially through all consistent interpretation M-tuples $(z_1, \cdots, z_M)$ in the constraint set A and for each one evaluate

$$\prod_{i=1}^{M} P(x_i | z_i)$$

to find that consistent labeling which maximizes the product. This approach, used in Duda and Hart [5], is sometimes called a dictionary-based method.

Maximizing a product is equivalent to maximizing the sum of the logorithms of the terms in a product. Maximizing a sum over possibilities from some constraint set can be accomplished by a branch and bound state space search.

For the least-squares loss function which is less common in usual artificial intelligence problems, but which may be more useful in complex signal processing applications, minimizing (4) is equivalent to minimizing

$$\sum_{(t_1, \cdots, t_M) \in A} \sum_{n=1}^{M} (t_n - z_n)^2 \prod_{m=1}^{M} P(x_m | t_m). \tag{29}$$

Notice that the context equal probability of ignorance assumption has eliminated the prior probability $P(z_1, \cdots, z_M | Q)$. Expression (29) is minimized by

$$z_j = \frac{\sum_{(t_1, \cdots, t_M) \in A} t_j \prod_{m=1}^{M} P(x_m | t_m)}{\sum_{(t_1, \cdots, t_M) \in A} \prod_{m=1}^{M} P(x_m | t_m)}. \tag{30}$$

A difficulty with the solution (30) is that the labeling $z_1, \cdots, z_M$ determined by (30) may not be a member of the constraint set A. A natural variation, therefore, is to restrict the assigned interpretations to come from the constraint set A. This is easily put in the loss framework by having an infinitely high loss for any assignment not in the constraint set.

To accomplish finding $(z_1, \cdots, z_M) \in A$ minimizing (29) we can define the.constraint set A parametrically by

$$A = \left\{ (z_1, \cdots, z_M) \in L^M \middle| \text{ for some } a_1, \cdots, a_K, z_m = \sum_{k=1}^{K} a_k f_k(m) \right\} \tag{31}$$

where the basis functions satisfy the orthogonality conditions

$$\sum_m f_i(m) f_j(m) = \begin{cases} 0 & i \neq j \\ 1 & i = j. \end{cases}$$

Minimizing (29) under the constraint that $(z_1, \cdots, z_M) \in A$ is then equivalent to finding $a_1, \cdots, a_K$ which minimizes

$$\sum_{(t_1, \cdots, t_M) \in A} \sum_{n=1}^{M} \left(\sum_{k=1}^{K} a_k f_k(n) - t_n \right)^2 \prod_{m=1}^{M} P(x_m | t_m). \tag{32}$$

The minimizing values are given by

$$a_j = \frac{\sum_{n=1}^{M} f_j(n) \sum_{(t_1, \cdots, t_M) \in A} t_n \prod_{m=1}^{M} P(x_m | t_m)}{\sum_{(t_1, \cdots, t_M) \in A} \sum_{m=1}^{M} P(x_m | t_m)}. \tag{33}$$

The solution is then given by

$$z_m = \sum_{k=1}^{K} a_k f_k(m).$$

Modeling the constraints among the units by a general subset $A \subseteq L^M$ which specifies the allowable interpretation M-tuples is a powerful concept. By itself, however, it may not lead to any practical implementations. The problem is one of size. For the loss function we have considered, we need to have a way to access each of the M-tuples in A, a set too large to store in memory directly, without having to generate all the M-tuples in L^M and test each one to see if it satisfies the conditions defining A. Artificial intelligence researchers sometimes accomplish this by specifying A through a small set of production rules or constraint rules. Accessing each of the M-tuples in A is then accomplished as a constrained search which searches a space slightly larger than A but much smaller than L^M. In the next section we discuss a way of specifying A by relational constraints.

A. Specifying the Constraint Set

One possibility is to specify A by specifying possibly overlapping pieces of the M-tuples in A. If this decomposition of A matches a corresponding decomposition in the loss function, the two decompositions can then lead to algorithms of the form: solve a classic optimization problem of the form (3) for each of the small pieces and then determine a total solution from the solution on each of the pieces.

In this framework, there is a natural layering. The layering occurs because there is a unit change from the lowest level observational unit to the higher level piece which consists of a group of related observational units. Relatively little is known about the varieties of possible unit change or the naturally associated loss functions.

We now describe one way to specify A by specifying possibly overlapping pieces of the M-tuples in A. Each piece is a tuple of related units. Each piece is a higher level unit. Let T be a set containing all tuples of indexes of related units. Thus, if $(i_1, \cdots, i_K) \in T$, then units $i_1, \cdots, i_k$ are all related and $(i_1, \cdots, i_K)$ is the higher level unit. Corresponding to each $(i_1, \cdots, i_K) \in T$ is a set $R(i_1, \cdots, i_K)$, the projection of A onto the $(i_1, \cdots, i_K)$ subspace, which contains all tuples of legal or allowable interpretations for units $i_1, \cdots, i_K$. The set A is then represented by the set of all interpretation tuples $(t_1, \cdots, t_M)$ such that $(i_1, \cdots, i_K) \in T$ implies $(t_{i_1}, \cdots, t_{i_M}) \in R(i_1, \cdots, i_K)$. Such an A is equal to the intersection of its inverse projections taken over all $(i_1, \cdots, i_K) \in T$.

This formulation is the one used by Waltz [28] in line labeling problems and by Tenenbaum and Barrow [25] in their interpretation guided segmentation experiments. Riseman and Ehrich [20] in a character recognition situation used a related approach. There, each $R(i_1, \cdots, i_k)$ corresponds to a set of allowable K-gram of characters in the $i_1, \cdots, i_k$ positions of a word. They set $K = 2$ and used the bigrams to eliminate nonsense alternatives from possible labels produced by a context independent classifier. Finding all the M-tuples in A is a consistent labeling problem (Haralick and Shapiro [31]). Because the consistent labeling problem is an *NP*-complete problem, it is a powerful way to specify constraints.

B. Corresponding Loss Functions

Corresponding to this decomposition of the constraint set A is a corresponding representation of the loss functions as a sum of partial losses.

$$L(z_1, \cdots, z_M, t_1, \cdots, t_M) = \sum_{(i_1, \cdots, i_K) \in T} L(z_{i_1}, \cdots, z_{i_K}, t_{i_1}, \cdots, t_{i_K}). \quad (34)$$

The loss function of (34) leads to the problem of determining interpretations $z_1, \cdots, z_M$ which minimize

$$\sum_{(t_1, \cdots, t_M) \in A} \sum_{(i_1, \cdots, i_K) \in T} \cdot L(z_{i_1}, \cdots, z_{i_K}, t_{i_1}, \cdots, t_{i_K}) \prod_{m=1}^{M} P(t_m | x_m). \quad (35)$$

With a couple of notational definitions, (35) can be rewritten. For each $(i_1, \cdots, i_K) \in T$, denote its extension by $i_{K+1}, \cdots, i_M$. The extension makes $i_1, \cdots, i_M$ a permutation of $1, \cdots, M$. For each $(t_{i_1}, \cdots, t_{i_K}) \in R(i_1, \cdots, i_K)$, we define the set $S_{i_1, \cdots, i_K}(t_{i_1}, \cdots, t_{i_K})$ of its extension in A by

$$S_{i_1, \cdots, i_K}(t_{i_1}, \cdots, t_{i_K}) = \{(t_{i_{K+1}}, \cdots, t_{i_M}) | (t_1, \cdots, t_M) \in A\}.$$

Using the extension concept (35) becomes

$$\sum_{(i_1, \cdots, i_K) \in T} \sum_{(t_{i_1}, \cdots, t_{i_K}) \in R(i_1, \cdots, i_K)} \cdot L(z_{i_1}, \cdots, z_{i_K}, t_{i_1}, \cdots, t_{i_K}) \prod_{k=1}^{K} P(t_{i_k} | x_{i_k}) \cdot \left[\sum_{(t_{i_{K+1}}, \cdots, t_{i_M}) \in S_{i_1, \cdots, i_K}(t_{i_1}, \cdots, t_{i_K})} \cdot \prod_{m=K+1}^{M} P(t_{i_m} | x_{i_m}) \right]. \quad (36)$$

Determining the minimizing $z_1, \cdots, z_M$ is computationally difficult because of the last bracketed term in (36).

If we neglect this term (by assuming they are all equal for each $t_{i_1}, \cdots, t_{i_K}$) we can solve a suboptimal problem of choosing $z_1, \cdots, z_M$ which minimizes

$$\sum_{(i_1, \cdots, i_K) \in T} \sum_{(t_{i_1}, \cdots, t_{i_K})(i_1, \cdots, i_K) \in R} \cdot L(z_{i_1}, \cdots, z_{i_K}, t_{i_1}, \cdots, t_{i_K}) \prod_{k=1}^{K} P(t_{i_k} | x_{i_k}). \quad (37)$$

This suboptimal problem can be solved as a dynamic programming problem or a branch and bound search. For each $z_{i_1}, \cdots, z_{i_K}$ the function

$$f_{i_1, \cdots, i_K}(z_{i_1}, \cdots, z_{i_K}) = \sum_{(t_{i_1}, \cdots, t_{i_K}) \in R(i_1, \cdots, i_K)} \cdot L(z_{i_1}, \cdots, z_{i_K}, t_{i_1}, \cdots, t_{i_K}) \prod_{k=1}^{K} P(t_{i_k} | x_{i_k})$$

is evaluated. Expression (30) just becomes

$$\sum_{(i_1, \cdots, i_K) \in T} f_{i_1, \cdots, i_K}(z_{i_1}, \cdots, z_{i_K}),$$

the form of a standard multidimensional dynamic programming problem.

If we are not interested in any interpretation which unconditionally minimizes (37), but only an interpretation $(z_1, \cdots, z_M) \in A$ which suboptimally minimizes (37), the state space branch and bound search can be done even more efficiently (Shapiro and Haralick [32]). Note that this formulation is closer to an optimal formulation than that in Riseman and Ehrich [20] or Hanson *et al.* [10], and to our knowledge it has not previously appeared in the literature.

C. Loss Function Using Context and Leading to More Efficient Computation

In this section we discuss two alternative kinds of loss functions which permit the combining of the alternative solutions for the small pieces to occur in linear time rather than the exponential time of the combinational search required in Section IV-B. The formulations arising out of these loss functions have not previously appeared in the literature.

These loss function decompositions are different than the simple sum of (34). Letting T_m be that subset of T containing tuples one of whose components has value m

$$T_m = \{(i_1, \cdots, i_K) \in T \mid \quad \text{for some} \quad k, i_k = m\},$$

we first consider the loss function given by

$$L(z_1, \cdots, z_M, t_1, \cdots, t_M) = \sum_{m=1}^{M} \sum_{(i_1, \cdots, i_K) \in T_m} \cdot L_{m(i_1, \cdots, i_K)}(z_m, t_{i_1}, \cdots, t_{i_K}). \tag{38}$$

In this loss function, the loss for the assignment of interpretations $z_1, \cdots, z_M$ given that the true interpretations are $t_1, \cdots, t_M$ is calculated as a sum of partial losses, one partial loss term for each unit to be assigned. The partial loss associated with assigning interpretation z_m to unit m is obtained by summing the loss of each problem piece containing unit m. These problem pieces are those associated with the higher level units $(i_1, \cdots, i_K) \in \mathrm{T}_m$. Each problem piece $(i_1, \cdots, i_K)$ associated with unit m has a loss

$$L_{m(i_1, \cdots, i_K)}(z_{m_1}, \cdots, t_{i_K}),$$

the loss of assigning the interpretation z_m to unit u_m where the true interpretation for the higher level unit $(i_1, \cdots, i_K)$ is $(t_{i_1}, \cdots, t_{i_K})$.

Letting e^* be the minimal loss, we have

$$e^* = \min_{z_1, \cdots, z_M} \sum_{(t_1, \cdots, t_M) \in A} \sum_{m=1}^{M} \sum_{(i_1, \cdots, i_K) \in T_m} \cdot L_{m(i_1, \cdots, i_K)}(z_m, t_{i_1}, \cdots, t_{i_K}) \prod_{n=1}^{M} P(t_n \mid x_n). \tag{39}$$

The order of the summation on $(t_1, \cdots, t_M)$, m, and $(i_1, \cdots, i_K)$ can be interchanged.

$$e^* = \min_{z_1, \cdots, z_M} \sum_{m=1}^{M} \left[\sum_{(i_1, \cdots, i_K) \in T_m} \sum_{(t_1, \cdots, t_M) \in A} \cdot L_{m(i_1, \cdots, i_K)}(z_m, t_{i_1}, \cdots, t_{i_K}) \prod_{n=1}^{M} P(t_n \mid x_n) \right]. \tag{40}$$

The term in square brackets is a function of z_m alone. Hence, the minimum of the sum is the sum of the minimums.

$$e^* = \sum_{m=1}^{M} \min_{z_m} \sum_{(i_1, \cdots, i_K) \in T_m} \sum_{(t_1, \cdots, t_M) \in A} \cdot L_{m(i_1, \cdots, i_K)}(z_m, t_{i_1}, \cdots, t_{i_K}) \prod_{n=1}^{M} P(t_n \mid x_n). \tag{41}$$

Using the same notational convention for the extension as before, we can rewrite (41) as

$$e^* = \sum_{m=1}^{M} \min_{z_m} \sum_{(i_1, \cdots, i_K) \in T_m} \sum_{(t_{i_1}, \cdots, t_{i_K}) \in R(i_1, \cdots, i_K)} \cdot L_{m(i_1, \cdots, i_K)}(z_m, t_{i_1}, \cdots, t_{i_K}) \prod_{k=1}^{K} P(t_{i_k} \mid x_{i_k}) \cdot \left[\sum_{(t_{i_{K+1}}, \cdots, t_{i_M}) \in S_{i_1, \cdots, i_K}(t_{i_1}, \cdots, t_{i_K})} \cdot \prod_{m=K+1}^{M} P(t_{i_m} \mid x_{i_m}) \right]. \tag{42}$$

Assuming that the last term in square brackets takes the same value for each $t_{i_1}, \cdots, t_{i_K}$ we obtain an approximate minimizing $z_1, \cdots, z_M$. Take each interpretation z_m to be that z_m minimizing

$$\sum_{(i_1, \cdots, i_K) \in T_m} \left[\sum_{(t_{i_1}, \cdots, t_{i_K}) \in R(i_1, \cdots, i_K)} \cdot L_{m(i_1, \cdots, i_K)}(z_m, t_{i_1}, \cdots, t_{i_K}) \prod_{k=1}^{K} P(t_{i_k} \mid x_{i_k}) \right]. \tag{43}$$

This approximate formulation is computationally advantageous. The term in square brackets is the expected loss obtained by assigning unit m to interpretation z_m for problem piece $(i_1, \cdots, i_K)$. Thus, each problem piece $(i_1, \cdots, i_K)$ can be computed independently for each possible value of z_m. The global minimizing z_m is then just that z_m which minimizes the sum of the expected losses of interpretation z_m, the sum being taken over all the problem pieces.

Another alternative loss function to (38) is given by

$$L(z_1, \cdots, z_M, t_1, \cdots, t_m) = \sum_{m=1}^{M} \min_{(i_1, \cdots, i_K) \in T_m} \cdot L_{m(i_1, \cdots, i_K)}(z_m, t_{i_1}, \cdots, t_{i_K}). \tag{44}$$

In this loss function the loss for the assignment of interpretations $z_1, \cdots, z_M$, given that the true interpretations are $t_1, \cdots, t_M$, is calculated as a sum of partial losses, one partial loss term for each unit to be assigned. The partial loss associated with assigning interpretation z_m to unit m is obtained by going through each problem piece containing unit m. These problem pieces are then represented by the higher level units $(i_1, \cdots, i_K) \in T_m$. Each problem piece $(i_1, \cdots, i_K)$ associated with unit m has a loss

$$L_{m(i_1, \cdots, i_K)}(z_m, t_{i_1}, \cdots, t_{i_K})$$

the loss of assigning the interpretation z_m to unit m when the true interpretation of the higher level unit $(i_1, \cdots, i_K)$ is $(t_{i_1}, \cdots, t_{i_K})$. Of all the problem pieces unit m participates in, one of them will have the smallest loss. This smallest loss is the partial loss associated with unit m.

The loss function of (44) does not permit a computationally efficient solution for the exact minimizing $z_1, \cdots, z_M$. However, it does provide for a computationally efficient solution for interpretations $z_1, \cdots, z_M$ whose expected loss bounds the minimizing ones. Letting e^* be the minimal expected total loss, we have

$$e^* = \min_{z_1,\cdots,z_M} \sum_{(t_1,\cdots,t_M)\in A} \sum_{m=1}^{M} \min_{(i_1,\cdots,i_K)\in T_m} \cdot L_m(z_m, t_{i_1}, \cdots, t_{i_K}) \prod_{n=1}^{M} P(t_n|x_n). \tag{45}$$

Performing interchanges similar to before, there results

$$e^* = \sum_{m=1}^{M} \min_{z_m} \sum_{(t_1,\cdots,t_M)\in A} \min_{(i_1,\cdots,i_K)\in T_m} \cdot L_m(z_m, t_{i_1}, \cdots, t_{i_K}) \prod_{n=1}^{M} P(t_n|x_n). \tag{46}$$

In general, the sum of the minimums is less than the minimum of the sums. This lets us bound e^* by

$$e^* < \sum_{m=1}^{M} \min_{z_m} \min_{(i_1,\cdots,i_K)\in T_m} {}_{(t_1,\cdots,t_M)\in A} \cdot L_{m(i_1,\cdots,i_K)}(z_m, t_{i_1}, \cdots, t_{i_K}) \prod_{n=1}^{M} P(t_n|x_n). \tag{47}$$

Using the same notational conventions as before we can rewrite (40) as

$$e^* < \sum_{m=1}^{M} \min_{z_m} \min_{(i_1,\cdots,i_K)\in T_m} \sum_{(t_{i_1},\cdots,t_{i_K})\in R(i_1,\cdots,i_K)} \cdot \sum_{(t_{i_{K+1}},\cdots,t_{i_M})\in S_{i_1,\cdots,i_K}(t_{i_1},\cdots,t_{i_K})} \cdot L_{m(i_1,\cdots,i_K)}(z_m, t_{i_1}, \cdots, t_{i_K}) \prod_{n=1}^{M} P(t_n|x_n) \tag{48}$$

which results in

$$e^* \leqslant \sum_{m=1}^{M} \min_{z_m} \min_{(i_1,\cdots,i_K)\in T_m} \sum_{(t_{i_1},\cdots,t_{i_K})\in R(i_1,\cdots,i_K)} \cdot L_{m(i_1,\cdots,i_K)}(z_m, t_{i_1}, \cdots, t_{i_K}) \prod_{k=1}^{K} P(t_{i_k}|x_{i_k}) \cdot \left[\sum_{(t_{i_{K+1}},\cdots,t_{i_M})\in S_{i_1,\cdots,i_K}(t_{i_1},\cdots,t_{i_K})} \cdot \prod_{m=K+1}^{M} P(t_{i_m}|x_{i_m}) \right]. \tag{49}$$

Assuming that the last term in square brackets takes the same values for each $t_{i_1}, \cdots, t_{i_K}$ we obtain an approximate minimizing $z_1, \cdots, z_M$. Take each interpretation z_m to be that z_m minimizing

$$\min_{(i_1,\cdots,i_K)\in T_m} \sum_{(t_{i_1},\cdots,t_{i_K})\in R(i_1,\cdots,i_K)} \cdot L_{m(i_1,\cdots,i_K)}(z_m, t_{i_1}, \cdots, t_{i_K}) \prod_{k=1}^{K} P(t_{i_k}|x_{i_k}). \tag{50}$$

This formulation is computationally advantageous because it allows each interpretation z_m to be obtained independently. The expected loss for each value of interpretation z_m is computed for each problem piece $(i_1, \cdots, i_K)$. There will be some $(i_1, \cdots, i_K)$ and interpretation value z_m having the smallest expected loss. Assign this interpretation value z_m to unit m. This formulation is related to the facet iterations of Haralick and Watson [12] which are generalizations of the smoothing technique introduced by Tomita and Tsuji [26] and Nagao and Matsuyama [16].

V. Probabilistic Relaxation

In image analysis there have been numerous papers on the effective use of cooperative processing through the mechanism of probabilistic relaxation. The idea was first introduced by Rosenfeld *et al.* [21]. There have been a variety of papers analyzing some aspects of the underlying theory in Haralick *et al.* [11], Peleg [18], Zucker *et al.* [29], and Zucker *et al.* [30].

In cooperative processing, neighboring information positively or negatively reinforces the weights for each local unit of information depending on the compatibility of the neighboring information with the local information. After each relaxation iteration, the resulting values are more consistent with the prior knowledge of information dependencies and global context.

Probabilistic relaxation has been a mechanism whose theory has not been well understood. In this section we state some general conditional independence conditions which give probabilistic relaxation the interpretation that each iteration computes the conditional probability of each units category interpretation given a new context which is the context of the previous iteration enlarged by one neighborhood width. This interpretation implies that relaxation iterations must only continue until either the conditional independence assumptions no longer hold or until the entire context is taken into account, whichever comes first. When the entire context is taken into account, the computed probabilities are the conditional probabilities of a unit having a category interpretation given the measurements made of all the units. Thus, assigning that category label having highest computed probability is a Bayes decision rule under the total error loss function.

Section V-A makes precise this interpretation of probabilistic relaxation and Section V-B states the conditional probability assumption and shows that these conditional probability assumptions lead to the interpretation given in Section IV-A.

A. Interpretation of Probabilistic Relaxation

In this section we develop an interpretation for the relaxation equation

$$P(q_i, t+1) = \frac{P(q_i, t) \prod_{j\in N(i)} \sum_{q_j} P(q_j, t) J_{ij}(q_i, q_j)}{\sum_{s_i} P(s_i, t) \prod_{j\in N(i)} \sum_{q_j} P(q_j, t) J_{ij}(s_i, q_j)} \tag{51}$$

where

$$J_{ij}(q_i, q_j) = \frac{P(q_i, q_j)}{P(q_i)\,P(q_j)}$$

and $N(i)$ is the set of neighbors for unit i.

Our interpretation states that $P(q_i, t)$ is the conditional probability that unit i takes label q_i given the tth level context. Furthermore, the context at each iteration grows by an entire neighborhood width surrounding the previous level context.

To make these remarks precise, we will have to make a change in notation in which the context is explicitly written. Context means the units and their corresponding measurements where the units come from some general neighborhood. Initially, a measurement is made of each unit. We denote by d_i the measurement made of unit i. This is its immediate context. The neighborhood context for unit i is the measurement d_i plus all the measurements of units in the neighborhood of unit i. The next larger context for unit i is measurement d_i, plus all the measurements of units in the neighborhood of unit i, plus all the measurements of units in the neighborhood of the neighbors of unit i. The global context consists of all the units $1, \cdots, M$.

We denote by $Z_i(t)$, the units in the tth level context for unit i and by $N(i)$ the set of neighbors for unit i. $Z_i(1) = \{i\}$. The units in the successive level contexts can be defined iteratively by $Z_i(t+1) = \{j \mid \text{for some } k \in Z_i(t), j \in N(k)\}$.

The purpose of the probabilistic relaxation is to compute for each unit i and label q_i the conditional probability $P(q_i | d_1, \cdots, d_M)$, where it is understood that a subscript n on a label or measurement designates that the label or measurement is for unit n. Thus, $P(q_2)$ designates a generally different probability value than $P(q_3)$ even if $q_2 = q_3$. A more complete notation would write $P_2(q_2)$ for $P(q_2)$. We use the shorter notation to keep from writing unnecessarily complex expressions.

We will need to write conditional probabilities like $P(q_i | d_1, \cdots, d_M)$ but where the condition is on measurements for some arbitrary subset S of units whose names are not explicitly known. We denote this kind of conditional probability by $P(q_i | d_k : k \in S)$. Thus, if $S = \{1, 3, 6, 7\}$, we write $P(q_i | d_k : k \in S)$ for $P(q_i | d_1, d_3, d_6, d_7)$. Likewise, if $T = \{2, 3, 4\}$ we will write $P(q_n : n \in T | d_k : k \in S)$ for $P(q_2, q_3, q_4 | d_1, d_3, d_6, d_7)$.

In this notation, the relaxation begins with $P[q_i | d_k : k \in Z_i(1)]$ and terminates with the probabilities $P(q_i | d_k : k \in \{1, \cdots, I\})$. Therefore, we have the following interpretation for the relaxation equation (51):

$$P(q_i | d_k : k \in Z_i(t+1)) = \frac{P(q_i | d_k : k \in Z_i(t)) \prod_{j \in N(i)} \sum_{q_j} P(q_j | d_k : k \in Z_j(t))\, J_{ij}(q_i, q_j)}{\sum_{t_i} P(t_i, | d_k : k \in Z_i(t)) \prod_{j \in N(i)} \sum_{q_j} P(q_j | d_k : d \in Z_j(t))\, J_{ij}(q_i, q_j)}. \tag{52}$$

B. Basis for the Interpretation

In this section, we state the two conditional probability assumptions which make the relaxation equation (52) a valid equation. The assumptions are

$$P(q_i, q_k : k \in N(i)) = \frac{\prod_{j \in N(i)} P(q_i, q_j)}{P(q_i)^{|N(i)|-1}} \tag{53}$$

and

$$P(d_k : k \in Z_i(t+1) | q_i, q_k : k \in N(i)) = P(d_k : k \in Z_i(t) | q_i) \prod_{j \in N(i)} P(d_k : k \in Z_j(t) | q_j). \tag{54}$$

To see how this leads to the relaxation equation (52), consider the conditional probability that unit i takes label q_i given its level $t+1$ context. By definition of conditional probability,

$$\begin{aligned} P(q_i | d_k : k \in Z_i(t+1)) &= \frac{P(q_i, d_k : k \in Z_i(t+1))}{P(d_k : k \in Z_i(t+1))} = \frac{\sum_{j \in N(i)} \sum_{q_j} P(q_i, q_k : k \in N(i), d_k : k \in Z_i(t+1))}{P(d_k : k \in Z_i(t+1))} \\ &= \frac{\sum_{j \in N(i)} \sum_{q_j} P(d_k : k \in Z_i(t+1) | q_i, q_k : k \in N(i))\, P(q_i, q_k : k \in N(i))}{P(d_k : k \in Z_i(t+1))}. \end{aligned} \tag{55}$$

Upon using assumptions (53) and (54) by substituting into (55), there results

$$\begin{aligned} &P(q_i | d_k : k \in Z_i(t+1)) \\ &= \frac{\propto P(d_k : k \in Z_i(t) | q_i)}{P(d_k : k \in Z_i(t+1))\, P(q_i)^{|N(i)|-1}} \\ &\quad \times \sum_{j \in N(i)} \sum_{q_j} \prod_{n \in N(i)} (P(d_k : k \in Z_n(t) | q_n)\, P(q_i, q_n). \end{aligned} \tag{56}$$

Again using the definition of conditional probability, we may rewrite (8).

$$\begin{aligned} &P(q_i | d_k : k \in Z_i(t+1)) \\ &= \frac{\propto P(d_k : k \in Z_i(t)) \prod_{n \in N(i)} P(d_k : k \in Z_n(t))}{P(d_k : k \in Z_i(t+1))} \\ &\quad \cdot P(q_i | d_k : k \in Z_i(t)) \\ &\quad \times \sum_{j \in N(i)} \sum_{q_j} \prod_{n \in N(i)} P(q_n | d_k : k \in Z_n(t)) \\ &\quad \cdot \frac{P(q_i, q_n)}{P(q_i)\, P(q_n)}. \end{aligned} \tag{57}$$

The sums of products in (57) can be simplified. The product contains terms each of which depends simply upon n. All other variables involved are constant with respect to sums and product. Hence,

$$\sum_{j\in N(i)} \sum_{q_j} \prod_{n\in N(i)} P(q_n|d_k: k\in Z_n(t)) \frac{P(q_i, q_n)}{P(q_i)P(q_n)}$$

$$= \prod_{n\in N(i)} \sum_{q_n} P(q_n|d_k: k\in Z_n(t)) \frac{P(q_i, q_n)}{P(q_i)P(q_n)}. \tag{58}$$

Finally, noticing that

$$\sum_{q_i} P(q_i|d_k: k\in Z_i(t+1)) = 1 \tag{59}$$

we can divide both sides of (57) by the sum in (59). The first term in square brackets on the right-hand side of (57) is a constant with respect to the summation and, therefore, cancels in the division. Thus, upon making the substitution of (58) and the division of (59) there results the relaxation equation

$$P(q_i|d_k: k\in Z_i(t+1)) = \frac{P(q_i|d_k: k\in Z_i(t)) \prod_{j\in N(i)} \sum_{q_j} P(q_j|d_k: k\in Z_j(t)) \frac{P(q_i, q_j)}{P(q_i)P(q_j)}}{\sum_{s_i} P(s_i|d_k: k\in Z_i(t)) \prod_{j\in N(i)} \sum_{q_j} P(q_j|d_k: k\in Z_j(t)) \frac{P(s_i, q_j)}{P(s_i)P(q_j)}}.$$

C. Probabilistic Relaxation Summary

We have shown that with the conditional independence assumptions of (53) and (54), the relaxation equation (52) results. This equation states that the probability of a category interpretations given the $(t+1)$-level context for any unit can be computed from the same kind of t-label probabilities of the unit and its neighbors. By iterating (52) until the entire context is taken into account, it becomes possible to compute the probability that a unit has a label given the entire context.

The relaxation iterations can be continued until either the entire context has been taken into account or until a context level is reached where one of the conditional independence assumptions of (53) or (54) no longer holds.

The consequence of this explanation of cooperative processing and relaxation is that we now must begin to determine for each application the precise context level at which the assumptions (53) and (53) no longer hold.

VI. Conclusion

We have reviewed the Bayesian framework for decision making in context and have surveyed the usual loss functions and assumptions on the joint prior probability function classically employed in the pattern recognition literature. By making these typical choices and assumptions explicit, it became obvious that they are inadequate for decision making in context.

We then reviewed from the perspective of Bayesian decision theory, one of the ways in which the artificial intelligence researchers take context into account. The corresponding choices for loss function and assumptions on the joint probability prior probability function are much more complex. The algorithms for determining the smallest expected loss decisions are complicated state space searches. Often, suboptimal more easily computed solutions are sought.

In summary, we have seen that the way to handle context is to make more realistic assumptions on the joint prior probability function. Using the more complex prior, decisions are made by computing the conditional probability of the interpretation label for a unit given all the measurements from all the units. Depending upon the prior probability assumption there may be a corresponding natural form for the loss function which permits some kind of partial problem decomposition.

References

[1] M. Askar and H. Derin, "A recursive algorithm for the Bayes solution of the smoothing problem," *IEEE Trans. Automat. Contr.*, vol. AC-26, pp. 558-561, Apr. 1981.

[2] W. W. Bledsoe and I. Browning, "Pattern recognition and reading by machine," in *Proc. Eastern Joint Comput. Conf.*, 1959, vol. 16, pp. 225-232.

[3] R. Bellman and S. Dreyfus, *Applied Dynamics Programming.* Princeton, NJ: Princeton Univ. Press, 1963.

[4] C. K. Chow, "A recognition method using neighbor dependence," *IEEE Trans. Comput.*, vol. C-11, pp. 683-690, 1962.

[5] R. Duda and P. Hart, "Experiments in the recognition of handprinted text: Part II–Context analysis," in *Proc. AFIPS Conf.*, 1968, vol. 33, pp. 1139-1149.

[6] L. D. Erman, F. Hayes Roth, V. Lesser, and R. Reddy, "The hearsay II speech-understanding system: Integrating knowledge to resolve uncertainty," *Comput. Surv.*, vol. 12, pp. 213-253, June 1980.

[7] J. A. Feldman and Y. Yakimovsky, "Decision theory and artificial intelligence: I. A semantics-based region analyze," *Art. Intell.*, vol. 5, pp. 349-371, 1974.

[8] G. Forney, "The Viterbi algorithm," *Proc. IEEE*, vol. 6, pp. 268-278, 1973.

[9] K. S. Fu, "Recent developments in pattern recognition," *IEEE Trans. Comput.*, vol. C-29, pp. 845-854, Oct. 1980.

[10] A. R. Hanson, E. M. Riseman, and E. Fisher, "Context in word recognition," *Pattern Recog.*, vol. 8, pp. 35-46, Jan. 1976.

[11] R. M. Haralick, J. C. Mohammed, and S. W. Zucker, "Compatibilities and the fixed points of arithmetic relaxation processes," *Comput. Graphics Image Processing*, vol. 13, pp. 242-256, 1980.

[12] R. M. Haralick and L. Watson, "A facet model for image data," *Comput. Graphics Image Processing*, vol. 15, pp. 113-129, 1981.

[13] L. N. Kanal, "Problem solving models and search strategies for pattern recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-1, no. 2, pp. 192-201, 1979.

[14] F. Lehan, personal communication, 1980.

[15] B. T. Lowerre and R. Reddy, "The HARPY speech understanding system," *Trends in Speech Understanding*, Lea, Ed. Englewood Cliffs, NJ: Prentice-Hall, 1980.

[16] M. Nagao and T. Matsuyama, "Edge preserving smoothing," *Comput. Graphics Image Processing*, vol. 9, pp. 394-407, 1979.

[17] G. Nagy, "State of the art in pattern recognition," *Proc. IEEE*, vol. 56, pp. 836-854, May 1968.

[18] S. Peleg, "A new probabilistic relaxation scheme," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-2, pp. 362-369, July 1980.

[19] J. Raviv, "Decision making in Markov chains applied to the problem of pattern recognition," *IEEE Trans. Inform. Theory*, vol. IT-3, pp. 536-551, Oct. 1962.

[20] E. M. Riseman and R. W. Ehrich, "Contextual word recognition using binary digrams," *IEEE Trans. Comput.*, vol. C-20, pp. 397-403, Apr. 1971.

[21] A. Rosenfeld, R. Hummel, and S. W. Zucker, "Scene labeling by relaxation operation," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, pp. 420-433, 1976.

[22] S. M. Rubin and R. Reddy, "The locus models of search and its use in image interpretation," *in Proc. Int. Joint. Conf. Art. Intell.*, 1977, pp. 590-595.

[23] R. Shinghal, D. Roseberg, and G. T. Toussaint, "A simplified heuristic version of a recursive bayes algorithm for using contest in test recognition," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-8, pp. 412-414, May 1978.

[24] G. Stockman, L. Kanal, and M. Kyle, "Structural pattern recognition of caratid pulse waves using a general waveform parsing system," *Commun. Ass. Comput. Mach.*, vol. 19, pp. 688-695, Dec. 1976.

[25] J. M. Tenenbaum and H. G. Barrow, "Experiments in interpretation guided segmentation," *Art. Intell.*, vol. 8, pp. 241-274, 1977.

[26] F. Tomita and S. Tsuji, "Extraction of multiple regions by smoothing in selected neighborhoods," *IEEE Trans. Syst., Man., Cybern.*, vol. SMC-7, pp. 107-109, Feb. 1977.

[27] G. T. Toussaint, "The use of context in pattern recognition," *Pattern Recog.*, vol. 10, no. 3, pp. 189-204, 1978.

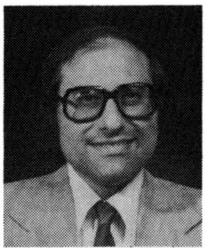
[28] D. Waltz, "Generating semantic descriptions from drawings of scenes with shadows," Artifical Intelligence Lab., Mass. Inst. Technol., Cambridge, MA, Rep. AI TR-271, 1972.

[29] S. W. Zucker, E. V. Krishnamurthy, and R. L. Haar, "Relaxation processes for scene labeling: Convergence, speed, and stability," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-8, pp. 41-48, 1978.

[30] S. W. Zucker and J. L. Mohammed, "Analysis of probabilistic relaxation labeling processes," presented at the Pattern Recognition and Image Processing Conf., Chicago, IL, June 1978.

[31] R. M. Haralick and L. G. Shapiro, "The consistent labeling problem I," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-1, pp. 178-184, 1979.

[32] —, "Structured descriptions and inexact matching," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-3, pp. 504-519, 1981.



Robert M. Haralick (M'69-SM'76) was born in Brooklyn, NY, on September 30, 1943. He received the B.S. and Ph.D. degrees from the University of Kansas, Lawrence, in 1966 and 1969, respectively.

He has worked at Autonetics and IBM. In 1965 he worked for the Center for Research, University of Kansas, as a Research Engineer and in 1969 he joined the faculty of the Department of Electrical Engineering, where he served as a Professor from 1975 to 1978. In 1979 he joined the faculty of the Department of Electrical Engineering, Virginia Polytechnic Institute and State University, Blacksburg, where he is now a Professor and Director of the Spatial Data Analysis Laboratory. He has done research in pattern recognition, multiimage processing, remote sensing, texture analysis, data compression, clustering, artificial intelligence, and general systems theory. He is responsible for the development of GIPSY (general image processing system), a multiimage processing package which runs on a minicomputer system.

Dr. Haralick is a member of the Association for Computing Machinery, Sigma Xi, the Pattern Recognition Society, and the Society for General Systems Research.

Correspondence

Application of the Conditional Population-Mixture Model to Image Segmentation

STANLEY L. SCLOVE

Abstract—**The problem of image segmentation is considered in the context of a mixture of probability distributions. The segments fall into classes. A probability distribution is associated with each class of segment. Parametric families of distributions are considered, a set of parameter values being associated with each class. With each observation is associated an unobservable label, indicating from which class the observation arose. Segmentation algorithms are obtained by applying a method of iterated maximum likelihood to the resulting likelihood function. A numerical example is given. Choice of the number of classes, using Akaike's information criterion (AIC) for model identification, is illustrated.**

Index Terms—**Cluster analysis, image processing, image segmentation, isodata procedure, *k*-means procedure, Mahalanobis distance, mixtures of distributions, multivariate statistical analysis, pattern recognition, pixel classification, relaxation methods.**

Manuscript received December 1, 1980; revised September 1, 1982. This work was supported by the Office of Naval Research under Contract N00014-80-C-0408 and by the Army Research Office under Contract DAAG29-82-K-0155.

The author is with the Department of Quantitative Methods, University of Illinois at Chicago, Chicago, IL 60680.

I. Introduction

A digital (i.e., numerical) image may be considered as a rectangular array of picture elements (pixels), indexed by (i, j). At each pixel the same p features are observed. We denote the features by

$$X_1, X_2, \cdots, X_p.$$

The vector of features is

$$\boldsymbol{X} = (X_1, X_2, \cdots, X_p).$$

The observed digital image is

$$\{\boldsymbol{x}_{ij}, i = 1, 2, \cdots, I, j = 1, 2, \cdots, J\},$$

where

$$\boldsymbol{x}_{ij} = (x_{1_{ij}}, x_{2ij}, \cdots, x_{pij})$$

is the vector of numerical values of the p features at pixel (i, j).

Examples

1) In color television, $p = 3$ colors, the pixels are the dots on the screen, and for pixel (i, j), x_{1ij} = red level, x_{2ij} = green level, and x_{3ij} = blue level.

(2) In Landsat data, $p = 4$ spectral channels, one in the green/yellow visible range, the second in the red visible range, and the other two in the near infrared range.

An *object* is a set of contiguous pixels which may be assumed to be members of a common class. One task of image processing is segmentation, grouping of pixels with a view toward identifying objects.

In this context the *conceptual model* is that the image is a