Propagating Possibilistic Temporal Constraints

Rasiah Loganantharaj
Automated Reasoning laboratory
The Center for Advanced Computer Studies
University of SouthWestern Louisiana
Lafayette, LA - 70504

1 Introduction

The notion of time plays an important role in any intelligent activities. Time is represented either implicitly or explicitly. We are interested in explicit representation of time. The popular approaches for such representation are based on points or intervals or a hybrid of both. Propositional temporal assertions are represented as relations among the points, or among the intervals. The indefinite information among either the points or the intervals are represented as disjunctions.

In real world, information is often incomplete, imprecise, uncertain and approximate. Temporal knowledge is not an exception to this reality. For example, consider the following information: John often drinks coffee during his breakfast. Sometimes, he drinks his coffee before the breakfast and drinks orange juice during his breakfast. There were few occasions he drank water during his breakfast and drank coffee after the breakfast. Mike talked to John over the phone while John was having coffee. Suppose, we are interested in finding out how Mik's telephone conversation was related to John's breakfast. From the given information, we have definite relation between the telephone call and John's coffee, but there is no information about the relationship between the coffee and the breakfast on that particular day. In the absence of such information, we can use John's habitual pattern to infer plausible relations.

Suppose, I_c and I_b respectively represent the interval over which John was having coffee, and John was having breakfast. Let I_t represents the interval over which Mike was having telephone conversation with John. In interval logic, the information is represented as I_t is during I_c, and I_c is before or during or after I_b. In such representation, the disjunctive relations do not provide any clue about which relation is highly probable than the others. Instead, the representation may let us believe that all the relations of a disjunction have equal probability, for example, having coffee before, during or after breakfast has equal probability. This is not what the original information tells us. Based on John's habit, having coffee during breakfast is much more probable than having coffee either before or after the breakfast. This issues have not been studied in temporal reasoning. In this paper we will provide a representation to specify uncertain information and to propagate them over temporal constraint network.

This paper is organized as the following. We introduce interval-based logic in Section 2. In Section 3, we describe the representation of uncertain temporal knowledge and its propagation. This paper is concluded by a summary and discussion in section 4.

2 Background on Interval Based System

Allen [1] has proposed an interval logic that uses time intervals as primitives. In this logic, the following seven relations and their inverses are defined to express the temporal relations between two intervals: before(after), meets(met-by), overlaps(overlapped-by), starts(started-by), during(contains), ends(ended-by), and equals. Here, the inverse relations are indicated within parentheses. Since the inverse of *equal* is same as itself, there are, in fact, only thirteen relations.

Temporal inferencing is performed by manipulating the network corresponding to the intervals. Each interval maps onto a node of a network called *temporal constraint network (TCN)*. A temporal relation, say R, from an interval, say I_i, to another interval, say I_j, is indicated by the label R_{ij} on the directed arc from I_i to I_j. Obviously, the label R_{ji} of the directed arc from I_j to I_i is the inverse relation of R_{ij}. If we have definite information about the relation I_i to I_j then R_{ij} will be a primitive interval relation, otherwise it will be disjunctions of two or more primitive interval relations.

Suppose the relation I_i to I_j (R_{ij}) and the relation I_j to I_k (R_{jk}) are given. The relation between I_i and I_k, constrained by R_{ij} and R_{jk}, is given by composing R_{ij} and R_{jk}. In general, R_{ij} and R_{jk} can be disjunctions of primitive relations, they are represented as: $R_{ij} = \{r_{ij}^1, r_{ij}^2, \ldots, r_{ij}^m\}$ and $R_{jk} = \{r_{jk}^1, r_{jk}^2, \ldots, r_{jk}^n\}$ where r_{ij} is one of the primitive relations defined in the system. The interval I_i is related to the interval I_k by the temporal relation given by the following expression:

$$R_{ij} \circ R_{jk} = \bigcup_{l=1\ldots m} \left\{ \bigcup_{p=1\ldots n} (r_{ij}^l \circ r_{jk}^p) \right\}$$

where $(r_{ij}^l \circ r_{jk}^p)$ is a composition (transitive relation) of r_{ij}^l and r_{jk}^p, and is obtained from the entry of the transitivity table [1] at the r_{ij}^i row and the r_{jk}^j column. Alternatively this could be written as

$$R_{ij} \circ R_{jk} = \{T(r_{ij}, r_{jk}) | r_{ij} \in R_{ij} \wedge r_{jk} \in R_{jk}\}$$

where $T(r_{ij}, r_{jk})$ is the value of Allen's look up composition table of row r_{ij} and column r_{jk}.

Temporal constraints are propagated to the rest of the network to obtain the minimal temporal network in which each label between a pair of intervals is minimal with respect to the given constraints. Vilain et al. [7] have shown that the problem of obtaining minimal labels for an interval-based temporal constraint network is NP-complete. Approximation algorithms, however, are available for temporal constraint propagation. Allen proposed an approximate algorithm that has an asymptotic time complexity of $O(N^3)$ where N is the number of intervals. His algorithm is an approximate one in the sense that it is not guaranteed to obtain the minimum relations, but it always find the superset of the minimal label. Since any set is a super set of a null set it is not very comforting because it is possible that global inconsistency may be hiding under 3-consistency.

3 Representation of Uncertainty

The problem of uncertainty is not new to AI problem solving. In many expert system applications, uncertainty have been studied under approximate reasoning. Mycin system [6, 2] has used certainty factor whose value varies from -1 to 1 through 0 to represent the confidence on an evidence or on a rule. The value 1 indicates the assertion is true while the value -1 indicates the evidence is false. 0 indicates no opinion on the evidence. The other values correspond to some mapping of the belief on the evidence onto the scale of -1 to 1. Prospector model [4, 5] uses probabilistic theory with Bayes' theorem and other approximation techniques to propagate evidences over causal network. Fuzzy logic [8] has also been used in expert systems to capture knowledge with fuzzy quantifiers such as 'very much', 'somewhat' etc. Other techniques have also been used to handle uncertainty in expert systems.

Let us look back at the same example presented in the introduction of this paper.

> John often drinks coffee during his breakfast. Sometimes he drinks his coffee before the breakfast and drinks orange juice during his breakfast. There were few occasions he drank water during his breakfast and drank coffee after the breakfast.

The statement has fuzzy quantifies indicating that the frequency of John having coffee during his breakfast is much higher than he is having coffee either before or after his breakfast. We should capture the fuzzy quantifiers into probabilistic measures in our temporal constraint representation such that the summation of the probabilities of the relations between a pair of intervals is equal to 1.

Let us represent this idea more formally. The relation R_{ij}, the relation I_i to I_j, is represented as $\{r_{ij}^1(w_1), r_{ij}^2(w_2), ..., r_{ij}^m(w_m)\}$ where $r_{ij}^l(w_l)$ is a primitive relation with its probability or the relative weight of w_l. Since each primitive relation between a pair of intervals is associated with a weight to represent the probability or the relative strength, the summation of the weights must be equal to 1 which we call *probabilistic condition* for the weights. That is, $\sum_{i=1}^m w_i = 1$. The boundary value of the weight 1 indicates that the relation is true while the value 0 indicates that the relation is false. Further, the inverse relation of R_{ij} will be the inverse of all the primitive relations of R_{ij} with the same weights. In the presence of new evidences, the probability values of the relations are modified to take account of the new evidences. That is, when the relations between a pair of intervals are modified or refined because of other constraining relations, the probability or the weights of the relations are adjusted to satisfy the probabilistic condition. This process is called *normalization*. Let us explain this with an example: Suppose (1) $R_{12} = \{b(0.6), O(0.3), d(0.1)\}$, (2) $R_{13} \circ R_{32} = \{b(0.4), o(0.5), m(0.1)\}$. The relation R_{12} is refined to $\{b(w_1), o(w_2)\}$. The weights w_1 and w_2 are computed using the intersection operation and then the weights are normalized such that $w_1^{'} + w_2^{'} = 1$.

When propagating constraints with probabilistic weights, we may need to define such as union, intersection, composition and normalization operations which are used in propagation.

Union operation:

$$R_{ij} \cup_p R_{ij}^{'} = \{r(w) | r(w_{ij}) \in R_i \wedge r(w_{ij}^{'}) \in R_j \wedge w = max(w_{ij}, w_{ij}^{'})\}$$

Intersection Operation:

$$R_{ij} \cap_p R_{ij}^{'} = \{r(w) | r(w_{ij}) \in R_i \wedge r(w_{ij}^{'}) \in R_j \wedge w = min(w_{ij}, w_{ij}^{'})\}$$

Composition Operation:

$$R_{ij} \odot R_{jk} = \{\cup_p r(w) | r_{ij}^l(w_l) \in R_{ij} \wedge r_{jk}^m(w_m) \in R_{jk} \wedge r = T(r_{ij}^l, r_{jk}^m) \wedge w = min(w_l, w_m)\}$$

Normalization operation: Suppose the label R_{ij} takes the following form after refinement $\{r_{ij}^1(w_1), r_{ij}^2(w_2), ..., r_{ij}^m(w_m)\}$. Let $w = \sum_{i=1}^m w_i$. After normalization operation the label becomes $\{r_{ij}^1(w_1^{'}), r_{ij}^2(w_2^{'}), ..., r_{ij}^m(w_m^{'})\}$ where $w_i^{'} = w_i / w$ which ensures that $\sum_{i=1}^m w_i^{'} = 1$

In this approach we use possibilistic ways of combining the constraints as has been used in many expert systems under uncertainty.

3.1 Temporal Propagation

A temporal constraint network (TCN) is constructed from the given temporal assertion such that each node of the constraint network represents each interval of the temporal assertions. The labels on each arc of a TCN corresponds to the relations between the corresponding intervals. Further the summation of the weights of each component in each arc should be equal to 1. If no constraint is specified between a pair of intervals it will take a universal constraint[1] as label in the TCN and it will not be used for the purpose of propagation. When propagating a constraint, other label of the network may get updated to a subset of its label. The process of updating of a label as a result of propagating a constraint is called *label refinement.* The label refinement takes the following forms: (1) The weights of the labels of an arc get changed, or (2) the primitive relations of a label get reduced. In either case normalization operation is performed to ensure the summation of the weights adds to 1.

Suppose we are propagating the label R_{ij} of the arc $< I, J >$ to the arc $< I, K >$ of the triangle IJK. Let the labels of the arcs $< J, K >$ and $< I, K >$ are respectively R_{jk} and R_{ik}. The new label of the arc $< I, K >$ is computed as $R_{ik} \cap \{R_{ij} \circ R_{jk}\}$. When the new relation is not null the weights on the label are normalized.

3.2 Temporal Constraints Propagation Algorithm for uncertain constraints

This is an extension of Allen's propagation algorithm. The algorithm uses a first in first out (FIFO) queue to maintain the constraints that need to be propagated. Initially all the pairs of constrained intervals are placed into the queue. The propagation of constraints is initiated by removing an arc, say $< I_i, I_j >$, from the queue and checking whether the relation between I_i to I_j can constrain the relations on all the arcs incident to either I_i or I_j except the arc $< I_i\ I_j >$. When a new relation is constrained, that is, the old label is modified, the arc (the pair of intervals related by this relation) is placed in the queue. The main propagation algorithm is described in Figure 1. In this algorithm, we use the notation R_{ij} to denote the label of the arc i to j.

When we omit the weights on the labels and use the union, the intersection operations of set, and the composition operation of temporal logic, the algorithm becomes identical to the one of Allen's [1]. One may expect the asymptotic complexity to be $O(N^3)$ where N is the number of nodes of the TCN. Intuitively one may make a conclusion that this algorithm will also converge in $O(N^3)$ time. This may be misleading because we have not yet considered the instability effect of the algorithm due to it normalization operation. An arc is placed back in the queue when either the number of primitive relations of the label is reduced or the weight of the primitive relation of the label is modified even when the label remains unchanged. An arc may be placed in the queue at most 12 times as a result of the disjunctive relations being refined one at a time till it becomes a singleton relation. On the other hand, the number of times an arc is placed in the queue due to the change of weight depends on the following parameters: (1) the resolution of the weights (the number of decimal places that counts) and (2) the error bound we are prepared to tolerate. A complete study on these issues can be found in one of our report [3].

Let us consider an example. The probability of 'having coffee' before breakfast is 0.15, during breakfast is 0.8 and after breakfast is 0.05. The probability of 'having a coffee' overlapping 'reading morning newspaper' is 0.8 and 'having coffee' meets 'reading the newspaper' is 0.2. Suppose we want to find the relation between having breakfast and reading newspaper. Let us propagate the constraints and find out how the labels gets refined. Suppose I_b, I_c, I_r respectively represent the intervals of having breakfast, coffee and reading newspaper. Using the notations defined in this paper we can define the labels of the initial TCN as following.

Initial TCN
$R_{cb} = \{b(0.15), d(0.8), a(0.05)\}$
$R_{cr} = \{o(0.8), m(0.2)\}$
$R_{br} =$ to be computed
TCN after propagating R_{bc} to $< I_b, I_r >$
$R_{cb} = \{b(0.15), d(0.8), a(0.05)\}$
$R_{cr} = \{o(0.8), m(0.2)\}$
$R_{br} = \{o(.8), oi(.15), d(.15), di(.8), f(.15), fi(.8), b(.05), a(.15)\}$
After normalizing R_{br}
$R_{br} = \{o(.26), oi(.049), d(.049), di(.26), f(.049), fi(.26), b(.024), a(.049)\}$
This is also the 3-consistent TCN labels.

4 Summary and Discussion

In many real world applications we are faced with the information that is incomplete, indefinite, imprecise and uncertain. When explicit time was used for temporal reasoning, indefinite information is accommodated as disjunctions. For example, drinking coffee is either before, during or after the breakfast. The disjunctive information implicitly assume equal probability of occurrence even though exactly one can be true between a pair of intervals (points). In such representation we fail to distinguish or differentiate the highly probable one from the remotely possible one. According to our example, having coffee during breakfast is highly probable than having coffee before

[1] a universal constraint is the weakest constraint and thus it is a disjunctions of all the primitive relations of the time model

```
procedure propagate1()
While queue is not empty Do
{    get next < i, j > from the queue
     /* propagation begins here */
     For ( ( k ∈ set_of_intervals ) ∧k ≠ i ∧ k ≠ j )Do
     {    temp ← R_ik ∩_p (R_ij ⊙ R_jk)
          If temp is null Then signal contradiction and Exit
          Normalize temp
          If R_ik ≠ temp Then
          {    place < i, k > on queue
               R_ik ← temp }
          temp ← R_kj ∩_p (R_ki ⊙ R_ij)
          If temp is null Then signal contradiction and Exit
          Normalize temp
          If R_kj ≠ temp Then
          {    place < k, j > on queue
               R_kj ← temp }
     }
}
```

Figure 1: propagation algorithm

or after a breakfast.

In this paper we have proposed a formalism to represent temporal constraints with the associated probabilistic weights and use them to propagate to the rest of the network to obtain 3-consistency. We have extended Allen's algorithm to handle probabilistic relations among intervals. The operations we have defined for the labels with weights are applicable to both the points and the intervals. Therefore, our formalism will be applicable to both the points and the intervals.

Acknowledgement

This research was partially supported by a grant from Louisiana Education Quality Support Fund (LEQSF).

References

[1] J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of ACM*, 26(11):832–843, 1983.

[2] B. G. Buchanan and E. H. Shortliffe Eds. *Rule-Based Expert Systems.* Addision-Wesley, 1984.

[3] R. Loganantharaj. Complexity issues of possibilistic temporal reasoning. *preperation*, 1994.

[4] J. Gaschnig R. O. Duda and P. E. Hart. Model design in the prospector consultant system for mineral exploration. In D. Michie, editor, *Expert Systems in the Microelectronics Age.* Edinburgh University Press, 1979.

[5] P. E. Hart R. O. Duda and N. J. Nilsson. Subjective bayesian methods for rule-based inference systems. In *Proceedings of the AFIPS National Computer Conference*, volume 45, 1976.

[6] E. H. Shortliffe. *Computer Based Medical Consultations:MYCIN.* Elsvier, 1976.

[7] M. Vilain and H. Kautz. Constraint propagation algorithm for temporal reasoning. In *Proceedings of AAAI-86*, pages 377–382, 1986.

[8] L. A. Zadeh. Making computers think like people. *IEEE Spectrum*, pages 26–32, August 1984.