Eliciting and Combining Influence Diagrams: Tying Many Bowties Together

Krol Kevin Mathias*, **Cynthia Isenhour**†, **Alex Dekhtyar***, **Judy Goldsmith***, **Beth Goldstein**‡
{kevin9,Cynthia.Isenhour}@uky.edu, {dekhtyar,goldsmit}@cs.uky.edu, bethg@uky.edu
University of Kentucky, Lexington, KY

Abstract

The work presented here is part of a research project to develop decision-support software for case managers in the Kentucky social welfare system. Welfare case managers help their clients plan participation in activities such as volunteer work, job readiness programs, substance abuse counselling, or study in high school or college, for example. The case manager's advice is guided by her perception of how different client characteristics, as well as the client's history prior to and within the program, may affect the client's probability of success in the proposed activities.

This paper focuses on knowledge transfer from welfare domain experts to formal representation of the welfare domain. In the process of modelling decision-making in the Kentucky welfare system, we have elicited influence diagrams from multiple welfare case managers. We have developed a distinctive elicitation procedure and influence diagram format, which we refer to as a "bowtie action fragment". We present the format, the procedure, and a discussion of methods for combining influence diagrams from multiple experts into one consistent and size-bounded diagram.

1 Introduction

Stochastic planning with Markov Decision Processes (MDPs) has been a productive research area recently (space limitations prohibit a comprehensive survey.) While planning with MDPs is tractable, tractability requires explicit instantiation of all states, which is infeasible for large complex planning problems. Dynamic Bayes Networks (DBNs) can be used for factored representation of MDPs. DBNs are small in size, but exact planning algorithms become intractable [9]. Good heuristics, however, make planning with DBNs attractive even when the planning problems are large.

In [3] we have discussed the problem of decision-theoretic planning with uncertainty and constraints. The problem involves four key components: (a) an agent called "advisor" is charged with suggesting a plan of future activities for another agent, called "advisee"; (b) the advisee's goals are based on successful completion of activities, and success in completing an activity *stochastically* depends on personal characteristics, as well as on successful completion of certain prior activities; (c) proposed plans must satisfy existing constraints; (d) proposed plans must respect, as much as possible, the advisee's preferences.

The work of case managers with welfare-to-work clients is an application of decision-theoretic planning with uncertainty and constraints. The Welfare-to-Work scenario, the subject of this paper, considers the advising done by welfare case managers to their clients. Since the implementation of welfare reform in 1996, neoliberal policies have reshaped the welfare system from one in which cash grants were the primary vehicle of assistance, to one in which the primary tool for assisting welfare clients has taken the form of social services such as employment training, counselling, or educational services. This shift in policy has translated into a significant increase in case manager work load and responsibility. In addition to processing applications, payments and other paperwork for a nearly unmanageable case load, case managers are expected to act as advisors and information brokers, informing their clients about the myriad support services available. Without adequate information support, however, case managers often find it difficult to serve their clients

* Department of Computer Science.
† Department of Anthropology.
‡ Department of Educational Policy.

well. Thus the welfare domain presents a significant opportunity for the application of decision theoretic planning.

Welfare-to-work case managers[1] provide advice and place participants (called "clients") in training, counselling, educational or work programs designed to make the client employable, and, eventually, employed. Case mangers also control access to payment for basic necessities such as housing, food, health care, childcare, and transportation. However, these programs must be consistent with the client's abilities and locally available programs. Case manager advice is therefore guided by: federal and state welfare policies which specify what the client must do to stay eligible for benefits; the availability of local service programs, the case manager's assessment of the client's goals, preferences, abilities, skills, and limitations; and the case manager's perception of how these characteristics and past performance in KTAP activities may affect the probability of the client's success in the proposed activities.

In this paper we describe our work on building the Welfare-to-Work domain model for the purpose of decision support planning. The domain model consists of several components: (a) the stochastic model of client performance in KTAP-supported activities, (b) the model of a client, (c) the rules and constraints that apply to the program and (d) the means for expressing client goals and preferences. Our present study concentrates on the first two components. Here, we describe:

• Our approach to building stochastic models for planning (Section 2).
• The process of elicitation we have employed in working with the KTAP case managers to obtain their opinions about the model components (Section 3).
• The methods we employed to combine the expertise of different case managers in order to produce a single unified model (Section 4).
• The results of a study of these methods (Section 5).

2 A Bayes Net Model for Advising Applications

2.1 Basics

A *Markov Decision Process* (MDP) is defined as a 4-tuple (S, A, R, P) where S is a set of states; A is a set of actions; R is a reward function $R : S \mapsto \mathcal{R}$, such that $R(s)$ represents the reward obtained by the agent in state s; and P is a transition model where $P_a(s' \mid s)$ represents the probability of going from state s to state s' with action a [6]. In factored MDPs, the set of states is described by a set of random variables. This results in a large and inefficient matrix representation of transition probabilities. The transitions and reward function are captured concisely by *dynamic Bayesian networks* (DBNs) [2]. The transition graph of a DBN is a two layered directed acyclic graph whose nodes are $\{X_1, X_2, \ldots, X_m, X_1', X_2', \ldots, X_m'\}$ where X_i denotes the random variable X_i at time t and X_i' denotes the random variable X_i at time $t + 1$. Figure 1 represents the DBN model for the action "Volunteer Placement (VOP)". In this action, the client participates in volunteer work relevant to the client's job goals.

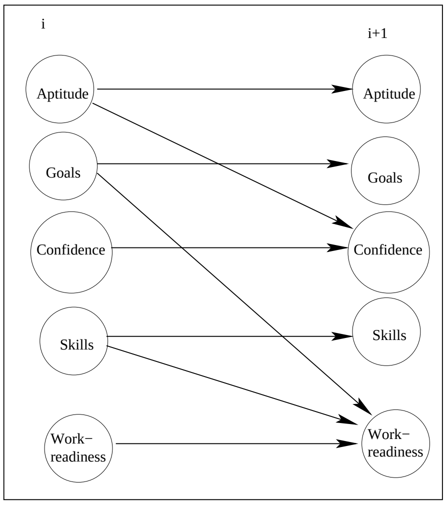

Figure 1: DBN model for the action "Volunteer Placement".

Qualitative probabilistic networks(QPNs) are qualitative abstractions of their quantitative counterparts. A QPN is defined as a pair $G = (V, Q)$, where V is the set of variables or nodes in the graph and Q is the set of qualitative influences [10]. Qualitative influences define the sign of direct influence between two variables and correspond to an arc in a belief network [4].

Definition (qualitative influence) [4]:A random variable a positively influences a random variable c, written $S^+(a, c)$, iff for all values $a_1 > a_2$, c_0, and x, which is the set of all c's predecessors other than a, $\mathrm{P}(c \geq c_0 \mid a_1 x) \geq \mathrm{P}(c \geq c_0 \mid a_2 x)$. This definition expresses the fact that increasing the value of a, makes higher values of c more probable. Negative qualitative influence, S^-, and zero qualitative influence, S^0, are defined analogously by substituting $\geq$ by $\leq$ and $=$ respectively. If a qualitative property is not "+", "-", or 0 then it is represented as "?" by default (non-monotonic or unknown relation) [4]. Specifying signs is known to require considerably less effort from do-

[1]For instance those working in the Kentucky Temporary Assistance Program (KTAP), the "experts" in this study.

main experts than specifying numbers [4].

2.2 The Bowtie Model

In this section we describe the *bowtie* models we have elected to elicit, and explain how we arrived to the specific shape of the models.

Let $D = \langle d_1, \ldots d_m \rangle$ be a list of fluents we call *client characteristics* (Figure 3). Each fluent d_i takes values from a finite ordinal domain[2] $dom(d_i) = \mathcal{D}_i$. A *client state* is a vector $s = (s_1, \ldots, s_m)$, where $s_i \in \mathcal{D}_i \cup \{\perp\}$. If $s_i = \perp$ for some $1 \leq i \leq m$, we say that the value of client characteristic d_i is undefined or unknown in state s.

A *plan* for a client is a mapping from client states into collections of *actions*. The list $\mathcal{A} = \{A_1, \ldots, A_n\}$ forms the universe of actions, and a plan π is a mapping $\mathcal{D}_1 \cup \{\perp\} \times \ldots \times \mathcal{D}_m \cup \{\perp\} \longrightarrow 2^{\mathcal{A}}$. To properly construct plans, we must identify the influences of client characteristics on actions. We model these influences stochastically. That is, we believe that *client performance in a specific action depends stochastically on client characteristics*, or, in other words, *values of client characteristics can be used as predictors for client performance in a specific action.*

We use a representation model we call a *bowtie action fragment* instead of DBN representations of factored MDPs. A *bowtie model* for an action A is a tuple $\mathcal{B}_A = \langle \mathcal{I} \cup \mathcal{O} \cup \{s_A\}, E_{\mathcal{I}} \cup E_{\mathcal{O}}, w, inf \rangle$, where $\mathcal{I}$ and $\mathcal{O}$ are subsets of D called *inputs* and *outputs* respectively, and s_A is a node we call a *success node of action A*, together form the set of nodes for the bowtie model; $E_{\mathcal{I}} = \{(x, s_A) | x \in \mathcal{I}\}$ and $E_{\mathcal{O}} = \{(s_A, y) | y \in \mathcal{O}\}$ are the edges of the model; $w : E_{\mathcal{I}} \rightarrow \mathcal{R}$ is *input weight* function and $inf : E_{\mathcal{O}} \rightarrow \{+, -, 0, ?\}$ is the *output influence* function.

There are two key reasons why bowtie models are used here instead of DBNs. First, in advising scenarios, each action has a specific result (a success/failure or taking in, etc.) associated with it. This result is not part of the list of input client characteristics but it is part of the problem domain. In particular, information about it must be maintained throughout the process. We represent the success of an action using a special *success node* in our graphical models. The domain of each success node is determined by the action itself. It can be binary: *success/failure*, but it can also have more values, e.g., a GPA from {less than 2.0, between 2.0 and 3.0, between 3.0 and 4.0, 4.0} in case of action "High School".

[2]The specific properties of the domains are not necessary for the study described in this paper. However, they are used later in building the model.

The second reason for using the bowtie model is that all information about individual action fragments in our study is elicited from the experts, the case managers, whose goals and objectives are tied directly to helping a client succeed in a given action. Each small success is seen by case managers as having a profound positive impact on the client's state, and therefore, ability to succeed in future actions. During our elicitation process we have determined that the bowtie model makes sense to the case managers since the action in the welfare system mediates the transformation between two client states. In Section 3 we describe the elicitation process in more detail.

Informally, the *bowtie* model consists of a set of input nodes linked to a central action success node. The success node, in return, is linked to the set of output nodes. The *bowtie* model is similar to the QPN model except that in the *bowtie* model, 'weight' is associated with each of the influence factors.

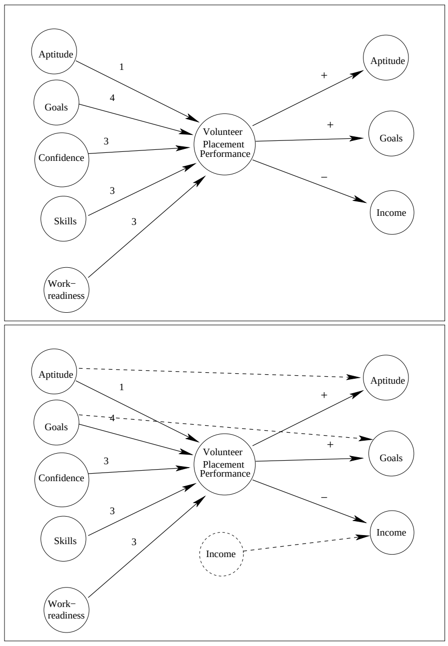

Figure 2: Bowtie model for the action fragment "Volunteer Placement (VOP)" and the actual influence diagram it represents.

The bowtie model makes a convenient picture of the action fragments. However, some influences in the fragment are not explicitly represented. In particular, *each output node in an action fragment is influenced by two nodes: the action success node and the node representing the value of this characteristic prior to the action being taken.* The first influence is represented explicitly in the bowtie diagram, but the second is implicit. Figure 2 shows an example of a bowtie model for the action fragment "Volunteer Placement (VOP)", and the actual influence diagram represented by this fragment. In the actual diagram, each output node has two parents. In this case, we needed to add a node representing a client's income prior to taking the action.

3 Elicitation of Model

Name	Abbrevn.	Name	Abbrevn.
Job & Post-Sec. Edu.	JSE	English Second Language	ESL
Short Term Training	JST	Volunteer Placement	VOP
Voc. Rehabilitation	RHB	Job Readiness Class	JRA
On Job Training	OJT	Literacy Training	LIT
College	COL	Community Service	COM
Adult Basic Edu.	ABE	Community College	CCO
High School	HSC	Voc. School	VOC
Group Job Search	GJS		

Table 1: The list of 15 elicited actions.

The elicitation of case manager knowledge was done in three stages: (i) a manual pilot study, (ii) design and implementation of elicitation software, and (iii) software-directed elicitation. We briefly outline these stages below.

First, our team of social scientists conducted a pilot study to test our elicitation methodology. In the pilot study, welfare case managers were asked to free list client characteristics which would affect the client's likelihood of success in the action "Get GED", which includes attending preparatory classes and eventually taking the General Education Development (GED) test.[3]

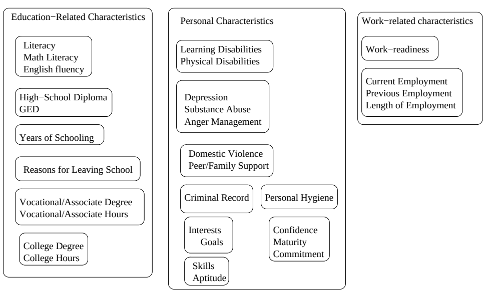

Figure 3: Client characteristics used in the experiments.

The nearly 200 characteristics free-listed as affecting the client's likelihood of success in the action "Get GED" were then evaluated and ranked by small groups. The five client characteristics cited by case managers as most important were weighted, thus forming the input structure of the bowtie model for the "Get GED" action fragment. The list of client characteristics generated in the pilot study and extracted from previous expert interviews were then merged into a more manageable list of approximately 50 variables. These variables, in turn, we loosely categorized as education-related, work-related and personal characteristics. Within each category, we have tentatively outlined subcategories. The list of most often considered characteristics and their classification into categories and subcategories is shown in Figure 3. The elicitation methodology was verified and the experiment was replicated for the additional fifteen actions in which a welfare client can participate (Table 1). A High Level Elicitor (HELL), the special-purpose elicitation software developed by our team for eliciting bowtie action fragments, was utilized to replicate this methodology.

The High Level Elicitor was designed to obtain bowtie fragments by eliciting the input of 18 participating case managers. Each manager was assigned 5 of the 15 actions from Table 1. For each of their five actions, they were asked to complete a three-step process. First, they were asked to pick, from a list of approximately 50, the five characteristics which were most important for success in the given action (Figure 4). Second, upon selecting the five most important characteristics for a given action, case managers were then asked to assign a weight to each characteristic so we could determine the relative importance of the five selected characteristics.[4] Finally, output nodes were elicited for each action. Experts were asked to indicate which of the client characteristics were most likely to change positively or negatively as the result of successful completion of the assigned action. Thus for example, experts reported that, upon successful completion of Volunteer Placement (VOP), the client would experience an improvement in Aptitude and Goals.

The case managers' answers were collected and stored for further processing. In the next section we discuss the methods used to combine the action fragments elicited from different experts.

4 Fusion

Using the procedure described in Section 3 we obtained multiple action fragments for each of the 15 actions. The next step was to produce for each activity a single action fragment, that incorporated the expertise of multiple case managers. The challenge we faced was to

[3] We have elicited a total of 16 actions from case managers. Out of these, "Get GED" was used in the pilot study as a "training example" and then excluded from further elicitation procedures. The remaining 15 actions, shown in Table 1 were used in the remainder of the study.

[4] Scale of 1 to 4 was used.

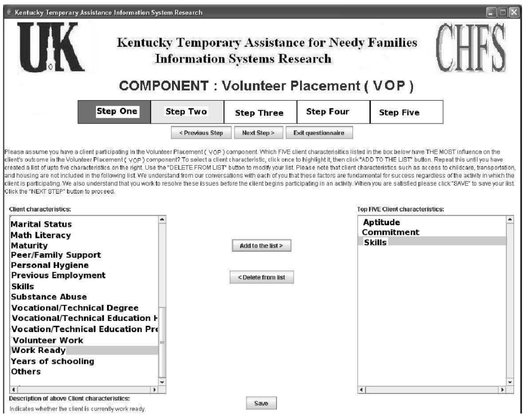

Figure 4: Step 1 of the elicitation process for the action fragment "Volunteer Placement (VOP)".

restrict the number of input nodes in order to keep the number of parent nodes of the success (central) node to a minimum and to incorporate maximum information provided by the experts in the final action fragment. We observed that the more parents the success node had, the harder it was to construct the conditional probability table (CPT), and the less reliable such a CPT became. Because of this, we *restricted the size of the set of input nodes in a bowtie action fragment to **five***. We now, address the problem of selecting the appropriate five client characteristics from a significantly larger pool provided by multiple experts.

More formally, we are solving the following problem. Given an action A and a set $\{\mathcal{B}_A^1, \ldots \mathcal{B}_A^k\}$ of bowtie action fragments for A, we need to construct a single bowtie action fragment $\mathcal{B}_A^*$ which has the following properties: (i) the number of parents of the action success node is no more than five[5] and (ii) as much information from $\{\mathcal{B}_A^1, \ldots \mathcal{B}_A^k\}$ as possible is used. We call this the action fragment fusion problem.

We consider two ways of reducing the number of input nodes in the bowtie model. First, we consider *node removal*, i.e., straightforward pruning of the set of input nodes. Our second operation, *merge* is more complex. A merge of two or more input nodes replaces them in the bowtie model with a single node. Optionally, the replaced nodes may be left in the model as parents of the new node. Schematically, the operations of node removal and merge are shown in Figure 5.

Node removal. Decisions on which nodes are removed can be made in different ways. For each input node x, we construct the vector of *mentions*: $m^n = (m_1^n, \ldots, m_k^n)$, $m_i^n = 1$ if $n \in \mathcal{I}$ for $\mathcal{B}_A^i$, otherwise, $m_i^n = 0$. We also construct the vector of weights: $w^n = (w_1^n, \ldots, w_k^n)$, $w_i^n = 0$ if $m_i^n = 0$, otherwise $w_i^n = w((n, s_A))$ for the w from $\mathcal{B}_A^i$. Finally, we consider the vector of expert experiences $E = (e_1, \ldots, e_k)$, where e_i is some measure of experience assigned to expert i (the source of the fragment $\mathcal{B}_A^i$). To simplify notation, we let $N = \{n_1, n_2, \ldots, n_m\}$ denote the complete set of input nodes specified by k experts for an action fragment.

[5]In our work, action fragments $\mathcal{B}_A^i$ had up to five input nodes as well.

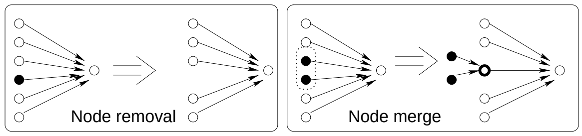

Figure 5: Operations of node removal and node merge.

All node removal operations considered here work as follows. The combined list of input nodes is ordered according to some ordering criteria. The top five nodes in this ordering are retained, all others removed. We have 10 methods to order the nodes, described in Table 3. The measures used for the different ordering methods are described in Table 2.

Measure	Expression
no. of mentions (*nom*)	$\sum_{j=1}^{k}(m_j^{n_i})$
total importance (*timp*)	$\sum_{j=1}^{k}(w_j^{n_i})$
experience-based wt. (*we*)	$\sum_{j=1}^{k}(w_j^{n_i} * e_j) / \sum_{j=1}^{k} e_j$
relative expertise (*re*)	$\sum_{j=1}^{k}(e_j * m_j^{n_i}) / \sum_{j=1}^{k} e_j$

Table 2: Measures used for the ordering methods.

Method	Primary sort	Secondary sort
$S1$	*nom*	*timp*
$S2$	*timp*	*nom*
$S3$	*nom*	*we*
$S4$	*we*	*nom*
$S5$	*re*	*timp*
$S6$	*timp*	*re*
$S7$	*re*	*we*
$S8$	*we*	*re*
$S9$	*nom* * *timp*	-
$S10$	*we* * *nom*	-

Table 3: The 10 Ordering methods.

Node Merge. Two or more client characteristics can be merged into a single node if two conditions are satisfied:
1. Proximity: the characteristics are similar in their meaning.
2. Use: the pattern of mentions of the characteristics by the experts allows us to hypothesize that they can be merged.

We address each issue in turn. We considered client characteristics to be *similar* if they share the same category and subcategory in the classification scheme of Figure 3. In order to merge nodes, it is not enough to simply discover that similar client characteristics had been used by experts. One must pay attention to *how* these characteristics had been used. In this paper we consider one such pattern of use, which we call *disjoint use*. A set of *similar client characteristics* exhibits the *pattern of disjoint use* if at most 20% of experts who mention one of these nodes, mention more than one. That is, the vast majority of experts uses only one of the similar characteristics as the influence on the action success node.

If such pattern is observed, we can hypothesize that experts were considering the same latent characteristic (the category that these nodes belong to), but selected different characteristics to serve as the representative. An example of such use is the case of an action A where three experts indicate that Disability has an important influence on it, while two experts state that Learning Disability has an important influence.

In this work we use a straightforward algorithm that scans the input action fragments and searches for the patterns of disjoint use for each group of similar nodes. If it succeeds, it replaces the group of similar nodes with a node representing their supercategory as the parent of the action success node. The original nodes then become the parents of the new node. The vector of mentions for the new node is constructed as the disjunction of the vectors of mentions for its parent nodes. The weights are averaged: the weight assigned to the new node by an expert is the average of the weights assigned to the parent nodes (for the disjoint use pattern, only few experts will have mentioned more than one parent). After all possible mergers are performed, we use one of the methods **S1**–**S10** to determine the top five input nodes.

Coverage. The different fragments for a particular action obtained from merging and sorting are validated against the *average coverage* measure. The coverage measure (CC) is evaluated for every expert (for a particular action). Coverage for expert j is $CC_j = \sum_{i=1}^{n}(m_j^{n_i} * w^{n_i})/\sum_{i=1}^{n}(w_j^{n_i})$. The average coverage ACC is then computed as $ACC = \left(\sum_{j=1}^{k} CC_j\right)/k$.

For a given action fragment the average coverage is calculated for each method by taking the average of the coverage criteria of all experts that elicited the given action fragment. High average coverage for a given action fragment means most of the important, or highly weighted, input nodes elicited from the experts for that action fragment are incorporated into the final fused model.

If we use only the ordering criteria for choosing the top x nodes for the final model representing the action fragment, we run into the problem of throwing out a lot of client characteristics (input nodes) elicited by experts. This reduces the average coverage for each action fragment. With merging, we are able to increase coverage.

5 Experiments

We have applied the fusion methods discussed in Section 4 to the action fragments for 15 actions (Table 1) elicited from the case managers. We looked at two groups of fusion methods: using only sorting and node removal, and using node merge with subsequent sorting and node removal. We now outline the results obtained. For each action fragment we used all ten methods **S1**–**S10** to sort the input nodes obtained from all experts, and select the top five. In the second run, we first used the merging criterion described in Section 4 to merge some of the nodes and reweigh the edges, and then used the same ten sorting methods.

For each of the two runs (which we refer to as "sort only" and "merge+sort"), all ten methods agreed on the top five input nodes for **nine** actions (the actual list of nine actions was somewhat different for each run). In the sort-only run, the remaining **six** actions were split in half: for **three**, we observed **two** different fragments, while for the remaining **three**, **three** different fragments were generated. In the merge+sort run, all **six** remaining actions yielded **two** fragments each. *In all our experiments that yielded more than one fragment, the fragments differed by exactly one input node.*

Figure 6 shows the four fragments generated for the action "Vocational Rehabilitation (RHB)" by sorting-only (top pair) and merging+sorting (bottom pair) methods. As seen from the picture, the top three inputs (Commitment, Literacy Level and Goals) remain intact in all four fragments. All sorting-only methods also agree on the fourth input, Learning Disabilities. The difference comes at the fifth input, where some methods favor Disabilities and others, Aptitude. Merge+sort methods generate immediate savings of one slot by merging together Learning Disabilities and Disabilities. In addition, in some merge+sort methods, Aptitude loses to the combo node obtained by merging Domestic Violence and Peer/Family Support.

Figures 7 and 8 show the correspondence between different methods and the fragments produced by them for each action. For each action, boxes of the same shade for two methods mean that the methods pro-

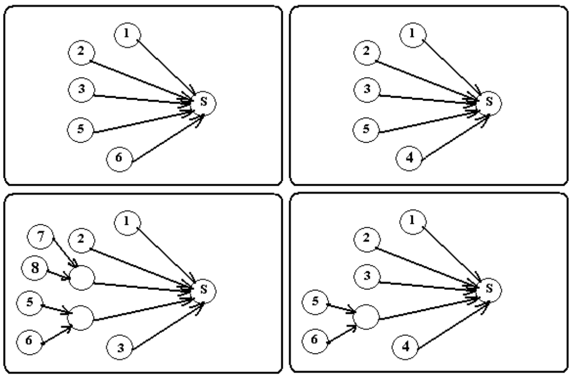

Figure 6: Fragments generated for the action "Vocational Rehabilitation (RHB)".

duced the same fragment. We note that expert experience, when used, had the highest chance of altering the list of top five inputs. In particular, on four out of six actions with multiple fragments returned, methods **S1, S2**, based solely on weights and frequency of mention produced different output than methods **S3, S4, S5, S7, S8**, which included, among their sorting criteria, measures weighted by expert experience. When merge+sort is considered, **S4, S7, S8** produce the second fragment in all six observed cases, while **S5** produces it in five cases and **S3** in three cases. In addition, method **S10** diverged from **S1, S2** on its output in four cases. In general we see a fairly stable picture. Methods that did not take experience into account tended to have the highest chance of providing different sets of five input nodes than those that did use experience vectors.

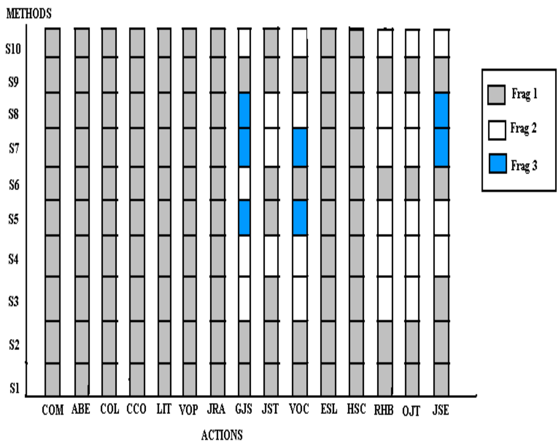

Figure 7: The figure shows which of the sorting methods yielded which fragments for each action.

Table 4 describes the average coverage for all fragments and actions. Numbers in boldface (for the merge+sort run) correspond to action fragments where a merge operation was used. We note, first, for 11 out of 15 actions, more than one action fragment was generated during both sorting and merging+sorting runs. Only four actions yield the same fragment for all twenty methods. Second, the merge operation for the pattern of disjoint mention tends to raise average coverage score, sometimes by as much as 10%.

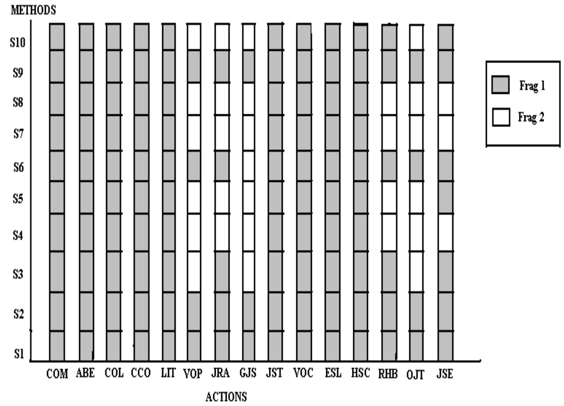

Figure 8: The figure shows which of the merge+sort methods yielded which fragments for each action.

Based on expert opinion, 12 out of the 16 fragments that applied the merge criterion were considered invalid (INV). In the case of actions ABE and LIT that are heavily centered on learning, the expert opinion did not agree on the merger of Disabilities and Learning Disabilities. In contrast this merge was considered valid for the actions VOP, JST, GJS, VOC and RHB that are not centered on learning. Expert opinion also disagreed with the merger of Commitment, Maturity, and Confidence, which made the fragments for actions VOP and GJS invalid. Finally, the merger of Aptitude and Skills that made the fragments for actions JRA, VOC, ESL, and JSE were aksi invalidated. However, the merger of College Degree and College Hours, Previous Employment and Current Employment were considered valid for actions JSE and OJT respectively. The first fragment for action RHB was considered invalid due to the merger of Domestic Violence and Peer/Family Support.

6 Related Work

There has been extensive research done in aggregating beliefs from different experts/sources of information to form models. Maynard-Zhang and Lehmann [8] provide a representation based on the class of modular, transitive relations for collective beliefs. They define an operator that combines belief states of informant

Action	sort-only			merge+sort	
	Frag1	Frag2	Frag3	Frag1	Frag 2
COM	0.72	-	-	0.72	-
ABE	0.63	-	-	**0.67 (INV)**	-
COL	0.67	-	-	0.67	-
CCO	0.67	-	-	0.67	-
LIT	0.65	-	-	**0.73 (INV)**	-
VOP	0.56	-	-	**0.6 (INV)**	**0.58 (INV)**
JRA	0.63	-	-	**0.65 (INV)**	0.64
GJS	0.524	0.535	0.52	**0.56 (INV)**	**0.57 (INV)**
JST	0.647	0.642	-	**0.675**	-
VOC	0.508	0.497	0.45	**0.59 (INV)**	-
ESL	0.571	-	-	**0.573 (INV)**	-
RHB	0.54	0.53	-	**0.66 (INV)**	**0.62**
OJT	0.56	0.55	-	**0.58**	**0.57**
JSE	0.45	0.43	0.37	**0.54 (INV)**	**0.48 (INV)**
HSC	0.623	-	-	0.623	-

Table 4: Average coverage for different combined fragments for each action.

sources preordered by credibility. In [5], the expert opinions are aggregated using weighted aggregation whereas [1] describes methods of scoring and aggregating expert opinions. Laskey and Mahoney [7] present a knowledge representation framework that represents knowledge as network fragments in the knowledge base. Their framework provides for representation of asymmetric independence and canonical intercausal interaction (combination methods in nodes) that is used for combining different network fragments to form problem-specific models.

7 Conclusions

In this paper, we presented the bowtie model and a distinctive elicitation procedure to build stochastic models for planning in the welfare domain. Our results shed some light both on the data elicited from the experts as well as the fusion methods we developed. In particular, we observe significant consistency in the output of different methods for same action fragments. Most of the input nodes are preserved from method to method. This, in our view, is due to the consistency with which experts responded to our model-building efforts. We also note that the use of merging methods has often led to reasonable fragments with improved coverage. The remaining steps of our model-building process will be: (i) elicitation/construction of CPTs for success nodes, and (ii) combination of different action fragments into a single network by merging and combining their output nodes.

Acknowledgements

We thank Russell Almond for clarifying communication between the social scientists and computer scientists in our group, and drawing the first bowtie diagram for us. This work is partially supported by NSF grant ITR-0325063.

References

[1] Bilal M Ayyub. *Elicitation of Expert Opinions for Uncertainty and Risks.* CRC Press, 2001.

[2] Craig Boutilier, Thomas Dean, and Steve Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *J. Artif. Intell. Res. (JAIR)*, 11:1–94, 1999.

[3] Alex Dekhtyar, Raphael Finkel, Judy Goldsmith, Beth Goldstein, and Cynthia Isenhour. Adaptive decision support for planning under hard and soft constraints. In *Proceedings of the AAAI Spring Symposium on Challenges to Decision Support in a Changing World*, pages 17–22, Stanford University, Palo Alto, CA, 2005.

[4] M.J. Druzdzel and M. Henrion. Efficient reasoning in qualitative probabilistic networks. In *Proceedings of the 11th National Conference on Artificial Intelligence*, pages 548–553, 1993.

[5] Siv Hilde Houmb. Combining disparate information sources when quantifying operational security. In *Proceedings of the 9th World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI'05)*, 2005.

[6] Daphne Koller and Ron Parr. Policy iteration for factored mdps. In *Proceedings of the 16th Annual Conference on Uncertainty in Artificial Intelligence (UAI-00)*, pages 326–334, San Francisco, CA, 2000. Morgan Kaufmann Publishers.

[7] Kathryn Laskey and Suzanne Mahoney. Network fragments: Representing knowledge for constructing probabilistic models. In *Proceedings of the 13th Annual Conference on Uncertainty in Artificial Intelligence (UAI-97)*, pages 334–341, San Francisco, CA, 1997. Morgan Kaufmann Publishers.

[8] Pedrito Maynard-Zhang and Daniel J. Lehmann. Representing and aggregating conflicting beliefs. *J. Artif. Intell. Res. (JAIR)*, 19:155–203, 2003.

[9] Martin Mundhenk, Judy Goldsmith, Christopher Lusena, and Eric Allender. Complexity of finite-horizon Markov decision process problems. *Journal of the ACM*, 47(4):681–720, 2000.

[10] M. P. Wellman. Fundamental concepts of qualitative probabilistic networks. *Artif. Intell.*, 44(3):257–303, 1990.