

# Acquiring Planning Knowledge via Crowdsourcing

Jie Gao<sup>a</sup> and Hankz Hankui Zhuo<sup>b</sup> and Subbarao Kambhampati<sup>c</sup> and Lei Li<sup>b</sup>

<sup>a</sup>Zhuhai college,  
Jilin Univ., Zhuhai, China  
jiiegao26@163.com

<sup>b</sup>School of Data & Computer Sci.,  
Sun Yat-sen Univ., China  
zhuohank@mail.sysu.edu.cn

<sup>c</sup>Dept. of Computer Sci. & Eng.,  
Arizona State Univ., US  
rao@asu.edu

## 1 Introduction

Plan synthesis often requires complete domain models and initial states as input. In many real world applications, it is difficult to build domain models and provide complete initial state beforehand. In this paper we propose to turn to the crowd for help before planning. We assume there are annotators available to provide information needed for building domain models and initial states. However, there might be a substantial amount of discrepancy within the inputs from the crowd. It is thus challenging to address the planning problem with possibly noisy information provided by the crowd. We address the problem by two phases. We first build a set of Human Intelligence Tasks (HITs)<sup>1</sup>, and collect values from the crowd. We then estimate the actual values of variables and feed the values to a planner to solve the problem.

In contrast to previous efforts [Zhang *et al.*, 2012; Manikonda *et al.*, 2014] that ask the crowd to do planning, we exploit knowledge about initial states and/or action models from the crowd and feed the knowledge to planners to do planning. We call our approach PAN-CROWD, stands for Planning by Acquiring kNowledge from the CROWD.

## 2 The Formulation of Our Planning Problem

We formulate our planning problem as a quadruple  $P = \langle \tilde{s}_0, g, \mathcal{O}, \bar{\mathcal{A}} \rangle$ , where  $\tilde{s}_0$  is an *incomplete* initial state which is composed of a set of *open propositions*. A proposition is called *open* if there exist variables in the parameter list of the proposition, e.g., “ $on(A, ?x)$ ”<sup>2</sup> is an open proposition since  $?x$  is a variable in the parameter list of proposition *on*. An open initial state can be *incomplete*, i.e., some propositions are missing. The set of variables in  $\tilde{s}_0$  is denoted by  $\mathcal{V}$ .  $\mathcal{O}$  is a set of objects that can be selected and assigned to variables in  $\mathcal{V}$ . We assume  $\mathcal{O}$  can be easily collected based on historical applications.  $\bar{\mathcal{A}}$  is a set of incomplete STRIPS action models.  $\bar{a} \in \bar{\mathcal{A}}$  is called “incomplete” if there are some predicates missing in the preconditions or effects of  $\bar{a}$ . A solution to the problem is a plan and a set of “refined” action models.

## 3 The PAN-CROWD approach

To acquire planning knowledge from the crowd, we first build HITs for action models and initial states, and then refine ini-

tial states and action models based on the answers to HITs given by the crowd. After that we feed the refined initial states and action models to planners to solve the planning problem.

**Building HITs for Action Models:** For this part, we build on our work with the CAMA system [Zhuo, 2015]. We enumerate all possible preconditions and effects for each action. Specifically, we generate actions’ preconditions and effects as follows [Zhuo and Yang, 2014]. If the parameters of predicate  $p$ , denoted by  $\text{Para}(p)$ , are included by the parameters of action  $a$ , denoted by  $\text{Para}(a)$ , i.e.,  $\text{Para}(p) \subseteq \text{Para}(a)$ ,  $p$  is likely a precondition, or an add effect, or a delete effect of  $a$ . We therefore generate three new proposition variables “ $p \in \text{Pre}(a)$ ”, “ $p \in \text{Add}(a)$ ” and “ $p \in \text{Del}(a)$ ”, the set of which is denoted by  $H_{pre} = \{p \in \text{Pre}(a) | \forall p \in P \text{ and } \forall a \in A\}$ ,  $H_{add} = \{p \in \text{Add}(a) | \forall p \in P \text{ and } \forall a \in A\}$ ,  $H_{del} = \{p \in \text{Del}(a) | \forall p \in P \text{ and } \forall a \in A\}$ , respectively. We put all the proposition variables together and estimate the label of each variable by querying the crowd or annotators.

For each proposition in  $H$ , we build a Human Intelligent Task (HIT) in the form of a short survey. For example, for the proposition “ $(\text{ontable } ?x) \in \text{Pre}(\text{pickup})$ ”, we generate a survey as shown below: *Is the fact that “ $x$  is on table” a precondition of picking up the block  $x$ ?* There are three possible responses, i.e., *Yes, No, Cannot tell*, out of which an annotator has to choose exactly one. Each annotator is allowed to annotate a given survey once. Note that the set of proposition variables  $H$  will not be large, since all predicates and actions are in the “variable” form rather than instantiated form, and we only consider predicates whose parameters are included by actions. For example, for the *blocks* domain, there are only 78 proposition variables in  $H$ .

**Building HITs for Initial States:** To generate surveys that are as simple as possible, we assume there are sets of objects known to our approach, each of which corresponds to a *type*. For example,  $\{A, B, C\}$  is a set of objects with *type* “Block” in the *blocks* domain, i.e., there are three blocks known to our approach. We can thus generate a set of possible propositions  $\mathcal{S}$  with the sets of objects and predicates of a domain. For each proposition, we formulate the Human Intelligent Task (HIT) as a short survey, the set of which is denoted by  $\mathcal{H}$ . To reduce the number of HITs, we start from the goal state  $g$ , we search for the action whose add effects match with  $g$ , and update  $g$  to be an internal state by deleting add effects and adding preconditions of the action into  $g$ . After that, we check whether open initial state matches with internal state. Sec-

<sup>1</sup><http://mturk.com>

<sup>2</sup>A symbol preceded by “?” denotes a variable that can be instantiated by an object.  $on(A, ?x)$  suggests  $A$  is on some block  $x$ .

only, from the matched internal state, we select those with unknown variables, and choose them with the same variables to build a set of propositions with unknown variables, which is finally transformed into a set of HITs.

**Estimating True Labels** Assume there are  $R$  annotators and  $N$  tasks with binary labels  $\{1, 0\}$ . The true labels of tasks are denoted by  $\mathcal{Z} = \{z_i \in \{1, 0\}, i = 1, 2, \dots, N\}$ . Let  $\mathcal{N}_j$  is the set of tasks labeled by annotator  $j$ , and  $\mathcal{R}_i$  is the set of annotators labeling task  $i$ . The task assignment scheme can be represented by a bipartite graph where an edge  $(i, j)$  denotes that the task  $i$  is labeled by the worker  $j$ . The labeling results form a matrix  $\mathcal{Y} \in \{1, 0\}^{N \times R}$ . The goal is to find an optimal estimator  $\hat{\mathcal{Z}}$  of the true labels  $\mathcal{Z}$  given the observation  $\mathcal{Y}$ , minimizing the average bit-wise error rate  $\frac{1}{N} \sum_{i \in \{1, 2, \dots, N\}} \text{prob}[\hat{z}_i \neq z_i]$ .

We model the accuracy of annotators separately on the positive and the negative examples [Raykar *et al.*, 2010]. If the true label is one, the true positive rate  $TP^j$  for the  $j$ th annotator is defined as the probability that the annotator labels it as one, i.e.,  $TP^j = p(y_i^j = 1 | z_i = 1)$ . On the other hand if the true label is zero, the true negative rate  $TN^j$  is defined as the probability that annotator labels it as zero, i.e.,  $TN^j = p(y_i^j = 0 | z_i = 0)$ . Suppose we have the training data set  $D = \{x_i, y_i^1, \dots, y_i^R\}_{i=1}^N$  with  $N$  instances from  $R$  annotators, where  $x_i \in \mathcal{X}$  is an instance (typically a  $d$ -dimensional feature vector),  $y_i^j$  is the label (1 or 0) given by the  $j$ th annotator. Considering the family of linear discriminating functions, the probability for the positive class is modeled as a logistic sigmoid, i.e.,  $p(y = 1 | x, w) = \sigma(w^T x)$ , where  $x, w \in R^d$ , and  $\sigma(z) = \frac{1}{1 + e^{-z}}$ .

The task is to estimate the parameter  $w$  as well as the true positive  $\mathcal{P} = \langle TP^1, \dots, TP^R \rangle$  and the true negative  $\mathcal{N} = \langle TN^1, \dots, TN^R \rangle$ . Let  $\theta = \{w, \mathcal{P}, \mathcal{N}\}$ , the probability of training data  $D$  can be defined by  $p(D | \theta) = \prod_{i=1}^N p(y_i^1, \dots, y_i^R | x_i, \theta)$ . The EM algorithm can be exploited to estimate the parameter  $\theta$  by maximizing the log-likelihood of  $p(D | \theta)$  [Raykar *et al.*, 2010]. Let  $\mu_i = p(z_i = 1 | y_i^1, \dots, y_i^R, x_i, \theta)$ . We simply set the threshold as 0.5, i.e., if  $\mu_i > 0.5$ , the value of  $\hat{z}_i$  is assigned with 1, otherwise 0.

**Refining Initial States and Action Models:** We refine the initial state and action models based on estimated true labels. Once the initial state and action models refined, we solve the corresponding revised planning problem using an off-the-shelf planner.

## 4 Experiment

Since the action model acquisition part has been evaluated in the context of CAMA already [Zhuo, 2015], here we focus on evaluating the initial state acquisition part. We evaluated our approach in three planning domains, i.e., *blocks*<sup>3</sup>, *depots*<sup>4</sup> and *driverlog*<sup>4</sup>. We first generated 150 planning problems with complete initial states, denoted by  $\bar{P}$ . After that we randomly removed propositions from the initial states in  $\bar{P}$  base on a ratio  $\alpha$ , resulting in *open* problems  $P$ . Propositions were

then changed to short surveys in natural language. We performed extensive simulations using 20 simulated annotators to complete the human intelligent tasks (HITs). We define the accuracy of our approach by comparing to ground-truth solutions (generated by simulators). In other words, we solve all 150 problems using our approach, and compare the 150 solutions to ground-truth solutions, viewing the ratio of identical solutions over 150 as the accuracy.

We ran our PAN-CROWD algorithm on testing problems by varying from 0.1 to 0.5 the ratio  $\alpha$ , setting the number of objects to be 10. The result is shown in Figure 1. From Figure 1, we can see that the accuracy generally becomes lower when the ratio increases in all three testing domains. This is consistent with our intuition, since the larger the ratio is, the more uncertain the information that is introduced to the original planning problem when using crowdsourcing. However, from the curves we can see the accuracies are no less than 70% when the ratio is smaller than 0.3.

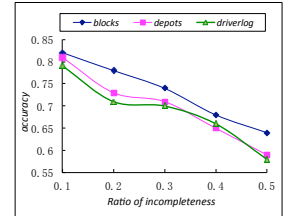


Figure 1: The accuracy w.r.t. ratio  $\alpha$ .

## 5 Discussion and Conclusion

We propose to acquire knowledge about initial states and action models from the crowd. Since the number of HITs sent to the crowd related to initial states could be large, in the future, we could consider how to reduce the number of HITs, e.g. by exploiting backward chaining planning techniques – which only might need to know small parts of the initial state, rather than the whole state. In addition, we could also think of ways of engaging the crowd in more active ways, rather than answering yes/no to HITs. For example, we can give them candidate plans and ask them to critique the plan and/or modify them to make them correct. We then learn knowledge from the modification process and use it for computing plans.

**Acknowledgements** Zhuo’s research is supported by NSFC (No. 61309011) and Fundamental Research Funds for the Central Universities (No. 14lgzd06). Kambhampati’s research is supported in part by a Google Research Award and the ONR grants N00014-13-1-0176 and N00014-15-1-2027.

## References

- [Manikonda *et al.*, 2014] L. Manikonda, T. Chakraborti, S. De, K. Talamadupula, and S. Kambhampati. AI-MIX: using automated planning to steer human workers towards better crowdsourced plans. In *IAAI*, pages 3004–3009, 2014.
- [Raykar *et al.*, 2010] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy. Learning from crowds. *JMLR*, 11:1297–1322, 2010.
- [Zhang *et al.*, 2012] H. Zhang, E. Law, R. Miller, K. Gajos, D. C. Parkes, and E. Horvitz. Human computation tasks with global constraints. In *CHI*, pages 217–226, 2012.
- [Zhuo and Yang, 2014] H. H. Zhuo and Q. Yang. Action-model acquisition for planning via transfer learning. *Artif. Intell.*, 212:80–103, 2014.
- [Zhuo, 2015] H. H. Zhuo. Crowdsourced action-model acquisition for planning. In *AAAI*, pages 3004–3009, 2015.

<sup>3</sup><http://www.cs.toronto.edu/aips2000/>

<sup>4</sup><http://planning.cis.strath.ac.uk/competition/>