A Study of Query Generation Strategies for Interactive Constraint Acquisition [⋆]

Sarah O'Connell, Barry O'Sullivan, and Eugene C. Freuder

Cork Constraint Computation Centre
Department of Computer Science
University College Cork, Ireland
{s.oconnell|b.osullivan|e.freuder}@4c.ucc.ie

Summary. In many practical applications users often find it difficult to articulate their constraints. While users can recognize examples of where a constraint should be satisfied or violated, they cannot articulate the constraint itself. In these situations we would like the computer to take an active role in acquiring the user's constraints. In this paper we present an approach to interactive constraint acquisition based on techniques from the field of machine learning. During interactive constraint acquisition we would like to employ a strategy which minimizes the dialog length between the user and the computer. In this paper, we compare a number of different heuristics for interactive query generation. We demonstrate that the interactive strategy used to acquire constraints depends on the topology of the hypothesis space and the degree to which the human user is being helpful.

Key words: Interactive Constraint Acquisition, Query Generation Strategies.

1 Introduction

In many practical applications users find it difficult to articulate their constraints. While users can recognize examples of where a constraint should be satisfied or violated, they cannot articulate the constraint itself. For example, a customer may be trying to specify a constraint to an engineer, without being able to use the correct engineering terms for the relevant concepts.

Recently, researchers have become more interested in techniques for solving problems where users have difficulties articulating constraints. Freuder and Wallace have considered suggestion strategies for applications where a user cannot articulate all constraints in advance, but can articulate additional constraints when confronted with something which is unacceptable [2]. However, that work assumes that the user can actually articulate each of the constraints when required. In essence their approach is model focussed, in the sense that they are concerned with acquiring a partial model of the user's CSP which is sufficient to be able to find a solution that

[⋆] This work has been supported by Science Foundation Ireland under Grant 00/PI.1/C075.

the user would find acceptable. Padmanabhuni *et al.* have proposed a framework for learning constraints [6]. However, they do not show how negative (unacceptable) examples can be handled by their approach. Rossi and Sperduti have studied the utility of using inductive learning, through the application of gradient descent techniques, for acquiring global and local preferences in the context of interactive soft constraint solving [7]. Their approach assumes that the constraints of the problem are known but that the local and global preferences are not. Freuder and O'Sullivan have begun studying the interactive acquisition of tradeoff constraints in interactive constraint-based configuration [1]. Their approach focuses on generating constraints which model tradeoffs between variables in problems which have become over-constrained during a interactive configuration session. Again, while some constraints in the problem are being acquired, it is assumed that the user can accurately articulate constraints, and that the issue is how to remove over-constrainedness in a model. Finally, a preliminary report on the research reported here has also been presented [5]. Therefore, while constraint acquisition, in one sense or another, is receiving some attention, there is a lack of work on techniques which can support the acquisition of individual constraints necessary to model the user's problem. It is this gap in the research which the work reported here is beginning to address.

The remainder of the paper is organized as follows. In Section 2 the model of interaction which we base our experiments is presented. Section 3 presents the set of query generation strategies that we consider here. Section 4 presents the results of our experiments. In Section 5 some brief concluding remarks are made.

2 A Model for Interactive Constraint Acquisition

In this paper we present an approach to interactive constraint acquisition based on techniques from the field of machine learning. In our system, constraint acquisition is modelled as search through an "hypothesis space" of constraints over which a general-to-specific ordering is explicitly known, or is implicit in its representation. Examples provided by the user can be used to identify a *version space* ([3]) of constraints that the user could be attempting to articulate. The system then attempts to generalize the user's examples by choosing an hypothesis from the current version space and attempting to validate it. This attempt at generalization involves proposing a *qualifying example* (a query) to the user based on the the candidate hypothesis. If the qualifying example is accepted by the user, the system attempts to generalize again. An accepted qualifying example is implicitly positive and is therefore used to further refine the version space for the constraint. Generalization continues until either the version space for the user's constraint has been reduced to a single hypothesis, or an example has been proposed to the user which he rejects. A rejected example is implicitly negative. Therefore, these negative examples are also used to further refine the version space for the constraint. If the user rejects an example he is invited to provide another positive example if one is available. The process terminates once the version space has converged upon a single hypothesis.

3 Generalization Strategies

In the work presented here, generalization is performed interactively through the computer's generation of a qualifying question (a query) and presenting it to the user. Mitchell [4] suggests that the optimal query generation strategy to adopt when learning using version spaces is one that involves generating instances that satisfy exactly half of the hypotheses in the version space. For the purposes of this paper we will refer to this strategy as *50/50*. Given that we are concerned with the interactive acquisition of constraints, one of the key metrics that is relevant to us is the total number of interactions with the user (the dialog length). Obviously, we would wish this number to be minimal. Therefore, query generation strategies which have the potential to minimize the length of the interaction (dialog) with the human are of signifi cant value. One could imagine that in the real-world the difference between a user using an interactive constraint acquisition system and abandoning it maybe their lack of tolerance of one more interaction.

Therefore, it is worthwhile posing the question: does the combination of acquisition problem, user profi le and version space topology have an effect on the performance of different query generation strategies? In particular, can we tailor our query generation strategy so that the total number of interactions with the human user is minimized? The query generation strategies that we have studied are: (a) *50/50* – our baseline case, (b) *Least Generalization (Random),* (c) *Least Generalization (Maximum Connectivity),* (d) *Least Generalization (50/50)* and (e) *Random Choice*. We outline the differences below:

50/50 (Baseline case) – This heuristic represents the conventional wisdom in the machine learning fi eld and generates queries which satisfy half of the hypotheses in the current version space. This is the best known general-purpose query generation strategy for version space learning.

Least Generalization (Random) – One disadvantage with the previous strategy is that while it is based on a sound principle, it can give rise to the generation of queries that are diffi cult for the user to explain from the perspective of perspicuity. Therefore, it may be worthwhile to use strategies that generate queries which the user would regard as "logical" generalizations. The *Least Generalization (Random)* strategy generates a query at random from the set of examples that could be used to qualify a minimally generalized hypothesis.

Least Generalization (Maximum Connectivity) – This strategy selects a query whose set of generalizations is maximal. This is a risky strategy since it relies on a negative response to the query to signifi cantly reduce the size of the version space.

Least Generalization (50/50) – This strategy attempts a least generalization step based on the query which reduces the version space by a similar amount regardless of whether it is accepted or rejected.

Random Choice – This strategy selects a query at random.

4 Evaluation

The objective of our evaluation was to compare the length of dialog (number of interactions) between the user and computer required to acquire a constraint. The evaluation was performed by varying (a) the topology of the original hypothesis space and (b) the extent to which the user was trying to be helpful.

We studied two alternative hypothesis space topologies. Each topology was parameterized with a single integer value representing the number of atomic examples that formed the vocabulary of the interaction. Figure 1 presents the hypothesis-space topologies used for the purposes of our evaluation. We will refer to the topology illustrated in Figure 1(a) as the *tree topology*, while we will refer to the topology illustrated in Figure 1(b) as the *lattice topology*.

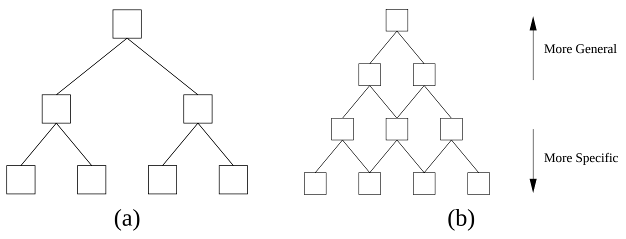


Fig. 1. The hypothesis space topologies used in our evaluation: (a) the tree topology and (b) the lattice topology. Note that we do not show the trivially most specific case, where nothing is acceptable.

For the purposes of our experiments, we developed an hypothesis-space based on the *tree topology* having 64 example nodes, giving a total of 127 possible constraints. We also generated an hypothesis space based on the *lattice topology* having 15 example nodes, giving a total of 120 possible constraints.

The extent to which the user was trying to be helpful was modelled as the number of examples that he would give to the computer when user-specifi ed examples were required. An *unhelpful user* gave a single example to the computer, while a *helpful user* gave a randomly chosen subset of the examples that exemplify the constraint being articulated. This is a reasonable approach to adopt since the user must be able to present at least one example of the constraint, but may be able to provide more than one in attempting to be helpful.

In the context of the hypothesis space based on the *tree topology* we compared the *50/50* and *Least Generalization (Random)* strategies. The other two variants of the *Least Generalization* strategy, namely the *Least Generalization (Maximum Connectivity)* and *Least Generalization (50/50)* would behave in exactly way as *Least Generalization (Random)* the context of this hypothesis space topology, since there is never a situation where a principled choice could be made between alternative queries. In the context of the *lattice topology*, all strategies were compared.

The model of interaction used in the evaluation was based on that described in Section 2. Each experiment involved counting the number of interactions, between the simulated user and our constraint acquisition system, required to acquire constraints from each of the hypothesis spaces described above. The number of interactions (dialog length) was computed as the sum of the number of interactions used for user-specifi ed examples and the number of interactions for system-generated queries. Since we are concerned with reducing the length of the dialog between the user and computer, we acquired each constraint in each hypothesis space, recording the total number of interactions required to acquire it and its depth in the hypothesis space. Depth, in this context, is a measure of the generality of the constraint. The more general the constraint, the deeper it is in the hypothesis space. The comparison presented below is based on averages over 50 runs of each experiment. In all the graphs, the x-axis represents the depth in the hypothesis space of the constraint being acquired.

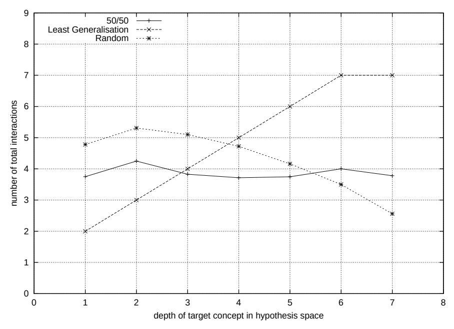
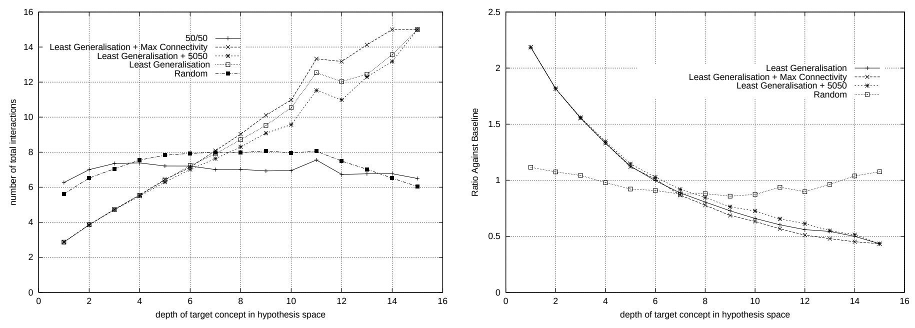


Fig. 2. Comparison of strategies on the tree topology when the user is not being helpful.

Figure 2 illustrates the average number of interactions required to acquire constraints at various depths in the hypothesis space. In this scenario, the hypothesis space is a (binary) tree and the user is not being helpful, only giving a single example at each point that a user-specifi ed example is sought. It can clearly be seen that by far the best strategy to adopt is *50/50*. However, the *Least Generalization* strategy performs well for less general constraints.

Figure 3(a) again illustrates the average number of interactions required to acquire constraints at various depths in the hypothesis space. In this scenario, the only change over that illustrated in Figure 2 is that we have a *lattice* topology. Again, from Figure 3(a), it can be seen that the best strategy across the entire version space is *50/50*. However, for constraints which are less general, the *Least Generalization* strategies perform best.

Figure 3(b) presents a comparison of the the dialog lengths of each of the strategies in Figure 3(a), which shows the relative saving of the various strategies over *50/50*. On this graph, all points with a value greater than 1 represent an improvement over the *50/50* baseline strategy. It is clear to see that the *Least Generalization* strategies perform very well for less general constraints.



(a) Dialog Length (b) Comparison against the *50/50* strategy

Fig. 3. Comparison of strategies on the lattice topology when the user is not being helpful.

Figure 4 again illustrates the average number of interactions required to acquire constraints at various depths in the hypothesis space for a *tree* topology. The user in this scenario is more helpful than in the previous two scenarios. Here the user presents a random set of examples. It can be seen clearly that the best strategy across the entire hypothesis space is *Least Generalization*. This is a signifi cant improvement over the scenario illustrated in Figure 2. It is clear evidence that the choice of generalization strategy can be dramatically informed by knowledge of the nature of the interaction with the user, in particular the degree to which he is helpful.

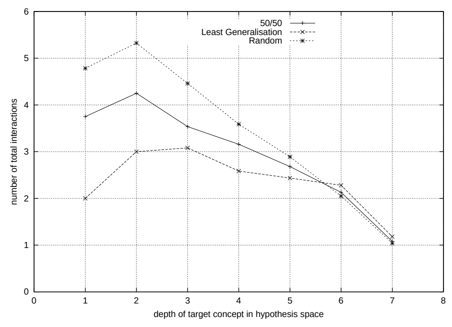
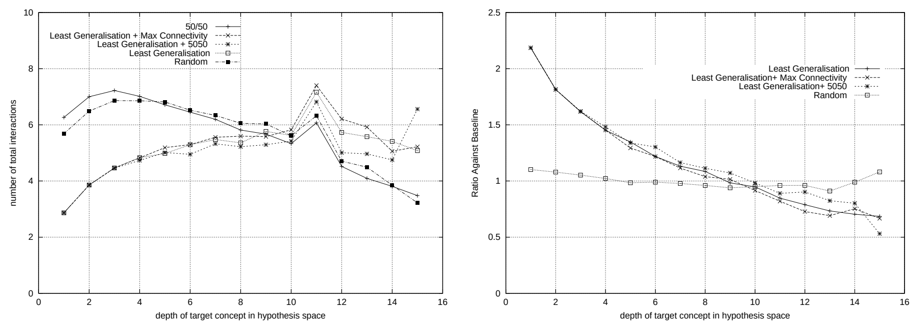


Fig. 4. Comparison of strategies on the tree topology when the user is being helpful.

Finally, Figure 5(a) illustrates the average number of interactions required to acquire constraints at various depths in the hypothesis space for a *lattice* topology. Again the user in this scenario is helpful. It can be seen clearly that while *50/50* performs consistently well across the entire hypothesis space, the best strategy is again based on *Least Generalization*. The improvement that these heuristics offer over *50/50* can be seen more clearly in Figure 5(b), which shows the relative saving of the various strategies over *50/50*. At almost every depth, the *Least Generalization (50/50)* strategies either save on interactions over the *50/50* approach, or closely follow it. This, again, is evidence that the choice of generalization strategy can be dramatically informed by knowledge of the nature of the interaction with the user. In general it can be seen that strategies based on *Least Generalization* are more powerful than *50/50* on constraints which are more specifi c in the version space and when the user is being helpful, perform competitively across the version space.



(a) Dialog Length (b) Comparison against the *50/50* strategy

Fig. 5. Comparison of strategies on the lattice topology when the user is being helpful.

In order to get an indication of the magnitude of these savings, Table 1 presents a comparison of the average savings in the learning rates for all constraints for each of the generalization strategies over the *50/50* baseline. It can be seen that on the average the *Random* strategy never beats *50/50*, regardless of the type of user from whom we are acquiring constraints. However, the performance of the variants of the *Least Generalization* strategy are interesting.

Table 1. Comparison of learning rates against baseline

Heuristic	*Tree One Ex.*	*Lattice One Ex.*	*Tree Multiple Ex.*	*Lattice Multiple Ex.*
Least Generalization	-26%	-27%	21%	10%
Least Generalization + 50-50	-	-22%	-	12%
Least Generalization + Max Conn.	-	-33%	-	8%
Random	-11%	-5%	-19%	0%

While there was no benefi t, on the average, from using a *Least Generalization* strategy when the user is not being helpful, once the user became helpful, there was a marked improvement in the performance of these strategies. While, on the average we can see that the benefi t is not large, we have seen above that *Least Generation* strategies are best for more specifi c constraints. We can also see from Table 1, that on tree-structured hypothesis spaces, a helpful user coupled with a *Least Generalisation* query strategy can give good reductions in dialog length. While acknowledging the general purpose power of the *50/50* strategy, we have some evidence to hypothesise that there are circumstances when it is worthwhile tailoring the query generation strategy of the acquisition system based on the degree to which the user (teacher) is being helpful, the topology of the hypothesis space and the generality of the target constraint.

5 Conclusions

The work presented in this paper is concerned with interactive constraint acquisition. In this paper, we have compared a number of different heuristics for query generation for interactive constraint acquisition. We have demonstrated that the choice of generalization strategy is closely related to the degree to which the user is being helpful and version space topology. Future work will address the utility of hybrid strategies for query generation for constraint acquisition. In addition, we will focus on learning disjunctive constraints and also, more importantly, sets of constraints. While in this paper we have compared a helpful teacher with one which is more helpful, a further investigation focused on how the teachers ability to provide good examples can affect the acquisition process needs to be undertaken. Such a study would more formally study the interaction between an intelligent learner with teachers of varying abilities and, in particular, study whether skill is more important than helpfulness.

References

1. Eugene C. Freuder and Barry O'Sullivan. Generating Tradeoffs for Interative Constraint-Based Configuration. In Toby Walsh, editor, *Proceedings of CP-2001*, pages 590–594, November 2001.
2. Eugene C. Freuder and Richard J. Wallace. Suggestion Strategies for Constraint-Based Matchmaker Agents. In *Proceedings of CP-98*, pages 192–204, October 1998.
3. Tom Mitchell. Generalization as Search. *Artificial Intelligence*, 18(2):203–226, 1982.
4. Tom Mitchell. Concept Learning and the General-to-Specific Ordering. In *Machine Learning*, Chapter 2, pages 20–51. McGraw Hill, 1997.
5. Barry O'Sullivan, Eugene C. Freuder, and Sarah O'Connell. Interactive Constraint Acquisition – Position Paper. In *Proceedings of the CP-2001 Workshop on User-Interaction in Constraint Satisfaction*, pages 73–81, December 2001.
6. Srinivas Padmanabhuni, Jia-Huai You, and Ghose Aditya. A Framework for Learning Constraints. In *Proceedings of the PRICAI-96 Workshop on Induction of Complex Representations*, August 1996.
7. Francesca Rossi and Alessandro Sperduti. Acquiring both Global and Local Preferences in Interactive Constraint Solving via Machine Learning. In *Proceedings of the CP-2001 Workshop on User-Interaction in Constraint Satisfaction*, pages 83–95, December 2001.