Position Paper
Interactive Constraint Acquisition

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

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

Abstract. 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. Constraint acquisition is modeled as search through a "hypothesis space" of constraints over which specific-to-general ordering is known. We have begun studying the issues involved in interactively acquiring constraints. We describe our basic approach in the context of a simple example involving the acquisition of a placement constraint. Finally, we lay out a program for further study, outlining our research agenda.

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. There are many reasons for this. 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. By pointing out to the engineer examples of what is acceptable, the customer can be assisted in finding the right term for what she is trying to articulate. This can be regarded as an instance of interactive constraint acquisition. The customer cannot articulate her constraint but by interacting with the engineer, her examples of what she would accept or reject help to define her constraint.

Recently, researchers have become more interested in the 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 of her constraints in advance, but can articulate additional constraints when confronted with something which she would find unacceptable [1]. However, that work assumes that the user can actually articulate a

* This work was performed while this author was at the University of New Hampshire.

constraint. Padmanabhuni *et al* have proposed a framework for learning constraints [4]. However, they do not show how negative (unacceptable) examples can be handled by their approach.

In this paper we present an approach to interactive constraint acquisition based on techniques from the field of machine learning [3]. In particular, constraint acquisition is regarded as search through a "hypothesis space" of constraints over which an general-to-specific ordering is known. Examples provided by the user can be used to identify a version space ([2]) of candidate constraints that the user could be attempting to articulate. We employ an interactive model similar to that proposed by Sammut and Banerji [5]. As the user provides examples of what she would find acceptable (positive examples), we attempt to generalize these examples by asking questions which attempt to qualify potential generalizations. If the user finds a generalization unacceptable, this provides us with a negative example that can be used to further refine the version space of the constraint being acquired.

The remainder of the paper is organized as follows. Section 2 presents an overview of concept learning and version spaces. Section 3 presents the model of interaction used in the work presented here. Section 4 presents the details of our current approach using an example involving the acquisition of a simple placement constraint. Section 5 presents an agenda for future research. In Section 6 a number of concluding remarks are made.

2 An Overview of Concept Learning using Version Spaces

Concept learning can be defined as inferring a boolean-valued function from training examples of its input and output [2]. Concept learning can be regarded as search through a space of hypotheses implicitly defined by the hypothesis representation. A number of algorithms for concept learning are known: the *Find-S* algorithm, the *List-Then-Eliminate* algorithm, the *Candidate-Elimination* algorithm [2].

In the preliminary research presented here Mitchell's *List-Then-Eliminate* algorithm is used as the basis for an interactive constraint acquisition system. This algorithm outputs a description of the set of all hypothesis consistent with the training examples. A hypothesis is *consistent* with the training examples if it correctly classifies these examples: negative examples are classified as negative, positive examples are classified as positive. The term *version space* is used to describe the subset of the hypothesis space consistent with the training examples.

3 Model of Interaction

The model of interaction employed here is as follows. The user is assumed to be incapable of articulating her constraint, but can provide and recognize positive and negative examples of it. The interaction begins with the user providing a positive example of the constraint that she wish to articulate. Using the *List-Then-Eliminate* algorithm this example can be used to determine a version space for the constraint. The system then attempts to generalize the user's example. This generalization step involves determining whether there exists a maximally specific, but more general that the current, hypothesis which the user would find acceptable. This generalization involves proposing a *qualifying example* to the user based on the the candidate hypothesis. This qualifying example is generated based on the current version space. 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 she 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 she is asked to provide another positive example.

The set of possible examples, that can be provided by the user or used as qualifying examples, is equivalent to the least "trivially" general hypotheses of the original hypothesis space. In other words, removing the most specific hypothesis, the remaining next most specific hypotheses can be used as either a user-specified example or a system-generated qualifying example. Each such hypothesis can only be used once during a constraint acquisition session.

4 Case-Study – Acquiring Placement Constraints

Using the model to interaction presented in Section 3 let us consider a scenario where a hypothesis space related to the placement of queens around a box is known. Figure 1 illustrates the scenario. A queen can be placed in any box, for example in the *Above Right* position.

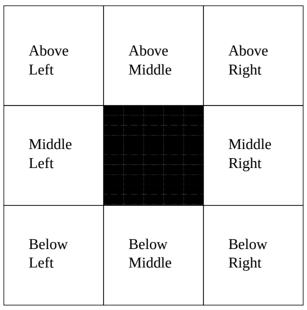

Fig. 1. An example constraint acquisition problem.

The hypothesis space presented in Figure 2 defines the various constraints that can be used to define the placement of the queens with respect to boxes. This

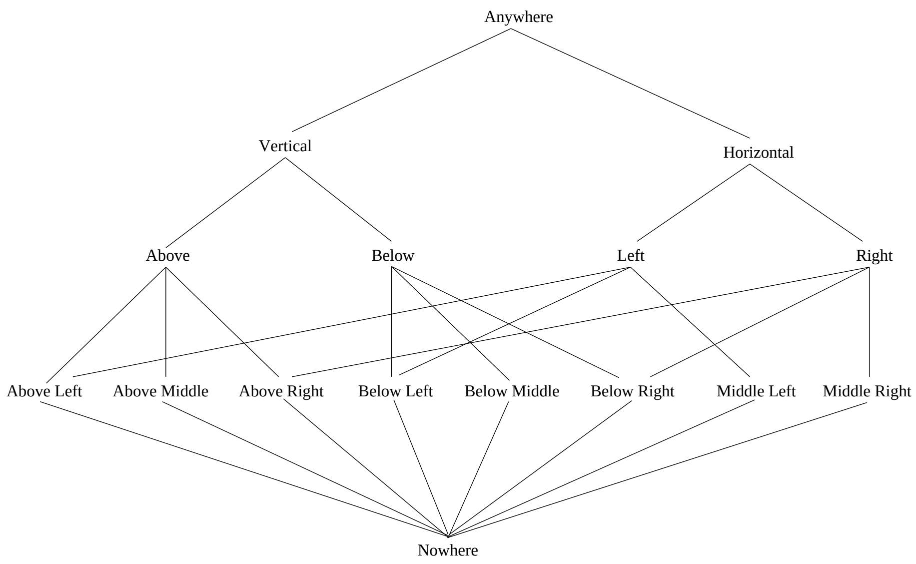

Fig. 2. An hypothesis space for the example illustrated in Figure 1.

hypothesis space is ordered general-specific from top to bottom. For example, if a queen was placed in the *Above Right* position, this indicates that the placement constraint is one of the following:

– the queen can be placed *Above Right* with respect to the box;
– the queen can be placed *Above* with respect to the box;
– the queen can be placed *Vertical* with respect to the box;
– the queen can be placed *Right* with respect to the box;
– the queen can be placed *Horizontal* with respect to the box;
– the queen can be placed *Anywhere* with respect to the box.

This is because each of these constraints is more general that the example given. The task is to acquire the maximally-specific constraint which is consistent with the user's examples.

The set of possible examples here comprises the hypotheses *Above Left*, *Above Middle*, *Above Right*, *Below Left*, *Below Middle*, *Below Right*, *Middle Left* and *Middle Right*.

Let us assume that the constraint that the user is attempting to articulate is that *queens can be placed to the right of the box*. Let us also assume that the user's first (positive) example of the constraint to be acquired is in the *Above Right* position. This scenario is illustrated in Figure 3(a). The version space corresponding to this example is illustrated in Figure 4 using boxes to indicate the hypotheses of the original hypothesis space that are consistent with the user's example.

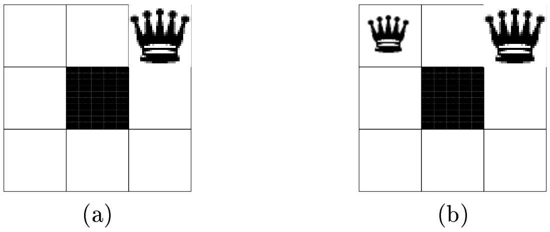

Fig. 3. An example constraint acquisition problem.

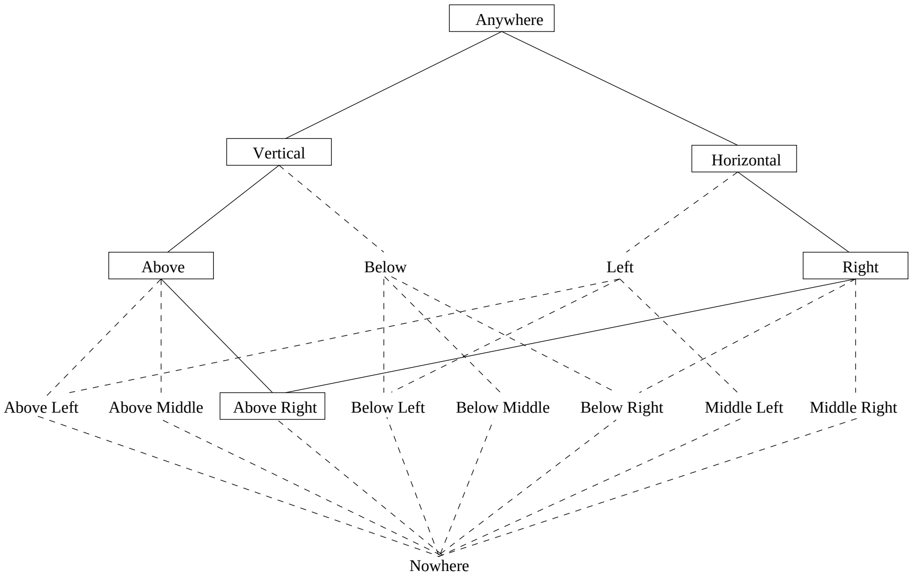

Fig. 4. An example constraint acquisition problem.

The maximally specific generalizations of the user's example are either *Above* or *Right*. The system selects, lexicographically, the hypothesis *Above*. This hypothesis can be qualified by asking if either *Above Left* or *Above Middle* are acceptable. The system selects, lexicographically, the qualifying example *Above Left*. This is illustrated in Figure 3(b).

Since this example is inconsistent with the constraint that the user is trying to articulate she would reject it. The effect of this is that *Above Left* is now regarded as a negative example of the user's constraint. Therefore, the user's constraint cannot be *Above*, which also implies that it cannot be *Vertical* or *Anywhere*. In addition, the user's constraint cannot be *Left*, which also implies that it cannot be *Horizontal*. Therefore, the only remaining possible generalization is that the constraint is that the queen can only be placed to the *Right* of the box. The version space corresponding to this scenario is illustrated in Figure 5.

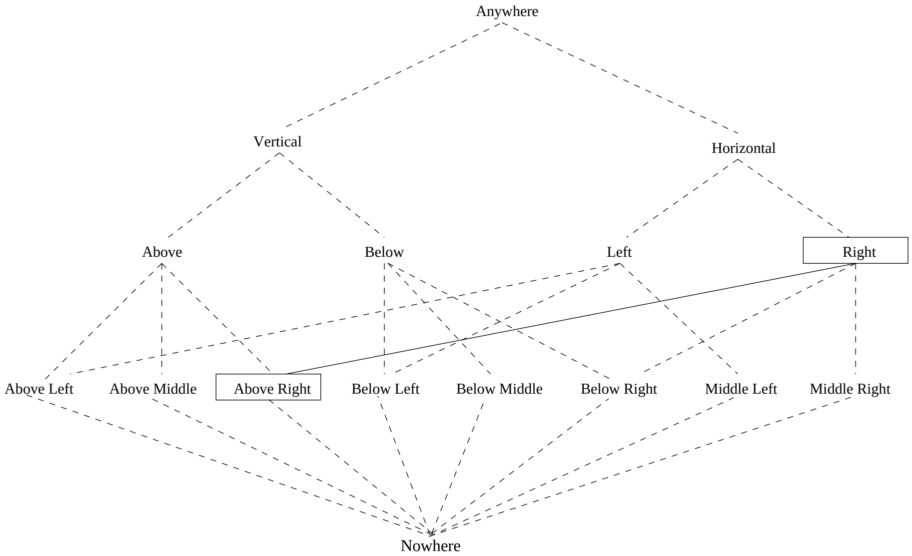

Fig. 5. An example constraint acquisition problem.

Since the qualifying example was rejected by the user, another example is requested. In Figure 6(a) the user presents *Below Right* as a positive example. The consequence of this example is that the version space is refined to contain the single hypothesis *Right* (Figure 7). Therefore, since the version space has been reduced to a single hypothesis the system terminates the interaction having acquired the constraint that queens can be placed to the *Right* of the box (Figure 6(b)).

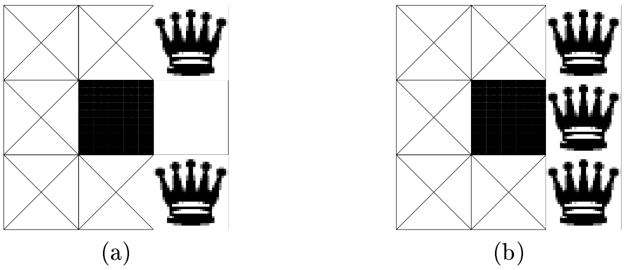

Fig. 6. An example constraint acquisition problem.

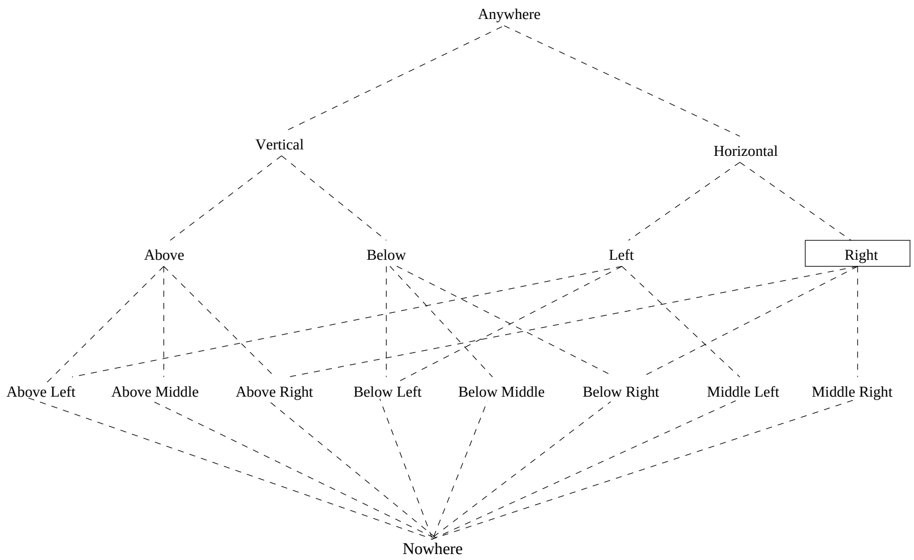

Fig. 7. An example constraint acquisition problem.

5 Agenda for Future Work

There are many interesting research questions raised by the preliminary work presented here. These will be discussed briefly:

- When attempting to generalize examples provided by the user, the choice of qualifying question is an important one. For example it is possible to select a question which can eliminate not only the current generalized hypothesis, but subsequent hypotheses which may be considered later during the acquisition process. In the example presented in Section 4 the choice of qualifying question, "is *Above Left* acceptable?" could be regarded as significant. The rejection of this question means that not only is *Above* not acceptable, but *Left* and consequently *Horizontal* also are not feasible. There may be some general principles which can guide selection of qualifying questions. We aim to investigate this issue.
- Related to the previous question is the level of generalization that should be attempted during the interaction process. At present we attempt to qualify a maximally-specific generalization of the user's input before requesting further examples. However, it may be worthwhile, under certain circumstances, to qualify maximally-general generalizations. There may be advantages to mixing the initiative here. Also, attempting more than a single generalization may be an effective way to prune the version space.
- We need to study metrics for evaluating alternative models of interaction with the user. The most obvious metric is the number of examples that the user needs to provide. Also of importance is the number of questions that

the system asks the user. Some combination of the above also is useful: for example, the number of correct generalizations that are attempted. Essentially all of these metrics are related to system-user dialog length. However, there are other metrics to be considered such as computational overhead and speed of response.
- Using constraint-based inference, such as arc-consistency may be of value. To some extent, inference already forms the basis of Mitchell's *List-Then-Eliminate* algorithm. However, other CSP techniques may also be useful to consider, e.g. forward checking, exploiting implied constraints etc.
- An important issue is how the "language", or "bias" in Machine Learning terms, is generated. Notice, for example, that the language presented in Section 4 here does not have a single concept for specifying a diagonal constraint. Issues relating to bias in the context of constraint acquisition need to be investigated. It has been assumed in this paper that the hypothesis space is known in advance. While in general this is not true, there is significant anecdotal evidence to support the validity of the assumption in many real-world applications. However, the acquisition of the hypothesis space is also an issue which requires attention.
- We are currently investigating a number of alternative test-beds for interactive constraint acquisition. It will be of considerable value to determine to what extent general-purpose techniques can be developed based on practical experience.

Our current research is directed towards providing answers to the questions listed above.

6 Concluding Remarks

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 presented an approach to interactive constraint acquisition based on techniques from the field of machine learning. Constraint acquisition was modeled as search through a "hypothesis space" of constraints over which specific-to-general ordering is known. We have begun studying the issues involved in interactively acquiring constraints. We described our basic approach in the context of a simple example involving the acquisition of a placement constraint. Finally, we described the parameters of a program for further study.

Acknowledgments

Dr. O'Sullivan is supported by Enterprise Ireland, under their International Collaboration Programme (Grant No. IC/2001/058). Professor Freuder is supported by a Principal Investigator Award from Science Foundation Ireland.

References

1. Eugene C. Freuder and Richard J. Wallace. Suggestion strategies for constraint-based matchmaker agents. In *Principles and Practice of Constraint Programming - CP98*, pages 192–204, October 1998.
2. Tom Mitchell. Concept learning and the general-to-specific ordering. In *Machine Learning*, chapter 2, pages 20–51. McGraw Hill, 1997.
3. Tom Mitchell. *Machine Learning*. McGraw Hill, 1997.
4. Srinivas Padmanabhuni, Jia-Huai You, and Ghose Aditya. A framework for learning constraints. In *Proceedings of the Fourth Pacific Rim International Conference on Artificial Intelligence (PRICAI-96) Workshop on Induction of Complex Representations*, August 1996.
5. Claude Sammut and Ranan Banerji. Learning concepts by asking questions. In Ryszard S. Michalski, Jamie G. Carbonell, and Tom M. Mitchell, editors, *Machine Learning : An Artificial Intelligence Approach*, volume 2, chapter 7, pages 187–191. Morgan Kaufmann, Los Altos, California, 1986.