REMEMBERING TO FORGET

A Competence-Preserving Deletion Policy for CBR Systems

Barry Smyth
Hitachi Dublin Laboratory,
Trinity College Dublin,
Dublin, IRELAND.
Phone: +353-1-6798911
Fax: +353-1-6798926
EMail: barry.smyth@hdl.ie

Mark T. Keane
Department of Computer Science,
Trinity College Dublin,
Dublin, IRELAND.
Phone: +353-1-7021534
EMail: mark.keane@cs.tcd.ie

ABSTRACT

The *utility problem* occurs when the cost associated with searching for relevant knowledge outweighs the benefit of applying this knowledge. Two common machine learning strategies have been used in knowledge-based systems to cope with this problem. One permits the unrestricted acquisition of knowledge but limits the depth of search. A second permits complete search but ensures that stored knowledge is useful, deleting any structures that do not contribute to performance in a positive sense, and essentially limiting the size of the knowledge-base. We will examine the second of these strategies in the context of CBR systems. In case-based reasoning the impact of the utility problem is very much dependant on the size and growth of the case-base; larger case-bases mean more expensive retrieval stages, an expensive overhead in CBR systems. In CBR, traditional deletion strategies will keep performance in check (and thereby control the classical utility problem) but they may cause problems for system competence. This effect is demonstrated experimentally and in reply two new deletion strategies are proposed that can take both competence and performance into consideration during deletion.

Keywords:

Case-Based Reasoning, Learning, The Utility Problem.

Declaration:

This paper has not already been accepted by and is not currently under review for a journal or another conference. Nor will it be submitted for such during IJCAI's review period.

REMEMBERING TO FORGET

A Competence-Preserving Deletion Policy for CBR Systems

ABSTRACT

The *utility problem* occurs when the cost associated with searching for relevant knowledge outweighs the benefit of applying this knowledge. Two common machine learning strategies have been used in knowledge-based systems to cope with this problem. One permits the unrestricted acquisition of knowledge but limits the depth of search. A second permits complete search but ensures that stored knowledge is useful, deleting any structures that do not contribute to performance in a positive sense, and essentially limiting the size of the knowledge-base. We will examine the second of these strategies in the context of CBR systems. In case-based reasoning the impact of the utility problem is very much dependant on the size and growth of the case-base; larger case-bases mean more expensive retrieval stages, an expensive overhead in CBR systems. In CBR, traditional deletion strategies will keep performance in check (and thereby control the classical utility problem) but they may cause problems for system competence. This effect is demonstrated experimentally and in reply two new deletion strategies are proposed that can take both competence and performance into consideration during deletion.

1 Introduction

The traditional wisdom in knowledge-based systems has been that *more knowledge is a good thing*. However, in recent years this view has been questioned with demonstrations that certain "harmful" knowledge may actually degrade system performance. This problem has been dubbed the *utility problem* (Minton, 1990; Tambe, Newell, Rosenbloom, 1990). In case-based reasoning systems (CBR) a special case of the utility problem arises called the *swamping problem* (Francis & Ram, 1993b). The swamping problem relates to the expense of searching large case-bases for appropriate cases with which to solve the current problem.

Markovitch and Scott (1993) have characterised strategies for dealing with the utility problem in terms of information filters applied at different stages in the problem solving process (e.g., acquisition, feature extraction, matching, learning). Two common strategies which are especially relevant to the swamping problem are *selective utilization*□and *selective retention.* Selective utilization filters (Markovitch

& Scott, 1989) prevent the problem solver from "seeing" certain knowledge items for some period of time, and are often implemented in CBR systems as indexing schemes (cf. Kolodner, 1994) or by time bounding the case-retrieval mechanism (cf. Brown, 1993; Veloso, 1992); in these approaches only a subset of the case-base is made available to the retrieval mechanism with the attendant disadvantage that certain cases may be missed at retrieval time. Unfortunately, these omissions can result in unnecessarily complex adaptation stages or even problem-solving failures. As an alternative, selective retention filters permanently remove knowledge from the knowledge-base (Holland, 1986; Markovitch & Scott, 1988; Minton, 1990) and are represented in CBR systems by techniques that only learn certain cases or that delete cases that appear to be redundant; the selection of a case to delete is controlled by a *deletion policy*. An advantage with selective retention is that the whole case-base is available at retrieval time but it is critical that only quality cases are learned to keep the case-base small and retrieval time at acceptable levels.

In this paper, we investigate a number of traditional deletion policies and their application to CBR. In section 2, we argue that such policies may not be directly applicable to CBR systems because they damage the competence potentials of such systems over time.. Section 3 introduces two new policies designed specifically with CBR in mind. These new policies recognise the possibility of competence degradation through case deletion and safe-guard against it by using an explicit model of case competence. Finally, in section 4, before a concluding look at the applicability and appropriateness of our approach, we demonstrate its effectiveness with the aid of experimental results.

2 Coping With The Utility Problem

Machine learning research offers a variety of different deletion policies, most of which should work well to preserve performance in CBR systems (and hence limit the classical utility problem). However, they can result in a gradual but irreversible reduction in the competence of CBR systems. In short, to control the utility problem in CBR systems, a deletion policy is required that preserves both performance and competence.

2.1 Traditional Deletion Policies

A very simple deletion policy is *random deletion*. According to this policy a random item is removed from the knowledge-base once the size of the knowledge-base exceeds some predefined limit. Surprisingly enough, random deletion can work very well and can often be as effective as more principled (and expensive) methods (Markovitch and Scott, 1988).