Title: Explanation-based learning

URL: https://en.wikipedia.org/wiki/Explanation-based_learning

PageID: 21638340

Categories: Category:Machine learning

Explanation-based learning (EBL) is a form of machine learning that exploits a very strong, or even perfect, domain theory (i.e. a formal theory of an application domain akin to a domain model in ontology engineering , not to be confused with Scott's domain theory) in order to make generalizations or form concepts from training examples. It is also linked with Encoding (memory) to help with Learning .

Details

An example of EBL using a perfect domain theory is a program that learns to play chess through example. A specific chess position that contains an important feature such as "Forced loss of black queen in two moves" includes many irrelevant features, such as the specific scattering of pawns on the board. EBL can take a single training example and determine what are the relevant features in order to form a generalization.

A domain theory is perfect or complete if it contains, in principle, all information needed to decide any question about the domain. For example, the domain theory for chess is simply the rules of chess. Knowing the rules, in principle, it is possible to deduce the best move in any situation. However, actually making such a deduction is impossible in practice due to combinatoric explosion . EBL uses training examples to make searching for deductive consequences of a domain theory efficient in practice.

In essence, an EBL system works by finding a way to deduce each training example from the system's existing database of domain theory. Having a short proof of the training example extends the domain-theory database, enabling the EBL system to find and classify future examples that are similar to the training example very quickly. The main drawback of the method—the cost of applying the learned proof macros, as these become numerous—was analyzed by Minton.

Basic formulation

EBL software takes four inputs:

a hypothesis space (the set of all possible conclusions)

a domain theory (axioms about a domain of interest)

training examples (specific facts that rule out some possible hypothesis)

operationality criteria (criteria for determining which features in the domain are efficiently recognizable, e.g. which features are directly detectable using sensors)

Application

An especially good application domain for an EBL is natural language processing (NLP). Here a rich domain theory, i.e., a natural language grammar—although neither perfect nor complete, is tuned to a particular application or particular language usage, using a treebank (training examples). Rayner pioneered this work. The first successful industrial application was to a commercial NL interface to relational databases. The method has been successfully applied to several large-scale natural language parsing systems, where the utility problem was solved by omitting the original grammar (domain theory) and using specialized LR-parsing techniques, resulting in huge speed-ups, at a cost in coverage, but with a gain in disambiguation.

EBL-like techniques have also been applied to surface generation, the converse of parsing.

When applying EBL to NLP, the operationality criteria can be hand-crafted, or can be

inferred from the treebank using either the entropy of its or-nodes or a target coverage/disambiguation trade-off (= recall/precision trade-off = f-score). EBL can also be used to compile grammar-based language models for speech recognition , from general unification grammars. Note how the utility problem, first exposed by Minton, was solved by discarding the original grammar/domain theory, and that the quoted articles tend to contain the phrase grammar specialization —quite the opposite of the original term explanation-based generalization. Perhaps the best name for this technique would be data-driven search space reduction. Other people who worked on EBL for NLP include Guenther Neumann, Aravind Joshi, Srinivas Bangalore, and Khalil Sima'an [citation needed] .

See also

One-shot learning in computer vision

Zero-shot learning

References