Chapter VII
Induction as a Search Procedure

Stasinos Konstantopoulos
NCSR ‘Demokritos’, Greece

Rui Camacho
Universidade do Porto, Portugal

Nuno A. Fonseca
Universidade do Porto, Portugal

Vítor Santos Costa
Universidade Federal do Rio de Janeiro, Brasil

ABSTRACT

This chapter introduces inductive logic programming (ILP) from the perspective of search algorithms in computer science. It first briefly considers the version spaces approach to induction, and then focuses on inductive logic programming: from its formal definition and main techniques and strategies, to priors used to restrict the search space and optimized sequential, parallel, and stochastic algorithms. The authors hope that this presentation of the theory and applications of inductive logic programming will help the reader understand the theoretical underpinnings of ILP, and also provide a helpful overview of the State-of-the-Art in the domain.

INTRODUCTION

Induction is a very important operation in the process of scientific discovery, which has been studied from different perspectives in different disciplines. Induction, or inductive logic, is the process of forming conclusions that reach beyond the data (facts and rules), i.e., beyond the current boundaries of knowledge. At the core of inductive thinking is the ‘inductive leap’, the stretch of imagination that draws a reasonable inference from the available information. Therefore, inductive conclusions are only probable, they may turn out to be false or improbable when given more data.

In this chapter we address the study of induction from an artificial intelligence (AI) perspective and, more specifically, from a machine learning perspective, which aims at automating the inductive inference process. As is often the case in AI,
Induction as a Search Procedure

this translates to mapping the ‘inductive leap’ onto a search procedure. Search, therefore, becomes a central element of the automation of the inductive inference process.

We consider two different approaches to the problem of induction as a search procedure: version spaces is an informal approach, more in line with traditional machine learning approaches; by contrast, search-based algorithms for inductive logic programming (ILP) rely on a formal definition of the search space. We compare the two approaches under several dimensions, namely, expressiveness of the hypothesis language underlying the space, completeness of the search, space traversal techniques, implemented systems and applications thereof.

Next, the chapter focuses on the issues related to a more principled approach to induction as a search. Given a formal definition and characterization of the search space, we describe the main techniques employed for its traversal: strategies and heuristics, priors used to restrict its size, optimized sequential search algorithms, as well as stochastic and parallel ones.

The theoretical issues presented and exposed are complemented by descriptions of and references to implemented and applied systems, as well as real-world application domains where ILP systems have been successful.

A Historical Road Map

Concept learning is a research area of machine learning that addresses the automation of the process of finding (inducing) a description of a concept (called the ‘target concept’ or hypothesis) given a set of instances of such concept. Concepts and instances have to be expressed in a concept description language or hypothesis language. Given a set of instances of some concept it is usually a rather difficult problem to (automatically) induce the ‘target concept’. The major difficulty is that there may be a lot, if not an infinite, number of plausible conjectures (hypotheses) that are ‘consistent’ with the given instances. Automating the induction process involves the generation of the candidate concept descriptions (hypotheses), their evaluation, and the choice of ‘the best one’ according to some criteria. The concept learning problem is often mapped into a search problem, that looks for a concept description that explains the given instances and lies within the concept description language.

An important step towards the automation of the learning process is the structuring of the elements of the hypothesis language in a manner that makes it possible to perform systematic searches, to justifiably discard some ‘uninteresting’ regions of candidate descriptions, and to have a compact description of the search space.

For this purpose, machine learning borrows from mathematics the concept of a lattice, a partially ordered set with the property that all of its non-empty finite subsets have both a supremum (called join) and an infimum (called meet). A semilattice has either only a join or only a meet. The partial order can be any reflexive, anti-symmetric, and transitive binary relation. In machine learning a lattice is usually defined as follows:

Definition 1: A lattice is a partially ordered set in which every pair of elements a, b has a greatest lower bound (glb, a ∩ b) and least upper bound (lub, a ∪ b).

A PRAGMATIC APPROACH: VERSION SPACES

This section is structured into three parts. A first part presents a set of definitions and concepts that lay the foundations for the search procedure into which induction is mapped. In a second part we mention briefly alternative approaches that have been taken to induction as a search procedure and finally in a third part we present the version spaces as a general methodology to implement induction as a search procedure.
Related Content

Understanding Phatic Aspects of Narrative when Designing Assistive and Augmentative Communication Interfaces
Benjamin Slotznick (2014). International Journal of Ambient Computing and Intelligence (pp. 75-94).
www.igi-global.com/article/understanding-phatic-aspects-of-narrative-when-designing-assistive-and-augmentative-communication-interfaces/147384?camid=4v1a

Blue Ocean Strategy: A Necessity Prescription for Companies
www.igi-global.com/chapter/blue-ocean-strategy/178300?camid=4v1a

Product Design Generation and Decision Making Through Strategic Integration of Evolutionary Grammars and Kano Model
www.igi-global.com/article/product-design-generation-decision-making/69389?camid=4v1a

Clustering Analysis and Algorithms
www.igi-global.com/chapter/clustering-analysis-algorithms/24291?camid=4v1a