Preface

The field of Probabilistic Logic Programming (PLP) was started in the early 1990s by seminal works such as those of [Dantsin, 1991], [Ng and Subrahmanian, 1992], [Poole, 1993b], and [Sato, 1995].

However, the problem of combining logic and probability has been studied since the 1950s [Carnap, 1950; Gaifman, 1964]. Then the problem became prominent in the field of Artificial Intelligence in the late 1980s to early 1990s when researchers tried to reconcile the probabilistic and logical approaches to AI [Nilsson, 1986; Halpern, 1990; Fagin and Halpern, 1994; Halpern, 2003].

The integration of logic and probability combines the capability of the first to represent complex relations among entities with the capability of the latter to model uncertainty over attributes and relations. Logic programming provides a Turing complete language based on logic and thus represents an excellent candidate for the integration.

Since its birth, the field of Probabilistic Logic Programming has seen a steady increase of activity, with many proposals for languages and algorithms for inference and learning. The language proposals can be grouped into two classes: those that use a variant of the Distribution Semantics (DS) [Sato, 1995] and those that follow a Knowledge Base Model Construction (KBMC) approach [Wellman et al., 1992; Bacchus, 1993].

Under the DS, a probabilistic logic program defines a probability distribution over normal logic programs and the probability of a ground query is then obtained from the joint distribution of the query and the programs. Some of the languages following the DS are: Probabilistic Logic Programs [Dantsin, 1991], Probabilistic Horn Abduction [Poole, 1993b], PRISM [Sato, 1995], Independent Choice Logic [Poole, 1997], pD [Fuhr, 2000], Logic Programs with Annotated Disjunctions [Vennekens et al., 2004], ProbLog [De Raedt et al., 2007], P-log [Baral et al., 2009], and CP-logic [Vennekens et al., 2009].

Instead, in KBMC languages, a program is seen as a template for generating a ground graphical model, be it a Bayesian network or a Markov network. KBMC languages include Relational Bayesian Network

[Jaeger, 1998], CLP(BN) [Costa et al., 2003], Bayesian Logic Programs [Kersting and De Raedt, 2001], and the Prolog Factor Language [Gomes and Costa, 2012]. The distinction among DS and KBMC languages is actually non-sharp as programs in languages following the DS can also be translated into graphical models.

This book aims at providing an overview of the field of PLP, with a special emphasis on languages under the DS. The reason is that their approach to logic-probability integration is particularly simple and coherent across languages but nevertheless powerful enough to be useful in a variety of domains. Moreover, they can be given a semantics in purely logical terms, without necessarily resorting to a translation into graphical models.

The book doesn't aim though at being a complete account of the topic, even when restricted to the DS, as the field has grown large, with a dedicated workshop series started in 2014. My objective is to present the main ideas for semantics, inference, and learning and to highlight connections between the methods.

The intended audience of the book are researchers in Computer Science and AI that want to get an overview of PLP. However, it can also be used by students, especially graduate, to get acquainted with the topic, and by practitioners that would like to get more details on the inner workings of methods.

Many examples of the book include a link to a page of the web application `cplint` on SWISH (http://cplint.eu) [Riguzzi et al., 2016a; Alberti et al., 2017], where the code can be run online using `cplint`, a system we developed at the University of Ferrara that includes many algorithms for inference and learning in a variety of languages.

The book starts with Chapter 1 that presents preliminary notions of logic programming and graphical models. Chapter 2 introduces the languages under the DS, discusses the basic form of the semantics, and compares it with alternative approaches in PLP and AI in general. Chapters 3 and 4 describe the semantics for more complex cases, the first of languages allowing function symbols and the latter allowing continuous random variables. Chapter 5 presents various algorithms for exact inference. Lifted inference is discussed in Chatper 6 and approximate inference in Chapter 7. Non-standard inference problems are illustrated in Chapter 8. Then Chapters 9 and 10 treat the problem of learning parameters and structure of programs, respectively. Chapter 11 presents some examples of use of the system `cplint`. Chapter 12 concludes the book discussing open problems.