FAIR2019 | PROGRAMME | TUTORIALS

Tuesday 3 December 2019
Venue: Snape Lecture Theatre 1, University of Cape Town, Upper Campus

09:30 Coffee & registration

10:00 – 12:00 Tutorial 1 | Knowledge Compilation: Principles and Applications
Presented by Adnan Darwiche, University of California, USA

12:00 Lunch

13:00 – 15:00 Tutorial 2 | Build, Train and Deploy A Machine Learning Model using Amazon SageMaker
Presented by Christian Kamwangala, Cloud Engineer, Amazon Web Services

15:00 Coffee

15:30 – 18:00 Tutorial 3 | Towards Commonsense Reasoning over Ontologies
Presented by Ivan Varzinczak, Artois University & CNRS, France

Tutorial 1 | Knowledge Compilation: Principles and Applications

Presented by Adnan Darwiche, University of California, USA

Knowledge compilation (KC) is a research area which aims to preprocess information to improve the time required to solve highly-demanding computational tasks (NP and Beyond NP problems). Pioneered more than two decades ago, KC is nowadays a very active field, being at the intersection of several areas of AI and computer science. I will discuss some key dimensions relating to KC: (1) the choice of a tractable language to compile into, which depends on its degree of tractability (operations it supports in polytime) and its succinctness (space efficiency of its representations); (2) the design of knowledge compilers; and (3) the applications of KC within AI, including probabilistic inference, machine learning and explanations.

Tutorial 2 | Build, Train and Deploy A Machine Learning Model using Amazon SageMaker

Presented by Christian Kamwangala, Cloud Engineer, Amazon Web Services

This tutorial aims to introduce Amazon SageMaker, a fully managed machine learning service that allows AWS to put machine learning in the hands of every developer, data scientists and researcher. With Amazon SageMaker, developers can quickly and easily build and train their machine learning models, then directly deploy them into a production-ready hosted environment. It provides an integrated Notebook Instance for authoring Jupytet Notebooks with easy access to your data sources for exploration and analysis, meaning you don't have to manage servers. It also provides prebuilt machine learning algorithms for common use case that are optimized to run efficiently against extremely large data in a distributed environment. With native support for bring-your-own algorithms and deep learning frameworks, Amazon SageMaker offers flexible distributed training options that adjust to your specific workflows. Deploy a model into a secure and scalable environment by launching it with a single click from the Amazon SageMaker console. Training and hosting are billed by minutes of usage, with no minimum fees and no upfront commitments.

The goal of this tutorial is to use Amazon SageMaker to demonstrate the entire machine learning lifecycle: label and prepare your data, choose an algorithm, train the model, tune and optimize it for deployment and make predictions, and take action. In this tutorial, we will use a publicly available dataset to train a machine learning model that is able to classify a given image into one of 5 categories or classes: berry, bird, dog, flower, and other (negative set). We will also explore the model optimization techniques and deployment strategies that are available in SageMaker.

Tutorial 3 | Towards Commonsense Reasoning over Ontologies

Presented by Ivan Varzinczak, Artois University & CNRS, France

This tutorial aims at providing an introduction to reasoning defeasibly over description logic ontologies in the context of knowledge representation and reasoning (KRR) in AI. Description Logics (DLs) are a family of logic-based knowledge representation formalisms with appealing computational properties and a variety of applications at the confluence of modern artificial intelligence and other areas. In particular, DLs are well-suited for representing and reasoning about ontologies and therefore constitute the formal foundations of the Semantic Web.

The different DL formalisms that have been proposed in the literature provide us with a wide choice of constructors in the object language. However, these are intended to represent only classical, unquestionable knowledge, and are therefore unable to express the different aspects of uncertainty and vagueness that often show up in everyday life. Examples of these comprise the various guises of exceptions, typicality (and atypicality), approximations and many others, as usually encountered in the different forms of human quotidian reasoning. A similar argument can be put forward when moving to the level of entailment, that of the sanctioned conclusions from a knowledge base. DL systems provide for a variety of (standard and non-standard) reasoning services, but the underlying notion of entailment remains classical and therefore, depending on the application one has in mind, DLs inherit most of the criticisms raised in the development of the so-called non-classical logics. In this regard, endowing DLs and their associated reasoning services with the ability to cope with defeasibility is a natural step in their development. Indeed, the past two decades have witnessed the surge of many attempts to introduce non-monotonic reasoning capabilities in a DL setting. Among these are default extensions, preferential approaches, circumscription-based ones, and others..

The goal of this tutorial is two-fold: (1) to provide an overview of the development of non-monotonic approaches to description logics from the past 25 years, in particular pointing out the difficulties that arise when naively transposing the traditional propositional approaches to the DLcase, and (2) present the latest results in the area, in particular those based on the preferential approach and related ones, as well as the new directions for investigation that have been opened.