Cognitive Automation of Data Science

ICML AutoML Workshop 2014

Horst Samulowitz[1]
Ashish Sabharwal[2]
Chandra Reddy[1]

When is an algorithmic framework *cognitive*?

1. <u>Integrates knowledge</u> **from (a) structured/unstructured sources, (b) past experience, and (c) current state, in order to reason with this knowledge as well as to adapt over time**
2. <u>Interacts</u> **with the user (e.g., through natural language or visualization) and reasons based on such interactions**
3. **Can generate <u>novel hypotheses and capabilities</u>, and test them**

Goal: Improve the Automated Data Science Processing Pipeline

- **Machine Learning Packages** such as WEKA offer a vast range of techniques that cover all aspects of data analytics starting with data processing and ending with performing predictions
- **Tools such as Auto-Weka** automate selection and configuration of machine learning tools mainly by purely data-driven methods and past experience
- **Cognitive Automation** tries to leverage additional sources of knowledge and support interactivity and computational creativity by:
  - **Understanding** basic properties of the underlying approaches
  - **Incorporating knowledge** from experts (e.g., common practices), structured and unstructured data (e.g., WEKA manual, ML papers), etc.
  - **Integrating End-User Objectives**
  - **Performing informed (re-)actions** based on observations and computational creativity
  - **Interaction with user to guide process**

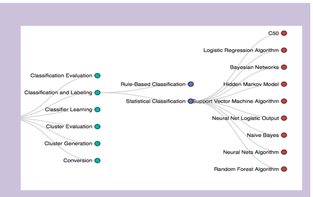
Examples

- **Basic yet Informed Reasoning:**
  - Automatically detect overfitting and **make informed decisions** based on it such as increasing neighborhood size in a nearest neighbor method.
  - **Directly take into account end-user constraints** (e.g., limited training time or understandable model) when searching for a suitable approach.

- **Unstructured data**
  - Suppose one wanted to use k-NN on a data set with discrete features. The system using web-scale NLP technology could search for “distance measure k-nn discrete data” and **automatically extract the knowledge** stated on a Wikipedia page: “A commonly used distance metric for continuous variables is Euclidean distance. For discrete variables, such as for text classification, another metric can be used, such as the overlap metric (or Hamming distance).” It could thus begin to “know” and exploit this knowledge in order to configure k-NN appropriately.
- **Interactivity**
  - In the above example, suppose only the Euclidean distance metric is provided by the user. The automated system should be able to exploit the Wikipedia knowledge and **interactively suggest** to the user that a Hamming distance metric may be more suitable.

- **Creativity**
  - In deep learning one recently introduced technique called “DropOut” randomly selects subsets of node activations in each layer and sets them to zero. Subsequently, a technique called “DropConnect” was introduced that instead sets a randomly selected subset of edge weights (or connections) within the network to zero. A cognitive automated system for ML could possibly **automatically explore such novel extensions** as DropConnect given knowledge about DropOut, as well as combinations of the individual approaches based on some fundamental knowledge on how the underlying approaches work.

Where to start? Develop system in stages



- **Extend purely data-driven methods with semantic knowledge**
- **Build on existing algorithm ‘cook books’ and structure**
  - Properties such as cost, input/output requirements, guarantees
  - Hierarchical structure developed by experts reveals grouping of techniques
  - Annotated by expert-knowledge whether a technique is applicable