Implementing and evaluating Sum-Product networks in Python

1 Description

Sum-Product Networks (SPNs) [1, 2] are a new type of deep learning paradigm that combines features from traditional probabilistic graphical models [3, 4, 5] and new developments in deep networks. Relevant information about SPNs is available from `http://spn.cs.washington.edu/`.

2 Objectives

The goal of the project is to implement, in Python, an algorithm for learning an SPN and test it as a classifier. The student is free to decide which classification problems will be used to test the implementation. Classification problems described in [2] could be used as a reference.

As in other projects, a report should describe the characteristics of the design, implementation, and results. A Jupyter notebook should include calls to the implemented function that illustrate the way it works.

3 Suggestions

- Implementations can use any Python library (e.g., scikit-learn, tensorflow, keras).
- Previous implementations of SPNs in Java and C++ can be used as a reference. However, be sure that these implementations are correct.
- The learning methods proposed in [1, 2] could serve as a reference.

References

[1] Hoifung Poon and Pedro Domingos. Sum-product networks: A new deep architecture. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 689–690. IEEE, 2011.

[2] Robert Gens and Domingos Pedro. Learning the structure of sum-product networks. In *International Conference on Machine Learning*, pages 873–880, 2013.

[3] E.M. Airoldi. Getting started in probabilistic graphical models. *PLoS Computational Biology*, 3(12):e252, 2007.

[4] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

[5] P. Larrañaga. *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*, chapter An introduction to probabilistic graphical models, pages 25–54. Kluwer Academic Publishers, Boston/Dordrecht/London, 2002.