Extending Mocapy++ with a mixed probability distribution
Advanced Topics In Data Modeling

Kasper Nybo Hansen
Dept. of Computer Science
University of Copenhagen
Copenhagen, Denmark
nybo@diku.dk

***Abstract*—Mocapy++ is a C++ toolkit for learning and inference in dynamic Bayesian networks. This report describes the implementation, testing and results of extending Mocapy++ with a new node.**

The new node is a mixed node, allowing both discrete and continuous values. The continuous part of the node is a gaussian distribution. The new node is used to calculate a probabilistic model of hydrogen bonding in protein structures. The probabilistic model is learned from a provided dataset.

***Index Terms*—Mocapy++; DIKU; Dynamic Bayesian networks; Mixed probability distribution**

I. Introduction

An introduction with a short discussion of the theory of inference and learning in Bayesian networks, relevant to Mocapy++.

When using the mixed node in protein bondings, the following

A. Mixed distribution

The mixed distribution can be divided into two parts. A discrete and continuous part. Let X be a random variable that takes values in the set S. We then define the discrete part as the countable set $D \subseteq S$, and the continuous part as $C \subseteq S$. We define a mixed distribution as a distribution that has the following two probabilities

- $0 < P(x \in D) < 1$
- $P(x \in C) = 0$

II. Implementation

A description of the implementation of the mixed distribution in Mocapy++.

We assume that the parent node can only take one value

A. ESS

B. Densities

CPD = 2*parent size, each parent can yield a value, the indicator shows if it is continuous or discrete

III. Testing of implementation

A description of some simple tests that show that the implementation is correct.

A. Test of inference

B. Test of sampling

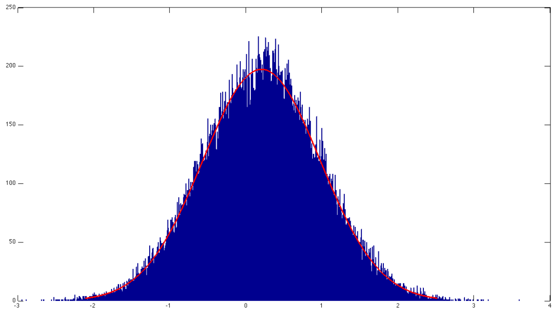

Figure 1 – Illustration of the samples drawn from the Gaussian part of the mixed distribution

C. Test of save/load date

IV. Results

A presentation and discussion of the results obtained from its application to the protein dataset.

V. Conclusion

VI. Improvements

The present implementation does not enable the user to specify which distributions should be part of the mixed node. Future work could involve making use of the existing distributions and using them in the mixed node. This would remove the duplicate code that is present in the current prototype of the mixed node, and make the mixed node more flexible.

References

[1] C. Igel and K. S. Pedersen. Statistical methods for machine learning. final exam assignment. recognition of traffic signs. March 2011.