Contents

Document history

Version	Date	Author (Unit)	Description
v0.3	21/11 2014	Helge Langseth, Thomas D. Nielsen, Antonio Salmerón	First draft

1 Executive summary

The aim of this document is to describe the progress of the software development related to learning in the AMIDST project at the end of the first year.

2 HL – Introduction

[[This is just the first couple of pages from D3.1, terminated as quickly as possible, basically. Need to define BNs and such stuff here...]]

Probabilistic graphical models provide a well-founded and principled approach for performing inference in complex domains endowed with uncertainty. A probabilistic graphical model is a framework consisting of two parts: a qualitative component in the form of a graphical model encoding conditional independence assertions about the domain being modelled as well as a quantitative component consisting of a collection of local probability distributions adhering to the independence properties specified in the graphical model. Collectively, the two components provide a compact representation of the joint probability distribution over the domain being modelled.

Bayesian networks (BNs) [**?**] are a particular type of probabilistic graphical model that has enjoyed widespread attention in the last two decades. Figure 1 shows a BN representing the joint distribution of variables $X_1, \ldots, X_5$. Attached to each node, there is a conditional probability distribution given its parents in the network, so that the joint distribution factorises as

$$p(X_1, \ldots, X_5) = p(X_1)p(X_2|X_1)p(X_3|X_1)p(X_4|X_2, X_3)p(X_5|X_3).$$

In general, for a BN with n variables $\mathbf{X} = \{X_1, \ldots, X_N\}$, the joint distribution factorises as

$$p(\mathbf{X}) = \prod_{i=1}^{N} p(X_i|\text{pa}\,(X_i)), \tag{1}$$

where $\text{pa}\,(X_i)$ denotes the set of parents of X_i in the network.

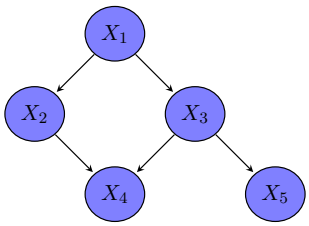

Figure 1: A Bayesian network with five variables.

We will use lowercase letters to refer to values or configurations of values, so that x denotes a value of X and $\mathbf{x}$ is a configuration of the variables in $\mathbf{X}$. Given a set of observed variables $\mathbf{X}_E \subset \mathbf{X}$ and a set of variables of interest $\mathbf{X}_I \subset \mathbf{X} \setminus \mathbf{X}_E$, *probabilistic inference* consists of computing the posterior distribution $p(x_i|\mathbf{x}_E)$ for each $i \in I$. A

thorough introduction to the state of the art for inference in Bayesian networks was given in [**?**]. In this document we assume that inference techniques for a given Bayesian network is available, and will rather consider how to define a Bayesian network model that fits a data set $\mathcal{D}$ as good as possible?

- Learning parameters
- Learning structure

3 HL – Design

[[This section will cover the design of the SW, as specified through the requirements engineering document. Only the parts related to learning will be covered.]]

4 HL – Learning as inference

Say that we recommend the use of a fully Bayesian approach. Approximate inference techniques (VB / EP, importance sampling, etc) can be optimized for the model class. Maximum likelihood based approaches do not go well hand in hand with approximate inference techniques...

Parameter learning in Bayesian networks generally comes in two different shapes:

Maximum likelihood learning attempts to find the model parameters that maximizes the likelihood. Let $\boldsymbol{\theta}$ denote the combination of all parameters of the model. Then $\mathcal{L}(\boldsymbol{\theta}|\mathcal{D}) = P(\mathcal{D}|\boldsymbol{\theta})$ is the *likelihood* of the parameters given the data set $\mathcal{D}$, and the maximum likelihood estimator is $\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}|\mathcal{D})$. Maximum likelihood learning in a model with missing data or latent variables (that are present in the AMIDST model class) is typically implemented using the Expectation Maximization (EM) algorithm or generalizations thereof. Implementations of the (generalized) EM algorithm typically iterates over the following two steps that are repeated until convergence: *i*) E-step: Inference in the model given parameter estimates; *ii*) M-step: Updates of the parameters using inferred states of the variables not observed in the dataset. The EM algorithm is a greedy algorithm that at each iteration guarantees that the likelihood of the present parameter estimates is not lower than the likelihood of the previous estimate as long as the inference algorithm employed by the E-step is exact. Convergence is therefore monitored by keeping track of the likelihood function. If approximate inference is used, no such guarantees exist. OR AM I MISTAKEN? `http://papers.nips.cc/paper/2404-approximate-expectation-maximization.pdf` looks interesting.

Bayesian learning hyper-parameters. Conjugate exponential model restriction. Learning as inference. "Dirty" inference is not a problem for convergence of learning (converges

if and only if inference algo converges, obviously).

5 AS – Task 4.1: Parallelization of structural learning

This section will give updates from Task 4.1, with focus on the actual tool-box implementation of parallel PC. We have two sources of information/ideas how to proceed:

- The slideset/poster we discussed some time ago.
- Hugin people are investigating lines for parallelizing the PC using multi-thread and relying on the TAN-PGM paper.

We should also discuss the learning of TAN classifiers (the PGM paper). Anders has suggested to Antonio that we could include the PGM paper here (except maybe the experiments, that could be reported in D5.1). In this way there would be no rush for having it implemented in the toolbox.

The setup of this section is that we show in practice how the initial procedures described in the design section is to be used.

6 HL – Initial results from the other tasks

[[This section will discuss any results or ideas that have been developed in Tasks 4.2, 4.3, 4.4. Not much to include, as the tasks only start at December 1st. Antonio proposes to only take in what has come out of the requirement analysis document.]]