International Journal of Approximate Reasoning 69 (2016) 58–80

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

Gated Bayesian networks for algorithmic trading

Marcus Bendtsen *, Jose M. Peña

Department of Computer and Information Science, Linköping University, Sweden

ARTICLE INFO

Article history:
Received 5 January 2015
Received in revised form 2 November 2015
Accepted 5 November 2015
Available online 11 November 2015

Keywords:
Probabilistic graphical models
Bayesian networks
Algorithmic trading
Decision support

ABSTRACT

This paper introduces a new probabilistic graphical model called *gated Bayesian network* (GBN). This model evolved from the need to represent processes that include several distinct phases. In essence, a GBN is a model that combines several Bayesian networks (BNs) in such a manner that they may be active or inactive during queries to the model. We use objects called *gates* to combine BNs, and to activate and deactivate them when predefined logical statements are satisfied. In this paper we also present an algorithm for semi-automatic learning of GBNs. We use the algorithm to learn GBNs that output buy and sell decisions for use in *algorithmic trading* systems. We show how the learnt GBNs can substantially lower risk towards invested capital, while they at the same time generate similar or better rewards, compared to the benchmark investment strategy *buy-and-hold*. We also explore some differences and similarities between GBNs and other related formalisms.

1. Introduction

Bayesian networks (BNs) can be interpreted as models of causality at the macroscopic level, where unmodelled causes add uncertainty. Cause and effect are modelled using random variables that are placed in a directed acyclic graph (DAG). The causal model implies some probabilistic independencies among the variables, that can easily be read off the DAG. Therefore, a BN does not only represent a causal model but also an independence model. The qualitative model can be quantified by specifying certain marginal and conditional probability distributions so as to specify a joint probability distribution, which can later be used to answer queries regarding posterior probabilities, interventions, counterfactuals, etc. The independencies represented in the DAG make it possible to compute these posteriors efficiently. Furthermore, they reduce the number of parameters needed to represent the joint probability distribution, thus making it easier to elicit the probability parameters needed from experts or from data. See [1–3] for more details.

A feature of BNs, known as the local Markov property, implies that a node is independent of all other non-descendent nodes given its parent nodes, where the relationships are defined with respect to the DAG of the BN. If we define the parents of X_i as $Parents(X_i)$, the local Markov property allows us to factorise the joint probability distribution according to Equation (1).

$$p(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} p(X_i|Parents(X_i)) \quad (1)$$

* Corresponding author.
E-mail addresses: marcus.bendtsen@liu.se (M. Bendtsen), jose.m.pena@liu.se (J.M. Peña).

http://dx.doi.org/10.1016/j.ijar.2015.11.002
0888-613X/

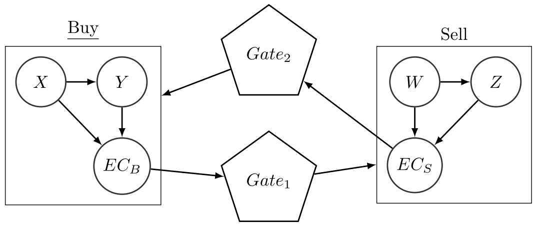

Fig. 1. GBN using two phases.

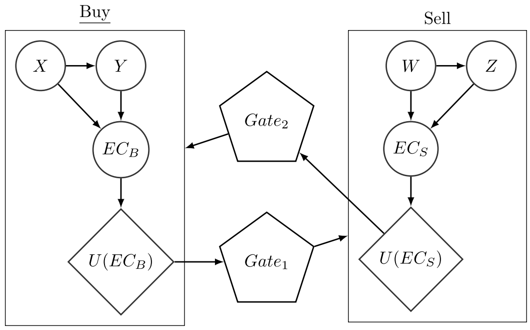

Fig. 2. GBN using utility nodes.

Despite their popularity and advantages, there are situations where a BN is not enough. For instance, when trying to model the process of a trader buying and selling stock shares, we wanted a model that switched between identifying buying opportunities and then, once such have been found, identifying selling opportunities. The trader can be seen as being in one of two distinct phases: either looking for an opportunity to buy shares and enter the market, or an opportunity to sell shares and exit the market. These two phases can be very different and the variables included in the BNs modelling them are not necessarily the same. Dynamic BNs have traditionally been used to model temporal processes, and as their name suggests, they model the dynamics among variables between typically equally spaced time steps. However, processes that entail different models at different phases, and where the transition between phases depends on the observations made, are not easily captured by dynamic BNs, as they assume the same static network at each time step. The need to switch between different BNs was the foundation for the probabilistic graphical model presented herein, which we call gated Bayesian networks (GBNs). In Fig. 1 we present a GBN that uses two different BNs (*Buy* and *Sell*). In Section 2.2 we will explain how decisions can be connected to the phase changes of a GBN, we will specifically show how buy and sell decisions are connected to the phase changes for the GBN in Fig. 1. It should however be noted that we will not always connect a phase change with a decision, as there will be an example of in Section 3.2. Sometimes a phase change is needed in order to use a different BN without any explicit decision connected to it.

Intuitively, a GBN makes explicit the possible transitions between the contained models, i.e. the phases, along with the driving variables in these phases. This is not only advantageous from a representational point of view, but since constraints are encoded in the model, parameter learning will be influenced by these constraints. For instance, when a transition from the *Sell* BN in Fig. 1 should occur will be dependent on when a transition from the *Buy* BN occurs, as one must happen before the other. Imagining two experts, where one gives recommendations of when to buy assets and the other when to sell assets, we would want the experts to work well together. If the first expert has a long-term view and the second expert has a short-term view, then recommendations to buy will be far apart, but as the second expert assumes that we are after short-term profits, sell recommendations come quickly after we have bought the assets. In extreme cases, this may end up in a strategy where over a year the assets are only held for a few hours. Thus, the fact that buying and selling places constraints on each other must be captured by the model, and single BNs are not able to encode these constraints.

The example of the trader is really a simplification of a more complex process known as *algorithmic trading*, which we will describe in more detail in the coming section. Our primary intention is to use GBNs as part of algorithmic trading, however for clarity, we will sometimes fall back to the more simple view of a single trader in this paper.

1.1. Algorithmic trading

Formally, the process we intend to model is part of a larger process commonly referred to as algorithmic trading. Algorithmic trading can be viewed as a process of actively deciding when to own assets and when to not own assets, so as to get better *risk* and *reward* on invested capital compared to holding on to the assets over a long period of time. At the other end of the spectrum is the *buy-and-hold* strategy, where one owns assets continuously over a period of time without making any decisions of selling or buying during the period.