Knowledge-Based Systems 58 (2014) 113–126

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Dynamic exploration designs for graphical models using clustering with applications to petroleum exploration

Gabriele Martinelli *, Jo Eidsvik

Dept. of Mathematical Sciences, Norwegian University of Science and Technology, Alfred Getz' vei 1, Trondheim, Norway

ARTICLE INFO

Article history:
Available online 28 August 2013

Keywords:
Bayesian networks
Markov random fields
Design
Sequential design
Dynamic programming
Gittins index
Petroleum exploration

ABSTRACT

The paper considers the problem of optimal sequential design for graphical models. Oil and gas exploration is the main application. Here, the outcomes at prospects or reservoir units are highly dependent on each other. The joint probability model for all node variables is considered known. As data is collected, this probability model is updated. The sequential design problem entails a dynamic selection of nodes for data collection, where the goal is to maximize utility, here defined via entropy or total expected profit. With a large number of nodes, the optimal solution to this selection problem is not tractable. An approximation based on a subdivision of the graph is considered. Within the small clusters the design problem can be solved exactly. The results on clusters are combined in a dynamic manner, to create sequential designs for the entire graph. The merging of clusters also gives upper bounds for the actual utility. Several synthetic models are studied, along with two real cases from the oil and gas industry. In these examples Bayesian networks or Markov random fields are used. The sequential model updating and data collection strategies provide useful guidelines to policy makers.

1. Introduction

Our interest is a sequential selection problem over dependent variables. The main motivation is to construct policies for oil and gas exploration, where the outcomes at prospects are dependent by spatial proximity or by common geological mechanisms. The probability of success for any exploration well is then highly influenced by the outcomes at other prospects.

More generally the challenge is to construct an optimal dynamic design of nodes in a graph. For instance, in the situation with a Bayesian Network (BN) or a Markov Random Field (MRF) we evaluate which variables are most useful to observe. We assume a fixed probability model a priori. As we acquire data at nodes in the BN or the MRF, the original probability distribution is updated, according to Bayes rule. Relevant design questions are then: Which nodes are more informative? Which sequence of nodes gives the best policy? In the petroleum industry drilling wells is extremely costly, and getting the right information is critical.

At each stage of the dynamic strategy, we choose to observe one additional variable, or quit the search. If we acquire data at a node, we incorporate the observation in the current (a priori) model to compute the updated (a posteriori) model. For the next stage, the updated model serves as a prior model, and so on. The sequential decisions account for two aspects: (i) the immediate profit in terms of monetary units or information gain by knowing the current variable and (ii) the expected future benefits induced by the predictive capacity, conditional on the current variable. These two aspects are combined in a utility function. If the expected utility of choosing one more node is too small, we stop collecting data. The trade off between (i) and (ii) is related to more general explore or exploit problems in decision making. An oil and gas company may want to target the most lucrative prospects, but it is also important to know the key variables, which give us the chance to make better, informed, decisions at the later stages. The future values in (ii) then play an important role in the utility function.

With our focus on oil and gas exploration we note some similarities and differences with common spatial design problems, e.g. Shewry and Wynn [27], Le and Zidek [18], and Zidek and Zimmerman [32]. The most common problem treated in the literature is to allocate a fixed number of monitoring stations to improve overall predictive performance in some sense. The selection is thus done in the static manner, not allowing the decision maker to modify her choices after observing the outcomes at the previously selected spatial sites. In this paper we consider the dynamic decision problem, with one observation at a time and the ability to make sequential decisions. Moreover spatial design problems the model typically rely on Gaussian models. The current paper studies graphical models with discrete outcomes at all nodes.

Our sequential design problem is a discrete optimization problem which is in theory solved via Dynamic Programming (DP). This

* Corresponding author. Tel.: +47 21609638 (O).
E-mail addresses: gabriele.martinelli@math.ntnu.no, gmartinelli@slb.com (G. Martinelli), joeid@math.ntnu.no (J. Eidsvik).

0950-7051/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.knosys.2013.08.020

method defines a forward–backward algorithm that constructs the optimal sequences and the expected utility. Bickel and Smith [6] present a DP algorithm tailored for our sequential design problem with dependent oil and gas prospects. However, their approach is not applicable when the number of variables gets too large. For more than, say, 10 variables, we must instead look for approximate strategies. The appropriate solution seems to be very case-specific. See e.g. Powell [24] for more background. Various heuristic approaches are important for special applications, but it is very difficult to assess the properties of these solutions. For graphical models it seems natural to utilize the structure. One approach is to split the original graph in several disjoint clusters. This clustering idea was originally presented in Brown and Smith [7], who solved the DP exactly for clusters, and combined the results to approximate the expected utilities on the full-size graph, with upper bounds.

Our main contribution in this paper is to use the clustering strategies to construct sequential designs for BNs and MRFs. A critical element in the method is to compute the cluster-wise Gittins index. This extends the original index pioneered by Gittins [12] and Whittle [31] for so-called bandit problems, and studied by Benkerhouf et al. [3] and Glazebrook and Boys [13] for oil and gas exploration problems. We consider the sensitivity of cluster orientation and size, and various levels of approximation in the Bayes updating scheme. We use utility functions based on entropy and more traditional cost/revenue aspects. The resulting sequential prospect designs can work as a road map for the petroleum exploration company. The presented methods are relevant for e.g. machine scheduling [2], medical treatments selection [9], real-time strategy games [23], subset selection problems and more generic search problems. The use of lattices and networks in decision making has been successfully applied in decision theory, see Fenton and Neil [11]. Finally, strategies based on clustering are typical in many fields, when dealing with classification problems: for recent cluster-based algorithms see Jain [14] and Karaboga and Ozturk [15].

The paper develops in the following way: in Section 2 we give the main ideas about sequential design, in Section 3 we discuss how the splitting in clusters can help in building approximate strategies, in Section 4 we provide results on synthetic examples, in Section 5 we show results on real case studies.

2. Sequential design

A sequential strategy is illustrated in Fig. 1. Here, we initially choose to drill one of three petroleum prospects, or nothing. If we start by drilling prospect 3, the design criterion for the next stage depends on the outcome of prospect 3. The decision is then to choose among prospect 1 and 2, or quit.

We first introduce the statistical notation and assumptions required to frame this sequential design problem. We then outline the theoretical solution given by DP. A small example is used to illustrate the sequential strategies resulting from different utility functions.

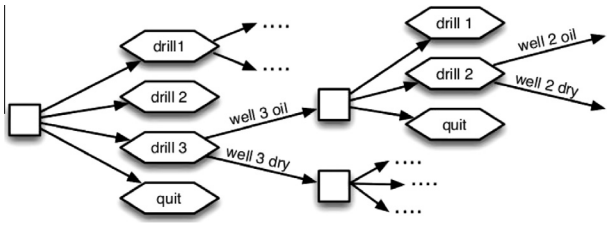

Fig. 1. Decision tree for a simple 3-nodes discrete example with two possible outcomes (*oil* or *dry*) per node.

2.1. Notation and modeling assumptions

Consider N nodes, and let $x_i \in \{1,\ldots,k_i\}$, $i = 1,\ldots,N$ denote the discrete random variables. Without loss of generality, we assume $k_i = k$ possible states for all nodes i. In Fig. 1, $k = 2$ with oil or dry outcomes. We represent the probabilistic structure for $\mathbf{x} = (x_1,\ldots,x_N)$ via a graph. For a BN defined by a directed acyclic graph the joint distribution is

$$p(\mathbf{x}) = \prod_{i=1}^{N} p(x_i|x_{\mathrm{pa}(i)}), \tag{1}$$

where pa(i) denotes the parent set of node i, which is empty for the top nodes. Undirected graphs are defined via the full conditionals over a neighborhood, or, by the Hammersley–Clifford theorem, via a joint distribution over clique potentials. For a first-order MRF [4] we use:

$$p(\mathbf{x}) \propto \exp\left\{\beta \cdot \sum_{i\sim j} \mathbb{I}(x_i = x_j) + \sum_{i=1}^{N} \alpha_i(x_i)\right\}, \tag{2}$$

where $i \sim j$ denotes neighboring lattice nodes (north, east, south, and west). The parameter β imposes spatial interaction, while the $\alpha_i(x_i)$ terms include prior preferences about states at node i.

We assume known, fixed, statistical model parameters in $p(\mathbf{x})$, such as β and $\alpha_i(x_i)$ in (2) and the conditional probabilities in (1). Associated with the probabilistic model we can of course compute several attributes that are important for design purposes. Assuming that we know the revenues or cost, denoted r_i^j, for outcomes $x_i = j$, the decision value is $DV(i) = \max\left(0, \sum_{j=1}^{k} r_i^j p(x_i = j)\right)$, $i = 1,\ldots,N$. This DV is useful for decision making. It is non-zero only when the expected profit is positive. The entropy (disorder) is defined by $H = -\sum \log(p(\mathbf{x}))p(\mathbf{x}) = -E(\log p(\mathbf{x}))$, see e.g. Wang and Suen [29].

In our sequential design situation, we rely on the ability to extract the marginal probabilities at all nodes, and to update the probability distributions when evidence is collected. Since we are going to update the model at each stage of the sequential strategy, for many different kinds of evidence, we require these computations to be reasonably fast. For BNs the updating of probabilities can be done effectively by the junction tree algorithm [17]. MRFs can similarly be updated by forward–backward algorithms, see e.g. Reeves and Pettitt [26] and Tjelmeland and Austad [28].

Assume we can acquire data at one node in the graph, and incorporate the outcome to get a posterior distribution. For the next stage, this updated distribution serves as a prior distribution. We can then select another node, acquire information, update the probabilities, and so on. The sequential design of nodes is constructed by optimizing the expected utility, which means that we integrate over all possible data when finding the optimal sequence. In our case, the utility is based on monetary profits or entropy reduction. One could of course imagine other selection criteria here. Minimum entropy entails a dynamic design that attempts to stabilize or minimize the uncertainty in the graph.

Let ω_i be the observable or evidence in node $i = 1,\ldots,N$. If node i is not yet observed, we set $\omega_i = -$. If we choose to observe node i, ω_i is the actual outcome of the random variable x_i at this node. For instance, in a petroleum example, $\omega_i = 1$ can mean that prospect i has been drilled and found dry, $\omega_i = 2$ if found gas, and $\omega_i = 3$ if oil. A priori, before acquiring any observables, we have $\boldsymbol{\omega} = \boldsymbol{\omega}_0 = (-,\ldots,-)$. When we observe nodes, we put the outcomes at the corresponding indices of the vector $\boldsymbol{\omega}$. Say, if node 2 is selected first, and observed in state $\omega_2 = x_2 = 1$, we set $\boldsymbol{\omega} = (-,1,-,\ldots,-)$. At each stage, one more entry of $\boldsymbol{\omega}$ is assigned. The posterior that is updated at every stage of the sequential design is generically denoted by $p(\mathbf{x}|\boldsymbol{\omega})$, with marginals $p(x_i = j|\boldsymbol{\omega})$,