Budgeted Distribution Learning of Belief Net Parameters

Liuyang Li LIUYANG@UALBERTA.CA
Barnabás Póczos POCZOS@UALBERTA.CA
Csaba Szepesvári SZEPESVA@UALBERTA.CA
Russ Greiner RGREINER@UALBERTA.CA
Department of Computing Science, University of Alberta, AB, Canada, T6G 2E8

Abstract

Most learning algorithms assume that a training dataset is given initially. We address the common situation where data is not available initially, but can be obtained, at a cost. We focus on learning Bayesian belief networks (BNs) over discrete variables. As such BNs are models of probabilistic distributions, we consider the "generative" challenge of learning the parameters for a fixed structure, that best match the true distribution. We focus on the *budgeted learning* setting, where there is a known fixed cost c_i for acquiring the value of the i^{th} feature for any specified instance, and a known total budget to spend acquiring all information. After formally defining this problem from a Bayesian perspective, we first consider *non-sequential* algorithms that must decide, before seeing any results, which features of which instances to probe. We show this is NP-hard, even if all variables are independent, then prove that the greedy allocation algorithm IGA is optimal here when the costs are uniform, but can otherwise be sub-optimal. We then show that general (sequential) policies perform better than non-sequential, and explore the challenges of learning the parameters for general belief networks in this sequential setting, describing conditions for when the obvious round-robin algorithm will, versus will not, work optimally. We also explore the effectiveness of this and various other heuristic algorithms.

Appearing in *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel, 2010.

1. Introduction

Consider the challenge of producing a Bayesian belief network (BN) (Verma & Pearl, 1991) for modeling a particular medical situation — *e.g.*, for encoding the various interactions between a given set of symptoms and diseases. Here, experts have identified the relevant symptom and disease variables $\mathbf{X} = (X_1, \ldots, X_d)$, and have structured them as nodes in a graph, whose directed arcs represent their dependencies. However, they have not specified the actual parameters (here, conditional probability table (CPtable) values, as these variables are all discrete), but have provided a prior distribution for each CPtable row $\theta_{X_i|\mathbf{u}_i}$, corresponding to the posterior distribution of the variable X_i given a specific assignment $\mathbf{u}_i$ to its parents $\mathbf{U}_i$ (Tong & Koller, 2001b).

Fortunately, there are many known techniques for estimating these parameters, *given a data sample* (Heckerman, 1999). Unfortunately, we do not initially have any data (perhaps this is just the start of a funded study). We do, however, have access to a set of patients $\{\mathbf{X}^1, \mathbf{X}^2, \ldots\}$, whose individual features we can "probe", at a cost. That is, we can acquire the value x_i^j of feature i for patient j, at known cost c_i. The total cost of the sequence of K probes $\langle x_{i_1}^{j_1}, \ldots, x_{i_K}^{j_K} \rangle$ then is $\sum_{k=1}^{K} c_{i_k}$. Our funders have provided a total budget $\mathcal{B}$ to spend on such probes; so we can consider any probe sequence where $\sum_{k=1}^{K} c_{i_k} \leq \mathcal{B}$. Our task, now, is to use this budget effectively, to (sequentially) purchase the probes that allow us to obtain good estimates of the parameters — in particular, to find the parameters that most closely match the true distribution; see Section 2. We refer to this as *Budgeted Distribution Learning* (BDL).

Section 2 provides the relevant framework. Section 3 then focuses on the simple case where the variables are independent, dealing first with "non-sequential algorithms" that must decide on all of the probes before seeing any of their responses. We provide an efficient

greedy algorithm, IGA, that provably returns the optimal allocation in the unit-cost case, then we show that this task is NP-hard when the costs are arbitrary. We also show that this optimal *allocation* is not the optimal policy, and provide empirical studies that illustrate the behaviour of this IGA algorithm, as well as several obvious sequential approaches.

Section 4 then extends these results to general belief networks. We describe conditions for when the simple round-robin algorithm (which acquires full data instances) will produce optimal results. We also explore the effectiveness of this and various other heuristic algorithms. The extended technical report (LPSzG, 2010) contains the proofs, and other material; in particular, showing that many of these results hold for Gaussian distributions, as well as Dirichlet.

We close this section by summarizing related work — in particular, placing our system within the context of active learning, and contrasting our system with discriminative budgeted learning.

1.1. Related Work

While most learning algorithms begin with a given data set, there is today a large literature on *active learning* (Muslea, 2002) and *experimental design* (Melas, 2006), which explore the challenges of first acquiring this training data. The standard example is "active discriminate label learning" (ADLL), in which the system attempts to produce a *classifier* — *i.e.*, a function that maps attributes of each instance $\mathbf{x}^j = \langle x^j_1, \ldots, x^j_n \rangle$ to a label y^j — and assumes the active learner initially has access to the attributes of a large number of instances $\{\mathbf{x}^j\}_j$, and needs to pay funds to purchase the labels.

Lizotte et al.'s (2003) "budgeted discriminative attribute learning" model (BDAL) is also seeking a good classifier; it differs from ADLL by initially providing the learner with all of the labels $\{y^j\}_j$ and allowing the learner to purchase the attribute values $\{x^j_i\}_{i,j}$ (called "probes") that it specifies. This task is potentially more difficult, as the BDAL learner has to consider dependencies among the attributes $\{x_i\}$ as well as dependencies connecting each attribute with the label y. Another complexity is that different attributes can have different costs. This work also differs from ADLL by imposing a *hard* limit on the purchases — *i.e.*, allowing the learner only a fixed budget to spend acquiring information.

Our BDL framework resembles BDAL as both are acquiring data to find the best parameters for a given belief net structure. Our goal, however, is a good *generative* model, rather than BDAL's accurate *discriminator* (that is, a good classifier). This eliminates the distinction between attribute versus label (they are all just "features"), and explains why our learner starts with no data at all (recall that BDAL assumes the learner initially has the labels $\{y^j\}_j$).

The simplest version of our BDL system deals with the trivial belief net where all variables are independent (*i.e.*, the nodes are not connected); see Section 3. This task relates directly to "bandit problems" (Berry & Fristedt, 1985), which basically uses a set of trials to identify the optimal "bandit" — *i.e.*, the "slot machine" that returns the maximal expected payout. While standard bandit problems combine exploration and exploitation (each "probe" or "trial" provides both information about the quality of the bandit played, and also a reward), our model differs as it uses its budget purely for exploration — *i.e.*, it does not acquire any reward during this time. Madani et al. (2004) investigated this "budgeted variant" of the standard bandit problem. As its loss function depends only on the single bandit selected, this system was able to essentially ignore apparently-inferior bandits. By contrast, our BDL loss function depends on *all* of the bandits (here, variables in the BN); hence, we need to learn information about all variables, rather than just one.

Our BDL also relates to the *interventional active learning of generative BNs* (IAL) framework (Tong & Koller, 2001a;b; Murphy, 2001; Steck & Jaakkola, 2002). Here, the learner has the option of setting the values of a fixed set of features ("interventions"), then requesting the values of the remaining instances, which it receives at "unit" cost (*i.e.*, the sum of the costs of the remaining features is a constant). The IAL objective is typically to quickly learn the parameters (or the structure) of the belief network that is as close to the correct distribution as possible; we use their criteria (Expected KL divergence; Equation 1) to evaluate the quality of our estimated parameters. Our BDL differs as (1) we do not get to set any values, but can only observe the results of our probes; (2) we have an explicit budget; and (3) we have the option of purchasing only a subset of the features for an instance. We will see that this makes computing the posterior distributions more complicated.

2. Budgeted Distribution Learning

In our BDL setting, we start with a parametric model $p_{\mathbf{X}}(\cdot|\theta)$ that defines a distribution over the random variables $\mathbf{X} = (X_1, \ldots, X_d)^T$, $X_i \in \mathbb{R}$, and a prior $p_\theta(\cdot)$ over the parameters $\theta \in \mathbb{R}^m$. In what follows, θ typically denotes a random variable drawn from $p_\theta(\cdot)$ and $\mathbf{X}$ denotes a random variable drawn from $p_{\mathbf{X}}(\cdot \,|\, \theta)$. (In the context of Figure 1, $\mathbf{X}$

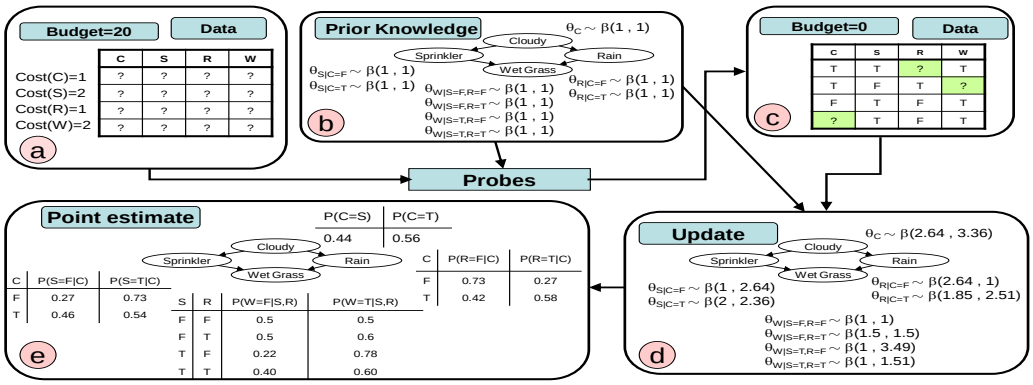


Figure 1. Budgeted learning of the parameters of the sprinkler network (Russell & Norvig, 2002).

is the set of four variables $\langle C, S, R, W\rangle$, and $\theta = \langle \theta_C, \theta_{S|c=T}, \theta_{S|c=F}, \ldots, \theta_{W|r=F,s=F}\rangle$ is the set of 9 parameters shown in Figure 1(b), corresponding to CPt-able entries of this belief net structure.) Further, we are given a fixed budget $\mathcal{B} \in \mathbb{R}^+$ (here $\mathcal{B} = 20$). We start with no data (Figure 1(a)), and must employ some "strategy" to collect the data. Moreover, we know it costs $c_i = \text{Cost}(X_i) \in \mathbb{R}^{\geq 0}$ to see the value of variable X_i for any specified data instance.

It is helpful to view the data set $\mathcal{D}$ as a growing matrix, where each row corresponds to a data instance and each column corresponds to an attribute (Fig. 1(a), 1(c)). We start with an empty matrix, where the values of the cells can only be determined through probes. A probe is defined as a purchase of the value of the i^{th} attribute and the j^{th} instance, cell (i, j), at cost c_i. Letting $\mathbf{X}^j$ refer to the j^{th} instance (*i.e.*, j^{th} row of this data set), a probe can be applied to a previously probed instance $\mathbf{X}^j$ (*e.g.*, probe $(5, 3)$ after probing $(2, 3)$) or an un-probed instance, which will increase the number of rows of $\mathcal{D}$ by one. The cost of a set of probes $A = \langle (i_k, j_k)\rangle_{k=1}^{|A|}$ is just the simple sum $c(A) = \sum_k c_{i_k}$. When the budget is exhausted (*i.e.*, when $c(A) = \mathcal{B}$), BDL then passes the collected data to a parameter learning system. The task of the BDL learner is to make the probes wisely, so that the distribution of $\mathbf{X}$, based on these learnt parameters, is estimated as accurately as possible.

Let $A \subset \{\, (i, j) : 1 \leq i \leq d,\ j \geq 1 \,\}$ be the attribute values probed by the learner, and $\mathbf{X}^A = (X_i^j)_{(i,j)\in A}$ be the responses obtained to these probes. We assume that the random variables $\mathbf{X}, \mathbf{X}^1, \mathbf{X}^2, \ldots$ are drawn from $p_{\mathbf{X}}(\cdot|\theta)$, conditionally independently of each other given θ. Let $\theta_{\mathbf{X}^A}$ denote a random variable drawn from the posterior distribution $p_\theta(\cdot|\mathbf{X}^A)$, and let $\bar{\theta}_{\mathbf{X}^A} = \mathbb{E}\left[\theta|\mathbf{X}^A\right]$ denote its expected value. Here, we use the objective function proposed by Tong & Koller (2001b):

$$J(A) \quad = \quad \mathbb{E}\left[\mathrm{KL}\left\{\, p_{\mathbf{X}}(\,\cdot\,|\,\theta_{\mathbf{X}^A}\,) \,\|\, p_{\mathbf{X}}(\,\cdot\,|\,\bar{\theta}_{\mathbf{X}^A}\,)\,\right\}\right]. \quad (1)$$

Hence, when the posterior $\theta_{\mathbf{X}^A}$ is well concentrated around its mean $\bar{\theta}_{\mathbf{X}^A}$, we expect the cost $J(A)$ to be small. (LPSzG, 2010) proves that this cost equals $J(A) = \mathbb{E}\left[\mathrm{KL}\left\{\, p_{\mathbf{X}}(\,\cdot\,|\,\theta\,) \,\|\, p_{\mathbf{X}}(\,\cdot\,|\,\bar{\theta}_{\mathbf{X}^A}\,)\,\right\}\right]$ over the prior θ.

We explore this task along two axes: One dimension is whether the variables are independent (Section 3) or not (Section 4); and the other, whether the learner is sequential. In the *sequential* (on-line) framework the set A can be selected in an incremental manner: when deciding about the $k+1^{st}$ probe, the learner can use the result of the previous k probes. In the *non-sequential* (aka off-line, allocation) version, the set A must be selected initially, and in particular, without knowing the values of any of the probes. Most of our theoretical results deal with non-sequential algorithms.

3. Independent Variable Model

In this section, we will assume that the features are independent of one another:

Assumption A1 The joint distribution of the variables $(X_1, \ldots, X_d)$ is the product of the marginal distributions. Further, the parameter vector $\theta = (\theta_1, \ldots, \theta_d)^T$ includes one parameter θ_i for each attribute X_i, in that the distribution of X_i depends only on θ_i. Formally, for any $\mathbf{x} = (x_1, \ldots, x_d)^T$, we have $p_{\mathbf{X}}(\,\mathbf{x}\,|\,\theta\,) = \prod_{i=1}^d p_{X_i}(\,x_i\,|\,\theta_i\,)$. The prior also factorizes: $p_\theta(\,\theta\,) = \prod_{i=1}^d p_{\theta_i}(\,\theta_i\,)$.

As a simple example to illustrate these conditions, consider the case when $\theta_i \in [0, 1]$, $p_{\theta_i}(\cdot)$ is a Beta-distribution, and $p_{X_i}(\cdot|\theta_i)$ is a Bernoulli distribution with parameter θ_i. We can view this as having d independent coins, each with a prior on its bias. Our task

is to learn as much about the distributions of the coins as is possible given a budget, assuming that each time coin i is flipped, the learner incurs a cost of $c_i > 0$.

3.1. Optimal Allocation Algorithm, IGA

Given this independence, it makes sense to consider just the subset of probes associated with the i^{th} random variable: $A_i = \{ (i,j) : \exists j \ (i,j) \in A \}$. We also let $V = \{ (i,j) : 1 \leq i \leq d, \ j \geq 1 \}$ be the set of all possible probes. Under these independence assumptions, the cost J decomposes into the sum of costs defined for the individual variables:[1]

Proposition 1. *For any* $A = \{A_1, \ldots, A_d\} \subset V$, $J(A) = \sum_{i=1}^{d} J_i(A_i)$, *where* $J_i(A_i) = \mathbb{E}\left[\mathrm{KL}\left\{ p_{\mathbf{X}_i}(\cdot|\theta_i) \,\|\, p_{\mathbf{X}_i}(\cdot|\mathbb{E}\left[\theta_i|\mathbf{X}^{A_i}\right]) \right\}\right]$ *is the objective function* (1) *applied to* X_i, *where* $\mathbf{X}^{A_i}$ *is set of values of the* X_i *feature in the datasample* $\mathcal{D} = \{\mathbf{X}^1, \mathbf{X}^2, \ldots\}$.

Proposition 2. *For any* $1 \leq i \leq d$, $|A_i| = |A'_i|$ *implies* $J_i(A_i) = J_i(A'_i)$.

That is, the cost associated with the i^{th} attribute depends only on how many times that attribute was probed. We therefore define $J_i(k)$ to be the cost associated with the i^{th} attribute after it has been probed k times, and note that the cost of any allocation A is the same as the cost of the corresponding "compact allocation" of the form $A' = \cup_{i=1}^{d} \{ (i,j) : 1 \leq j \leq |A_i| \}$, which depends only on the cardinalities $a_i = |A_i|$, $i = 1, \ldots, d$.

Definition 1. *Let* J *be a set function mapping subsets of* V *to reals. We say that* J *is* **monotone (non-increasing)** *if* $A \subset A' \subset V$ *implies* $J(A) \geq J(A')$, *and is* **supermodular** *if* $A \subset A' \subset V$ *and* $v \in V$ *implies* $J(A) - J(A \cup \{v\}) \geq J(A') - J(A' \cup \{v\})$. *(That is, adding* $\{v\}$ *to a small set* A *reduces the cost by more than adding* $\{v\}$ *to the larger set* A'*.)*

Let $Beta_x(\alpha, \beta)$ denote the density of a Beta distributed variable evaluated at x, and let $Ber_x(\theta)$ denote the probability mass function of a Bernoulli random variable with parameter θ, evaluated at x.

Proposition 3. *The objective function* J_i *for a variable* X_i *with* $p_{\theta_i}(\theta_i) = Beta_{\theta_i}(\alpha, \beta)$, $\alpha, \beta \in \mathbb{Z}^+$ *prior distribution, and* $Ber_{X_i}(\theta_i)$ *model likelihood is strictly monotonically decreasing in the number of probes, and supermodular.*

Let $\Delta_i(k) = J_i(k) - J_i(k+1)$ be the change of cost associated with the i^{th} attribute. The IGA algorithm, shown in Figure 2, initially computes these $\Delta_j(0)$ values for each variable j, and selects the largest $j^* = \arg\max_j \{\Delta_j(0)/c_j\}$, and assigns one probe to that variable. It then computes the expected $\Delta_{j^*}(1)/c_{j^*}$ value for that variable, and again finds the largest value along this modified frontier (replacing $\Delta_{j^*}(0)/c_{j^*}$ with $\Delta_{j^*}(1)/c_{j^*}$). The expectation is based on the chance of reaching a node, given the priors. To illustrate, consider the two unit-cost binary variables shown in Figure 3, with a budget $\mathcal{B} = 2$. IGA first computes $\Delta_A(0)$ and $\Delta_B(0)$. As $\Delta_A(0) \approx 1.51\text{E-}4 < \Delta_B(0) \approx 1.54\text{E-}4$, IGA allocates 1 probe to B. It then computes $\Delta_B(1)$ as the difference between the weighted sum of the 3 nodes at level 2 with the weighted sum at level 1, which here is $\Delta_B(1) \approx 1.49\text{E-}4$. As this is less that $\Delta_A(0)$, IGA allocates 1 probe to A. Having spent its entire budget, IGA terminates, returning the allocation: 1 probe to A and 1 to B.

IGA(budget $\mathcal{B}$; costs $\langle c_k \rangle$; reductions $\langle \Delta_k(\cdot) \rangle$)
$s := 0 \quad a_1 := 0, \ldots, a_d := 0$
while $s < \mathcal{B}$ **do**
  $j^* := \arg\max_j \{ \Delta_j(a_j)/c_j \}$
  $a_{j^*} := a_{j^*} + 1 \quad s := s + c_{j^*}$
end while
RETURN $(a_1, \ldots, a_d)$

Figure 2. Incremental Greedy Allocation algorithm, IGA

One challenge is computing $\Delta_j(k+1)$ from $\Delta_j(k)$. Fortunately, this is efficient in general — *e.g.*, when considering Beta priors, this takes only $O(k)$ time, where $k \leq \mathcal{B}$ (assuming each $c_i \geq 1$). Here, the entire algorithm, over n variables and a budget of $\mathcal{B}$, requires only $O((n+\mathcal{B})\mathcal{B} \ln n)$ time, and only $O(n)$ space.

Now consider the following assumptions:

Assumption A2

(i) All costs are equal and, in particular (without the loss of generality), $c_i \equiv 1$.

(ii) The objective functions J_i for each attribute are both monotone and supermodular.

Proposition 3 shows that Assumption A2(ii) is satisfied by the Beta prior distribution and Bernoulli model likelihood.[2]

Proposition 4. *Given Assumptions A1 and A2,* IGA *computes an optimal allocation.*

While IGA is the optimal *allocation* policy, it is not the optimal policy. To illustrate, note that the optimal policy for Figure 3 (for $\mathcal{B} = 2$) would first probe A; and if it is tails, probe A again, and otherwise probe B.

[1] Recall that the proofs of this claim, and the following ones, can all be found in the website (LPSzG, 2010).

[2] (LPSzG, 2010) shows it also holds for Gaussian distributions; we conjecture that it holds more generally.

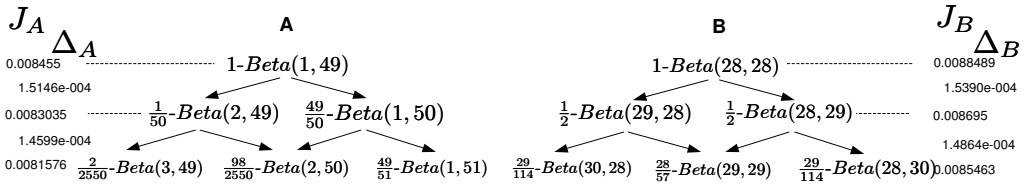


Figure 3. An example to illustrate IGA. We display J for each variable, at each level, beside each DAG. Indented numbers are the difference between each pair of Js. The notation "$a - Beta(c,d)$" means that the probability of arriving at this node is a, and the posterior distribution of the parameter is $Beta(c,d)$.

Everything above assumes that the variables have uniform cost. Otherwise:

Proposition 5. *It is NP-hard to compute the optimal budgeted allocation policy (for determining the parameters that minimize the $J(\cdot)$ objective function Equation 1), even if the variables are independent (Assumption A1).*

3.2. Other (Sequential) Algorithms

We can consider many other algorithms. *Round-robin* attempts to probe each variable the same number of times, until it terminates on the final iteration. *Random* just selects each variable uniformly at random. Finally, the *adaptive greedy* algorithm (AGA) policy picks $\arg\max_j \Delta_j(1)/c_j$ — *i.e.*, the variable with the largest expected one-step change of cost (per probe cost) associated with each attribute, given its current posterior distribution (*i.e.*, based on the responses observed to all previous probes). It then probes this variable *once*, updates its posterior using the outcome (using standard Bayesian update), and iterates until the learner runs out of budget.

We can use Figure 3 to help contrast these two algorithms. Recall IGA would probe A and B once here; AGA will do likewise. Now consider a slightly different problem, where B's prior is changed to $Beta(28,29)$. The optimal policy here is the same as for the original example. However, AGA would find this optimal policy here, but IGA would still allocate one probe to each of the two variables, which is suboptimal. This illustrates the benefit of adaptivity.

Unfortunately, AGA might not work well for the general problem, with arbitrary costs. Assume one variable has cost $c_1 = 1$ and the largest one-step $\Delta_1(1) = r$, while each the remaining $d-1$ variables has cost $c_j = \varepsilon \ll 1$ and $\Delta_j(1) = r - \varepsilon$. If the budget $\mathcal{B} = 1$, then AGA would probe the first coin and obtain a reduction of r; a better strategy, however, could probe each of the other coins $1/(d-1)\varepsilon$ times, obtaining a reduction of at least $(d-1)(r-\varepsilon)$, which is essentially $O(d)$ better than AGA.

3.3. Experiments on Independent Variables

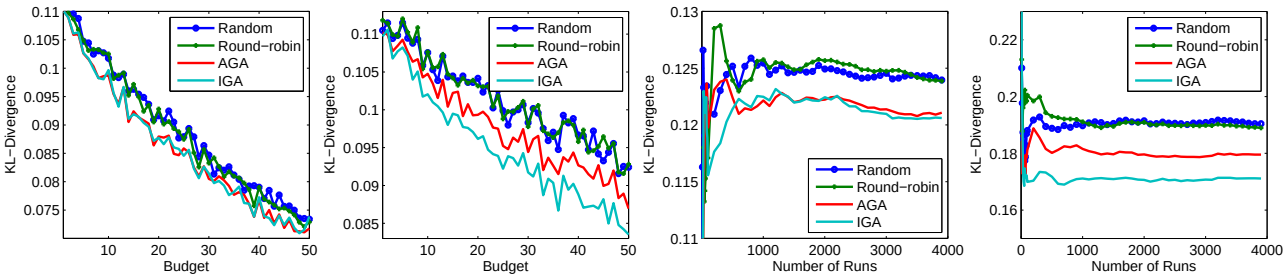
In this section, we report empirical results of a series of experiments that compare the effectiveness of these various algorithms for the independent variable model in the case when the priors are informative.

The first two experiments used 5 independent binary variables, with prior $Beta(\alpha_i, \beta_i)$. We generated non-uniform costs from a uniform discrete distribution on $\{1, 2, ..., 5\}$, where each integer cost has a probability of $\frac{1}{5}$. We generate non-uniform priors by first drawing an effective sample size e_i uniformly from $\{10, 11, \ldots, 30\}$, then drawing α_i uniformly from $\{1, 2, \ldots, e_i - 1\}$, and setting $\beta_i = e_i - \alpha_i$. The true θs are then generated from these $Beta(\alpha_i, \beta_i)$ priors.

We ran 4,000 experiments for each budget $\mathcal{B} \in \{1, \ldots, 50\}$; Fig. 4(a) – 4(b) plot the empirical means over the cost values J achieved. We repeated these experiments on a larger network using 10 independent nodes, and the fixed budget $\mathcal{B} = 200$. Fig. 4(c) – 4(d) show how the empirical means of the estimated costs J (that we can achieve after spending our budget $\mathcal{B} = 200$) converge in the number of runs for the different algorithms. As expected, we see that algorithms that use the distributional information (IGA and AGA) perform much better than ones that do not, *round-robin* and *random*. We ran a Wilcoxon signed rank test on this data, and found that in all of these subfigures, AGA and IGA are significantly better than Random and RoundRobin for essentially every number of probes (at $\alpha = 0.05$ significance level).

4. General Distributions

This section provides results when the joint probability distribution of $\mathbf{X}$ is given by a general (discrete) belief network, that can include dependencies among the variables. In general, a belief network specifies the joint probability distribution $p_{\mathbf{X}}(\cdot \,|\, \theta)$ over the random variable $\mathbf{X} = (X_1, \ldots, X_d)^T$ in a parsimonious manner. We assume we are given a labeled directed acyclic graph, $\mathcal{G} = \langle \mathcal{V}, E \rangle$ ($\mathcal{V} = \{1, \ldots, d\}$, $E \subset \mathcal{V} \times \mathcal{V}$), where node i is associated with the random variable X_i. Let $\mathbf{U}_i$ be the set of parent nodes



(a) Uniform cost (b) Different cost (c) Uniform cost, $\mathcal{B} = 200$ (d) Different cost, $\mathcal{B} = 200$

Figure 4. Empirical studies using Informative priors

of node i. The graph $\mathcal{G}$ encodes the conditional independencies of $\mathbf{X}$: each node is independent of its non-descendants, given its parents (Pearl, 1988). As we assume that $\mathbf{X}$ is discrete valued, we need to specify, for each node X_i, the conditional distributions, $\theta_{X_i|\mathbf{u}_i} = p_{X_i|\mathbf{u}_i}(\, x_i \,|\, \mathbf{u}_i; \theta\,)$, called the *conditional probability tables*. We can then write the probability of any complete tuple $\mathbf{x} = (x_1, \ldots, x_d)^T$, as a simple product $p_{\mathbf{X}}(\, \mathbf{x} \,|\, \theta\,) = \prod_{i=1}^{d} p_{X_i}(\, x_i \,|\, \mathbf{u}_i, \theta_i\,) = \prod_i \theta_{x_i|\mathbf{u}_i}$.

We assume that these parameters $\{\theta_{X_i|\mathbf{u}_i}\}$ are initially independent, which mean the belief network corresponds to a product of Dirichlet distributions. Given a data set $\mathcal{D}$, we can compute the posterior distribution; if this $\mathcal{D}$ consists of *complete* instances (*i.e.*, specifies a value for each variable), then the network remains factored as a product of Dirichlet distributions, each of which is updated: if the parameter is initially $\theta_{X_i|\mathbf{U}_i=\mathbf{u}} \sim \text{Beta}(\alpha, \beta)$, and $\mathcal{D}$ includes a instances matching $X_i = +$ and $\mathbf{U}_i = \mathbf{u}$ and b instances matching $X_i = -$ and $\mathbf{U}_i = \mathbf{u}$, then the posterior is $\theta_{X_i|\mathbf{u}_i=\mathbf{u}} \sim \text{Beta}(\alpha + a, \beta + b)$ (Heckerman, 1999).

Unfortunately, if there are omissions in the training data (that is, there is a training instance that includes the values for some, but not all of the features), then the posterior distribution can become a mixture, which in general does not have a simple analytic form. Indeed, if the data instance specifies only $d - k$ values (*i.e.*, there are k omissions), then the posterior distribution corresponds to a mixture of 2^k products of Dirichlet distributions (if each variable is binary). We therefore need a way to estimate this objective function. (As computing $J(\cdot)$ requires estimating the distribution itself, and not just its mean values, we cannot use EM (Dempster et al., 1977).) See Section 4.3.

4.1. Complete 2-Node Belief Networks with BDe Priors and Uniform Costs

A network has *BDe Dirichlet priors* if the CPtables entries are "matching" (Tong & Koller, 2001b). For example, in the structure $A \to B$, the parameters $\theta_A \sim \text{Beta}(3,4)$, $\theta_{B|+a} \sim \text{Beta}(1,2)$, and $\theta_{B|-a} \sim \text{Beta}(3,1)$ are BDe as the "effective" number of pseudo-instances corresponding to $+a$ here is 3 based both on the "3" in θ_A's first parameter and the fact that $\theta_{B|+a}$ has an effective sample size of $1 + 2 = 3$. Similarly the "effective" number of pseudo-instances corresponding to $-a$ here is 4 based both on the "4" in θ_A's second parameter and $\theta_{B|-a}$'s effective sample size of $3 + 1 = 4$.

Proposition 6. *For a complete 2-node belief network with BDe Beta priors and uniform costs, when the budget $\mathcal{B}$ is an even number, an allocation algorithm that takes full data instances gives a posterior distribution with the minimum expected risk $J(\cdot)$.*

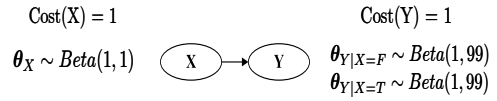
This claim shows that, for some situations, the best allocation algorithm involves the obvious round-robin approach: if our budget is $\mathcal{B}$ and there are $d = 2$ variables, probe each variable $\mathcal{B}/d$ times.

4.2. Non-BDe, Non-uniform Costs, Incomplete

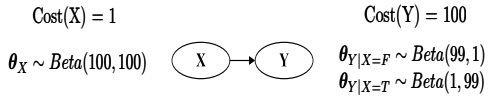
The BDe and uniform cost constraints are crucial for a round-robin like algorithm, which takes full data instances, to perform well. As one counter-example, consider the non-BDe distribution in Fig. 5(a). Here, X and Y are basically independent from each other. As we are very certain about the probability of Y but we know little about X, and their costs are the same, it makes sense to allocate more probes to X. (That is, if the budget $\mathcal{B} = 2$, we will do much better probing X twice, rather than the round-robin approach of probing X once and Y once.)

Now consider the BDe distribution shown in Fig. 5(b), where the costs are very different. As X and Y are highly correlated, and the cost of probing X is significantly cheaper, we clearly get more "information per unit cost" by probing X more. (So given the budget $\mathcal{B} = 101$, we would do much better proving X 101 times, rather than probing X once and Y once.)

Finally, this RoundRobin also requires that the struc-



(a) Uniform costs, non-BDe priors



(b) Different costs, BDe priors

Figure 5. Situations where RoundRobin is suboptimal

ture be complete — *i.e.*, that every node is connected to every other. As a counter-example, imagine X and Y are independent, where $\theta_X \sim \text{Beta}(1, 999)$ and $\theta_Y \sim \text{Beta}(500, 500)$. Given the budget $\mathcal{B} = 2$, we will do much better probing Y twice, rather than the round-robin approach of probing X once and Y once.

4.3. Estimating $J(\cdot)$ from Partial Data

The above counter-examples show that the round-robin algorithm is not always optimal, which means the optimal algorithm will probably produce partial instances. To evaluate such algorithms, we need to compute the objective function $J(\cdot)$ on the resulting distribution; as discussed above, this is trickly as the resulting distribution is no longer simple. This section provides a sampling algorithm for estimating this expected KL value.

In general, we can write the final datasample as $\mathcal{D} = \{\mathcal{D}_o, \mathcal{D}_m\}$, where $\mathcal{D}_o$ and $\mathcal{D}_m$ are the observed and missing elements, respectively. Our goal is to obtain the observations $\mathcal{D}_o$ that lead to a posterior over $\boldsymbol{\theta}$ that is well concentrated around its mean (in expectation) — *i.e.*, that minimize

$$J(\mathcal{D}_o) = \mathbb{E}_{\boldsymbol{\theta}_1 \sim p_{\boldsymbol{\theta}}(\cdot \mid \mathcal{D}_o)} \text{KL}\left\{ p_{\mathbf{X}}(\cdot \mid \boldsymbol{\theta}_1) \,\|\, p_{\mathbf{X}}(\cdot \mid \bar{\boldsymbol{\theta}}) \right\}, \quad (2)$$

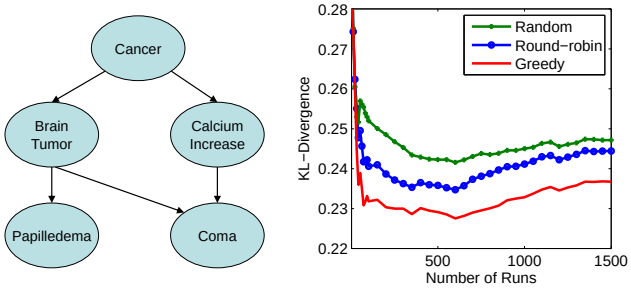
where $\bar{\boldsymbol{\theta}} \doteq \mathbb{E}_{p_{\boldsymbol{\theta}}(\cdot \mid \mathcal{D}_o)}[\boldsymbol{\theta}]$. Observe that the posterior is a mixture of Dirichlets:

$$p(\boldsymbol{\theta} \mid \mathcal{D}_o) = \sum_{\mathcal{D}_m} p(\boldsymbol{\theta} \mid \mathcal{D}_o, \mathcal{D}_m)\, p(\mathcal{D}_m \mid \mathcal{D}_0) \quad (3)$$

Since the number of components of the posterior can be exponential in the number of omissions in the dataset, $|\mathcal{D}_m|$, its analytical computation is not feasible for a dataset with a large number of omissions. We there define the $\widehat{J}$ sampling algorithm (shown in Figure 6) to estimate the objective function in (2), and to serve as the basis for the obvious **Greedy** algorithm: At each time, consider probing each unspecified feature X_i^j. Each of the $k \geq 2$ possible outcomes of this probe will occur with probability $p(X_i^j | \boldsymbol{\theta}, \mathcal{D}_o)$. While we do not know the true $\boldsymbol{\theta}$, we can use the estimate $\hat{\boldsymbol{\theta}}(X^j \cap \mathcal{D}_o)$ based on the data, where $X^j \cap \mathcal{D}_o$ are the features specified in the partial instance X^j. Then for each $\{X_i^j = v\}$ outcome, we compute $\widehat{J}(p_{\mathbf{X}}(\cdot|\hat{\boldsymbol{\theta}});\ \mathcal{D}_o \cup \{X_i^j = v\};\ \dots)$ as an estimate of the resulting objective function. The average

$$J(i,j) = \sum_v p(X_i^j = v | \hat{\boldsymbol{\theta}}) \widehat{J}(p_{\mathbf{X}}(\cdot|\hat{\boldsymbol{\theta}}); \mathcal{D}_o \cup \{X_i^j = v\}; \dots)$$

estimates the quality of probing this X_i^j; we then probe the optimal $\langle \hat{i}, \hat{j} \rangle = \arg\max_{i,j} J(i,j)$.



(a) Cancer BN (b) KL divergences

Figure 7. (a) Cancer belief network. (b) KL divergences from the true parameters.

4.4. Empirical Studies on General BNs

We performed many experiments using 3 algorithms: the Greedy defined above, as well as the obvious Random and Round-robin, over a number of belief network structures. Here we report our findings on the "Cancer" structure (Fig. 7(a)), a BN with 5 correlated binary variables and 11 parameters (Pearl, 1988).

Fig. 7(b) compares these 3 algorithms (where Greedy uses $S = 5$) with a budget $\mathcal{B} = 10$. The plots show how the running averages of the $\widehat{J}(\dots)$ functions converge for each algorithm. We can see that Greedy performs better than the other two algorithms, which shows that considering the defined risk can help the learner to obtain a better estimate of the parameters. A Wilcoxon signed rank test confirmed that Greedy significantly outperformed Random and Round-robin at $\alpha = 0.05$ significance level.

5. Conclusions

This paper theoretically and empirically explores the *budgeted distribution learning* task, where the learner is allowed to sequentially probe specified attributes of data instances to obtain their values, with the goal of learning the joint distribution over the attributes, subject to the fixed total budget. We first studied the simple case where all variables are independent, and proved that the simple IGA is the optimal allocation algorithm when the costs are uniform, but that even this simple allocation task is NP-hard for general costs. We then showed that sequential algorithms can do better than allocation algorithms, and presented empirical studies to show that algorithms that use the distribu-

$\widehat{J}(\ p_{\mathbf{X}}(\cdot|\theta_{\mathbf{0}});\ \mathcal{D}_{\mathbf{o}};\ S)$ % $\theta_{\mathbf{0}}$= *initial param,* $\mathcal{D}_o$ *= observed data,* S *= number of imputed datasets*
for $i = 1..S$ **do**
  Draw i^{th} imputed datasample $\mathcal{D}_m^i$ from $p_{\mathbf{X}}(\cdot \mid \mathcal{D}_o, \theta_{\mathbf{0}})$
  Draw $\hat{\boldsymbol{\theta}}(i)$ from $p_{\boldsymbol{\theta}}(\cdot \mid \mathcal{D}_o, \mathcal{D}_m^i)$ % *Each* $\hat{\boldsymbol{\theta}}(i)$ *is a product of Dirichlet distr'ns*
end for
$\bar{\boldsymbol{\theta}} := \frac{1}{S}\sum_{i=1}^{S} \hat{\boldsymbol{\theta}}(i)$ % *... = mean*$[\hat{\boldsymbol{\theta}}]$ *...*$\approx$ *estimate of* $\bar{\boldsymbol{\theta}}(p_{\boldsymbol{\theta}}(\cdot \mid \mathcal{D}_o)) = \mathbb{E}_{p_{\boldsymbol{\theta}}(\cdot \mid \mathcal{D}_o)}[\boldsymbol{\theta}]$
for $i = 1..S$ **do**
  $\mathbf{K}(i) := \sum_{n=1}^{d} \sum_{\boldsymbol{u_n}} p_{\mathbf{X}}(\boldsymbol{u_n} \mid \hat{\boldsymbol{\theta}}(i)) \sum_{x_n} \hat{\boldsymbol{\theta}}(i)_{x_n|\boldsymbol{u_n}} \ln \frac{\hat{\boldsymbol{\theta}}(i)_{x_n|\boldsymbol{u_n}}}{\theta_{X_n|\boldsymbol{u_n}}}$
  % *%* n *indexes the variables;* $\boldsymbol{u_n}$ *ranges over instantiations of the parents of the* n^{th} *variable* X_n
  % $\hat{\boldsymbol{\theta}}(i)_{x_n|\boldsymbol{u_n}}$ *= parameter of* $\hat{\boldsymbol{\theta}}(i)$ *associated with* $[X_n = x_n, \boldsymbol{U_n} = \boldsymbol{u_n}]$*; similarly for* $\bar{\theta}_{x_n|\boldsymbol{u_n}}$
end for
RETURN $\hat{J} := \frac{1}{S}\sum_{i=1}^{S} \mathbf{K}(i)$ % *...* $=$ *mean*$[\mathbf{K}]$

Figure 6. Sampling algorithm for estimating the EKL of a posterior distribution

tional information (IGA and AGA) perform much better than ones that do not, *round-robin* and *random*.

We then considered more general belief networks with dependencies, and showed restricted situations where the obvious *round-robin* algorithm is guaranteed to be the optimal allocation algorithm, followed by examples where round-robin is not optimal. This means we will need to consider partially specified training instances, which lead to complex posterior distributions that do not have a simple closed form. We provided a stochastic algorithm that approximates the evaluation (Expected KL divergence), which we used to produce a greedy algorithm. While round robin algorithm does work well in practice for certain belief nets, our empirical results show that our greedy algorithm can be significantly superior. The results of these explorations reveal a number of insights about the challenges of acquiring the information needed to learn a distribution. The webpage (LPSzG, 2010) shows that these results hold for other relevant distributions as well.

This paper presents many initial steps in addressing the challenges of the budgeted distribution learning framework. However, much remains to be done. For example, our results show that, except for a few special cases, learning in the Bayesian framework can be hard. In the future, we plan to study the challenge of approximating this solution.

Acknowledgements

This work was supported in part by AICML, AITF (formerly iCore and AIF), NSERC and the PASCAL2 Network of Excellence under EC grant no. 216886. Cs. Szepesvári is on leave from SZTAKI, Hungary.

References

Berry, A. and Fristedt, B. *Bandit Problems: Sequential Allocation of Experiments*. Springer, 1985.

Dempster, A., Laird, N., and Rubin, D. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statistics Soc, B*, 39, 1977.

Heckerman, D. A tutorial on learning with Bayesian networks. In *Learning in Graphical Models*. 1999.

Krause, A. and Guestrin, C. Near-optimal nonmyopc value of information in graphical models. In *UAI*, 2005a.

Krause, A. and Guestrin, C. Optimal nonmyopic value of information in graphical models - efficient algorithms and theoretical limits. In *IJCAI*, 2005b.

Lizotte, D., Madani, O., and Greiner, R. Budgeted learning of Naive-Bayes classifiers. In *UAI*, 2003.

LPSzG, 2010. http://sites.google.com/site/BudgetedDistributionLearning/.

Madani, O., Lizotte, D., and Greiner, R. Active model selection. In *UAI*, 2004.

Melas, V.B. *Functional Approach to Optimal Experimental Design*. Springer, 2006.

Murphy, K. Active learning of causal Bayes net structure. Technical report, UCB, 2001.

Muslea, I. *Active learning with multiple views*. PhD thesis, USC, 2002.

Pearl, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. 1988.

Russell, S. and Norvig, P. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2002.

Steck, H. and Jaakkola, T. Unsupervised active learning in large domains. In *UAI*, 2002.

Tong, S. and Koller, D. Active learning for structure in Bayesian networks. In *IJCAI*, 2001a.

Tong, S. and Koller, D. Active learning for parameter estimation in Bayesian networks. In *NIPS*, 2001b.

Verma, T. and Pearl, J. Equivalence and synthesis of causal models. In *UAI*, 1991.