Optimal Nonmyopic Value of Information in Graphical Models – Efficient Algorithms and Theoretical Limits

Andreas Krause
Carnegie Mellon University

Carlos Guestrin
Carnegie Mellon University

Abstract

Many real-world decision making tasks require us to choose among several expensive observations. In a sensor network, for example, it is important to select the subset of sensors that is expected to provide the strongest reduction in uncertainty. It has been general practice to use heuristic-guided procedures for selecting observations. In this paper, we present the first efficient optimal algorithms for selecting observations for a class of graphical models containing Hidden Markov Models (HMMs). We provide results for both selecting the optimal subset of observations, and for obtaining an optimal conditional observation plan. For both problems, we present algorithms for the filtering case, where only observations made in the past are taken into account, and the smoothing case, where all observations are utilized. Furthermore we prove a surprising result: In most graphical models tasks, if one designs an efficient algorithm for chain graphs, such as HMMs, this procedure can be generalized to polytrees. We prove that the value of information problem is $\mathbf{NP^{PP}}$-hard even for discrete polytrees. It also follows from our results that even computing conditional entropies, which are widely used to measure value of information, is a $\#\mathbf{P}$-complete problem on polytrees. Finally, we demonstrate the effectiveness of our approach on several real-world datasets.

1 Introduction

In probabilistic reasoning, where one can choose among several possible but expensive observations, it is often a central issue to decide which variables to observe in order to most effectively increase the expected utility. In a medical expert system [14], for example, multiple tests are available, and each test has a different cost. In such systems, it is thus important to decide which tests to perform in order to become most certain about the patient's condition, at a minimum cost. Occasionally, the cost of testing can even exceed the value of information for any possible outcome.

The following running example motivates our research and is empirically evaluated in Section 7. Consider a temperature monitoring task, where wireless temperature sensors are distributed across a building. The task is to become most certain about the temperature distribution, whilst minimizing energy expenditure, a critically constrained resource [4].

Many researchers have suggested the use of myopic (greedy) approaches to select observations [13; 15; 5; 1]. Unfortunately, this heuristic does not provide any performance guarantees. In this paper, we present efficient algorithms, which guarantee optimal nonmyopic value of information in chain graphical models such as Hidden Markov Models (HMMs). We address two settings: *subset selection*, where the optimal subset of observations is obtained in an open-loop fashion, and *conditional plans*, a closed-loop plan where the observation strategy depends on the actual value of the observed variables (*c.f.* Fig. 1(a)). To our knowledge, these are the first optimal and efficient algorithms for these tasks for this class of graphical models. For both settings, we address the *filtering* and the *smoothing* versions: Filtering is important in online decision making, where our decisions can only utilize observations made in the past. Smoothing arises for example in structured classification tasks, where there is no temporal dimension in the data, and hence all observations can be taken into account. We evaluate our algorithms empirically on three real-world datasets, and also show that they are well-suited for interactive classification of sequential data.

Most problems in graphical models, such as probabilistic inference and the most probable explanation, that can be solved efficiently for chain-structured graphs, can also be solved efficiently for polytrees. We prove that the problem of maximizing value of information is $\mathbf{NP^{PP}}$-hard even for discrete polytree graphical models, giving a complexity theoretic classification of a core artificial intelligence problem. $\mathbf{NP^{PP}}$-hard problems are believed to be significantly harder than $\mathbf{NP}$-complete or even $\#\mathbf{P}$-complete problems commonly arising in the context of graphical models. As a special case, we also prove that computing conditional entropies is $\#\mathbf{P}$-complete even in the case of discrete polytrees. This is a surprising result about a measure of uncertainty that is frequently used in practice.

2 Optimization criteria

In order to maximize value of information, our objective functions should depend on probability distributions over variables. Let $S = \{X_1, \ldots, X_n\}$ be a set of discrete random variables. We consider a class of *local reward* functions R, which are defined on the marginal probability distributions of the variables. This class has the computational advantage that local rewards can be evaluated using probabilistic inference techniques. The total reward will then be the sum of all local rewards.

Let O be a subset of S. Then $P(X_j \mid O = o)$ denotes the marginal distribution of variable X_j conditioned on observations o. For classification purposes, it can be more appropriate to consider the max-marginals $P^{max}(X_j = x_j \mid O = o) = \max_{\mathbf{x}} P(\mathbf{X} = \mathbf{x}, X_j = x_j \mid O = o)$, that is, for X_j set to value

x_j, the probability of the most probable assignment to all other random variables conditioned on the observations o. The *local reward* R_j is a functional on the probability distribution P or P^{max} over X_j. We write

$$R_j(X_j \mid O) \triangleq \sum_o P(O = o) R_j(P(X_j \mid O = o))$$

as an abbreviation to indicate *expected local rewards*, where the expectation is taken over all assignments o to the observations O. Important measures for value of information include:

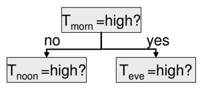
- **Entropy.** If we set $R_j(P(X_j \mid O)) = -H(X_j \mid O) = \sum_{x_j,o} P(x_j, o) \log P(x_j \mid o)$, the objective in the optimization problem becomes to minimize the sum of residual entropies. We choose this reward function in our running example to measure the uncertainty about the temperature distribution.
- **Maximum expected utility.** The concept of local reward functions also includes the concept of utility nodes in influence diagrams. If $U_j : A_j \times \operatorname{dom} X_j \to \mathbb{R}$ is a utility function mapping an action $a \in A_j$ and an outcome $x \in \operatorname{dom} X_j$ to a reward, then the *maximum expected utility principle* states that actions should be selected as to maximize $EU_j(a \mid o) = \sum_x P(x \mid o) U_j(a, x)$. The more certain we are about X_j, the more economically we can choose our action. Hence we can define our local reward function $R(P(X_j \mid O)) = \sum_o P(o) \max_a EU_j(a \mid o)$.
- **Margin.** We can also consider the margin of confidence: $R_j(P^{max}(X_j \mid O)) = \sum_o P(o)[P^{max}(x_j^* \mid o) - P^{max}(\bar{x}_j \mid o)]$, where $x^* = \operatorname{argmax}_{x_j} P^{max}(x_j \mid o)$ and $\bar{x} = \operatorname{argmax}_{x_j \neq x^*} P^{max}(x_j \mid o)$, which describes the margin between the most likely outcome and the closest runner up. This reward function is very useful for structured classification purposes, as shown in Sec. 7.

These examples demonstrate the generality of our notion of local reward. One can generalize the algorithms even more, e.g., to measure the total entropy or the margin between the most probable explanation and its runner up. Details are omitted here due to space limitations.

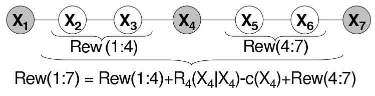
We also want to capture the constraint that observations are expensive. This can mean that each observation X_j has an associated positive *penalty* C_j that effectively decreases the reward. In our example, we might be interested in trading off accuracy with sensing energy expenditure. Alternatively, it is also possible to define a *budget* B for selecting observations, where each one is associated with an integer *cost* β_j. Here, we want to select observations whose sum cost is within the budget, but these costs do not decrease the reward. In our running example, the sensors could be powered by solar power, and regain a certain amount of energy per day, which allows a certain amount of sensing. Our formulation of the optimization problems allows both for penalties and budgets. To simplify notation we also write $C(O) = \sum_{X_j \in O} C_j$ and $\beta(O) = \sum_{X_j \in O} \beta_j$ to extend C and β to sets.

3 Decomposing Rewards

In the following Sections 4 and 5, we present efficient algorithms for two problems of optimizing value of information in the class of chain graphical models.



(a) Example conditional plan.



(b) Decomposition of the reward.

Figure 1: Example, and decomposing reward idea

The set of random variables $S = \{X_1, \ldots, X_n\}$ forms a chain graphical model (a chain), if X_i is conditionally independent of X_k given X_j whenever $i < j < k$. We can assume that the joint distribution is specified by the prior $P(X_1)$ and conditional probability distributions $P(X_{i+1} \mid X_i)$. The time series model for the temperature measured by one sensor in our example can be formulated as a chain graphical model.

Chain graphical models originating from time series have additional, specific properties: In a system for online decision making, only observations from the past and present time steps can be taken into account, not observations which will be made in the future. This is in general referred to as the *filtering* problem. In this setting, the notation $P(X_i \mid O)$ will refer to the distribution of X_i conditional on observations in O prior to and including time i. For structured classification problems as discussed in Section 7, in general observations made anywhere in the chain must be taken into account. This situation is usually referred to as the *smoothing* problem. We will provide algorithms both for filtering and smoothing.

We will now describe the key insight, which allows for efficient optimization in chains. Consider a set of observations $O \subseteq S$. If the j variable is observed, i.e., $X_j \in O$, then the local reward is simply $R(X_j \mid O) = R(X_j \mid X_j)$. Now consider $X_j \notin O$, and let O_j be the subset of O containing the closest ancestor (and for the smoothing problem also the closest descendant) of X_j in O. The conditional independence property of the graphical model implies that, given O_j, X_j is independent of the rest of the observed variables, i.e., $P(X_j \mid O) = P(X_j \mid O_j)$. Thus, it follows that $R(X_j \mid O) = R(X_j \mid O_j)$.

These observations imply that the expected reward of some set of observations decomposes along the chain. For simplicity of notation, we add two independent dummy variables X_0 and X_{n+1}, where $R_0 = C_0 = \beta_0 = R_{n+1} = C_{n+1} = \beta_{n+1} = 0$. Let $O = \{X_{i_0}, \ldots, X_{i_{m+1}}\}$ where $i_l < i_{l+1}$, $i_0 = 0$ and $i_{m+1} = n + 1$. Using this notation, the total reward $R(O) = \sum_j R_j(X_j \mid O)$ for the smoothing case is given by:

$$\sum_{v=0}^{m} \left(R_{i_v}(X_{i_v} \mid X_{i_v}) - C_{i_v} + \sum_{j=i_v+1}^{i_{v+1}-1} R_j(X_j \mid X_{i_v}, X_{i_{v+1}}) \right).$$

In filtering settings, we simply replace $R_j(X_j \mid X_{i_v}, X_{i_{v+1}})$ by $R_j(X_j \mid X_{i_v})$. Figure 1(b) illustrates this decomposition.

Consider now a Hidden Markov Model unrolled for n time steps, i.e., S can be partitioned into the hidden variables $\{X_1, \ldots, X_n\}$ and the emission variables $\{Y_1, \ldots, Y_n\}$. In HMMs, the Y_i are observed and the variables X_i form a chain. In many applications, some of which are discussed in Section 7, we can observe some of the hidden variables, e.g., by asking an expert, in addition to observing the emission variables. In these cases, the problem of selecting expert labels also belongs to the class of chain graphical models addressed by this paper.

4 Subset Selection

In the *subset selection* problem, we want to find a most informative subset of the variables to observe *in advance*, i.e., before any observations are made. In our running example, we would, before deploying the sensors, identify k time points that are expected to provide the most informative sensor readings according to our model.

First, define the objective function L on subsets of S by

$$L(O) = \sum_{j=1}^{n} R_j(X_j \mid O) - C(O). \tag{4.1}$$

The *subset selection* problem is to find the optimal subset

$$O^* = \operatorname*{argmax}_{O \subseteq S, \beta(O) \leq B} L(O)$$

maximizing the sum of expected local rewards minus the penalties, subject to the constraint that the total cost must not exceed the budget B.

We solve this optimization problem using a dynamic programming algorithm, where the chain is broken into sub-chains using the insight from Sec. 3. Consider a sub-chain from variable X_a to X_b. We define $L_{a:b}(k)$ to represent the expected total reward for the sub-chain $X_a, \ldots, X_b$, where X_a (and X_b in the smoothing case) is observed, and with a budget level of k. More formally:

$$L_{a:b}(k) = \max_{\substack{O \subset \{X_{a+1} \cdots X_{b-1}\} \\ \beta(O) \leq k}} \sum_{j=a+1}^{b-1} R_j(X_j \mid O \cup \{X_a\}) - C(O),$$

for the filtering version, and

$$L_{a:b}(k) = \max_{\substack{O \subset \{X_{a+1} \cdots X_{b-1}\} \\ \beta(O) \leq k}} \sum_{j=a+1}^{b-1} R_j(X_j \mid O \cup \{X_a, X_b\}) - C(O),$$

for the smoothing version. Note that $L_{0:n+1}(B) = \max_{O:\beta(O) \leq B} L(O)$, as in Eq. (4.1), i.e., by computing the values for $L_{a:b}(k)$, we compute the maximum expected total reward for the entire chain.

We can compute $L_{a:b}(k)$ using dynamic programming. The base case is simply:

$$L_{a:b}(0) = \sum_{j=a+1}^{b-1} R_j(X_j \mid X_a),$$

for filtering, and

$$L_{a:b}(0) = \sum_{j=a+1}^{b-1} R_j(X_j \mid X_a, X_b),$$

for smoothing. The recursion for $L_{a:b}(k)$ has two cases: we can choose not to spend any more of the budget, reaching the base case, or we can break the chain into two sub-chains, selecting the optimal observation X_j, where $a < j < b$:

$$\begin{aligned} L_{a:b}(k) = \max\{L_{a:b}(0), \max_{j:a<j<b, \beta_j \leq k} \{-C_j + \\ + R_j(X_j \mid X_j) + L_{a:j}(0) + L_{j:b}(k - \beta_j)\}\}. \end{aligned}$$

At first, it may seem that this recursion should consider the optimal split of the budget between the two sub-chains. However, since the subset problem is open-loop and the order of the observations is irrelevant, we only need to consider split points where the first sub-chain receives zero budget.

```
Input: Budget $B$, rewards $R_j$, costs $\beta_j$ and penalties $C_j$
Output: Optimal selection $O$ of observation times
begin
    for $0 \leq a < b \leq n+1$ do compute $L_{a:b}(0)$;
    for $k = 1$ to $B$ do
        for $0 \leq a < b \leq n+1$ do
            $sel(-1) := L_{a:b}(0)$;
            for $j = a+1$ to $b-1$ do $sel(j) :=$
            $-C_j + R_j(X_j \mid X_j) + L_{a:j}(0) + L_{j:b}(k - \beta_j)$;
            $L_{a:b}(k) = \max_{a<j<b} sel(j)$;
            $\Lambda_{a:b}(k) = \operatorname{argmax}_{a<j<b} sel(j)$;
        end
    end
    $a := 0$; $b := n+1$; $k := B$; $O := \emptyset$;
    repeat
        $j := \Lambda_{a:b}(k)$;
        if $j \geq 0$ then $O := O \cup \{X_j\}$; $k := k - \beta_j$;
    until $j = -1$;
end
```

Algorithm 1: Optimal subset selection.

A pseudo code implementation is given in Alg. 1. If we do not consider different costs β, we would simply choose $\beta_j = 1$ for all variables and compute $L_{a:b}(N)$. Alg. 1 uses the quantities $\Lambda_{a:b}$ to recover the optimal subset by tracing the maximal values occurring in the dynamic programming equations. Using an induction proof, we obtain:

Theorem 1. *The dynamic programming algorithm described above computes the optimal subset with budget B in $(\frac{1}{6}n^3 + \mathcal{O}(n^2))B$ evaluations of expected local rewards.* $\square$

If the variables X_i are continuous, our algorithm is still applicable when the integrations and inferences necessary for computing the expected rewards can be performed efficiently.

5 Conditional Plan

In the *conditional plan* problem, we want to compute an optimal querying policy: We sequentially observe a variable, pay the penalty, and depending on the observed values, select the next query as long as our budget suffices. The objective is to find the plan with the highest expected reward, where, for each possible sequence of observations, the budget B is not exceeded. For filtering, we can only select observations in the future, whereas in the smoothing case, the next observation can be anywhere in the chain. In our running example, the filtering algorithm would be most appropriate: The sensors would sequentially follow the conditional plan, deciding on the most informative times to sense based on the previous observations. Fig. 1(a) shows an example of such a conditional plan.

The formal definition of the objective function J is given recursively. The base case considers the exhausted budget:

$$J(O = o; 0) = \sum_{X_j \in S} R_j(X_j \mid O = o) - C(O).$$

The recursion, $J(O = o; k)$, represents the maximum expected reward of the conditional plan for the chain where $O = o$ has been observed and the budget is limited to k:

$$J(O=o;k) = \max\{J(O=o;0), \max_{X_j \notin O}\{ \sum_y P(X_j = y \mid O=o) \cdot J(O=o, X_j = y; k-\beta_j)\}\}.$$

The optimal plan has reward $J(\emptyset; B)$.

We propose a dynamic programming algorithm for obtaining the optimal conditional plan that is similar to the subset algorithm presented in Sec. 4. Again, we utilize the decomposition of rewards described in Section 3. The difference here is that the observation selection and budget allocation now depend on the actual values of the observations.

We again consider sub-chains $X_a, \ldots, X_b$. The base case deals with the zero budget setting:

$$J_{a:b}(x_a;0) = \sum_{j=a+1}^{b-1} R_j(X_j \mid X_a = x_a),$$

for filtering, and

$$J_{a:b}(x_a, x_b;0) = \sum_{j=a+1}^{b-1} R_j(X_j \mid X_a = x_a, X_b = x_b),$$

for smoothing. The recursion defines $J_{a:b}(x_a;k)$ ($J_{a:b}(x_a,x_b;k)$ for smoothing), the expected reward for the problem restricted to the sub-chain $X_a, \ldots, X_b$ conditioned on the values of X_a (and X_b for smoothing), and with budget limited by k. To compute this quantity, we again iterate through possible split points j, such that $a < j < b$. Here we observe a notable difference between the filtering and the smoothing case. For smoothing, we now must consider all possible splits of the budget between the two resulting sub-chains, since an observation at time j might require us to make an additional, earlier observation:

$$J_{a:b}(x_a, x_b;k) = \max\{J_{a:b}(x_a,x_b;0), \max_{a<j<b}\{-C_j + \sum_{x_j} P(X_j = x_j \mid X_a = x_a, X_b = x_b)\{R_j(X_j \mid X_j) + \max_{0 \le l \le k-\beta_j} [J_{a:j}(x_a, x_j; l) + J_{j:b}(x_j, x_b; k-l-\beta_j)]\}\}\}.$$

Looking back in time is not possible in the filtering case, hence the recursion simplifies to

$$J_{a:b}(x_a;k) = \max\{J_{a:b}(x_a;0), \max_{a<j<b:\beta_j \le k}\{-C_j + \sum_{x_j} P(X_j = x_j \mid X_a = x_a)\{R_j(X_j \mid X_j) + J_{a:j}(x_a;0) + J_{j:b}(x_j; k-\beta_j)\}\}\}.$$

The optimal reward is obtained by $J_{0:n+1}(\emptyset; B) = J(\emptyset; B)$. Alg. 5 presents a pseudo code implementation for the smoothing version – the filtering case is a straight-forward modification. The plan itself is compactly encoded in the quantities $\pi_{a:b}$ which determines the next variable to query and $\sigma_{a:b}$, which determines the allocation of the budget. Considering the exponential number of possible sequences of observations, it is remarkable that the optimal plan can even be represented using only polynomial space. Alg. 5 indicates how the computed plan can be executed. The procedure is recursive, requiring the parameters $a := 0$, $x_a := 1$, $b := n+1$, $x_b := 1$ and $k := B$ for the initial call. Again, by induction, we obtain:

Theorem 2. *The algorithm for smoothing presented above computes an optimal conditional plan in $d^3 \cdot B^2 \cdot (\frac{1}{6}n^3 + \mathcal{O}(n^2))$ evaluations of local rewards, where d is the maximum domain size of the random variables $X_1, \ldots, X_n$. In the filtering case, or if no budget is used, the optimal plan can be computed using $d^3 \cdot B \cdot (\frac{1}{6}n^3 + \mathcal{O}(n^2))$ or $d^3 \cdot (\frac{1}{6}n^4 + \mathcal{O}(n^3))$ evaluations respectively.* □

The faster computation for the no budget case is obtained by observing that we do not require the third maximum computation, which distributes the budget into the sub-chains.

```
Input: Budget $B$, rewards $R_j$, costs $\beta_j$ and penalties $C_j$
Output: Optimal conditional plan ($\pi_{a:b}$, $\sigma_{a:b}$)
begin
    for $0 \le a < b \le n+1, x_a \in \mathrm{dom}\, X_a, x_b \in \mathrm{dom}\, X_b$ do
    compute $J_{a:b}(x_a, x_b; 0)$;
    for $k = 1$ to $B$ do
        for $0 \le a < b \le n+1, x_a \in \mathrm{dom}\, X_a, x_b \in \mathrm{dom}\, X_b$ do
            $sel(-1) := J_{a:b}(0)$;
            for $a < j < b$ do
            $sel(j) := -C_j + R_j(X_j \mid X_j)$;
            for $a < j < b, x_j \in \mathrm{dom}\, X_j$ do
                for $0 \le l \le k - \beta_j$ do ;
                $bd(l) :=$
                    $J_{a:j}(x_a, x_j; l) + J_{j:b}(x_j, x_b; k - l - \beta_j)$;
                $sel(j) := sel(j) + P(x_j \mid x_a, x_b) \cdot \max_l bd(j)$;
                $\sigma(j, x_j) = \mathrm{argmax}_l\, bd(j)$;
            end
            $J_{a:b}(k) = \max_{a<j<b} sel(j)$;
            $\pi_{a:b}(x_a, x_b; k) = \mathrm{argmax}_{a<j<b}\, sel(j)$;
            for $x_j \in \mathrm{dom}\, X_{\pi_{a:b}(k)}$ do
            $\sigma_{a:b}(x_a, x_b, x_j; k) = \sigma(\pi_{a:b}(k), x_j)$;
        end
    end
end
```

Algorithm 2: Computation of optimal conditional plan.

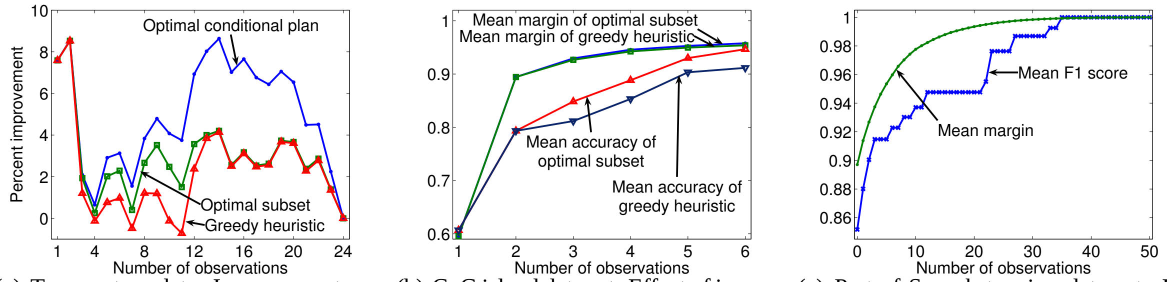
```
Input: Budget $k$, observations $X_a = x_a$, $X_b = x_b$, $\sigma$, $\pi$
begin
    $j := \pi_{a:b}(x_a, x_b; k)$;
    if $j \ge 0$ then
        Observe $X_j = x_j$;
        $l := \sigma_{a:b}(x_a, x_b, x_j; k)$;
        Recurse with $k := l$, $a := a$, $b := j$;
        Recurse with $k := k - l - \beta_j$, $a := j$, $b := b$;
    end
end
```

Algorithm 3: Observation selection using conditional plan.

6 Theoretical Limits

Many problems that can be solved efficiently for discrete chain graphical models can also be efficiently solved for discrete polytrees. Examples include probabilistic inference and the most probable explanation (MPE). Surprisingly, we prove that for the optimization problems discussed in this paper, this generalization is not possible, unless $\mathbf{P} = \mathbf{NP}$. All proofs in this section are stated in the Appendix.

In order to solve the optimization problems, we will most likely have to evaluate the objective function, i.e., the expected local rewards. Our first result states that this problem is intractable even for discrete polytrees.



(a) Temperature data: Improvement over the uniform spacing heuristic. (b) CpG island data set: Effect of increasing the number of observations on margin and classification accuracy. (c) Part-of-Speech tagging data set: Effect of increasing the number of observations on margin and F1 score.

Figure 2: Experimental results.

Theorem 3. *The computation of expected local rewards for discrete polytrees is* $\#\mathbf{P}$*-complete.*[1] □

This negative result can be specialized to the conditional entropy, one of the most frequently used reward function to characterize the residual uncertainty in value of information problems.

Corollary 4. *The computation of conditional entropy for discrete polytrees is* $\#\mathbf{P}$*-complete.* □

Since evaluating local rewards is $\#\mathbf{P}$-complete, it can be suspected that the subset selection problem is at least $\#\mathbf{P}$-hard. We show that it is even $\mathbf{NP}^{\mathbf{PP}}$-complete[2], a complexity class containing problems that are believed to be significantly harder than $\mathbf{NP}$ or $\#\mathbf{P}$ complete problems. This result provides a complexity theoretic classification of value of information, a core AI problem.

Theorem 5. *Subset Selection is* $\mathbf{NP}^{\mathbf{PP}}$*-complete even for discrete polytrees.* □

For our running example, this implies that the generalized problem of optimally selecting k sensors from a network of correlated sensors is most likely computationally intractable without resorting to heuristics. A corollary extends the hardness of subset selection to the hardness of conditional plans.

Corollary 6. *Computing conditional plans is* $\mathbf{NP}^{\mathbf{PP}}$*-hard even for discrete polytrees.* □

7 Experiments

In this section, we evaluate the proposed methods for several real world data sets. A special focus is on the comparison of the optimal methods with the greedy heuristic and other heuristic methods for selecting observations, and on how the algorithms can be used for interactive structured classification.

7.1 Temperature time series

The first data set consists of temperature time series collected from a sensor network deployed at Intel Research Berkeley [4] as described in our running example. Data was continuously collected for 19 days, linear interpolation was used in case of missing samples. The temperature was measured once every 60 minutes, and it was discretized into 10 bins of 2 degrees Kelvin. To avoid overfitting, we used pseudo counts $\alpha = 0.5$ when learning the model. Using parameter sharing, we learned four sets of transition probabilities: from 12 am - 7am, 7 am - 12 pm, 12 pm - 7 pm and 7 pm - 12 am. Combining the data from three adjacent sensors, we got 53 sample time series.

The goal of this task was to select k out of 24 time points during the day, during which sensor readings are most informative. The experiment was designed to compare the performance of the optimal algorithms, the greedy heuristic, and a uniform spacing heuristic, which distributed the k observations uniformly over the day. Fig. 2(a) shows the relative improvement of the optimal algorithms and the greedy heuristic over the uniform spacing heuristic. The performance is measured in decrease of expected entropy, with zero observations as the baseline. It can be seen that if k is less than about the half of all possible observations, the optimal algorithms decreased the expected uncertainty by several percent over both heuristics. The improvement gained by the optimal plan over the subset selection algorithms appears to become more drastic if a large number of observations (over half of all possible observations) is allowed. Furthermore, for a large number of observations, the optimal subset and the subset selected by the greedy heuristic were almost identical.

7.2 CpG-Island detection

We then studied the bioinformatics problem of finding CpG islands in DNA sequences. CpG islands are regions in the genome with a high concentration of the cytosine-guanine sequence. These areas are believed to be mainly located around the promoters of genes, which are frequently expressed in the cell. In our experiment, we considered the gene loci HS381K22, AF047825 and AL133174, for which the GenBank annotation listed three, two and one CpG islands each. We ran our algorithm on a 50 base window at the beginning and end of each island, using the transition and emission probabilities from [6] for our Hidden Markov Model, and we used the sum of margins as reward function.

The goal of this experiment was to locate the beginning and ending of the CpG islands more precisely by asking experts, whether or not certain bases belong to the CpG region or not. Fig. 2(b) shows the mean classification accuracy and mean margin scores for an increasing number of observations. The results indicate that, although the expected margin scores are similar for the optimal algorithm and the greedy heuristic, the mean classification performance of the optimal algorithm was still better than the performance of the greedy heuristic.

[1] $\#\mathbf{P}$ contains problems such as counting the number of satisfying assignments to a Boolean formula.

[2] $\mathbf{NP}^{\mathbf{PP}}$ is natural for AI planning problems [9]. A complete problem is $EMAJSAT$, where one has to find an assignment to the first k variables of a 3CNF formula, such that the formula is satisfied under the majority of assignments to the remaining variables.

7.3 Part-of-Speech Tagging

In our third experiment, we investigated the structured classification task of part-of-speech (POS) tagging [3]. Problem instances are sequences of words (sentences), where each word is part of an entity (e.g., "United States of America"), and each entity belongs to one of five categories: Location, Miscellaneous, Organization, Person or Other. Imagine an application, where automatic information extraction is guided by an expert: Our algorithms compute an optimal conditional plan for asking the expert, trying to optimize classification performance while requiring as little expert interaction as possible.

We used a conditional random field for the structured classification task, where each node corresponds to a word, and the joint distribution is described by node potentials and edge potentials. The sum of margins was used as reward function. Measure of classification performance was the F1 score, the geometric mean of precision and recall. The goal of this experiment was to analyze how the addition of expert labels increases the classification performance, and how the indirect, decomposing reward function used in our algorithms corresponds to real world classification performance.

Figure 2(c) shows the increase of the mean expected margin and F1 score for an increasing number of observations, summarized over ten 50 word sequences. It can be seen that the classification performance can be effectively enhanced by optimally incorporating expert labels. Requesting only three out of 50 labels increased the mean F1 score from by more than five percent. The following example illustrates this effect: In one scenario both words of an entity, the sportsman 'P. Simmons', were classified incorrectly – 'P.' as *Other* and 'Simmons' as *Miscellaneous*. The first request of the optimal conditional plan was to label 'Simmons'. Upon labeling this word correctly, the word 'P.' was automatically labeled correctly also, resulting in an F1 score of 100 percent.

8 Related Work

Decision Trees [12] popularized the value of information as a criterion for creating conditional plans. Unfortunately, there are no guarantees on the performance of this greedy method. Bayer-Zubek [1] proposed a heuristic method based on the Markov Decision Process framework. Several researchers [15; 5] suggested myopic, i.e., greedy approaches for selectively gathering evidence in graphical models. Heckerman *et al.* [7] propose a method to compute the maximum expected utility for specific sets of observations. While their work considers more general graphical models than this paper, they provide only large sample guarantees for the evaluation of a given sequence of observations, and use a heuristic without guarantees to select such sequences The subset selection problem as an instance of feature selection is a central issue in machine learning, with a vast amount of literature (see [10] for a survey). The problem of choosing observations also has a strong connection to the field of active learning [2] in which the learning system designs experiments based on its observations.

9 Conclusions

We have described novel efficient algorithms for optimal subset selection and conditional plan computation in chain graphical models, including HMMs. Empirical evaluation indicates that these algorithms can improve upon commonly used heuristics for decreasing expected uncertainty. Our algorithms can also effectively enhance performance in interactive structured classification tasks.

Unfortunately, the optimization problems become intractable for even a slight generalization of chains. We presented surprising theoretical limits, which indicate that commonly used local reward functions, such as conditional entropies, cannot be efficiently computed even in discrete polytree graphical models. We also identified optimization of value of information as a new class of problems that are intractable ($\mathbf{NP}^{\mathbf{PP}}$-complete) for polytrees.

Our hardness results, along with other recent results for polytree graphical models, the $\mathbf{NP}$-completeness of maximum a posteriori assignment [11] and $\mathbf{NP}$-hardness of inference in conditional linear Gaussian models [8], suggest the possibility of developing a generalized complexity characterization of problems that are hard in polytree graphical models.

In light of these theoretical limits for computing optimal solutions, it is a natural question to ask whether approximation algorithms with non-trivial performance guarantees can be found. We are currently focusing our research in this direction.

Acknowledgements. We would like to thank Ben Taskar for providing the part-of-speech tagging model, and Reuters for making their news archive available. We would also like to thank Brigham Anderson and Andrew Moore for helpful comments and discussions.

Appendix

Proof of Theorem 3. Membership in $\#\mathbf{P}$ is straightforward. To show hardness, we use a construction similar to the one presented in [11] for the maximum a posteriori problem. Let ϕ be an instance of $\#3SAT$, where we have to count the number of assignments to $X_1, \ldots, X_n$ satisfying ϕ. Let $C = \{C_1, \ldots, C_m\}$ be the set of clauses. Now create a Bayesian network with nodes U_i for each X_i, each with uniform Bernoulli prior. Add variables Y_0, which uniformly varies over $\{1, \ldots, m\}$ and $Y_1, \ldots, Y_n$ with CPTs defined the following way:

$$Y_i \mid [Y_{i-1} = j, U_i = u_i] \sim \begin{cases} 0, & \text{if } j = 0, \text{ or } U_i = u_i \\ & \text{satisfies clause } C_j; \\ j, & \text{otherwise.} \end{cases}$$

In this model, $Y_n = 0$ iff $U_1, \ldots, U_n$ encode a satisfying assignment of ϕ. Let all nodes have zero reward, except for Y_n, which is assigned the following reward:

$$R(Y_n \mid O = o) = \begin{cases} 2^n, & \text{if } P(Y_n = 0 \mid O = o) = 1; \\ 0, & \text{otherwise.} \end{cases}$$

Since the prior probability of any assignment is 2^{-n}, the expected reward $R(Y_n \mid U_1, \ldots, U_n)$ is exactly the number of satisfying assignments to ϕ. □

Proof of Corollary 4. We start from the same construction as in the proof of Theorem 3, and add an additional random variable Z after Y_n on the chain. $Z \mid Y_n$ is 0 if $Y_n = 0$, and takes uniformly random values in $\{0, 1\}$ if $Y_n \neq 0$. Then $H(Z \mid U = u)$ is 0 if u is a satisfying assignment, and 1 otherwise. Hence $H(Z \mid O) = 1 - K2^{-n}$, where K is the number of satisfying assignments to ϕ. □

Proof of Theorem 5. Membership follows from Theorem 3. Let ϕ be an instance of $EMAJSAT$, where we have to find an instantiation of $X_1, \dots, X_n$ such that $\phi(X_1, \dots, X_{2n})$ is true for the majority of assignments to $X_{n+1}, \dots, X_{2n}$. Let $C = \{C_1, \dots, C_m\}$ be the set of 3CNF clauses. Create the Bayesian network shown in Fig. 3, with nodes U_i, each having a uniform Bernoulli prior. Add bivariate variables $Y_i = (sel_i, par_i)$, $0 \le i \le 2n$, where sel_i takes values in $\{0, \dots, m\}$ and par_i is a parity bit. The CPTs for Y_i are defined as: sel_0 uniformly varies over $\{1, \dots, m\}$, $par_0 = 0$, and for $Y_1, \dots, Y_{2n}$:

$$sel_i \mid [sel_{i-1} = j, U_i = u_i] \sim \begin{cases} 0, & \text{if } j = 0, \text{ or } u_i \text{ satisfies } C_j; \\ j, & \text{otherwise;} \end{cases}$$

$$par_i \mid [par_{i-1} = b_{i-1}, U_i] \sim b_{i-1} \oplus U_i,$$

where $\oplus$ denotes the parity (XOR) operator.

We now add variables Z_i^T and Z_i^F for $1 \le i \le n$ and let

$$Z_i^T \mid [U_i = u_i] \sim \begin{cases} \mathcal{I}(\{0,1\}), & \text{if } u_i = 1; \\ 0, & \text{otherwise;} \end{cases}$$

where $\mathcal{I}$ denotes the uniform distribution. Similarly, let

$$Z_i^F \mid [U_i = u_i] \sim \begin{cases} \mathcal{I}(\{0,1\}), & \text{if } u_i = 0; \\ 0, & \text{otherwise.} \end{cases}$$

Intuitively, $Z_i^T = 1$ guarantees us that $U_i = 1$, whereas $Z_i^T = 0$ leaves us uncertain about U_i. The case of Z_i^F is symmetric.

We use the subset selection algorithm to choose the Z_is that encode the solution to $EMAJSAT$. If Z_i^T is chosen, it will indicate that X_i should set to true, similarly Z_i^F indicates a false assignment to X_i. The parity function is going to be used to ensure that exactly one of $\{Z_i^T, Z_i^F\}$ is observed for each i.

We first assign penalties ∞ to all nodes except Z_i^T, Z_i^F for $1 \le i \le n$, and U_j for $n+1 \le j \le 2n$, which are assigned zero penalty. Let all nodes have zero reward, except for Y_{2n}, which is assigned the following reward:

$$R(Y_{2n} \mid O = o) = \begin{cases} 4^n, & \text{if } P(sel_{2n} = 0 \mid O = o) = 1 \text{ and} \\ & [P(par_{2n} = 1 \mid O = o) = 1 \text{ or} \\ & \;P(par_{2n} = 0 \mid O = o) = 1]; \\ 0, & \text{otherwise.} \end{cases}$$

Note that $sel_{2n} = 0$ with probability 1 iff $U_1, \dots, U_{2n}$ encode a satisfying assignment of ϕ, as in the proof of Theorem 3. Furthermore, we get positive reward only if we are both certain that $sel_{2n} = 0$, i.e., the chosen observation set must contain a proof that ϕ is satisfied, and we are certain about par_{2n}. The parity certainty will only occur if we are certain about the assignment $U_1, \dots, U_{2n}$. It is only possible to infer the value of each U_i with certainty by observing one of U_i, Z_i^T or Z_i^F. Since, for $i = 1, \dots, n$, the cost of observing U_i is ∞, to receive any reward we must observe at least one of Z_i^T or Z_i^F. Assume that we compute the optimal subset $\hat{O}$ for budget $2n$, then we can only receive positive reward by observing exactly one of Z_i^T or Z_i^F.

We interpret the selection of Z_i^T and Z_i^F as an assignment to the first n variables of $EMAJSAT$. Let $\hat{R} = R(Y_{2n} \mid \hat{O})$. We claim that $\phi \in EMAJSAT$ if and only if $\hat{R} > 0.5$. First let $\phi \in EMAJSAT$, with assignment $x_1, \dots, x_n$ to the first n variables. Now add $U_{n+1}, \dots, U_{2n}$ to O and add Z_i^T to O iff $x_i = 1$ and Z_i^F to O iff $x_i = 0$. This selection guarantees $\hat{R} > 0.5$.

Now assume $\hat{R} > 0.5$. We call an assignment to $U_1, \dots, U_{2n}$ *consistent* if for any $1 \le i \le n$, if $Z_i^T \in \hat{O}$, then $U_i = 1$ and if $Z_i^F \in \hat{O}$ then $U_i = 0$. For any consistent assignment, the chance that the observations Z_i prove the consistency is 2^{-n}. Hence $\hat{R} > 0.5$ implies that the majority of all provably consistent assignments satisfy ϕ and hence $\phi \in EMAJSAT$. This proves that subset selection is $\mathbf{NP^{PP}}$ complete. □

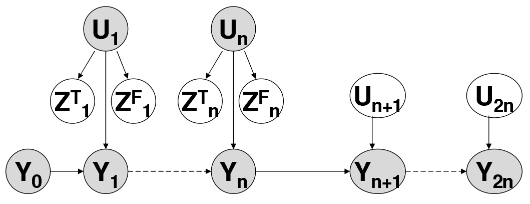


Figure 3: Graphical model used in proof of Theorem 6.

Proof of Corollary 6. The construction in the proof of Theorem 5 also proves that computing conditional plans is $\mathbf{NP^{PP}}$-hard, since, in this instance, any plan with positive reward must observe all $U_{n+1}, \dots, U_{2n}$ and one each of the $Z_1, \dots, Z_n$, to satisfy the parity condition. In this case, the order of selection is irrelevant, and, hence, the conditional plan effectively performs subset selection. □

References

[1] V. Bayer-Zubek. Learning diagnostic policies from examples by systematic search. In *UAI*, 2004.

[2] D. A. Cohn, Z. Gharamani, and M. I. Jordan. Active learning with statistical models. *J AI Research*, 4:129–145, 1996.

[3] CoNLL. Conference on computational natural language learning shared task. http://cnts.uia.ac.be/conll2003/ner/, 2003.

[4] A. Deshpande, C. Guestrin, S. Madden, J. Hellerstein, and W. Hong. Model-driven data acquisition in sensor networks. In *VLDB*, 2004.

[5] S. Dittmer and F. Jensen. Myopic value of information in influence diagrams. In *UAI*, pages 142–149, San Francisco, 1997.

[6] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis : Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1999.

[7] D. Heckerman, E. Horvitz, and B. Middleton. An approximate nonmyopic computation for value of information. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15:292–298, 1993.

[8] U. Lerner and R. Parr. Inference in hybrid networks: Theoretical limits and practical algorithms. In *UAI*, 2001.

[9] M. Littman, J. Goldsmith, and M. Mundhenk. The computational complexity of probabilistic planning. *Journal of Artificial Intelligence Research*, 9:1–36, 1998.

[10] L. Molina, L. Belanche, and A. Nebot. Feature selection algorithms: A survey and experimental evaluation. In *ICDM*, 2002.

[11] J. D. Park and A. Darwiche. Complexity results and approximation strategies for map explanations. *Journal of Aritificial Intelligence Research*, 21:101–133, February 2004.

[12] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[13] T. Scheffer, C. Decomain, and S. Wrobel. Active learning of partially hidden markov models for information extraction. In *ECML/PKDD Workshop on Instance Selection*, 2001.

[14] P. D. Turney. Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. *Journal of Artificial Intelligence Research*, 2:369–409, 1995.

[15] L. van der Gaag and M. Wessels. Selective evidence gathering for diagnostic belief networks. *AISB Quart.*, 86:23–34, 1993.