Diagnose and Decide: An Optimal Bayesian Approach

Christopher Amato
CSAIL
MIT
camato@csail.mit.edu

Emma Brunskill
Computer Science Department
Carnegie Mellon University
ebrun@cs.cmu.edu

Abstract

Many real-world scenarios require making informed choices after some sequence of actions that yield noisy information about a latent state. Prior research has mostly focused on generic methods that struggle to scale to large domains. We focus on a subclass with two particular characteristics. First, once performed, an information gathering action (or test) will always yield the same result. This means it is sufficient to perform each test once. Second, we assume that test costs can be expressed in the same units as costs of the final decisions made. We call such scenarios diagnose-and-decide problems. We prove diagnose-and-decide problems are a special subclass of POMDPs for which the optimal policy can be computed in time polynomial in the number of possible test outcomes. We use a simple algorithm that takes advantage of the problem structure, and show our approach can equal or outperform a state-of-the-art POMDP planner. We demonstrate this performance on two simulations based on real-world data (colon cancer screening and object recognition) as well as a large synthetic domain.

Consider a doctor who may run diagnostic tests before treating a patient, or an autonomous surveillance helicopter that may fly closer to a target to get a better view before raising a security alert, or a drug discovery problem where a drug can be tested in various ways before deciding whether or not to develop and market it. These are all instances where there exists some hidden state that affects the quality of a resulting decision, and a decision maker has the opportunity run some, potentially costly, information-gathering operations before making that decision. In the above three cases, and many others, the results of the information-gathering operations will not change when they are repeated due to some underlying bias. In addition, we are interested in when the cost of the information-gathering operations is directly comparable to the cost of the final decision made, such as the time to raise an alert, or the financial cost of testing and then producing a drug. We call these problems diagnose-and-decide (DAD) problems. In this paper we focus on computing optimal conditional policies for DAD problems that specify what information-gathering operations to perform, and what final decision to make, in order to minimize the expected total cost. We show the special structure of DADs allows them to fall in a lower complexity class than generic partially observable Markov decision processes (POMDPs). We present a branch and bound approach to solving DADs, and show, somewhat surprisingly, that our approach achieves similar or better performance than POMCP [10], a state-of-the-art approximate POMDP solver, for the domains tried. Unlike POMCP, our approach also provides anytime bounds on the resulting performance.

1 Diagnose and decide model

We denote a diagnose-and-decide (DAD) model as a tuple consisting of the following components:

- Actions: $A_T \times A_D$. A_T is the set of possible tests ($|A_T| = T$). Each test can only be performed once. A_D ($|A_D| = D$) is the set of possible final decisions.
- States: $S_T \times S_C$, s_D. S_T is a vector representing the outcome (or not yet been taken status) of each test. S_T is fully observable. S_C ($|S_C| = Y$) is a finite set of classes with the underlying class hidden. s_D is an absorbing terminal state.

- Observations: L. The set of possible observed labels for a test. W.l.o.g. we assume all tests have the same labels.
- Transition model: (i) The hidden class s_c is static. (ii) If test $t_i \in A_T$ is performed, t_i's entry in the state is updated, $s_x(t_i) \rightarrow l$. No other values change. (iii) If decision $d_j \in A_D$ is made, transition to s_D.
- Observation model: $p(l|s_c, t_i)$, $t_i \in A_T$. The probability of a label depends only on the hidden state, unless the test has been previously performed, in which case a null observation is received.
- Cost model: (i) $R(s, a_{t_i}) = c(t_i)$ cost for a test. (ii) $R(s_c, a_{d_j})$: cost for a particular decision for a given state.
- Initial belief: b_0. $b_0(s_c)$ defines the initial probability of being in class s_c. $\sum_{s_c} b_0(s_c) = 1$.

We maintain a probability distribution (known as a belief) over the possible hidden class. Given any initial belief b, after performing test t_i and observing label l, the new belief state $b^{t_i,l}$ is

$$b^{t_i,l}(s_c) = \frac{p(l|s_c, t_i)b(s_c)}{p(l|b, t_i)}, \tag{1}$$

where $p(l|b, t_i) = \sum_{s_c} p(l|s_c, t_i)b(s_c)$. A policy π is a mapping from beliefs to actions. The value $V^\pi(b)$ is the expected total cost of executing π from belief b. Our goal is to determine the optimal π with the lowest V^π.

2 Related work

The diagnose and decide model is related to several previously considered problems. Adaptive sensing domains [3] are similar but, in contrast to DAD models, the underlying state may change. In sequential analysis "Wald problems" involve repeated sampling before making a decision [11], but assume that repeated tests will generate independent outcomes. In Bayesian experimental design there is a prior over the possible hypotheses (i.e., classes) and probabilities for test outcomes [2]. If the observations are noise-free, this is an optimal decision-tree problem, and greedy methods can produce a solution that utilizes within $O(\log n)$ tests of an optimal approach where n is the number of hypotheses [6]. The closest related work is Golovin et al.'s [4] work on Bayesian experimental design in the presence of noisy observations. Their objective is to only consider policies which guarantee that when a decision is chosen that it is identical to the decision that would be made if all tests were performed (risk minimization). A decision tree can then be generated that determines what tests should be performed to decide which equivalence class is the correct one while minimizing cost. They prove that finding the minimum cost decision tree can be solved using adaptive submodularity. This results in a greedy solution which is efficient and has performance bounds relative to their objective function, but it does not allow the cost of the tests to be balanced with the loss that results from making a decision. For our objective of minimizing the total expected cost, it is easy to construct an example where using the objective of minimizing risk (as discussed above) can result in an arbitrarily higher cost than the minimal possible expected cost.

3 Complexity of solving diagnose and decide problems

General POMDPs may have an infinite number of reachable beliefs and finite horizon planning is known to be PSPACE hard. Fortunately, certain subclasses of POMDPs have fewer belief states and lie in lower complexity classes. For example, as discussed by Littman [8] and Bonet [1], deterministic POMDPs have at most a number of beliefs that is an exponential function of the number of states, $O(|S|^{|S|})$. We can represent a DAD as a deterministic POMDP by introducing an additional static hidden observation selector variable o to the state of cardinality $|L|^T$. Each (o, hidden class s_c) pair maps to a unique label for each test, and a probability distribution over the hidden variable o represents the original observation noise. This yields a deterministic POMDP with a state space of $Y|L|^T$ (only one class and associated set of test outcomes is the true state, but we do not know which). More efficient is a forward search approach, but this will still be exponential in the horizon for both the number of actions and test outcomes. However, we now show that DAD problems lie in a lower computational subclass.

Theorem 1. *An optimal policy for a diagnose and decide problem can be computed in time that is a polynomial function of* $(T, D, Y, (|L| + 1)^{(T+1)})$.

Proof. We first reduce a DAD to a belief-state Markov decision process M with $(L + 1)^T$ states and $A_T \times A_D$ actions. In a DAD problem, a belief state is completely defined by the initial belief, the history of which tests have been

performed, and the outcomes of those tests. Since the hidden class is static, and the test outcomes are independent given the hidden class, all histories with the same tests and resulting outcomes lead to identical belief states, independent of the order of the prior tests. At a particular point, each test can either not have yet been performed, or have been performed and resulted in one of L possible values. Therefore the number of histories with the same set of performed tests and associated labels, also known as reachable beliefs, is $(L+1)^T$. Thus, we can map a DAD to a MDP where the MDP's states correspond to belief states in the original DAD problem, and the MDP actions are the same as the DAD problem actions. The transition model in the MDP will be the probability of transitioning between two belief states in the DAD problem given the action taken is a particular test t_i. As seen in Equation 1, given an observation and test, the belief update equation is deterministic. Therefore the probability of transitioning between two beliefs b and b' given a test t_i is either the probability of observing the label l if b' is equal to the updated belief state of taking t_i and receiving l from b ($p(l|b,t_i)$), or 0. The cost of performing tests $c(t_i)$ is independent of the state and is the same in the corresponding MDP M. The cost of taking a decision is dependent on the underlying hidden class, and so MDP costs for a decision d_j in a particular state (which corresponds to a belief state b in the DAD model) will be equal to the expected cost

$$R(b,d_j) = \sum_{s_c} R(s_c,d_j)b(s_c). \tag{2}$$

The state values computed in this MDP will exactly correspond to the belief state values of the DAD model.

It is therefore possible to compute an optimal policy for a DAD with a computational complexity equal to the cost of computing the transition and reward parameters of this equivalent MDP, plus the cost of computing an optimal policy in this MDP. The computational cost of computing the reward model of the equivalent MDP involves evaluating Equation 2, which takes $O(Y)$ time, for all beliefs and all decisions, yielding a total cost of $O(D(|L|+1)^T Y)$. To compute the transition parameters for each belief state, we enumerate all possible remaining tests (which is at most T), and all possible labels that could be observed ($|L|$) and use Equation 1 to compute the resulting successor belief in each case, and the associated likelihood of this transition ($p(l|b,t_i)$). A belief update takes $O(Y)$, and so computing the next possible beliefs and probabilities of each belief for a single starting belief takes $O(T|L|Y)$ time. Repeating this for all possible beliefs yields a total time of $O((|L|+1)^T T|L|Y)$. Therefore the total time to compute the MDP transition and reward parameters is $O(D(|L|+1)^T Y + (|L|+1)^{(T+1)}TY)$. An optimal policy for a MDP can be computed in time polynomial in the number of states ($(|L|+1)^T$) and actions ($T+D$), therefore a DAD problem can be solved in time which is a polynomial function of ($(|L|+1)^{(T+1)}$,T,D,Y). □

4 Planning in DADs

For now, we use a forward search branch and bound approach for solving DADs, though we are interested in later exploring other alternative algorithms. Our algorithm adaptively constructs a forward search tree of reachable beliefs by considering possible sequences of actions (which could be tests or decisions) and their associated labels. The value of a leaf node b after a decision d_j is equal to Equation 2. Nodes are expanded using a heuristic, which in our case is a lower bound estimate of their expected cost. We prune the action branching factor by not expanding previously taken tests. In addition, of the possible decisions, only the decision with the highest expected value $d^* = \arg\max_{d_j} Q(b,d_j)$ needs to be added to the tree. Now, the expected value of any possible remaining test t_i at b can at most be the value of taking that test, followed by taking the optimal policy after each possible label. Let $\bar{V}(b)$ be an upper bound on the value of the belief, ($\bar{V}(b) >= V^*(b)$). Then an upper bound on the value of any possible remaining test t_i at b is

$$\bar{Q}(b,t) = -c(t_i) + \sum_l p(l|b,t_i)\bar{V}(b^{t_i,l}), \tag{3}$$

where $b^{t_i,l}$ is the updated belief if test t_i is taken and label l is observed. If the expected value of the best decision at this belief is higher than $\bar{Q}(b,t)$, $\max_d Q(d,b) \geq \bar{Q}(b,t)$, then any policy from this point that starts with t_i is strictly dominated. This allows us to only expand test nodes in the search tree that are non-dominated. The children of each of the non-dominated tests (which are the labels) are also added to a priority queue of nodes using a lower bound on their value. In addition, the estimated expected costs of the ancestors of an expanded node are updated using the newly computed leaf values. The algorithm proceeds by popping a node to expand from the priority queue and repeating the above procedure. This whole process repeats until there are no more nodes to expand.

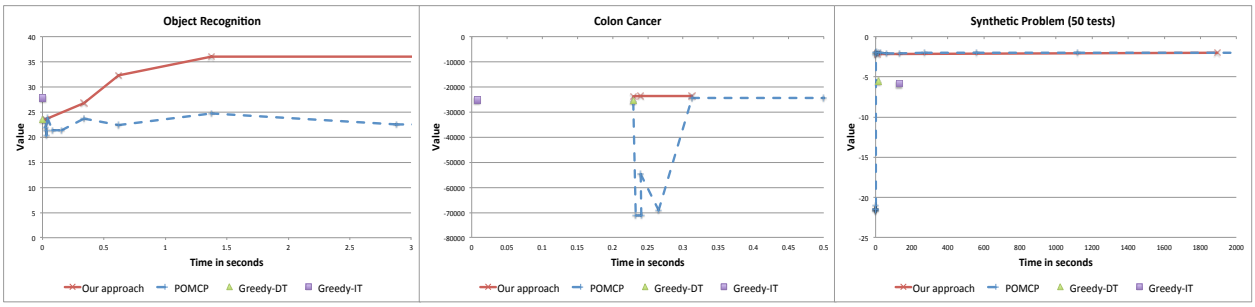

Figure 1: Results. Greedy-DT is the decision-theoretic method and Greedy-IT is the information theoretic method.

5 Experiments

We compare with three different methods. The greedy decision-theoretic algorithm chooses the action (i.e., test or decision) that has the highest one-step expected value given the current belief state. The greedy information-theoretic approach is an implementation of the Efficient Edge Cutting approXimate objective (EFFECXTIVE) algorithm of Golovin et al. [4]. We also compare to a state-of-the-art POMDP method called Partially Observable Monte-Carlo Planning (POMCP) [10]. POMCP is a Monte-Carlo tree search planning algorithm that has been shown to scale to extremely large domains, and achieve good performance in those domains. While POMCP has asymptotic convergence (with high probability), it has no guarantees given a finite number of trajectories. We also explored using MOMDP [9], but it took over 1 hr using 2GB of memory for a fairly small diagnose-and-decide problem (L=2,T=8,D=9,Y=9).

We considered three domains. One is an object recognition domain where the goal is to identify if one of three possible objects (or none) is in an image [5]: the tests are to run various computer vision techniques. The second domain is a colon cancer diagnosis problem based on real-world data [7]. There are 6 potential hidden patient states (healthy, low or high risk polyp, and localized, regionalized or distant cancer) and four diagnostic tests (colonoscopy, sigmoidoscopy, fecal occult blood test and a DNA stool test). The third domain is a simulated domain, with T tests, $T+1$ classes, $T+2$ decisions and 2 labels. We considered T to be 50 resulting in problems of size $2^{50} \times 51 \approx 5.74 \times 10^{16}$. The experimental results are shown in Figure 1 with value plotted vs running time. For our approach, anytime performance is produced by returning successively increasing lower bounds on the optimal solution. These intermediate values are given in the figures with the last data point representing the known optimal solution. The results from the greedy methods are given as points representing the one solution that is produced.

In the object recognition problem, our approach significantly outperforms the other methods in terms of value and does so in a relatively small amount of time. A near optimal solution is quickly found with our search method and then the optimal solution is found. POMCP failed to find a near optimal solution in this problem. In the colon cancer problem, all methods perform relatively well. Still the difference between the greedy solution and the optimal solution is thousands of dollars which will quickly add up to a very large amount of money over a number of patients. POMCP performs well, getting near optimal results relatively quickly, but our optimal approach outperforms it in terms of solution quality. In the synthetic domains, we again see that the greedy approaches perform well, but return suboptimal results. While POMCP performs almost identically to our approach in this problem, it does not provide a tight bound on the optimal solution.

6 Conclusion

We have described diagnose-and-decide problems, in which a number of information gathering tests can be performed and decisions can be made based on the various outcomes. These problems are relevant to a number of important application areas, including surveillance and medical treatment. Though DAD problems can be seen as a subclass of POMDPs, we proved that DAD problems can be optimally solved with a smaller computational complexity. We used an optimal algorithm that exploits the structure of DAD problems, and our empirical results demonstrated the benefit of our approach over greedy methods and a state-of-the-art POMDP planner. This suggests that large, real-world problems may be able to be solved optimally, rather than approximately or within only a loose bound of the optimal solution. We believe that further algorithmic improvements are possible that may lead to further performance gains and enhance scalability.

References

[1] B. Bonet and H. Geffner. Solving POMDPs: RTDP-Bel vs. point-based algorithms. In *IJCAI*, 2009.

[2] K. Chaloner and I. Verdinelli. Bayesian experimental design: a review. *Statistical Science*, 10(3):273304, 1995.

[3] E. K. Chong, C. M. Kreucher, and A. O. Hero, III. Partially observable markov decision process approximations for adaptive sensing. *Discrete Event Dynamic Systems*, 19(3):377–422, 2009.

[4] D. Golovin, A. Krause, and D. Ray. Near-optimal bayesian active learning with noisy observations. In *NIPS 23*. 2010.

[5] K. Kapoor, C. Amato, N. Srivastava, and P. Schrater. Using pomdps to control an accuracy-processing time tradeoff in video surveillance. In *IAAI*, 2012.

[6] S. R. Kosaraju, T. M. Przytycka, and R. S. Borgstrom. On an optimal split tree problem. In *Proceedings of the 6th International Workshop on Algorithms and Data Structures*, pages 157–168, 1999.

[7] M. Leshno, Z. Halpern, and N. Arber. Cost-effectiveness of colorectal cancer screening in the average risk population. *Health Care Management Science*, 6:165–174, 2003.

[8] M. Littman. *Algorithms for Sequential Decision Making*. PhD thesis, Brown University, Providence, RI, 1996.

[9] S. Ong, S. Png, D. Hsu, and W. Lee. Planning under uncertainty for robotic tasks with mixed observability. *International Journal of Robotics Research*, 29(8):1053–1068, 2010.

[10] D. Silver and J. Veness. Monte-carlo planning in large POMDPs. In *NIPS 23*. 2010.

[11] A. Wald. *Sequential Analysis*. John Wiley and Sons, 1947.