Lecture Notes: Influence Diagrams

Marc Toussaint
Machine Learning & Robotics group, TU Berlin
Franklinstr. 28/29, FR 6-9, 10587 Berlin, Germany

April 13, 2009

1 Influence Diagrams

Influence diagrams (Howard and Matheson, 1981) are similar to Bayesian Networks (BNs) but distinguish between random variables (graphically: circles), utility variables (diamonds), and decision variables (boxes), see Figure 1. An influence diagram is a DAG connecting these variables, where the arcs have different semantics depending on the type of the child variable:

- for random variables $x_{1,..,I}$, the incoming arcs define the conditional probability $P(x_i \,|\, \nu_{x_i})$ (ν=parents) as in Bayesian networks,
- for decision (action) variables $a_{1,..,K}$, the incoming arcs define on which inputs the *decision rule* (local policy) π_k depends, $a_k = \pi_k(\nu_{a_k})$,
- for utility (reward) variables $r_{1,..,T}$, the incoming arcs define on which inputs the utility depends, $r_t = R_t(\nu_{r_t}) \in \mathbb{R}$.

The set of all decision rules is called *policy* $\pi = (\pi_1, .., \pi_K)$. The objective is to find a policy that maximizes the expected accumulated utility

$$J(\pi) := \text{E}\{\sum_{t=1}^{T} r_t; \pi\} \ . \tag{1}$$

It has early been observed that in certain cases finding optimal decision rules can be solved by a series of inference queries (Cooper, 1988; Pearl, 1988; Shachter, 1988). The assumptions made are

- we have only *one utility variable* r,
- *regularity:* there exists a directed path containing all decision nodes, this introduces a decision order and w.l.o.g. we enumerate decisions $a_{1,..,K}$ along this path, and
- *no-forgetting:* all decisions get all previous decisions and their input as input

The first step is to reinterpret the utility variable as a binary random variable $\hat r$ with probability

$$P(\hat r\!=\!1 \,|\, \nu_R) = \frac{R(\nu_R) - \min(R)}{\max(R) - \min(R)} \ , \tag{2}$$

which rescales utilities to the interval $[0, 1]$. Shachter and Peot (1992) cite for this idea references as early as (Raiffa, 1968; von Neumann and Morgenstern, 1947). In this setup one can find optimal policies by recursing backward through the decisions: With respect to the last decision a_K the history is a single large random variable ν_{a_K} to which it has full access to (no-forgetting). The effective joint model reads P(history) P(decision|history) P(future|decision,history) P(utility|future,decision,history). Since the future does not depend on yet unknown decisions it can be eliminated and we have the joint

$$P(\hat r, a_K, \nu_{a_K}) = P(\hat r \,|\, a_K, \nu_{a_K})\ P(a_K \,|\, \nu_{a_K})\ P(\nu_{a_K}) \ . \tag{3}$$

The optimal last decision is

$$\pi^*_k(\nu_{a_k}) = \underset{a_k}{\text{argmax}}\, P(\hat r\!=\!1 \,|\, a_k, \nu_{a_k}) = \underset{a_k}{\text{argmax}}\, \frac{P(a_k, \nu_{a_k} \,|\, \hat r\!=\!1)\ P(\hat r\!=\!1)}{P(\nu_{a_k})\ 1/N_k} \ . \tag{4}$$

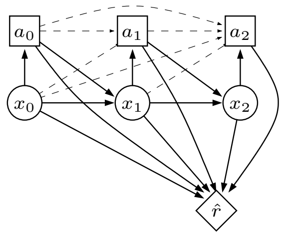

Figure 1: A finite-horizon MDP with non-stationary policy and single utility variable (e.g., as the sum of rewards) as an influence diagram. Original approaches (Cooper, 1988; Pearl, 1988; Shachter, 1988) assume no-forgetting (dashed arcs). Problem: no-forgetting assumption and moralization gives large clique.

The expression $P(a_k, \nu_{a_k} \mid \hat{r} = 1)$ is an inference query. The original algorithms recurse backward through the decisions, at each step solve the respective inference query and reassign the decision CPTs deterministically to the optimal decisions. (Tatman and Shachter, 1990) generalize this approach to the case of multiple utility variables (see also Zhang 1998). From a complexity point of view the no-forgetting assumption made in these classical approaches seems problematic: The last decision has many inputs. The respective clique essentially encompasses all the history which makes inference inefficient. Subsequent work (Zhang et al., 1992; Jensen et al., 1994; Zhang, 1994, 1998) developed methods to better exploit a given structure of the influence diagram. Kjaerulff and Madsen (2008) present a modern text book on influence diagrams including interesting work on solving continuous state problems similar to LQG methods in stochastic optimal control (Madsen and Jensen, 2005). Cano et al. (2006) use Monte Carlo methods for computing optimal policies in continuous state influence diagrams.

In summary, the early work on influence diagrams has formulated the problem of decision making as a series of inference queries, starting with the last decision and recursing backward through all decisions, where the inference query is a means to solve the argmax in equation (4) in each step. This backward recursion is a form of dynamic programming exploiting the optimality principle – in fact, applying the method on a finite-horizon Markov Decision processes is equivalent to Value Iteration where the argmax in each iteration is solved be means of an inference query. Our approach will adopt the idea of introducing a binary reward random variable as in equation (2). However, influence diagrams are significantly different from stationary, sequential, infinite-horizon decision processes like MDPs (see also section 4.3 in (Boutilier et al., 1999)):

- MDPs are infinite processes. In that respect, there is no "last decision". When trying to apply inference-in-influence-diagram methods, it is unclear where/when in the future to start the backward recursion. In fact, the *mixture* of variable length processes that we propose below will address exactly this issue.

- In MDPs the optimal policy is known to be stationary, i.e., independent of time. If one exploits this knowledge, then each decision at a time t is parameterized by the *same* CPT parameter $\pi_{ax} = P(a_t = a \mid x_t = x)$. In this setup one cannot choose or optimize some future decision a_K independent from an earlier decision (as in influence diagrams) because these should follow the same policy (parameter sharing in BN terms). A backward recursion to determine optimal policies is not possible in this setup. [1]

In textbooks on influence diagrams (Kjaerulff and Madsen, 2008) the infinite-horizon or stationary sequential decision problem is addressed based on the Bellman optimality equation and recursive value function computation instead of inference methods. To my knowledge, inference methods for solving infinite-horizon or stationary scenarios have not been developed in the context of influence diagrams.

References

C. Boutilier, T. Dean, and S. Hanks. Decision theoretic planning: structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, 11:1–94, 1999.

Andrés Cano, Manuel Gómez, and Serafín Moral. A forward-backward monte carlo method for solving influence diagrams. *International Journal of Approximate Reasoning*, 42:119–135, 2006.

G.F. Cooper. A method for using belief networks as influence diagrams. In *Procceedings of the Fourth Workshop on Uncertainty in Artificial Intelligence*, pages 55–63, 1988.

[1] One could decide not to exploit the knowledge of a stationary policy, optimize them separately, and let this implicitly converge to an optimal and stationary policy. However, this seems inefficient and again raises the question of where is the last decision in the infinite process.

R.A. Howard and J.E. Matheson. Influence diagrams. In R.A. Howard and J.E. Matheson, editors, *Readings on the Principles and Applications of Decision Analysis*, volume II. Menlo Park CA: Strategic Decisions Group (1984), 1981.

Frank Jensen, Finn V. Jensen, and S??ren L. Dittmer. From influence diagrams to junction trees. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pages 367–373. Morgan Kaufmann, 1994.

Uffe B. Kjaerulff and Anders L. Madsen. *Bayesian Networks and Influence Diagrams: A Guide to Construction and Analysis*. Information Science and Statistics. Springer, 2008.

A.L. Madsen and F. Jensen. Solving linear-quadratic conditional gaussian influence diagrams. *International Journal of Approximate Reasoning*, 38:263–282, 2005.

Judea Pearl. *Probabilistic Reasoning In Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

R. D. Shachter. Probabilistic inference and influence diagrams. *Operations Research*, 36:589–605, 1988.

R.D. Shachter and Peot. Decision making using probabilistic inference methods. In *Procceedings of the Eigth Conference on Uncertainty in Artificial Intelligence*, pages 276–283, 1992.

J.A. Tatman and R.D. Shachter. Dynamic programming and influence diagrams. *IEEE Transactions on Systems, Man and Cybernetics*, 20:365–379, 1990.

N. L. Zhang. Computational theory of decision networks, 1994.

N. L. Zhang. Probabilistic inference in influence diagrams. *Computational Intelligence*, 14(4):475–497, 1998.

N. L. Zhang, R. Qi, and D. Poole. Stepwise-decomposable influence diagrams. In *Proc. of the 3rd Conference on Knowledge representation*. Cambridge, Mass. USA, 1992.