Statistical Techniques in Robotics (16-831, F11)	Lecture 6 (9/17/12)
Graphical Models	
Lecturer: Drew Bagnell	*Scribe: Adam Stambler* [1]

1 Graphical Models

Graphical models are a framework for reasoning about uncertain quantities and the structural relationships between them. They are a union of probability and graph theory. Nodes represent random variables and edges represent the links, or relationships between these random variables. Graphical models are analogous to a circuit diagram — they are written down to visualize and better understand a problem.

Graphical models can be viewed as a:

- **Communication tool** that helps to *compactly* express dependencies between beliefs about a system.
- **Reasoning tool** that can be used to *extract* relationships that were not obvious when formulating the problem. In particular graphical models enable us to visualize conditional independence.
- **Computational skeleton** that helps organize how we perform computations on random variables.

We will examine four types of graphical models:

- **Bayes' Nets** (Directed Graphical Models)
- **Gibbs Fields** (Undirected Graphical Models)
- **Marlov Random Fields** (Undirected Graphical Models)
- **Factor Graphs** (Undirected Graphical Models)

2 Bayes' nets

One of the most common graphical models is called a Bayes' net. Bayes' nets are also known as Bayesian networks, belief networks, directed graphical models, and directed independence diagrams. In short, a Bayes' net is a directed acyclic graph with nodes representing uncertain quantities (random variables) and edges that encode relationships between them (often causal).

As an example, we take the random variables A, B, C, and D. Without any knowledge about their dependencies, we can write down the joint probability as $P(A, B, C, D)$. Every variable is

[1] Based on 2011 scribe of Yair Movshovitz-Attias and 2010 scribe by Brian Coltin and Hugh Cover

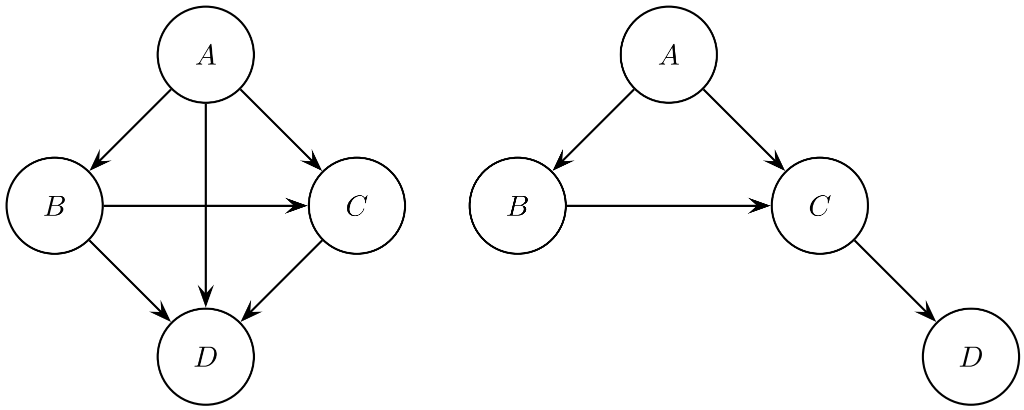


Figure 1: (Left) A generic Bayesian network of 4 random variables. (Right) A bayesian network which uses prior independence information to remove edges.

dependent on each other. In its graphical model, Figure 1 Left, every node is connected to each other. When we deal with this generic system $P(A, B, C, D)$, the chain rule of probability allows us to to factorize probability distribution as

$$P(A, B, C, D) = P(A)P(B|A)P(C|A, B)P(D|A, B, C)$$

This factorization *always* holds, and is not dependent on any particular graphical model. To verify, rename (B, C, D) to γ, and so

$$P(A, \gamma) = P(A)P(\gamma|A)$$

and continue recursively. This is just a recursively application of the definition of conditional probability to the probability distribution. It is known as *The Chain Rule of Probability.*

However, in real systems we know often know more about how they are related. In Figure 1 right, we encode this prior knowledge in the Bayes' net by removing edges (A,D) and (B,D) in the graph. These independences are then encoded in chain rule by expanding the probability as :

$$P(A, B, C, D) = P(A)P(B|A)P(C|A, B)P(D|C)$$

For an arbitrary Bayes' net with nodes $x_1, x_2, \ldots, x_n \in X$, we can derive the joint distribution $P(X)$ as the product of each node x_i given its parents $\pi(x_i)$.

$$P(X) = \prod_{x_i} P(x_i|\pi(x_i))$$

The simplest case of this expansion can be seen via Figure 2. $P(A, B, C, D) = P(A)P(B|A)P(C|B)P(D|C)$

Note that this factorization strategy only works if there are no cycles in the graph, and that Bayes' nets are acyclic by definition.

Simple Example

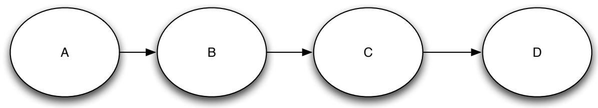


Figure 2: A simple Bayesian network.

One should note, that when we write these Bayes networks out based on our experience, we will often encode the relationships causally. We will draw an edge from A to B because A causes B normally. However, it is important to remember that the arrow's direction does not need to be causal. It just needs to show the correlation of the two variables probabilities. One should think of A as influencing B and C rather than A causing B and C. If all the arrows on a Bayes' net are flipped, then the resulting Bayes' net is equivalent to the original, since they both represent the same joint probability distribution.

In general, the absence of arrows is important in a Bayes net: *less* arrows mean *more* structure.

Note that in the Bayes net of Figure 1 if we add an edge between nodes A and D and between nodes B and D, we remove all structure form the graph!

2.1 Determining Dependencies

Bayes' nets can be used to quickly determine whether pairs of variables are dependent on each other. This is done by following all available paths between the two variables and checking if the path is 'blocked'. A path is any sequence of edge connected nodes leading from the first variable to the second (Note that edges can be traversed opposite to their direction). Blockages are determined by visiting each node on a path and checking the structure of surrounding nodes and edges against 3 rules explained below.

Two variables are independent if *all* available paths between them are blocked. Any unblocked paths show a possibility of dependence. It should be noted that this analysis can only be as good as the Bayes' net it is based on; an incomplete net may be missing paths that show a dependence.

2.1.1 Rule 1: Markov Chain

Figure 3 is a Bayes' net representation of a simple Markov chain.

An example of such a chain is the process of robot localization, although the usual z_i and u_i terms have been omitted for simplicity. If the robot knows the current state, x_2, then it does not need any information about past states, x_1, in order to determine the next state, x_3. For example if the robot can only be on the 1_{st} or 2_{nd} floor of a building and it knows that it's previous state was that it was on the 1_{st} floor. As there were no inputs any previous information about the robot's state is irrelevant, it must still be on the 1_{st} floor. This is the same as saying that x_1 and x_3 are

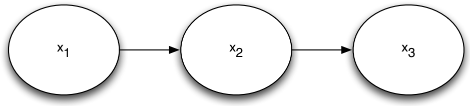


Figure 3: A Bayesian network representation of a Markov chain.

independent if x_2 is known.

$$P(x_3|x_2, x_1) = P(x_3|x_2)$$

In the case where x_2 is not known then knowledge of past states could provide information on the current state x_3. If our robot did not know it's state at x_2 but knew it was on the 1_{st} floor at x_1 then, given no inputs, x_3 has a high probability of being the 1_{st} floor. This is the same as saying that x_1 and x_3 could be dependent if x_2 is not known.

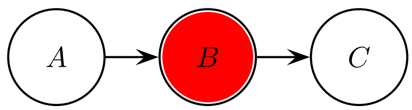


Figure 4: Markov chain is **BLOCKED** given B.

The rule is therefore that in a chain of nodes, as shown in Figure 4, C is independent from A if B is known. This means there is a blockage on any path passing through a Markov chain with a known middle node.

2.1.2 Rule 2: Two Parents, One Child, converging arrows (The Bagpipes Case)

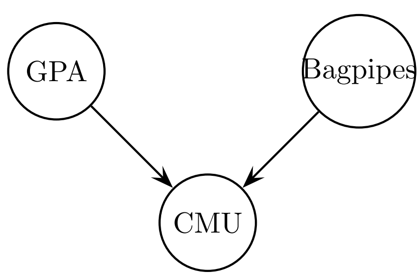


Figure 5: The Bagpipes Case.

Figure 5 is a simplified Bayes' net representation of the process of getting into CMU. Carnegie Mellon wants to admit students with a high GPA but it is also important to keep the school bagpipe band strong. A student's chances of getting into CMU can therefore be influenced by their GPA and also by their bagpipe playing skills.

Given any student applying to CMU knowing that they have good grades doesn't tell us anything about their bagpipe skills, the two are independent. This is changed if we then discover that the student was admitted. Now if we know they are good at the bagpipes our expectation of their grades is reduced as their admittance has been 'explained away'. The reverse is true if we know they have particularly high grades. Thus knowledge about admittance creates a dependence between the student's bag piping skills and their GPA.

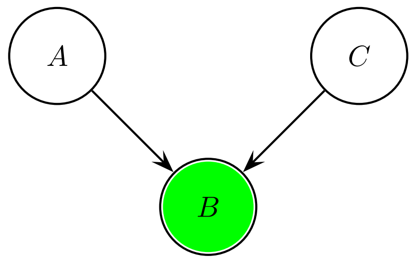


Figure 6: Two parents, one child case is **NOT BLOCKED** given B.

The rule is therefore that in a 'two parents, one child' case, as shown in Figure 6, A is dependent on C if B is known. The inverse is also true, A is independent from C if B is not known. This means that there is a blockage on a path passing through the 'two parent, one child' case if B is unknown.

2.1.3 Rule 2 Extension: Addition of Further Children

Rule 2 can be extended with the addition of descendents of the node. Figure 7 shows an example were an alarm can be set off by either an earthquake or a burglar and the police are called when the alarm goes off. As with the bagpipes example the presence of an earthquake and a burglar become dependent given the alarm going of. This is because if we know the alarm has been activated knowledge about a burglar reduces the liklihood of there having been an earthquake. Similarly, if we have no knowledge of the alarm going off, but there is a police visit, we can follow a chain of reasoning such as a police visit is more likely to have happened if an alarm went off, so the probability of the alarm going off increases, which then, coupled with knowledge about a burglar, reduces the probability of an earthquake.

The extension of rule 2 is therefore that if a descendant of the node into which the arrows converge is known the path is unblocked. With reference to Figure 8 the path is only blocked if B and any descendant of B are unknown.

2.1.4 Rule 3: One parent, Two Children

This rule is the case of some common cause for two events. The example in Figure 9 will help us understand the last rule. If we do not know the price of gas, information on the food price will

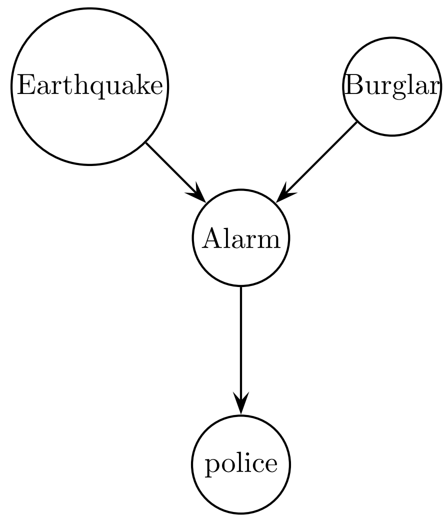


Figure 7: Home Alarm Example.

provide information on the likelihood of walking to work: If food price rises, it indicates that the gas price might have gone up, and then this can affect the likelihood of walking. If however, we know the price of gas, information about food price will not change our beliefs about the likelihood of walking.

Rule 3 is therefore that in a 'one parent, two children' case, as shown in Figure 10, A is independent from C given B. This means that there is a blockage on a path passing through this case if the parent, B, is known.

2.1.5 Example:

Some question we would like to know how to answer about a Bayes net, such as the one in Figure 1 are:

- Are two variables independent? $A \perp D$?
- Are two variables independent given some other variables? $A \perp D|C$?

1. $A \perp D$? No. The path $A \rightarrow c \rightarrow D$ is like a Markov Chain and we have not observed C.
2. $A \perp D|C$? Yes. Both paths are blocked.
3. $B \perp D|A$? No. The path $B \rightarrow C \rightarrow D$ is unblocked.

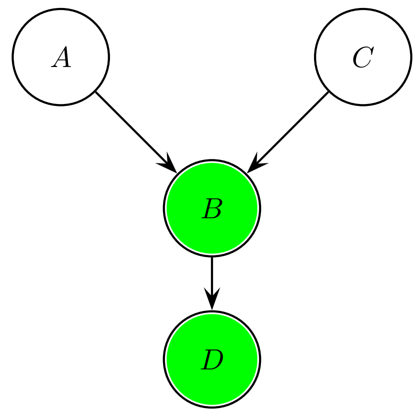


Figure 8: Path is **NOT BLOCKED** given either B or D.

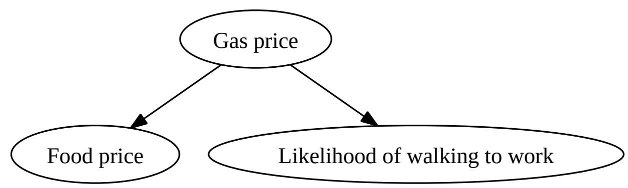


Figure 9: Rule 3, Gas price example.

2.1.6 Example: Localization

Now we will consider a set of Bayes nets for the navigation and localization of small car. Figure 11 (A) shows the remote controlled car scenario with a human driver sending inputs based on the car's actual state. The state is fully observable to the human so there is an edge connecting the state x_t to the control u_{t+1}. If the car is no longer remote controlled, but just follows a preprogrammed path, we get the Bayes net B. If the scenario is modified such that the car is not remote controlled, but is rather autonomous, we can reach the Bayes net (C). The robot cannot fully observe its state, it can only base its action on the sensor measurements.

For each of these models, we want to be able to model the state of the car using a recursive Bayes Filter. However, our original assumptions for the filter are not necessarily valid for these graphical models. We can test the assumptions using our new dependence rules. In the derivation of Bayes Filter in "Probabilistic Robotics" assumes that x_{t-1} is independent of u_t. When we look at the remote control example the two paths between x_1 and u_2 need to be tested. The path via x_2 is a case of rule 2 where both x_3 and z_3 are unknown and so is blocked, however the direct path cannot be blocked. Therefore there does exist a dependency and the assumption is incorrect.

On the other hand, with the preprogrammed path there is no edge between the control and prior state. This could be a preprogrammed path to move forward for 5 seconds then turn 90 degrees to the left and continue. The commands are independent of the position and would be the same regardless if the car hit a wall or continued without incident.

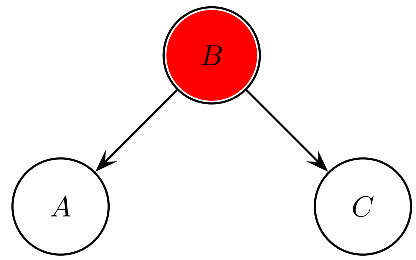


Figure 10: One parent, Two Children case is **BLOCKED** given B.

Finally, with the autonomous navigation, we get a more interesting case. In this case, we know x_2 and z_1. This means that the path via x_2 is blocked and the path via z_1 is a case of rule 1 where z_1 is known and therefore blocked. Since all paths are blocked and the assumption that x_{t-1} is independent from u_t is valid.

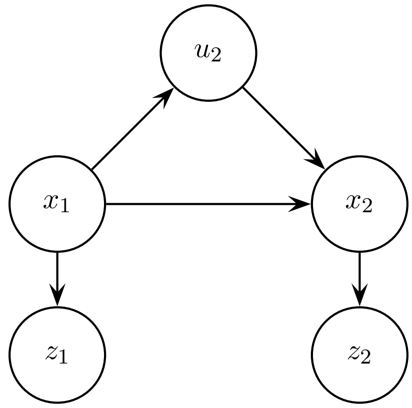
2.1.7 Example: Landmark Based Mapping

The Baysian network on Figure 12 displays a landmark based mapping scenario. Note that this network does not take into account that not seeing a landmark also provides information.

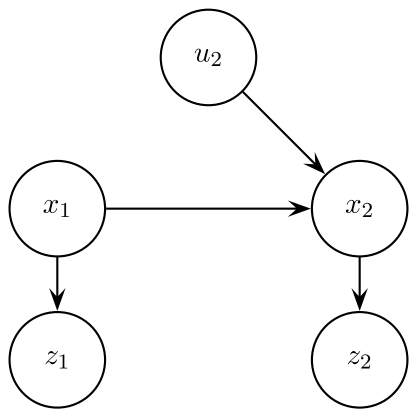
Some question we could ask about this network:

1. $l_1 \perp l_3$? Yes. Everything goes through z_1, use rule 2.
2. $l_1 \perp l_3 | z_1, \ldots, z_5$? No, but if we knew the states than yes.

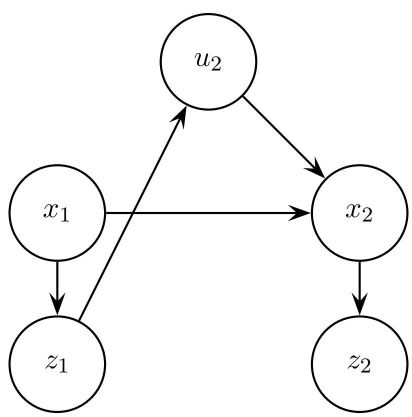
This means that if we know the states of the robot we can run a separate filter for each landmark, because they are independent. This is known as D-Separation.



(a) Remote Controlled



(b) Preprogrammed Path



(c) Autonomous Navigation

Figure 11: Three car navigation bayes nets. (A) The car is a remote control car. A human observes the full state of the car and directs it. (B) The car follows a preprogrammed path and does not use its sensor measurements to adjust its path. (C) The car is autonomous and uses its sensor measurements to guide its next control.

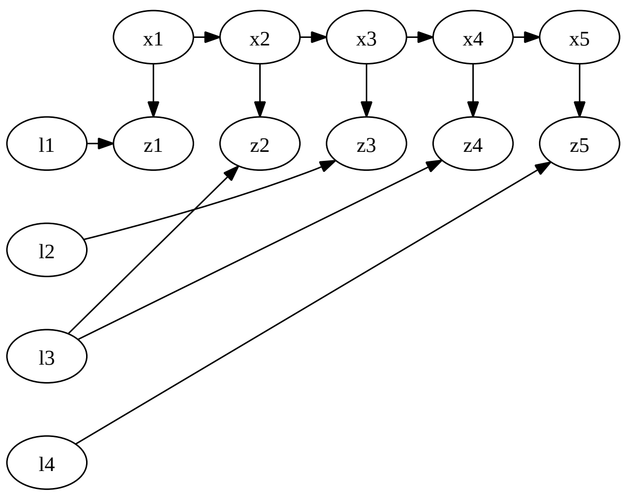


Figure 12: A Bayesian network representation of landmark based mapping.