Machine Learning (ML, F16) Lecture#15 (Tuesday, Nov. 1st)

Lecturer: Byron Boots *Graphical Models*

1 Graphical Models

Often, one is interested in representing a joint distribution P over a set of n random variables. Even in the simplest case, where the random variables are binary-valued, a joint distribution requires the specification of $2^n - 1$ numbers. This is completely unmanageable without further structural information about the distribution. 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 can be viewed as a:

- **Communication tool** that helps to *compactly* express 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 framework** that helps organize how we perform computations on random variables.

We will examine three types of graphical models:

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

Graphical models are a lot like a circuit diagram — they are written down to visualize and better understand a problem.

2 Bayesian Networks

One of the most common graphical models is called a Bayesian Network (Bayes' net). Bayes' nets are also known as 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).

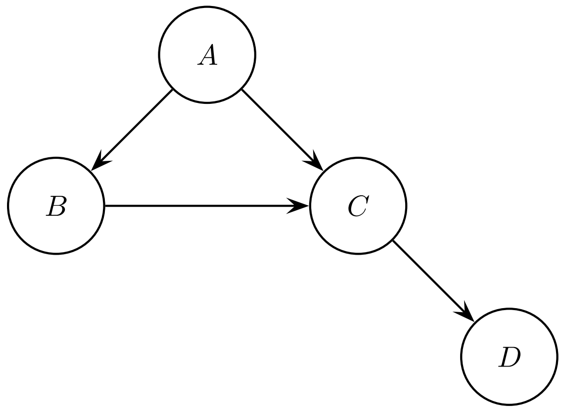

Figure 1: A Bayesian network.

In Figure 1, we indicate uncertain quantities A, B, C, and we draw directed arrows between them to represent relationships. A Bayesian network encodes a joint probability distribution over all the nodes in the graph. In this case, our Bayes' net encodes the joint probability distribution, $P(A, B, C, D)$.

The basic factorization of the probability distribution using the chain rule of probability is

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

This factorization always holds, and is not dependent on any particular graphical model.

In the network shown in Figure 1, we can use the edges in the graph to eliminate unnecessary conditional dependencies.

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

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 $\pi(x_i)$.

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

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

Bayes' net are often thought of as encoding causal relationships. However, these relationships are not necessarily causal. In our example, 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 edges is important in a Bayes net: *fewer* edges mean *more* structure.

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" or "d-separated" (directed separation). A path is any sequence of edge connected nodes leading from the first variable to the second. The Bayes' net in Figure 2 has two paths from A to E.

$$A \to B \to D \to E$$

$$A \to C \to D \to E$$

Blockages are determined by visiting each node on a path and comparing the structure of surrounding nodes and edges to the three rule situations explained below.

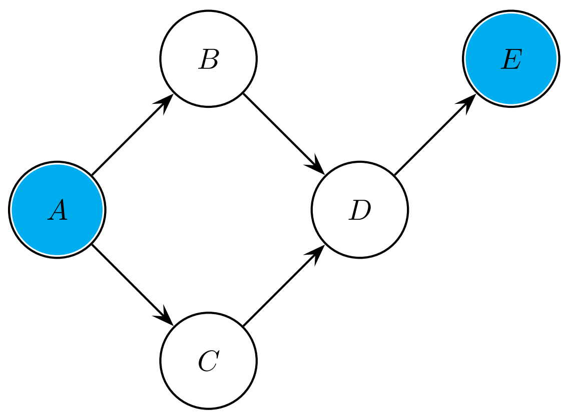

Figure 2: There are 2 paths from A to D.

2.1.1 Rule 1: Markov Chain

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

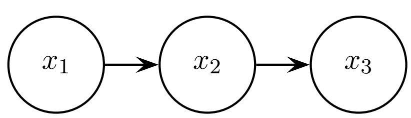

Figure 3: A Bayesian network representation of a Markov chain.

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. This is the same as saying that x_1 and x_3 are 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. This is the same as saying that x_1 and x_3 could be dependent if x_2 is not known.

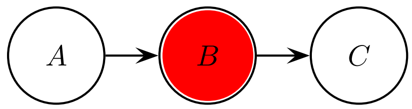

Figure 4: Markov chain is **BLOCKED** given B.

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

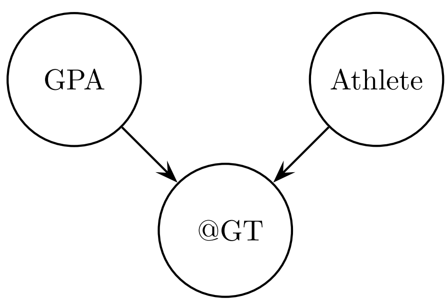

Figure 5: Admission to Georgia Tech

Figure 5 is a simplified Bayes' net representation of the process of getting into Georgia Tech. Georgia Tech wants to admit students with a high GPA but it is also wants good sports teams. A student's chances of getting into GT can therefore be influenced by their GPA and also by their athletic ability.

Given any student applying to GT knowing that they have good grades doesn't tell us anything about their athletic ability, the two are independent. This is changed if we then discover that the student was admitted. Now if we know they were a first team All-American football player 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 athletic ability and their GPA.

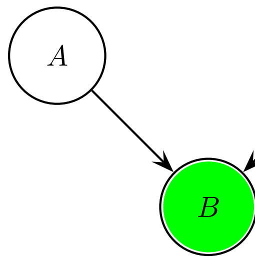

Figure 6: Two parents, one child case is **NOT BLOCKED** given B.

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

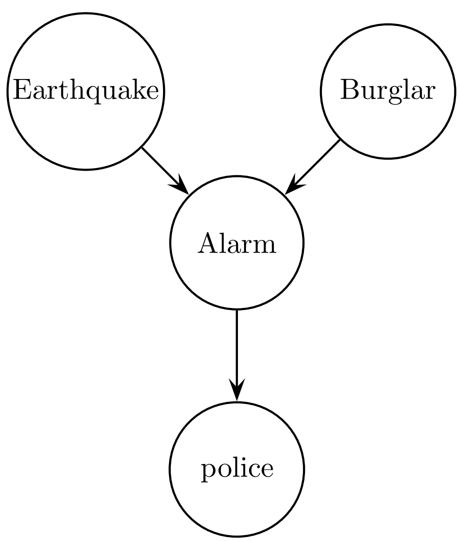

Figure 7: Home Alarm Example.

Rule 2 can be extended with the addition of children of the child. 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 admission 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 likelihood of there having been an earthquake. If however only the presence of the police is known the same dependency is formed as the police imply that the alarm has been activated.

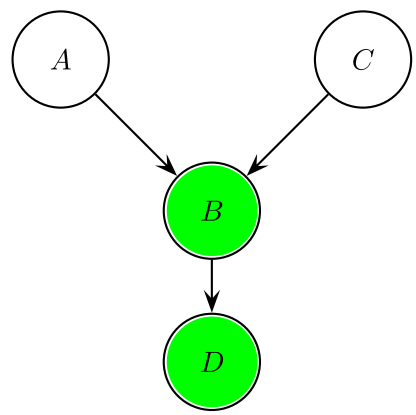

Figure 8: Path is **NOT BLOCKED** given either B or D.

The extension of rule 2 is therefore that if a descendent of the child is known the path is also unblocked. With reference to Figure 8 the path is only blocked if B **and all the descendants** of B are unknown.

2.1.4 Rule 3: One parent, Two Children

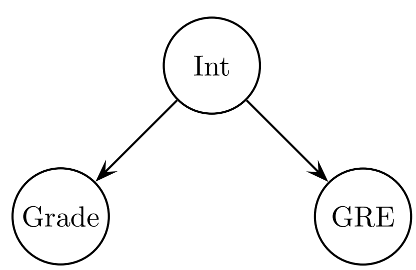

Figure 9: Rule 3, Grad-Student Example.

Figure 9 shows a Bayes' net representation of the use of graduate student performance. In this (highly fictional) example, GRE performance and course grade depend on intelligence. If a student scores highly on the GRE, then the probability of doing well in class in increased, and vice versa. If, however, we know that the student has high intelligence, knowing how the student performed on the GRE does not change the likelihood the student performing well in class.

Rule 3 is therefore that in a "one parent, two children" case, as shown in Figure 10, A is independent

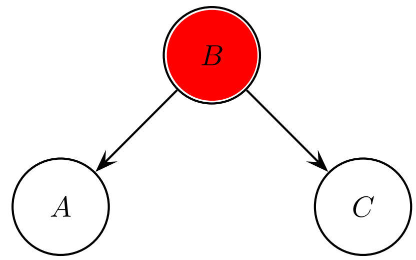

Figure 10: One parent, two Children case is **BLOCKED** given B.

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: Open-loop control

Figure 11 shows a typical dynamical system with controls u, observations y, and states x. Initially, if we haven't observed anything, x_1 is independent of u_2 and u_3, but not anything else. x_3, x_2, z_2, and z_3 are dependent of u_2. z_2 depends on everything except u_3.

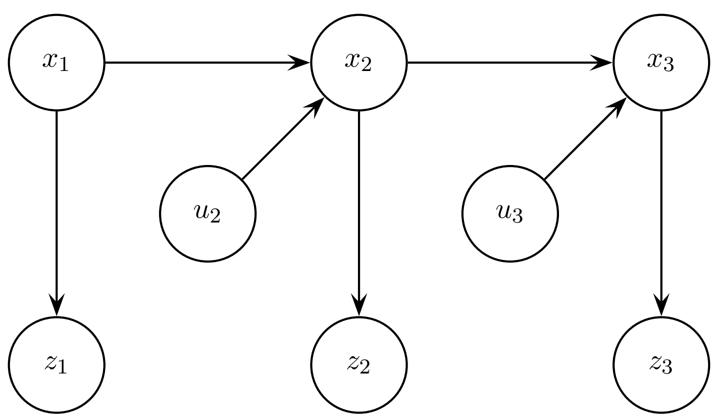

Figure 11: A dynamical system with open-loop control.

Now, if x_2 is observed (Figure 12), u_2 and x_1 are dependent, but x_1, x_3, and z_2 are independent. x_3 and z_2 are also independent of u_2.

2.1.6 Example: Closed-loop control

Figure 13 shows a simple remote controlled car scenario with a human driver sending inputs based on the car's actual state. The derivation of Bayes Filter in "Probabilistic Robotics" assumes that x_{t-1} is independent of u_t. To test that this is the case for the remote control example the two paths

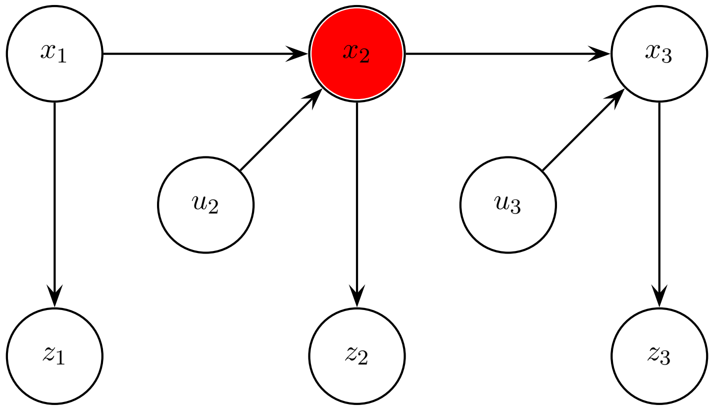

Figure 12: A dynamical system with open-loop control.

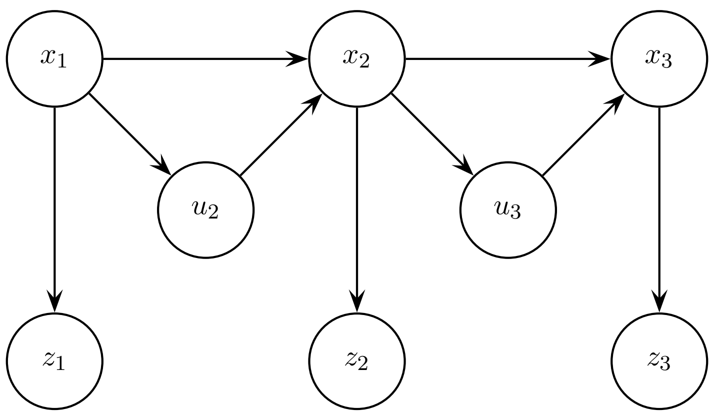

Figure 13: A dynamical system with closed-loop control.

between x_1 and u_2 need to be tested. The path via x_2 is a case of rule 2 where both x_2 and z_2 are unknown and so is blocked, however the direct path cannot be blocked.

2.1.7 Example: Controls based on Observations

If the scenario is modified such that the input is based on the previous observation and not a human who knows the actual state, the Bayes' net looks like Figure 14. In this case the path via x_2 is still blocked and the path via z_1 is a case of rule 1 where if z_1 is known the path to u_2 is blocked. If z_1 is observed, the assumption that x_{t-1} is independent from u_t is valid.

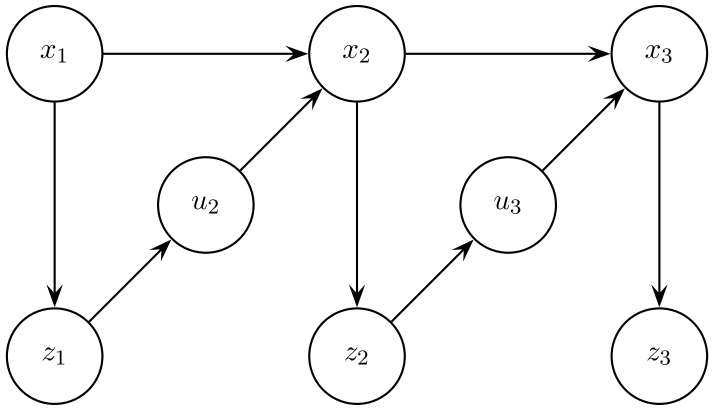

Figure 14: A dynamical system with controls based on observations.

2.1.8 Example: Landmark Based Navigation

Figure 15 shows a Bayesian network representation of a localization scenario with internal states x_i, observations z_i and landmarks l_i. If we see all the observations, $z_0, z_1, z_2, ...$, are the landmarks conditionally independent of each other? For example, is $l_1 \perp l_2 | Z$?

The converging arrows at z_0 are an example of rule 2 and the path remains unblocked as z_0 is known. Looking now at x_0, rule 3 can be used to show there is no blockage as x_0 is not known. The path extends through x_1 to z_1, a rule 1 case where x_1 is not known leaving the path unblocked. Finally z_1 to l_2 is another unblocked rule 2 case. Thus, l_1 and l_2 are not conditionally independent given z_0 and z_1.

The existence of conditional dependencies between landmarks introduces significant computational complexity due to high dimensionality. In this case if the values of X can be observed, the dependency between l_1 and l_2, and all the landmarks, is removed. Using a particle filter, samples of X can be taken making each landmark independent and allowing for seperate filters to be run for each landmark, greatly reducing the dimensionality of the problem.

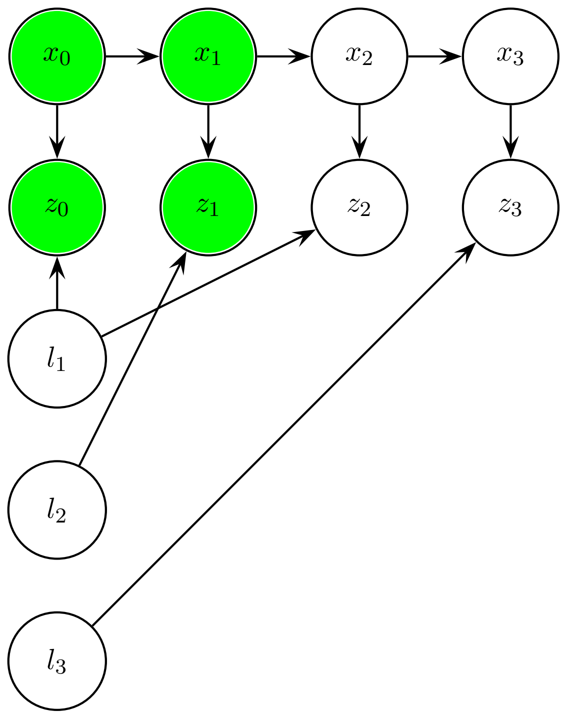

Figure 15: A Bayesian network representation of a localization scenario.