STA 414/2104:
Statistical Methods in Machine Learning II

Week 2 : Decision Theory & Graphical Models

Murat A. Erdogdu and Piotr Zwiernik

University of Toronto

Overview

- Today:
- Statistical decision theory
- Graphical Models

Decision making

We develop a small amount of theory that provides a framework for understanding many of the models we consider.

- Suppose we have a real-valued input vector x and a corresponding target (output) value c with joint probability distribution: $p(x, c)$.
- Our goal is to predict the output label c given a new value for x.
- For now, we focus on classification so c is a categorical variable, but the same reasoning applies to regression (continuous target).

The joint probability distribution $p(x, c)$ provides a complete summary of uncertainties associated with these random variables.

Inference

Estimating $p(x, c)$ from training data is an example of **inference**.

Example: Cancer screening from chest X-ray

Based on the X-ray image, we would like determine whether the patient has cancer or not.

- The input vector x is pixel intensities, and the output c represents the presence of cancer, class $\mathcal{C}_1$, or absence of cancer, class $\mathcal{C}_2$.

- $\mathcal{C}_1$ cancer present
- $\mathcal{C}_2$ cancer absent

We can use an "arbitrary" encoding for these classes $\mathcal{C}_1$ and $\mathcal{C}_2$, e.g. choose c to be binary: $c = 0$ correspond to class $\mathcal{C}_1$, and $c = 1$ corresponds to $\mathcal{C}_2$.

Inference

Inference Problem

Let's assume we estimated the joint distribution $p(x, c)$ using some ML method. In the end, we must make a decision of whether to give treatment to the patient or not.

- Given a new X-ray image, our goal is to decide which of the two classes that image should be assigned to. We could compute conditional probabilities of the two classes, given the input image:

$$p(\mathcal{C}_k|x) = \frac{p(x|\mathcal{C}_k)p(\mathcal{C}_k)}{p(x)} \quad \text{Bayes' rule.}$$

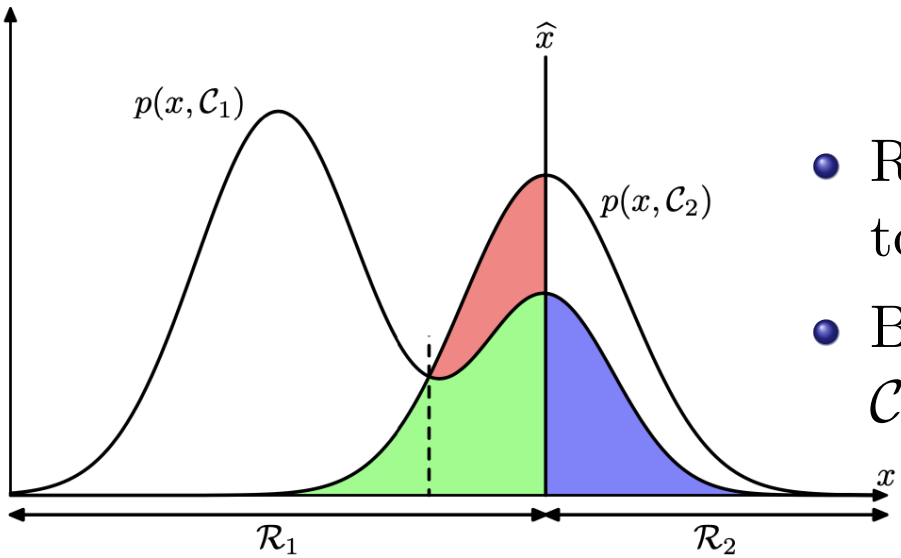
- If we minimize the expected number of mistakes, we can minimize the probability of assigning x to the wrong class.
  This suggests we minimize the **misclassification rate**.

Misclassification rate

Goal

Make as few misclassifications as possible. We need a rule that assigns each value of x to one of the available classes.

Divide the input space into regions $\mathcal{R}_k$ (decision regions) such that all points in $\mathcal{R}_k$ are assigned to class $\mathcal{C}_k$.

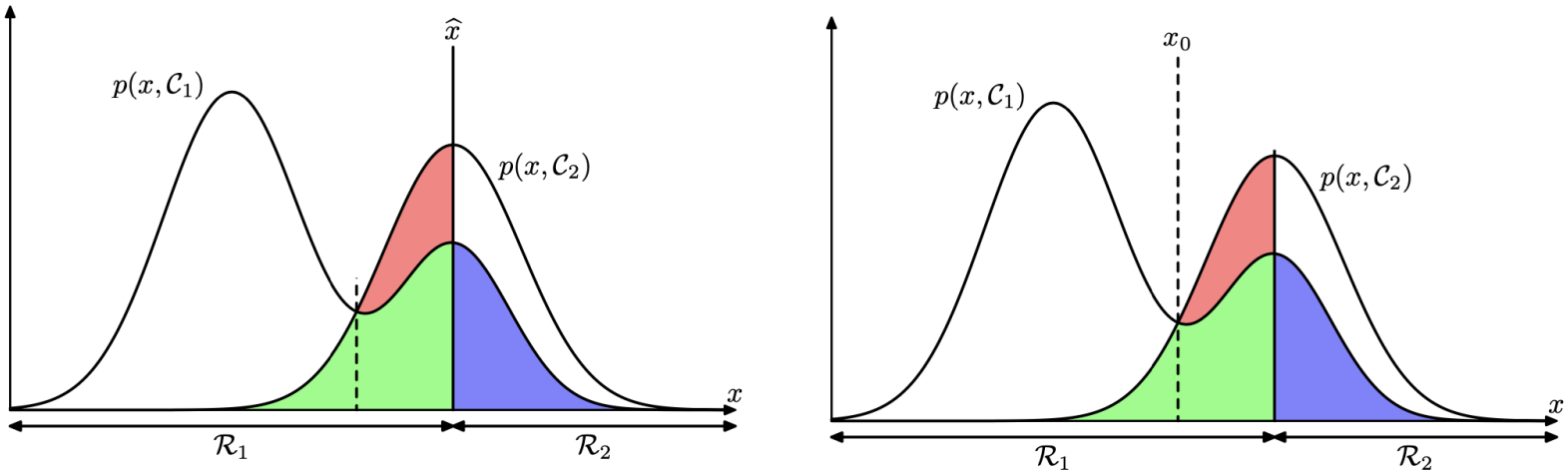


- Red + green regions: input belongs to class $\mathcal{C}_2$, but is assigned to $\mathcal{C}_1$.
- Blue region: input belongs to class $\mathcal{C}_1$, but is assigned to $\mathcal{C}_2$.

$$
\begin{aligned}
p(\text{mistake}) &= p(x \in \mathcal{R}_1, \mathcal{C}_2) + p(x \in \mathcal{R}_2, \mathcal{C}_1) \\
&= \int_{\mathcal{R}_1} p(x, \mathcal{C}_2) dx + \int_{\mathcal{R}_2} p(x, \mathcal{C}_1) dx
\end{aligned}
$$

Misclassification rate

Compare the following two decision rules:



- Blue + green area is always included in the p(mistake).
- Therefore, we aim to reduce the red area by moving the threshold $\hat{x}$ to x_0, which turns out to be optimal in this case.

Misclassification error

- Misclassification error:

$$p(\text{mistake}) = \underbrace{\int_{\mathcal{R}_1} p(x, \mathcal{C}_2) dx}_{\text{red+green}} + \underbrace{\int_{\mathcal{R}_2} p(x, \mathcal{C}_1) dx}_{\text{blue}}$$

  and the decision regions $\mathcal{R}_1$ and $\mathcal{R}_2$ are disjoint.
- Therefore, for a particular input x, if $p(x, \mathcal{C}_1) > p(x, \mathcal{C}_2)$, then we assign x to class $\mathcal{C}_1$. I.e. $\mathcal{R}_1 = \{x : p(x, \mathcal{C}_1) > p(x, \mathcal{C}_2)\}$.

Minimizing misclassification

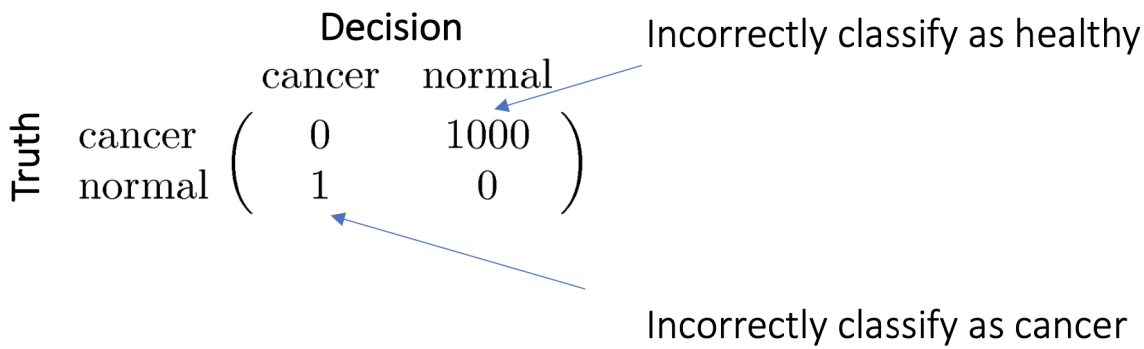
Since $p(x, \mathcal{C}_k) = p(\mathcal{C}_k|x)p(x)$, in order to minimize the probability of making mistake, we assign each x to the class for which the posterior probability $p(\mathcal{C}_k|x)$ is largest. This minimizes the misclassification rate.

Expected loss

How realistic is it to minimize the misclassification rate?

- We want a **loss function** to measure the loss incurred by taking any of the available decisions.
- Suppose that for x, the true class is $\mathcal{C}_k$, but we assign x to class $\mathcal{C}_j$ and incur loss of L_{kj} ((k, j)-th element of a loss matrix).

Consider medical diagnosis example: example of a loss matrix:



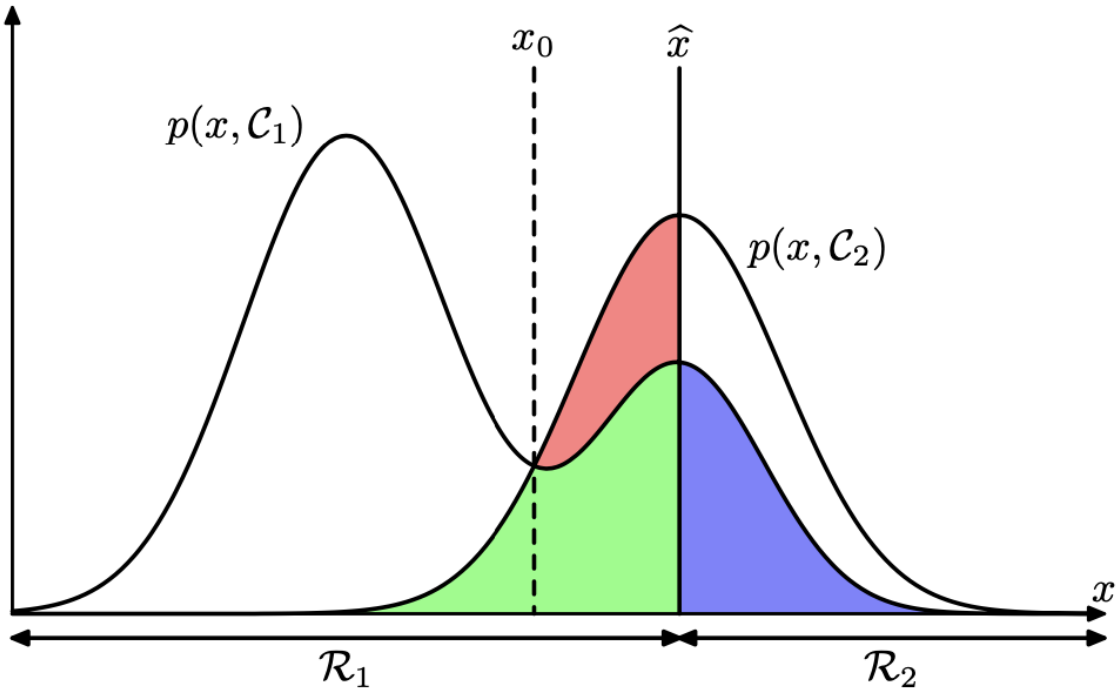
Thus the expected loss is given by

$$\mathbb{E}[L] = \sum_k \sum_j \int_{\mathcal{R}_j} L_{kj}\ p(x, \mathcal{C}_k) dx$$

New goal: Minimize expected loss

New objective:

Choose regions $\mathcal{R}_j$ as to minimize expected loss.



In the above figure, the blue region corresponds to L_{12}: the sample comes from class $\mathcal{C}_1$ but we classified as $\mathcal{C}_2$.

Minimize expected loss

Therefore, we want to minimize

$$\begin{aligned}\mathbb{E}[L] &= \sum_k \sum_j \int_{\mathcal{R}_j} L_{kj}\ p(x, \mathcal{C}_k) dx \\ &= \sum_j \int_{\mathcal{R}_j} \sum_k L_{kj}\ p(x, \mathcal{C}_k) dx.\end{aligned}$$

Define $g_j(x) = \sum_k L_{kj}\ p(x, \mathcal{C}_k)$ and notice that $g_j(x) \geq 0$. Then, the expected loss is equal to

$$\mathbb{E}[L] = \sum_j \int_{\mathcal{R}_j} g_j(x) dx$$

Thus, minimizing $\mathbb{E}[L]$ is equivalent to choosing

$$\mathcal{R}_j = \{x \ : \ g_j(x) < g_i(x) \ \text{ for all } \ i \neq j\}.$$

Simplifying further

We can also use the product rule $p(x, \mathcal{C}_1) = p(\mathcal{C}_1|x)p(x)$ and reduce the problem to:

Discriminant rules:

Find regions $\mathcal{R}_j$ such that the following is minimized:

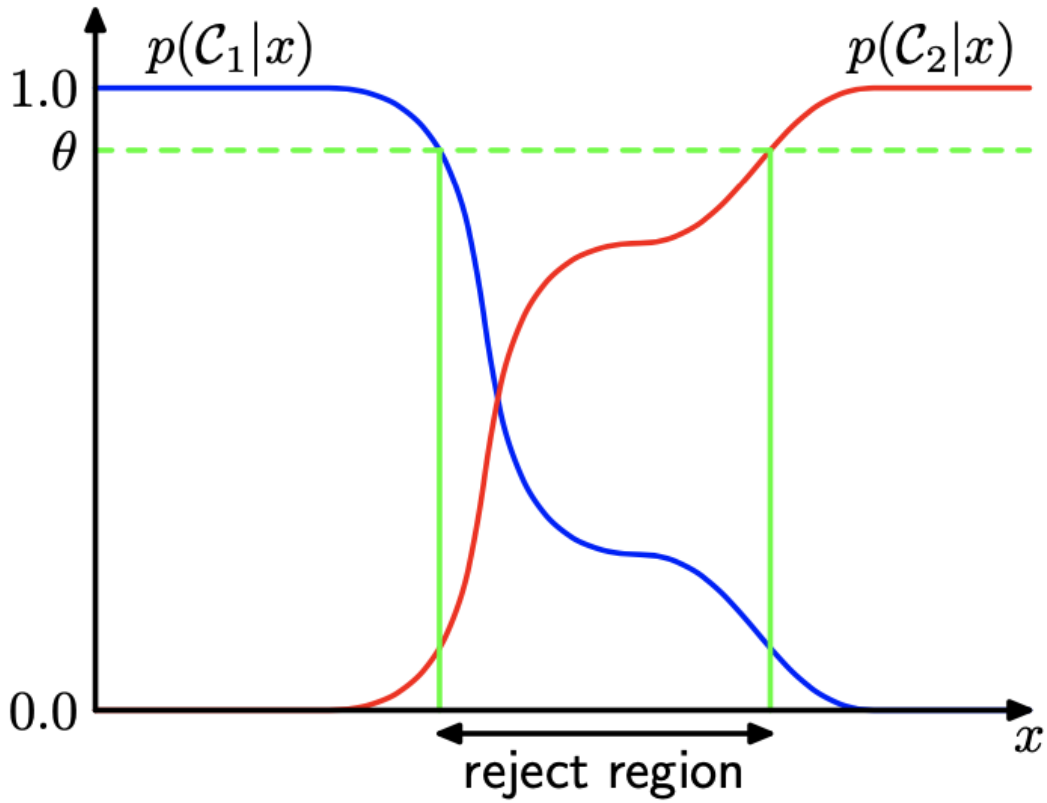
$$\sum_k L_{kj}\ p(\mathcal{C}_k|x).$$

That is

$$\mathcal{R}_j = \Big\{x\ : \sum_k L_{kj}\ p(\mathcal{C}_k|x) < \sum_k L_{ki}\ p(\mathcal{C}_k|x)\ \text{ for all }\ i \neq j\Big\}.$$

Reject option

For the regions where we are relatively uncertain about class membership, we don't have to make a decision.



Here, notice that we have a threshold θ and the conditional class probabilities fall below this threshold, we refuse to make a decision.

Loss functions for regression

- Now we consider an input/target setup (x, t) where the target (output) is continuous $t \in \mathbb{R}$, and the joint density is $p(x, t)$.
- Instead of decision regions, we aim to find a regression function $y(x) \approx t$ which maps inputs to the outputs.
- Consider the squared loss function L between $y(x)$ and t to assess the quality of our estimate $L(y(x), t) = (y(x) - t)^2$.

Goal:

What is the best function $y(x)$ that minimizes the expected loss?

$$\mathbb{E}[L] = \int \int L(y(x), t) p(x, t) dx dt.$$

Minimizing expected loss: Best regression function

We add and subtract $\mathbb{E}[t|x]$ and write

$$\begin{aligned}
\mathbb{E}[L] &= \int\int (y(x)-t)^2 p(x,t)dxdt \\
&= \int\int (y(x)-\mathbb{E}[t|x]+\mathbb{E}[t|x]-t)^2 p(x,t)dxdt \\
&= \int\int (y(x)-\mathbb{E}[t|x])^2 p(x,t)dxdt + \int\int (\mathbb{E}[t|x]-t)^2 p(x,t)dxdt \\
&\quad + 2\int\int (y(x)-\mathbb{E}[t|x])(\mathbb{E}[t|x]-t)p(x,t)dxdt
\end{aligned}$$

The last term is zero since

$$\begin{aligned}
&\int\int (y(x)-\mathbb{E}[t|x])(\mathbb{E}[t|x]-t)p(x,t)dxdt \\
&= \int\int (y(x)-\mathbb{E}[t|x])(\mathbb{E}[t|x]-t)p(t|x)p(x)dxdt \\
&= \int (y(x)-\mathbb{E}[t|x])\Big\{\underbrace{\int (\mathbb{E}[t|x]-t)p(t|x)dt}_{=0}\Big\}p(x)dx = 0
\end{aligned}$$

Best regression function

- We showed that the expected loss is given by the sum of two **non-negative** terms

$$\mathbb{E}[L] = \int\int (y(x) - \mathbb{E}[t|x])^2 p(x,t) dx dt + \int\int (\mathbb{E}[t|x] - t)^2 p(x,t) dx dt.$$

- The second term does not depend on $y(x)$ thus choosing the best regression function $y(x)$ is equivalent to minimizing the first term on the right hand side.
- Since that term is always non-negative, we can make it zero by choosing

$$y(x) = \mathbb{E}[t|x].$$

- The second term is the expectation of the conditional variance of $t|x$. It represents the intrinsic variability of the target data and can be regarded as noise.

Summary: Decision making

- Depending on the application, one needs to choose an appropriate loss function.
- Loss function can significantly change the optimal decision rule.
- One can always use the reject option and not make a decision.
- In case of regression, one can find the optimal map between x and t if one knows the conditional distribution $t|x$. The optimal map corresponds to the conditional expectation $\mathbb{E}[t|x]$.

Next:

- Graphical models notation
- Conditional independence
- Bayes Ball

Joint distributions

- The joint distribution of N random variables $(x_1, x_2, ..., x_N)$ is a very general way to encode knowledge about a system.
- Assume $x_i \in \{0, 1\}$ are binary, then it requires $2^N - 1$ parameters to specify the joint distribution

$$p(x_1, x_2, ..., x_N).$$

- This can be also written as

$$p(x_1, x_2, \ldots, x_N) = \prod_{j=1}^{N} p(x_j | x_1, x_2, \ldots, x_{j-1})$$

  for any ordering of the variables, where $p(x_1|x_0) = p(x_1)$.
- We can exploit dependencies among variables and reduce the number of parameters! (e.g. Naive Bayes)

Conditional Independence

- Assume there are N random variables $x_1, x_2, ..., x_N$.
- For set $A \subset \{1, 2, ..., N\}$, we denote by $x_A = \{x_i \ : \ i \in A\}$. Assume A, B, C are disjoint. In particular, we say that

$$x_A \perp x_B \mid x_C$$

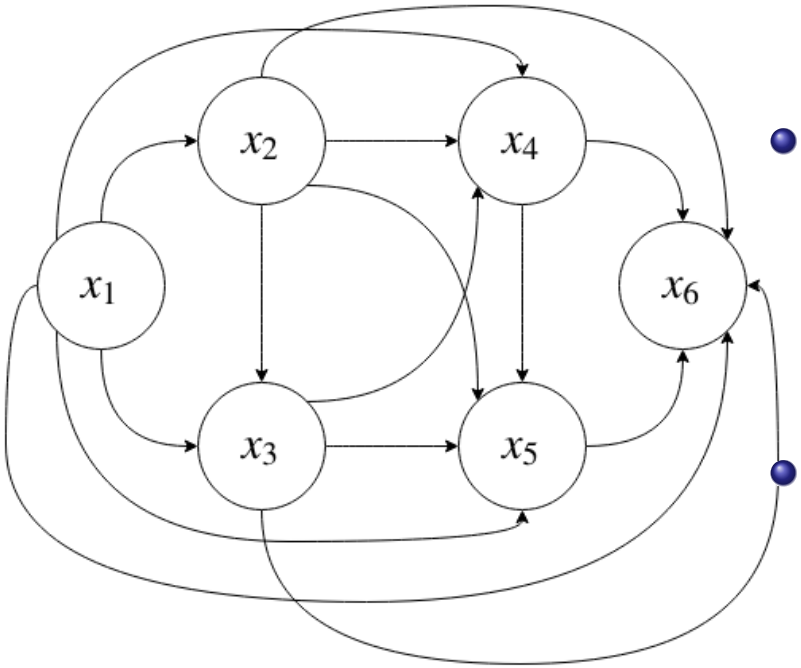
  if random variables x_A, x_B are conditionally independent given x_C.
- We have

$$x_A \perp x_B | x_C$$

  iff
  - $\Leftrightarrow p(x_A, x_B|x_C) = p(x_A|x_C)p(x_B|x_C)$
  - $\Leftrightarrow p(x_A|x_B, x_C) = p(x_A|x_C)$
  - $\Leftrightarrow p(x_B|x_A, x_C) = p(x_B|x_C)$

These are all equivalent!

Directed Acyclic Graphical Models (Bayes' Nets)



- A directed acyclic graphical model (DAG) implies a factorization of the joint distribution.
- Variables are represented by nodes, and edges represent dependence.

DAG induces the following factorization of the joint distribution of random variables $x_1, x_2, \ldots, x_N$, we can write:

$$p(x_1, \ldots, x_N) = \prod_{i=1}^{N} p(x_i | x_1, \ldots, x_{i-1}) = \prod_{i=1}^{N} p(x_i | \text{parents}(x_i))$$

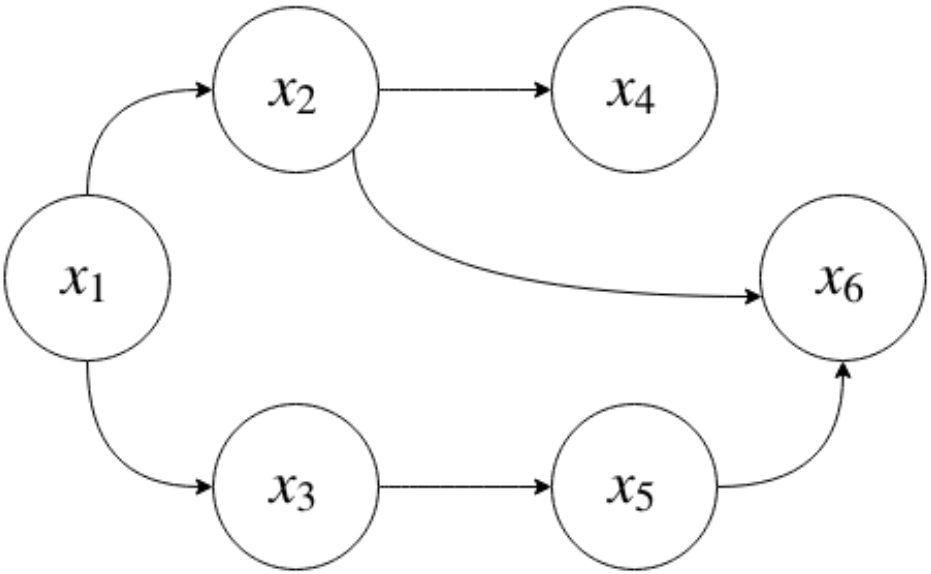
where $\text{parents}(x_i)$ is the set of nodes with edges pointing to x_i.

DAGs and Conditional Independence

In a directed acyclic graphical model (DAGs)

$$p(x_1, x_2, \ldots, x_N) = \prod_{i=1}^{N} p(x_i|\text{parents}(x_i))$$

where parents(x_i) is the set of nodes with edges pointing to x_i.

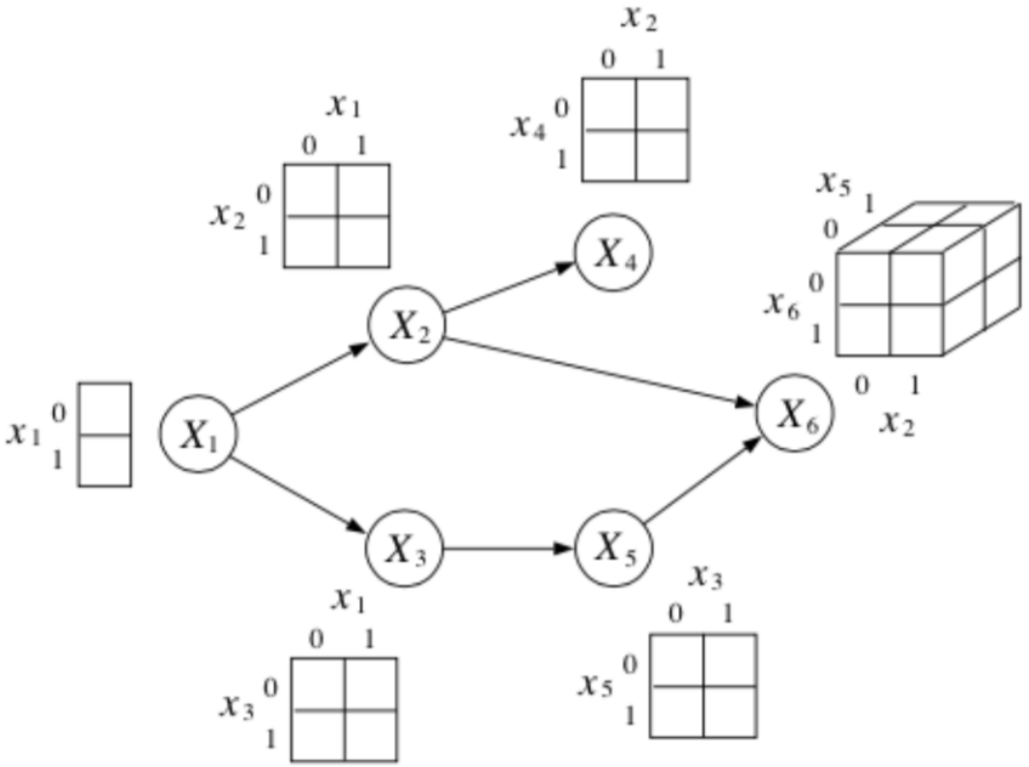


- This DAG corresponds to the following factorization of the joint distribution:

$$p(x_1, x_2, ..., x_6) = p(x_1)p(x_2|x_1)p(x_3|x_1)p(x_4|x_2)p(x_5|x_3)p(x_6|x_2, x_5)$$

Conditional Probability Tables (CPT)

Suppose each x_i is a binary random variable. How many parameters does it take to represent this joint distribution?



- For example, 2x2 CPT for the node x_4 corresponds to $p(x_4|x_2)$ requires 2 parameters.
- Each CPT with K_i parents requires 2^{K_i} parameters. In total, $\sim N2^{\max K_i}$ parameters.
- If we allow all possible dependencies (fully-connected DAG), which requires $2^N - 1$ parameters.

DAGs reduce the computational burden of making inferences by introducing conditional independencies.

Conditional Independence in DAGs

- **D-separation** (directed-separation) is a notion of connectedness in DAGs in which two (sets of) variables may or may not be connected conditioned on a third (set of) variable(s).
- D-separation implies conditional independence and vice versa.
- For a set $A \subset \{1, 2, ..., N\}$, we denote by $x_A = \{x_i \ : \ i \in A\}$. In particular, we say that

$$x_A \perp x_B \mid x_C$$

if every variable in A is d-separated from every variable in B conditioned on all the variables in C.

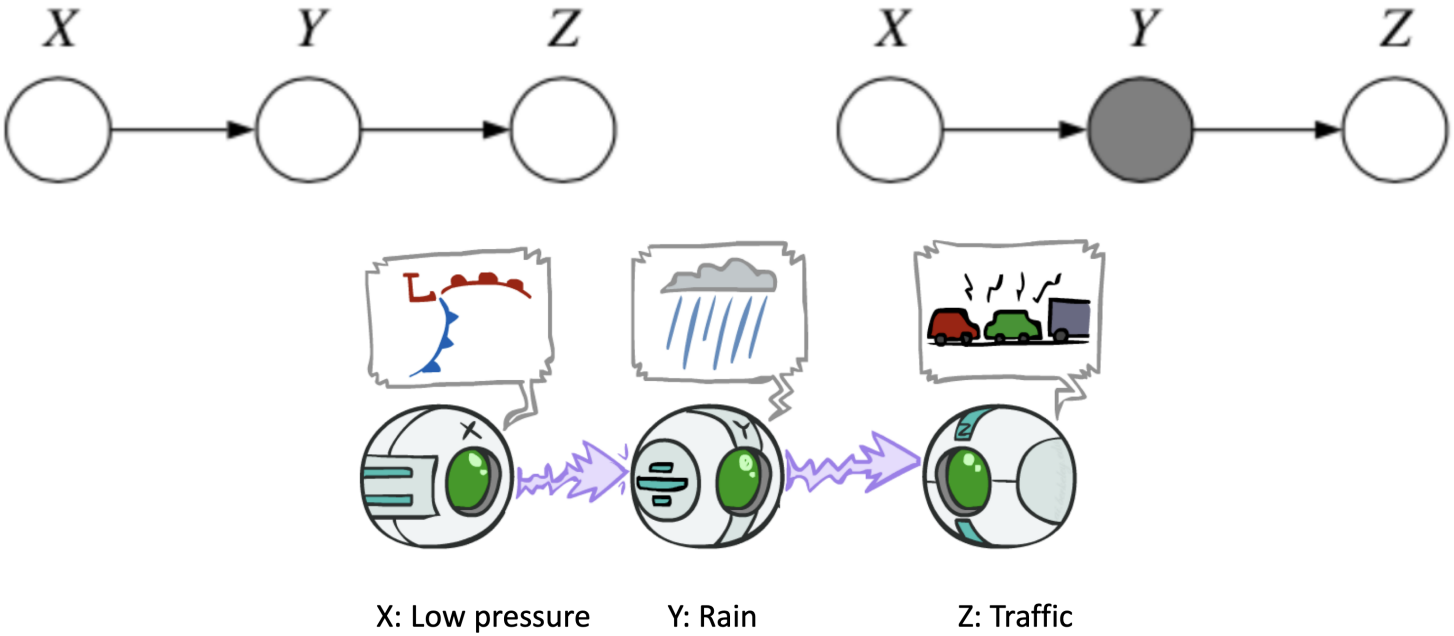
DFS Algorithm for Checking Independence

Let A, B, C be disjoint subsets of $\{1, 2, ..., N\}$.

- We cycle through each node in A, do a depth-first search to reach every node in B, and examine the path between them.
- If all of the paths have d-separated end points (i.e., conditionally independent nodes), then

$$x_A \perp x_B \mid x_C$$

Causal Chain

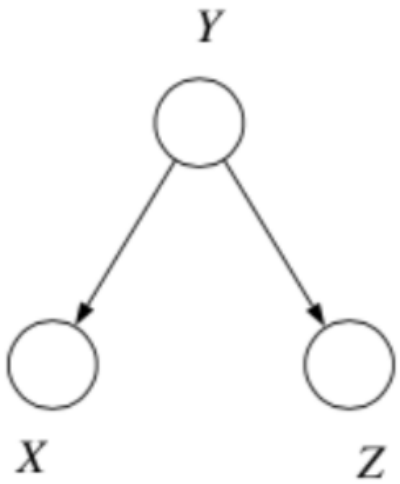


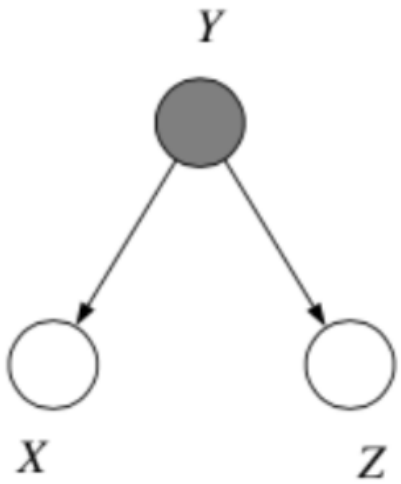
$$
\begin{aligned}
p(z|x,y) &= \frac{p(x,y,z)}{p(x,y)} \\
&= \frac{p(x)p(y|x)p(z|y)}{p(x)p(y|x)} \\
&= p(z|y) \quad X \text{ and } Z \text{ d-separated given } Y.
\end{aligned}
$$

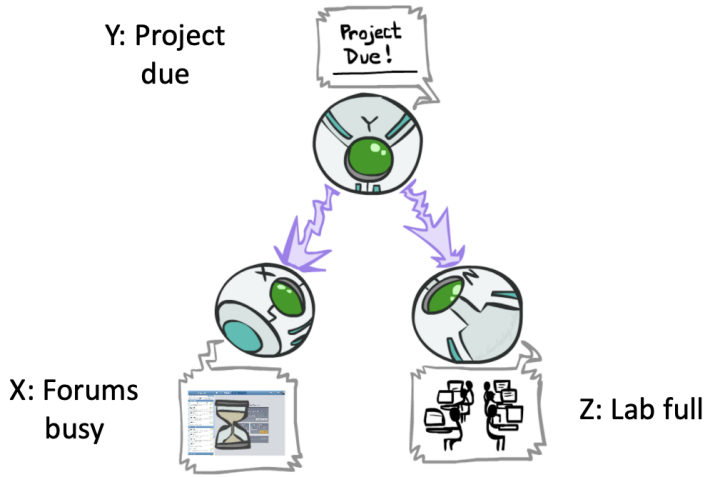
image credit Abbeel & Klein

Common Cause

Where we think of y as the "common cause" of the two independent effects x and z.





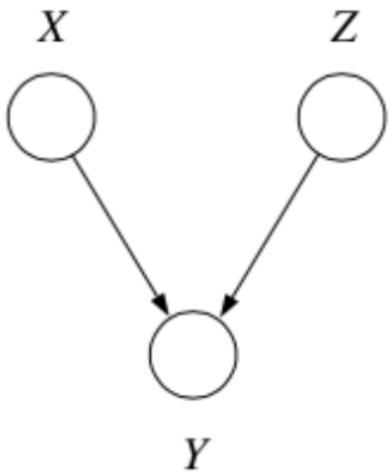


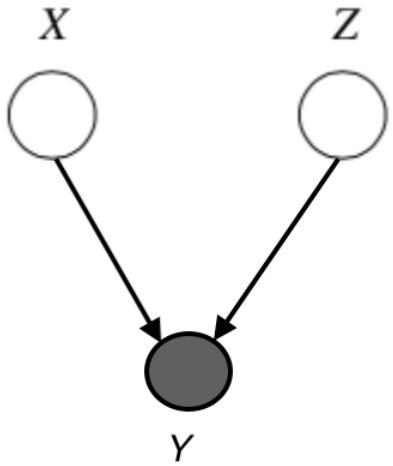
Question: When we condition on y, are x and z independent?
Answer: From the graph, we get

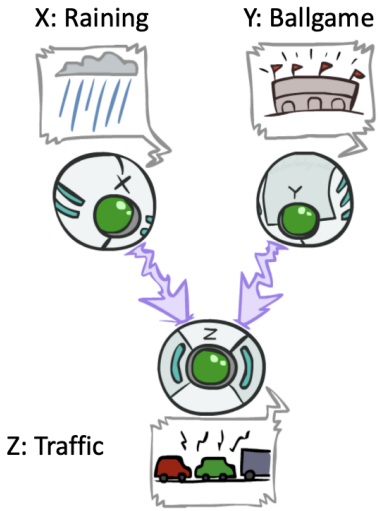
$$p(x,z|y) = \frac{p(x,y,z)}{p(y)} = \frac{p(y)p(x|y)p(z|y)}{p(y)} = p(x|y)p(z|y) \quad \text{yes!}$$

image credit Abbeel & Klein

Explaining Away (Common Effect)







Question: When we condition on y, are x and z independent?
Answer: From the graph, we get

$$p(z|x,y) = \frac{p(x)p(z)p(y|x,z)}{p(x)p(y|x)}$$
$$= \frac{p(z)p(y|x,z)}{p(y|x)} \neq p(z|y)$$

image credit Abbeel & Klein

Bayes Ball Algorithm

An algorithm for determining conditional independence in a DAG is the Bayes Ball algorithm.

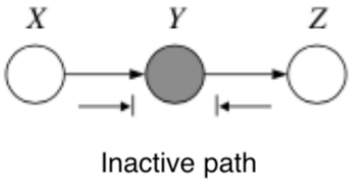
- To check if $x_A \perp x_B | x_C$ we need to check if every variable in A is d-seperated from every variable in B conditioned on all variables in C.
- In other words, given that all the nodes in x_C are "clamped", when we "wiggle" nodes x_A can we change any of the nodes in x_B?

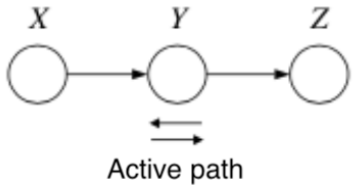
Bayes Ball: Rules

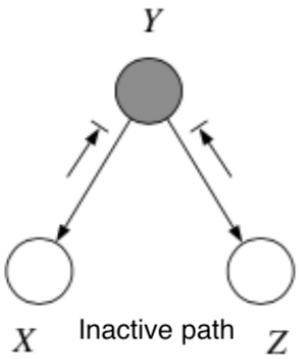
In general, the algorithm works as follows:

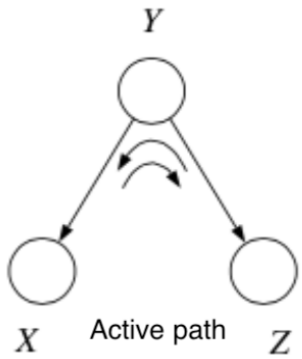
1. Shade all nodes x_C (these are observed)
2. Place "balls" at each node in x_A (or x_B)
3. Let the "balls" "bounce" around according to some rules
   - - If any of the balls reach any of the nodes in x_B from x_A (or x_A from x_B) then $x_A \not\perp x_B | x_C$
   - - Otherwise $x_A \perp x_B | x_C$

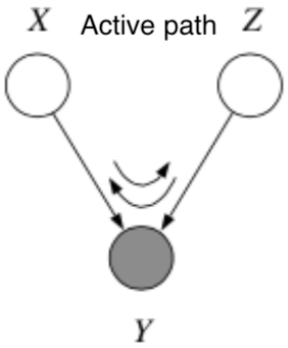
Bayes Ball: Rules for active/inactive triples

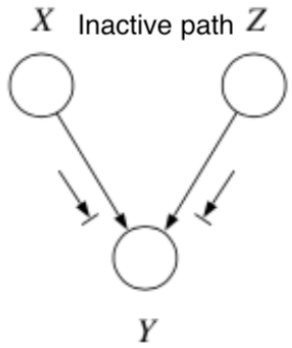




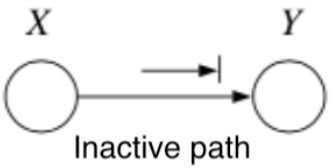


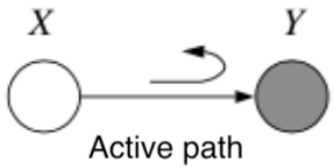


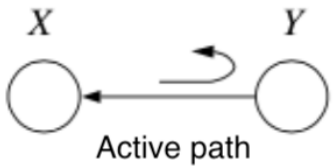


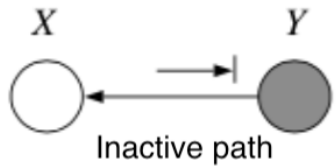


- Arrows: paths the balls can travel
- Arrows with bars: paths the balls cannot travel
- Notice balls can travel opposite to edge directions!
- Pairs are boundary cases.



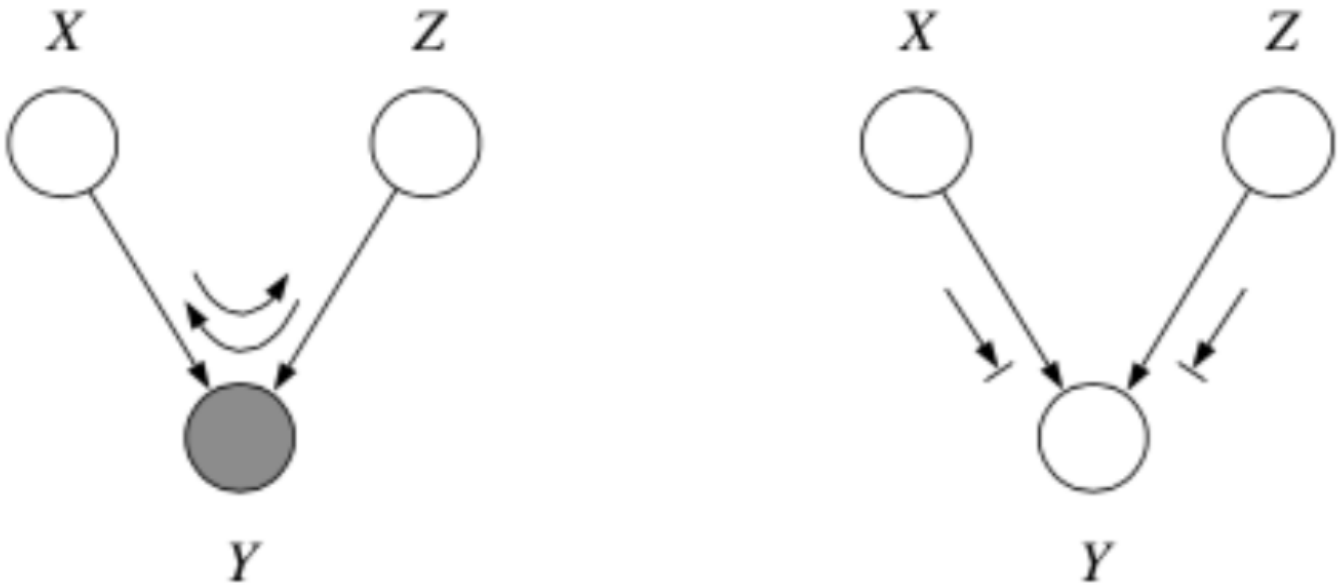






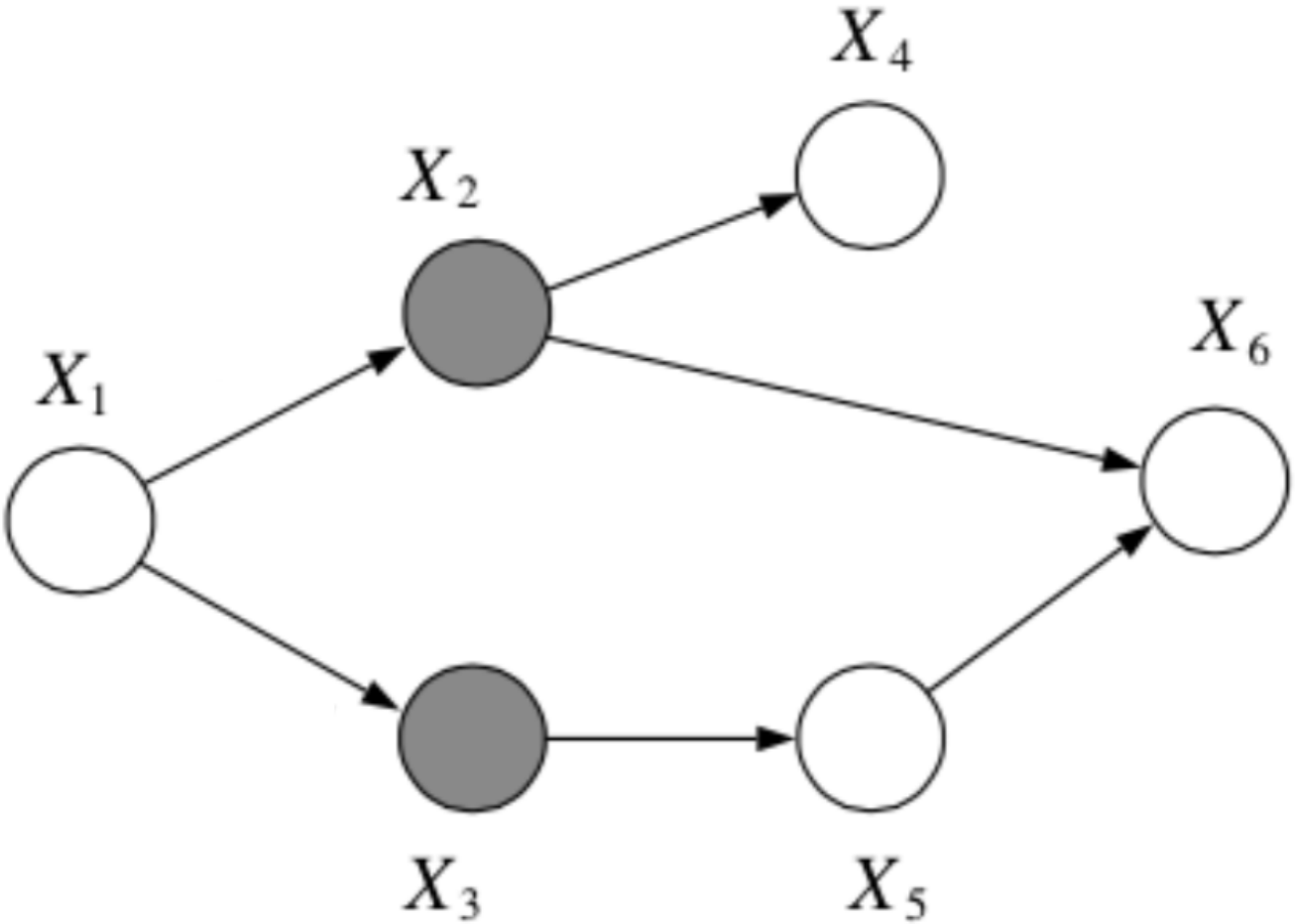
Example I: Explaining Away

If y or any of its descendants is shaded, the ball passes through.



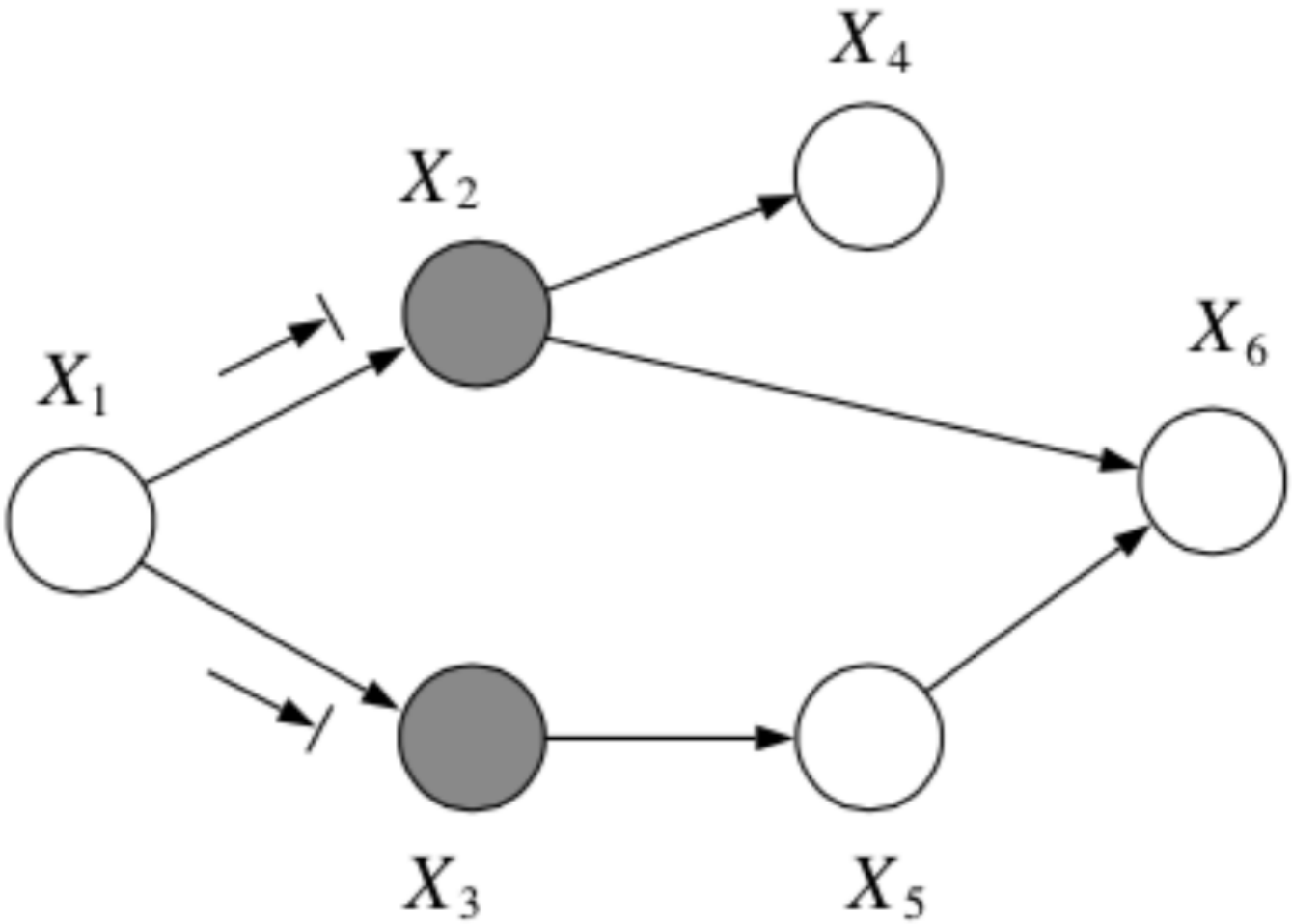
Example II: Large DAG

In the following graph, is $x_1 \perp x_6 | \{x_2, x_3\}$?



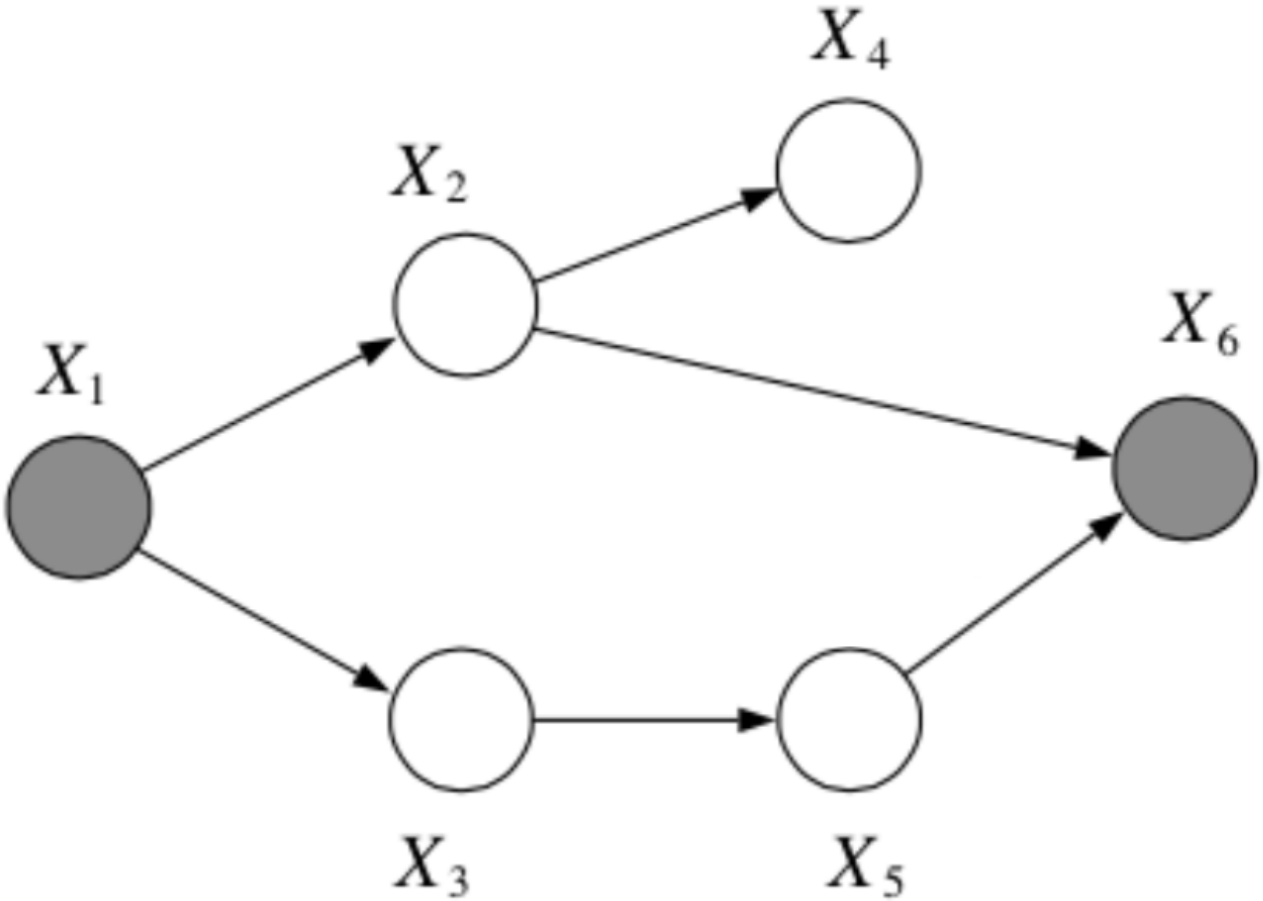
Example II: Solution

Yes, by the Bayes Ball algorithm.



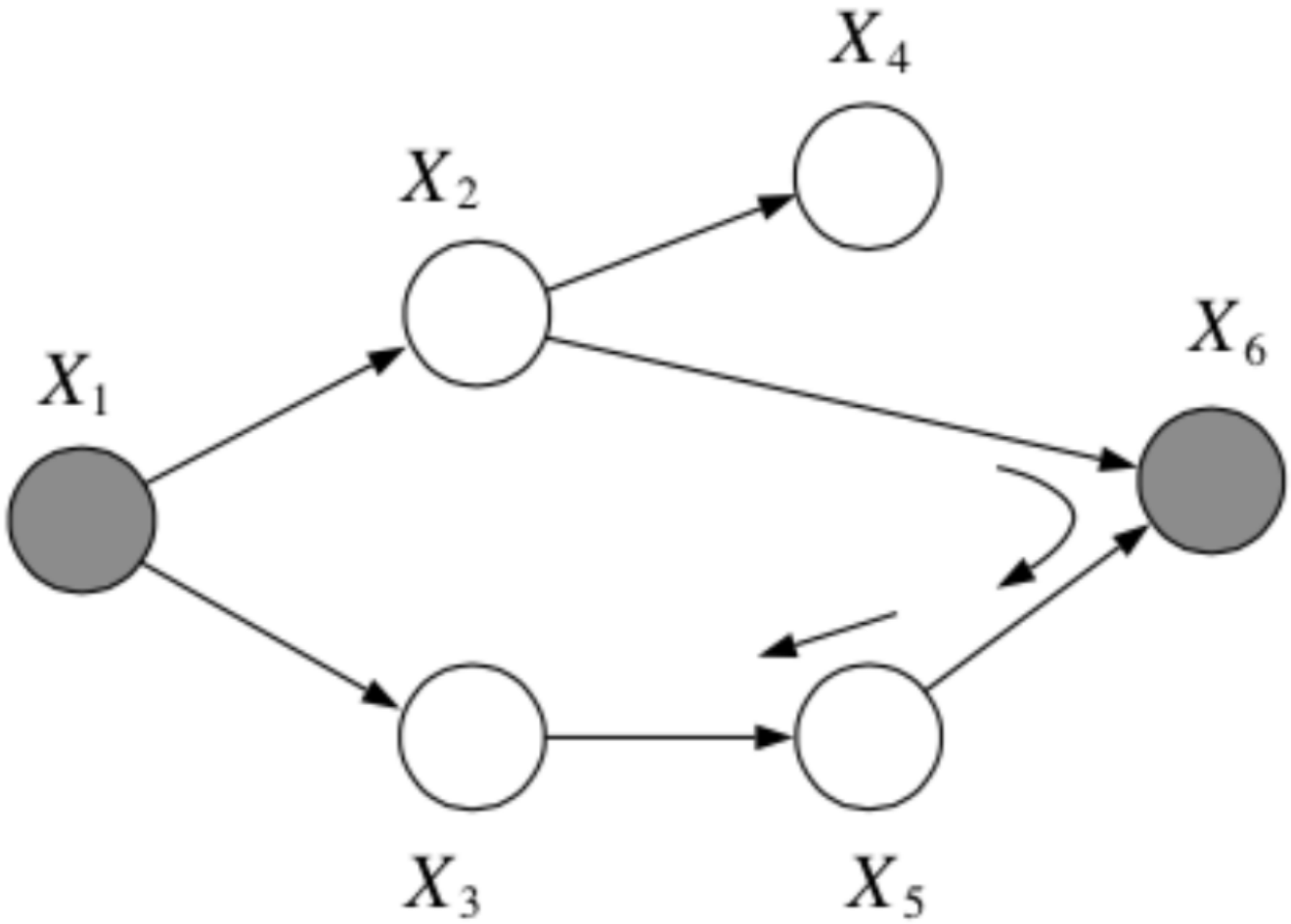
Example III:

In the following graph, is $x_2 \perp x_3 | \{x_1, x_6\}$?



Example III:

No, by the Bayes Balls algorithm.



Summary

- DAGs are great for encoding conditional independencies.
- They can reduce the number of parameters significantly.
- Conditional independence between two sets of variables on a DAG can be found using the Bayes ball method.
- Next lecture: Markov Random Fields.