Probabilistic graphical models

Quote

"Intelligence is not just about pattern recognition and function approximation. It's about modeling the world". — Josh Tenenbaum, NeurIPS 2021.

Graph

Graph is denoted $G = (V, E)$.

- V: set of nodes.
- E: set of edges describing relationship between nodes. Edges can be directed or undirected.
- Directed graph: all of the edges are directed, denoted $(u, v) = u \to v$ for $u, v \in V$. Each edge is an ordered pair, meaning $(u, v) \neq (v, u)$.
- Undirected graph: all of the edges are undirected, denoted $\{u, v\}$ for $u, v \in V$. An edge is represented as an unordered pair of vertices, i.e., $\{u, v\} = \{v, u\}$.

Graph

A **path** is a sequence of nodes where each pair of consecutive nodes are connected by an edge.

A simple path does not revisit nodes, whereas a general path may.

Directed graphs

- Parents are defined as $pa(v) = \{u : u \to v \in E\}$.
- Children: $ch(v) = \{u : v \to u \in E\}$.
- These concepts can be extended to define ancestors and descendants:

$$anc(v) = \{u : \exists \text{ a path from } u \text{ to v}\} \tag{1}$$

$$dec(v) = \{u : \exists \text{ a path from } v \text{ to u}\}. \tag{2}$$

Ancestors and descendants include indirect relationships established through one or more directed edges.

Directed acyclic graph (DAG)

A cycle is a path where the starting and ending nodes are the same, and no intermediate nodes are repeated.

DAG is a directed graph with no cycle.

- **Topological ordering**: DAG can be ordered such that parents come before children.

Directed graphical models

A Directed Graphical Model (DGM) is a DAG in which

- Each node $v \in V$ represents a random variable X_v.
- DGM encodes ordered **Markov property**:

$$X_v \perp X_u | X_{pa(v)} \text{ for } u \in anc(v) \quad pa(v)$$

This reduces the complexity of the model (e.g., number of parameters) by leveraging conditional independence.

Directed graphical models

- To specify a DGM, it suffices to specify conditional probabiliy distributions (CPD) $P(X_v|X_{pa(v)})$;
- The joint distribution factorizes as:

$$P(X) = \prod_v P(X_v|X_{pa(v)}); v \in V$$

This factorization reflects the dependency structure encoded in the graph, allowing for efficient representation and inference of the joint distribution.

Example: Markov chains

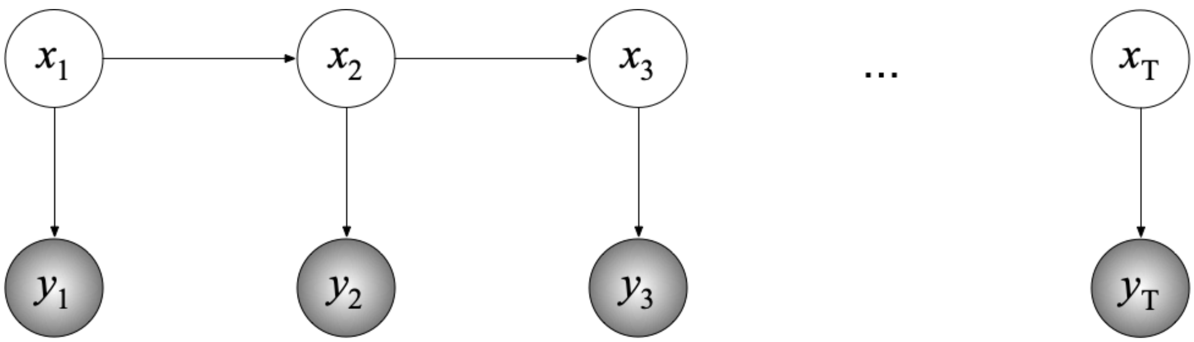
Time series data where the observation at time t depends on the past observations $1, ..., t-1$.

$$P(X_{1:T}) = P(X_1) \prod_{t=2}^{T} (X_t | X_{1:t-1})$$

First order Markov chain:

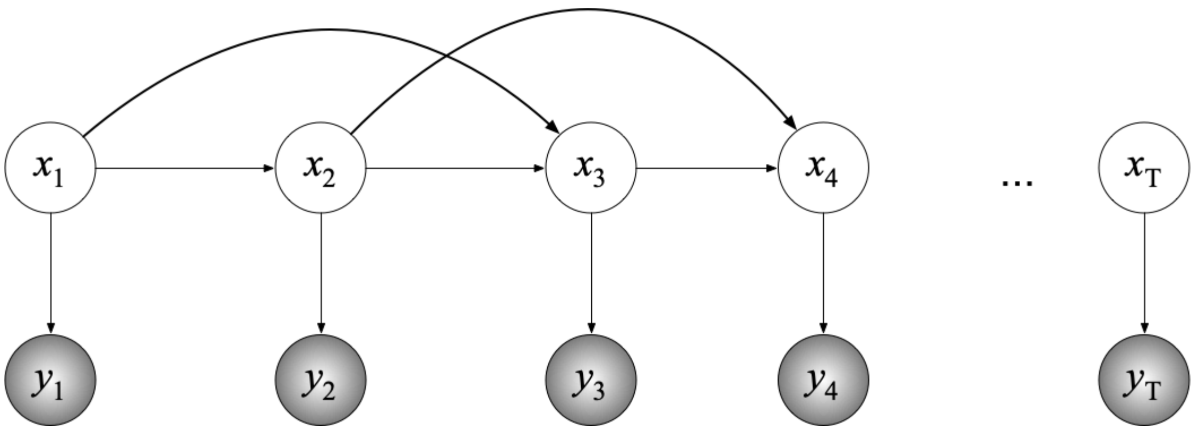
$$P(X_{1:T}) = P(X_1) \prod_{t=2}^{T} (X_t | X_{t-1})$$

Example: Hidden Markov model



First-order Markov chain on the latent state.

Example: Hidden Markov model



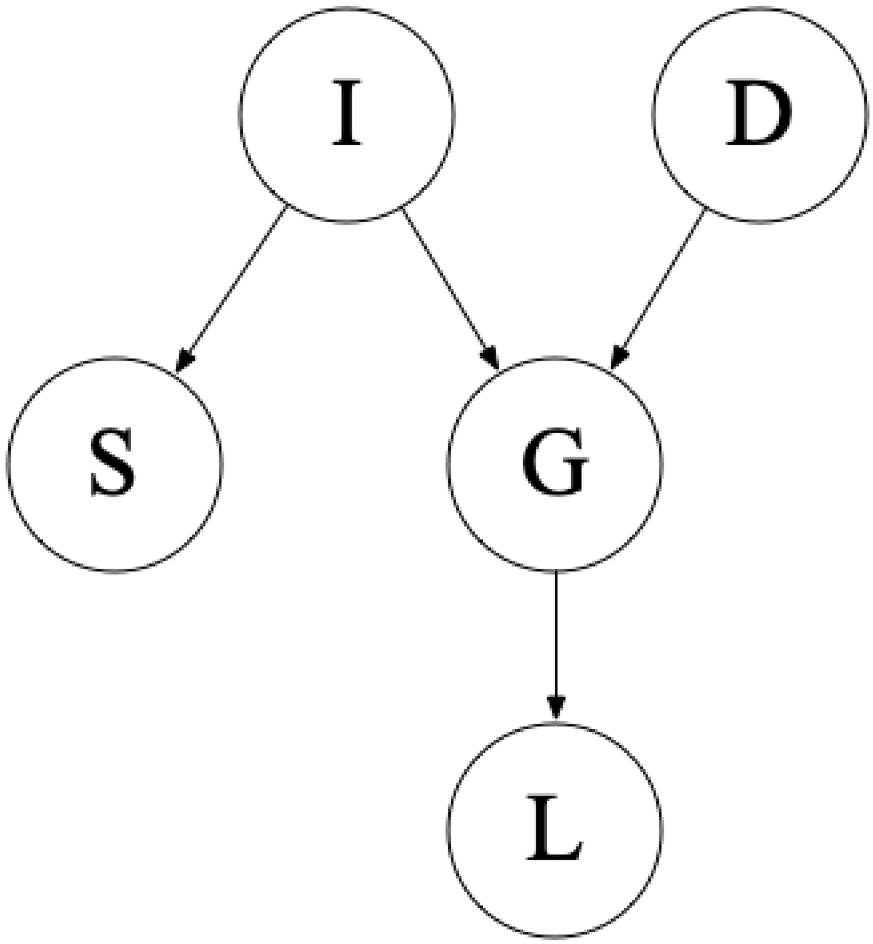
Second-order Markov chain on the latent state.

Example: The "student" network

- D: Difficulty of class (easy, hard)
- I: Intelligence (low, high)
- G: Grade (A, B, C)
- S: SAT score (bad, good)
- L: Letter of recommendation (bad, good)

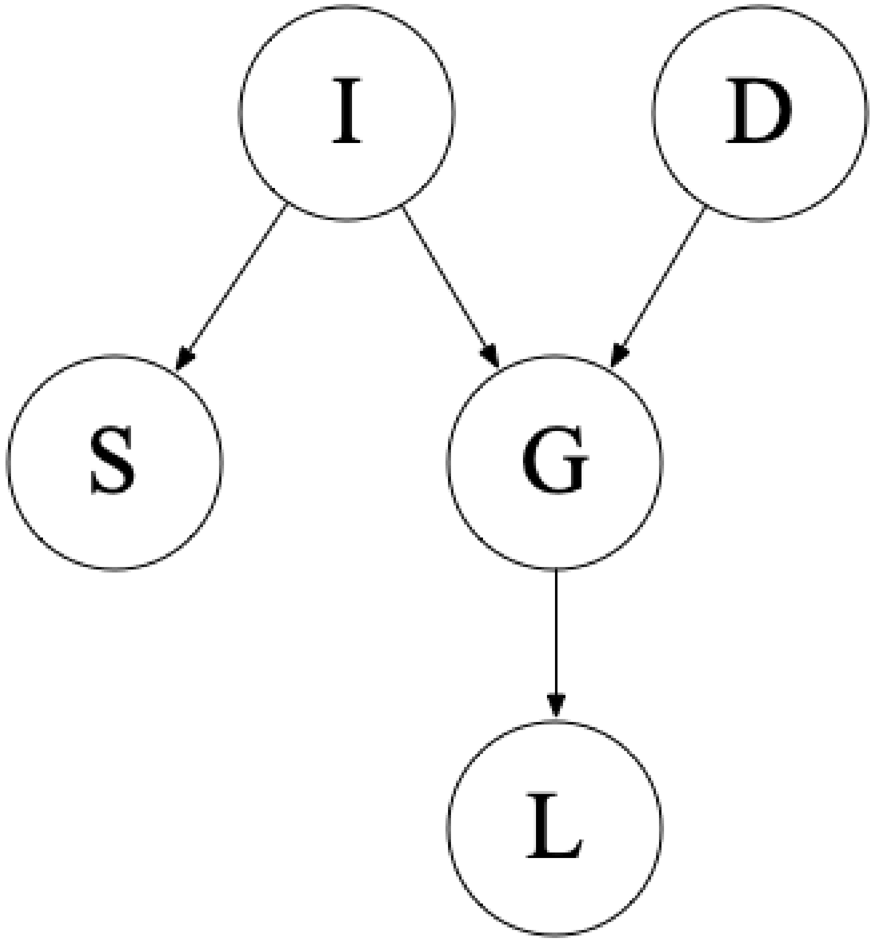
Exercise: construct a DGM on these variables. Think about which variables are the "causes" and which are the "outcomes."

Example: The "student" network



Difficult of class and Intelligence affects your Grade. Intelligence affects SAT score. Letter is determined by Grade.

Example: The "student" network



$$p(D, I, G, S, L) = p(I)p(D)p(G|D, I)p(S|I)p(L|G)$$

Conditional independence properties of DGMs

Let $G = (V, E)$ be a DAG and $A, B, C \subset V$. We want to determine if $A \perp_G B | C$.

Bayes ball algorithm.

- Shade all nodes in C as if though they are observed.
- Place a "ball" at each node in A.
- Let the balls bounce around according to the following rules and if any of the balls reach a node in B, the statement is not true.

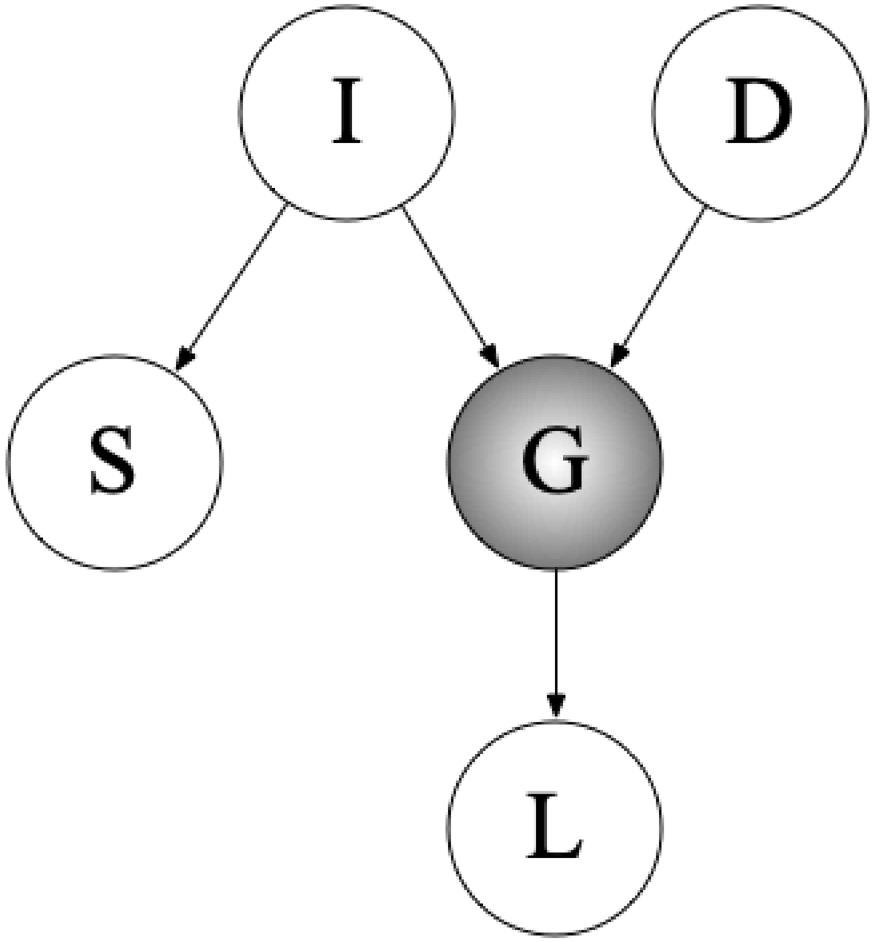
Bayes ball algorithm 1

Chain (pipe): $X \to Y \to Z$. Suppose we observe Y: $Y \in C$. Then, is $X \perp Z | Y$?

$$
\begin{aligned}
p(x, z|y) &= \frac{p(x, y, z)}{p(y)} && (3) \\
&= \frac{p(x)p(y|x)p(z|y)}{p(y)} && (4) \\
&= \frac{p(x, y)}{p(y)} p(z|y) && (5) \\
&= p(x|y)p(z|y). && (6)
\end{aligned}
$$

Therefore, if $Y \in C$, the ball cannot bounce from X to Z.

Bayes ball algorithm 1



Consider a path $D \to G \to L$ or $I \to G \to L$. The letter depends on the grade and if it is observed, the values of I and D are irrelevant (good grade will result in a good letter).

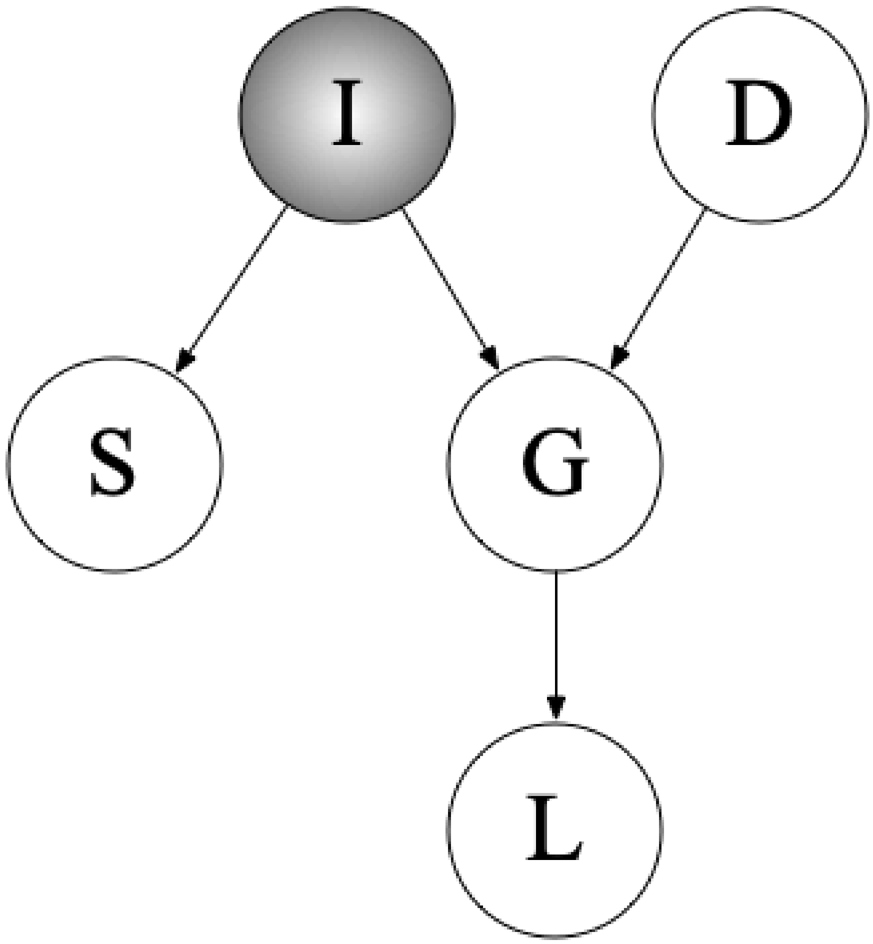
Bayes ball algorithm 2

Tent (fork): $X \leftarrow Y \to Z$. Suppose we Y is observed. Then, is $X \perp Z | Y$?

$$p(x, z|y) = \frac{p(y)p(x|y)p(z|y)}{p(y)} \tag{7}$$
$$= p(x|y)p(z|y). \tag{8}$$

So, YES! If we observe the root node, it separates the children and the ball cannot bounce from X to Z.

Bayes ball algorithm 2



The values taken by SAT score and Grade are independent given that we observe intelligence.

Bayes ball algorithm 3: Berkson's paradox

Collider (v-structure): $X \to Y \leftarrow Z$. Suppose we Y is observed. Then, is $X \perp Z | Y$?

$$p(x, z|y) = \frac{p(x)p(z)p(y|x, z)}{p(y)} \tag{9}$$

$$= \frac{p(x, z)p(y|x, z)}{p(y)} \tag{10}$$

If Y is observed, the ball can bounce from X to Z (and vice versa) via Y.

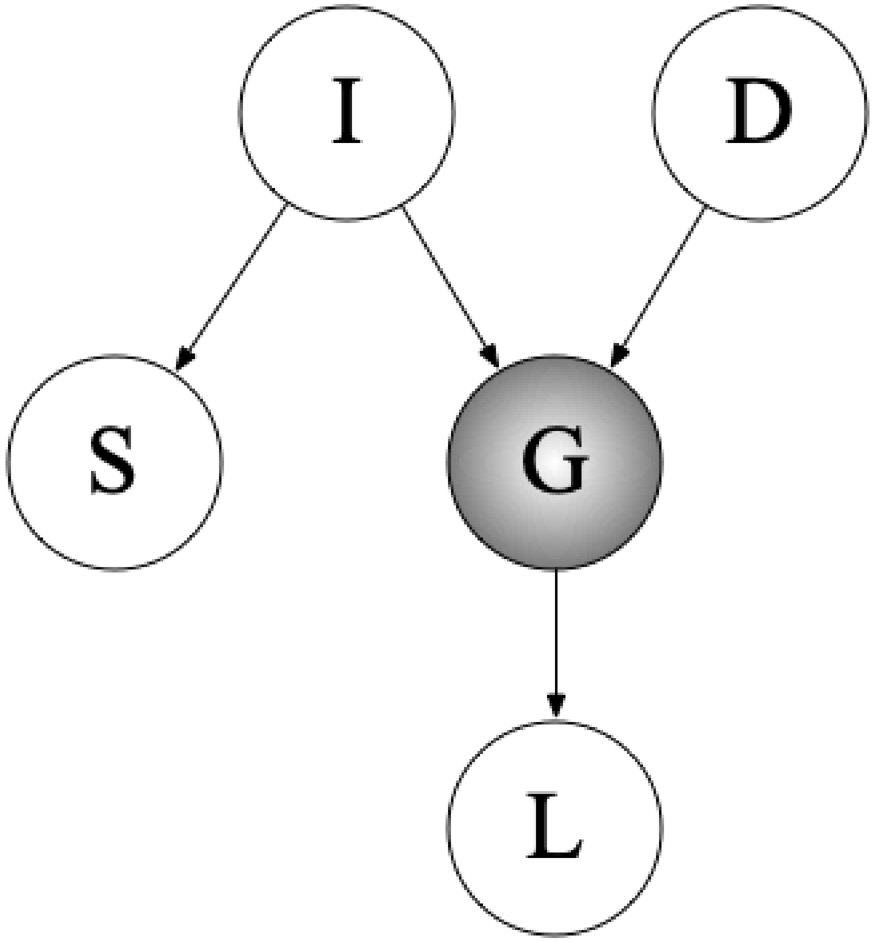
Bayes ball algorithm 3: Berkson's paradox

Note that marginally, $X \perp Z!!$ because $p(x, z) = p(x)p(z)$.

Therefore, ball cannot bounce from X to Z if Y is unobserved.

This is referred to as explaining away or Berkson' paradox, where the observed value can be due to either of the parent nodes.

Bayes ball algorithm 3: Berkson's paradox



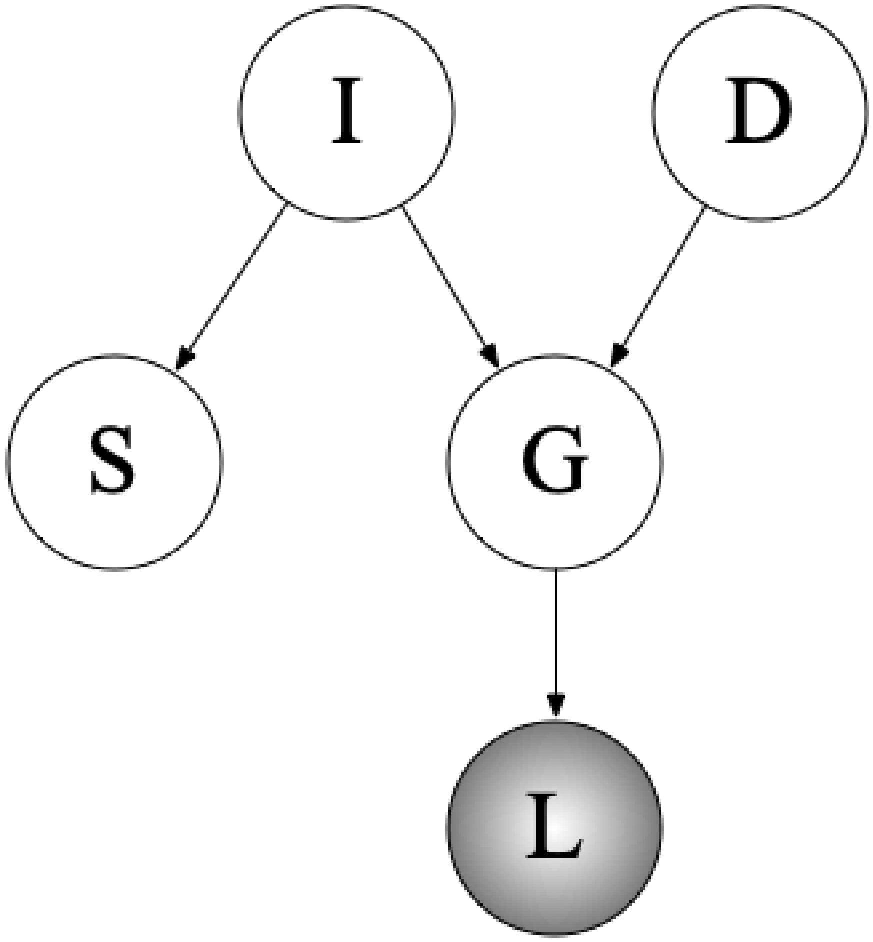
Consider $I \rightarrow G \leftarrow D$. Clearly intelligence of the student and difficulty of the class are unrelated (marginally independent). But if we observe $G = A$, then is it because the student is intelligent or because the course is easy? It can be explained away by either.

Bayes ball algorithm 3: Berkson's paradox

This conditional dependence arises because Y introduces a dependency between X and Z, even though they are marginally independent. This is a key property of colliders in DGMs.

For example, let $G = A$. Then, if I is known, then it will change our belief on D and vice versa. If $D = hard$ along with $G = A$, then we will update our belief so that $P(I = high)$ is high.

Bayes ball algorithm 4: boundary conditions



Can ball bounce from I to D and vice versa? Is $I \perp D | L$?

Bayes ball algorithm 4: boundary conditions

Essentially, observing L unblocks G as a collider so that ball can bounce from I to D.