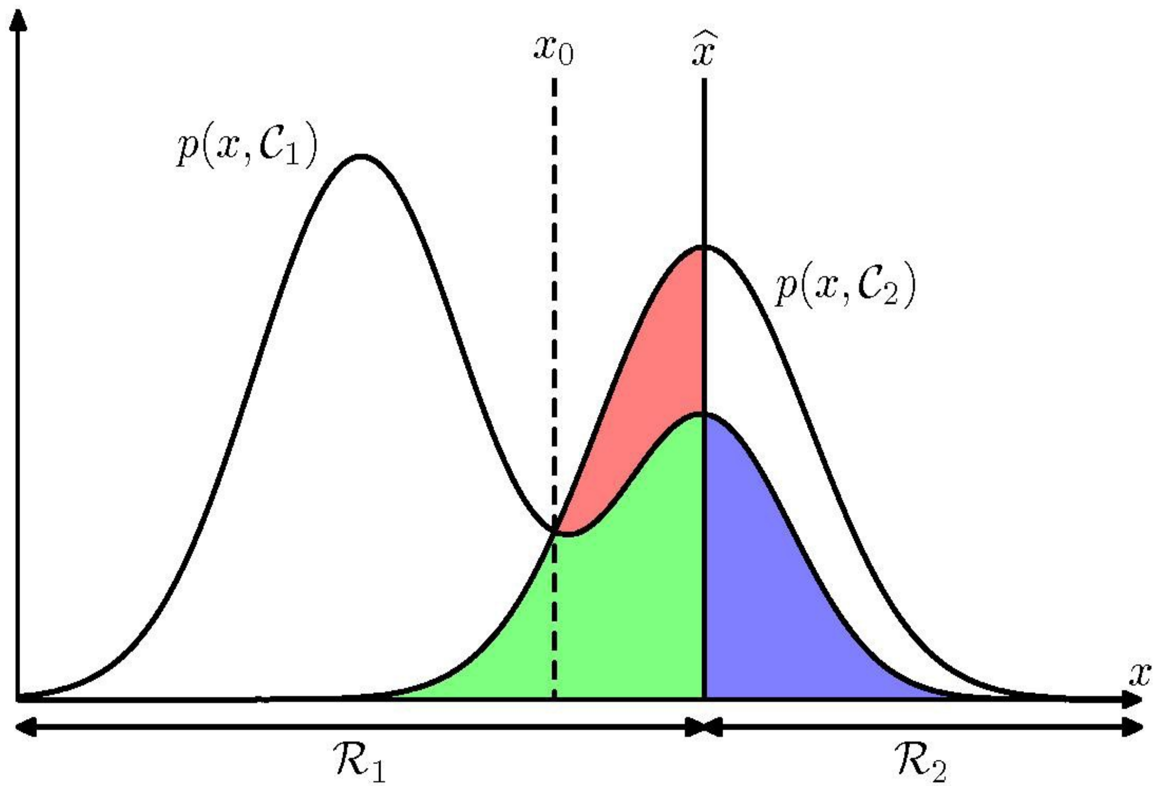
Lecture 1: PGM DAG

Probabilistic Graphical Models, Koller and Friedman:

- Chap 3

- Generative vs. Discriminative models, Bayes Nets (BN), Imap, D-separation, Markov Blankets.

Minimize Decision Error



$$P(\text{error}) = P(x \in R_2, C_1) + P(x \in R_1, C_2)$$

$$= P(x \in R_2 \mid C_1)\, P(C_1) + P(x \in R1, \mid C_2)\, P(C_2)$$

$$= \int_{R2} p(x \mid C_1)\, P(C_1) dx + \int_{R1} p(x \mid C_2)\, P(C_2) dx$$

Figure from the lectures of: Bjoern Andres and Bernt Schiele

3 Ways to Solve Decision Problems

1. Generative models:

   model (learn): $p(X|C), P(C)$;

   Use Bayes Theorem: $P(C|X) = p(X|C)P(C)/p(X)$

2. Discriminative models:

   infer posterior probabilities directly $P(C|X)$

3. Directly estimate a discriminating function that will minimize some loss.

   f(X)=C (so no probability, ie SVM...)

Generative Models

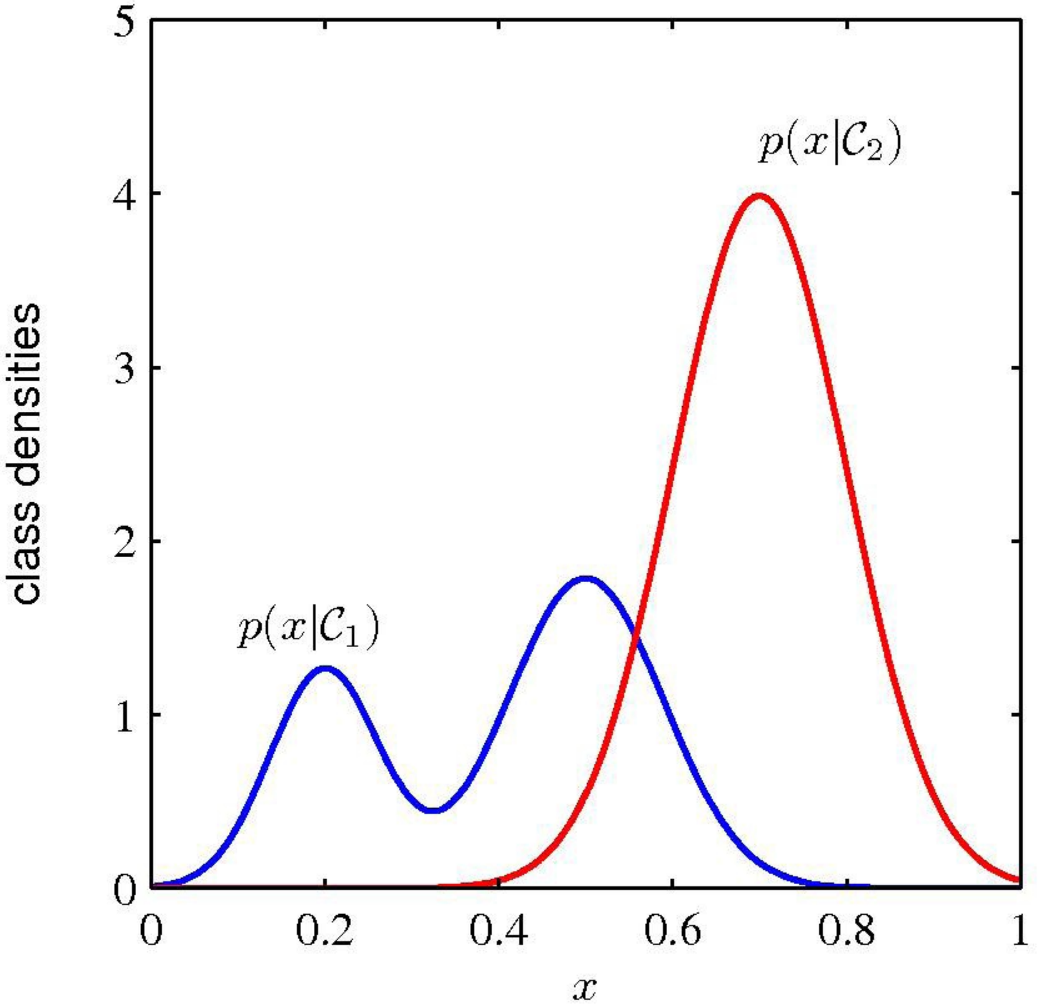
Pros:

- The name generative is because we can generate samples from the learned distribution
- Perhaps $p(\boldsymbol{X}|C)$ is easier to estimate

Cons:

- With high dimensionality of $\boldsymbol{X}$ we need a large training set to determine the class-conditionals
- We may not be interested in all quantities

[From (4)]



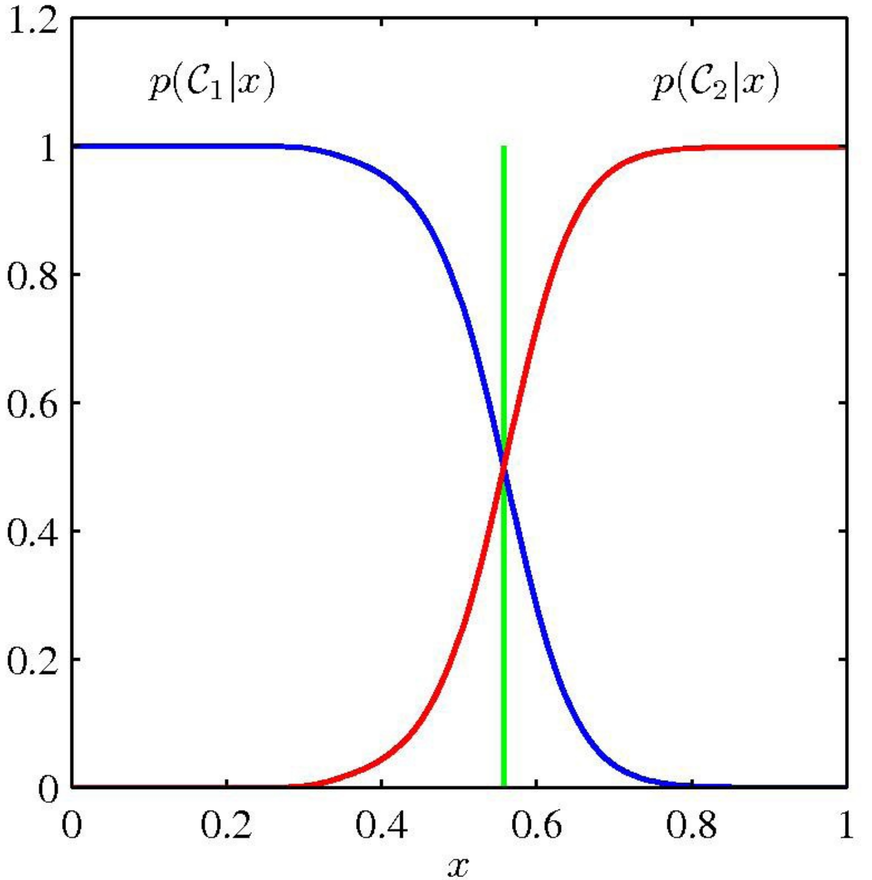
Discriminative Models

Pros:

- No need to model *p(x|C)* which is not technically needed.

Cons:

- No access to model p(x|C) which might be needed.



[From (4)]

PGMs: Core concepts

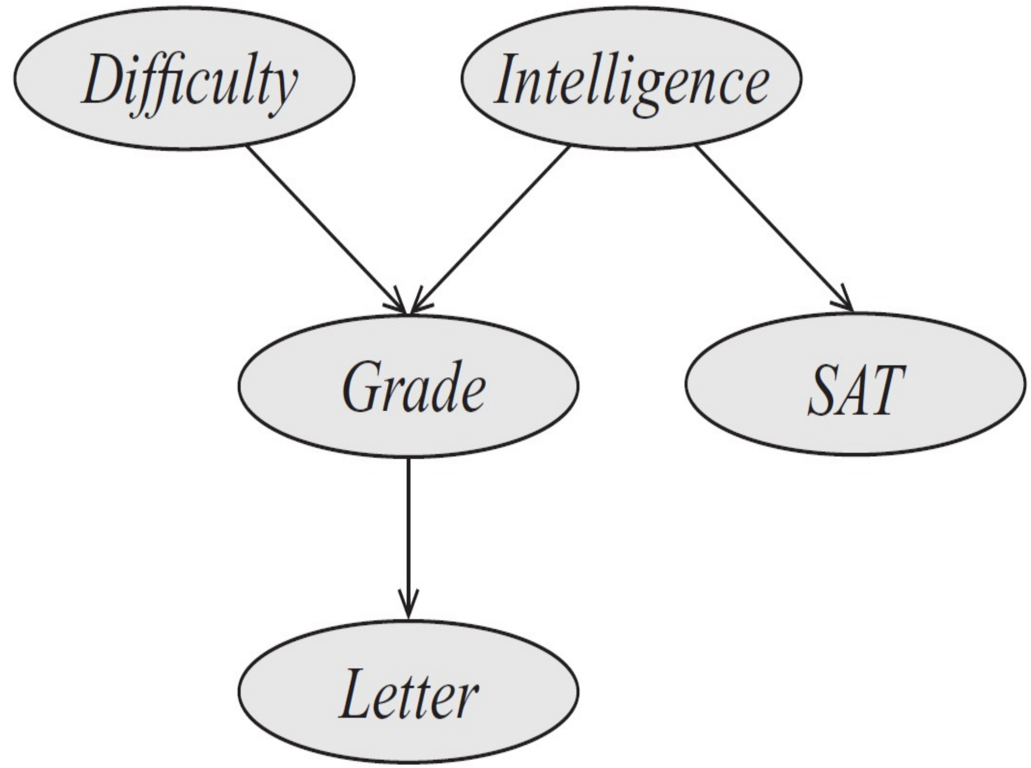
Key idea: PGMs effectively encode the information in a joint distribution:

E.g., $P(X_1, X_2, X_3, \ldots, X_N)$

- Conditional Prob: $P(X_1 \mid X_2) = P(X_1, X_2)/P(X_2)$
- Independence: $P(X_1 \mid X_2) = P(X_1)$
- Marginal: $P(X_1) = \Sigma_x P(X_1, X_2{=}x)$
- Conditional Indep: $P(X_1 \mid X_2, Z) = P(X_1 \mid Z)$

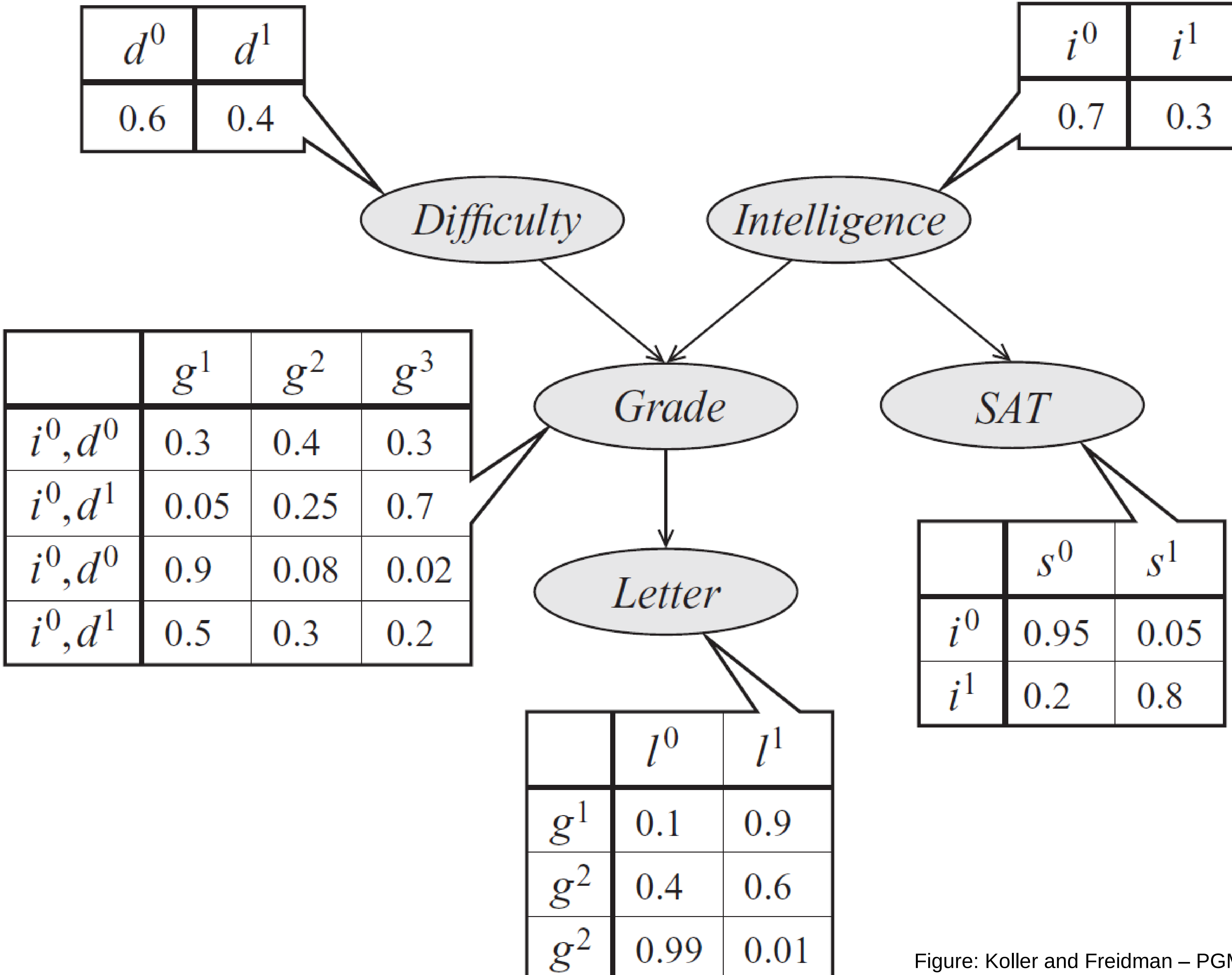
A Bayes Net

- Directed acyclic graph
  - DAG for short
- Encodes conditional (in)dependance:



- $P(L) = \Sigma_G P(L \mid G)P(G)$
- $P(L \mid G, S, D, I) = P(L \mid G)$
- G is the ‘parent’ of L and D and I are ‘parents’ of G.

Figure: Koller and Freidman – PGM Principles and Techniques



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

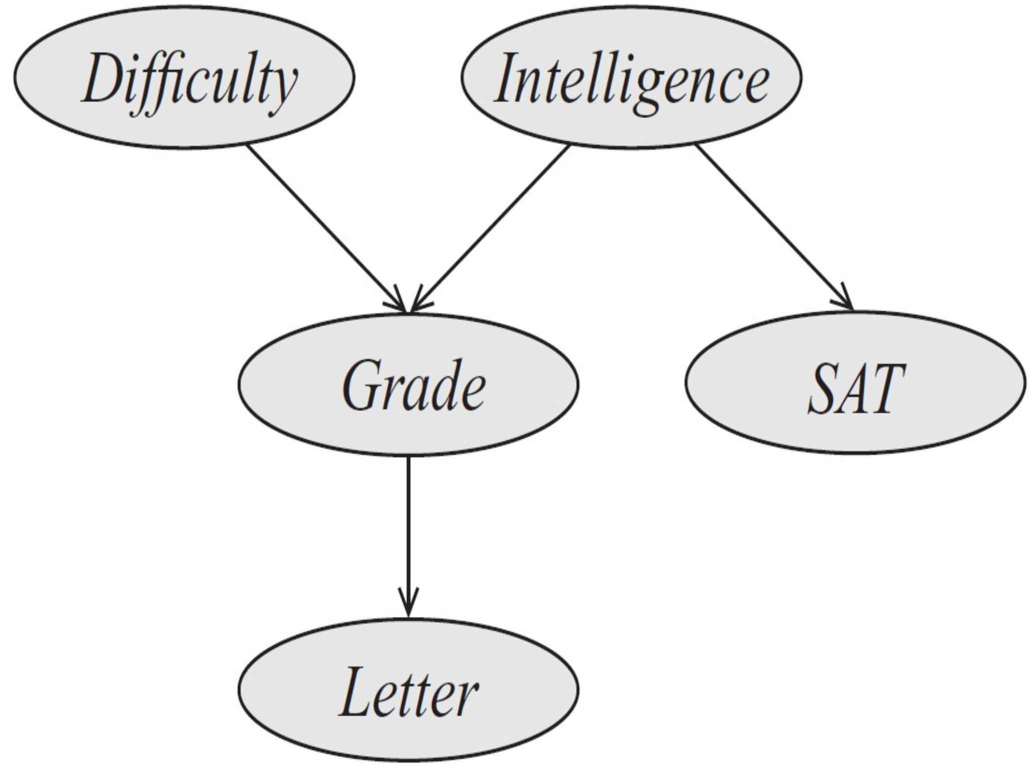
	g^1	g^2	g^3
i^0, d^0	0.3	0.4	0.3
i^0, d^1	0.05	0.25	0.7
i^0, d^0	0.9	0.08	0.02
i^0, d^1	0.5	0.3	0.2

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^2	0.99	0.01

Figure: Koller and Freidman – PGM Principles and Techniques

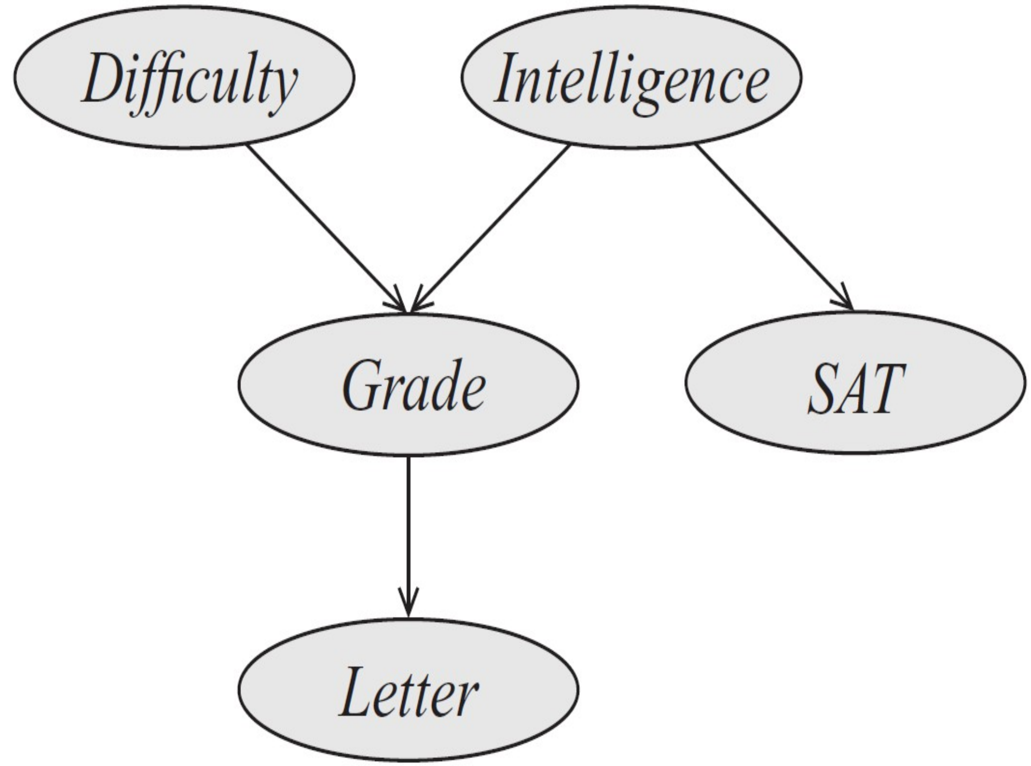
A Bayes Net



- $P(G \mid L,D,I) \neq P(G \mid D, I)$

Figure: Koller and Freidman – PGM Principles and Techniques

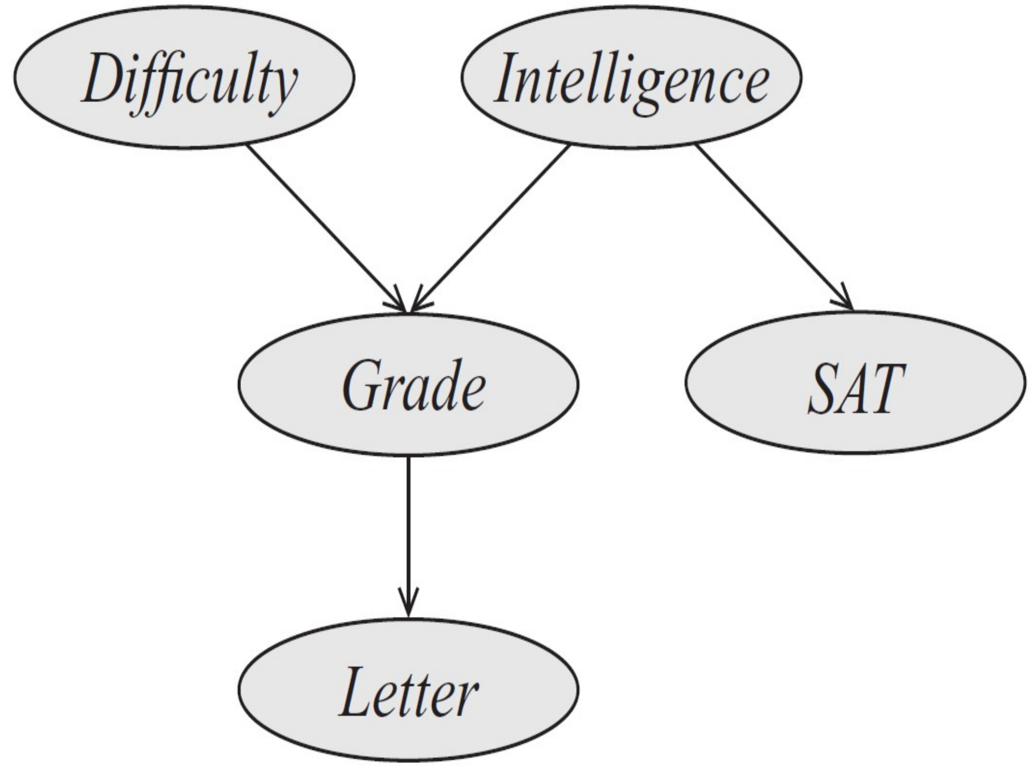
A Bayes Net



- $P(G \mid L,D,I) \neq P(G \mid D, I)$
- *A node and its descendants are conditionally independent of the non-descendants given the node's parents.*

Figure: Koller and Freidman – PGM Principles and Techniques

A Bayes Net



- Chain rule for Bayes Nets:
- $P(I,D,G,L,S) = P(I, D, G, L)\, P(S \mid I, D, G, L)$; *any joint can be factored so*
- $P(I,D,G,L,S) = P(I)\, P(D \mid I)\, P(G \mid I, D)\, P(L \mid I, D, G)\, P(S \mid I, D, G, L)$; repeat

$= P(I)\, P(D)\, P(G \mid I, D)\, P(L \mid G)\, P(S \mid I)$; use the net

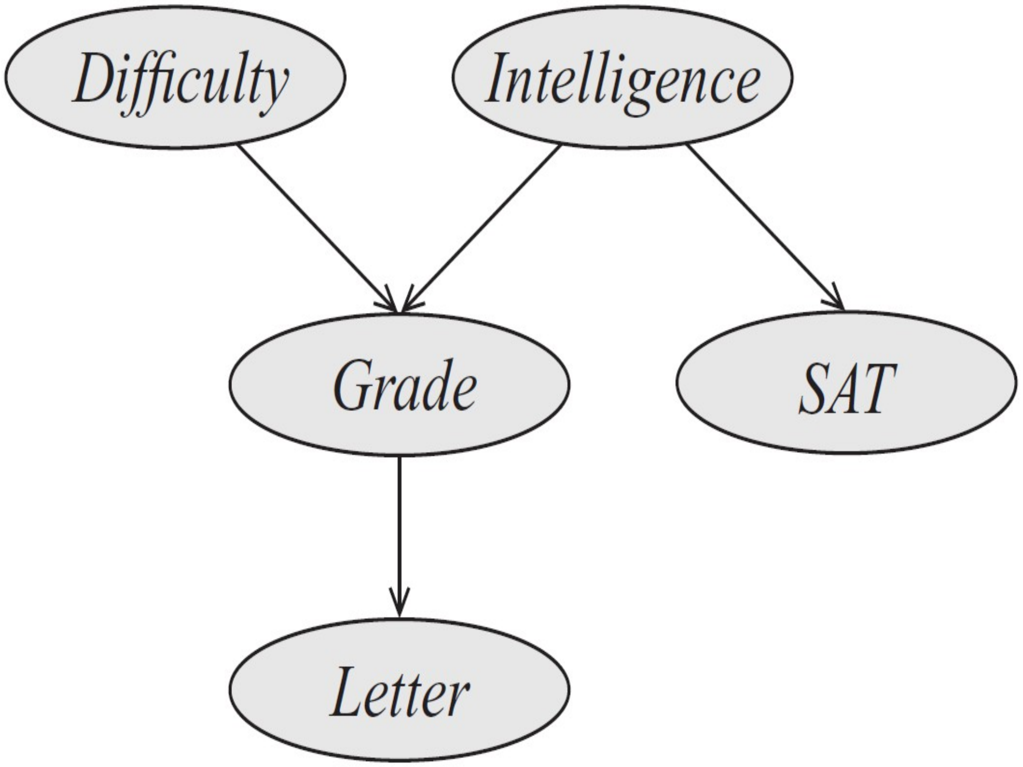
$= \Pi_i\ P(X_i \mid Pa_{X_i})$; general BN factorization

Figure: Koller and Freidman – PGM Principles and Techniques

Did you pay attention?

- That last slide is the essence of why we use BN

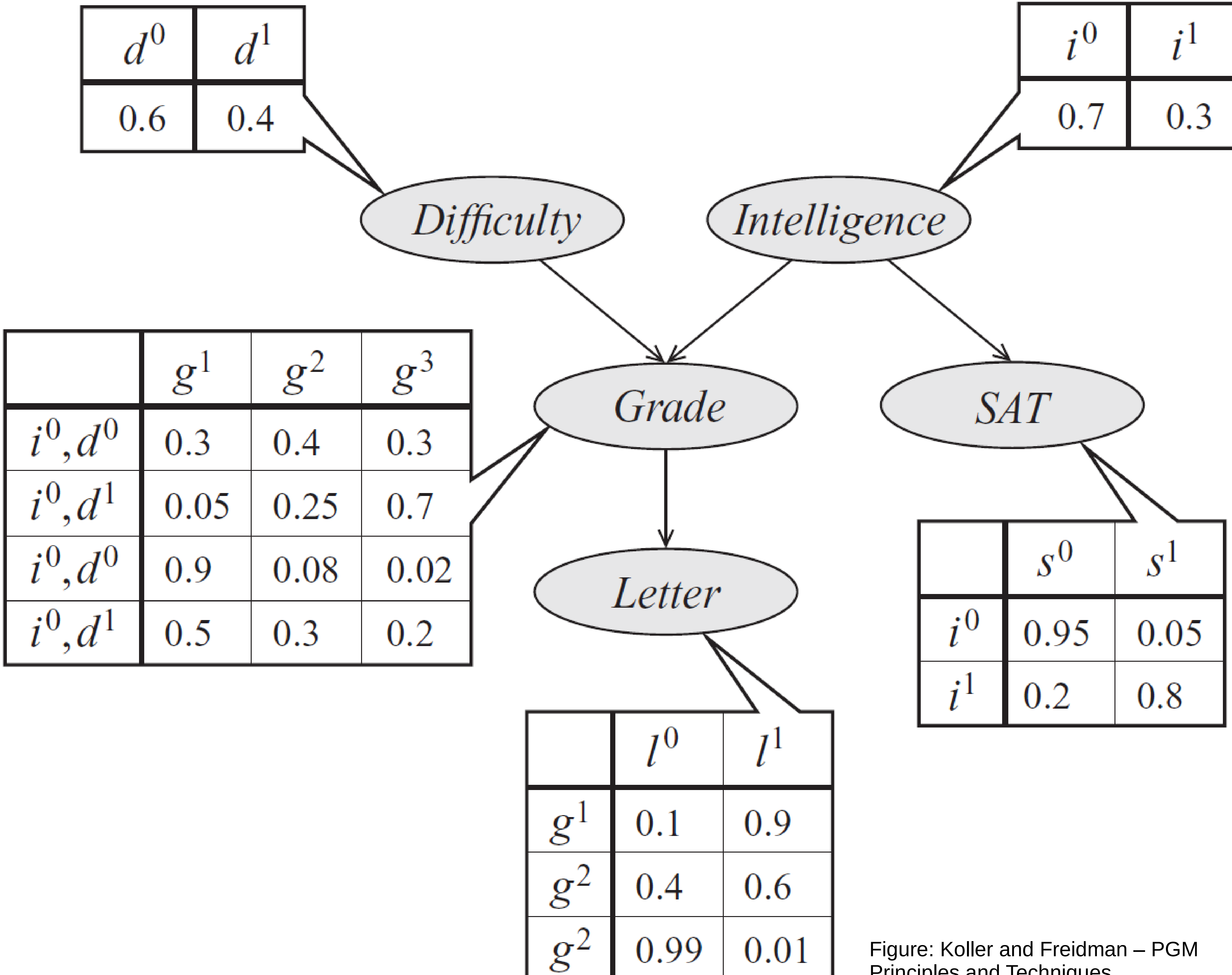
**** Bayes Net Factorization ****



- $P(I,D,G,L,S) = P(I)\,P(D)\,P(G \mid I, D)\,P(L \mid G)\,P(S \mid I);$

  $= \Pi_i\ P(X_i \mid Pa_{X_i});$

Figure: Koller and Freidman – PGM Principles and Techniques



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

	g^1	g^2	g^3
i^0, d^0	0.3	0.4	0.3
i^0, d^1	0.05	0.25	0.7
i^0, d^0	0.9	0.08	0.02
i^0, d^1	0.5	0.3	0.2

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^2	0.99	0.01

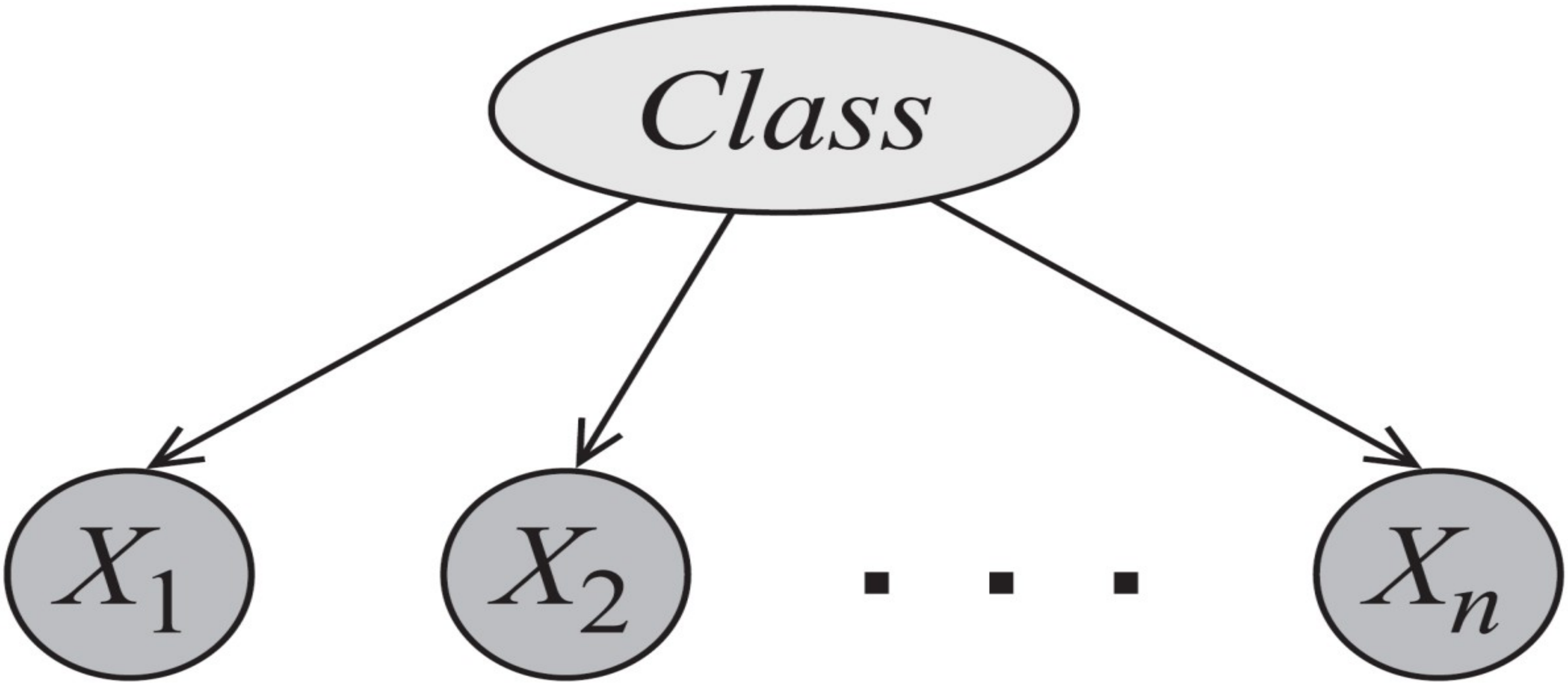
Figure: Koller and Freidman – PGM Principles and Techniques

Independence Relations

- A DAG, call it G, tells us the 'local independencies'
  - $X_i \perp$ Non descendants of $X_i \;\; | \;$ Pa(X_i)
  - $\mathcal{I}_\ell$(G) is the set of all such 'local independencies' given G
- This then get connected to a similar concept for P, a probability distribution:
  - The set of all independence statements

    $\mathbf{X} \perp \mathbf{Y} \mid \mathbf{Z}$ is called $\mathcal{I}$(P)
- $\mathcal{I}_\ell(G) \subseteq \mathcal{I}(P)$ means G is an **I-map** for P
- Point is now we can use G to factor P iff G is an I-map for P

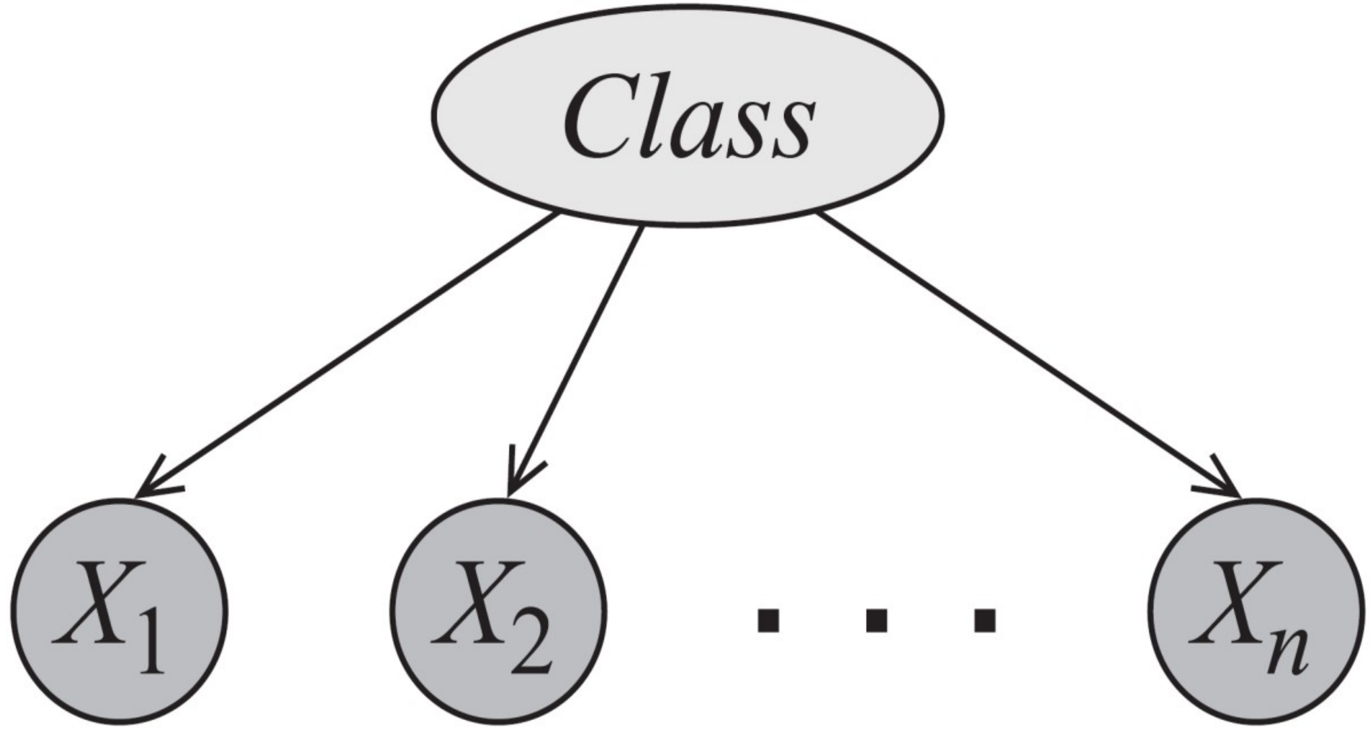
Naive Bayes Net



$$P(X_1, X_2, X_3, \ldots, X_N, C) = \prod_{i=0..N} P(X_i \mid Pa_{X_i})$$

$$= P(C)\prod_{i=1..N} P(X_i \mid C)$$

Figure: Koller and Freidman – PGM Principles and Techniques



$$P(X_1, X_2, X_3 \ldots, X_N)$$

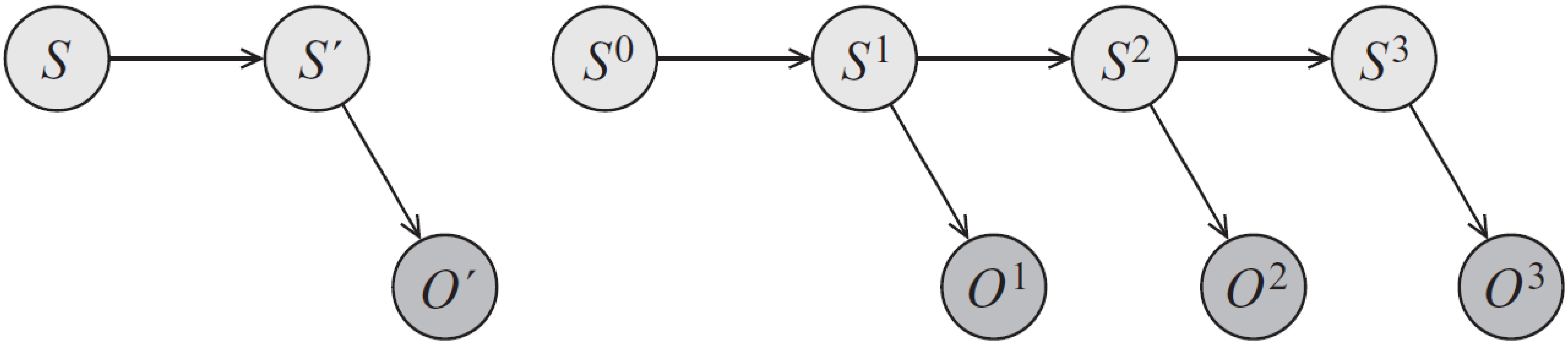
$$= \sum_{c=1\ldots M} P(C{=}c) \prod_{i=1..N} P(X_i \mid C{=}c)$$

$$P(C{=}c \mid X_1, X_2, X_3 \ldots, X_N)$$

$$= P(X_1, X_2, X_3 \ldots, X_N, C{=}c) / P(X_1, X_2, X_3 \ldots, X_N)$$

Figure: Koller and Freidman – PGM Principles and Techniques

Hidden Markov Models



(a) Discriminative Models (b)

- Left is a Dynamic Bayes Net that is the model for each link in the right PGM (Stationary)
- $P(S \mid S')$ called Transition
- $P(O'|S')$ called Emission
- S's are the 'hidden' states

Figure: Koller and Freidman – PGM Principles and Techniques

Collisions

- $A \perp C \mid B, D$? yes
- $D \perp B \mid A$? yes
- $D \perp B \mid C, A$? no!

  C is a 'collider'
- $A \perp C \mid B$? no

  why? Because
- $P(A,B,C,D) = P(A)P(B \mid A)P(D \mid A)\,P(C \mid B, D)$

  huh?

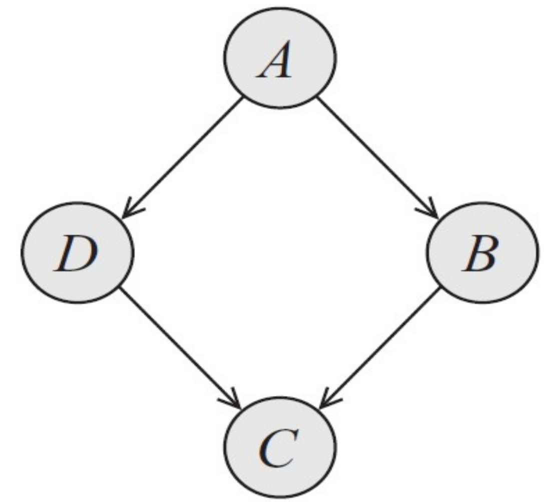
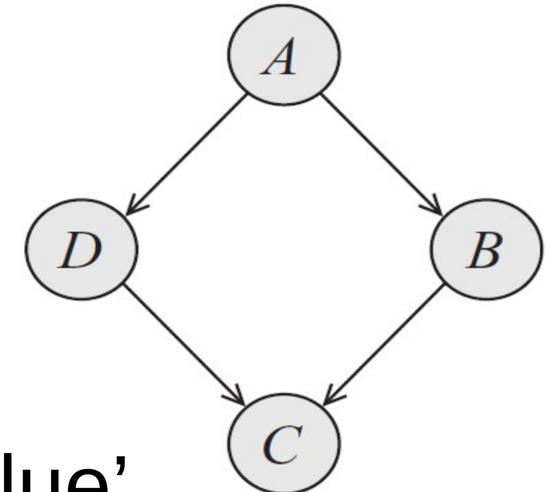


Figure from the lectures of: Bjoern Andres and Bernt Schiele

Collisions



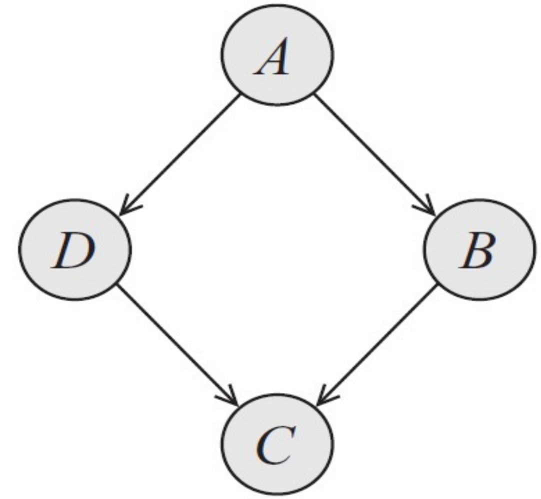
- $A \perp C \mid B, D$? yes
- Lets use a shorthand:
- Small letter will mean 'that random variable takes on that value'
- P(A | B=b) = P(A | b)

  So then we reason:
- $P(A,B,C,D) = P(A)P(B \mid A)P(D \mid A)\,P(C \mid B, D)$
- $P(A,C,b,d) = \{P(A)P(b \mid A)P(d \mid A)\} \times \{P(C \mid b, d)\}$
- And $P(A,C \mid b, d) = P(A,C,b,d) / P(b,d)$

Figure from the lectures of: Bjoern Andres and Bernt Schiele

Collisions

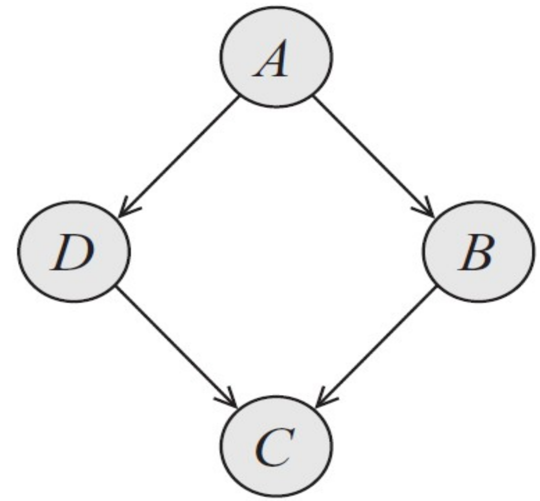
- $D \perp B \mid A$? yes
- When we are missing a variable it must be marginalized.



- P(A,B,C,D)= P(A)P(B | A)P(D | A) P(C | B, D)
- P(D, B, a)= P(a) P(B |a) P(D|a) $\Sigma_C P(C \mid B, D)$
- *P(D, B, a)= {P(a) P(B |a)} x {P(D|a)}*
- *Why?*

Figure from the lectures of: Bjoern Andres and Bernt Schiele

Collisions



- $D \perp B \mid C, A$? no!

  C is a 'collider'

  Sometimes what is true for the sum is

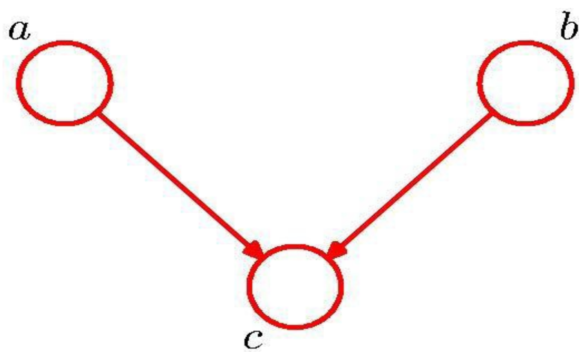
  not true for the individual values.

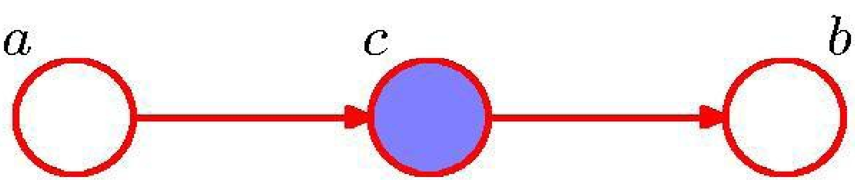
- P(A,B,C,D)= P(A)P(B | A)P(D | A) P(C | B, D)
- P(a,B,c,D)= P(a)P(B | a)P(D | a) P(c | B, D)
- Since we can not sum over the c we get B and D all mixed up in that table and so it will not factor into a B part and a D part.

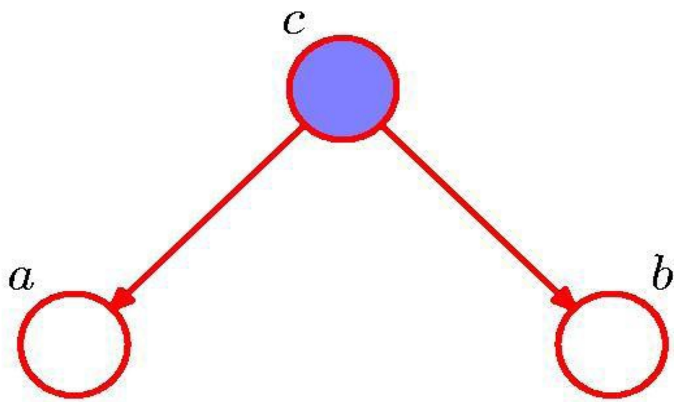
Figure from the lectures of: Bjoern Andres and Bernt Schiele

D-separation a ⊥ b | filled node ?

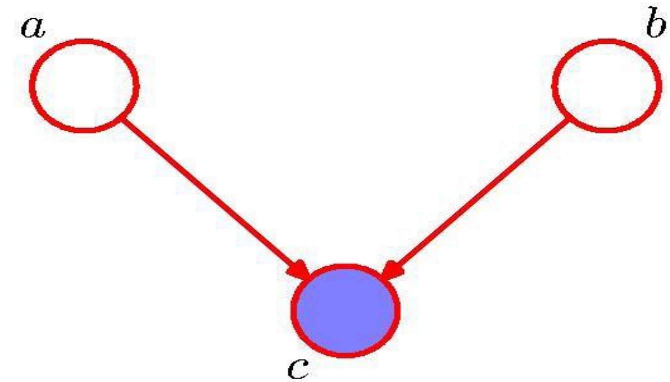
Yes, Blocked:

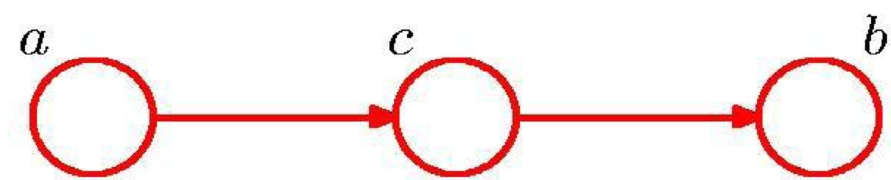






No, d-Connected





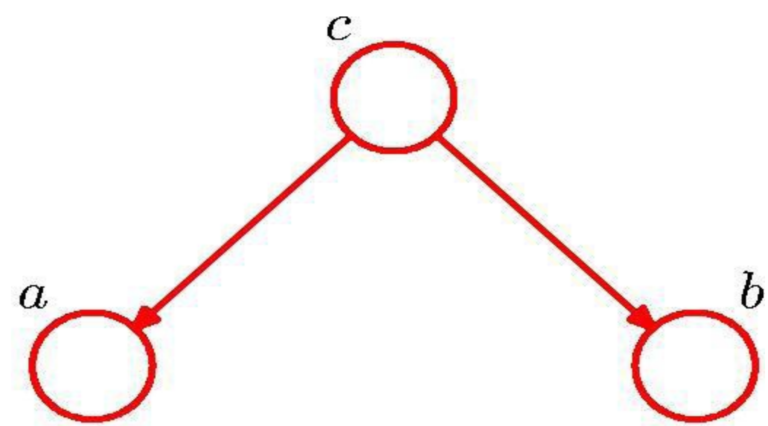


Figure from Bishop-
Pattern Recognition
and ML

**** D separation ****

For every $x \in$ X , $y \in$ Y and a set of nodes Z, check every path U between x and y.

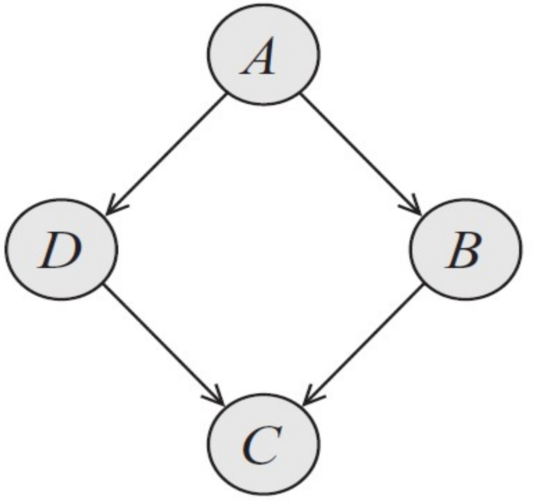
A path is blocked if there is a node w on U such that either

1. w is a collider and neither w nor any descendant is in Z, or
2. w is not a collider on U and w is in Z

If all such paths are blocked then X and Y are d-separated by Z. (Otherwise they are d-connected)

- D-separated $\rightarrow$ X $\perp$ Y | Z

d-separation



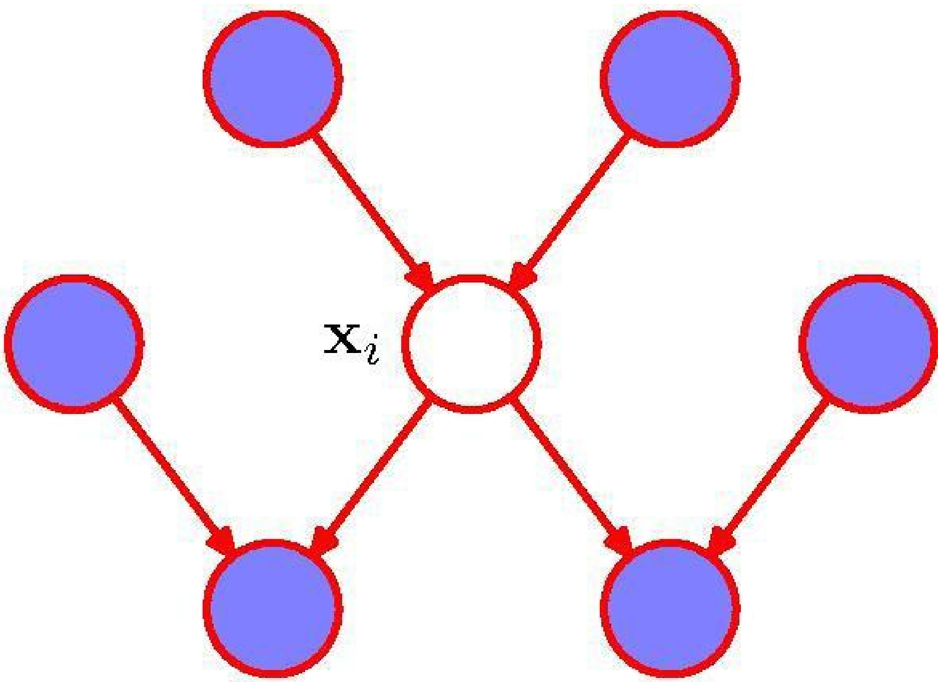
- $D \perp B \mid C$? no! C is
  a 'collider' and in Z
  So {C} d-connects {D,B}

- $D \perp B \mid A$? yes
  A is not a collider in Z
  and C is a collider not in Z
  So {A} d-separates {D,B}

Markov Blanket

e.g. $p(x_1, | x_{2...M}) = p(x_1, x_2, x_3, x_4) / p(x_2, x_3, x_4)$

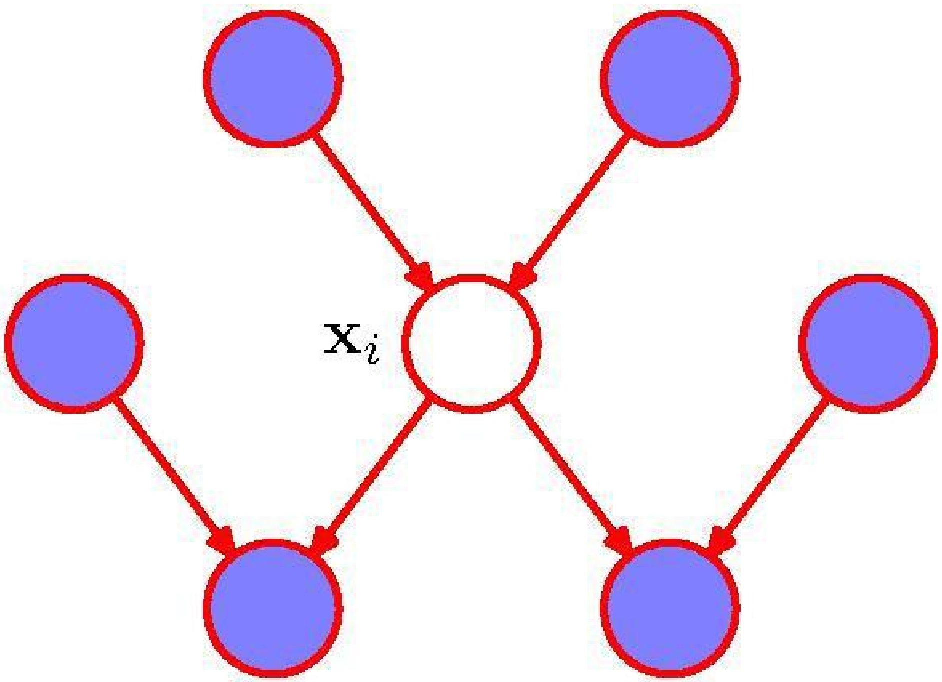
$$p(\mathbf{x}_i | \mathbf{x}_{\{j \neq i\}}) = \frac{p(\mathbf{x}_1, \ldots, \mathbf{x}_M)}{\int p(\mathbf{x}_1, \ldots, \mathbf{x}_M)\,\mathrm{d}\mathbf{x}_i} = \frac{\prod_k p(\mathbf{x}_k | \mathrm{pa}_k)}{\int \prod_k p(\mathbf{x}_k | \mathrm{pa}_k)\,\mathrm{d}\mathbf{x}_i}$$



Factors independent of x_i cancel between numerator and denominator.
That is not in some node's table along with x_i.

Figure from Bishop- Pattern Recognition and ML

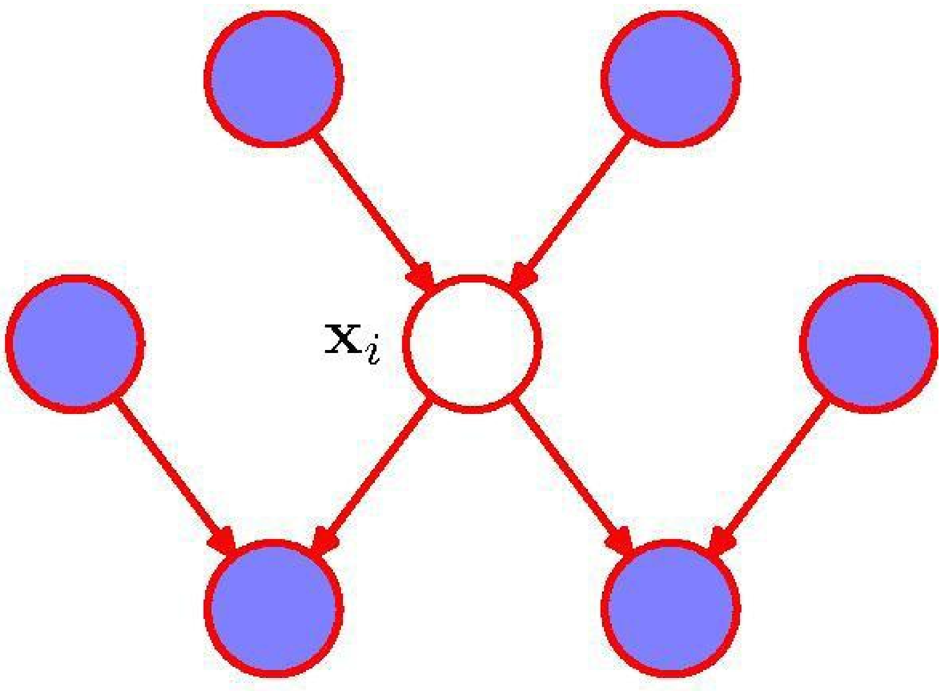
Markov Blanket



- We have now shown there are more independencies in the graph.
- Not just between x and its non-descendants.

Figure from Bishop- Pattern Recognition and ML

Skeleton and V-Structures



- Skeleton is the graph with no arrowheads
- V-Structures are where two arrows point into a node (a collider).
- They define the D-separation and all independancies.

Figure from Bishop- Pattern Recognition and ML

Local vs Global Independencies

- The set of edges in a BN define, $\mathcal{I}_\ell(G)$, local independences over a set of random variables.
- ie. between X and it's non-descendants given it's parents.
- A given probability distribution, *P*, algebraically has a set of, $\mathcal{I}(P)$ independences: $\mathbf{X} \perp \mathbf{Y} \mid \mathbf{Z}$.
- $\mathcal{I}_\ell(G) \subseteq \mathcal{I}(P)$ means G is an **I-map** for P.

Local vs Global Independencies

- D-separation in the graph defines a new set $\mathcal{I}(G)$.
- If P factorizes according to G then $\mathcal{I}(G) \subseteq \mathcal{I}(P)$ (Theorem 3.3)
- Lots of theorems about this in the book but one interesting one is 3.5:
- If P factorizes according to G then $\mathcal{I}(G) = \mathcal{I}(P)$
  - Except for a set of ‘measure zero’ (ie weird cases with ‘accidental’ independencies)

Minimal I-map

- Less edges ‘increases’ $\mathcal{I}$(G).
  - ie. less dependencies is more independencies.
- Two graphs are I-Equivalent if independences are the same $\rightarrow$ same skeleton and v-structures.
- Minimial I-map says remove any edge and it is no longer an I-map, (i.e. NOT $\mathcal{I}_\ell$(G) $\subseteq$ $\mathcal{I}$(P)).
- If $\mathcal{I}$(G) = $\mathcal{I}$(P) we say G is a P-map (perfect map). We can read off all independence from G.
- Minimal I-map does not imply a perfect map.

An Example: Bayes nets for Driver Intention

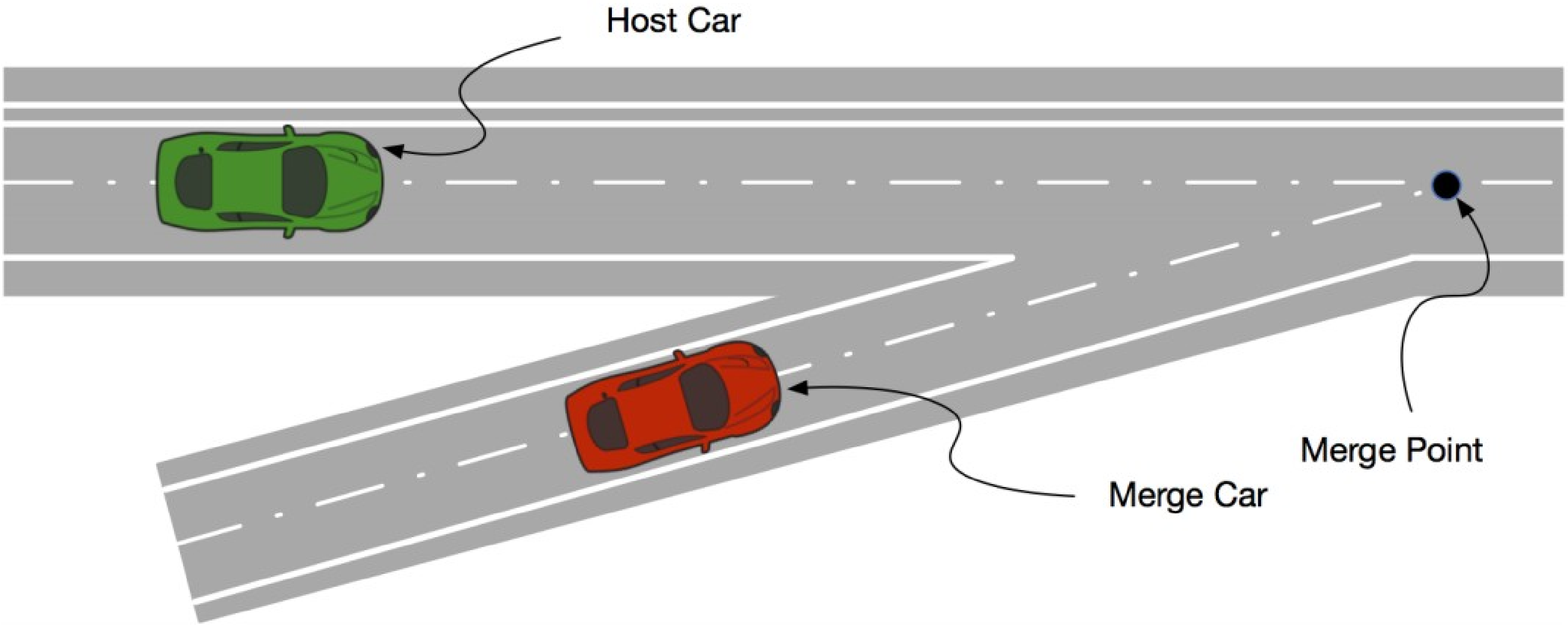


Fig. 1: Merge scenario. The host car (green) is an autonomous vehicle, running on the main road; the merge car (red) is a human-driven car, running on the ramp.

From: Intention Estimation For Ramp Merging Control In Autonomous Driving, Chiyu Dong, John M. Dolan, and Bakhtiar Litkouhi

Bayes Nets for Driver Intention

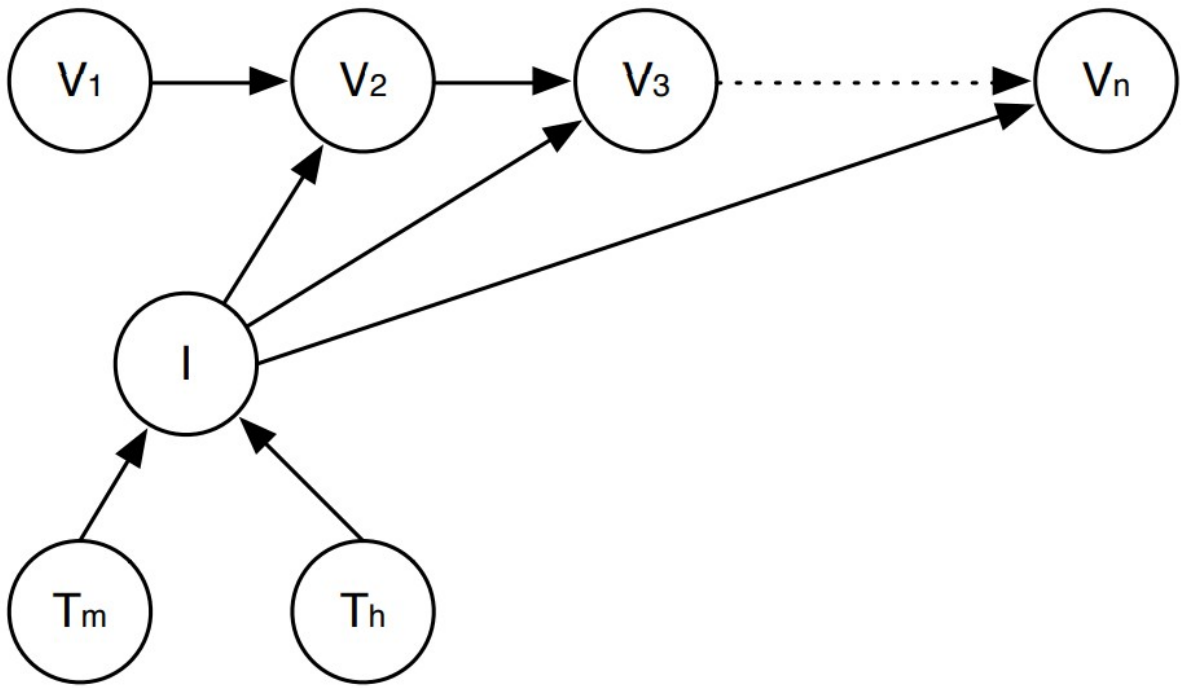


Fig. 2: Probabilistic Graphical Model of the social behavior of an autonomous vehicle.; V_n is the current speed, V_i is the speed at the previous time step; T_m, T_h are the current time-to-arrival for merging and host car respectively; I is the latent intention which needs to be estimated.

From: Intention Estimation For Ramp Merging Control In Autonomous Driving, Chiyu Dong, John M. Dolan, and Bakhtiar Litkouhi

Bayes Nets for Driver Intention

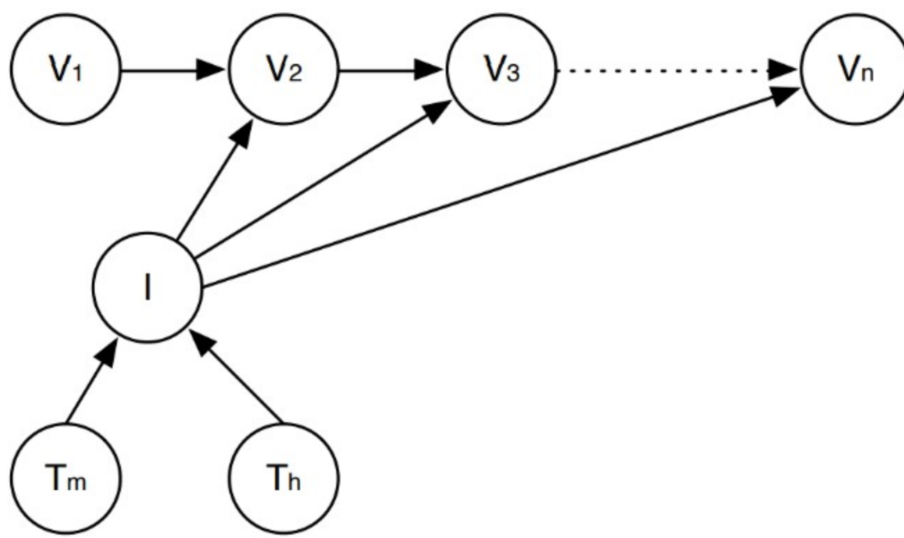
$$P(I \mid \mathbf{V}, T_m, T_h) = P(I, \mathbf{V}, T_m, T_h) / P(\mathbf{V}, T_m, T_h)$$

$$\propto P(\mathbf{V}, T_m, T_h \mid I)\, P(I)$$

$$P(\mathbf{V}, T_m, T_h \mid I) = P(\mathbf{V} \mid I) P(T_m, T_h \mid I)$$

$$P(\mathbf{V} \mid I) = P(V_1, V_2, \ldots, V_n \mid I)$$

$$= P(V_1) P(V_2 \mid V_1, I) \ldots P(V_n \mid V_{n-1}, I)$$



Who will find the error after this step in the paper?

Bayes Nets for Driver Intention

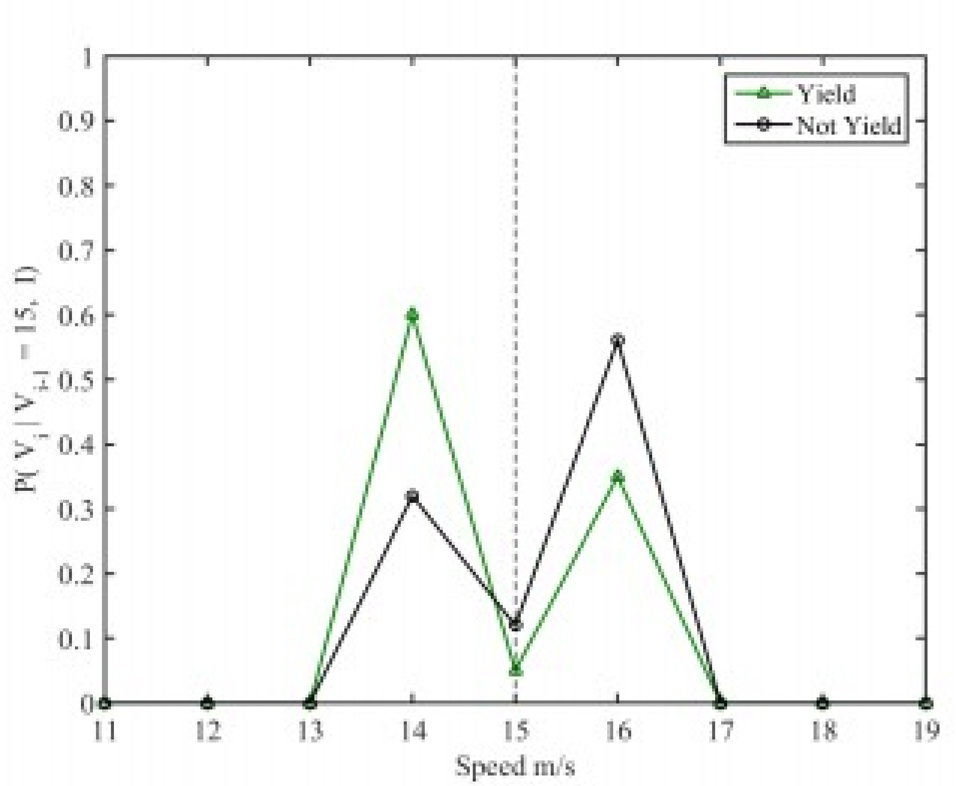


Fig. 4: Example of speed transition probability $P(V_t|V_{t-1}, I)$, which is learned from training data. The vertical dashed line is the previous speed; the x-axis indicates the current speed; Two colors indicate different intentions.

From: Intention Estimation For Ramp Merging Control In Autonomous Driving, Chiyu Dong, John M. Dolan, and Bakhtiar Litkouhi

Bayes nets for Driver Intention

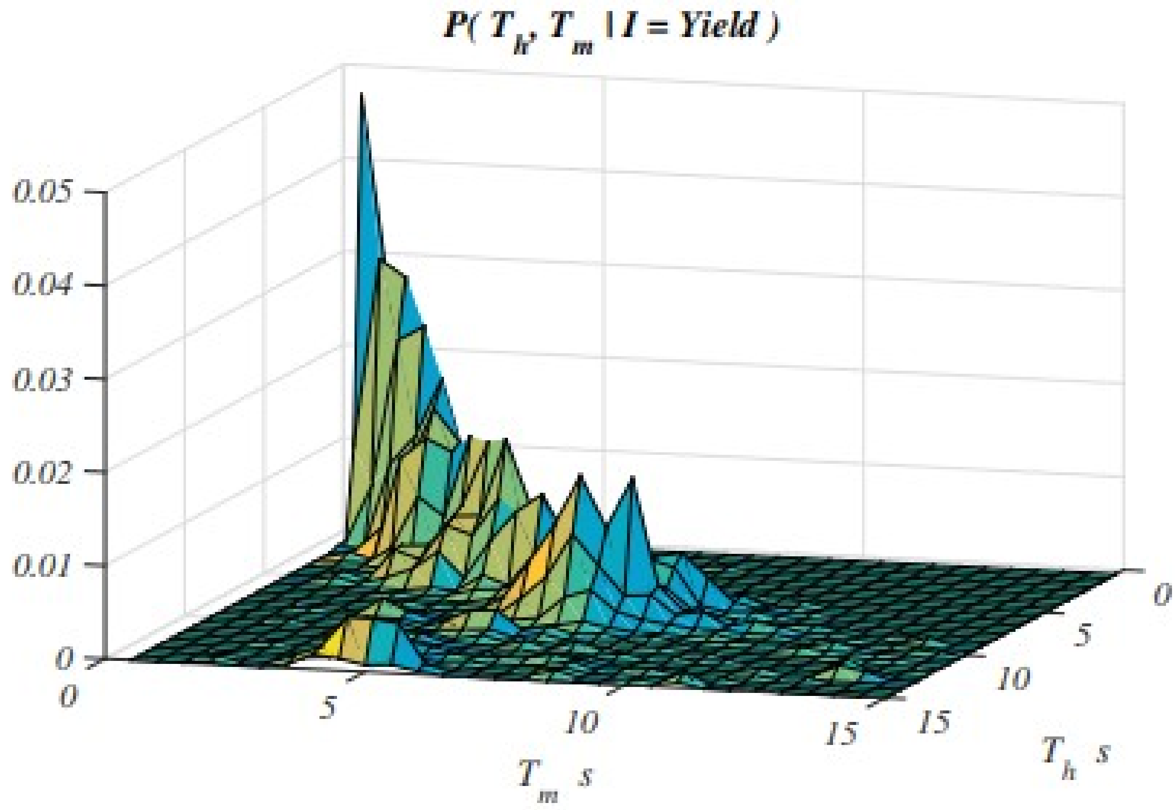


Fig. 5: $P(T_m, T_h | I = \text{Yield})$. Time-to-arrival transition probability distribution when the merge car yields the host.

From: Intention Estimation For Ramp Merging Control In Autonomous Driving, Chiyu Dong, John M. Dolan, and Bakhtiar Litkouhi