CPSC 540: Machine Learning

Directed Acyclic Graphical Models

Mark Schmidt

University of British Columbia

Winter 2016

Admin

- Assignment 3:
  - Due today, 1 late day to hand it in Thursday.
- Assignment 4:
  - Out, due in 2 weeks.
- Thursday;
  - Rich Sutton in DMP 110 at 3:30 (cancelling class):
    "The Future of Artificial Intelligence"
- Friday:
  - Julien Mairal in ICICS 146 at 5:00.
- Monday:
  - Monday: CVPR Area Chair Workshop;
    `http://cvpr2016.thecvf.com/events/ac_workshop`.

Course Roadmap

- Topic 1 was supervised learning: modeling $p(y^i|x^i)$.
  - Regresion models, change of basis, cross-validation, regularization/MAP.
  - Robust/logistic losses, structured sparisty, convex optimization, kernels, duality.

Course Roadmap

- Topic 1 was supervised learning: modeling $p(y^i|x^i)$.
  - Regresion models, change of basis, cross-validation, regularization/MAP.
  - Robust/logistic losses, structured sparisty, convex optimization, kernels, duality.
- Topic 2 has been density estimation: modeling $p(x^i)$.
  - Simple Bernoulli/Gaussian and product distributions.
  - Mixture models, EM, probbalistic latent-factor models.

Course Roadmap

- Topic 1 was supervised learning: modeling $p(y^i|x^i)$.
  - Regresion models, change of basis, cross-validation, regularization/MAP.
  - Robust/logistic losses, structured sparisty, convex optimization, kernels, duality.
- Topic 2 has been density estimation: modeling $p(x^i)$.
  - Simple Bernoulli/Gaussian and product distributions.
  - Mixture models, EM, probbalistic latent-factor models.
- Some things that should be bothering you:

Course Roadmap

- Topic 1 was supervised learning: modeling $p(y^i|x^i)$.
  - Regresion models, change of basis, cross-validation, regularization/MAP.
  - Robust/logistic losses, structured sparisty, convex optimization, kernels, duality.
- Topic 2 has been density estimation: modeling $p(x^i)$.
  - Simple Bernoulli/Gaussian and product distributions.
  - Mixture models, EM, probbalistic latent-factor models.
- Some things that should be bothering you:
  - How does conditional independence actually work?

Course Roadmap

- Topic 1 was supervised learning: modeling $p(y^i|x^i)$.
  - Regresion models, change of basis, cross-validation, regularization/MAP.
  - Robust/logistic losses, structured sparisty, convex optimization, kernels, duality.
- Topic 2 has been density estimation: modeling $p(x^i)$.
  - Simple Bernoulli/Gaussian and product distributions.
  - Mixture models, EM, probbalistic latent-factor models.
- Some things that should be bothering you:
  - How does conditional independence actually work?
  - Can we combine Topics 1 and 2?
  - Topic 3: Probabilistic graphical models.

Course Roadmap

- Topic 1 was supervised learning: modeling $p(y^i|x^i)$.
    - Regresion models, change of basis, cross-validation, regularization/MAP.
    - Robust/logistic losses, structured sparisty, convex optimization, kernels, duality.
- Topic 2 has been density estimation: modeling $p(x^i)$.
    - Simple Bernoulli/Gaussian and product distributions.
    - Mixture models, EM, probbalistic latent-factor models.
- Some things that should be bothering you:
    - How does conditional independence actually work?
    - Can we combine Topics 1 and 2?
    - Topic 3: Probabilistic graphical models.
    - Why aren't we learning about deep learning?
    - Topic 4: Deep learning.

Course Roadmap

- Topic 1 was supervised learning: modeling $p(y^i|x^i)$.
  - Regresion models, change of basis, cross-validation, regularization/MAP.
  - Robust/logistic losses, structured sparisty, convex optimization, kernels, duality.
- Topic 2 has been density estimation: modeling $p(x^i)$.
  - Simple Bernoulli/Gaussian and product distributions.
  - Mixture models, EM, probbalistic latent-factor models.
- Some things that should be bothering you:
  - How does conditional independence actually work?
  - Can we combine Topics 1 and 2?
  - Topic 3: Probabilistic graphical models.
  - Why aren't we learning about deep learning?
  - Topic 4: Deep learning.
  - Why do we optimize w, cross-validate λ, and integrate over z?

Course Roadmap

- Topic 1 was supervised learning: modeling $p(y^i|x^i)$.
  - Regresion models, change of basis, cross-validation, regularization/MAP.
  - Robust/logistic losses, structured sparisty, convex optimization, kernels, duality.
- Topic 2 has been density estimation: modeling $p(x^i)$.
  - Simple Bernoulli/Gaussian and product distributions.
  - Mixture models, EM, probbalistic latent-factor models.
- Some things that should be bothering you:
  - How does conditional independence actually work?
  - Can we combine Topics 1 and 2?
  - Topic 3: Probabilistic graphical models.
  - Why aren't we learning about deep learning?
  - Topic 4: Deep learning.
  - Why do we optimize w, cross-validate λ, and integrate over z?
  - We'll start clarifying this in topic 5, where we'll also start relaxing IID...

Independence of Random Variables

- Let A and B are random variables taking values $a \in \mathcal{A}$ and $b \in \mathcal{B}$.
- We say that A and B are independent if we have

$$p(a, b) = p(a)p(b),$$

  for all a and b.
- This is true iff $p(a, b) = f(a)g(b)$ for some functions f and g.

Independence of Random Variables

- Let A and B are random variables taking values $a \in \mathcal{A}$ and $b \in \mathcal{B}$.
- We say that A and B are independent if we have

$$p(a, b) = p(a)p(b),$$

  for all a and b.
- This is true iff $p(a, b) = f(a)g(b)$ for some functions f and g.
- Let's solve for $p(a)$,

$$p(a) = \frac{p(a, b)}{p(b)} = p(a|b).$$

- This gives us a more intuitive/useful definition: A and B are independent if

$$p(a|b) = p(a)$$

  for all a and b.

Independence of Random Variables

- Let A and B are random variables taking values $a \in \mathcal{A}$ and $b \in \mathcal{B}$.
- We say that A and B are independent if we have

$$p(a, b) = p(a)p(b),$$

  for all a and b.
- This is true iff $p(a, b) = f(a)g(b)$ for some functions f and g.
- Let's solve for $p(a)$,

$$p(a) = \frac{p(a, b)}{p(b)} = p(a|b).$$

- This gives us a more intuitive/useful definition: A and B are independent if

$$p(a|b) = p(a)$$

  for all a and b.
- By the same logic it's also equivalently to $p(b|a) = p(b)$.
- We sometimes write this as $A \perp B$.

Independence of Random Variables

- A and B are independent if $p(a|b) = p(a)$:
  - In words: knowing b tells us nothing about a (and vice versa).

Independence of Random Variables

- A and B are independent if $p(a|b) = p(a)$:
  - In words: knowing b tells us nothing about a (and vice versa).
- Example:
  - If we flip coin 1 then flip coin 2, the results of the two tosses are independent.
  - If we flip coin if coin 1 lands heads and otherwise flip coin 3, not independent.

Independence of Random Variables

- A and B are independent if $p(a|b) = p(a)$:
  - In words: knowing b tells us nothing about a (and vice versa).
- Example:
  - If we flip coin 1 then flip coin 2, the results of the two tosses are independent.
  - If we flip coin if coin 1 lands heads and otherwise flip coin 3, not independent.
- If we are talking about d variables x_j, we say they're mutually independent if

$$p(x_1, x_2, \dots, x_d) = \prod_{j=1}^{d} p(x_j), \text{ or } p(x_j | x_{-j}) = p(x_j) \text{ for all } j.$$

Independence of Random Variables

- A and B are independent if $p(a|b) = p(a)$:
  - In words: knowing b tells us nothing about a (and vice versa).
- Example:
  - If we flip coin 1 then flip coin 2, the results of the two tosses are independent.
  - If we flip coin if coin 1 lands heads and otherwise flip coin 3, not independent.
- If we are talking about d variables x_j, we say they're mutually independent if

$$p(x_1, x_2, \dots, x_d) = \prod_{j=1}^{d} p(x_j), \text{ or } p(x_j | x_{-j}) = p(x_j) \text{ for all } j.$$

  - In a product of Bernoullis model we have

$$p(x) = \prod_{j=1}^{n} p(x_j),$$

  so the x_j are independent and $p(x_j|x_{-j}) = p(x_j)$.

Independence of Random Variables

- A and B are independent if $p(a|b) = p(a)$:
  - In words: knowing b tells us nothing about a (and vice versa).
- Example:
  - If we flip coin 1 then flip coin 2, the results of the two tosses are independent.
  - If we flip coin if coin 1 lands heads and otherwise flip coin 3, not independent.
- If we are talking about d variables x_j, we say they're mutually independent if

$$p(x_1, x_2, \ldots, x_d) = \prod_{j=1}^{d} p(x_j), \text{ or } p(x_j | x_{-j}) = p(x_j) \text{ for all } j.$$

  - In a product of Bernoullis model we have

$$p(x) = \prod_{j=1}^{n} p(x_j),$$

    so the x_j are independent and $p(x_j|x_{-j}) = p(x_j)$.
  - In a mixture of (product of Bernoullis) the x_j are not independent:
    - Knowing x_{-j} can tell you something about x_j.

Conditional Independence

- We say that A is conditionally independent of B given C if

$$p(a, b|c) = p(a|c)p(b|c),$$

or equivalently we have

$$p(a|b, c) = p(a|c) \quad \left(\text{both equal } \frac{p(a, b|c)}{p(b|c)}\right).$$

Conditional Independence

- We say that A is conditionally independent of B given C if
$$p(a, b|c) = p(a|c)p(b|c),$$
or equivalently we have
$$p(a|b, c) = p(a|c) \quad \left(\text{both equal } \frac{p(a, b|c)}{p(b|c)}\right).$$
- If you know C, then *also* knowing B would tell you nothing about A.
- We often write this as $A \perp B \mid C$ or equivalently $B \perp A \mid C$.
- In a mixture of (product of Bernoullis) model
$$p(x) = \sum_{c=1}^{k} p(z = c) \prod_{j=1}^{d} p(x_j|z = c),$$
we have that $x_i \perp x_j \mid z$ (conditionally independent given the cluster)

Conditional Independence

- We say that A is conditionally independent of B given C if

$$p(a, b|c) = p(a|c)p(b|c),$$

or equivalently we have

$$p(a|b, c) = p(a|c) \quad \left(\text{both equal } \frac{p(a, b|c)}{p(b|c)}\right).$$

- If you know C, then *also* knowing B would tell you nothing about A.
- We often write this as $A \perp B \mid C$ or equivalently $B \perp A \mid C$.
- In a mixture of (product of Bernoullis) model

$$p(x) = \sum_{c=1}^{k} p(z = c) \prod_{j=1}^{d} p(x_j|z = c),$$

we have that $x_i \perp x_j \mid z$ (conditionally independent given the cluster)
- In particular, we can show that

$$p(x_i, x_j|z) = p(x_i|z)p(x_j|z) \text{ and } p(x_i|x_j, z) = p(x_i|z).$$

Outline

DAG Models

- Directed acyclic graphical (DAG) use product rule, $p(a, b, c) = p(b, c|a)p(a)$, to write

$$\begin{aligned} p(x_1, x_2, \ldots, x_d) &= p(x_1)p(x_2, x_3, \ldots, x_d|x_1) \\ &= p(x_1)p(x_2|x_1)p(x_3, x_4, \ldots, x_d|x_1, x_2) \\ &= p(x_1)p(x_2|x_1)p(x_3|x_2, x_1)p(x_4, x_5, \ldots, x_d|x_1, x_2, x_3) \end{aligned}$$

DAG Models

- Directed acyclic graphical (DAG) use product rule, $p(a, b, c) = p(b, c|a)p(a)$, to write

$$\begin{aligned} p(x_1, x_2, \dots, x_d) &= p(x_1)p(x_2, x_3, \dots, x_d|x_1) \\ &= p(x_1)p(x_2|x_1)p(x_3, x_4, \dots, x_d|x_1, x_2) \\ &= p(x_1)p(x_2|x_1)p(x_3|x_2, x_1)p(x_4, x_5, \dots, x_d|x_1, x_2, x_3) \end{aligned}$$

and so on until we get

$$p(x) = \prod_{j=1}^{d} p(x_j|x_{1:j-1}).$$

DAG Models

- Directed acyclic graphical (DAG) use product rule, $p(a,b,c) = p(b,c|a)p(a)$, to write

$$\begin{aligned} p(x_1, x_2, \ldots, x_d) &= p(x_1)p(x_2, x_3, \ldots, x_d|x_1) \\ &= p(x_1)p(x_2|x_1)p(x_3, x_4, \ldots, x_d|x_1, x_2) \\ &= p(x_1)p(x_2|x_1)p(x_3|x_2, x_1)p(x_4, x_5, \ldots, x_d|x_1, x_2, x_3) \end{aligned}$$

  and so on until we get

$$p(x) = \prod_{j=1}^{d} p(x_j|x_{1:j-1}).$$

- The above always holds, but it has too many parameters:
  - For binary x_i, we need 2^d parameters for $p(x_j|x_1, x_2, \ldots, x_{j-1})$ alone

DAG Models: Parsimonious Parameterization

- Directed acyclic graphical (DAG) models use product rule to write

$$p(x) = \prod_{j=1}^{d} p(x_j | x_{1:j-1}).$$

- Two main approaches for simplifying these d probability distributions.
- Approach 1: we can treat $p(x_j | x_{1:j-1})$ as supervised learning problem.

DAG Models: Parsimonious Parameterization

- Directed acyclic graphical (DAG) models use product rule to write

$$p(x) = \prod_{j=1}^{d} p(x_j | x_{1:j-1}).$$

- Two main approaches for simplifying these d probability distributions.
- Approach 1: we can treat $p(x_j | x_{1:j-1})$ as supervised learning problem.
    - The features are $x_{1:j-1}$ and the label is x_j.
    - If we use linear model, only need $(j-1)$ parameters.
    - We can apply our tricks from Topic 1 to Topic 2.
        - Nonlinear bases, robust/logistic losses, structured sparsity, kernels, etc.

DAG Models: Conditional Independence

- Directed acyclic graphical (DAG) models use product rule to write

$$p(x) = \prod_{j=1}^{d} p(x_j | x_{1:j-1}).$$

- Two main approaches for simplifying these d probability distributions.
- Approach 2: assume conditional independence to write

$$p(x_j | x_{1:j-1}) = p(x_j | x_{\mathsf{pa}(j)}),$$

where $\mathsf{pa}(j)$ are the parents of j.

DAG Models: Conditional Independence

- Directed acyclic graphical (DAG) models use product rule to write

$$p(x) = \prod_{j=1}^{d} p(x_j | x_{1:j-1}).$$

- Two main approaches for simplifying these d probability distributions.
- Approach 2: assume conditional independence to write

$$p(x_j | x_{1:j-1}) = p(x_j | x_{\text{pa}(j)}),$$

  where $\text{pa}(j)$ are the parents of j.
- Specifically, we assume that $x_j \perp x_{\text{np}(j)} | x_{\text{pa}(j)}$, where $\text{np}(j)$ are non-parents.
- In binary case, if we have k parents then only need 2^{k+1} parameters.
  - We can also combine both approaches: use regression on parents.

Special Cases of DAG Models

- We can write a lot of models as special cases DAG models,

$$p(x) = \prod_{j=1}^{d} p(x_j | x_{\mathsf{pa}(j)}).$$

  - Product of independent: if $\mathsf{pa}(j) = \varnothing$ then all variables are independent,

$$p(x) = \prod_{j=1}^{d} p(x_j).$$

Special Cases of DAG Models

- We can write a lot of models as special cases DAG models,

$$p(x) = \prod_{j=1}^{d} p(x_j | x_{\text{pa}(j)}).$$

  - Product of independent: if $\text{pa}(j) = \varnothing$ then all variables are independent,

$$p(x) = \prod_{j=1}^{d} p(x_j).$$

  - Markov chain: If $\text{pa}(j) = \{j-1\}$ then each j only depends on the previous value,

$$p(x) = p(x_1) \prod_{j=2}^{d} p(x_j | x_{j-1}).$$

Special Cases of DAG Models

- We can write a lot of models as special cases DAG models,

$$p(x) = \prod_{j=1}^{d} p(x_j | x_{\text{pa}(j)}).$$

  - Product of independent: if $\text{pa}(j) = \varnothing$ then all variables are independent,

$$p(x) = \prod_{j=1}^{d} p(x_j).$$

  - Markov chain: If $\text{pa}(j) = \{j-1\}$ then each j only depends on the previous value,

$$p(x) = p(x_1) \prod_{j=2}^{d} p(x_j | x_{j-1}).$$

  - Naive Bayes: Add an extra variable y with $\text{pa}(y) = \emptyset$ and $\text{pa}(x_j) = y$,

$$p(y, x) = p(y) \prod_{j=1}^{n} p(x_j | y).$$

Special Cases of DAG Models

- Instead of factorizing by variables j, could factor into blocks b:

$$p(x) = \prod_{b=1} p(x_b | x_{\mathsf{pa}(b)}).$$

Special Cases of DAG Models

- Instead of factorizing by variables j, could factor into blocks b:

$$p(x) = \prod_{b=1} p(x_b | x_{\text{pa}(b)}).$$

  - Generative models (Classification using Topic 2): $\text{pa}(y) = \emptyset$ and $\text{pa}(x) = y$,

$$p(y, x) = p(y)p(x|y).$$

Special Cases of DAG Models

- Instead of factorizing by variables j, could factor into blocks b:

$$p(x) = \prod_{b=1} p(x_b | x_{\text{pa}(b)}).$$

  - Generative models (Classification using Topic 2): $\text{pa}(y) = \emptyset$ and $\text{pa}(x) = y$,

$$p(y, x) = p(y)p(x|y).$$

  - Discriminative models (Classification using topic 1): $\text{pa}(y) = x$ and $\text{pa}(x) = \emptyset$,

$$p(y, x) = p(y|x)p(x).$$

Special Cases of DAG Models

- Instead of factorizing by variables j, could factor into blocks b:

$$p(x) = \prod_{b=1} p(x_b | x_{\text{pa}(b)}).$$

  - Generative models (Classification using Topic 2): $\text{pa}(y) = \emptyset$ and $\text{pa}(x) = y$,

$$p(y,x) = p(y)p(x|y).$$

  - Discriminative models (Classification using topic 1): $\text{pa}(y) = x$ and $\text{pa}(x) = \emptyset$,

$$p(y,x) = p(y|x)p(x).$$

  - Mixture models: $\text{pa}(z) = \emptyset$ and $\text{pa}(x) = z$,

$$p(x,z) = p(z)p(x|z).$$

From Probability Factorizations to Graphs

- DAG models are also known as “Bayesian networks” and “belief networks”.
- Called graphical because we can visualize independence assumptions as a graph:
  - We have a vertex for each variable j (or block b).

From Probability Factorizations to Graphs

- DAG models are also known as "Bayesian networks" and "belief networks".
- Called graphical because we can visualize independence assumptions as a graph:
  - We have a vertex for each variable j (or block b).
  - We place an edge from i to j if i is a parent of j.
  - By construction, the graph will be acyclic.

From Probability Factorizations to Graphs

- DAG models are also known as "Bayesian networks" and "belief networks".
- Called graphical because we can visualize independence assumptions as a graph:
  - We have a vertex for each variable j (or block b).
  - We place an edge from i to j if i is a parent of j.
  - By construction, the graph will be acyclic.
- Two interesting properties of the structure of this graph:
  1. Can be used to test conditional independence between arbitrary sets.
  2. Nice structures allow efficient calculation using dynamic programming.

Graph Structure Examples

With product of independent we have

$$p(x) = \prod_{j=1}^{d} p(x_j).$$

The corresponding graph structure is:

Graph Structure Examples

With Markov chain we have

$$p(x) = p(x_1) \prod_{j=2}^{d} p(x_j | x_{j-1}).$$

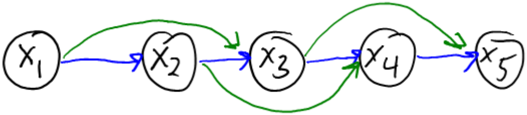
The corresponding graph structure is:

Graph Structure Examples

With second-order Markov chain we have

$$p(x) = p(x_1)p(x_2|x_1)\prod_{j=3}^{d} p(x_j|x_{j-1}, x_{j-2}).$$

The corresponding graph structure is:

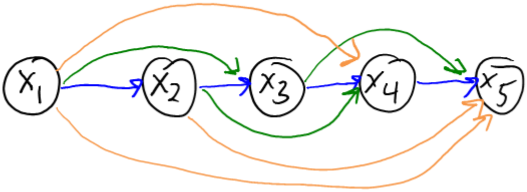


Graph Structure Examples

With general distribution we have

$$p(x) = \prod_{j=1}^{d} p(x_j | x_{1:j-1}).$$

The corresponding graph structure is:

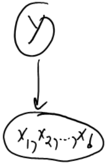


Graph Structure Examples

With Gaussian generative classifier we have

$$p(y, x) = p(y)p(x|y).$$

The corresponding graph structure is:

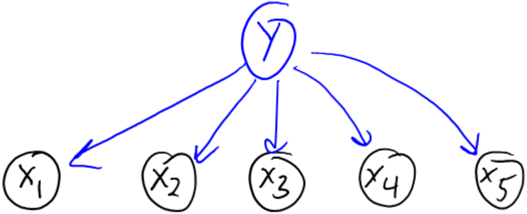


Graph Structure Examples

With naive Bayes or diagonal Gaussian generative classifier we have

$$p(y, x) = p(y) \prod_{j=1}^{d} p(x_j | y).$$

The corresponding graph structure is:

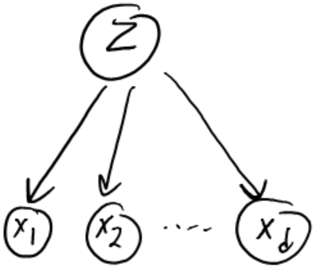


Graph Structure Examples

With mixture of independent we have

$$p(z, x) = p(z) \prod_{j=1}^{d} p(x_j | z).$$

The corresponding graph structure is:

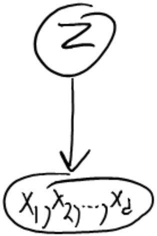


Graph Structure Examples

With mixture of Gaussian we have

$$p(z, x) = p(z)p(x|z).$$

The corresponding graph structure is:

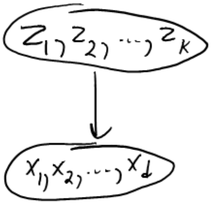


Graph Structure Examples

With probabilistic PCA we have

$$p(z, x) = p(z)p(x|z).$$

The corresponding graph structure is:

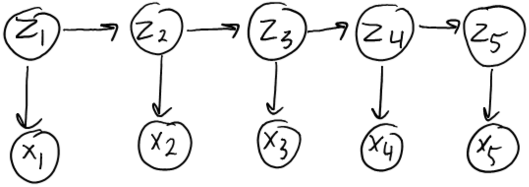


Graph Structure Examples

With hidden Markov models we have

$$p(z, x) = p(z_1) \left(\prod_{j=2}^{d} p(z_j | z_{j-1}) \right) \left(\prod_{j=1}^{n} p(x_j | z_j) \right).$$

The corresponding graph structure is:

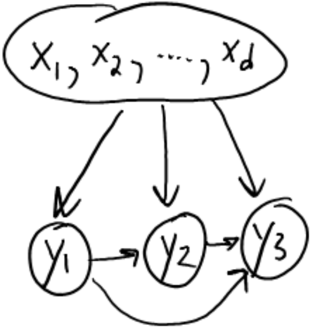


Graph Structure Examples

We can do multi-output regression/classification via conditional DAGs,

$$p(y, x) = p(x) \prod_{c=1}^{k} p(y_c | y_{\text{pa}(c)}, x)$$

The corresponding graph structure is:

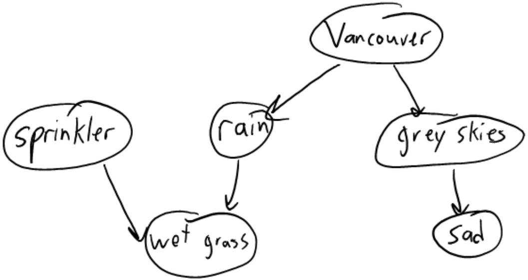


Graph Structure Examples

We can consider less-structured examples,

$$p(S, V, R, W, G, D) = p(S)p(V)p(R|V)p(W|S, R)p(G|V)p(D|G).$$

The corresponding graph structure is:

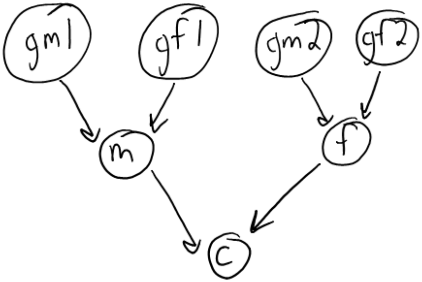


Graph Structure Examples

We can consider phylogeny (family trees):

$$
\begin{aligned}
&p(gm1, gf1, gm2, gf2, m, f, c) \\
&= p(gm1)p(gf1)p(gm2)p(gf2)p(m|gm1, gf1)p(f|gm2, gf2)p(c|m, f).
\end{aligned}
$$

The corresponding graph structure is:



Outline

D-Separation: From Graphs to Conditional Independence

- The graph represents conditional independence implied by factorization.
- Can we use the graph to test generic conditional independence statements?
  - Yes, variables are independent if all paths are block by d-separation.
- The rules are best illustrated by example...

D-Separation Case 0 (No Paths and Direct Links)

Are genes for eye colour in person x independent of these genes in person y?

D-Separation Case 0 (No Paths and Direct Links)

Are genes for eye colour in person x independent of these genes in person y?

- No path: x and y are not related (independent),

We have $x \perp y$: there are no paths to be blocked.

D-Separation Case 0 (No Paths and Direct Links)

Are genes for eye colour in person x independent of these genes in person y?

- No path: x and y are not related (independent),

  We have $x \perp y$: there are no paths to be blocked.

- Direct link: x is the parent of y,

  We have $x \not\perp y$: knowing x tells you about y (direct paths aren't blockable).

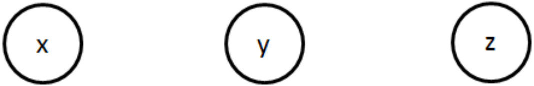
D-Separation Case 0 (No Paths and Direct Links)

Neither case changes if we have a third independent person z:

D-Separation Case 0 (No Paths and Direct Links)

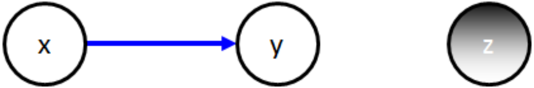
Neither case changes if we have a third independent person z:

- No path: If x and y are independent,



  We have $x \perp y$: adding z doesn't make a path.

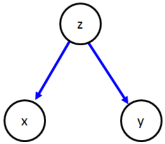
- Direct link: x is the parent of y,



  We have $x \not\perp y$: adding z doesn't block path.

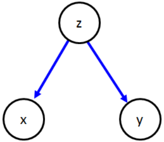
D-Separation Case 1: Common Parent

- Case 1: x and y are sibilings.
  - If z is a common unobserved parent:



D-Separation Case 1: Common Parent

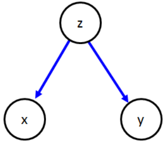
- Case 1: x and y are sibilings.
  - If z is a common unobserved parent:



We now have $x \not\perp y$: knowing x would give information about y.

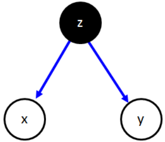
D-Separation Case 1: Common Parent

- Case 1: x and y are sibilings.
  - If z is a common unobserved parent:



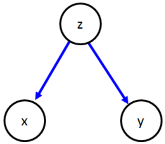
We now have $x \not\perp y$: knowing x would give information about y.

  - But if z is *observed*:

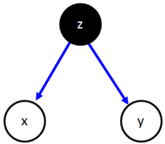


D-Separation Case 1: Common Parent

- Case 1: x and y are sibilings.
  - If z is a common unobserved parent:



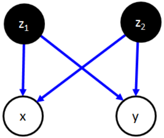
    We now have $x \not\perp y$: knowing x would give information about y.
  - But if z is *observed*:



    In this case $x \perp y \mid z$: knowing z "breaks" dependence between x and y.

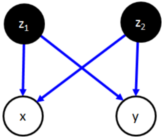
D-Separation Case 1: Common Parent

- Case 1: x and y are sibilings.
  - If z_1 and z_2 are common observed parents:



D-Separation Case 1: Common Parent

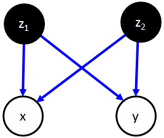
- Case 1: x and y are sibilings.
  - If z_1 and z_2 are common observed parents:



We have $x \perp y \mid z_1, z_2$: knowing z_1 and z_2 breaks dependence between x and y.

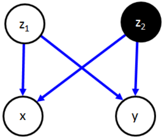
D-Separation Case 1: Common Parent

- Case 1: x and y are sibilings.
  - If z_1 and z_2 are common observed parents:



We have $x \perp y \mid z_1, z_2$: knowing z_1 and z_2 breaks dependence between x and y.

  - But if we only observe z_2:



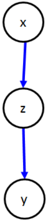
Then we have $x \not\perp y \mid z_2$: dependence still "flows" through z_1.

D-Separation Case 2: Chain

- Case 2: x is the grandmother of y.

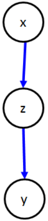
D-Separation Case 2: Chain

- Case 2: x is the grandmother of y.
  - If z is the mother we have:



D-Separation Case 2: Chain

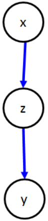
- Case 2: x is the grandmother of y.
  - If z is the mother we have:



We have $x \not\perp y$: knowing x would give information about y because of z

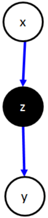
D-Separation Case 2: Chain

- Case 2: x is the grandmother of y.
  - If z is the mother we have:



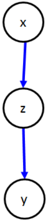
We have $x \not\perp y$: knowing x would give information about y because of z

  - But if z is *observed*:



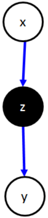
D-Separation Case 2: Chain

- Case 2: x is the grandmother of y.
  - If z is the mother we have:



We have $x \not\perp y$: knowing x would give information about y because of z

  - But if z is *observed*:



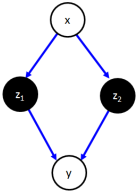
In this case $x \perp y \mid z$: knowing z "breaks" dependence between x and y.

D-Separation Case 2: Chain

- Consider weird case where parents z_1 and z_2 share mother x:

D-Separation Case 2: Chain

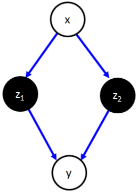
- Consider weird case where parents z_1 and z_2 share mother x:
  - If z_1 and z_2 are observed we have:



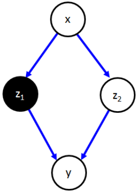
We have $x \perp y \mid z_1, z_2$: knowing both parents breaks dependency.

D-Separation Case 2: Chain

- Consider weird case where parents z_1 and z_2 share mother x:
  - If z_1 and z_2 are observed we have:



    We have $x \perp y \mid z_1, z_2$: knowing both parents breaks dependency.
  - But if only z_1 is *observed*:



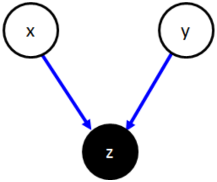
    We have $x \not\perp y \mid z_1$: dependence still "flows" through z_2.

D-Separation Case 3: Common Child

- Case 3: x and y share a child z:

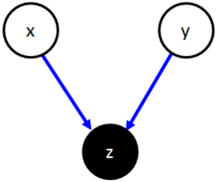
D-Separation Case 3: Common Child

- Case 3: x and y share a child z:
  - If we observe z then we have:



D-Separation Case 3: Common Child

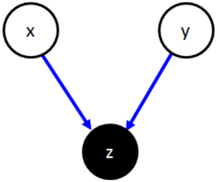
- Case 3: x and y share a child z:
  - If we observe z then we have:



We have $x \not\perp y \mid z$: if we know z, then knowing x gives us information about y.

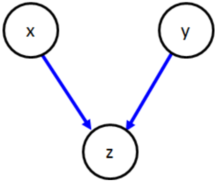
D-Separation Case 3: Common Child

- Case 3: x and y share a child z:
  - If we observe z then we have:



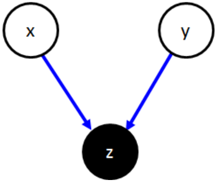
We have $x \not\perp y \mid z$: if we know z, then knowing x gives us information about y.

  - But if z is not observed:

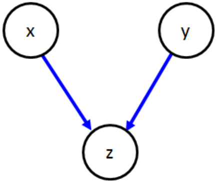


D-Separation Case 3: Common Child

- Case 3: x and y share a child z:
  - If we observe z then we have:



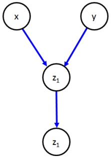
    We have $x \not\perp y \mid z$: if we know z, then knowing x gives us information about y.
  - But if z is not observed:



    We have $x \perp y$: if you don't observe z then x and y are independent.
- Different from Case 1 and Case 2: not observing the child blocks path.

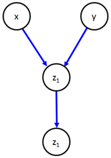
D-Separation Case 3: Common Child

- Case 3: x and y share a child z_1:
  - If there exists an unobserved grandchild z_2:



D-Separation Case 3: Common Child

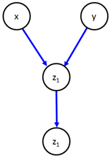
- Case 3: x and y share a child z_1:
  - If there exists an unobserved grandchild z_2:



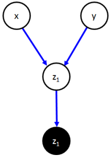
We have $x \perp y$: the path is still blocked by not knowing z_1 or z_2.

D-Separation Case 3: Common Child

- Case 3: x and y share a child z_1:
  - If there exists an unobserved grandchild z_2:

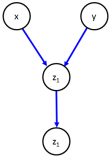


    We have $x \perp y$: the path is still blocked by not knowing z_1 or z_2.
  - But if z_2 is observed:

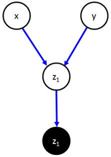


D-Separation Case 3: Common Child

- Case 3: x and y share a child z_1:
  - If there exists an unobserved grandchild z_2:



    We have $x \perp y$: the path is still blocked by not knowing z_1 or z_2.
  - But if z_2 is observed:



    We have $x \not\perp y \mid z_2$: grandchild creates dependence even with unobserved parent.
- Case 3 needs to consider descendants of child.

D-Separation

- We say that A and B are d-separated given E if for *all paths* P from A to B, *at least one* of the following holds:

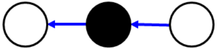
D-Separation

- We say that A and B are d-separated given E if for *all paths* P from A to B, *at least one* of the following holds:
  1. P includes a "fork" with an observe parent node:

D-Separation

- We say that A and B are d-separated given E if for *all paths* P from A to B, *at least one* of the following holds:
  1. P includes a "fork" with an observe parent node:

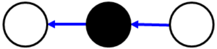
  2. P includes a "chain" with an observed middle node:



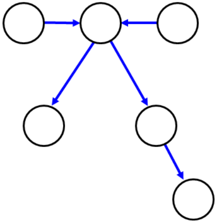
D-Separation

- We say that A and B are d-separated given E if for *all paths* P from A to B, *at least one* of the following holds:
  1. P includes a "fork" with an observe parent node:

  2. P includes a "chain" with an observed middle node:



  3. P includes a "collider":



where C and all its descendants are unobserved.

Alarm Example

Alarm Example

- Earthquake $\not\perp$ Call.

Alarm Example

- Earthquake $\not\perp$ Call.
- Earthquake $\perp$ Call $|$ Alarm.

Alarm Example

- Earthquake $\not\perp$ Call.
- Earthquake $\perp$ Call $|$ Alarm.
- Alarm $\not\perp$ Stuff Missing.

Alarm Example

- Earthquake $\not\perp$ Call.
- Earthquake $\perp$ Call $\mid$ Alarm.
- Alarm $\not\perp$ Stuff Missing.
- Alarm $\perp$ Stuff Missing $\mid$ Burglary.

Alarm Example

Alarm Example

- Earthquake $\perp$ Burglary.

Alarm Example

- Earthquake $\perp$ Burglary.
- Earthquake $\not\perp$ Burglary $|$ Alarm.
  - Explaining away: Knowing Earthquake would make Burglary is less likely.

Alarm Example

- Earthquake $\perp$ Burglary.
- Earthquake $\not\perp$ Burglary | Alarm.
  - Explaining away: Knowing Earthquake would make Burglary is less likely.
- Call $\not\perp$ Stuff Missing.

Alarm Example

- Earthquake $\perp$ Burglary.
- Earthquake $\not\perp$ Burglary | Alarm.
  - Explaining away: Knowing Earthquake would make Burglary is less likely.
- Call $\not\perp$ Stuff Missing.
- Earthquake $\perp$ Stuff Missing.

Alarm Example

- Earthquake $\perp$ Burglary.
- Earthquake $\not\perp$ Burglary | Alarm.
  - Explaining away: Knowing Earthquake would make Burglary is less likely.
- Call $\not\perp$ Stuff Missing.
- Earthquake $\perp$ Stuff Missing.
- Earthquake $\not\perp$ Stuff Missing | Call.

Outline

Discussion of D-Separation

- D-separation lets you say if conditional independence is implied by factorization:

$$(A \text{ and } B \text{ are d-separated given } E) \Rightarrow A \perp B \mid E.$$

Discussion of D-Separation

- D-separation lets you say if conditional independence is implied by factorization:

$$(A \text{ and } B \text{ are d-separated given } E) \Rightarrow A \perp B \mid E.$$

- However, there might be extra conditional independences in the distribution:
  - These would depend on specific choices of the $p(x_j|x_{\text{pa}(j)})$.
  - Or some orderings may to non-equivalent graphs.

Discussion of D-Separation

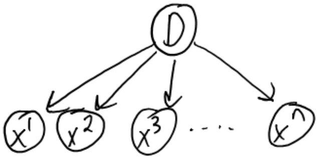
- D-separation lets you say if conditional independence is implied by factorization:

$$(A \text{ and } B \text{ are d-separated given } E) \Rightarrow A \perp B \mid E.$$

- However, there might be extra conditional independences in the distribution:
  - These would depend on specific choices of the $p(x_j | x_{\text{pa}(j)})$.
  - Or some orderings may to non-equivalent graphs.
- Nevertheless, we can do a lot with d-separation:
  - Implies every instance of independence/conditional-independence/IID we've used.

IID Assumption in DAG and Plate Notation

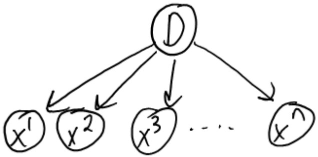
- Graphical representation of the IID assumption:



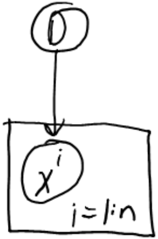
- Test samples from D would be related to training x^i because D is unobserved:
  - With this understanding we can start to relax IID assumption.

IID Assumption in DAG and Plate Notation

- Graphical representation of the IID assumption:

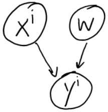


- Test samples from D would be related to training x^i because D is unobserved:
  - With this understanding we can start to relax IID assumption.
- We can concisely represent repeated parts of graphs using plate notation:



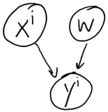
Tilde Notation in DAG and Plate Notation

- When we write
$$y^i \sim \mathcal{N}(w^T x^i, 1),$$
we can interpret it as the DAG model:

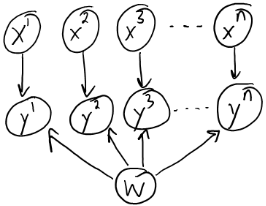


Tilde Notation in DAG and Plate Notation

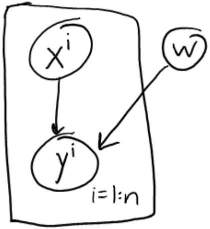
- When we write
$$y^i \sim \mathcal{N}(w^T x^i, 1),$$
we can interpret it as the DAG model:



- If the x^i are IID then we can represent supervised learning as



or



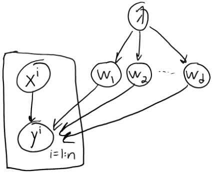
- From d-separation on this graph we have $p(y|X, w) = \prod_{i=1}^{n} p(y^i|x^i, w)$.

Tilde Notation in DAG and Plate Notation

- When we do MAP estimation under the assumptions

$$y^i \sim \mathcal{N}(w^T x^i, 1), \quad w_j \sim \mathcal{N}(0, 1/\lambda),$$

we can interpret it as the DAG model:

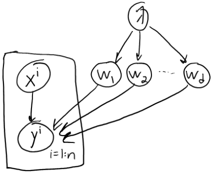


Tilde Notation in DAG and Plate Notation

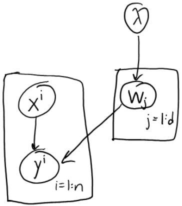
- When we do MAP estimation under the assumptions

$$y^i \sim \mathcal{N}(w^T x^i, 1), \quad w_j \sim \mathcal{N}(0, 1/\lambda),$$

  we can interpret it as the DAG model:



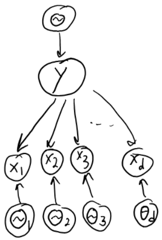
- Or introducing a second plate using:



Other Models in DAG/Plate Notation

- For naive Bayes or Gaussian discriminant analysis with diagonal Σ_c we have

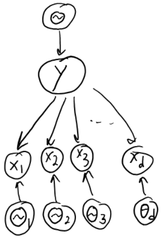
$$y^i \sim \mathrm{Cat}(\theta), \quad x^i | y^i = c \sim D(\theta_c).$$



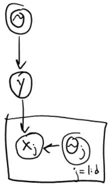
Other Models in DAG/Plate Notation

- For naive Bayes or Gaussian discriminant analysis with diagonal Σ_c we have

$$y^i \sim \mathsf{Cat}(\theta), \quad x^i | y^i = c \sim D(\theta_c).$$



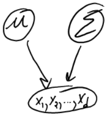
- Or in plate notation as



Other Models in DAG/Plate Notation

- In a full Gaussian model for a single x we have

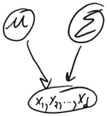
$$x^i \sim \mathcal{N}(\mu, \Sigma).$$



Other Models in DAG/Plate Notation

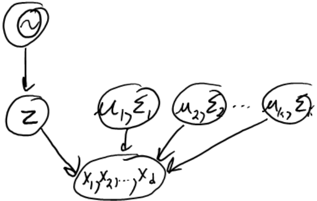
- In a full Gaussian model for a single x we have

$$x^i \sim \mathcal{N}(\mu, \Sigma).$$

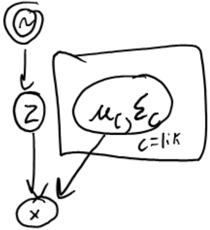


- For mixture of Gaussians we have

$$z^i \sim \text{Cat}(\theta), \quad x^i | z^i = c \sim \mathcal{N}(\mu_c, \Sigma_c).$$



or



Summary

- Conditional independence of A and B given C:
    - Knowing B tells us nothing about A if we already know C.

Summary

- Conditional independence of A and B given C:
  - Knowing B tells us nothing about A if we already know C.
- DAG models factorize joint distribution into product of conditionals.
  - Assume conditionals are regression models or depend on small number "parents".
  - Joint distribution of models we've discussed can be written as DAG models.

Summary

- Conditional independence of A and B given C:
  - Knowing B tells us nothing about A if we already know C.
- DAG models factorize joint distribution into product of conditionals.
  - Assume conditionals are regression models or depend on small number "parents".
  - Joint distribution of models we've discussed can be written as DAG models.
- D-separation allows us to test conditional independences based on graph.

Summary

- Conditional independence of A and B given C:
  - Knowing B tells us nothing about A if we already know C.
- DAG models factorize joint distribution into product of conditionals.
  - Assume conditionals are regression models or depend on small number "parents".
  - Joint distribution of models we've discussed can be written as DAG models.
- D-separation allows us to test conditional independences based on graph.
- Plate Notation lets compactly draw graphs with repeated patterns.
  - There are fancier versions of plate notation called "probabilistic programming".

- Next time: undirected graphical models and how we use graphical models.