PGMs for Deep Learning: Representations

Sargur N. Srihari
srihari@cedar.buffalo.edu

Role of PGMs in Deep Learning

- Deep learning draws upon many modeling formalisms to guide design efforts and describe algorithms
- Structured PGMs are a key ingredient of the most important topics in deep learning

What are structured probabilistic models?

- A way of describing probability distributions using a graph to describe which variables interact with each other directly
- Graph is used in the sense of graph theory
  - Vertices connected to one another by edges
- Because structure is described by a graph they are called *graphical models* or *PGMs*

Treatment of PGMs in Deep Learning

- Deep learning practitioners tend to use very different model structures, learning algorithms and inference procedures than rest of PGM community

Plan of PGM Discussion

1. Challenges of building large scale probability models
2. How to use a graph to describe the structure of a probability distribution
   - It has its own complications
     - E.g., which variables need to interact directly?
   - Discuss two approaches to overcome difficulty
     - By learning about dependencies
3. Unique emphasis placed by deep learning researchers to specific approaches

Challenge of Unstructured Modeling

- To scale machine learning to solve AI problems requires ability to understand high-dimensional data with rich structure
  - E.g., understand natural images, documents containing multiple words and punctuations
- The classification task of machine learning is a limited goal

Classification is a limited goal

- Classification algorithms take input from a rich high-dimensional distribution and summarize it with a categorical label
  - What object is in the photo, What word is spoken in a recording, What topic a document is about
- Classification discards most of the input
  - E.g., ignore background when recognizing object
- Produces a single output
  - Or a distribution over values of that single output

Probabilistic models for other tasks

- More expensive than classification
- Require producing many output values
- Require complete understanding of structure of entire input without ignoring sections of it
- Some tasks are:
  1. Density estimation
  2. Denoising
  3. Missing value imputation
  4. Sampling
- Each of these four tasks are described next

1. The Density Estimation Task

- From a given input $\boldsymbol{x}$, return an estimate of $p(\boldsymbol{x})$, under the data generating distribution
- Although it requires only a single output, it requires a complete understanding of the entire input
- Even if one element of vector $\boldsymbol{x}$ is unusual, the system must assign to it a low probability

2. The Denoising Task

- Given a damaged or incorrectly observed $\boldsymbol{x}'$, return an estimate of the correct $\boldsymbol{x}$
  - E.g., image may have dust or scratches
- This requires
  - Multiple outputs
    - Every element of the clean output
  - An understanding of the entire input
    - If even a part remains damaged the output is damaged

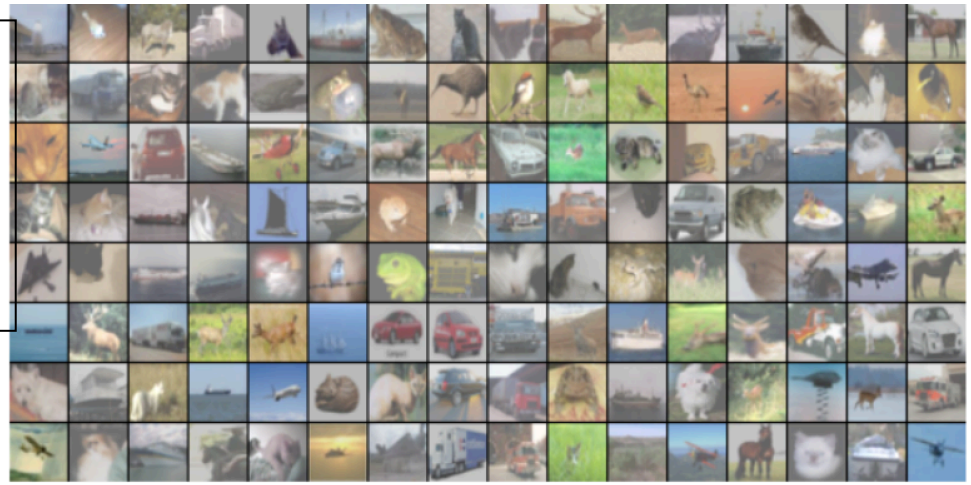
3. Missing value imputation task

- Given observations of some elements of $\boldsymbol{x}$ return estimates of unobserved parts of $\boldsymbol{x}$
  - Or a probability distribution over those parts
- ML algorithm must understand entire input to be able to restore any elements of $\boldsymbol{x}$

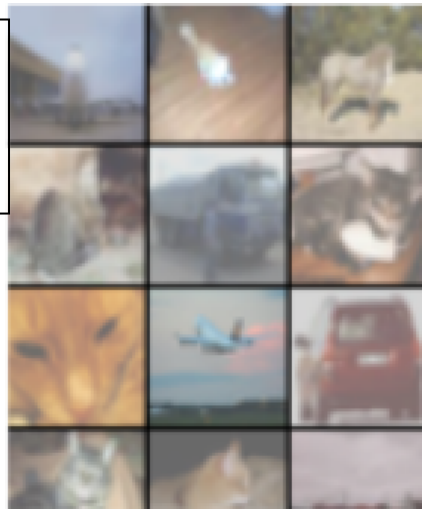
4. The sampling task

- Generate new samples from a distribution $p(\boldsymbol{x})$
  - E.g., probabilistic modeling of natural images

$p(\boldsymbol{x})$:color images from CIFAR data set (32x32 pixels)



Some enlarged images



Samples generated by PGM trained on this CIFAR data set

Synthesized mages displayed in same order as those that are closest in input set

Model is truly synthesizing new images rather than memorize the training data

Intractability of Rich Distributions

- Modeling a distribution over 1000s of variables is challenging computationally/ statistically
  - Consider 32x32x3 binary image
    - There are 2^{3072} possible images
    - (10^{800} larger than no. of atoms in universe)
- If we have n discrete variables with k possible values each, naiive approach of storing a table of k^n values is infeasible for several reasons described next

Intractability of Naiive method

- *Memory*: cost of storing the representation
  - Unless n and k are small, table would be too large to store
- *Statistical Efficiency*: As the number of parameters increases, so does the amount of training data needed

Runtime intractability of naiive method

- Runtime: Cost of inference
  - We have a model of $p(\boldsymbol{x})$ and need to infer $p(x_1)$ and $p(x_2|x_1)$
  - Requires summing across the entire table
- Runtime: Cost of sampling
  - Naiive way is to draw a sample from $u \sim \mathrm{U}(0,1)$
  - Iterate through the table adding probabilities until they exceed u and return the outcome corresponding to that position in the table
  - Requires reading through entire table--same cost

Full Table approach is overkill

- Table-based approach models every possible interaction between variables
- Probability distributions in real tasks are much simpler than this
- Usually variables influence each other only indirectly

Direct and indirect interaction

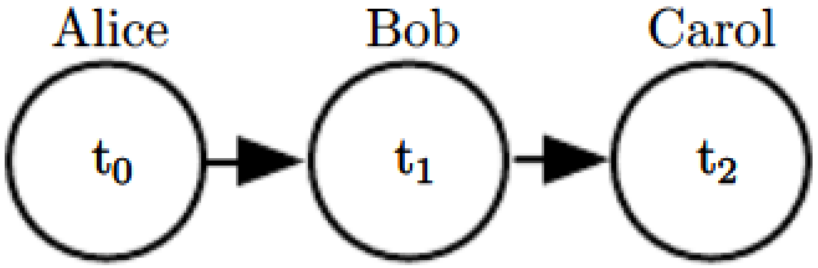
- Consider modeling finishing times in a relay
- Team has three runners: Alice, Bob, Carol
  - Alice hands baton to Bob, Bob hands to Carol who finishes the lap
    - Alice's finishing time does not depend on anyone else, Bob's finishing time depends on Alice's, Carol's depends on Bob's
    - Carol's finishing time depends only indirectly on Alice's
    - If we already know Bob's finishing time, we will not be able to better estimate Carol's finishing time by finding out what Alice's finishing time was

Using graphs to describe model structure

- Each node represents a random variable
- Each edge represents a direct interaction
  - These direct interactions imply other indirect interactions
  - But only direct interactions need be represented
- Graphical models can be largely divided into two categories
  - Models based on directed acyclic graphs
  - Models based on undirected graphs

Directed Graphical Model

- Also called belief network or Bayesian network
- Relay race example



- Bob's finishing time t_1 depends on Alice's finishing time t_0, Carol's finishing time t_2 depends on Bob's finishing time t_1

Meaning of directed edges

- Drawing an arrow from a to b means we define a conditional probability distribution (CPD) over b via a conditional distribution with a as one of the variables on the right side of the conditional bar
  - i.e., distribution over b depends on the value of a

Formal directed graphical model

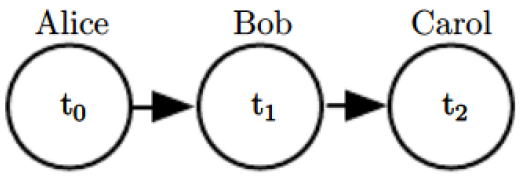
- Defined on variables $\mathbf{x}$ is defined by a directed graphical acyclic graph G
  - whose vertices are random variables in the model and a set of local CPDs

    $$p(x_i|Pa_G(x_i))$$

    - where $Pa_G(x_i)$ gives the parents of x_i in G
- The probability distribution over $\mathbf{x}$ is given by

$$p(\mathbf{x}) = \prod_i p(x_i \mid Pa_G(x_i))$$

- In the relay race example
  - $p(t_0,t_1,t_2)=p(t_0)p(t_1|t_0)p(t_2|t_1)$



Savings achieved by directed model

- If t_0, t_1 and t_2 are discrete with 100 values then single table would require $999{,}999$ values
  - By making tables for only conditional probabilities we need only $18{,}999$ values
- To model n discrete variables each having k values, cost of a single table is $O(k^n)$
- If m is maximum no. of variables appearing on either side of conditioning bar in a single CPD then cost of tables for directed PGM is $O(k^m)$
  - So long as each variable has few parents in graph, distribution represented by few parameters

Undirected models

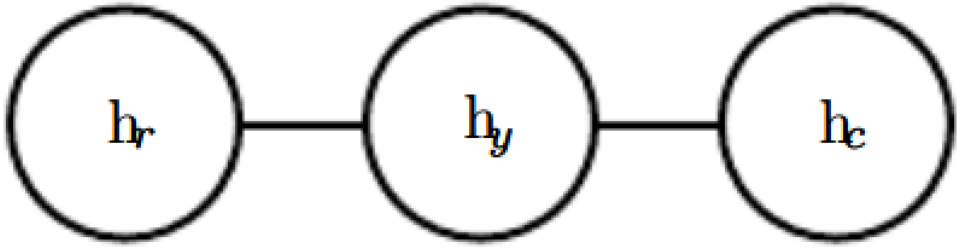
- Also known as Markov Random Fields or Markov Networks
- Use graphs whose edges are undirected
- Directed models have clear directionality as in the case of the runners
- When interactions have no clear directionality, more appropriate to use an undirected graph

Ex: Undirected Model for Health

- A model over three binary variables:
  - Whether or not you are sick, h_y
  - Whether or not your coworker is sick, h_c
  - Whether or not your roommate is sick, h_r
- Assuming coworker and roommate do not know each other, very unlikely one of them will give a cold to the other
  - Event is so rare we do not model it
- There is no clear directionality either
- This motivates using an undirected model

The health undirected graph

- You and your rommmate may infect each other with a cold
- You and your work colleague may do the same
- Assuming room-mate and colleague do not know each other they can only get infected through you



Undirected graph definition

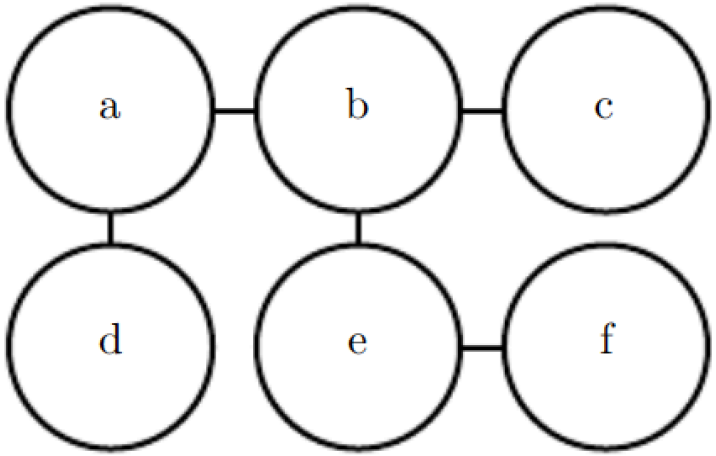
- If two variables directly interact with each other then the nodes are connected
- Edge has no arrow and has no CPD
- An undirected PGM is defined on a graph G
  - For each clique C in the graph, a factor $\phi(C)$, or clique potential, measures the affinity for being in each of their joint states
    - A clique is a subset of nodes all connected to each other
  - Together they define an unnormalized distribution

$$\tilde{p}(\mathbf{x}) = \prod_{C \in G} \phi(C)$$

Efficiency of Unnormalized Distribution

- Efficient to work with so long as the cliques are small
- It encodes that states with higher affinity $\phi(C)$ are more likely
- Since there is little structure to the definition of the cliques, there is no guarantee that multiplying them together will yield a probability distribution

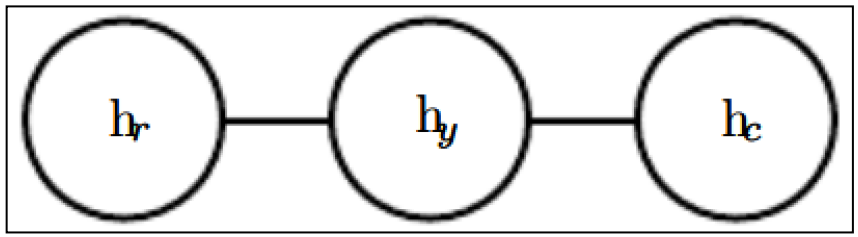
Reading factorization information from an undirected graph



- This graph implies that

$$p(a,b,c,d,e,f) =$$
$$\phi_{a,b}(a,b) + \phi_{b,c}(b,c) + \phi_{a,d}(a,d) + \phi_{b,e}(b,e) + \phi_{e,f}(e,f)$$

Ex: Clique potential



- In this example, one clique is between h_y and h_c
- Factor for this clique can be defined by a table

	$h_y = 0$	$h_y = 1$
$h_c = 0$	2	1
$h_c = 1$	1	10

State of 1 indicates healthy
Both are usually healthy, thus highest affinity
State of both being sick has higher afinity than one being sick

- Similar factor is needed for the other clique

The partition function

- The unnormalized probability distribution
  - Is guaranteed to be non-negative everywhere
  - It is not guranted to sum or integrate to 1
- To obtain a valid probability distribution we must use the normalized (or Gibbs) distribution

$$p(\mathbf{x}) = \frac{1}{Z}\hat{p}(\mathbf{x}) \qquad \tilde{p}(\mathbf{x}) = \prod_{C \in G} \phi(C)$$

  - Where Z causes the distribution to sum to one

$$Z = \int \tilde{p}(\mathbf{x})\, d\mathbf{x}$$

    - Z is a constant when the ϕ functions are constant
    - If ϕ has parameters then Z is a function of those parameters, commonly written without arguments
    - Known as the partition function in statistical physics

Intractability of Z

- Since Z is an integral or sum over all possible values of $\mathbf{x}$ it is intractable to compute
- In order to compute a normalized probability of an undirected model:
  - Model structure and definitions of ϕ functions must be conducive to computing Z efficiently
  - In deep learning applications Z is intractable and we must resort to approximations

Key difference: directed vs undirected modeling

- Directed models are defined directly in terms of probability distributions
- Undirected models are defined loosely in terms of ϕ functions which must then be converted into probability distributions
- This changes intuitions one must use in working with these models
- Domain of the variables has a dramatic effect on the kind of probability distributions a given set of ϕ functions corresponds to

Effect of domain on partition function

- If $\mathbf{x}$ is a n-dimensional vector
- Suppose we have one clique for element of $\mathbf{x}$

$$\phi_i(x_i) = \exp(b_i x_i)$$

- If $x \in R^n$, integral defining Z diverges and no probability distribution exists
- If $x \in \{0,1\}^n$ then $p(x)$ factorizes into n independent distributions with

$$p(x_i = 1) = \text{sigmoid}\ (b_i)$$

  - If domain of x is one-of-k then $p(\mathbf{x}) = \text{softmax}(\mathbf{b})$
    - A large value of b_i reduces to $p(x_j) = 1$ for $j \neq i$

Energy-based Models (EBMs)

- Many interesting theoretical results of undirected models depend on assumption that

$$\forall \mathbf{x}, \tilde{p}(\mathbf{x}) > 0$$

- We can enforce this using an EBM where

$$\tilde{p}(\mathbf{x}) = \exp(-E(\mathbf{x}))$$

  - $E(\mathbf{x})$ is known as the Energy function
  - Because $\exp(z) > 0 \ \forall z$, no energy function will result in a probability of zero for any $\mathbf{x}$
    - If we were to learn clique potentials directly we would need to impose constraints for minimum probability value
    - By learning the energy function we can use unconstrained optimization: probabilities can approach 0

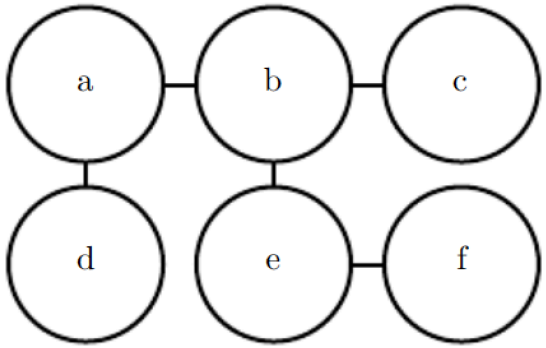
Boltzmann Machines (BMs)

- Any distribution of the form $\tilde{p}(\mathbf{x}) = \exp(-E(\mathbf{x}))$
  - is referred to as a Boltzmann distribution
- For this reason, many energy-based models are referred to as Boltzmann machines
  - No consensus on when to call it a energy-based model or a BM
- BMs were first defined for binary variables but today mean-covariance restricted BMs deal with real-valued variables
  - Today BMs refer to models with latent variables and those without are referred to as MRFs or log-linear

Cliques,factors and energy

- Cliques in the undirected graph correspond to factors in the uunnormalized probability function
- Cliques in the undirected graph also correspond to different terms of an energy function, because $\exp(a)\exp(b)=\exp(a+b)$
- i.e., energy-based model is a special case of a Markov network
  - The exponentiation makes each term of the energy function correspond to a factor for a different clique

Reading the form of the energy function from an undirected graph



- This graph implies that

$$E(a,b,c,d,e,f)=$$

$$E_{a,b}(a,b)+E_{b,c}(b,c)+E_{a,d}(a,d)+E_{b,e}(b,e)+E_{e,f}(e,f)$$

We can obtain the ϕ functions by setting each ϕ to the exponential of the corresponding negative energy, e.g.,

$$\phi_{a,b}(a,b)=\exp(-E(a,b))$$

Separation in PGMs

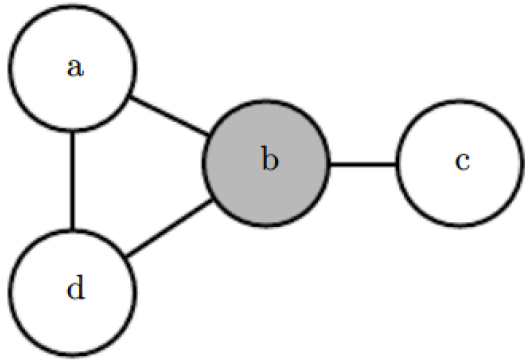
- Edges in a directed graph tell which variables directly interact
- We often need to know which variables indirectly interact
- Some of these interactions can be enabled or disabled by observing other variables
- More formally we would like to know which variables are conditionally independent of each other given the values of other sets of variables

Separation in undirected models

- Identifying conditional independences is very simple in the case of undirected models
  - In this case conditional independence implied by the graph is called separation
  - Set of variables A is separated from variables B given third set of variables S if the graph implies that A is independent of B given S
  - If two variables a and b are connected by a path involving only unobserved variables then those variables are not separated
    - If no path exists between them, or all paths contain an observed variable then they are separated

Separation in undirected graphs

- b is shaded to indicate it is observed



- b blocks path from a to c, so a and c are separated given b
- There is an active path from a to d, so a and d are not separated given b

Separation in Directed Graphs

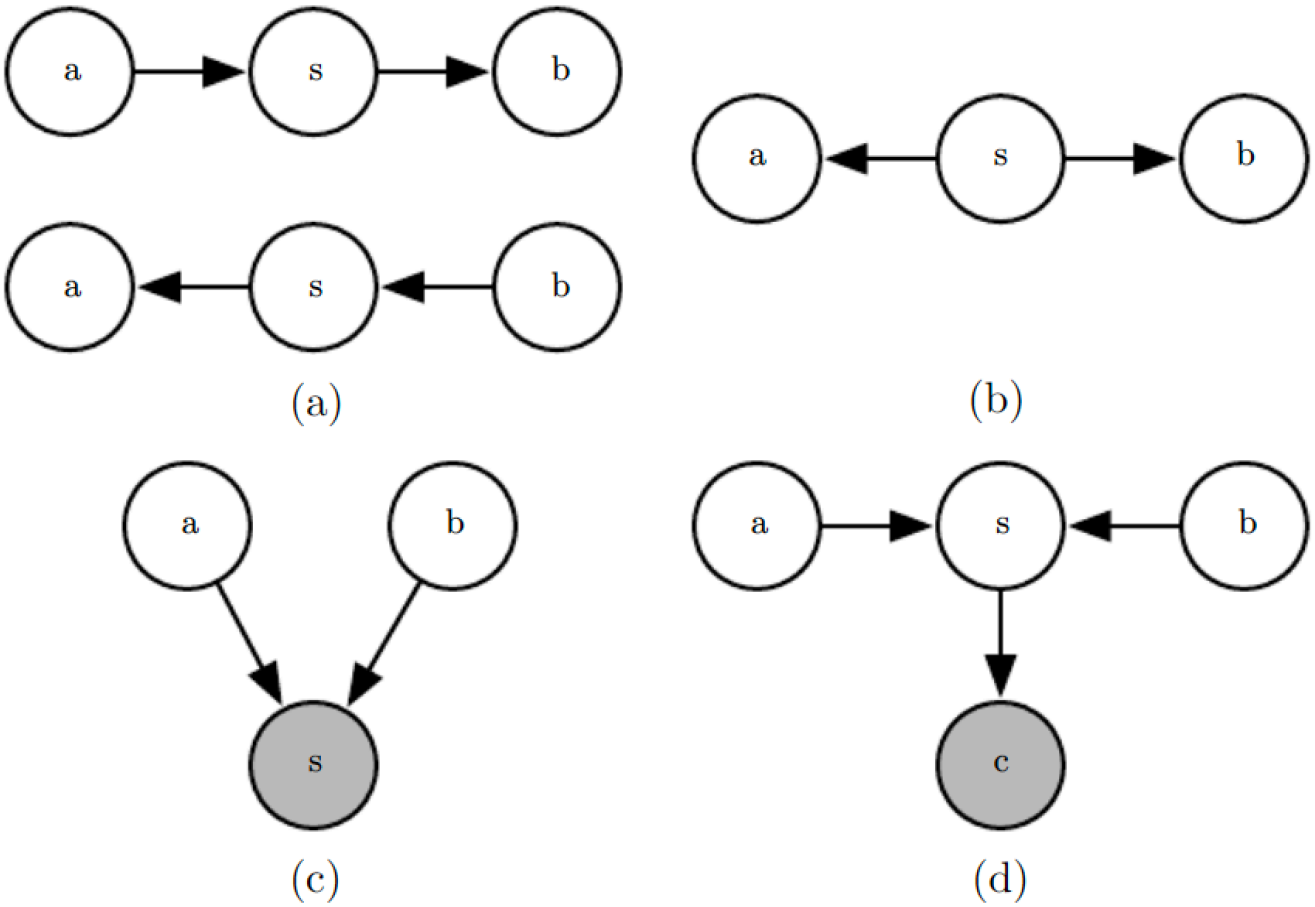
- In the context of directed graphs, these separation concepts are called d-separation
  - d stands for "dependence"
- D-separation is defined the same as separation for undirected graphs:
- A set of variables A is d-separated from a set of variables B given a third set of variables S if the graph structure implies that A is independent of B given S

Examining Active Paths

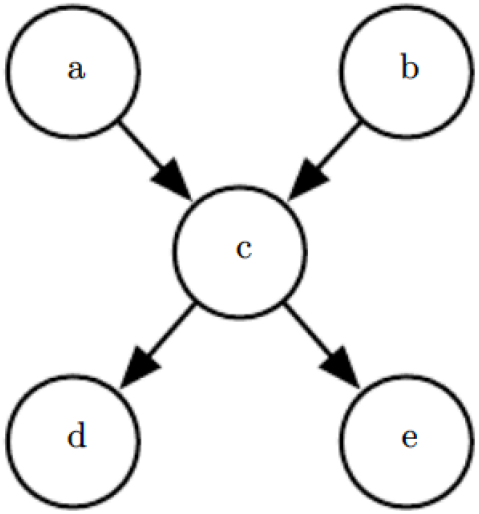
- Two variables are dependent if there is an active path between them
- They are d-separated if there is no path between them
- In directed nets determining whether a path is active is more complicated
- A guide to identifying active paths in a directed model is given next

All active paths of length 2

Active paths between random variables a and b



Reading properties from a graph



- a and b are d-separated given the empty set.
- a and e are d-separated given c.
- d and e are d-separated given c.

- a and b are not d-separated given c.
- a and b are not d-separated given d.