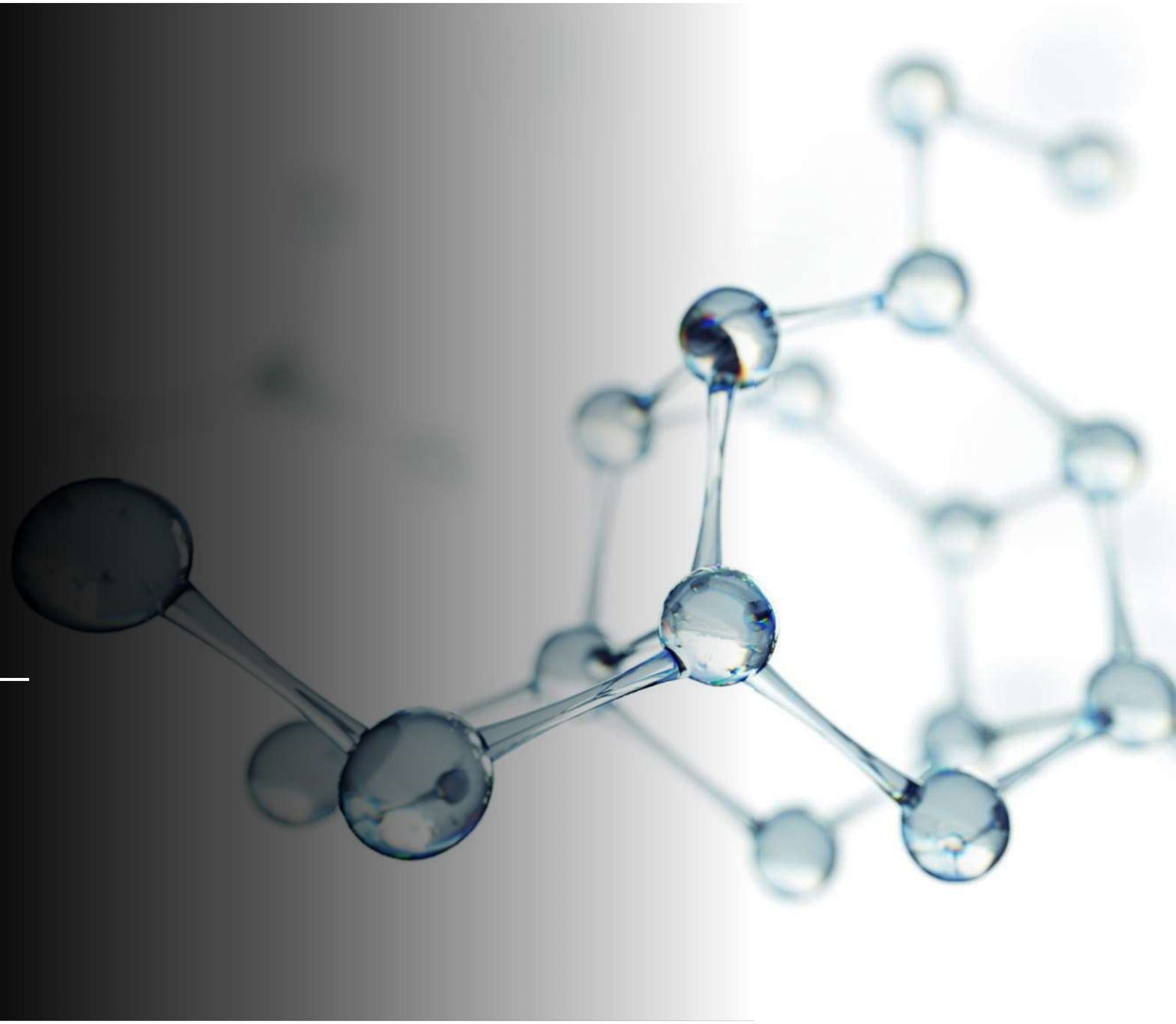
Graphical Models

Jiatong



Probabilistic Models

- We have considered many probabilistic models
  - Linear Regression
  - Logistic Regression
  - Gaussian Mixture Models

- Most of these have been very simple
  - Assume a label (observed or unobserved)
  - Estimate probabilities from data

Model Representations

- No formal language to talk about model
  - We've described the models and given intuition

- Example: Gaussian Mixture Models
  - Assume that we first select a cluster
  - We then generate an example (features) given the cluster

- How can we describe this model formally

Example Probabilistic System

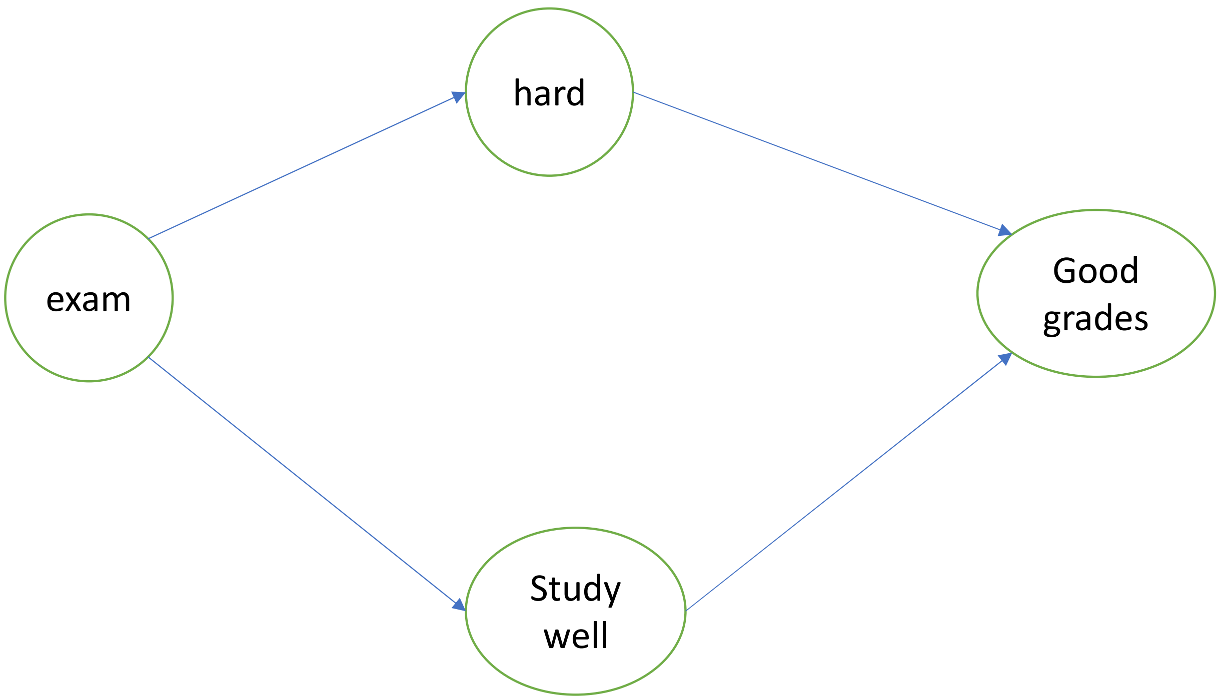
- A collection of related binary random variables
  - We will have an exam today
  - Student 1 studies hard
  - The exam is hard
  - The grades of student 1 are good

Example

- How do we answer these questions
  - What is the structure of these variables
  - What probabilities do I need to compute
  - Are any of the variables independent of each other

- We need some representation for these variables

Graphical Models



Graphical Models

- Combination of probability theory and graph theory
  - Combines uncertainty (probability) and complexity (graphs)
  - Represent a complex system as a graph
    - Gives Modularity
  - Standard algorithms for solving graph problems

- Your favorite algorithms are graphical models
  - Logistic regression, linear Regression, GMMs, etc.

Representation

- A probabilistic system is encoded as a graph
  - Nodes
    - Random Variables
      - Could be discrete or continuous
  - Edges
    - Connections between two nodes
    - Indicates a direct relationship between two random variables
    - Note: the lack of an edge is very important
      - No direct relationship

Graph Types

- Edge type determines graph type
- Directed graphs
  - Edges have directions (A -> B)
  - Assume DAGs (no cycles)
  - Typically called Bayesian Networks
- Undirected graphs
  - Edge don't have directions (A – B)
  - Typically called Markov Random Fields (MRFs)

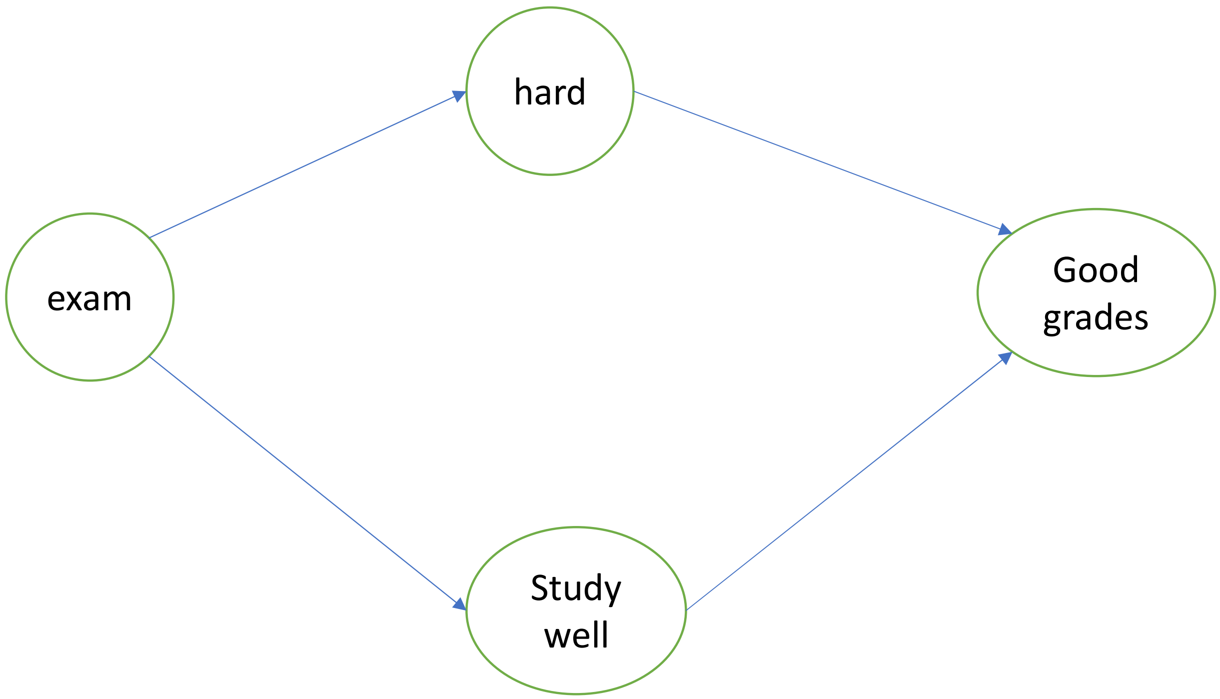
Directed Graphs

- The direction of the edge indicates causation

- Causation can be very intuitive
  - We may know which random variable causes the other
  - Use this intuition to create a graph structure

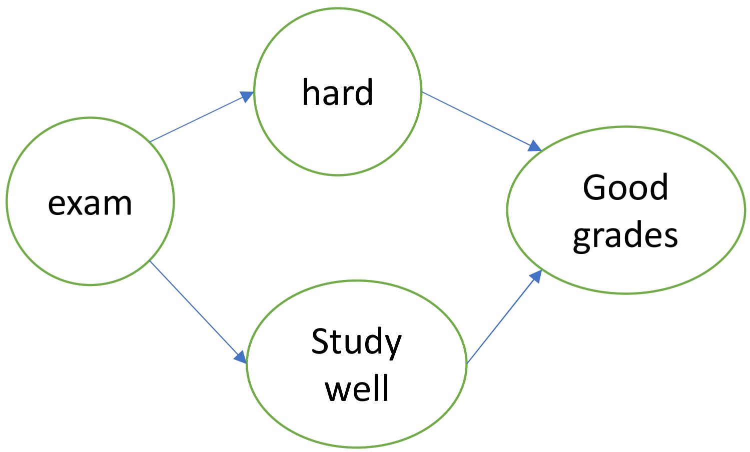
Example

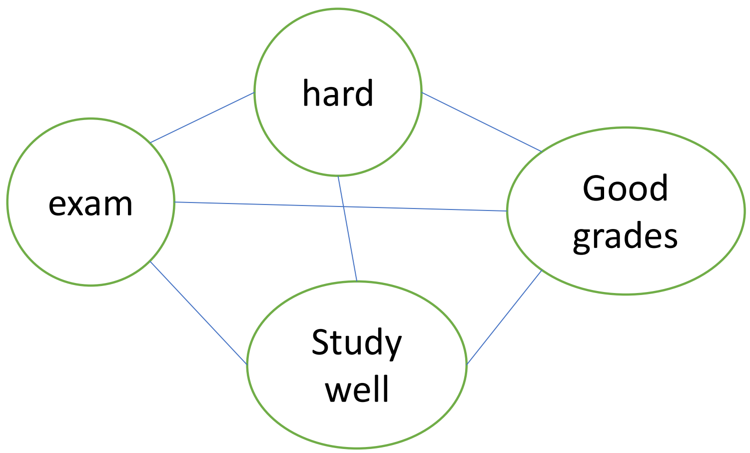
- Generative Model



Advantages

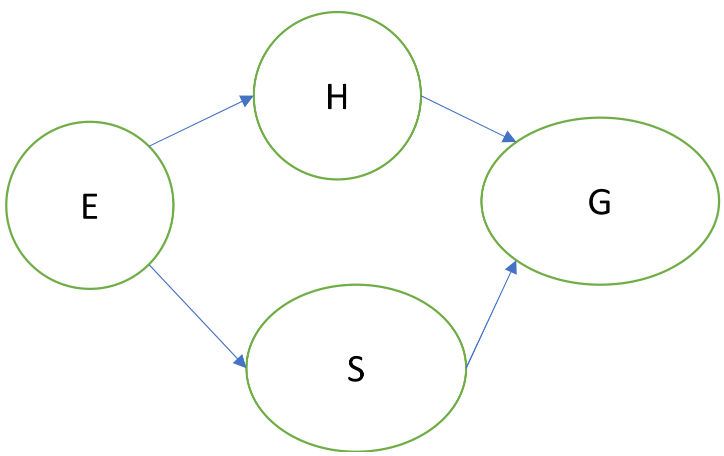
- What have we gained by this representation
  - We could just draw a graph where everything is connected





Factorization

- Consider the joint probability of our example
  - P(E, H, S, G), this is complex
  - What can we do to simplify?
  - Notice that H, S are independent given E



Product Rule

- Can use the product rule to decompose joint probabilities
  - $P(a,b,c) = P(c|a,b)P(a, b)$
  - $P(a,b,c)=P(c|a,b)P(b|a)P(a)$

- This is true for any distribution
- Same for K variables

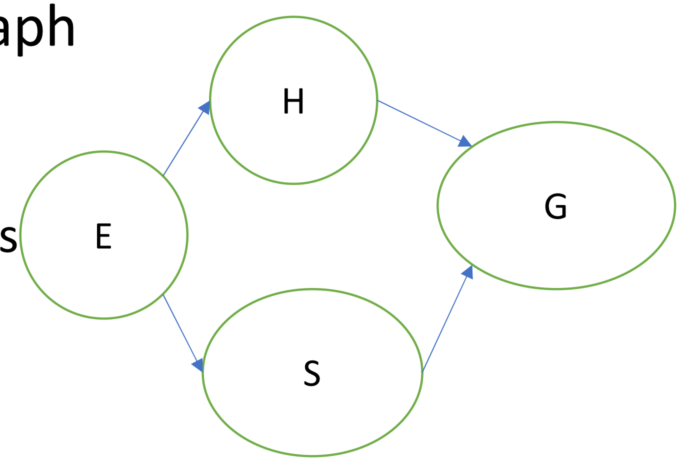
Factorization

- For any graphical model, we can write the joint distribution using conditional probabilities
  - We just need conditional probabilities for a node given its parents

  - $p(x) = \prod_k p(x_k | parents_k)$

Factorization

- Consider the joint probability of our example
  - P(E, H, S, G), this is complex
  - What can we do to simplify?
  - Notice that H, S are independent given E
- Factor the joint probability according to the graph
  - $P(E, H, S, G)=p(G|H, S)\ p(H|E)p(S|E)p(E)$
  - This is much simpler to compute
  - We are likely to have these conditional probabilities



Observed Variables

- Variables are either
  - Observed- we observe values in data
  - Hidden- we cannot see values in data

- Indicate observed variables by shading
- Compute the remaining probabilities given shaded value

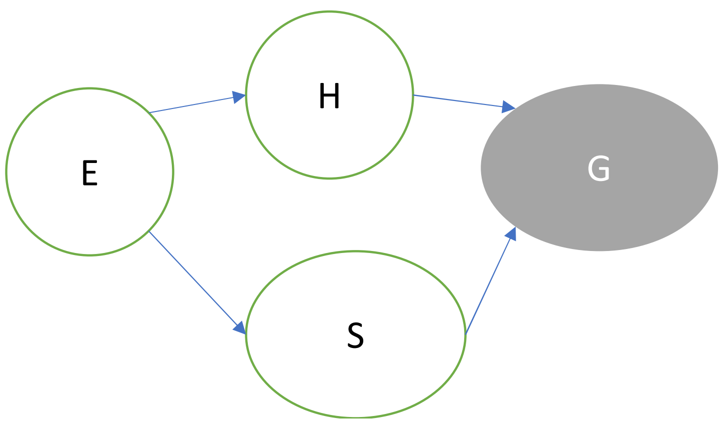
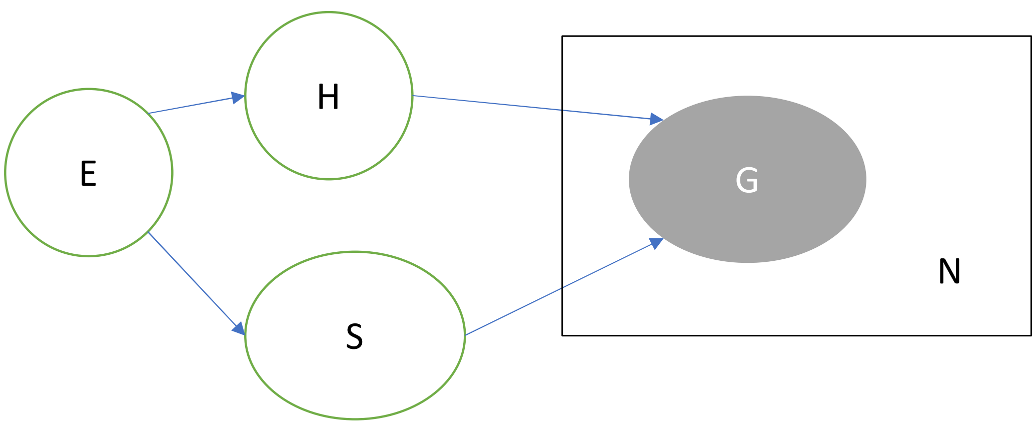


Plate Notation

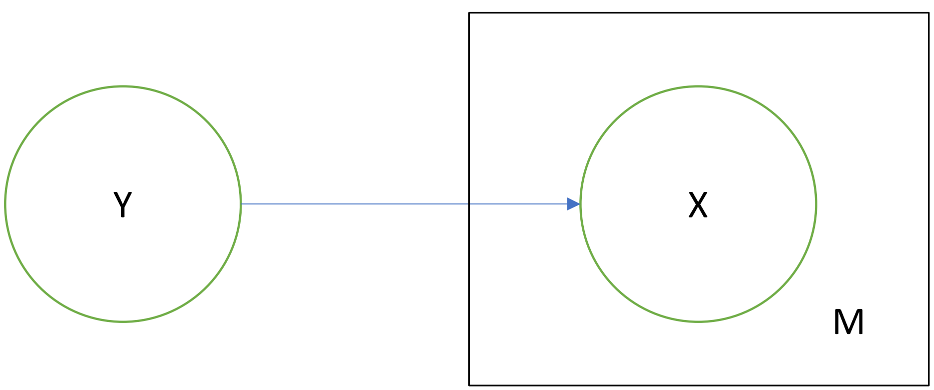
- Plates in Graphical Models
  - When many variables have same structure, we replace them with a plate
  - The plate indicates repetition

  - There are n problems presented in the exam
  - Each conditioned on the same H,S



Example

- A model where we have label Y and example X
- Test time, no Y
  - Estimate Y using X
- What model is this?



Naïve Bayes

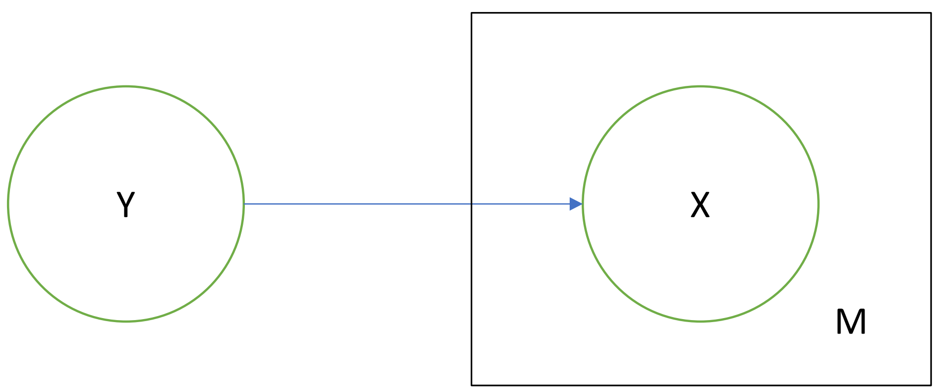
- Generative Story
  - Generated a label Y
  - Given Y, generate each feature X independently

- Learning
  - We observe X and Y, maximum likelihood solution

- Prediction
  - Compute most likely value for Y given X

Factorization

- $p(y, x) = p(x|y)p(y) = \prod p\left(x_j \middle| y\right)p(y)$



Learning

- We assumed both examples (X) and labels (Y) for learning naïve Bayes
  - Maximum likelihood solution

- What if we only have X
  - General purpose method for maximizing likelihood where we have missing variables
  - EM
  - Unsupervised NB: clustering
  - Some labels: semi-supervised NB

Naïve vs. Reality

- Positive: we now can parameterize our model using conditional independence
- Reality: naïve assumption very unlikely to be true

- Example
  - Document classification: sports vs. finance
  - Each word in a document is a feature
  - Naïve assumption: once I know the topic is sports, every word is conditionally independent
    - Not true! Would be total nonsense.

Naïve vs. Reality

- Reality: works pretty well in practice
- Caution: features that are too dependent are difficult for model
  - Create features that are minimally dependent
  - Limits the expressiveness of features

Assumptions

- Naïve Bayes makes an assumption
- Features (X) conditionally independent given label (Y)

Undirected Networks

- Conditional Independence
  - D-separation used directionality of the edges
- Can we define it without directionality?
  - Yes!
- If all path from set A to set B pass through set C, then A is d-separated from B given C
  - Notice no distinction between head to head and tail to tail
  - “Explaining away” not an issue since no causation
  - Actually easier to check

Factorization

- How can we express the joint distribution as a product of functions over local sets of variables
  - The directions in directed graphs indicated conditional relationship
- Step 1 is easier than with directed graphs
  - Given two nodes xi and xj , they are conditionally independent given the entire graph if they are not neighbors

  - $p\left(x_i, x_j \middle| X_{\{i,j\}}\right) = p\left(x_i \middle| X_{\{i,j\}}\right) p\left(x_j \middle| X_{\{i,j\}}\right)$

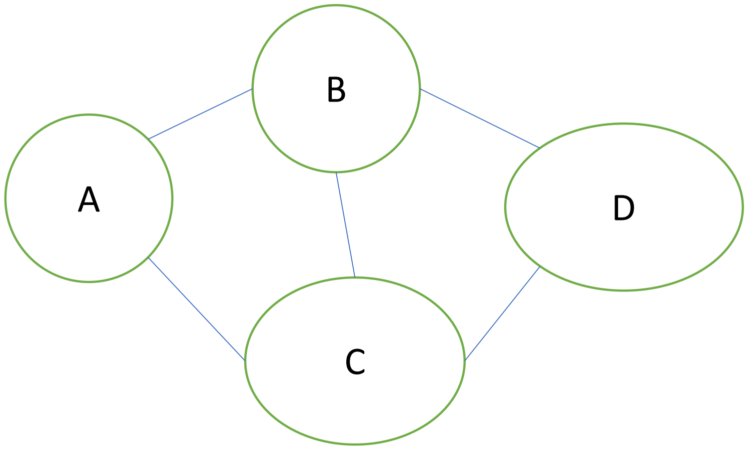
Defining Factors

- The factorization of the joint distribution is such that xi and xj do not appear in the same factor

- Graph concept: clique
  - A set of nodes that are fully connected
    - There exists an edge between every pair of nodes
- Maximal clique
  - A clique such that adding any other node means it is no longer a clique

Cliques

- Cliques in the graph
  - A/B, A/C, B/D, B/C, C/D
  - Maximal Cliques: A/B/C, B/C/D
  - A/B/C/D is not a clique, since no edge from A to D



- We just need to use maximal cliques, since they contain all other cliques

Factorization

- Define
  - Xc as all nodes in clique C
  - $\psi_C(\mathrm{x}_c)$ is a potential function over clique C

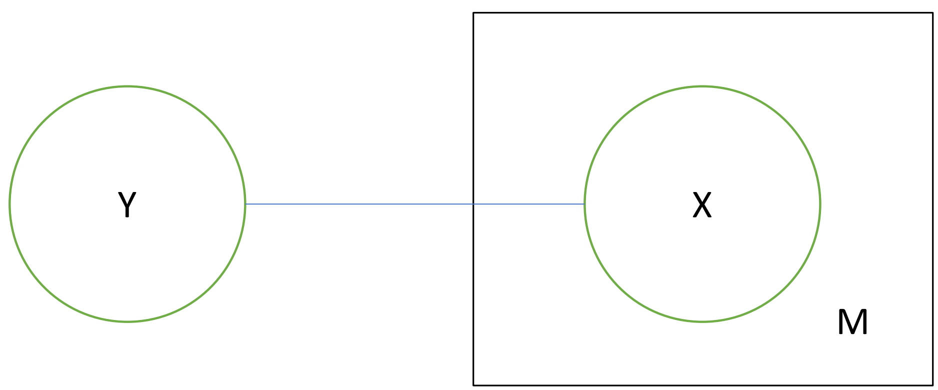
- We can define the joint distribution of the graph as a product of potential functions over maximal cliques
  - $p(x) = \frac{1}{Z}\prod_C \psi_C(\mathrm{x}_c)$

Partition Function

- $p(x) = \frac{1}{Z}\prod_C \psi_C(\mathrm{x}_c)$
- Notice that $\psi_C(\mathrm{x}_c)$ is not a probability
  - Must be non-negative
  - Will not sum to 1
- Therefore, we need to normalize to get a probability
  - $Z = \sum_x \prod_C \psi_C(\mathrm{x}_c)$
- Z is called the partition function

Example

- A model where we have label Y and example X
- Test time, no Y
  - Estimate Y using X
- What model is this?

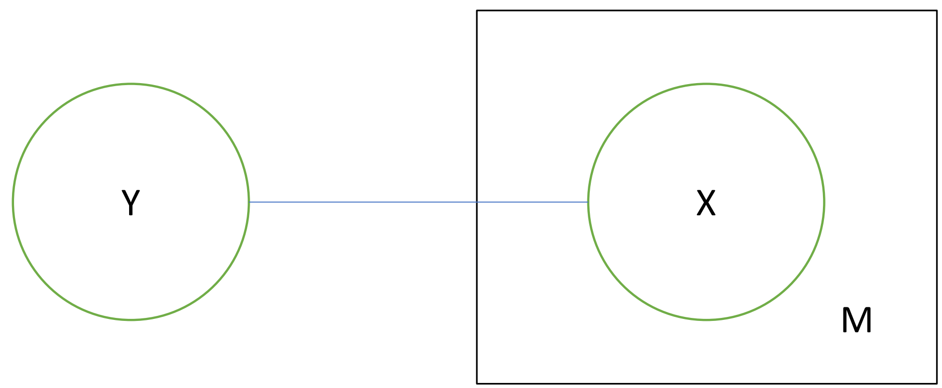


Logistic Regression

- No Generative Story
  - Graph does not encode causality
  - We can say that X and Y are related
- Learning
  - We observe X and Y, maximum likelihood solution
- Prediction
  - Compute most likely value for Y given X

Factorization

- $p(y|x) = \frac{1}{Z} \prod_m \psi_m(\mathrm{x}_m, y)$
- $\psi_m(\mathrm{x}_m, y) = \exp\left(w_m, f(x_m, y)\right)$
- $Z = \sum_y \prod_m \psi_m(\mathrm{x}_m, y)$



MRF

- Pros:
  - Define arbitrary potential functions
  - Much more flexibility than directed models
  - Easier to compute condition independence
  - Don't need to express causation relationship
- Cons
  - Z!!!
  - Sum over all states x
  - M discrete nodes, each with K discrete states, K^M
    - However, we just need Z for learning
    - To evaluate, we need most likely option, so Z cancels

Inference

- Computing probabilities of network configurations
  - We know some values of the network (observed)
  - How do we compute the posterior of a set of nodes?
- Previously, we did this by explicitly working out the probabilities
- How can we do this in an efficient and general way?

Two Approaches

- Exact inference
  - We get the exact value of the probability we want
  - While some efficient algorithms exist, very slow for some graphs
- Approximate inference
  - Compute an approximation of the desired probability
  - The only solution for some types of graphs

Chain Graphical Model

- Consider a linear chain of random variables
  - Notice this is undirected
  - We can convert directed models to undirected model

- Joint distribution of the Chain
  - $p(x) = \frac{1}{Z} \psi_{1,2}(\mathrm{x}_1, x_2)\, \psi_{2,3}(\mathrm{x}_2, x_3) \ldots\, \psi_{N-1,N}(\mathrm{x}_{N-1}, x_N)$
  - Given N nodes with K states
  - Each potential function is a K*K table
  - Joint has $(N-1)K^2$ parameters

Chain Inference

- What is p(xn) for some node n in the chain
  - Assuming no observed nodes

- Sum over all other nodes in the chain
  - $p(x_n) = \sum_{x_1} \dots \sum_{x_{n-1}} \sum_{x_{n+1}} \dots \sum_{x_N} p(x)$
  - Notice there are K^N values to consider in the summation
  - Our computations are exponential in the length of the chain

Conditional Independence

- Conditional independence to the rescue!
  - We can write the joint in terms of potentials
  - Each potential depends on 2 nodes

- Plug the factorized joint into the marginal for $p(x_n)$

Grouping Potential Function

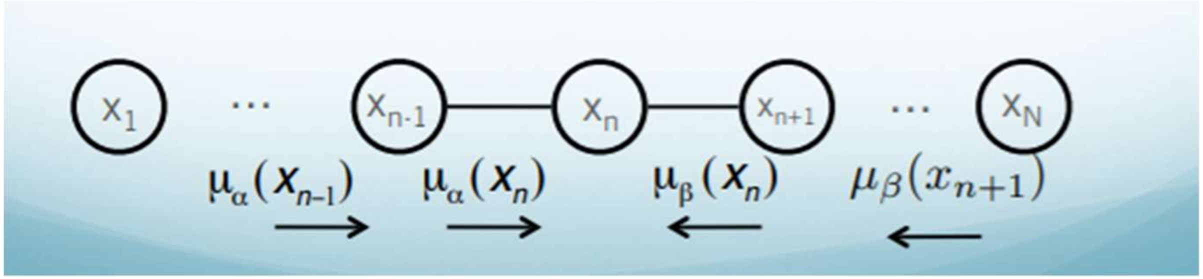
- $p(x_n) = \frac{1}{Z}\mu_\alpha \cdot \mu_\beta = \frac{1}{Z}$

$$\left[\sum_{x_{n-1}} \psi_{n-1,n}(\mathrm{x}_{n-1}, x_n) \dots \left[\sum_{x_2} \psi_{2,3}(\mathrm{x}_2, x_3) \left[\sum_{x_1} \psi_{1,2}(\mathrm{x}_1, x_2)\right]\right]\right]$$

$$\left[\sum_{x_{n+1}} \psi_{n,n+1}(\mathrm{x}_n, x_{n+1}) \dots \left[\sum_{x_N} \psi_{N-1,N}(\mathrm{x}_{N-1}, x_N)\right]\right]$$

Rewriting as Factors

- The marginal can be written in terms of two factors
  - $p(x_n) = \frac{1}{Z}\mu_\alpha(x_n) \cdot \mu_\beta(x_n)$
  - Each Factor depends only on the odes to one side
  - This information is passed along the network to xn
  - We call this a message
    - Each message contains K values (for every xn)



Normalization Constant

- Z is a sum over all states in
  - $p(x_n) = \frac{1}{Z}\mu_\alpha(x_n) \cdot \mu_\beta(x_n)$

Computing Probabilities

- What about observed variables
  - Clamp their value, remove the sum

- What about joint distribution over two neighboring nodes
- $p(x_{n-1}, x_n) = \frac{1}{Z}\mu_\alpha(x_{n-1}) \cdot \psi_{n-1,n}(\mathrm{x}_{n-1}, x_n) \cdot \mu_\beta(x_n)$

Most Likely Configuration

- We know how to find the probability of configuration
- How do we find the most likely configuration
  - Run the sum product algorithm, for each marginal
  - Select the most probable value for each node
  - Problem: this gives a node specific max probability

- How to do:
- Simply replace the sum to max

Structured Prediction

What is structured prediction

- Input x
  - Typically a structured input
  - Maintain structure of input in x
    - Do not flatten into list of features in an instance
- Output y
  - Y is now from a large set of possible outputs
  - Outputs defined based on input
    - Often exponential in size of input

Previous Approaches

- Naturally multi-class algorithm,
  - Neural Networks
  - Decision Tree
- Reduction to binary
  - E.g. one classifier per class
- These methods don’t work when
  - Exponential number of output
  - Outputs defined based on input

Structured Prediction Challenges

- Scoring
  - How do we assign a score/probability to a possible output structure

- Search/Inference
  - Find the best scoring output structure
  - How do we search through an exponential number of options

Sequence Models

- Many events happen in sequence
  - Weather on consecutive days
  - Words in a written sentence
  - Spoken sounds by a person
  - Movements in the stock market
  - DNA base pairs

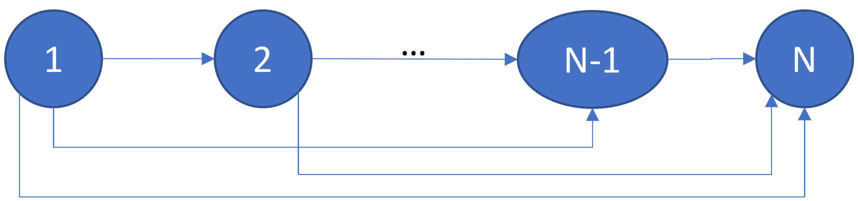
Sequential Model

- Simple Approach
  - Each event is independent

  - Simple but not helpful

Sequential Model

- Complex Approach
  - Each event is dependent on previous events



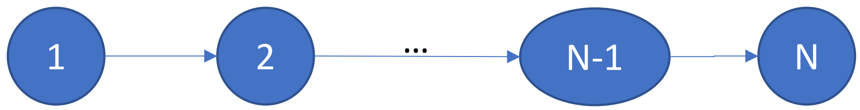
  - Captures dependencies, but way too complex

Markov Assumption

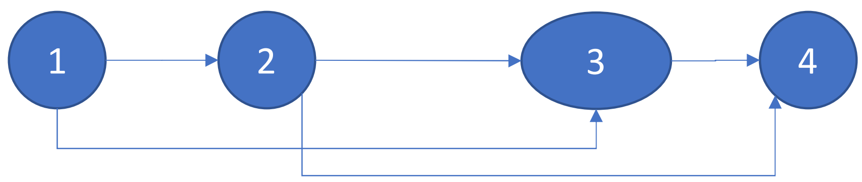
- The current state depends on a fixed number of previous states
  - The weather today depends on the past three days, but not two weeks ago
  - The next word in the sentence depends on the past three words, but nothing before
- Pros: make for simple models
- Cons: doesn’t capture full history

Markov Chains

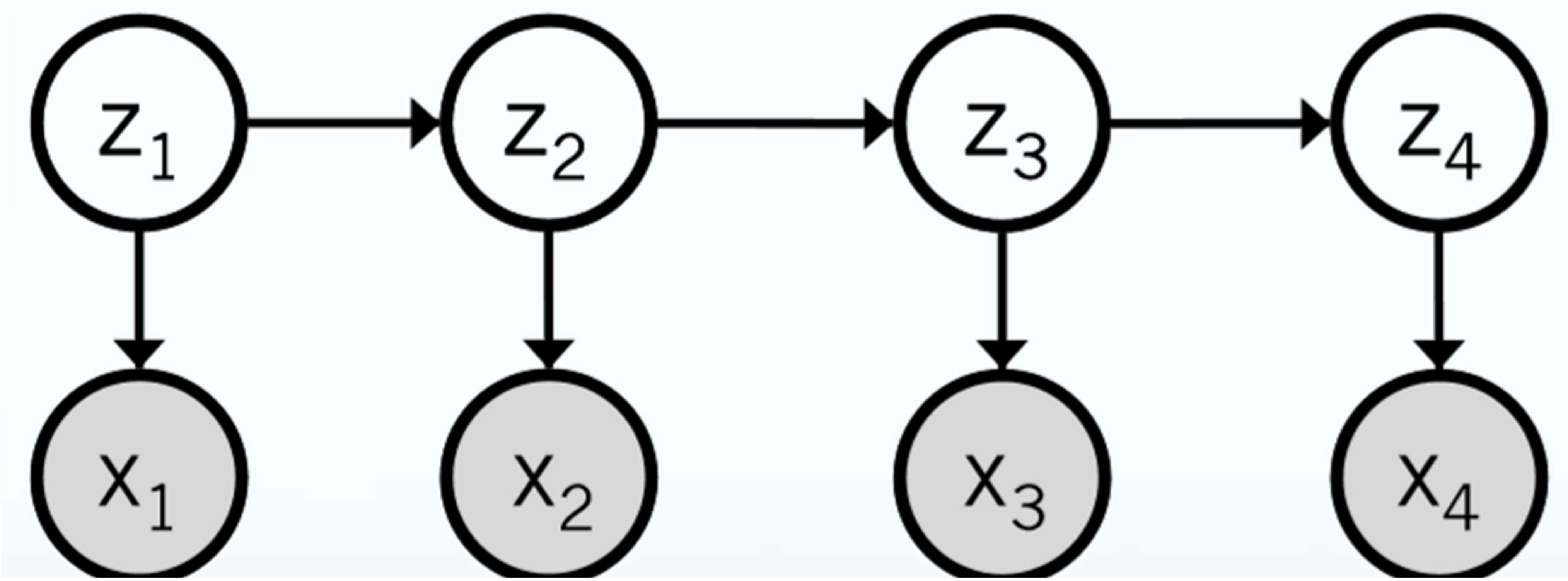
- First order Markov chain



- Second order Markov chain



Hidden Markov Model



Joint Probability of an HMM

- The joint probability of an HMM
- $p(X, Z|\theta) = p(z_i|\pi)[\prod_n p(z_n|z_{n-1}, A)] \prod_m p(x_m|z_m, \phi)$
- A transition probabilities
  - The probability of moving from state i to j
- Pi – vector with starting probabilities
- Phi – emission probabilities

Unsupervised Training

- How do we train a probabilistic model
  - Maximum likelihood
    - $\max_{\theta} p(X, Z|\theta)$
  - Problem: we don't know z

- Solution: EM

Step1: Write the likelihood

- $p(X|\theta) = \sum_Z p(X, Z|\theta)$
- $p(X, Z|\theta) = p(z_i|\pi)[\prod_n p(z_n|z_{n-1}, A)] \prod_m p(x_m|z_m, \phi)$

EM for HMMs

- E step
  - Find the expected values for the hidden variables Z given the model parameters
    - The most likely Z given X and current model parameters

- M-step
  - Pretend to observe the values for Z
  - Update model parameters to maximize complete data likelihood

E step

- Given Q function, evaluate probabilities for Z
- $Q\left(\theta, \theta^{old}\right) = \sum_Z p\left(Z \middle| X, \theta^{old}\right) \log p(X, Z|\theta)$

- For HMM:

$$Q(\theta, \theta^{old}) = \sum_{k=1}^{K} \gamma(z_{1k}) \log \pi_k + \sum_{n=2}^{N} \sum_{j=1}^{K} \sum_{k=1}^{K} \xi(z_{n-1,j}, z_{nk}) \log A_{jk} + \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk}) \log p(x_n \mid \phi_k)$$

$$\gamma(z_n) = p(z_n \mid X, \theta^{old})$$

π, Φ and $\mathbf{A}$ are model parameters

$$\xi(z_{n-1}, z_n) = p(z_{n-1}, z_n \mid X, \theta^{old})$$

How can we get these values

- What is the probability of being in state zn
- What is the probability of bing in state zn and zn+1

- Inference
  - Forward-backward algorithm

M-step

Maximize Q function with respect to pi, phi and A

Assuming values for Z, we get:

$$\pi_k = \frac{\gamma(z_{1k})}{\sum_{j=1}^{K} \gamma(z_{1j})} \qquad A_{jk} = \frac{\sum_{n=2}^{N} \xi(z_{n-1,j}, z_{nk})}{\sum_{l=1}^{K} \sum_{n=2}^{N} \xi(z_{n-1,j}, z_{nl})} \qquad \phi_{ik} = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) x_{ni}}{\sum_{n=1}^{N} \gamma(z_{nk})}$$

Prediction

- Given a new sequence X, find the most likely set of states to have generated X
  - Find the sequence Z with the maximum probability given X

- Viterbi Decoding! (replace sum to max)

Supervised Training

- We actually observe Z
    - Just compute M step a single time
    - Very fast and easy
- What if we observe only some Z
    - Case 1: only some examples are labeled with Z
    - Case 2: each example has only some labels for Z
- Semi-supervised Learning ● Use EM algorithm but fix Z when know

Notes on HMMs

An HMM can have continuous or discrete emissions

- Discrete- base pair, word in sentence
- Continuous- stock price, frequency of a sound

An HMM has discrete hidden states

A Linear Dynamical System has continuous hidden states

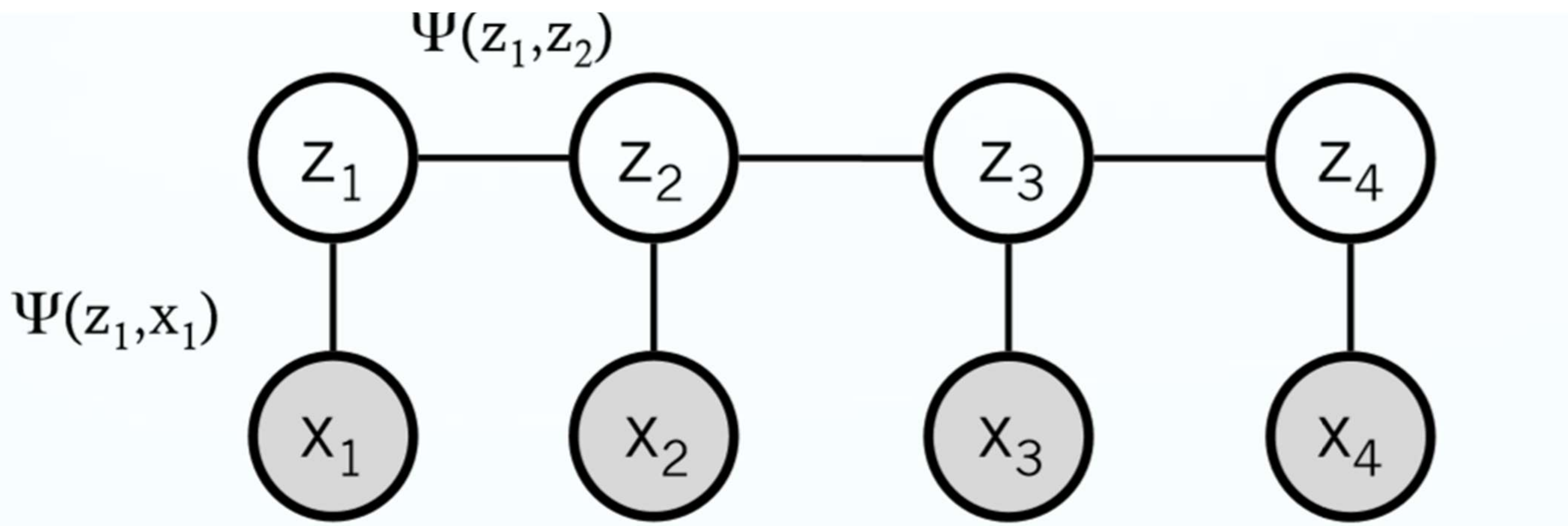
Sequence Models

- An HMM is a directed graphical model
  - Bayesian Network

  - What happens if we have an undirected graphical model
    - Markov Random Field

Conditional Random Fields

- Replace conditional probabilities with potential functions
- The joint probability is a product of potential functions



Conditional Random Fields: Learning

- Given some data, we want to learn a CRF
- How should we learn the model
  - Maximum Likelihood

- Questions
  - What are the parameters of our model
    - The potential functions
  - What is the objective
  - How do we compute model probabilities efficiently

Model Parameters

- Parameterize the potential functions
  - Learn the parameters
    - $\phi(x, z) = \exp(\sum_k \theta_k f_k(x, z))$

  - Parameter theta determine value of potential function
  - Fk is a feature function
  - Notice: linear combination of features(linear model!)

Objective

- Let's maximize the likelihood of the data
  - For a single example
    - $p(x, z|\theta) = \frac{1}{Z} \prod_A \phi_A(x_A, z_A)$
  - What about Z?
    - $Z = \sum \prod_A \phi_A(x_A, z_A)$

- Sum over all X!
  - All possible sequences of base pairs
- It's too hard to learn X

Discriminative Training

- Solution: don't learn p(x)

- Maximize the conditional likelihood of the data
  - For a single example

- CRF conditional log likelihood of all examples

$$p(\mathbf{Z} \mid \mathbf{X}, \theta) = \frac{1}{Z} \prod_{A} \psi_A(\mathbf{x}_A, \mathbf{z}_A)$$

$$\mathbf{Z} = \sum_{Z} \prod_{A} \psi_A(\mathbf{x}_A, \mathbf{z}_A)$$

$$\log p(z \mid x) = \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{k=1}^{K} \theta_k f_k(z_{it}, z_{it-1}, x_{it}) - \sum_{i=1}^{N} \log Z(x_i)$$

Problem with Objective

- Recall the logistic regression (discriminative training) maximum likelihood over-fit the data
- Solution: regularization
- Gaussian prior

$$\log p(z \mid x) = \sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{k=1}^{K}\theta_k f_k(z_{it}, z_{it-1}, x_{it}) - \sum_{i=1}^{N}\log Z(x_i) - \sum_{k=1}^{K}\frac{\theta_k^2}{2\sigma^2}$$

Training a CRF

- The conditional log likelihood is convex
  - Take the derivative and solve for θ

$$\frac{\partial L}{\partial \theta_k} = \sum_{i=1}^{N}\sum_{t=11}^{T} f_k(z_{it}, z_{it-1}, x_{it}) - \sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{z,z'} f_k(z, z', x_{it})\, p(z, z' \mid x_{it}) - \sum_{k=1}^{K}\frac{\theta_k}{\sigma^2}$$

  - The derivative is 0 when
    - The last term (regularizer) is 0
    - The first term and the second term cancel each other
      - First term: the expected value for fk under the empirical distribution (from the data)
      - Second term: expectation for fk given odel distribution

Computing Probabilities

- What do we need to compute the values in the derivative?
  - Marginal probability of $p(z, z'|x_{it})$
  - The normalization constant Z
    - Total score for all possible labeling of the sequence
    - Forward-Backward

- Prediction
  - Sequence of states with max probability
  - Prediction
    - How do we find the highest probability sequence?
    - Viterbi Decoding

CRF summary

- CRFs are
  - Markov Random Fields
  - The MRF equivalent of a supervised HMM
  - Discriminatively trained using conditional log likelihood
  - Linear model(recall linear potential functions)

Why CRFs

- CRF training is much harder than HMM
  - Computing gradients, optimization vs. counting
  - 11 labels, 200k tokens: 2 hours / 45 labels, 1m tokens: 1 week
- Why bother?
  - HMMs require
    - Assumptions of causation / generative story
    - Independence assumptions for observations
  - These aren't problems for CRFs!
    - Can allow arbitrary dependencies
    - Transition can depend on x and z
    - Can condition on the whole sequence x
    - Recall:
      - Generative models limit the features
      - Discriminative models can have any types of features

HMMs and CRFs

Generative/Discriminative pairs

- A generative and discriminative parametric model family that can represent the same set of conditional probability distributions
- Naïve Bayes/Logistic Regression
- HMM/CRF

HMM is a Naïve Bayes classifier at each node

CRF is a Logistic Regression classifier at each node

Q&A