CS 583: Probabilistic Graphical Models

Topic: Parameter Estimation

Mustafa Bilgic

http://www.cs.iit.edu/~mbilgic

https://twitter.com/bilgicm

Overview

- Bayesian networks
  - Maximum likelihood estimation
  - Bayesian estimation
  - Incomplete data
- Markov networks
  - Maximum likelihood estimation
  - Gradient optimization
  - Bayesian estimation
  - Regularization

Bayesian Networks

Parameter estimation for BNs

- Assume the network structure is given
- The data $\mathcal{D}$ consists of fully observed instances of the network variables
  - $\mathcal{D} = \{x[1], x[2], \ldots, x[n]\}$
- Estimate the network parameters, i.e., learn the CPDs
- Two approaches
  1. Maximum likelihood estimation
  2. Bayesian estimation

Simplest case – one variable

- Imagine we have a thumbtack
- Flip it, and it comes as heads or tails

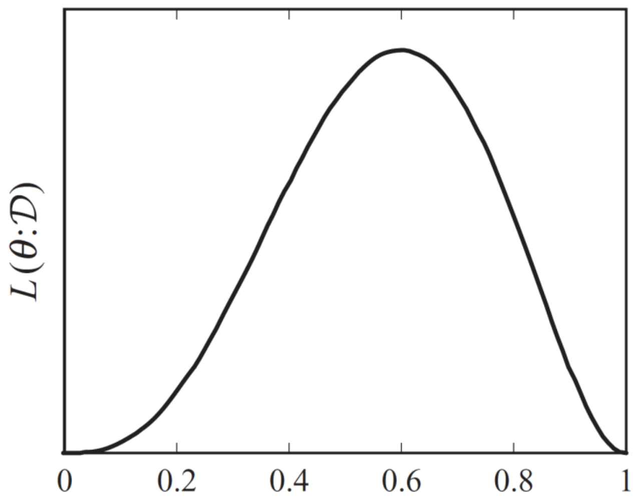
- Assume we flip it 100 times and it comes head 30 times
- What is θ?

THUMBTACK TOSSES

- Assume we have a set of thumbtack tosses
  - $\mathcal{D} = \{x[1], \ldots, x[n]\}$
- Also assume each toss, x[i], is IID
- We define a *hypothesis space* Θ
  - Θ is the set of all parameters $\theta \in [0, 1]$
- We formulate an *objective function*
  - The objective function tells us how good a given hypothesis (in this case θ) is

Likelihood

- What is the probability, or *likelihood*, of seeing the sequence H, T, T, H, H?
  - $\theta * (1-\theta) * (1-\theta) * \theta * \theta = \theta^3(1-\theta)^2$



When is $L(\theta : \mathcal{D})$ maximum?

Likelihood/Log-likelihood

- Number of heads = h, number of tails = t
- Likelihood: $L(\theta:\mathcal{D}) = \theta^h(1-\theta)^t$
- Log-likelihood: $l(\theta:\mathcal{D}) = h\ln\theta + t\ln(1-\theta)$
- Find θ that maximizes the log-likelihood
- Take derivate of $l(\theta:\mathcal{D})$ with respect to θ and set it to zero

Maximum likelihood for a multinomial

- Domain of X is {A, B, C}
- We see A a times, B b times, and C c times.
- P(X=A) is p, P(X=B) is q, and P(C) = $1 - p - q$
- What are p and q?
- Proof?

ML FOR BNS

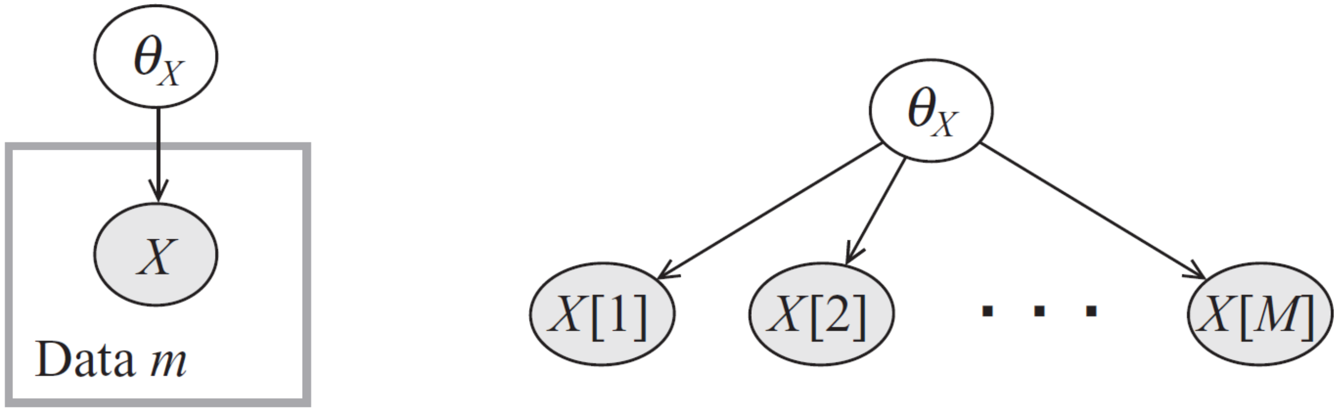
- Simple structure
  - $X \rightarrow Y$
- General structure
  - The key is that the parameters for each variable can be optimized independently
  - Examples

Bayesian estimation

- Assume we flip a coin 10 times and we get 4 Heads, 6 Tails
  - What is P(C=H)?
- Assume we flip a thumbtack 10 times and we get 4 Heads, 6 Tails
  - What is P(T=H)?
- What if we repeat the flips 10M times and we get 4M Heads and 6M Tails?
- Bayesian estimation will let us encode our *prior knowledge*

INDEPENDENCE?

- Earlier, we assumed the tosses are independent
- This is true if we know θ
- If we don't know θ, then each toss tells us something about θ, thus the next toss

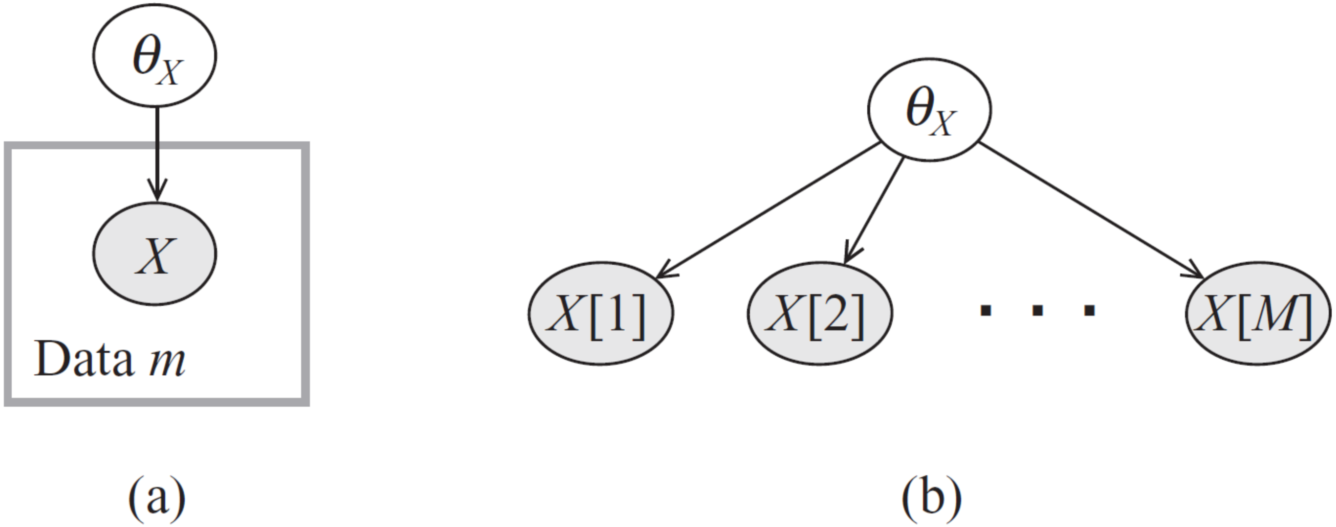


(a) (b)

Bayesian Estimation

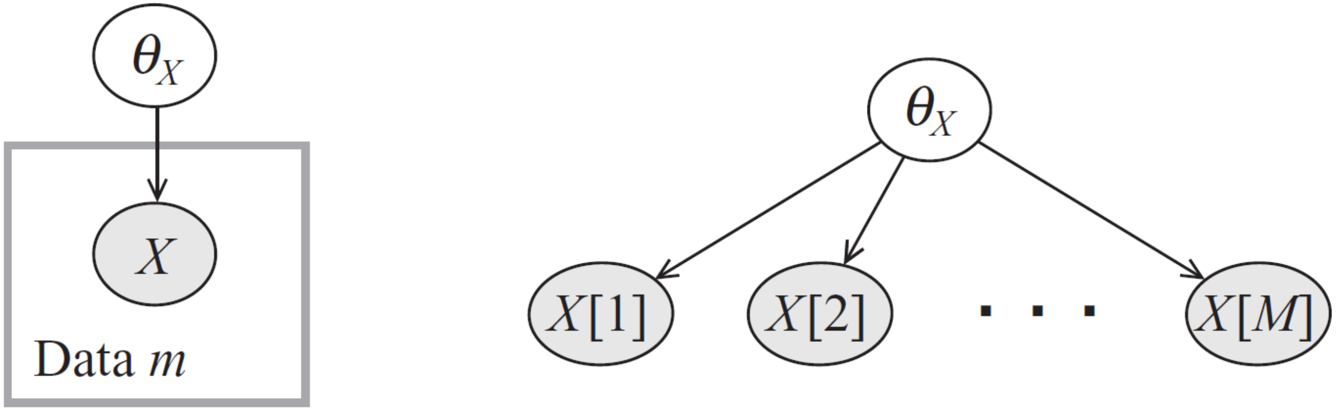
- Rather than a single θ, we will instead have a probability distribution, $p(\theta)$, over θ

Bayesian estimation



- We treat the parameter θ as a random variable
- We ascribe a prior probability to θ, $p(\theta)$, encoding our prior knowledge

PARAMETERS



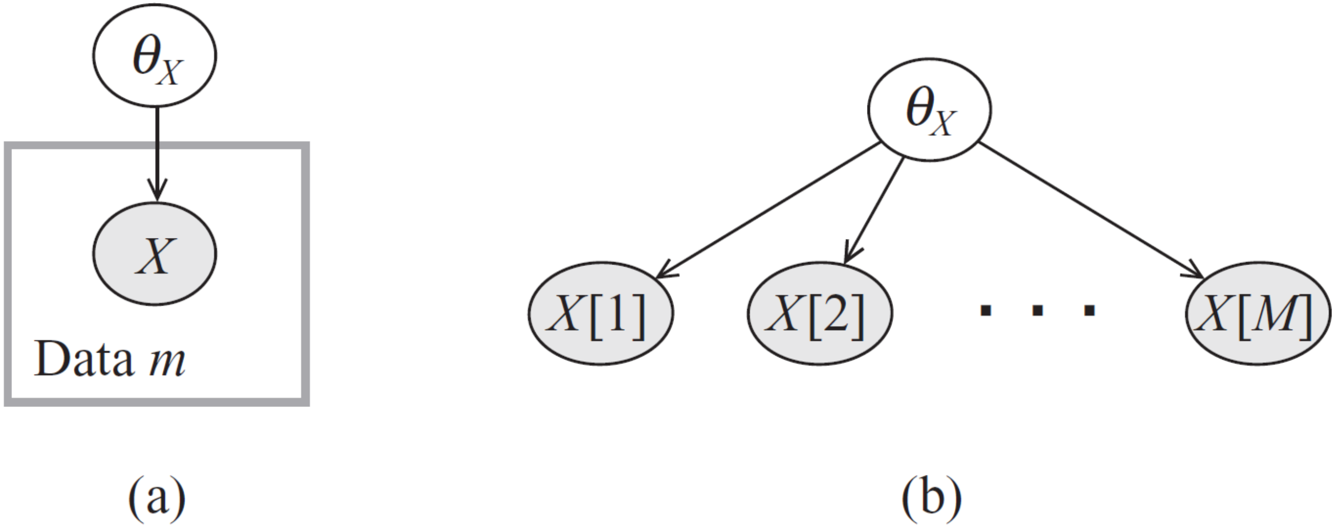
(a) (b)

- $P(X[i] = x^1 \mid \theta_x) = \theta$; $P(X[i] = x^0 \mid \theta_x) = (1 - \theta)$
- $p(\theta_X)$?
  - A continuous distribution over the interval [0,1]

Posterior and prediction

- We are interested in
  - The probability of the next instance, given data
    - $P(x[M+1] \mid D)$
  - The posterior distribution of θ given data
    - $p(\theta \mid D)$

Factorization



$$
\begin{aligned}
P(x[1], x[2], \ldots, x[M], \theta_X) &= P(\theta_X) \prod_{m=1}^{M} P(x[m] \mid \theta_X) \\
&= P(\theta_X) \theta^{M[1]} (1-\theta)^{M[0]}
\end{aligned}
$$

POSTERIOR AND P(X[M+1] | D)

Posterior distribution

$$P(\theta_X \mid D) = \frac{P(x[1],\ldots,x[M] \mid \theta_X)P(\theta_X)}{P(x[1],\ldots,x[M])}$$

$$\begin{aligned} P(x[M+1] \mid D) &= \int_0^1 P(x[M+1] \mid \theta_X, x[1],\ldots x[M])P(\theta_X \mid x[1],\ldots,x[M])d\theta \\ &= \int_0^1 \underbrace{P(x[M+1] \mid \theta_X)}_{\theta \text{ or } 1-\theta \text{ (if binary)}} \underbrace{P(\theta_X \mid x[1],\ldots,x[M])}_{\text{Posterior}} d\theta \end{aligned}$$

Think of taking a weighted average

P(X[M+1] | D)

$$P(x[M+1] \mid x[1],\ldots,x[M]) \;=\; \int_0^1 P(x[M+1] \mid \theta_X) P(\theta_X \mid x[1],\ldots,x[M]) d\theta$$

$$= \int_0^1 P(x[M+1] \mid \theta_X) \frac{P(\theta_X) P(x[1],\ldots,x[M] \mid \theta_X)}{P(x[1],\ldots,x[M])}$$

P(x[1], …, x[M]) is a constant

$$P(x[M+1] \mid x[1],\ldots,x[M]) \propto \int_0^1 P(x[M+1] \mid \theta_X) P(\theta_X) P(x[1],\ldots,x[M] \mid \theta_X) d\theta$$

UNIFORM PRIOR

- We have a uniform prior over θ_X. That is, $p(\theta_X)=1$
- $P(X[M+1]=x^1 \mid x[1],\ldots,x[M])$?

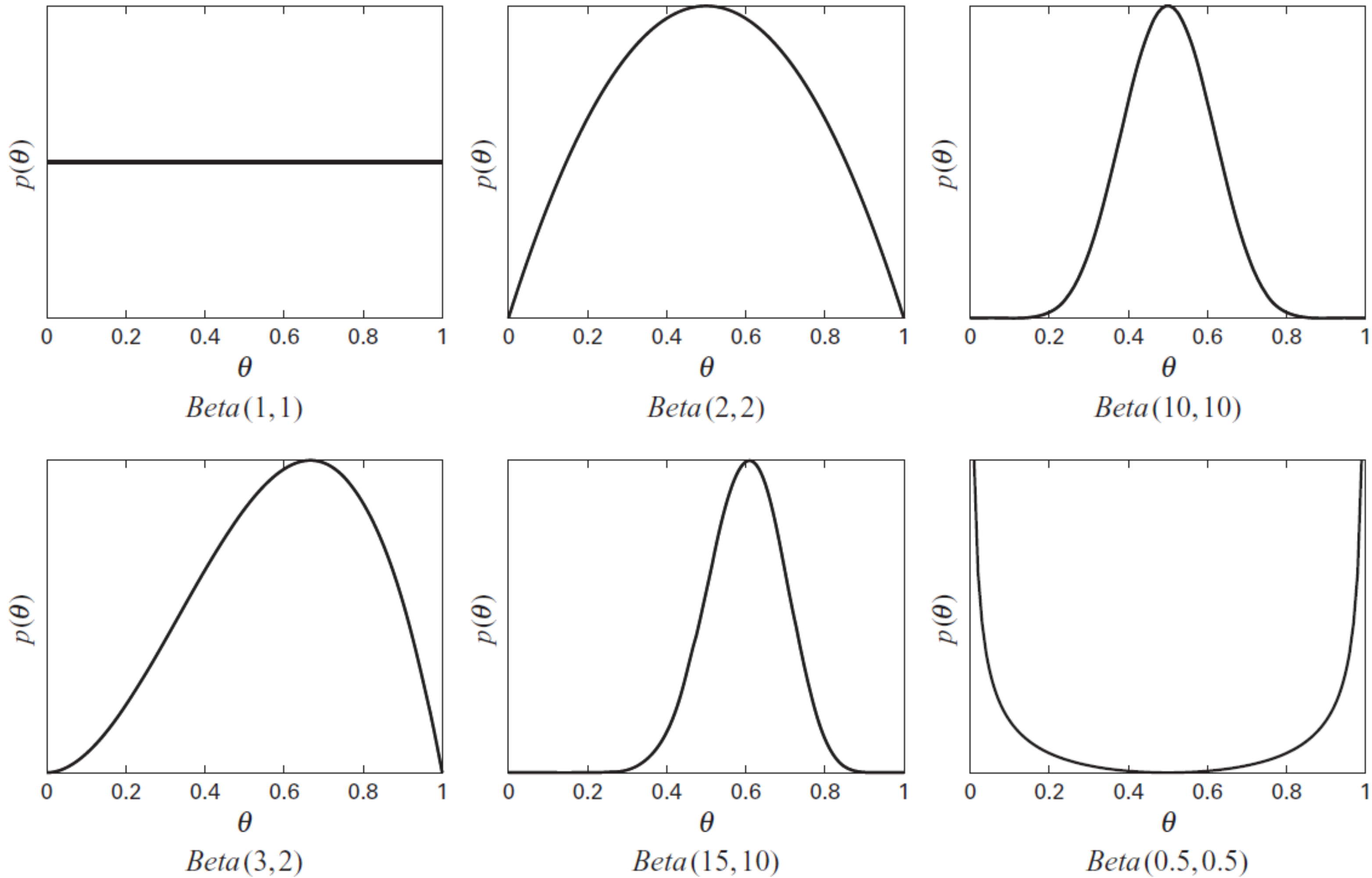
Uniform prior

- We have a uniform prior over θ_X. That is, $p(\theta_X)=1$
- $P(X[M+1]=x^1 \mid x[1],\ldots,x[M])$? That is, $P(X[M+1]=x^1 \mid D)$?
- For the binary case, $P(X[M+1]=x^1 \mid D) = (t + 1) / (t + f + 2)$, where t is the number of True cases and f is the number of False cases in D
- This is also called *Laplace smoothing*
- What about the posterior, $P(\theta \mid D)$, if the prior $P(\theta)$ is uniform?

Beta Distribution

- $\theta \sim \text{Beta}(\alpha,\beta)$ if $p(\theta) = \gamma\theta^{\alpha-1}(1-\theta)^{\beta-1}$ where γ is a normalizing constant
- Mean: $\alpha/(\alpha+\beta)$
- Mode: $(\alpha-1)/(\alpha+\beta-2)$
- Note that the mode is closer to the mean when α and β are large
- Read more at
  - https://en.wikipedia.org/wiki/Beta_distribution

BETA DISTRIBUTION



Beta distribution

- What is $P(X[M+1]=x^1 \mid D)$ if the prior is Beta(α,β)?
  - $P(X[M+1]=x^1 \mid D) = (p + \alpha) / (p + \mathrm{n} + \alpha + \beta)$
- What is the posterior, $P(\theta \mid D)$, if the prior is Beta(α,β)?
  - $P(\theta \mid D) = \mathrm{Beta}(p + \alpha, n + \beta)$
- α and β work like pseudo-counts for the positive and negative cases respectively
- What values to choose for α and β?
  - It depends on our belief and the strength of our belief

DIRICHLET PRIORS

- Generalizes the Beta distribution for multinomials

$$\theta \sim Dirichlet(\alpha_1, \ldots, \alpha_K) \text{ if } P(\theta) \propto \prod_{k=1}^{K} \theta_k^{\alpha_k - 1}$$

- What is P($X[M$+1]=x^i | D) if the prior is Dirichlet?
  - P($X[M$+1]=x^i | D) = (n_i+α_i) / (| D | + α) where n_i is the number of times the i^{th} case appears in D and $\alpha = \alpha_1 + \alpha_2 + \ldots + \alpha_K$
- What is the posterior, P(θ | D), if the prior is Dirichlet?
  - P(θ | D) = Dirichlet($n_1 + \alpha_1$, $n_2 + \alpha_2$, …, $n_K + \alpha_K$)

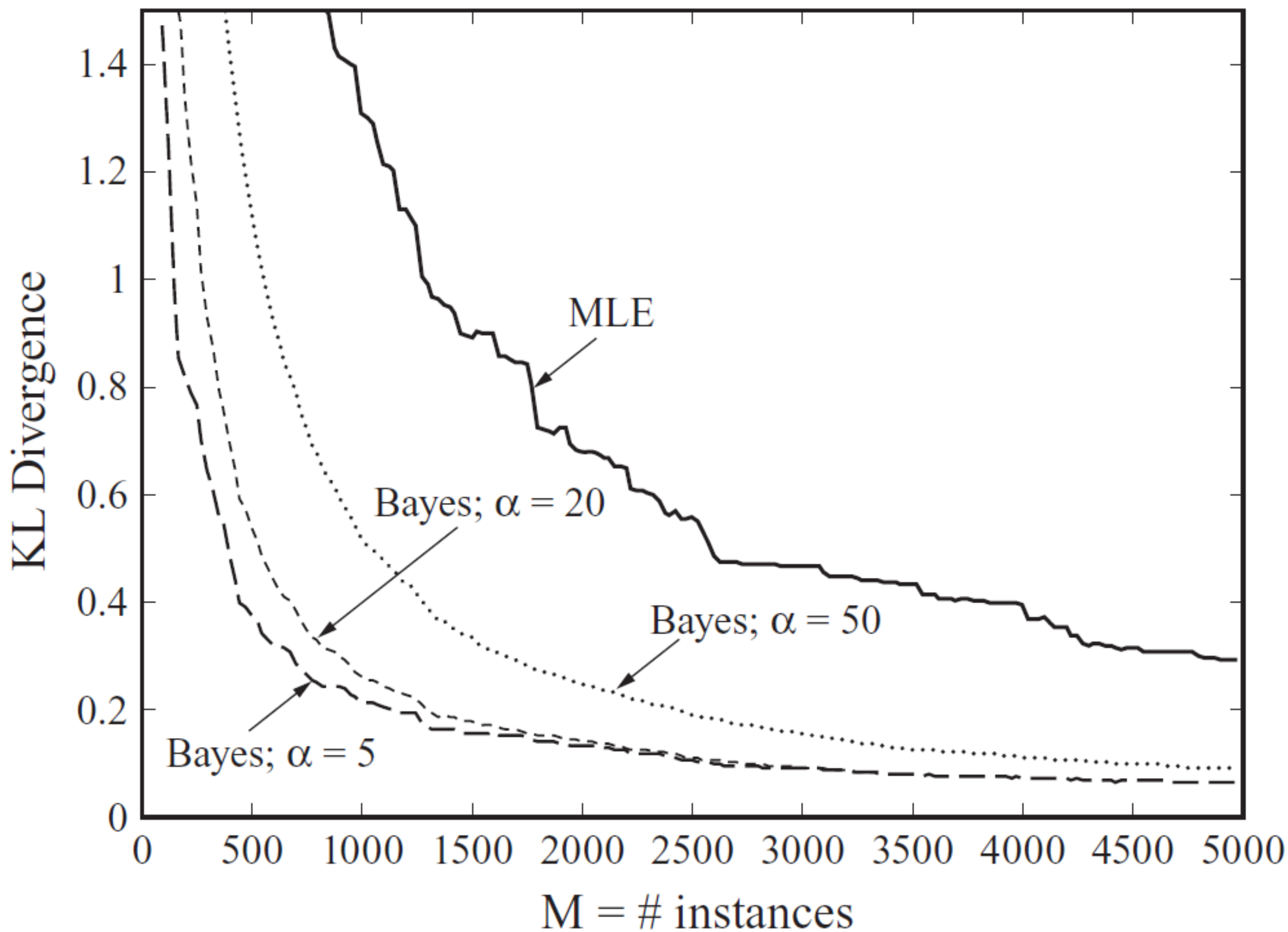
Bayesian estimation

- In MLE for BNs, we optimized each parameter independently
- Can we do the same for Bayesian estimation for BNs?
  - Only if the prior also factorizes wrt the BN
- What about the priors? How do we choose them?
  1. Ask the prior for each variable to an expert
  2. Use the same prior for all variables
     - This is called the *K2 prior*
  3. Imagine a dataset D' of imaginary instances
     - The number of imaginary instances for x is |D'|*P'(x, pa(x))
     - This is called the *BDe prior*
     - What is P'?
       - Could be anything; e.g., a marginally independent distribution

Bayesian estimation examples

- Try a dataset using
  - MLE
  - Bayesian
    - K2
    - BDe

ICU Alarm network – fig 17.c.1



Incomplete Data

MISSING DATA

1. Accidental
   - E.g., sensor failure
2. Intentional
   - E.g., not all tests are ordered for medical diagnosis
3. Hidden variable
   - E.g., cluster assignment; hidden cause

APPROACHES

1. Ignore the data points with missing values
   - Not economical (throwing data away), not necessarily accurate (if missing intentionally), and might not be possible (hidden variables)
2. Imputing
3. Gradient optimization
4. Expectation Maximization

Examples

- Simple network
  - $X \rightarrow Y$
  - Consider these cases:
    - Missing completely at random (MCAR), missing based on a condition
- A more complicated network:
  - A disease (D) variable and three test variables (T_1, T_2 and T_3)
  - T_1 is the fastest but least accurate test. T_2 requires more time but is more accurate. T_3 requires the most time but also is the most accurate test.
  - The doctors order T_1 for everyone first. Depending on the results, they might order T_2, and depending on its results, they might order T_3

Expectation Maximization (EM)

- Initialize θ
- Iterate
  - Expectation
    - $M[x, pa(x)] = \sum P(x, pa(x) \mid observed)$ for each X
  - Maximization
    - $\theta_x = M[x, pa(x)] / M[pa(x)]$ for each X

Markov Networks

Overview

- Compared with Bayesian networks, the same principles apply but the issues and solutions are quite different
- The most important reason for the differences is the use of global normalization constant Z
- Z couples all parameters together, preventing us from optimizing each parameter independently
- Even simple maximum likelihood parameter estimation does not have a closed form solution
- We often resort to iterative approaches such as gradient ascent
- The good news is that the likelihood objective is concave; iterative approaches converge to global optimum

SINGLE VARIABLE CASE

X	D	$\phi(X)$
T	a	θ_1
F	b	θ_2

$Z = \theta_1 + \theta_2$

- We have a binary variable X with domain(X) = {T, F}.
- Parameters are θ_1 and θ_2
  - Remember, the only constraint on θ_1 and θ_2 is that they need to be non-negative
- Dataset D has a T and b F instances.
- What are the maximum likelihood estimates for θ_1 and θ_2?

$$P(X=T)^a \cdot P(X=F)^b$$

$$\left(\frac{\theta_1}{\theta_1+\theta_2}\right)^a \left(\frac{\theta_2}{\theta_1+\theta_2}\right)^b$$

$\theta_1 = a \cdot k$

$\theta_2 = b \cdot k$

LOG-LINEAR MODELS

- A distribution is a *log-linear model* over a Markov network $\mathcal{H}$ is it is associated with
  - A set of features $\mathcal{F} = \{f_1(\mathbf{C}_1), \ldots, f_k(\mathbf{C}_k)\}$, where each $\mathbf{C}$ is a complete subgraph in $\mathcal{H}$,
  - A set of weights $w_1, \ldots, w_k$

$$P(X_1, \ldots, X_n) = \frac{1}{Z(\mathbf{w})} e^{-\sum_{i=1}^{k} w_i f_i(\mathbf{C}_i)}$$

  - It is common to have several features over the same scope

LOG-LINEAR MODELS – LOG-LIKELIHOOD

- Given a domain $\mathcal{X}$={X_1, …, X_n} and a dataset $\mathcal{D}$ = {ξ[1], ξ[2], …, ξ[M]}, where ξ[i] is an instance, i.e., a complete assignment to the variables $\mathcal{X}$

- The log-likelihood is

$$l(\mathbf{w}:D) = -\sum_i w_i \left(\sum_m f_i(\xi[m]) \right) - M \ln Z\,(\mathbf{w})$$

- We are abusing the notation a little for clarity. The feature functions are defined over cliques, but here we passed them the whole instance. They just ignore the irrelevant portions of the instance

DERIVATIVE OF $l(\mathbf{w}:D)/M$ WRT TO w_i

$$\frac{\partial}{\partial w_i}\frac{1}{M}l(\mathbf{w}:D) = \mathbf{E}_{\mathbf{w}}[f_i] - \mathbf{E}_D[f_i]$$

Let's prove it.

$\mathbf{E}_{\mathbf{w}}[f_i] - \mathbf{E}_D[f_i]$

- $\mathbf{E}_D[f_i]$ can be computed by summing f_i over the instances in D and dividing by M
- How can we compute $\mathbf{E}_{\mathbf{w}}[f_i]$?
- $\mathbf{E}_{\mathbf{w}}\left[f_i\right] = \sum_{x} P_{\mathbf{w}}(\mathcal{X}) f_i(\mathcal{X})$
- It is impossible to iterate over all possible values of $\mathcal{X}$
- Remember f_i is defined over a set of variables $\mathbf{C}_i$ and it ignores (i.e. equals zero for) the rest of the variables
- $\mathbf{E}_{\mathbf{w}}\left[f_i\right] = \sum_{\mathbf{c}_i} P_{\mathbf{w}}(\mathbf{c}_i) f_i(\mathbf{c}_i)$
- Perform inference to compute $P_{\mathbf{w}}(\mathbf{c}_i)$
- Now we have the gradient, we can optimize $\mathbf{w}$ using gradient ascent

GRADIENT ASCENT

$\vec{w} = \langle w_1, \ldots, w_n \rangle$

- Find maximum of *f(*θ*)* where there is no closed form solution
- Start with some initial guess θ_0

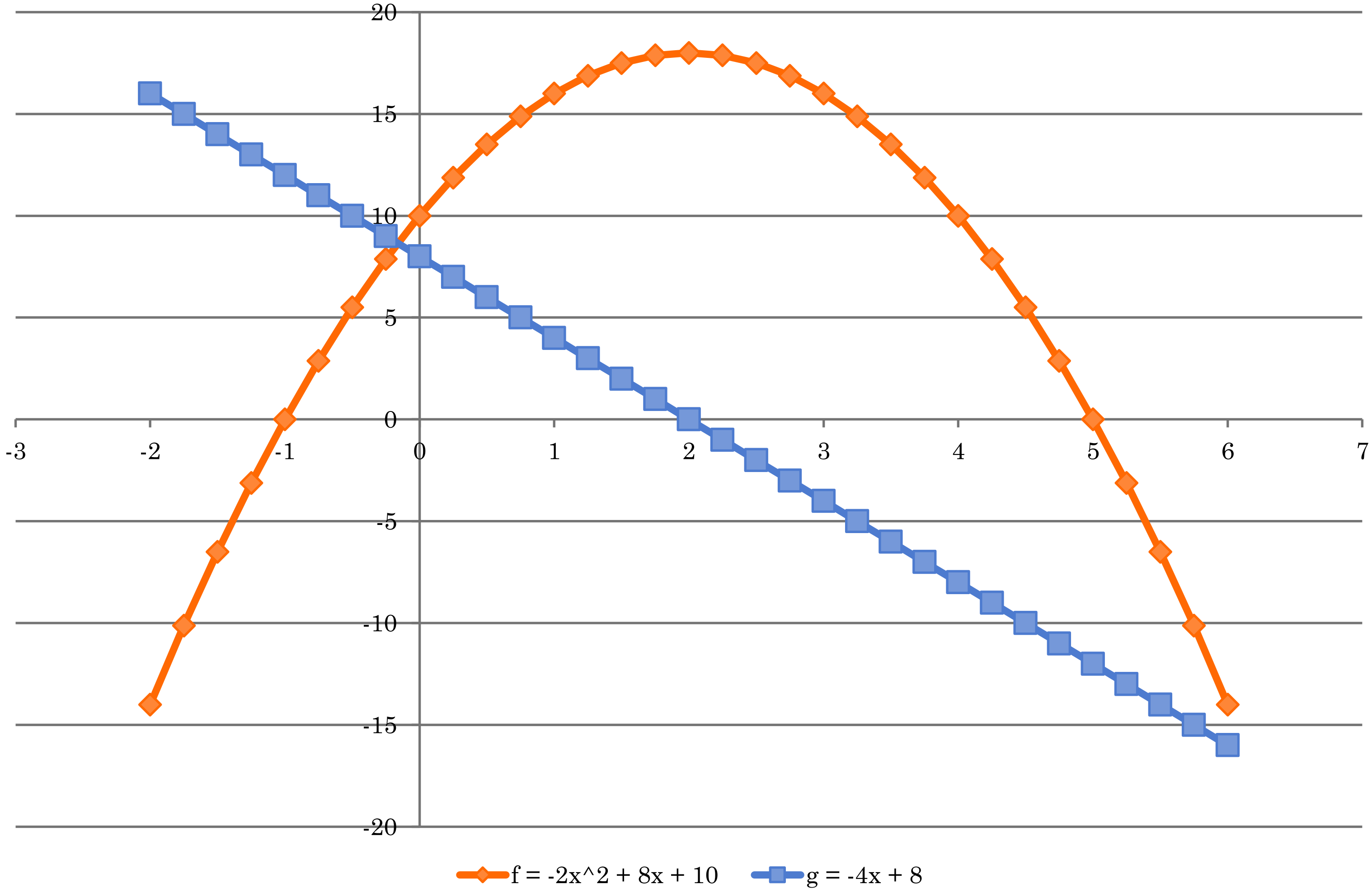
  $\vec{w}[0]$

- While change is not much
  - $\theta_{i+1} = \theta_i + \eta * f'(\theta_i)$

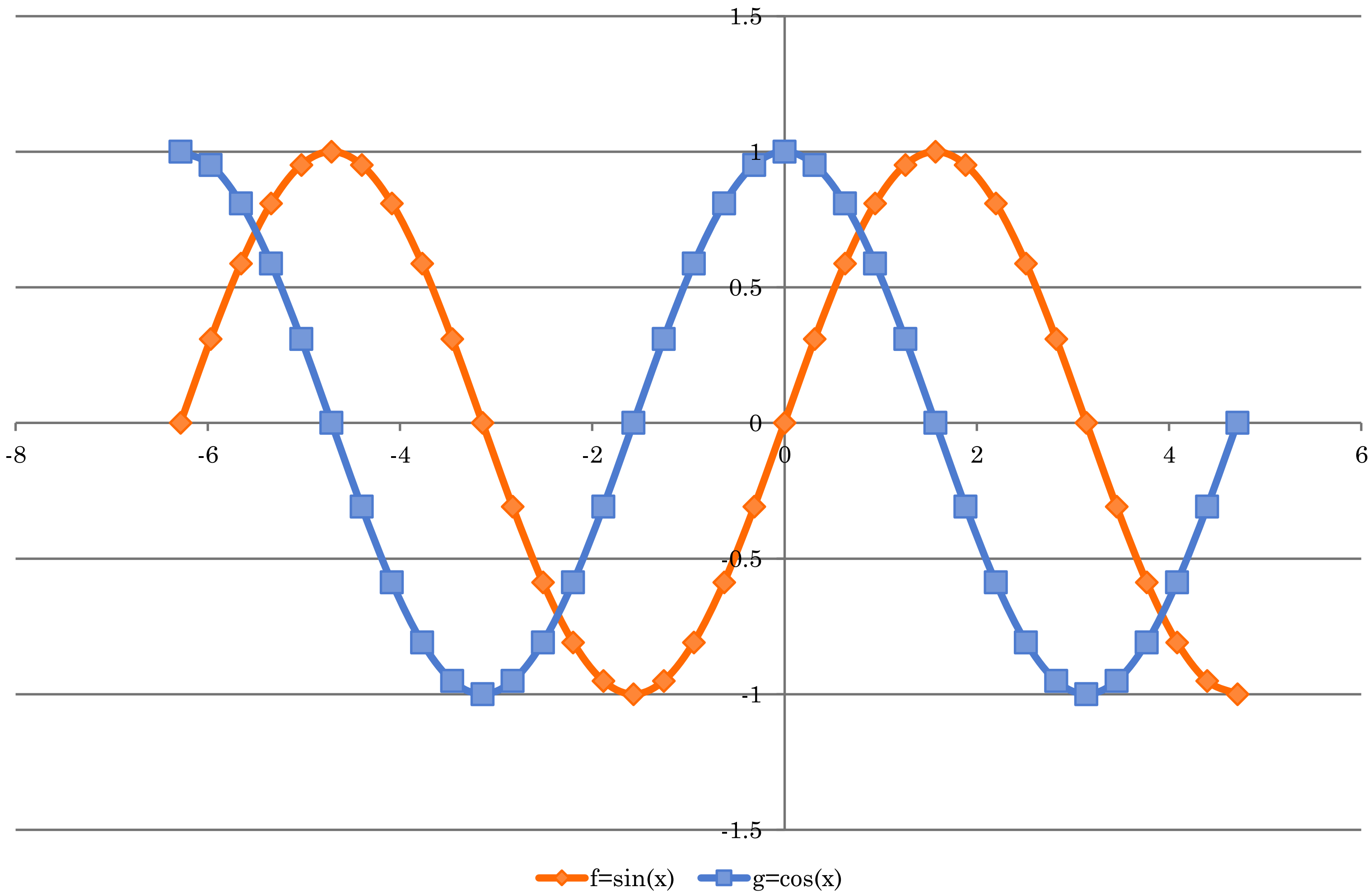
  $\vec{w}[1] = \vec{w}(0) + \eta \cdot \frac{\partial LL/m}{\partial \vec{w}(0)}$

- η is called the learning rate and it is a user specified parameter

$f = -2x^2 + 8x + 10$



f = sin(x)



$f = -2x^2 + 8x + 10$

- $f' = -4x + 8$
- Start with $x_0 = 6$
- Use $\eta = 0.2$
- $x_1 = x_0 + \eta * f'(x_0)$
  - $x_1 = 6 + 0.2*(-4*6 + 8) = 6 – 0.2*16 = 6 – 3.2 = 2.8$
- $x_2 = x_1 + \eta * f'(x_1)$
  - $x_2 = 2.8 + 0.2*(-4*2.8 + 8) = 2.8 – 0.2*3.2 = 2.8 – 0.64 = 2.16$
- $x_3 = x_2 + \eta * f'(x_2)$
  - $x_3 = 2.16 + 0.2*(-4*2.16 + 8) = 2.16 – 0.2*0.64 = 2.16 – 0.128 = 2.032$
- $x_4 = x_3 + \eta * f'(x_3)$
  - $x_4 = 2.032 + 0.2*(-4*2.032 + 8) = 2.032 – 0.2*0.128 = 2.032 – 0.0256 = 2.0064$

BAYESIAN PRIORS

- There was no closed form solution for the maximum likelihood formulation
- There is no closed form solution for full Bayesian approach either
- We instead find the parameters that maximize $p(\theta)P(D \mid \theta)$

$p(\theta \mid D)$

$p(x_{new} \mid D)$

mode of the poster

$p(\theta)$?

$p(\vec{w})$

$\arg\max_{\vec{w}} p(\vec{w} \mid D)$

$\arg\max_{\theta} \frac{f(\theta)}{10}$

L_2-Regularization

- Assume $p(\mathbf{w})$ is zero-mean diagonal Gaussian with equal variances
  $\mathcal{N}\left(\langle 0, 0, 0 \cdots 0 \rangle, \begin{pmatrix} \sigma^2 & 0 & \cdots \\ & \ddots & \\ \cdots & & \sigma^2 \end{pmatrix}\right)$
- In the log space, it gives rise to a penalty of the form

$$\frac{1}{2\sigma^2}\sum_{i=1}^{k} w_i^2$$

$$\underset{\vec{w}}{\operatorname{argmax}}\ f(\vec{w}) = \underline{LL} - \underline{\frac{1}{2\sigma^2}\sum_i w_i^2}$$

- It penalizes large weights

$$\frac{\partial f}{\partial w_i} = \frac{\partial LL}{\partial w_i} - \underline{\frac{1}{2\sigma^2} \cdot 2 \cdot w_i}$$

L_1-Regularization

$$\arg\max_{w} p(w) P(D|w)$$

- Assume p(**w**) is zero-mean Laplace distribution
- In the log space, it gives rise to a penalty of the form

$$\frac{1}{\beta}\sum_{i=1}^{k}|w_i|$$

$$\arg\max LL - \frac{1}{\beta}\sum|w_i|$$

- It penalizes large weights

WHY SMALL WEIGHTS?

$1 \qquad 10$

$e^{\varepsilon} \qquad e^{10\varepsilon}$

- The prior for the weights is zero; unless the data tells us otherwise, we will treat the feature as not needed by the model
- Large weights are more susceptible to the noise in the data
  - A small difference in the feature value can cause big changes in the probability
- Small weights give rise to smoother probabilities

L_2 VS L_1

- L_2 forces the large weights to get closer to zero and places an emphasis on the large weights
  - Even though the weights get closer to zero, they are often not zero
- L_1 also penalizes large weights but the emphasis is not necessarily on the large weights
  - Some of the weights become zero
  - Leads to a sparser representation

LEARNING RATE

- As we have seen with examples, the learning rate is an important parameter
- If it is too large, then we can overshoot
- If it is too small, then it takes a long time to converge
- There is no single value that works for all datasets and domains
- There are approaches that chooses an appropriate learning rate at each step
  - E.g., line search

MN Parameter estimation summary

- There is no closed form solution for both maximum likelihood estimation and Bayesian estimation
- The likelihood function is concave; there is a single global optimum (note that the objective function has a single global optimum value but there might be many parameter values that achieve the same global optimum)
- Gradient ascent methods are applied to estimate the parameters
- The gradient computation requires running inference, which is costly
- In practice, regularization (L_2, L_1) is used to avoid overfitting