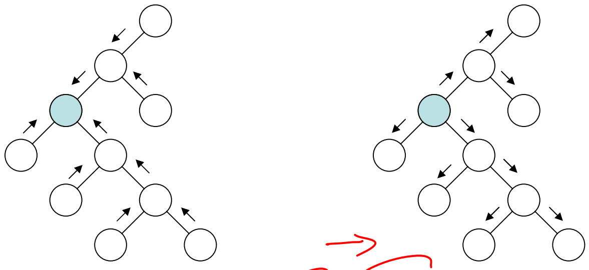
A Sketch of the Junction Tree Algorithm

- **The algorithm**

  1. Moralize the graph (trivial)
  2. Triangulate the graph by elimination (good heuristic exist, but actually NP hard)
  3. Build a clique tree (using a maximum spanning tree algorithm
  4. Propagation of probabilities --- a local message-passing protocol

- Thm: Maximum-spanning tree of cliques ←→ Junction tree
- Results in marginal probabilities of all cliques --- solves all queries in a single run
- A **generic** exact inference algorithm for any GM
- **Complexity**: exponential in the size of the maximal clique --- a good elimination order often leads to small maximal clique, and hence a good (i.e., thin) JT

Message-passing algorithms



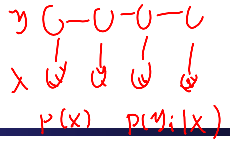
- Message update
  - The Hugin update

    $$\phi_S^* = \sum_{V\setminus S}\psi_V \qquad \psi_W^* = \frac{\phi_S^*}{\phi_S}\psi_W$$

  - The Shafer-Shenoy update

    $$m_{i\to j}(S_{ij}) = \sum_{C_i\setminus S_{ij}}\psi_{C_i}\prod_{k\neq j} m_{k\to i}(S_{ki})$$

$y_t = \begin{pmatrix} \vdots \end{pmatrix}$ y_t^i



The Forward Algorithm

- We want to calculate $P(\mathbf{x})$, the likelihood of $\mathbf{x}$, given the HMM
  - Sum over all possible ways of generating $\mathbf{x}$:

$$p(\mathbf{x}) = \sum_{\mathbf{y}} p(\mathbf{x},\mathbf{y}) = \sum_{y_1}\sum_{y_2}\cdots\sum_{y_N} \pi_{y_1}\prod_{t=2}^{T} a_{y_{t-1},y_t}\prod_{t=1}^{T} p(x_t \mid y_t)$$

  - To avoid summing over an exponential number of paths $\mathbf{y}$, define

$$\alpha(y_t^k = 1) = \alpha_t^k \overset{\text{def}}{=} P(x_1,\ldots,x_t,y_t^k = 1)$$

(the forward probability)

$P(x) = \sum_k \alpha_T^k$

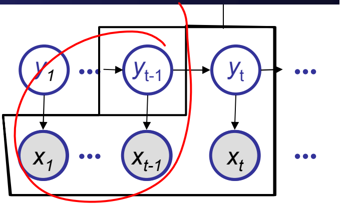
  - The recursion:

$$\alpha_t^k = p(x_t \mid y_t^k = 1)\sum_i \alpha_{t-1}^i a_{i,k}$$

$$P(\mathbf{x}) = \sum_k \alpha_T^k$$

The Forward Algorithm – derivation

- Compute the forward probability:



α_{t-1}^{i}
$\shortparallel$
$p(x_1 \cdots x_{t-1}, y_{t-1}^i = 1)$

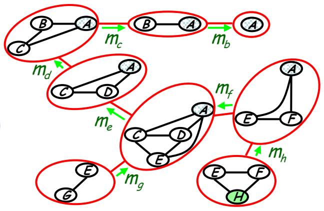
$$\begin{aligned}
\alpha_t^k &= P(x_1,\ldots,x_{t-1},x_t,y_t^k = 1)\\
&= \sum_{y_{t-1}} P(x_1,\ldots,x_{t-1},x_t,y_{t-1},y_t^k = 1)\\
&= \sum_{y_{t-1}} P(x_1,\ldots,x_{t-1},y_{t-1})P(y_t^k = 1 \mid y_{t-1},x_1,\ldots,x_{t-1})P(x_t \mid y_t^k = 1,x_1,\ldots,x_{t-1},y_{t-1})\\
&= \sum_{y_{t-1}} P(x_1,\ldots,x_{t-1},y_{t-1})P(y_t^k = 1 \mid y_{t-1})P(x_t \mid y_t^k = 1)\\
&= P(x_t \mid y_t^k = 1)\sum_i P(x_1,\ldots,x_{t-1},y_{t-1}^i = 1)P(y_t^k = 1 \mid y_{t-1}^i = 1)\\
&= P(x_t \mid y_t^k = 1)\sum_i \alpha_{t-1}^i a_{i,k}
\end{aligned}$$

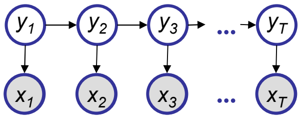
Chain rule: $P(A,B,C) = P(A)P(B \mid C)P(C \mid A,B)$

Recall the Elimination and Message Passing Algorithm

- Elimination ≡ message passing on a **clique tree**

$$m_e(a,c,d) = \sum_e p(e \mid c,d) m_g(e) m_f(a,e)$$





$$P(\mathbf{x}) = \sum_k \alpha_T^k$$

$$\alpha_t^k = p(x_t \mid y_t^k = 1) \sum_i \alpha_{t-1}^i a_{i,k}$$

The Forward Algorithm

- We can compute α_t^k for all k, t, using dynamic programming!

Initialization:

$$\alpha_1^k = P(x_1 \mid y_1^k = 1)\pi_k$$

$$\alpha_1^k = P(x_1, y_1^k = 1)$$
$$= P(x_1 \mid y_1^k = 1) P(y_1^k = 1)$$
$$= P(x_1 \mid y_1^k = 1)\pi_k$$

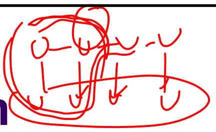
Iteration:

$$\alpha_t^k = P(x_t \mid y_t^k = 1) \sum_i \alpha_{t-1}^i a_{i,k}$$

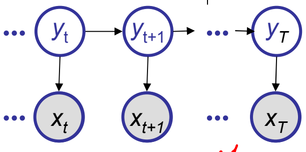
Termination:

$$P(\mathbf{x}) = \sum_k \alpha_T^k$$

The Backward Algorithm



- We want to compute $P(y_t^k=1\,|\,\mathbf{x})$, the posterior probability distribution on the t th position, given $\mathbf{x}$
  - We start by computing



$P(X_1 \cdot\cdot X_t \cdot y_t^k=1)$

A

$$\frac{P(y_t^k=1,\mathbf{x})}{P(X)} = P(x_1,\ldots,x_t,y_t^k=1,x_{t+1},\ldots,x_T)$$

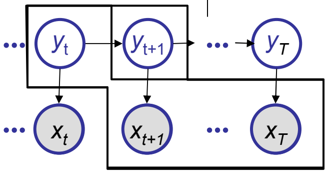
$$= P(x_1,\ldots,x_t,y_t^k=1)P(x_{t+1},\ldots,x_T\,|\,x_1,\ldots,x_t,y_t^k=1)$$

$$= \underbrace{P(x_1\ldots x_{t,}y_t^k=1)}_{\text{Forward, } \alpha_t^k}\underbrace{P(x_{t+1}\ldots x_T\,|\,y_t^k=1)}_{\text{Backward, } \beta_t^k = P(x_{t+1},\ldots,x_T\,|\,y_t^k=1)}$$

- The recursion: $\beta_t^k = \sum_i a_{k,i}\,p(x_{t+1}\,|\,y_{t+1}^i=1)\beta_{t+1}^i$ $O(Tn^2)$

The Backward Algorithm – derivation

- Define the backward probability:



$$\beta_t^k = P(x_{t+1},\ldots,x_T\,|\,y_t^k=1)$$

$$= \sum_{y_{t+1}} P(x_{t+1},\ldots,x_T,y_{t+1}\,|\,y_t^k=1)$$

$$= \sum_i P(y_{t+1}^i=1\,|\,y_t^k=1)p(x_{t+1}\,|\,y_{t+1}^i=1,y_t^k=1)P(x_{t+2},\ldots,x_T\,|\,x_{t+1},y_{t+1}^i=1,y_t^k=1)$$

$$= \sum_i P(y_{t+1}^i=1\,|\,y_t^k=1)p(x_{t+1}\,|\,y_{t+1}^i=1)P(x_{t+2},\ldots,x_T\,|\,y_{t+1}^i=1)$$

$$= \sum_i a_{k,i}\,p(x_{t+1}\,|\,y_{t+1}^i=1)\beta_{t+1}^i$$

Chain rule: $P(A,B,C\,|\,\alpha) = P(A,\alpha)P(B\,|\,C,\alpha)P(C\,|\,A,B,\alpha)$

The Backward Algorithm

- We can compute β_t^k for all k, t, using dynamic programming!

Initialization:

$$\beta_T^k = 1, \ \forall k$$

Iteration:

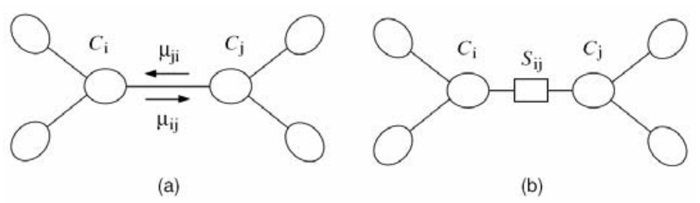
$$\beta_t^k = \sum_i a_{k,i} P(x_{t+1} \mid y_{t+1}^i = 1) \beta_{t+1}^i$$

Termination:

$$P(\mathbf{x}) = \sum_k \alpha_1^k \beta_1^k$$

Shafer Shenoy for HMMs

- Recap: Shafer-Shenoy algorithm



  - Message from clique i to clique j :

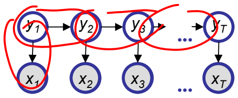
$$\mu_{i \to j} = \sum_{C_i \setminus S_{ij}} \psi_{C_i} \prod_{k \neq j} \mu_{k \to i}(S_{ki})$$

  - Clique marginal

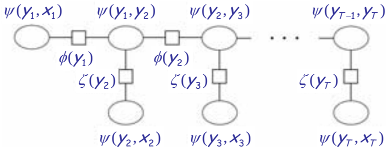
$$p(C_i) \propto \psi_{C_i} \prod_k \mu_{k \to i}(S_{ki})$$

Message Passing for HMMs (cont.)

- A junction tree for the HMM



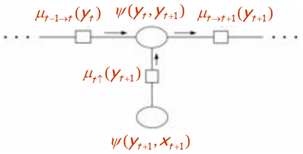
$\Rightarrow$



- Rightward pass

$$\mu_{t\to t+1}(y_{t+1}) = \sum_{y_t}\psi(y_t,y_{t+1})\mu_{t-1\to t}(y_t)\mu_{t\uparrow}(y_{t+1})$$
$$= \sum_{y_t} p(y_{t+1}\mid y_t)\mu_{t-1\to t}(y_t)p(x_{t+1}\mid y_{t+1})$$
$$= p(x_{t+1}\mid y_{t+1})\sum_{y_t} a_{y_t,y_{t+1}}\mu_{t-1\to t}(y_t)$$

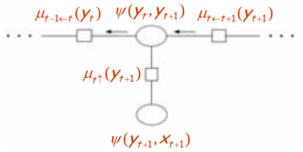
  - This is exactly the *forward algorithm*!



- Leftward pass …

$$\mu_{t-1\leftarrow t}(y_t) = \sum_{y_{t+1}}\psi(y_t,y_{t+1})\mu_{t\leftarrow t+1}(y_{t+1})\mu_{t\uparrow}(y_{t+1})$$
$$= \sum_{y_{t+1}} p(y_{t+1}\mid y_t)\mu_{t\leftarrow t+1}(y_{t+1})p(x_{t+1}\mid y_{t+1})$$

  - This is exactly the *backward algorithm*!

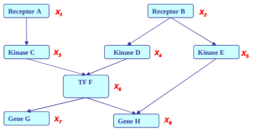

$\mu_{t-1\leftarrow t}(y_t)$
$\psi(y_t,y_{t+1})$
$\mu_{t\leftarrow t+1}(y_{t+1})$
$\mu_{t\uparrow}(y_{t+1})$
$\psi(y_{t+1},x_{t+1})$

School of Computer Science
Carnegie Mellon

Statistical learning with basic graphical models

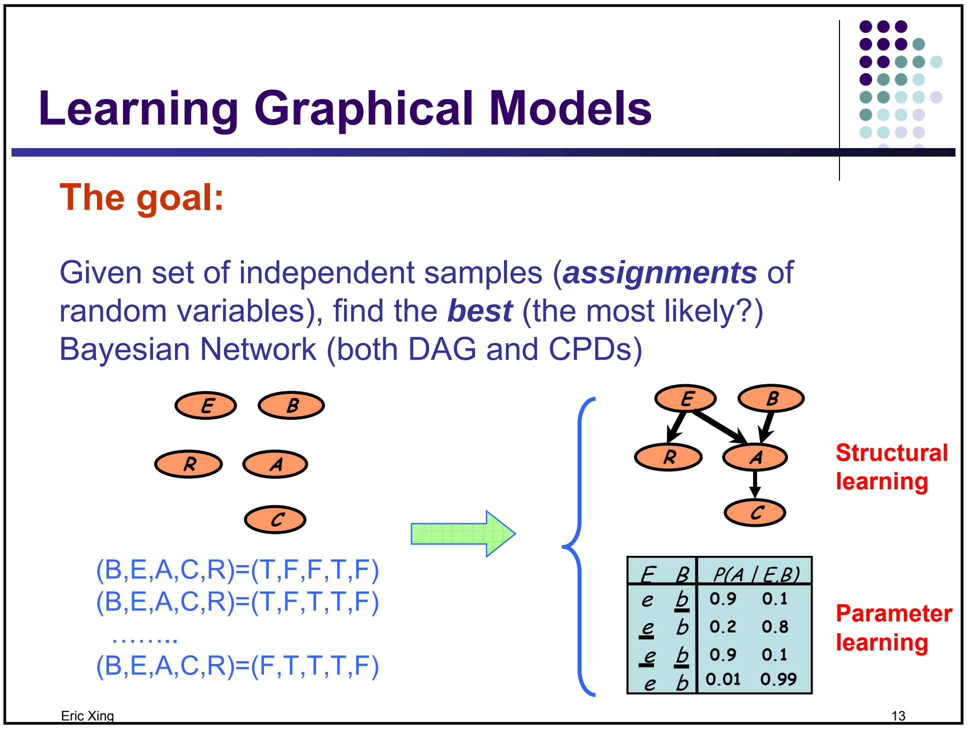
Probabilistic Graphical Models (10-708)

Lecture 7, Oct 8, 2007



Eric Xing

Reading: J-Chap. 5,6, KF-Chap. 8



Learning Graphical Models

The goal:

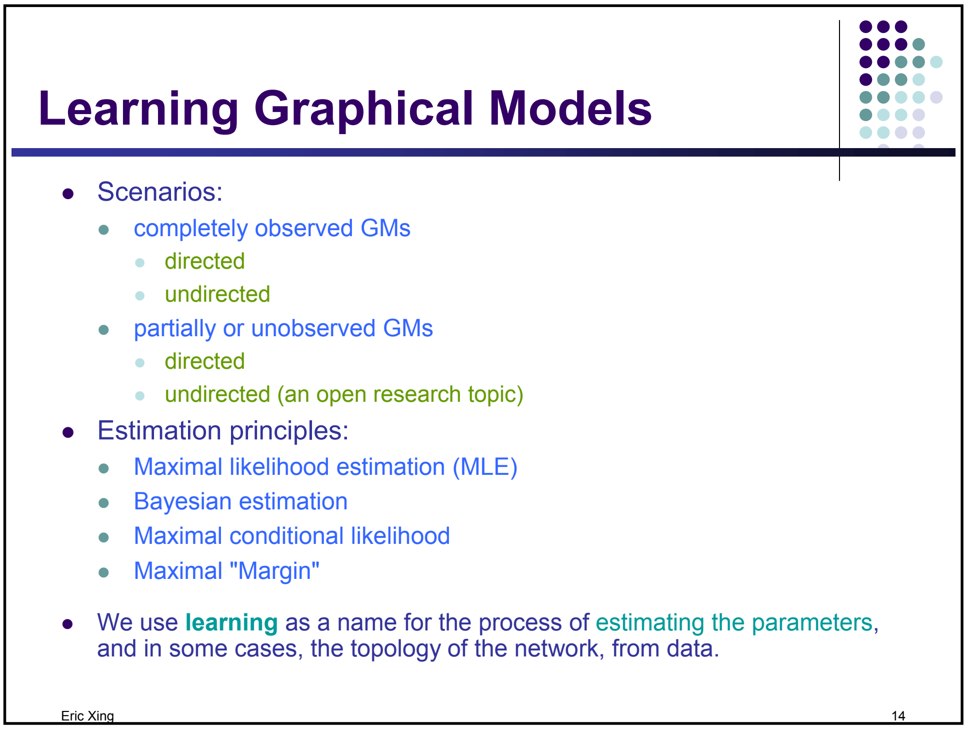
Given set of independent samples (***assignments*** of random variables), find the ***best*** (the most likely?) Bayesian Network (both DAG and CPDs)

(B,E,A,C,R)=(T,F,F,T,F)
(B,E,A,C,R)=(T,F,T,T,F)
……..
(B,E,A,C,R)=(F,T,T,T,F)

Structural learning

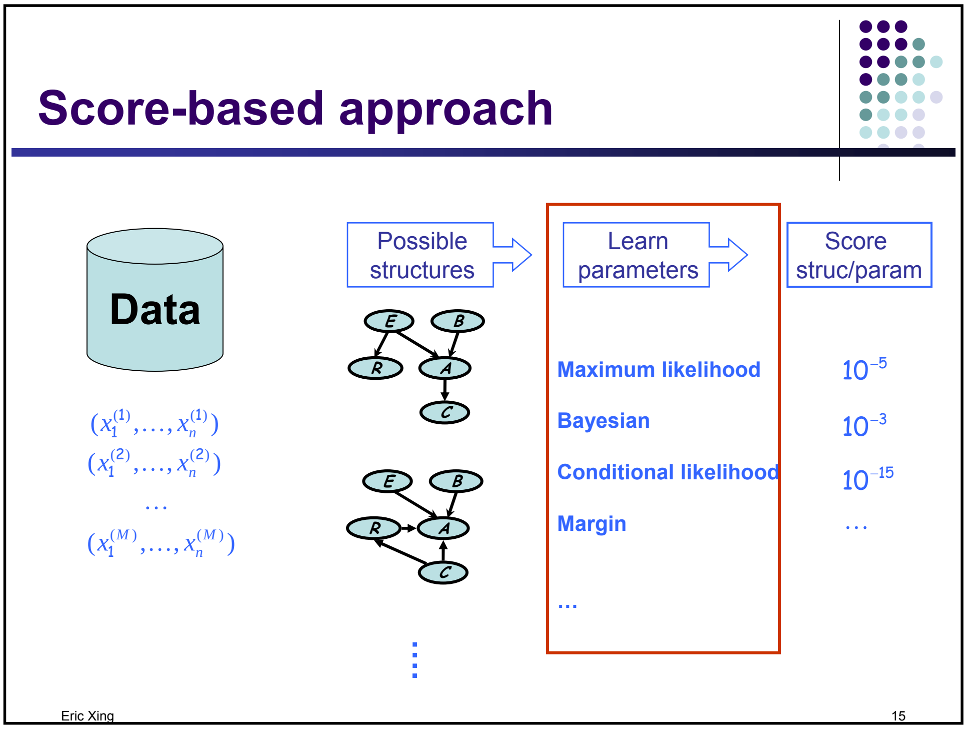
E	B	P(A | E,B)	
e	$\underline{b}$	0.9	0.1
$\underline{e}$	b	0.2	0.8
$\underline{e}$	$\underline{b}$	0.9	0.1
e	b	0.01	0.99

Parameter learning



Learning Graphical Models

- Scenarios:
  - completely observed GMs
    - directed
    - undirected
  - partially or unobserved GMs
    - directed
    - undirected (an open research topic)
- Estimation principles:
  - Maximal likelihood estimation (MLE)
  - Bayesian estimation
  - Maximal conditional likelihood
  - Maximal "Margin"
- We use **learning** as a name for the process of estimating the parameters, and in some cases, the topology of the network, from data.



Score-based approach

Data

$(x_1^{(1)},\dots,x_n^{(1)})$

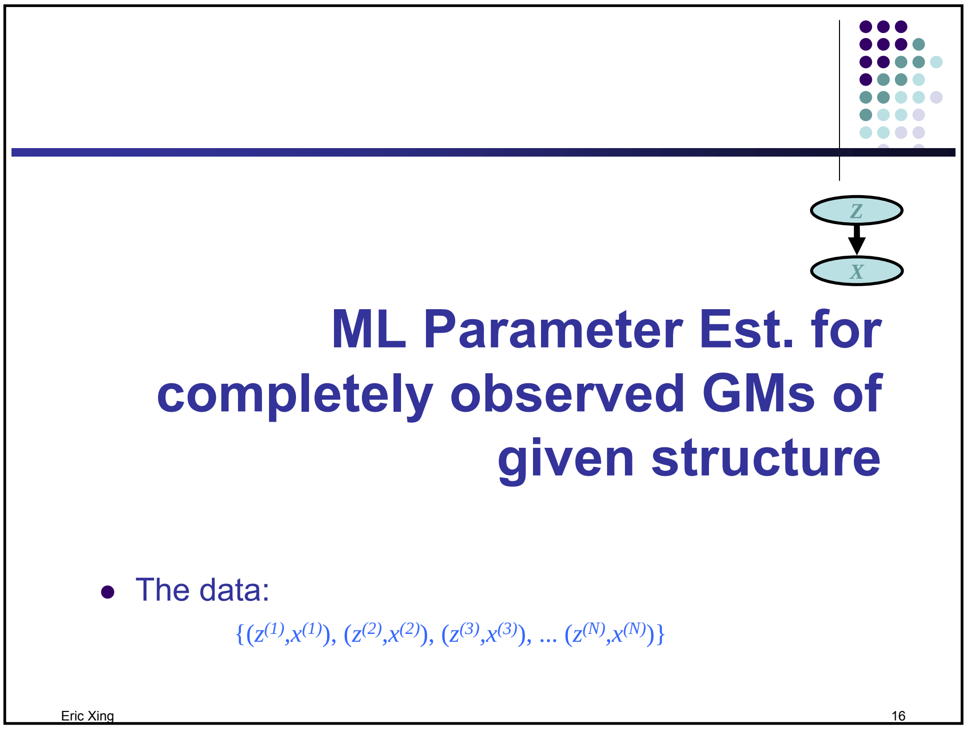
$(x_1^{(2)},\dots,x_n^{(2)})$

$\dots$

$(x_1^{(M)},\dots,x_n^{(M)})$

Possible structures → Learn parameters → Score struc/param

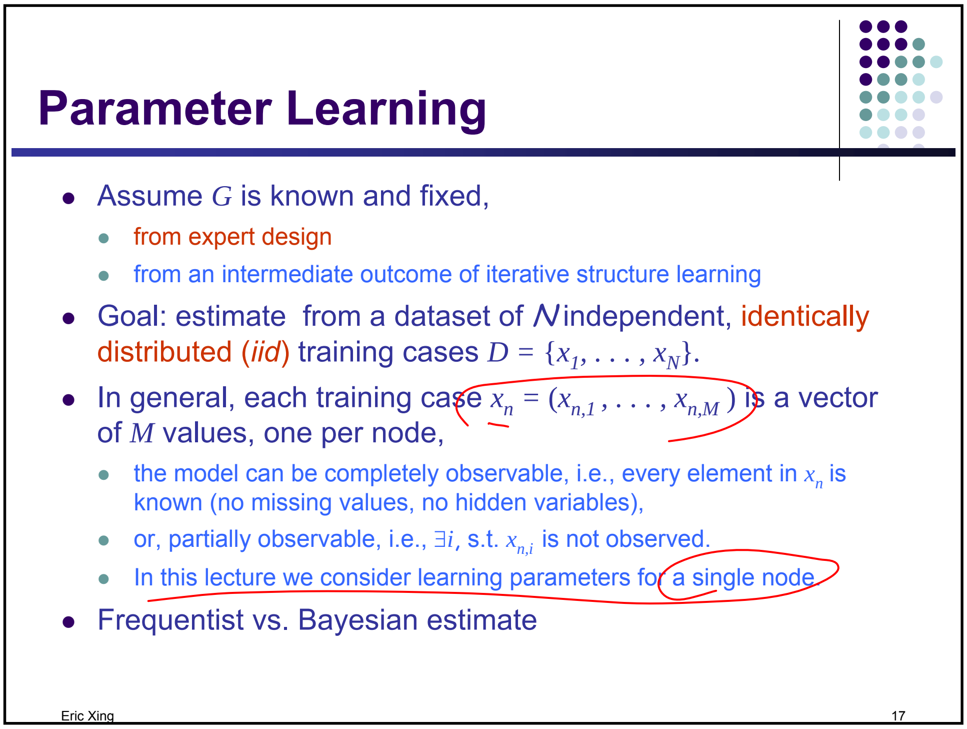
Learn parameters	Score struc/param
Maximum likelihood	10^{-5}
Bayesian	10^{-3}
Conditional likelihood	10^{-15}
Margin	…
…	



ML Parameter Est. for completely observed GMs of given structure

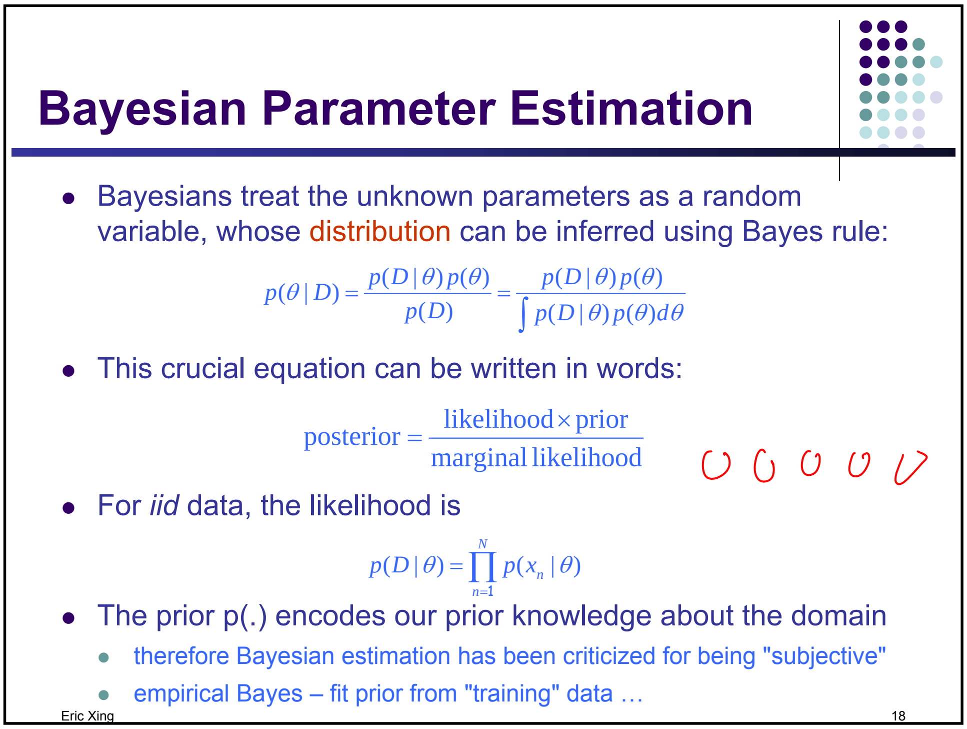
- The data:

  $\{(z^{(1)},x^{(1)}), (z^{(2)},x^{(2)}), (z^{(3)},x^{(3)}), \dots (z^{(N)},x^{(N)})\}$



Parameter Learning

- Assume G is known and fixed,
  - from expert design
  - from an intermediate outcome of iterative structure learning
- Goal: estimate from a dataset of N independent, identically distributed (*iid*) training cases $D = \{x_1, . . . , x_N\}$.
- In general, each training case $x_n = (x_{n,1}, . . . , x_{n,M})$ is a vector of M values, one per node,
  - the model can be completely observable, i.e., every element in x_n is known (no missing values, no hidden variables),
  - or, partially observable, i.e., $\exists i$, s.t. $x_{n,i}$ is not observed.
  - In this lecture we consider learning parameters for a single node
- Frequentist vs. Bayesian estimate



Bayesian Parameter Estimation

- Bayesians treat the unknown parameters as a random variable, whose distribution can be inferred using Bayes rule:

$$p(\theta \mid D) = \frac{p(D \mid \theta)\, p(\theta)}{p(D)} = \frac{p(D \mid \theta)\, p(\theta)}{\int p(D \mid \theta)\, p(\theta) d\theta}$$

- This crucial equation can be written in words:

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}$$

- For *iid* data, the likelihood is

$$p(D \mid \theta) = \prod_{n=1}^{N} p(x_n \mid \theta)$$

- The prior p(.) encodes our prior knowledge about the domain
  - therefore Bayesian estimation has been criticized for being "subjective"
  - empirical Bayes – fit prior from "training" data …

Frequentist Parameter Estimation

Two people with different priors $p(\theta)$ will end up with different estimates $p(\theta|D)$.

- Frequentists dislike this “subjectivity”.
- Frequentists think of the parameter as a fixed, unknown constant, not a random variable.
- Hence they have to come up with different "objective" **estimators** (ways of computing from data), instead of using Bayes’ rule.
  - These estimators have different properties, such as being “unbiased”, “minimum variance”, etc.
- A very popular estimator is the maximum likelihood estimator, which is simple and has good statistical properties.

Discussion

θ or $p(\theta)$, this is the problem!

Maximum Likelihood Estimation

- The log-likelihood is monotonically related to the likelihood:

$$\ell(\theta; D) = \log p(D \mid \theta) = \sum_{n=1}^{N} \log p(x_n \mid \theta)$$

- The Idea underlying maximum likelihood estimation (MLE): pick the setting of parameters most likely to have generated the data we saw: $\widehat{\theta}_{ML} = \arg\max_{\theta} \ell(\theta; D)$

- Problem of MLE:
  - Overfitting: means that "some of the relationships that appear statistically significant are actually just noise. It occurs when the complexity of the statistical model is too great for the amount of data that you have"
  - Often the MLE **overfits** the training data, so it is common to maximize a **regularized** log-likelihood instead: $\widehat{\theta}' = \arg\max_{\theta} \ell(\theta; D) - c(\theta)$
  - Insufficient training data can lead to spurious estimator (e.g., certain possible values are not observed due to data sparsity), so it is common to **smooth** the estimated parameter

Example: Bernoulli model

- Data:
  - We observed *N* ***iid*** coin tossing: *D*={1, 0, 1, …, 0}
- Representation:

Binary r.v: $x_n = \{0,1\}$

- Model: $P(x) = \begin{cases} 1-\theta & \text{for } x = 0 \\ \theta & \text{for } x = 1 \end{cases} \Rightarrow P(x) = \theta^{x}(1-\theta)^{1-x}$

- How to write the likelihood of a single observation x_i?

$$P(x_i) = \theta^{x_i}(1-\theta)^{1-x_i}$$

- The likelihood of dataset *D*={*x*₁, …,*x*_N}:

$$P(x_1, x_2, \ldots, x_N \mid \theta) = \prod_{i=1}^{N} P(x_i \mid \theta) = \prod_{i=1}^{N} \left(\theta^{x_i}(1-\theta)^{1-x_i}\right) = \theta^{\sum_{i=1}^{N} x_i} (1-\theta)^{\sum_{i=1}^{N} 1-x_i} = \theta^{\#\text{head}} (1-\theta)^{\#\text{tails}}$$

MLE

- Objective function:

$$\ell(\theta;D)=\log P(D\mid\theta)=\log\theta^{n_h}(1-\theta)^{n_t}\;=n_h\log\theta+(N-n_h)\log(1-\theta)$$

- We need to maximize this w.r.t. θ
- Take derivatives wrt θ

$$\frac{\partial\ell}{\partial\theta}=\frac{n_h}{\theta}-\frac{N-n_h}{1-\theta}=0 \quad\Rightarrow\quad \widehat{\theta}_{MLE}=\frac{n_h}{N} \quad\text{or}\quad \widehat{\theta}_{MLE}=\frac{1}{N}\sum_i x_i$$

Frequency as sample mean

- Sufficient statistics
  - The counts, n_h, where $n_k=\sum_i x_i$, are **sufficient statistics** of data *D*

Being a pragmatic frequentist

- Maximum *a posteriori* (MAP) estimation:

$$\widehat{\theta}_{MAP}=\arg\max_\theta\, p(\theta\mid D)=\arg\max_\theta\, \ell(\theta;D)+\log p(\theta)$$

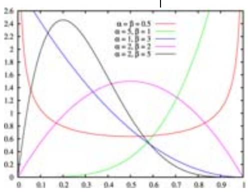
- Smoothing with pseudo-counts
  - Recall that for Binomial Distribution, we have

$$\widehat{\theta}^{head}_{MLE}=\frac{n^{head}}{n^{head}+n^{tail}}$$

  - What if we tossed too few times so that we saw zero head? We have $\widehat{\theta}^{head}_{ML}=0$, and we will predict that the probability of seeing a head next is zero!!!
  - The rescue: $\widehat{\theta}^{head}_{MLE}=\dfrac{n^{head}+n'}{n^{head}+n^{tail}+n'}$ **But are we still objective?**
    - Where n' is know as the pseudo- (imaginary) count

Bayesian estimation for Bernoulli

- Beta distribution:



P(θ)

1/B

$$\int P(\theta;\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1} = B(\alpha,\beta)\theta^{\alpha-1}(1-\theta)^{\beta-1}$$

  - = 1 When x is discrete $\Gamma(x+1) = x\Gamma(x) = x!$

- Posterior distribution of θ:

$$P(\theta \mid x_1,\ldots,x_N) = \frac{p(x_1,\ldots,x_N \mid \theta)\,p(\theta)}{p(x_1,\ldots,x_N)} \propto \theta^{n_h}(1-\theta)^{n_t} \times \theta^{\alpha-1}(1-\theta)^{\beta-1} = \theta^{n_h+\alpha-1}(1-\theta)^{n_t+\beta-1}$$

$$\frac{\Gamma(\alpha+\beta+N)}{\Gamma(\alpha+N^+)\Gamma(\beta+N^-)}$$

  - Notice the isomorphism of the posterior to the prior,
  - such a prior is called a **conjugate prior**
  - *α and β* are hyperparameters (parameters of the prior) and correspond to the number of "virtual" heads/tails (pseudo counts)

Bayesian estimation for Bernoulli, con'd

- Posterior distribution of θ:

$$P(\theta \mid x_1,\ldots,x_N) = \frac{p(x_1,\ldots,x_N \mid \theta)\,p(\theta)}{p(x_1,\ldots,x_N)} \propto \theta^{n_h}(1-\theta)^{n_t} \times \theta^{\alpha-1}(1-\theta)^{\beta-1} = \theta^{n_h+\alpha-1}(1-\theta)^{n_t+\beta-1}$$

- Maximum *a posteriori* (MAP) estimation:

$$\theta_{MAP} = \arg\max_{\theta} \log P(\theta \mid x_1,\ldots,x_N)$$

- Posterior mean estimation:

Bata parameters can be understood as pseudo-counts

$$\underline{\theta_{Bayes}} = \int \theta p(\theta \mid D)d\theta = C\int \theta \times \theta^{n_h+\alpha-1}(1-\theta)^{n_t+\beta-1}d\theta = \frac{n_h+\alpha}{N+\alpha+\beta}$$

- Prior strength: A=α+β
  - A can be interoperated as the size of an imaginary data set from which we obtain the **pseudo-counts**

Effect of Prior Strength

- Suppose we have a uniform prior (α=β=1/2), and we observe $\vec{n} = (n_h = 2, n_t = 8)$
- Weak prior A = 2. Posterior prediction:

$$p(x = h \mid n_h = 2, n_t = 8, \vec{\alpha} = \vec{\alpha}' \times 2) = \frac{1+2}{2+10} = 0.25$$

- Strong prior A = 20. Posterior prediction:

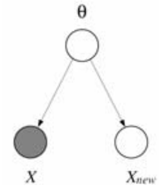
$$p(x = h \mid n_h = 2, n_t = 8, \vec{\alpha} = \vec{\alpha}' \times 20) = \frac{10+2}{20+10} = 0.40$$

- However, if we have enough data, it washes away the prior. e.g., $\vec{n} = (n_h = 200, n_t = 800)$. Then the estimates under weak and strong prior are $\frac{1+200}{2+1000}$ and $\frac{10+200}{20+1000}$, respectively, both of which are close to 0.2

How estimators should be used?

- $\hat{\theta}_{MAP}$ is not Bayesian (even though it uses a prior) since it is a point estimate.
- Consider predicting the future. A sensible way is to combine predictions based on all possible values of θ, weighted by their posterior probability, this is what a Bayesian will do:

$$\begin{aligned} p(x_{new} \mid \mathbf{x}) &= \int p(x_{new}, \theta \mid \mathbf{x}) d\theta \\ &= \int p(x_{new} \mid \theta, \mathbf{x}) p(\theta \mid \mathbf{x}) d\theta \\ &= \int p(x_{new} \mid \theta) p(\theta \mid \mathbf{x}) d\theta \end{aligned}$$



- A frequentist will typically use a “plug-in” estimator such as ML/MAP:

$$p(x_{new} \mid \mathbf{x}) = p(x_{new} \mid \hat{\theta}_{ML}), \quad \text{or,} \;\; p(x_{new} \mid \mathbf{x}) = p(x_{new} \mid \hat{\theta}_{MAP})$$

  - The Bayesian estimate will collapse to MAP for concentrated posterior

Frequentist vs. Beyesian

- This is a “theological” war.
- Advantages of Bayesian approach:
  - Mathematically elegant.
  - Works well when amount of data is much less than number of parameters (e.g., one-shot learning).
  - Easy to do incremental (sequential) learning.
  - Can be used for model selection (max likelihood will always pick the most complex model).
- Advantages of frequentist approach:
  - Mathematically/ computationally simpler.
  - "objective", unbiased, invariant to reparameterization
- As $|\mathcal{D}| \rightarrow \infty$, the two approaches become the same:

$$p(\theta | D) \rightarrow \delta(\theta, \widehat{\theta}_{ML})$$

Simplest GMs: the building blocks

Density estimation

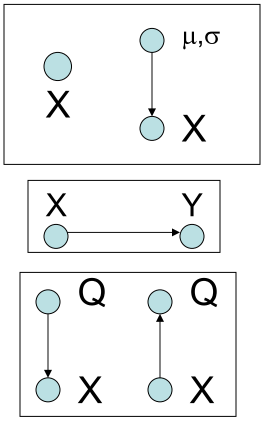
Parametric and nonparametric methods

Regression

Linear, conditional mixture, nonparametric

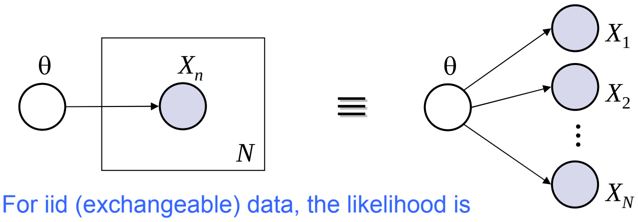
Classification

Generative and discriminative approach



Plates

- A plate is a “macro” that allows subgraphs to be replicated



  - For iid (exchangeable) data, the likelihood is

$$p(D \mid \theta) = \prod_n p(x_n \mid \theta)$$

  - We can represent this as a Bayes net with *N* nodes.
    - The rules of plates are simple: repeat every structure in a box a number of times given by the integer in the corner of the box (e.g. N), updating the plate index variable (e.g. n) as you go.
    - Duplicate every arrow going into the plate and every arrow leaving the plate by connecting the arrows to each copy of the structure.

Discrete Distributions

- Bernoulli distribution: Ber(*p*)

$$P(x) = \begin{cases} 1-p & \text{for } x = 0 \\ p & \text{for } x = 1 \end{cases} \quad \Rightarrow \quad P(x) = p^x (1-p)^{1-x}$$

- Multinomial distribution: Mult(1,*θ*)
  - Multinomial (indicator) variable:

$$X = \begin{bmatrix} X^1 \\ X^2 \\ X^3 \\ X^4 \\ X^5 \\ X^6 \end{bmatrix}, \qquad \text{where} \qquad \begin{array}{l} X^j = [0,1], \quad \text{and} \quad \sum_{j \in [1,\dots,6]} X^j = 1 \\ X^j = 1 \text{ w.p. } \theta_j, \quad \sum_{j \in [1,\dots,6]} \theta_j = 1 \ . \end{array}$$

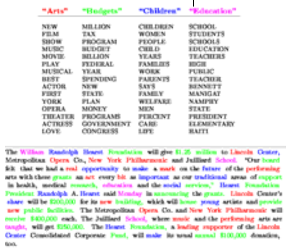
$$p(x(j)) = P\left(\{X_j = 1, \text{where } j \text{ index the dice-face}\}\right)$$
$$= \theta_j = \theta_1^{x^1} \times \theta_2^{x^2} \times \theta_3^{x^3} \times \theta_4^{x^4} \times \theta_5^{x^5} \times \theta_6^{x^6} = \prod_k \theta_k^{x^k} = \theta^x$$

Discrete Distributions

- Multinomial distribution: Mult(n,θ)
  - Count variable:

$$n = \begin{bmatrix} n_1 \\ \vdots \\ n_K \end{bmatrix}, \qquad \text{where } \sum_j n_j = N$$

$$p(n) = \frac{N!}{n_1!n_2!\cdots n_K!}\theta_1^{n_1}\theta_2^{n_2}\cdots\theta_K^{n_K} = \frac{N!}{n_1!n_2!\cdots n_K!}\theta^n$$



"Arts"	"Budgets"	"Children"	"Education"
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

The William Randolph Hearst Foundation will give $1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. "Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services," Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center's share will be $200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive $400,000 each. The Juilliard School, where music and the performing arts are taught, will get $250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual $100,000 donation, too.

Example: multinomial model

- Data:
  - We observed N ***iid*** die rolls (K-sided): D={5, 1, K, …, 3}
- Representation:

Unit basis vectors: $$x_n = \begin{pmatrix} x_n^1 \\ x_n^2 \\ \vdots \\ x_n^K \end{pmatrix}, \quad \text{where } x_n^k = \{0,1\}, \text{ and } \sum_{k=1}^{K} x_n^k = 1$$

- Model: $$X_n^k = 1 \text{ w.p. } \theta_k, \text{ and } \sum_{k\in\{1,\ldots K\}} \theta_k = 1$$

- How to write the likelihood of a single observation x_n?

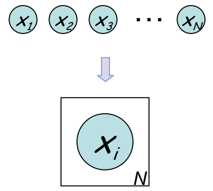
$$P(x_i) = P(\{x_n^k = 1, \text{where } k \text{ index the die-side of the } n\text{th roll}\})$$

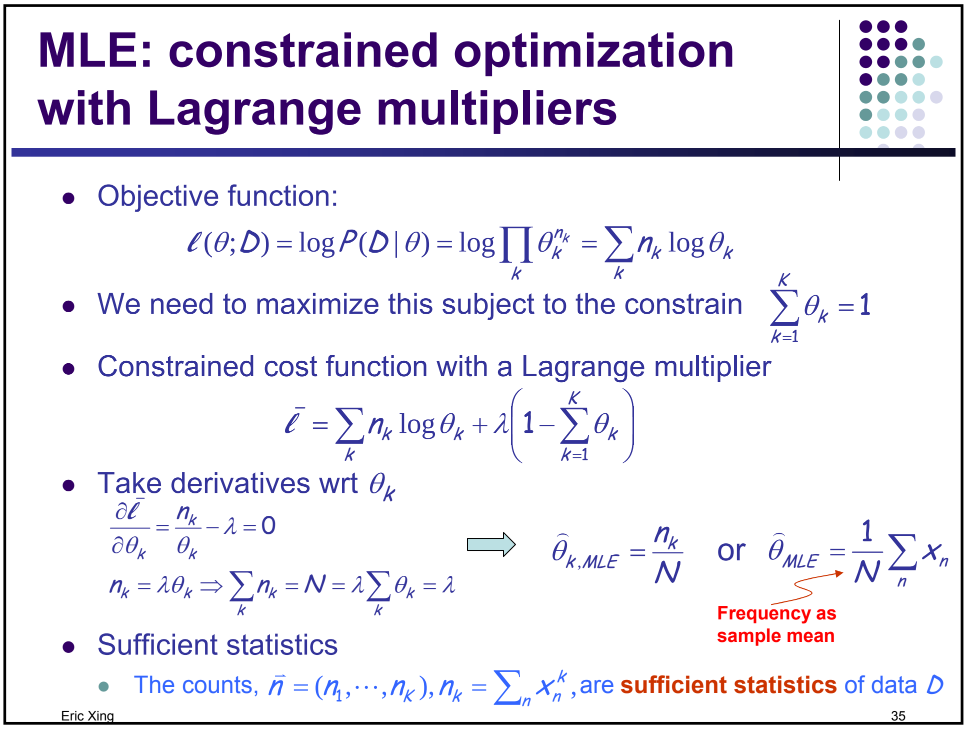
$$= \theta_k = \theta_1^{x_n^1} \times \theta_2^{x_n^2} \times \cdots \times \theta_K^{x_n^K} = \prod_{k=1}^{K} \theta_k^{x_n^k}$$

- The likelihood of dataset $D=\{x_1, \ldots, x_N\}$:

$$P(x_1, x_2, \ldots, x_N \mid \theta) = \prod_{n=1}^{N} P(x_n \mid \theta) = \prod_{n=1}^{N}\left(\prod_k \theta_k^{x_n^k}\right) = \prod_k \theta_k^{\sum_{n=1}^{N} x_n^k} = \prod_k \theta_k^{n_k}$$

GM:





MLE: constrained optimization with Lagrange multipliers

- Objective function:

$$\ell(\theta;D)=\log P(D\mid\theta)=\log\prod_k \theta_k^{n_k}=\sum_k n_k\log\theta_k$$

- We need to maximize this subject to the constrain $\sum_{k=1}^{K} \theta_k=1$
- Constrained cost function with a Lagrange multiplier

$$\bar{\ell}=\sum_k n_k\log\theta_k+\lambda\left(1-\sum_{k=1}^{K}\theta_k\right)$$

- Take derivatives wrt θ_k

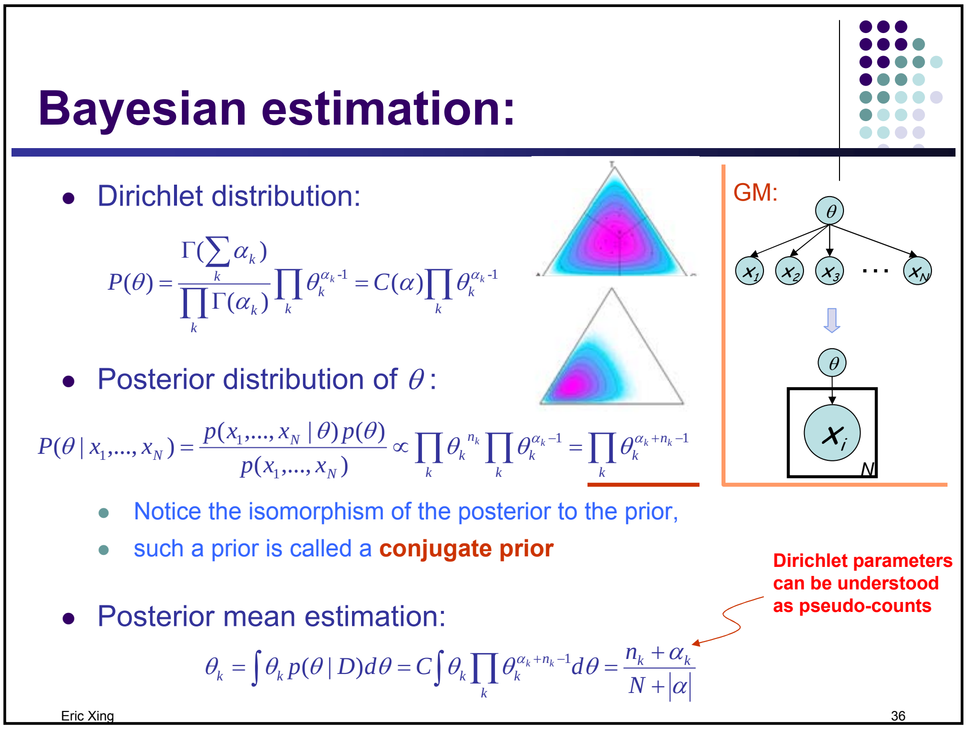
$$\frac{\partial\bar{\ell}}{\partial\theta_k}=\frac{n_k}{\theta_k}-\lambda=0$$

$$n_k=\lambda\theta_k\Rightarrow\sum_k n_k=N=\lambda\sum_k\theta_k=\lambda$$

⇨ $\hat{\theta}_{k,MLE}=\frac{n_k}{N}$ or $\hat{\theta}_{MLE}=\frac{1}{N}\sum_n x_n$

Frequency as sample mean

- Sufficient statistics
  - The counts, $\vec{n}=(n_1,\cdots,n_K), n_k=\sum_n x_n^k$, are **sufficient statistics** of data D



Bayesian estimation:

- Dirichlet distribution:

$$P(\theta)=\frac{\Gamma(\sum_k\alpha_k)}{\prod_k\Gamma(\alpha_k)}\prod_k\theta_k^{\alpha_k-1}=C(\alpha)\prod_k\theta_k^{\alpha_k-1}$$

- Posterior distribution of θ:

$$P(\theta\mid x_1,\ldots,x_N)=\frac{p(x_1,\ldots,x_N\mid\theta)p(\theta)}{p(x_1,\ldots,x_N)}\propto\prod_k\theta_k^{n_k}\prod_k\theta_k^{\alpha_k-1}=\underline{\prod_k\theta_k^{\alpha_k+n_k-1}}$$

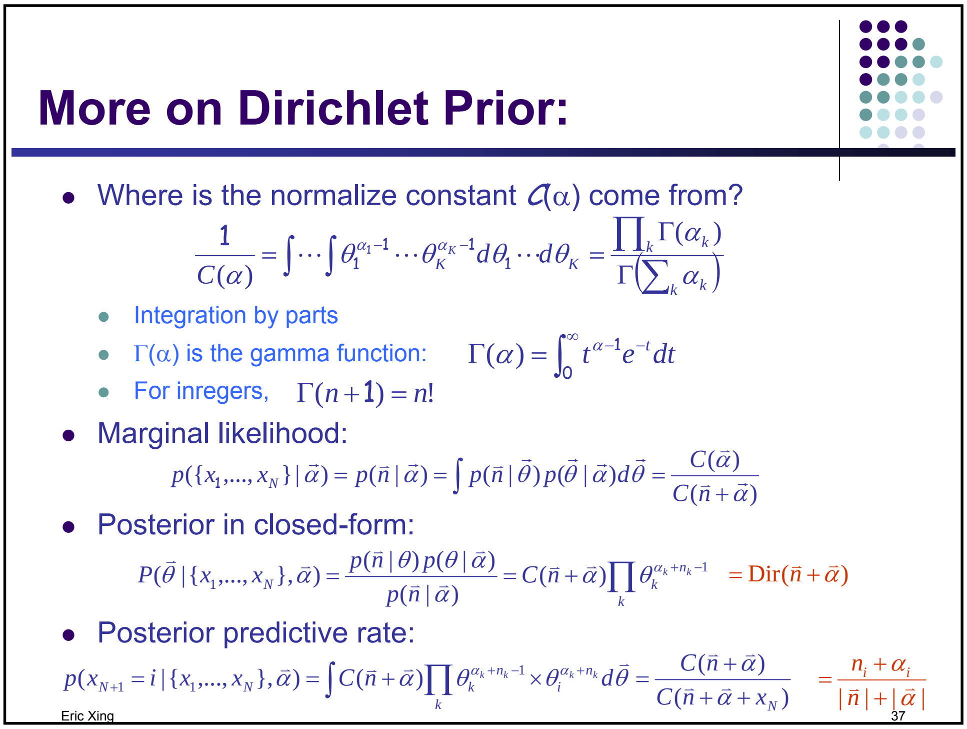
  - Notice the isomorphism of the posterior to the prior,
  - such a prior is called a **conjugate prior**

- Posterior mean estimation:

$$\theta_k=\int\theta_k p(\theta\mid D)d\theta=C\int\theta_k\prod_k\theta_k^{\alpha_k+n_k-1}d\theta=\frac{n_k+\alpha_k}{N+|\alpha|}$$

Dirichlet parameters can be understood as pseudo-counts

GM:



More on Dirichlet Prior:

- Where is the normalize constant $C(\alpha)$ come from?

$$\frac{1}{C(\alpha)} = \int \cdots \int \theta_1^{\alpha_1 - 1} \cdots \theta_K^{\alpha_K - 1} d\theta_1 \cdots d\theta_K = \frac{\prod_k \Gamma(\alpha_k)}{\Gamma\left(\sum_k \alpha_k\right)}$$

  - Integration by parts
  - $\Gamma(\alpha)$ is the gamma function: $\Gamma(\alpha) = \int_0^\infty t^{\alpha-1} e^{-t} dt$
  - For inregers, $\Gamma(n+1) = n!$
- Marginal likelihood:

$$p(\{x_1,\dots,x_N\} \mid \vec{\alpha}) = p(\vec{n} \mid \vec{\alpha}) = \int p(\vec{n} \mid \vec{\theta}) p(\vec{\theta} \mid \vec{\alpha}) d\vec{\theta} = \frac{C(\vec{\alpha})}{C(\vec{n} + \vec{\alpha})}$$

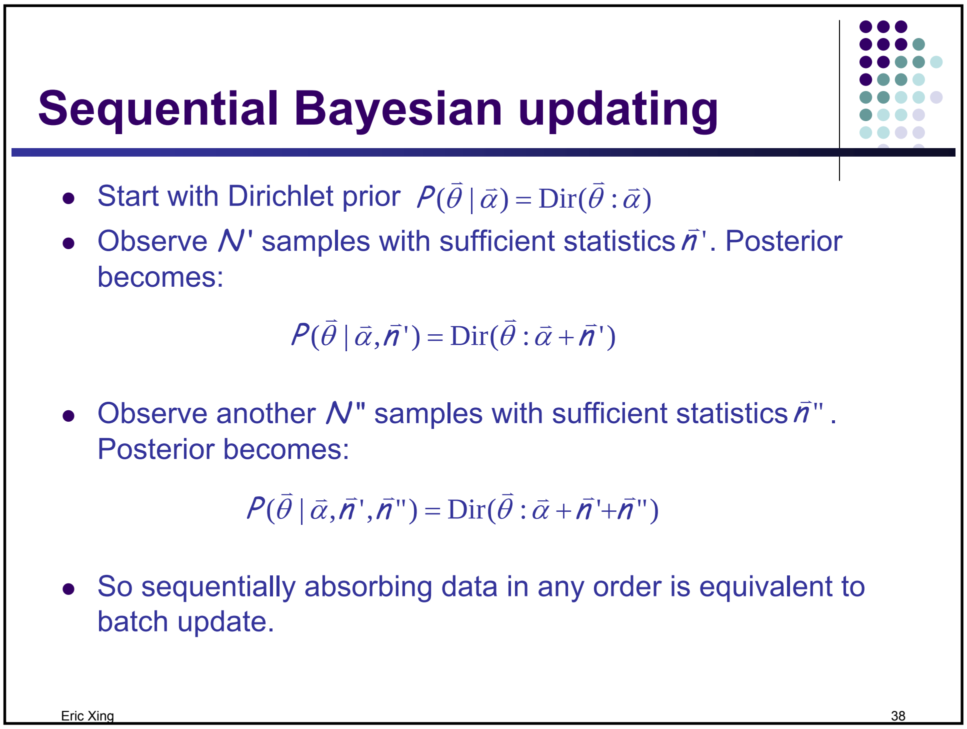
- Posterior in closed-form:

$$P(\vec{\theta} \mid \{x_1,\dots,x_N\}, \vec{\alpha}) = \frac{p(\vec{n} \mid \theta) p(\theta \mid \vec{\alpha})}{p(\vec{n} \mid \vec{\alpha})} = C(\vec{n} + \vec{\alpha}) \prod_k \theta_k^{\alpha_k + n_k - 1} \quad = \mathrm{Dir}(\vec{n} + \vec{\alpha})$$

- Posterior predictive rate:

$$p(x_{N+1} = i \mid \{x_1,\dots,x_N\}, \vec{\alpha}) = \int C(\vec{n} + \vec{\alpha}) \prod_k \theta_k^{\alpha_k + n_k - 1} \times \theta_i^{\alpha_k + n_k} d\vec{\theta} = \frac{C(\vec{n} + \vec{\alpha})}{C(\vec{n} + \vec{\alpha} + x_N)} \quad = \frac{n_i + \alpha_i}{|\vec{n}| + |\vec{\alpha}|}$$

Eric Xing 37



Sequential Bayesian updating

- Start with Dirichlet prior $P(\vec{\theta} \mid \vec{\alpha}) = \mathrm{Dir}(\vec{\theta} : \vec{\alpha})$
- Observe N' samples with sufficient statistics $\vec{n}'$. Posterior becomes:

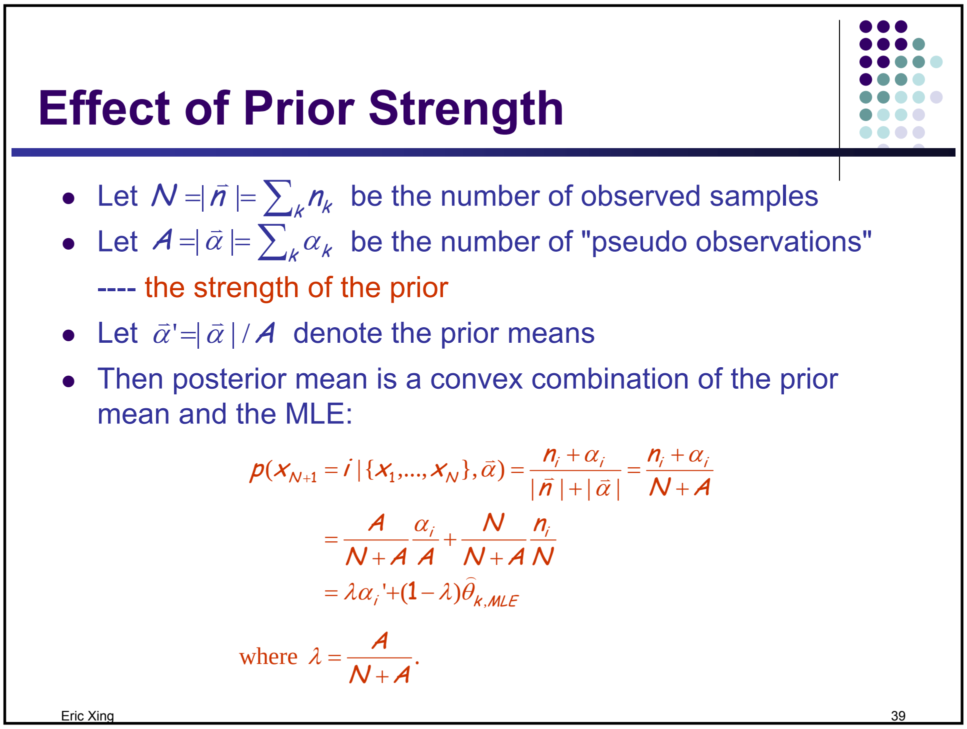
$$P(\vec{\theta} \mid \vec{\alpha}, \vec{n}') = \mathrm{Dir}(\vec{\theta} : \vec{\alpha} + \vec{n}')$$

- Observe another N'' samples with sufficient statistics $\vec{n}''$. Posterior becomes:

$$P(\vec{\theta} \mid \vec{\alpha}, \vec{n}', \vec{n}'') = \mathrm{Dir}(\vec{\theta} : \vec{\alpha} + \vec{n}' + \vec{n}'')$$

- So sequentially absorbing data in any order is equivalent to batch update.

Eric Xing 38



Effect of Prior Strength

- Let $N = |\vec{n}| = \sum_k n_k$ be the number of observed samples
- Let $A = |\vec{\alpha}| = \sum_k \alpha_k$ be the number of "pseudo observations"

---- the strength of the prior

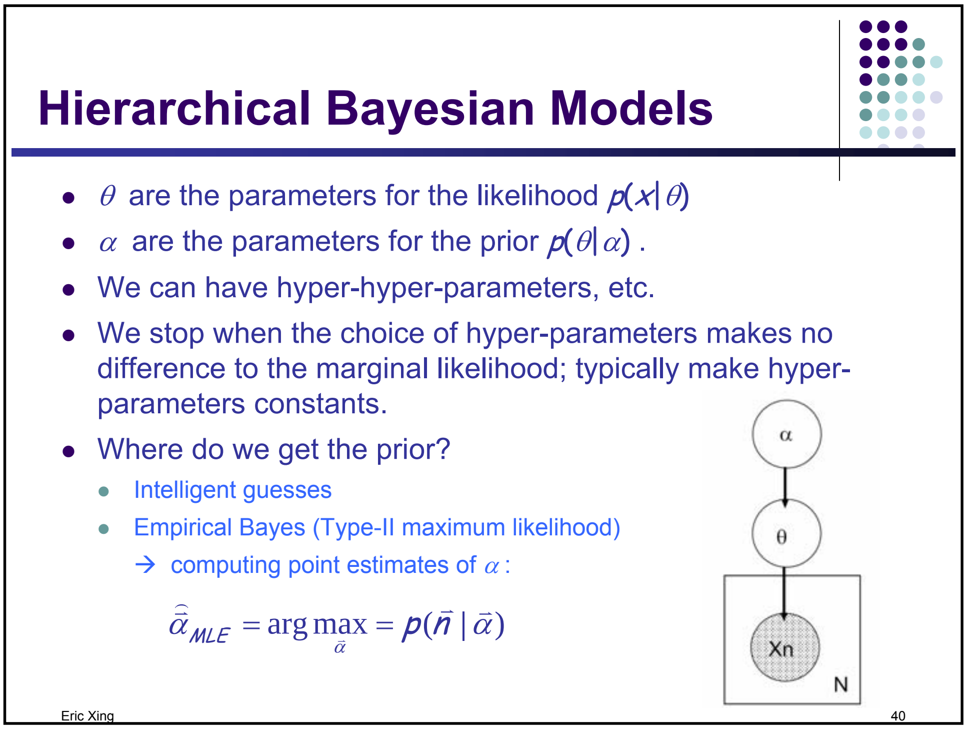
- Let $\vec{\alpha}' = |\vec{\alpha}| / A$ denote the prior means
- Then posterior mean is a convex combination of the prior mean and the MLE:

$$p(x_{N+1} = i \mid \{x_1, \dots, x_N\}, \vec{\alpha}) = \frac{n_i + \alpha_i}{|\vec{n}| + |\vec{\alpha}|} = \frac{n_i + \alpha_i}{N + A}$$

$$= \frac{A}{N+A}\frac{\alpha_i}{A} + \frac{N}{N+A}\frac{n_i}{N}$$

$$= \lambda \alpha_i' + (1-\lambda)\hat{\theta}_{k,MLE}$$

where $\lambda = \frac{A}{N+A}$.



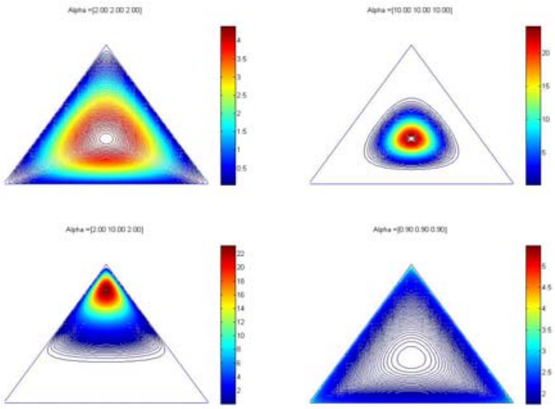
Hierarchical Bayesian Models

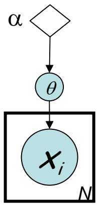
- θ are the parameters for the likelihood $p(x|\theta)$
- α are the parameters for the prior $p(\theta|\alpha)$.
- We can have hyper-hyper-parameters, etc.
- We stop when the choice of hyper-parameters makes no difference to the marginal likelihood; typically make hyper-parameters constants.
- Where do we get the prior?
  - Intelligent guesses
  - Empirical Bayes (Type-II maximum likelihood)

    → computing point estimates of α :

$$\hat{\vec{\alpha}}_{MLE} = \arg\max_{\vec{\alpha}} = p(\vec{n} \mid \vec{\alpha})$$

Limitation of Dirichlet Prior:





The Logistic Normal Prior

$$\theta \sim LN_K(\mu, \Sigma)$$

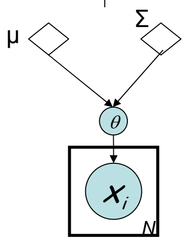
$$\gamma \sim N_{K-1}(\mu, \Sigma) \qquad \gamma_K = 0$$

$$\theta_i = \exp\left\{\gamma_i - \log\left(1 + \sum_{i=1}^{K-1} e^{\gamma_i}\right)\right\}$$

$$C(\gamma) = \log\left(1 + \sum_{i=1}^{K-1} e^{\gamma_i}\right)$$

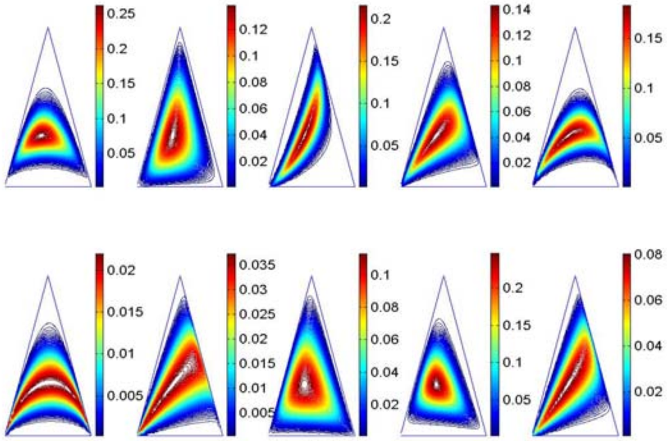
Problem

- Log Partition Function
- Normalization Constant



- Pro: co-variance structure
- Con: non-conjugate (we will discuss how to solve this later)

Logistic Normal Densities



Logistic Normal

Example 2: univariate-Gaussian

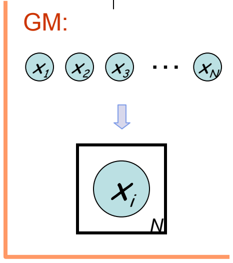
- Data:
  - We observed *N* ***iid*** real samples:
    D={-0.1, 10, 1, -5.2, …, 3}
- Model: $P(x) = \left(2\pi\sigma^2\right)^{-1/2} \exp\left\{-(x-\mu)^2 / 2\sigma^2\right\}$
- Log likelihood:

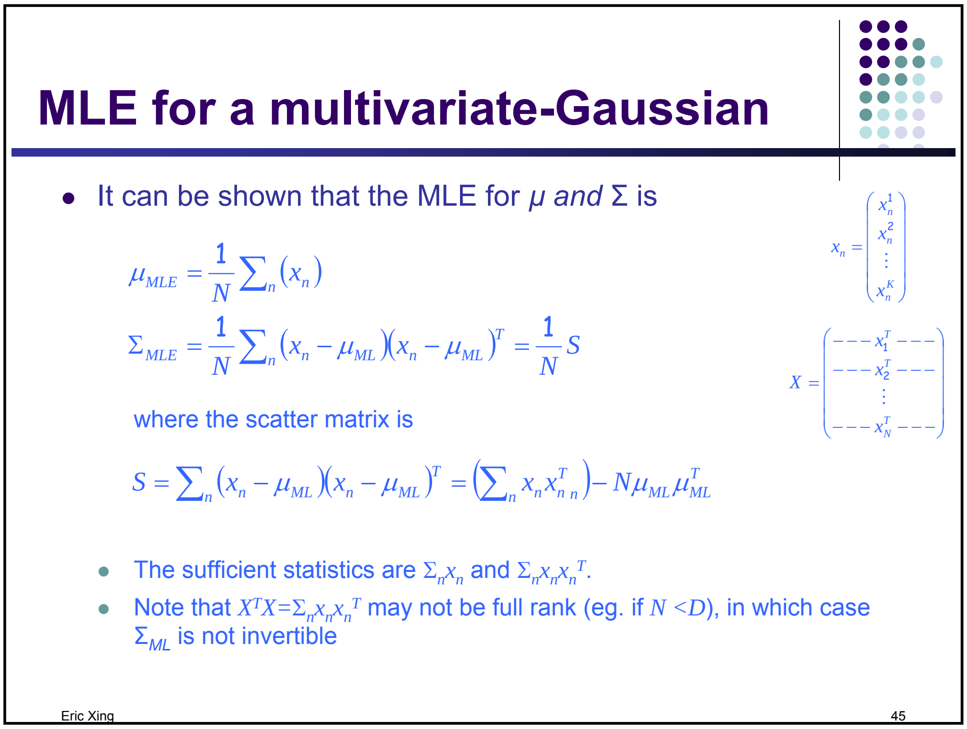
$$\ell(\theta; D) = \log P(D \mid \theta) = -\frac{N}{2}\log(2\pi\sigma^2) - \frac{1}{2}\sum_{n=1}^{N}\frac{(x_n-\mu)^2}{\sigma^2}$$

- MLE: take derivative and set to zero:

$$\frac{\partial \ell}{\partial \mu} = (1/\sigma^2)\sum_n (x_n - \mu)$$

$$\frac{\partial \ell}{\partial \sigma^2} = -\frac{N}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_n (x_n-\mu)^2 \quad \Rightarrow \quad \mu_{MLE} = \frac{1}{N}\sum_n (x_n) \qquad \sigma^2_{MLE} = \frac{1}{N}\sum_n (x_n - \mu_{ML})^2$$





MLE for a multivariate-Gaussian

- It can be shown that the MLE for *μ and* Σ is

$$x_n = \begin{pmatrix} x_n^1 \\ x_n^2 \\ \vdots \\ x_n^K \end{pmatrix}$$

$$\mu_{MLE} = \frac{1}{N}\sum\nolimits_n (x_n)$$

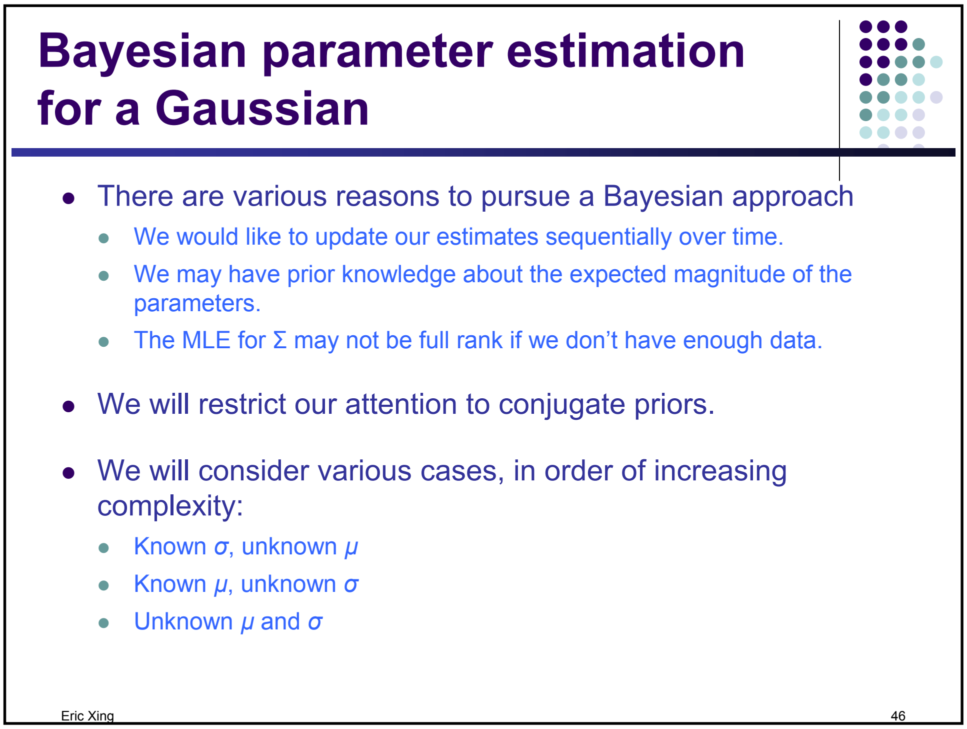
$$\Sigma_{MLE} = \frac{1}{N}\sum\nolimits_n (x_n - \mu_{ML})(x_n - \mu_{ML})^T = \frac{1}{N}S$$

$$X = \begin{pmatrix} ---x_1^T--- \\ ---x_2^T--- \\ \vdots \\ ---x_N^T--- \end{pmatrix}$$

where the scatter matrix is

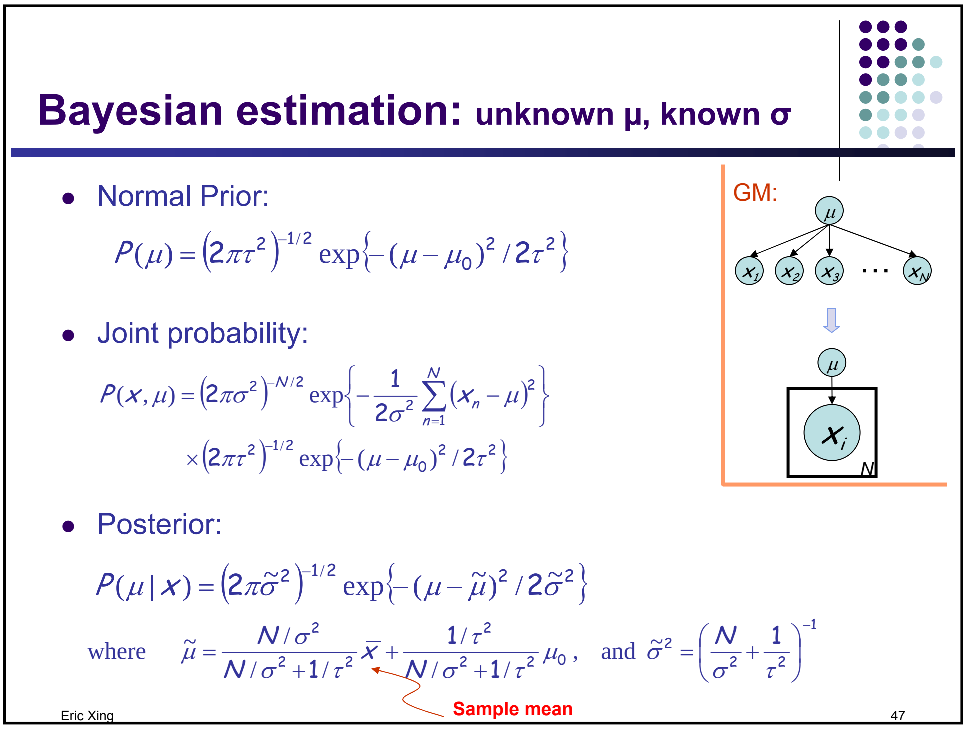
$$S = \sum\nolimits_n (x_n - \mu_{ML})(x_n - \mu_{ML})^T = \left(\sum\nolimits_n x_n x^T_{n\ n}\right) - N\mu_{ML}\mu_{ML}^T$$

  - The sufficient statistics are $\Sigma_n x_n$ and $\Sigma_n x_n x_n^T$.
  - Note that $X^T X = \Sigma_n x_n x_n^T$ may not be full rank (eg. if $N < D$), in which case Σ_{ML} is not invertible



Bayesian parameter estimation for a Gaussian

- There are various reasons to pursue a Bayesian approach
  - We would like to update our estimates sequentially over time.
  - We may have prior knowledge about the expected magnitude of the parameters.
  - The MLE for Σ may not be full rank if we don't have enough data.
- We will restrict our attention to conjugate priors.
- We will consider various cases, in order of increasing complexity:
  - Known *σ*, unknown *μ*
  - Known *μ*, unknown *σ*
  - Unknown *μ* and *σ*



Bayesian estimation: unknown μ, known σ

- Normal Prior:

$$P(\mu) = \left(2\pi\tau^2\right)^{-1/2} \exp\left\{-(\mu-\mu_0)^2 / 2\tau^2\right\}$$

- Joint probability:

$$P(x,\mu) = \left(2\pi\sigma^2\right)^{-N/2} \exp\left\{-\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2\right\}$$
$$\times\left(2\pi\tau^2\right)^{-1/2} \exp\left\{-(\mu-\mu_0)^2 / 2\tau^2\right\}$$

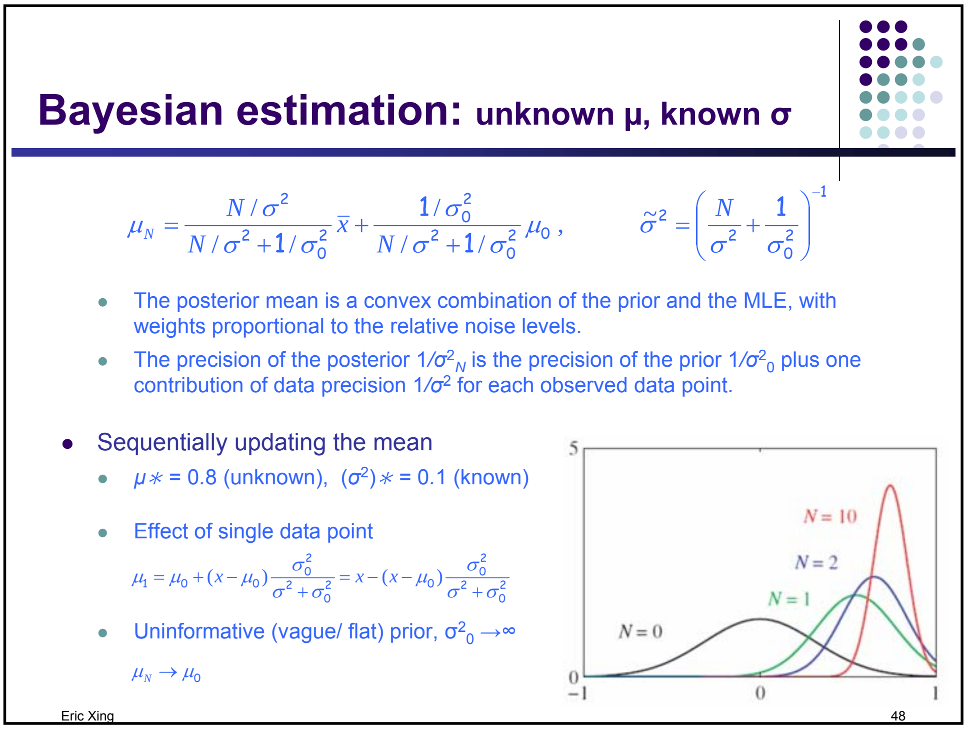
- Posterior:

$$P(\mu \mid x) = \left(2\pi\tilde{\sigma}^2\right)^{-1/2} \exp\left\{-(\mu-\tilde{\mu})^2 / 2\tilde{\sigma}^2\right\}$$

where $\tilde{\mu} = \frac{N/\sigma^2}{N/\sigma^2 + 1/\tau^2}\bar{x} + \frac{1/\tau^2}{N/\sigma^2 + 1/\tau^2}\mu_0$, and $\tilde{\sigma}^2 = \left(\frac{N}{\sigma^2} + \frac{1}{\tau^2}\right)^{-1}$

Sample mean

GM:



Bayesian estimation: unknown μ, known σ

$$\mu_N = \frac{N/\sigma^2}{N/\sigma^2 + 1/\sigma_0^2}\bar{x} + \frac{1/\sigma_0^2}{N/\sigma^2 + 1/\sigma_0^2}\mu_0, \qquad \tilde{\sigma}^2 = \left(\frac{N}{\sigma^2} + \frac{1}{\sigma_0^2}\right)^{-1}$$

- The posterior mean is a convex combination of the prior and the MLE, with weights proportional to the relative noise levels.
- The precision of the posterior $1/\sigma^2_N$ is the precision of the prior $1/\sigma^2_0$ plus one contribution of data precision $1/\sigma^2$ for each observed data point.

- Sequentially updating the mean
  - $\mu* = 0.8$ (unknown), $(\sigma^2)* = 0.1$ (known)
  - Effect of single data point

    $$\mu_1 = \mu_0 + (x-\mu_0)\frac{\sigma_0^2}{\sigma^2+\sigma_0^2} = x - (x-\mu_0)\frac{\sigma_0^2}{\sigma^2+\sigma_0^2}$$

  - Uninformative (vague/ flat) prior, $\sigma^2_0 \to \infty$

    $$\mu_N \to \mu_0$$

Other scenarios

- Known *μ*, unknown *λ* = 1/*σ*$_2$
  - The conjugate prior for *λ* is a Gamma with shape *a*$_0$ and rate (inverse scale) *b*$_0$

$$p(\lambda|a,b) = \frac{1}{\Gamma(a)} b^a \lambda^{a-1} \exp(-b\lambda)$$

  - The conjugate prior for *σ*^{2} is Inverse-Gamma

$$IG(\sigma^2|a,b) = \frac{1}{\Gamma(a)} b^a (\sigma^2)^{-(a+1)} \exp(-b/(\sigma^2))$$

- Unknown *μ* and unknown *σ*$_2$
  - The conjugate prior is Normal-Inverse-Gamma

$$\begin{aligned} P(\mu,\sigma^2) &= P(\mu|\sigma^2)P(\sigma^2) \\ &= \mathcal{N}(\mu|m,\sigma^2 V)\ \ IG(\sigma^2|a,b) \end{aligned}$$

  - Semi conjugate prior

- Multivariate case:
  - The conjugate prior is Normal-Inverse-Wishart

$$\begin{aligned} P(\mu,\Sigma) &= P(\mu|\Sigma)P(\Sigma) \\ &= \mathcal{N}(\mu|\mu_0,\frac{1}{\kappa_0}\Sigma)\ \ \mathcal{IW}(\Sigma|\Lambda_0^{-1},\nu_0) \end{aligned}$$

Summary

- Learning scenarios:
  - Data
  - Objective function
  - Frequetist and Bayesian

- Learning single-node GM – density estimation
  - Typical discrete distribution
  - Typical continuous distribution
  - Conjugate priors