Machine Learning in Image Analysis
Day 2

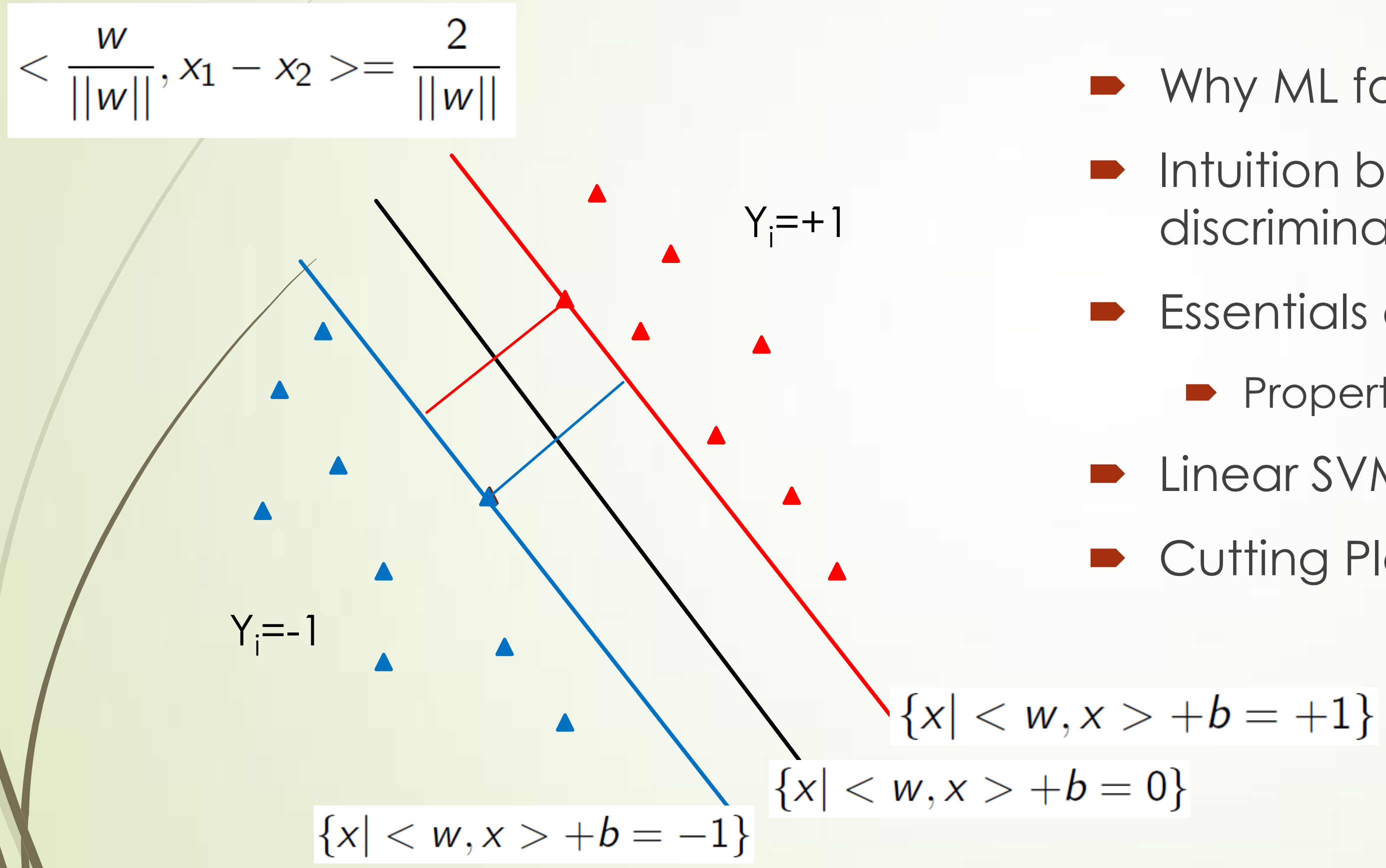
Anirban Mukhopadhyay

Zuse Institute Berlin

Organization

- Recap
- Basic Mathematical Structures of ML, MAP and Bayesian
  - Basics
  - ML vs MAP vs Bayesian
  - Simple model fitting example using ML
- Expectation Maximization algorithm
  - Basics
  - EM derivation
- Importance Sampling and MC Integration
  - Bayesian Practicalities

Recap Day 1



- Why ML for IA?
- Intuition behind choosing either discriminative or generative
- Essentials of Convex sets and functions
  - Properties of 1st order Taylor Approximation
- Linear SVM Formulation
- Cutting Plane algo to solve linear SVM

Basic Mathematical Structures of ML, MAP and Bayesian

- Fitting probability models to data
- Generative Machine Learning
- This is called learning because we learn about parameters (Training)
- Also concerns calculating the probability of a new data point
  - Evaluating a predictive distribution (Testing)

Basic Bayesian

$$\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{I}$$

where each $\mathbf{x}_i$ is a realization of a random variable $\mathbf{x}$. **Each observation $\mathbf{x}_i$ is, in general, a data point in a multidimensional space.**

Basic Bayesian

$$\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{I}$$

where each $\mathbf{x}_i$ is a realization of a random variable $\mathbf{x}$. **Each observation $\mathbf{x}_i$ is, in general, a data point in a multidimensional space.**

We may wish to estimate the parameters Θ with the help of the Bayes' Rule

$$prob(\Theta|\mathcal{X}) = \frac{prob(\mathcal{X}|\Theta) \cdot prob(\Theta)}{prob(\mathcal{X})}$$

Basic Bayesian

$$\mathcal{X} \quad = \quad \{\mathbf{x}_i\}_{i=1}^{I}$$

where each $\mathbf{x}_i$ is a realization of a random variable $\mathbf{x}$. **Each observation $\mathbf{x}_i$ is, in general, a data point in a multidimensional space.**

We may wish to estimate the parameters Θ with the help of the Bayes' Rule

$$prob(\Theta|\mathcal{X}) \quad = \quad \frac{prob(\mathcal{X}|\Theta) \cdot prob(\Theta)}{prob(\mathcal{X})}$$

$$posterior \quad = \quad \frac{likelihood \cdot prior}{evidence}$$

ML vs MAP vs Bayesian

We seek that value for Θ which maximizes the likelihood shown on the previous slide. That is, we seek that value for Θ which gives largest value to

$$prob(\mathcal{X}|\Theta)$$

We denote such a value of Θ by $\widehat{\Theta}_{ML}$.

ML vs MAP vs Bayesian

$$
\begin{aligned}
\widehat{\Theta}_{MAP} &= \underset{\Theta}{\operatorname{argmax}}\ prob(\Theta|\mathcal{X}) \\
&= \underset{\Theta}{\operatorname{argmax}}\ \frac{prob(\mathcal{X}|\Theta)\ \cdot\ prob(\Theta)}{prob(\mathcal{X})} \\
&= \underset{\Theta}{\operatorname{argmax}}\ prob(\mathcal{X}|\Theta)\ \cdot\ \boxed{prob(\Theta)} \\
&= \underset{\Theta}{\operatorname{argmax}}\ \prod_{\mathbf{x}_i \in \mathcal{X}} prob(\mathbf{x}_i|\Theta)\ \cdot\ prob(\Theta)
\end{aligned}
$$

ML vs MAP vs Bayesian

$$\underset{\Theta}{\operatorname{argmax}} \frac{prob(\mathcal{X}|\Theta) \cdot prob(\Theta)}{prob(\mathcal{X})}$$

$$prob(\mathcal{X}) = \int_{\Theta} prob(\mathcal{X}|\Theta) \cdot prob(\Theta) \; d\Theta$$

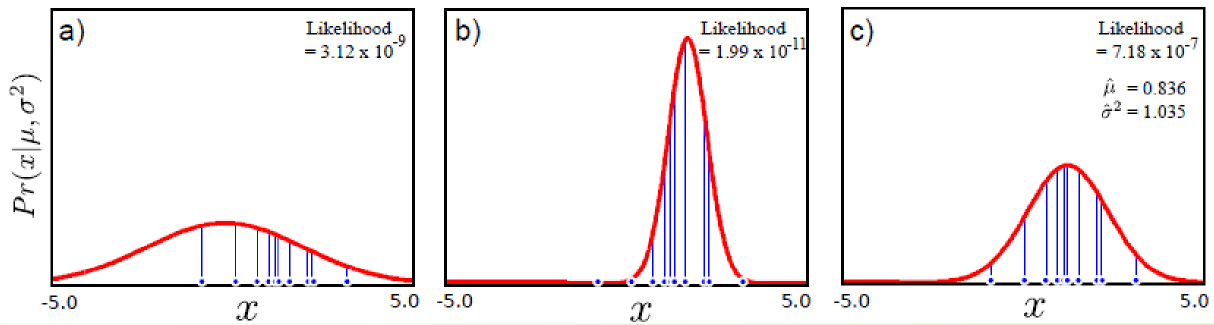
Example of calculating ML

- Fitting a univariate normal with pdf: $Pr(x|\mu,\sigma^2) = \text{Norm}_x[\mu,\sigma^2] = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[-0.5\frac{(x-\mu)^2}{\sigma^2}\right]$
- Quiz time: Parameters?
- **Simplest Strategy:**
  - Evaluate pdf for each data point separately
  - Take the product

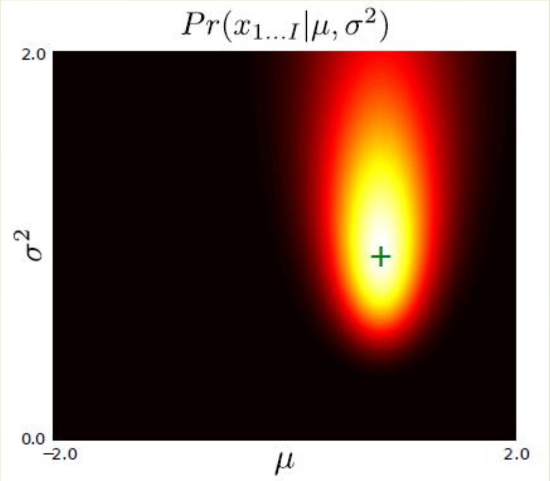
Example of calculating ML

- Fitting a univariate normal with pdf: $Pr(x|\mu,\sigma^2) = \text{Norm}_x[\mu,\sigma^2] = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[-0.5\frac{(x-\mu)^2}{\sigma^2}\right]$
- Quiz time: Parameters?
- **Simplest Strategy:**
  - Evaluate pdf for each data point separately
  - Take the product

$$
\begin{aligned}
Pr(x_{1\ldots I}|\mu,\sigma^2) &= \prod_{i=1}^{I} Pr(x_i|\mu,\sigma^2)\\
&= \prod_{i=1}^{I} \text{Norm}_{x_i}[\mu,\sigma^2]\\
&= \frac{1}{(2\pi\sigma^2)^{I/2}}\exp\left[-0.5\sum_{i=1}^{I}\frac{(x_i-\mu)^2}{\sigma^2}\right]
\end{aligned}
$$



Log-likelihood



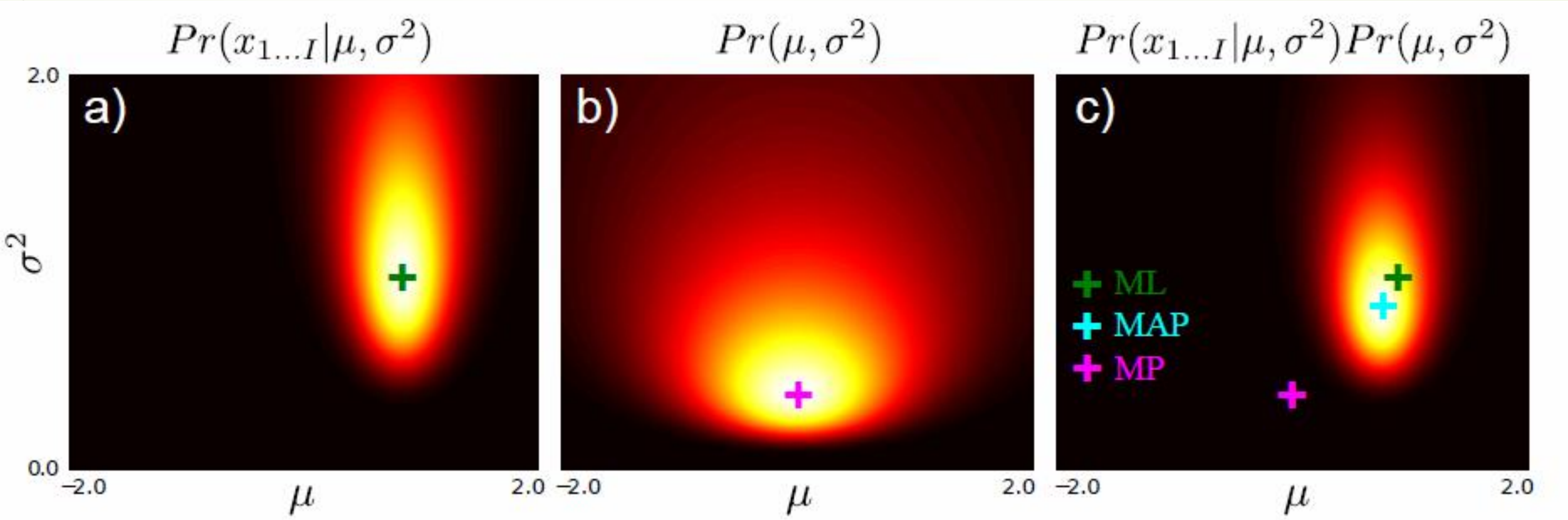
- Maximum likelihood solution occurs at peak
- How to find peak? By taking derivative and equating to 0
- Resulting eqns are messy
  - Take logarithm of the expression (monotonically increasing, so position of max in transformed space remains same)
  - Logarithm also decouples contribution by changing product to sum

$$
\begin{aligned}
\hat{\mu}, \hat{\sigma}^2 &= \underset{\mu, \sigma^2}{\operatorname{argmax}} \left[\sum_{i=1}^{I} \log \left[\text{Norm}_{x_i}[\mu, \sigma^2] \right] \right] \\
&= \underset{\mu, \sigma^2}{\operatorname{argmax}} \left[-0.5 I \log[2\pi] - 0.5 I \log \sigma^2 - 0.5 \sum_{i=1}^{I} \frac{(x_i - \mu)^2}{\sigma^2} \right]
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial L}{\partial \mu} &= \sum_{i=1}^{I} \frac{(x_i - \mu)}{\sigma^2} \\
&= \frac{\sum_{i=1}^{I} x_i}{\sigma^2} - \frac{I\mu}{\sigma^2} = 0
\end{aligned}
$$

Differentiating log likelihood L w.r.t. mean, similar for var

Comparing ML with MAP



Likelihood Prior Posterior

Log MaP derivations + its relation to Empirical Risk Minimization

$$\widehat{\Theta}_{MAP} = \underset{\Theta}{\text{argmax}} \left(\sum_{\mathbf{x}_i \in \mathcal{X}} log\ prob(\mathbf{x}_i|\Theta) + log\ prob(\Theta) \right)$$

$$\text{minimize} \left(-\sum_{\mathbf{x}_i \in \mathcal{X}} log\ prob(\mathbf{x}_i|\Theta) - log\ prob(\Theta) \right)$$

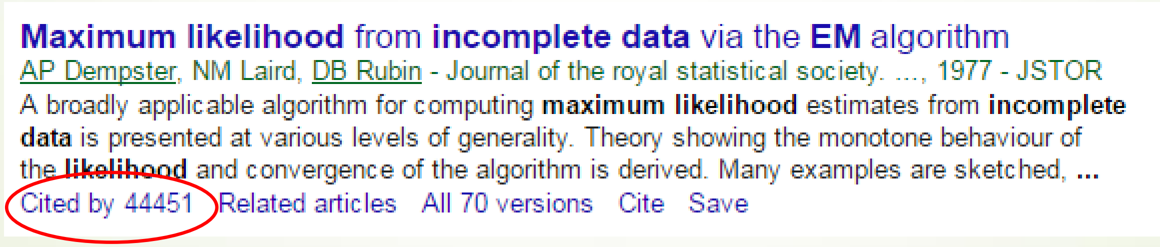
$$\underset{w}{\text{minimize}} \quad \lambda\omega(w) + \frac{1}{m}\sum_{i=1}^{m} l(x_i, y_i, w)$$

Regularizer

Risk

Expectation Maximization algorithm

- Quick facts:
  - Computes **Maximum Likelihood** estimate in the presence of missing data
  - Efficient iterative procedure for **maximizing log-likelihood**



Why EM?

- Despite the fact that EM can occasionally get stuck in a local maximum, 3 super cool stuffs about EM

- ability to simultaneously optimize a large number of variables
- the ability to find good estimates for any missing information in data at the same time
- GMM: the ability to create both the traditional "hard" clusters and not-so-traditional "soft" clusters.
  - "Hard": disjoint partition of Data
  - "Soft": allowing a data point to belong to two or more clusters at the same time, the "level of membership"

Main Idea of EM (Iterative Procedure)

- E-Step
  - Estimate missing data given observed data and current estimate
- M-Step
  - Maximize likelihood function under the assumption that missing data is known

Derivation of EM

- Maximizing L ≡ update s.t. $L(\theta) > L(\theta_n)$ ≡ maximize $L(\theta) - L(\theta_n) = \ln \mathcal{P}(\mathbf{X}|\theta) - \ln \mathcal{P}(\mathbf{X}|\theta_n)$
- Hidden / latent variable (Z) can be introduced here
  - As unobserved / missing variable
  - Artifact to make the solution tractable

$$\mathcal{P}(\mathbf{X}|\theta) = \sum_{\mathbf{z}} \mathcal{P}(\mathbf{X}|\mathbf{z}, \theta)\mathcal{P}(\mathbf{z}|\theta)$$

$$L(\theta) - L(\theta_n) = \ln \left(\sum_{\mathbf{z}} \mathcal{P}(\mathbf{X}|\mathbf{z}, \theta)\mathcal{P}(\mathbf{z}|\theta) \right) - \ln \mathcal{P}(\mathbf{X}|\theta_n).$$

Jensen's Inequality

$$\ln \sum_{i=1}^{n} \lambda_i x_i \geq \sum_{i=1}^{n} \lambda_i \ln(x_i)$$ if $\lambda_i \geq 0$ with $\sum_{i=1}^{n} \lambda_i = 1$

Contd.

$$
\begin{aligned}
L(\theta) - L(\theta_n) &= \ln\left(\sum_{\mathbf{z}} \mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta)\right) - \ln \mathcal{P}(\mathbf{X}|\theta_n) \\
&= \ln\left(\sum_{\mathbf{z}} \mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta) \cdot \frac{\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)}{\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)}\right) - \ln \mathcal{P}(\mathbf{X}|\theta_n) \\
&= \ln\left(\sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)\frac{\mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta)}{\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)}\right) - \ln \mathcal{P}(\mathbf{X}|\theta_n) \\
&\geq \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln\left(\frac{\mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta)}{\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)}\right) - \ln \mathcal{P}(\mathbf{X}|\theta_n) \\
&= \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln\left(\frac{\mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta)}{\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)\mathcal{P}(\mathbf{X}|\theta_n)}\right) \\
&\triangleq \Delta(\theta|\theta_n).
\end{aligned}
$$

λ_i

$\sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) = 1$ so that $\ln \mathcal{P}(\mathbf{X}|\theta_n) = \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln \mathcal{P}(\mathbf{X}|\theta_n)$

Inside ln, subtraction means division

Contd.

$$L(\theta) \geq L(\theta_n) + \Delta(\theta|\theta_n)$$

$$l(\theta|\theta_n) \triangleq L(\theta_n) + \Delta(\theta|\theta_n)$$ [To simplify notations]

$$L(\theta) \geq l(\theta|\theta_n)$$

$l(\theta|\theta_n)$ is bounded above by the likelihood function $L(\theta)$

value of the functions $l(\theta|\theta_n)$ and $L(\theta)$ are equal at $\theta = \theta_n$

And last bit of PAIN!! i.e. "more formally"

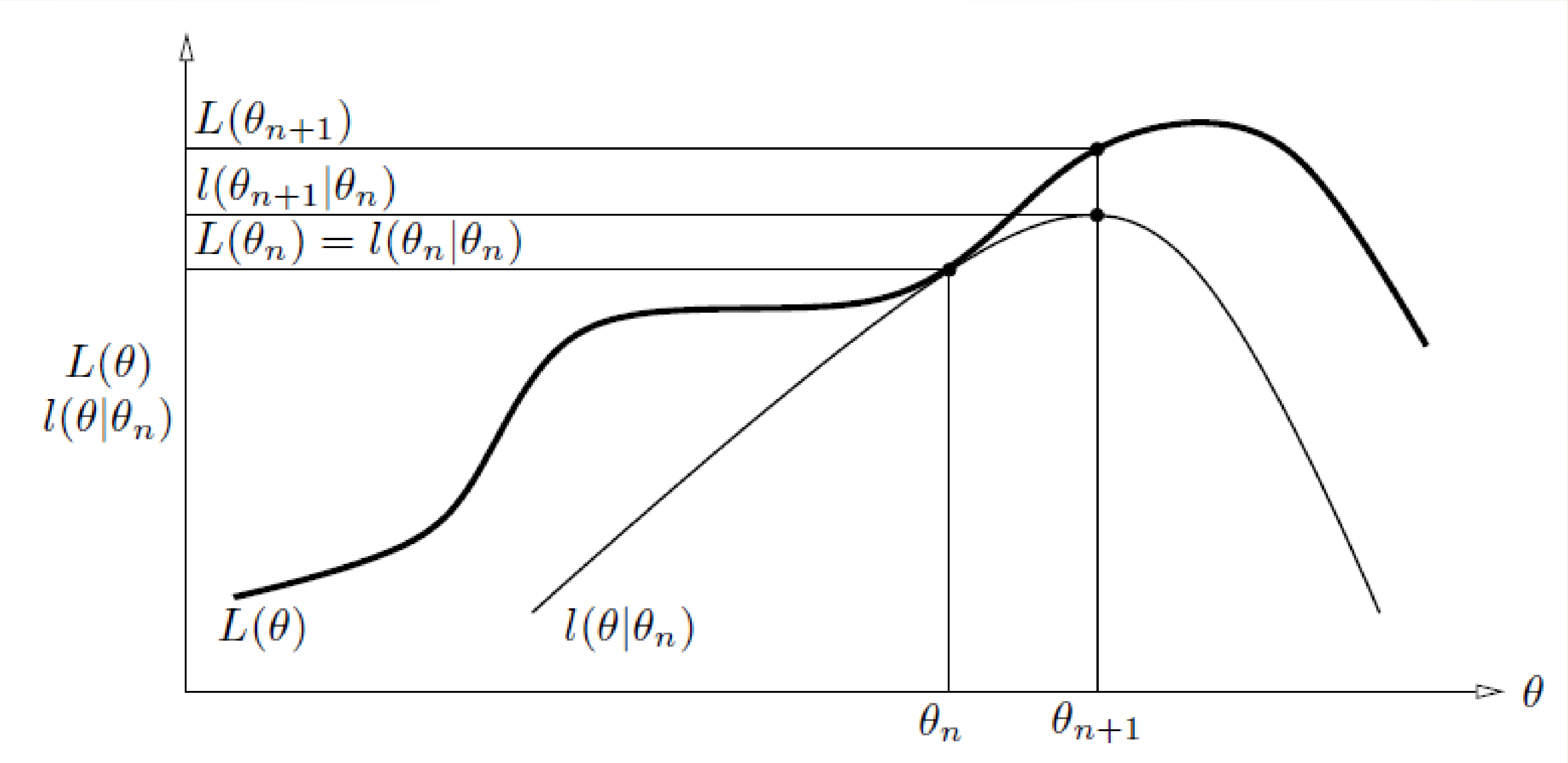
$$
\begin{aligned}
\theta_{n+1} &= \arg\max_{\theta} \{ l(\theta|\theta_n) \} \\
&= \arg\max_{\theta} \left\{ L(\theta_n) + \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln \frac{\mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta)}{\mathcal{P}(\mathbf{X}|\theta_n)\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)} \right\} \\
&\text{Now drop terms which are constant w.r.t. } \theta \\
&= \arg\max_{\theta} \left\{ \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln \mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta) \right\} \\
&= \arg\max_{\theta} \left\{ \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln \frac{\mathcal{P}(\mathbf{X},\mathbf{z},\theta)}{\cancel{\mathcal{P}(\mathbf{z},\theta)}} \frac{\cancel{\mathcal{P}(\mathbf{z},\theta)}}{\mathcal{P}(\theta)} \right\} \\
&= \arg\max_{\theta} \left\{ \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln \mathcal{P}(\mathbf{X},\mathbf{z}|\theta) \right\} \\
&= \arg\max_{\theta} \left\{ \mathbb{E}_{\mathbf{Z}|\mathbf{X},\theta_n} \{ \ln \mathcal{P}(\mathbf{X},\mathbf{z}|\theta) \} \right\}
\end{aligned}
$$

The latent/ missing variable Z is taken into account by maximizing this rather than log likelihood L

M-step: Maximize this exprsn w.r.t. θ

E-step: Determine this conditional Expectation

Graphically one iteration of EM



θ↑ $l(\theta|\theta_n)$↑ $L(\theta)$↑ ➡

At each iteration of EM

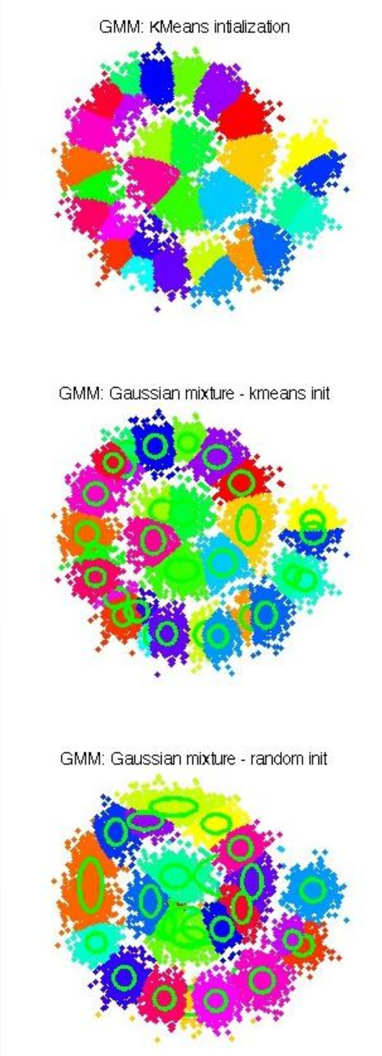
to achieve the greatest possible increase in the value of $L(\theta)$

EM algorithm calls for selecting θ such that $l(\theta|\theta_n)$ is maximized

GMM with K-means initialization vl-feat

- http://www.vlfeat.org/overview/gmm.html

```
numClusters = 30;
numData = 1000;
dimension = 2;
data = rand(dimension,numData);

% Run KMeans to pre-cluster the data
[initMeans, assignments] = vl_kmeans(data, numClusters, ...
    'Algorithm','Lloyd', ...
    'MaxNumIterations',5);

initCovariances = zeros(dimension,numClusters);
initPriors = zeros(1,numClusters);

% Find the initial means, covariances and priors
for i=1:numClusters
    data_k = data(:,assignments==i);
    initPriors(i) = size(data_k,2) / numClusters;

    if size(data_k,1) == 0 || size(data_k,2) == 0
        initCovariances(:,i) = diag(cov(data'));
    else
        initCovariances(:,i) = diag(cov(data_k'));
    end
end

% Run EM starting from the given parameters
[means,covariances,priors,ll,posteriors] = vl_gmm(data, numClusters, ...
    'initialization','custom', ...
    'InitMeans',initMeans, ...
    'InitCovariances',initCovariances, ...
    'InitPriors',initPriors);
```



Parameter Estimation and predictionof future values from evidence

$$\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{I}$$

where each $\mathbf{x}_i$ is a realization of a random variable $\mathbf{x}$. **Each observation $\mathbf{x}_i$ is, in general, a data point in a multidimensional space.**

Bayes' Rule (Reminder)

$$prob(\Theta|\mathcal{X}) = \frac{prob(\mathcal{X}|\Theta) \cdot prob(\Theta)}{prob(\mathcal{X})}$$

$$posterior = \frac{likelihood \cdot prior}{evidence}$$

Bayes w.r.t. ML and MAP

- ML considers the parameter vector to be a constant and seeks out that value for the constant that provides maximum support for the evidence.

Bayes w.r.t. ML and MAP

- ML considers the parameter vector to be a constant and seeks out that value for the constant that provides maximum support for the evidence.
- MAP allows the parameter vector to take values from a distribution that expresses our prior beliefs regarding the parameters. MAP returns that parameter value which maximizes the posterior.

Bayes w.r.t. ML and MAP

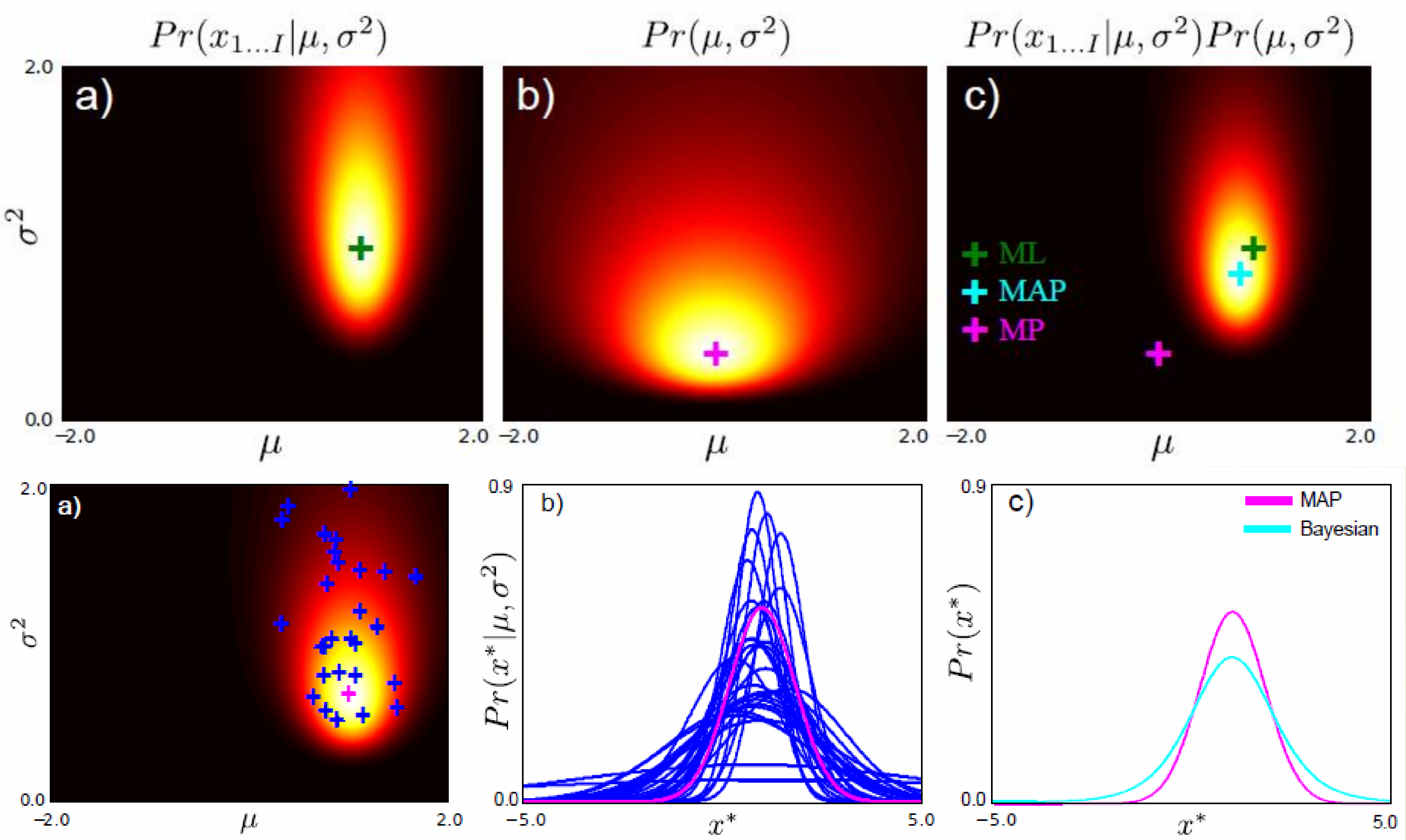
- ML considers the parameter vector to be a constant and seeks out that value for the constant that provides maximum support for the evidence.
- MAP allows the parameter vector to take values from a distribution that expresses our prior beliefs regarding the parameters. MAP returns that parameter value which maximizes the posterior.
- Both ML and MAP return only single and specific values

Bayes w.r.t. ML and MAP

- ML considers the parameter vector to be a constant and seeks out that value for the constant that provides maximum support for the evidence.
- MAP allows the parameter vector to take values from a distribution that expresses our prior beliefs regarding the parameters. MAP returns that parameter value which maximizes the posterior.
- Both ML and MAP return only single and specific values

- Bayesian estimation, by contrast, calculates fully the posterior distribution
  - Our job is to select the value that we consider “best” in certain sense

ML, MAP and Bayesian for Normal Parameter Estimation



Difficulties of Bayesian

- Theoretical
  - Integration at the denominator of the equation (probability of evidence)

$$prob(\mathcal{X}) \;=\; \int_{\Theta} prob(\mathcal{X}|\Theta) \cdot prob(\Theta)\; d\Theta$$

  - Conjugate prior: If we have a choice in how we express our prior beliefs, we must use that form which allows to carry out the integration

Difficulties of Bayesian

- Theoretical
  - Integration at the denominator of the equation (probability of evidence)

$$prob(\mathcal{X}) = \int_{\Theta} prob(\mathcal{X}|\Theta) \cdot prob(\Theta)\ d\Theta$$

  - Conjugate prior: If we have a choice in how we express our prior beliefs, we must use that form which allows to carry out the integration
- Practical
  - Integration in denominator is trivial as it is just a normalizer if you have reasonably high number of samples
  - Main problem: observation model you want to use

Importance Sampling and Monte Carlo Integration to the rescue

Solving Probabilistic Integrals Numerically

- Integrals that involve probability density functions in the integrands are ideal for solution by Monte Carlo methods.

$$E(g(\mathcal{X},\Theta)) \quad = \quad \int g(\mathcal{X},\Theta) \cdot prob(\Theta)\; d\Theta$$

- Monte Carlo approach to solving the integration is
  - draw samples from the probability distribution
  - estimate the integral with the help of these samples.

Problems

- When the distribution is simple, such as uniform or normal, it is trivial to draw such samples from the distribution and use the following as approximation

$$E(g(\mathcal{X}, \Theta)) \approx \frac{1}{n} \sum_{i=1}^{n} g(\mathcal{X}, \Theta^i)$$

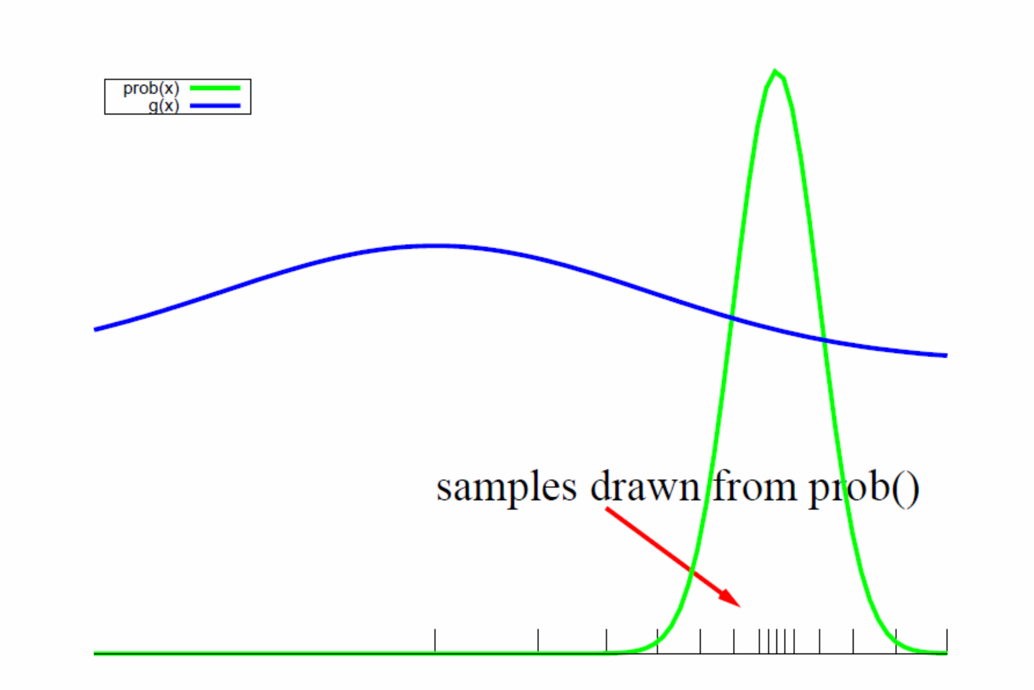
Problems

- When the distribution is simple, such as uniform or normal, it is trivial to draw such samples from the distribution and use the following as approximation

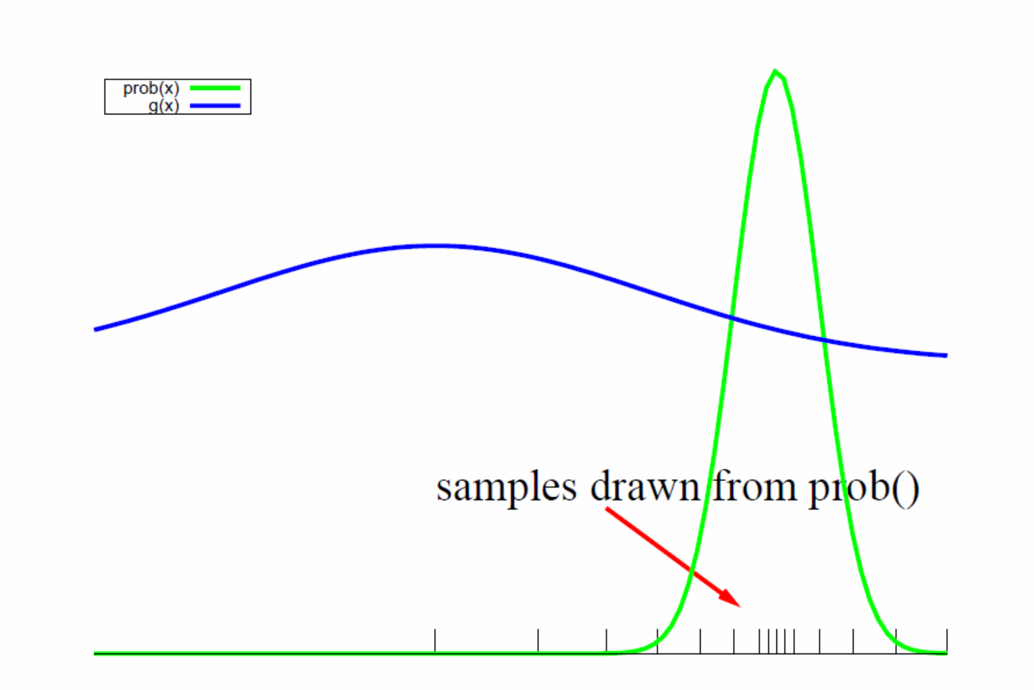
$$E(g(\mathcal{X}, \Theta)) \approx \frac{1}{n} \sum_{i=1}^{n} g(\mathcal{X}, \Theta^i)$$

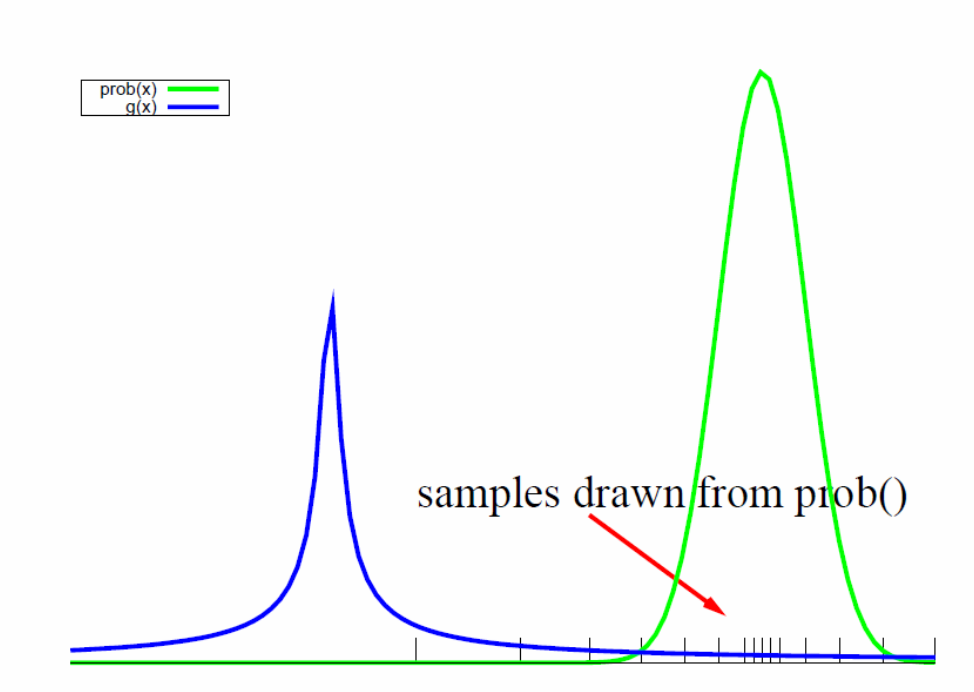
- However, in Bayesian estimation, probability distribution can be expected to be arbitrary
- Even if some samples are drawn, the approximation won't work any more

Deeper Explanation of the Problem



Deeper Explanation of the Problem





Importance Sampling

- Sampling not only based on priors, but also where function g() acquires significant values
  - Situations where we have no reason to believe that g() is compatible with ‘prior’

Importance Sampling

- Sampling not only based on priors, but also where function g() acquires significant values
  - Situations where we have no reason to believe that g() is compatible with 'prior'
- Importance sampling brings into play another distribution q(), known as the sampling distribution or the proposal distribution,
  - Help us do a better job of randomly sampling the values spanned by Θ

Integral remains unchanged

$$\frac{\int g(\mathcal{X}, \Theta)\ \frac{prob(\Theta)}{q(\Theta)}\ q(\Theta)\ \ d\Theta}{\int \frac{prob(\Theta)}{q(\Theta)}\ q(\Theta)\ d\Theta}$$

- As long as dividing by q() does not introduce any singularities

Practicalities of q()

- We can use "any" proposal distribution q() to draw random samples provided we now think:

$$s(\Theta) \quad = \quad g(\mathcal{X}, \Theta)\, \frac{prob(\Theta)}{q(\Theta)}$$

- We must now also estimate the integration in the denominator

$$\int t(\Theta) q(\Theta) d\Theta \qquad\qquad t(\Theta) \; = \; prob(\Theta)/q(\Theta)$$

- Implication: we must now first construct the weights ('importance weights') at the random samples drawn according to the probability distribution q()

$$w^i \quad = \quad \frac{prob(\Theta^i)}{q(\Theta^i)} \quad \Rightarrow \quad \frac{\frac{1}{n}\sum_{i=1}^{n} w^i \cdot g(\Theta^i)}{\frac{1}{n}\sum_{i=1}^{n} w^i}$$

Comparing different proposals for q()

- Monte-Carlo integration is an expectation of some entity g()

$$\int g(\Theta) \cdot prob(\Theta) \, d\Theta \;=\; E(g(\Theta)) \;\approx\; \sum_{i=1}^{n} W^i \cdot g(\Theta^i)$$

- associate a variance with this estimate, the Monte Carlo variance

$$\int [g(\Theta) - E(g(\Theta))]^2 \cdot prob(\Theta) \, d\Theta \;=\; Var(g(\Theta))$$

- Discrete approximation of the variance similar to MC Integration
- Goal: Choose the proposal distribution q() that minimizes the MC variance.

Still with the Problem of Having to Draw Samples According to a Prescribed Distribution

- For simplicity, p(x) denotes the distribution whose samples we wish to draw from for the purpose of Monte Carlo integration, f(x) arbitrary function
- Goal: Estimate the integral $\int_{\mathbf{x}\in\mathcal{X}} p(\mathbf{x}) f(\mathbf{x}) d\mathbf{x}$
- Trivial, if p(x) is simple
- Non trivial in complex cases
- Modern Approach: Markov-Chain Monte-Carlo

Markov-Chain Monte-Carlo (MCMC)

- For the very first sample x_1, any value that belongs to the domain of p(x), that is, any randomly chosen value x where $p(x) > 0$ is acceptable.

Markov-Chain Monte-Carlo (MCMC)

- For the very first sample x_1, any value that belongs to the domain of p(x), that is, any randomly chosen value x where $p(x) > 0$ is acceptable.
- Next sample, randomly choose a value from the interval where $p(x) > 0$ but must "reconcile" it with x_1. Let's denote the value we are now looking at as x^* and refer to it as our candidate for x_2.

Markov-Chain Monte-Carlo (MCMC)

- For the very first sample x_1, any value that belongs to the domain of p(x), that is, any randomly chosen value x where p(x) > 0 is acceptable.
- Next sample, randomly choose a value from the interval where p(x) > 0 but must "reconcile" it with x_1. Let's denote the value we are now looking at as x* and refer to it as our candidate for x_2.
- "reconcile": select a large number of samples in the vicinity of the peaks in p(x) and, relatively speaking, fewer samples where p(x) is close to 0. Capture this intuition by the ratio a1 = p(x*)/p(x_1).
  - If a1 > 1, then accepting x* as x_2

Markov-Chain Monte-Carlo (MCMC)

- For the very first sample x_1, any value that belongs to the domain of p(x), that is, any randomly chosen value x where p(x) > 0 is acceptable.
- Next sample, randomly choose a value from the interval where p(x) > 0 but must "reconcile" it with x_1. Let's denote the value we are now looking at as x* and refer to it as our candidate for x_2.
- "reconcile": select a large number of samples in the vicinity of the peaks in p(x) and, relatively speaking, fewer samples where p(x) is close to 0. Capture this intuition by the ratio a1 = $p(x^*)/p(x_1)$.
  - If a1 > 1, then accepting x* as x_2
- If a1 < 1, exercise some caution in accepting x* for x_2, as explained on the next slide.

MCMC contd.

- Want to accept x* as x_2 with some hesitation when a1 < 1
  - hesitation being greater the smaller the value of a1 in relation to unity
  - capture this intuition by saying that let's accept x* as x2 with probability a1.

Check out the board for Intuition

MCMC contd.

- Want to accept x* as x_2 with some hesitation when a1 < 1
  - hesitation being greater the smaller the value of a1 in relation to unity
  - capture this intuition by saying that let's accept x* as x2 with probability a1.

Check out the board for Intuition

Why Markov Chain?

Gibbs sampler – special case of MCMC

- Idea: The Gibbs sampler samples each dimension of X separately through the univariate conditional distribution along that dimension vis-a-vis the rest.

Gibbs sampler – special case of MCMC

- Idea: The Gibbs sampler samples each dimension of X separately through the univariate conditional distribution along that dimension vis-a-vis the rest.
- Individual components of $X = (x_1, \ldots, x_n)^T$
- Also, $X^{(-i)}=(x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n)^T$

Gibbs sampler – special case of MCMC

- Idea: The Gibbs sampler samples each dimension of X separately through the univariate conditional distribution along that dimension vis-a-vis the rest.
- Individual components of $X = (x_1, \ldots, x_n)^T$
- Also, $X^{(-i)}=(x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n)^T$
- Focus: Univariate conditional distribution: $p(x_i \mid X^{(-i)})$, for $i=1,\ldots,n$

Gibbs sampler – special case of MCMC

- Idea: The Gibbs sampler samples each dimension of X separately through the univariate conditional distribution along that dimension vis-a-vis the rest.
- Individual components of $X = (x_1, \ldots, x_n)^T$
- Also, $X^{(-i)}=(x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n)^T$
- Focus: Univariate conditional distribution: $p(x_i \mid X^{(-i)})$, for $i=1,\ldots,n$
- Keep in mind: Conditional distribution for x_i makes sense only when the other $n - 1$ variables in $X^{(-i)}$ are given constant values.

Gibbs sampler – special case of MCMC

- Idea: The Gibbs sampler samples each dimension of X separately through the univariate conditional distribution along that dimension vis-a-vis the rest.
- Individual components of $X = (x_1, \ldots, x_n)^T$
- Also, $X^{(-i)}=(x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n)^T$
- Focus: Univariate conditional distribution: $p(x_i \mid X^{(-i)})$, for $i=1,\ldots,n$
- Keep in mind: Conditional distribution for x_i makes sense only when the other $n - 1$ variables in $X^{(-i)}$ are given constant values.
- Main Observation: Even when the joint distribution $p(x)$ is multimodal, the univariate conditional distribution for each x_i, when all the other variables are held constant, is likely to be approximable by an unimodal distribution

Gibbs sampler – special case of MCMC

- Idea: The Gibbs sampler samples each dimension of X separately through the univariate conditional distribution along that dimension vis-a-vis the rest.
- Individual components of $X = (x_1, \ldots, x_n)^T$
- Also, $X^{(-i)}=(x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n)^T$
- Focus: Univariate conditional distribution: $p(x_i \mid X^{(-i)})$, for $i=1,\ldots,n$
- Keep in mind: Conditional distribution for x_i makes sense only when the other $n - 1$ variables in $X^{(-i)}$ are given constant values.
- Main Observation: Even when the joint distribution $p(x)$ is multimodal, the univariate conditional distribution for each x_i, when all the other variables are held constant, is likely to be approximable by an unimodal distribution
- Implication: Individual scalar variables can be approx. by std. rand gen SW

Gibbs Sampling

- Initialization: Choose random values for $x_2^{(0)},\dots,x_n^{(0)}$
- For k=1… K scans
  - Draw a sample for x_1 by: $x_1^{(k)} \sim p(x_1 \mid x^{(-1)}=(x_2^{(k-1)},\dots,x_n^{(k-1)}))$
  - Draw a sample for x_2 by: $x_2^{(k)} \sim p(x_2 \mid x_1=x_1^{(k)},\ \ x^{(-1,-2)}=(x_3^{(k-1)},\dots,x_n^{(k-1)}))$
  - Keep doing it for next j scalars: j = 3 … n
- End For

- In this manner, after K scans, we end up with K sampling points for vector variable X

References

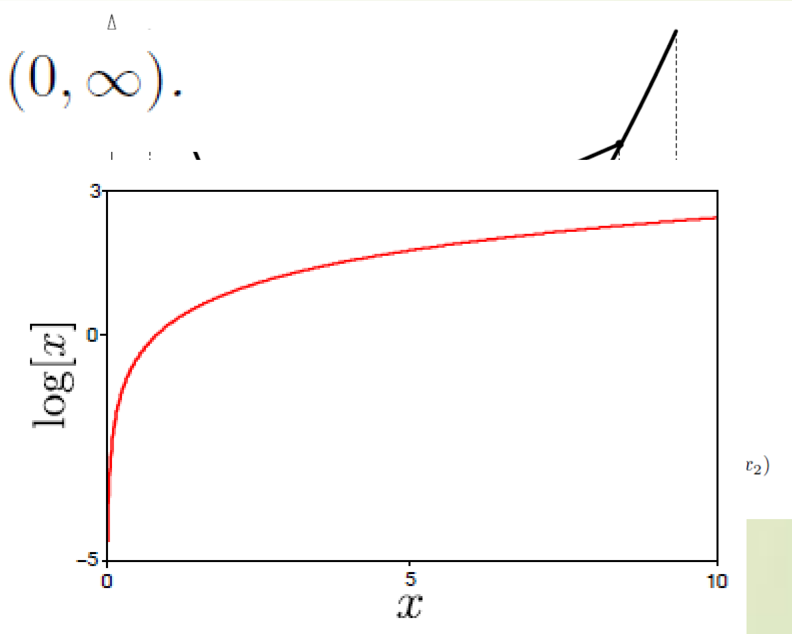
- Avinash Kak, ML, MAP and Bayesian – The Holy Trinity of Parameter Estimation and Data Prediction, Purdue University, 2014
- Avinash Kak, Monte Carlo Estimation in Bayesian Integration, Purdue University, 2014
- Sean Borman, The Expectation Maximization Algorithm A short tutorial, 2004
- A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society: Series B, 39(1):1–38, November 1977.
- Computer vision: models, learning and inference, Simon J.D. Prince, Cambridge University Press, 2012
- Optimization for Machine Learning, Sra, Nowozin, Wright, MIT Press, 2012

Mathematical developments that lead to the EM algorithm

Proposition 1 $-\ln(x)$ *is strictly convex on* $(0, \infty)$.

Theorem 2 (Jensen's inequality) *Let* f *be a convex* .
interval I. *If* $x_1, x_2, \ldots, x_n \in I$ *and* $\lambda_1, \lambda_2, \ldots, \lambda_n \geq 0$ *wi*

$$f\left(\sum_{i=1}^{n} \lambda_i x_i\right) \leq \sum_{i=1}^{n} \lambda_i f(x_i)$$



$$
\begin{aligned}
l(\theta_n|\theta_n) &= L(\theta_n) + \Delta(\theta_n|\theta_n) \\
&= L(\theta_n) + \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln \frac{\mathcal{P}(\mathbf{X}|\mathbf{z},\theta_n)\mathcal{P}(\mathbf{z}|\theta_n)}{\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)\mathcal{P}(\mathbf{X}|\theta_n)} \\
&= L(\theta_n) + \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln \frac{\mathcal{P}(\mathbf{X},\mathbf{z}|\theta_n)}{\mathcal{P}(\mathbf{X},\mathbf{z}|\theta_n)} \\
&= L(\theta_n) + \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln 1 \\
&= L(\theta_n),
\end{aligned}
$$

MCMC contd.

- Want to accept x^* as x_2 with some hesitation when $a1 < 1$
  - hesitation being greater the smaller the value of a1 in relation to unity
  - capture this intuition by saying that let's accept x* as x2 with probability a1.
- Algorithmically:
  - fire up a random-number generator that returns floating-point numbers in the interval (0, 1).
  - Let's say the number returned by the random-number generator is u.
  - accept x^* as x_2 if $u < a1$.
- Intuition towards original Metropolis Algorithm

Comparison contd.

- Goal: Choose the proposal distribution q() that minimizes the MC variance.
- proposal distribution that minimizes the Monte-Carlo variance is given by

$$q(\Theta) \quad \propto \quad |g(\Theta) \cdot prob(\Theta)|$$

- Not a complete solution to the choosing of the proposal distribution, the product g()prob() may not sample g() properly because the former goes to zero where it should not.