EM (Expectation Maximization) algorithm (& its diverse applications in bioinformatics)

Optional !Saturday Track!

CS5691 PRML Jul-Nov 2025

"Algorithmic Thinking in Bioinformatics" Course

Manikandan Narayanan

Associate Professor, Computer Science and Engineering, IIT Madras

Context: Seen so far…

- Session on Motivation and Background for “Algorithmic Thinking in Bioinformatics” course
- Sessions on several biological questions & their algorithmic solutions

- This session: Take a ***computation-first*** approach:
  - computational problem (“*learn the parameters of a mixture model aka latent variable model from data*”),
  - its algorithmic solution (*EM algorithm*), and
  - several biological applications (from motif-finding to soft k-means clustering, genes/isoforms’ expression quantitation, haplotype estimation, etc.).

Lecture style and sources

- Probabilistic (mixture) modeling and EM algo. is front-and-center:
  - Specifically, generative modeling + inference + parameter-learning using a mixture/latent-variable model.
  - Propose EM algorithm as a generic tool to learn the (MLE) parameters of a mixture model from data.
  - Apply this generic EM algo. to analyze various types of biological datasets with missing/incomplete data, and hidden structure/patterns.

- Sources for this lecture:
  - Main sources:
    - Do & Batzoglou. What is the expectation maximization algorithm?. *Nat Biotechnol* **26,** 897–899 (2008). - **cited as [DB08]**
    - Fan, Yuan & Liu. The EM Algorithm and the Rise of Computational Biology. Statistical Science. 25(4): 476-491 (2010). – **cited as [FYL10]**
  - Other sources:
    - David J. C. MacKay. Information Theory, Inference and Learning Algorithms (Chapters 20-22). 2003. – **cited as [DJM]** for content/figures taken as is from this book.
    - Stanford CS228 Probabilistic Graphical Models Course Lecture Notes. https://ermongroup.github.io/cs228-notes/ - cited as **[ECL]**

Outline of EM Algo. & Appns.

- **Expectation-Maximization (EM) Algorithm (Algo.)**
  - **Background: Mixture Model (MM) or Latent Variable Model (LVM)**
  - Generic algorithm for parameter-learning of a MM
  - Algorithm analysis (correctness)

- Bioinformatics Applications (Appns.) of EM Algo.
  - Application 0: Bernoulli mixture of two coins (toy/warmup appn.)
  - Application 1: EM meets motif-finding in DNA sequences
  - Application 2: Soft k-means clustering of gene expression data (revisit appn.)
  - Application 3: Genes/isoforms' expression quantitation (deconvolution)
  - Application 4: Haplotype estimation from genotype data
  - .
  - .
  - .

Recall: Probabilistic Mixture Model (MM) or Latent Variable Model (LVM)

- What? A complex yet elegant model built from simpler (tractable) components.
  - Any model with joint distbn. $P(X, Z) = \mathrm{P(Z)P(X|Z)}$, where X is observed and Z is latent.
  - Marginal $P(X) = \sum_{\mathrm{z}} \mathrm{P(Z)P(X|Z)}$

- Why? i.e., why include variables that are always unobserved/hidden/latent in a model?
  - To make model more expressive, yet interpretive; and
  - To capture prior knowledge about hidden patterns/structure in our data (e.g., on clusters in a dataset)!

[ECL]

Recall: MM or LVM Tasks

- Generate *iid* data from model parameters (generative model): $\theta \rightarrow \boldsymbol{x} \coloneqq \left\{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\right\}$

$$\{\pi_k\}_{k=1}^{K} \rightarrow z^{(n)} \ (= k')$$

$$\mathrm{Model}_{\mathrm{k}'}(\theta_{k'}) \rightarrow x^{(n)}$$

- Inference:

$$r_k^{(n)} \coloneqq P(Z = k \mid X = x^{(n)}; \theta) = \frac{\pi_k \, P\left(X = x^{(n)} \,\middle|\, Z = k; \theta\right)}{\sum_{k'=1}^{K} \pi_{k'} P(X = x^{(n)} \mid Z = k'; \theta)}$$

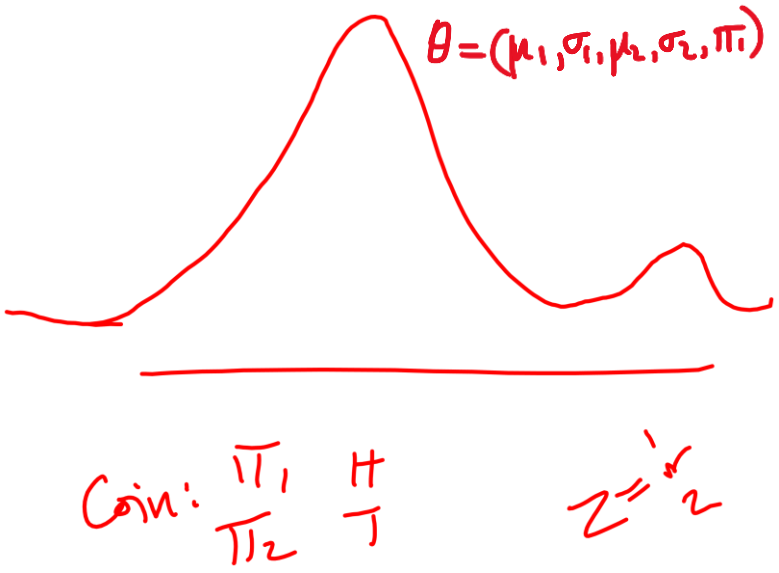
$P\left(X = x^{(n)}; \theta\right)$ = Dr. of above expn.

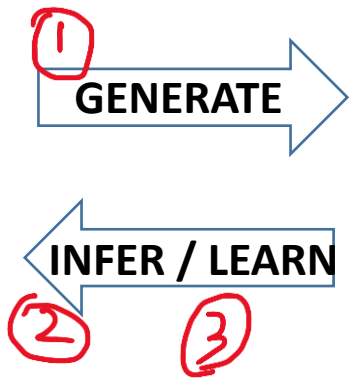
- Learning model parameters from data (MLE approach; EM algorithm)

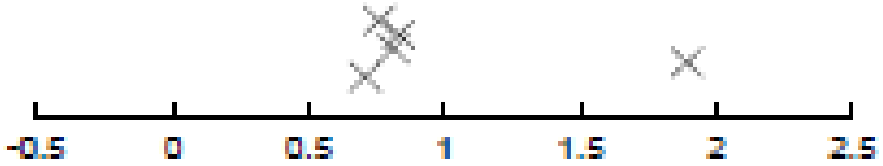
$$\hat{\theta}_{ML} = \arg\max_{\theta} \ P(\boldsymbol{x}; \theta)$$

$$= \arg\max_{\theta} \ \sum_{\boldsymbol{z}} P(\boldsymbol{x}, \boldsymbol{z}; \theta)$$

Recall: MM Tasks – Example (Mixture of two 1D Gaussians - GMM)







[DJM]

Outline of EM Algo. & Appns.

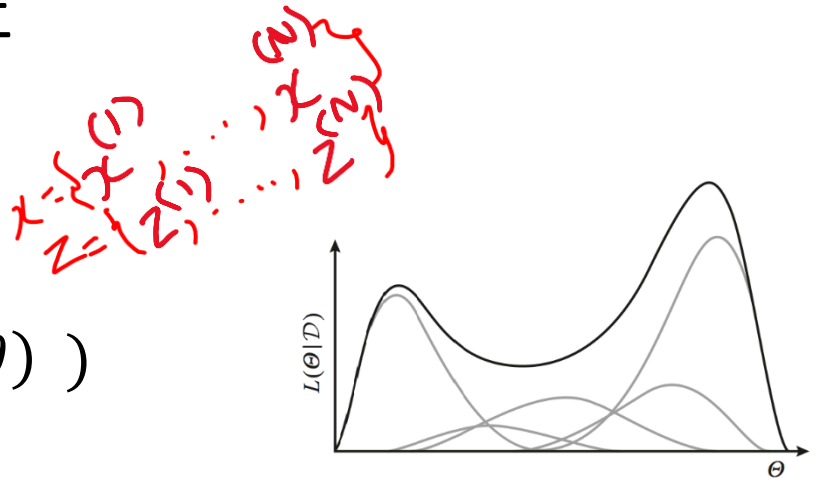
- **Expectation-Maximization (EM) Algorithm (Algo.)**
  - Background: Mixture Model (MM) or Latent Variable Model (LVM)
  - **Generic algorithm for parameter-learning of a MM**
  - Algorithm analysis (correctness)

- Bioinformatics Applications (Appns.) of EM Algo.
  - Application 0: Bernoulli mixture of two coins (toy/warmup appn.)
  - Application 1: EM meets motif-finding in DNA sequences
  - Application 2: Soft k-means clustering of gene expression data (revisit appn.)
  - Application 3: Genes/isoforms' expression quantitation (deconvolution)
  - Application 4: Haplotype estimation from genotype data
  - .
  - .
  - .

Goal: Parameter Learning of a Mixture model (aka MM or LVM such as GMM) via MLE

$$\hat{\theta}_{ML} = \arg\max_{\theta} \; \mathbf{log} \; (\; P(\boldsymbol{x} \, ; \, \theta) \;)$$

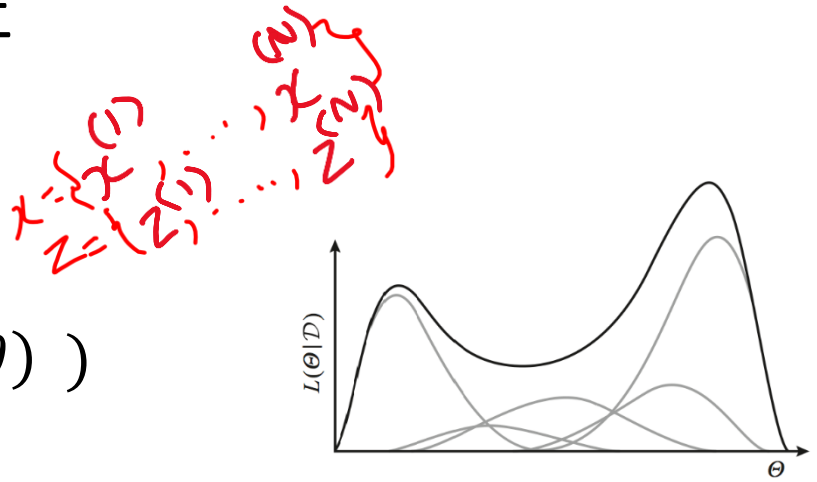
$$= \arg\max_{\theta} \; \mathbf{log} \; (\; \sum_{\boldsymbol{z}} \; P(\boldsymbol{x}, \boldsymbol{z} \, ; \, \theta) \;)$$



[ECL]

Goal: Parameter Learning of a Mixture Model (aka MM or LVM such as GMM) via MLE

$$\hat{\theta}_{ML} = \arg\max_{\theta} \; \mathbf{log} \; (\; P(\boldsymbol{x}\,;\,\theta) \;)$$
$$= \arg\max_{\theta} \; \mathbf{log} \; (\; \textstyle\sum_{\boldsymbol{z}} \; P(\boldsymbol{x}, \boldsymbol{z}\,;\,\theta) \;)$$



If iid:

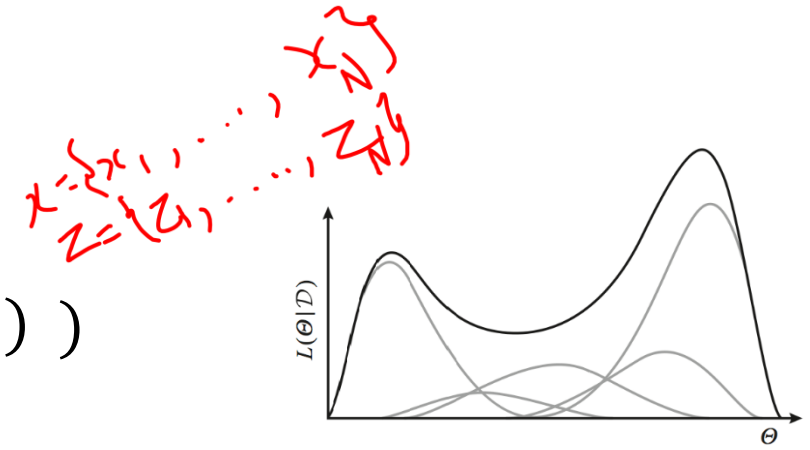
$$P(\boldsymbol{x};\theta) = \prod_{n=1}^{N} P(\boldsymbol{x}^{(n)};\theta)$$
$$= \prod_{n} \sum_{z=1}^{K} P(\boldsymbol{x}^{(n)}, z^{(n)};\theta)$$

[ECL]

Parameter-learning of a mixture model: Why is it hard?

$$\hat{\theta}_{ML} = \arg\max_{\theta} \; \mathbf{log} \; (\; P(\boldsymbol{x} \, ; \, \theta) \;)$$

$$= \arg\max_{\theta} \; \mathbf{log} \; (\; \sum_{\boldsymbol{z}} \; P(\boldsymbol{x}, \boldsymbol{z} \, ; \, \theta) \;)$$



- "log of sum"
  - calls for an approximate learning approach
  - EM algo. is one such approach that pushes the log across the sum!

[ECL]

How to solve the MLE optimization problem of a mixture model in general?

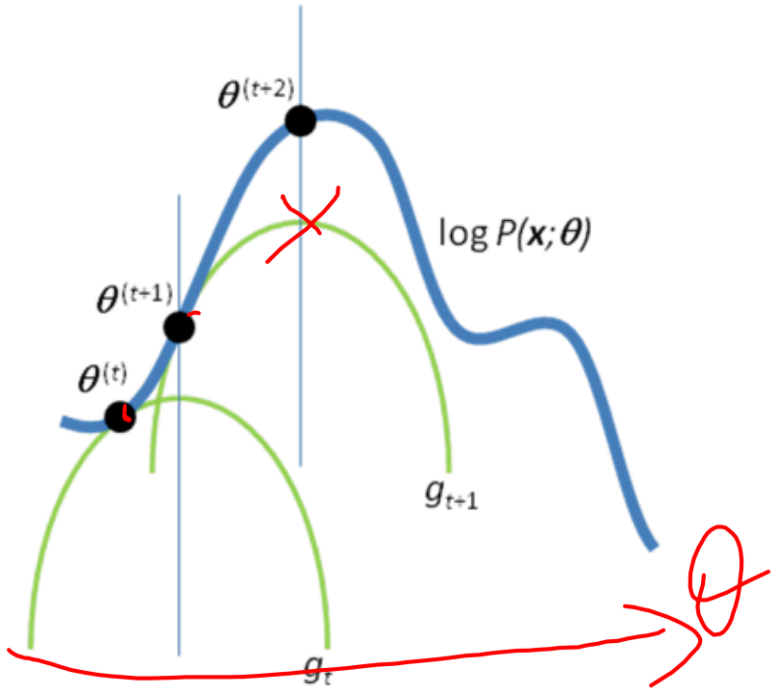
Why is loglikl. **hard to optimize**?: $\hat{\theta}_{ML} = \arg\max_{\theta} \quad \mathbf{log}\ (\ \sum_{\boldsymbol{z}}\ P(\boldsymbol{x}, \boldsymbol{z}\ ;\ \theta)\ )$

EM algorithm finds a stationary point (mostly a local maxima) of the log likelihood function.

Pushes log across the sum

Finds a series of "nice" lower bounds (iterative E/M steps)

.



[from Do & Batzoglou. What is the expectation maximization algorithm?. *Nat Biotechnol* 2008]

How to solve the MLE optimization problem of a mixture model in general?

Why is loglikl. **hard to optimize**?: $\hat{\theta}_{ML} = \arg\max_{\theta} \quad \mathbf{log} \ (\ \sum_{\boldsymbol{z}} \ P(\boldsymbol{x}, \boldsymbol{z} \, ; \, \theta) \ )$

EM algorithm finds a stationary point (mostly a local maxima) of the log likelihood function.

Pushes log across the sum

Finds a series of "nice" lower bounds (iterative E/M steps)

E-step: q_t

M-step: θ_t

$\rightarrow$ q_{t+1} . . .

θ_{t+1}

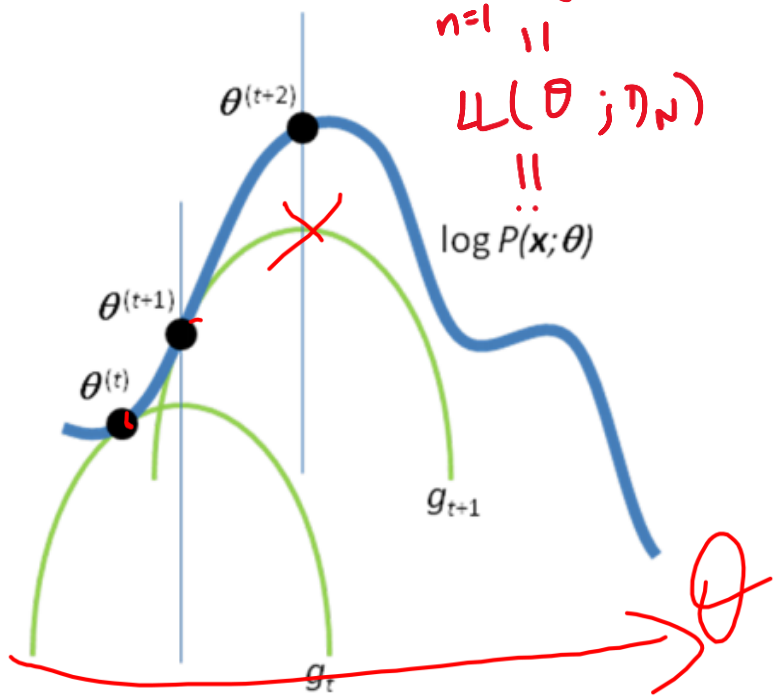
$\sum_{n=1}^{N} \log P(x^{(n)}; \theta)$

$=$

$LL(\theta \, ; \eta_N)$

$=$

$\log P(\boldsymbol{x}; \theta)$



[from Do & Batzoglou. What is the expectation maximization algorithm?. *Nat Biotechnol* 2008]

EM algo. – generic version (*iid* data)

- Versatile and wide-spread approach for
  - (approximate) learning of parameters θ in not just Gaussian Mixture Models (GMM), but also any other MM or LVM; and so
  - enjoys many applications in bioinformatics & beyond

- EM algo. – generic version pseudocode:

  - Starting at an initial θ_0, repeat until convergence for $t = 1, 2, \ldots$:
  - *E-Step*: For each $x \in D$, compute the posterior $p(z \mid x; \theta_t)$.
  - *M-Step*: Compute new weights via

$$\theta_{t+1} = \arg\max_{\theta} \sum_{x \in D} \mathbb{E}_{z \sim p(z|x;\theta_t)} \log p(x, z; \theta).$$

[ECL]

Outline of EM Algo. & Appns.

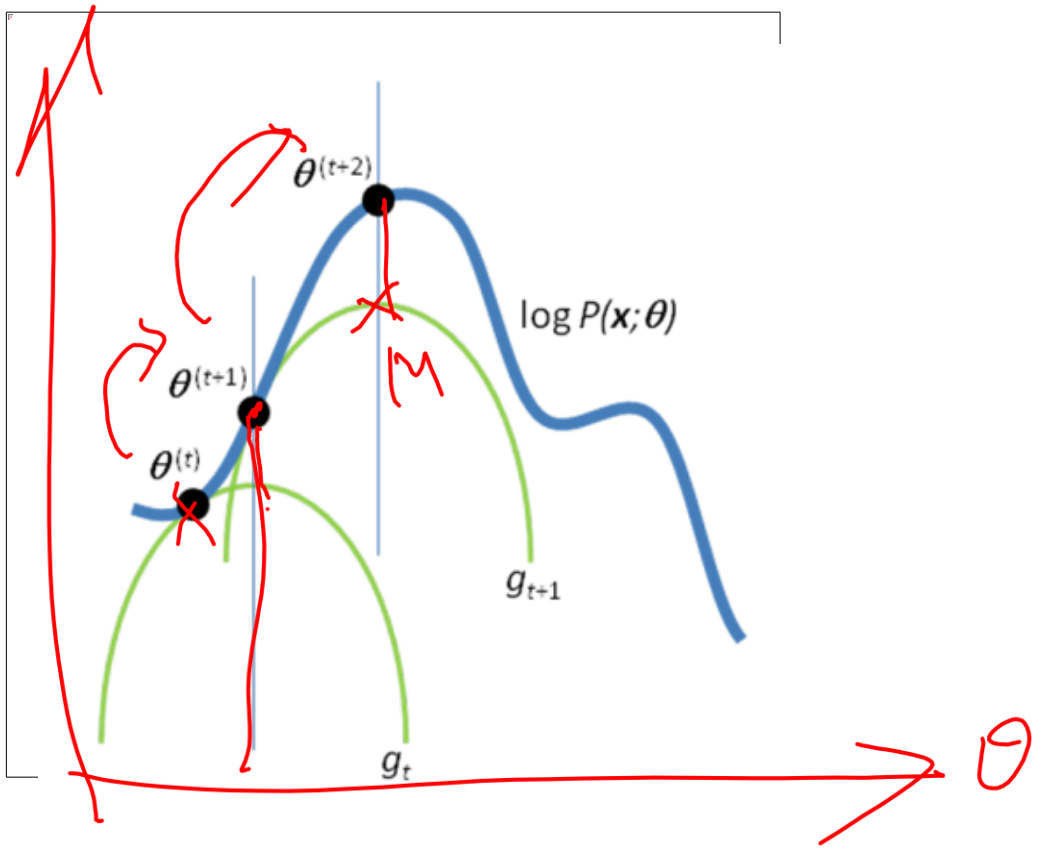
- **Expectation-Maximization (EM) Algorithm (Algo.)**
  - Background: Mixture Model (MM) or Latent Variable Model (LVM)
  - Generic algorithm description
  - **Algorithm analysis (correctness)**

- Bioinformatics Applications (Appns.) of EM Algo.
  - Application 0: Bernoulli mixture of two coins (toy/warmup appn.)
  - Application 1: EM meets motif-finding in DNA sequences
  - Application 2: Soft k-means clustering of gene expression data (revisit appn.)
  - Application 3: Genes/isoforms' expression quantitation (deconvolution)
  - Application 4: Haplotype estimation from genotype data
  - .
  - .
  - .

EM algorithm Correctness (Idea in pictures)

$$\hat{\theta}_{ML} = \arg\max_{\theta} \quad \mathbf{log} \; \left(\sum_{\boldsymbol{z}} P(\boldsymbol{x}, \boldsymbol{z}\,;\, \theta) \right)$$

Find a series of “nice” lower bounds (based on Jensen’s Inequality)!



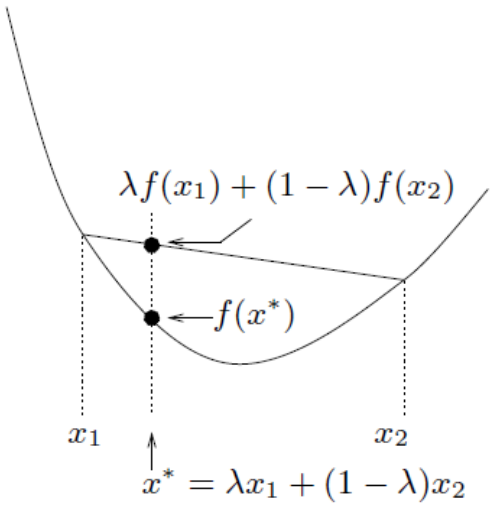
[DB08]

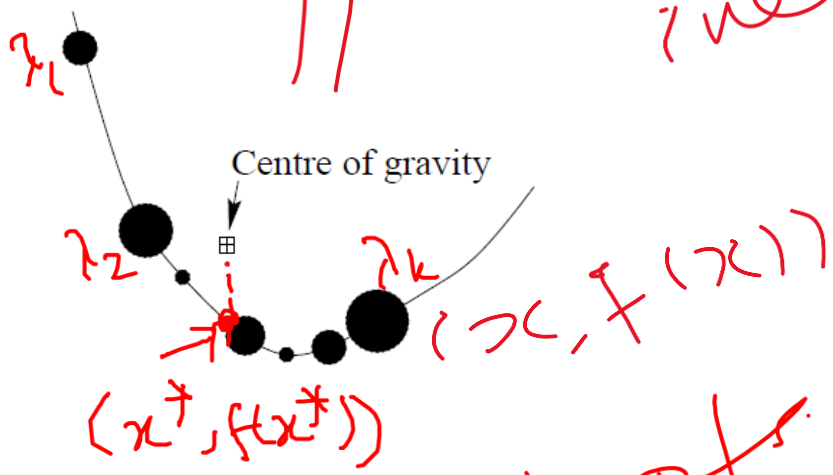
Jensen’s inequality background

Jensen's inequality for convex functions f – in pictures

$$\mathcal{E}\left[f(x)\right] \geq f(\mathcal{E}[x])\,,$$

Jensen's ineq.





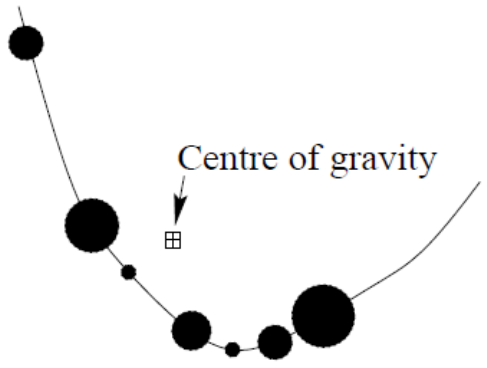
Defn. of convex fn.

multiple pts

[DJM]

When is Jensen's inequality an equality??

$$\mathcal{E}\left[f(x)\right] \geq f(\mathcal{E}[x])\,,$$



Jensen's inequality - formally

Convex ⌣ functions. A function $f(x)$ is *convex* ⌣ over (a, b) if every chord of the function lies above the function, as shown in figure 2.10; that is, for all $x_1, x_2 \in (a, b)$ and $0 \leq \lambda \leq 1$,

$$f(\lambda x_1 + (1 - \lambda)x_2) \quad \leq \quad \lambda f(x_1) + (1 - \lambda) f(x_2). \qquad (2.47)$$

A function f is *strictly convex* ⌣ if, for all $x_1, x_2 \in (a, b)$, the equality holds only for $\lambda = 0$ and $\lambda = 1$.

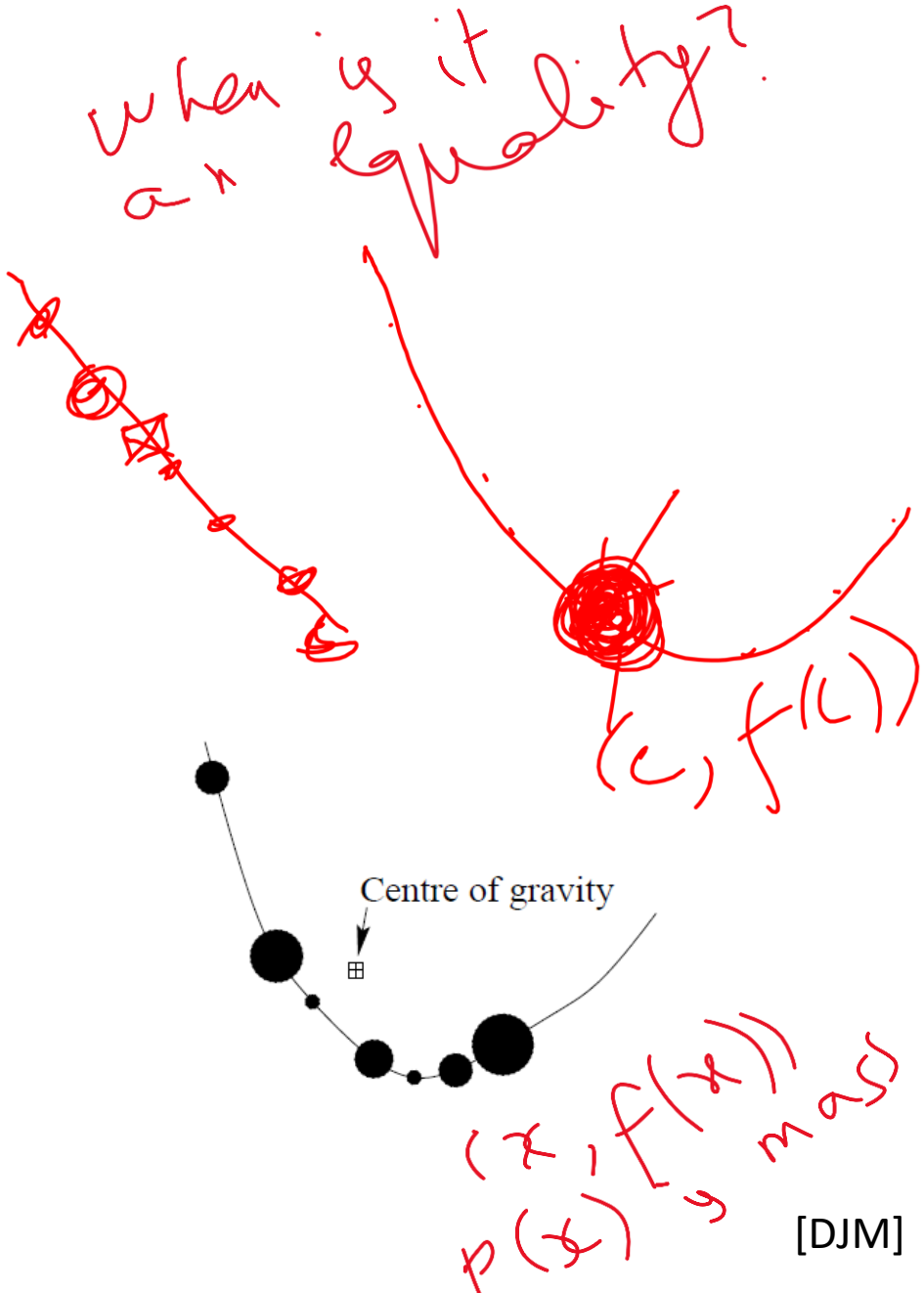
Similar definitions apply to concave ⌢ and strictly concave ⌢ functions.

Jensen's inequality. If f is a convex ⌣ function and x is a random variable then:

$$\mathcal{E}\left[f(x)\right] \geq f(\mathcal{E}[x])\,, \qquad (2.48)$$

where $\mathcal{E}$ denotes expectation. If f is strictly convex ⌣ and $\mathcal{E}\left[f(x)\right] = f(\mathcal{E}[x])$, then the random variable x is a constant.

Jensen's inequality can also be rewritten for a concave ⌢ function, with the direction of the inequality reversed.



[DJM]

EM algorithm Correctness (Idea in eqns):

- Jensen's inequality (RHS is called Evidence Lower BOund or ELBO):

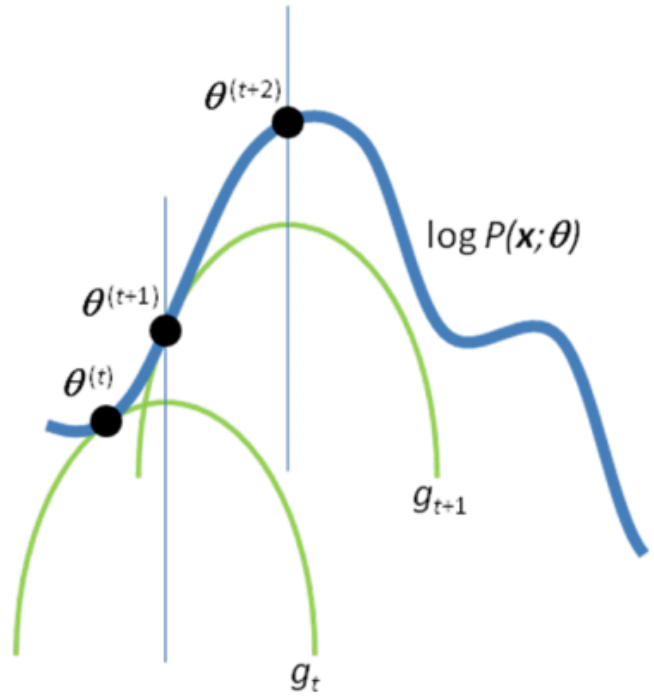
$$\log\left(\sum_z P(x, z; \theta)\right) = \log\left(\sum_z Q(z) \cdot \frac{P(x, z; \theta)}{Q(z)}\right) \geq \sum_z Q(z) \log\left(\frac{P(x, z; \theta)}{Q(z)}\right).$$

[DB08]

EM algo. Correctness – view as alternate max.

- E-step maximizes ELBO over $Q(z)$ (setting it to $P(z|x;\theta_t)$), and
- M-step maximizes ELBO over θ $\quad \theta^{(t+1)} = \arg\max_{\theta} g_t(\theta)$ where

$$g_t(\theta) = \sum_z P\left(z|x;\theta^{(t)}\right) \log\left(\frac{P(x,z;\theta)}{P\left(z|x;\theta^{(t)}\right)}\right).$$



[DB08]

Blank space for illustration

- To prove:

EM algo. convergence to a **stationary** point

$\log P(x;\theta) \geq \sum_z Q(z) \log \frac{P(x,z;\theta)}{Q(z)}$ $\forall x, Q, \theta$ $\Leftarrow$ g_t: Tight ELBO at $\theta^{(t)}$

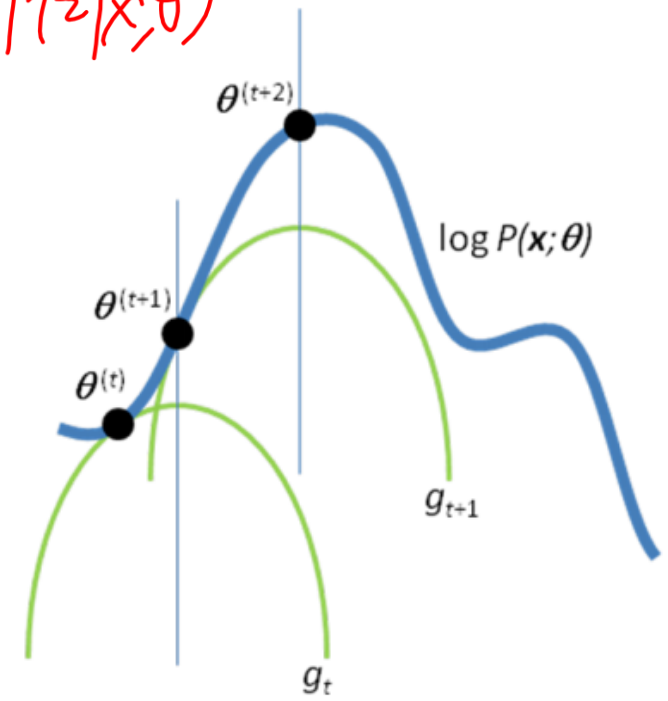
$Q(z) = P(z|x;\theta^t)$

To prove: $\log P(x;\theta^{t+1}) \geq \log P(x;\theta^t)$

$\log P(x;\theta^{t+1}) \geq g_t(\theta^{t+1})$ ELBO

$\geq g_t(\theta^t)$ max. g_t

$= \log P(x;\theta^t)$ tight ELBO



Alternate resource for EM algo.: http://cs229.stanford.edu/notes-spring2019/cs229-notes8-2.pdf

Two final details in the algorithm's correctness

- M-step ($\mathrm{argmax}_\theta\ \mathrm{g_t}(\theta)$) can be simplified
- EM algo. also works with non-iid data (e.g., Hidden Markov Models with structured data)

Recap: EM algo. – generic version (*iid* data)

- Versatile and wide-spread approach for
  - (approximate) learning of parameters θ in not just GMM, but also other LVMs; and so
  - enjoys many appns. in clustering, bioinformatics & beyond

- EM algo. – generic version pseudocode:

  - Starting at an initial θ_0, repeat until convergence for $t = 1, 2, \ldots$:
  - *E-Step*: For each $x \in D$, compute the posterior $p(z \mid x; \theta_t)$.
  - *M-Step*: Compute new weights via

$$\theta_{t+1} = \arg\max_{\theta} \sum_{x \in D} \mathbb{E}_{z \sim p(z|x;\theta_t)} \log p(x, z; \theta).$$

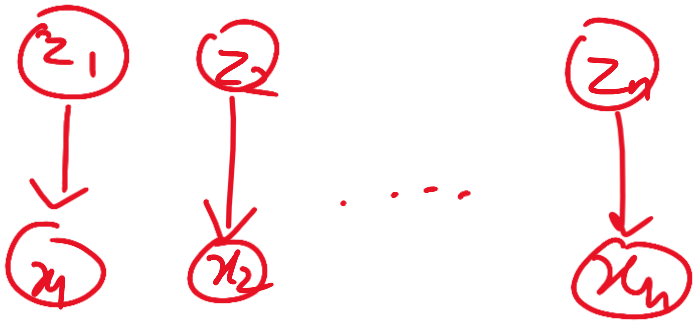
$\tilde{g}_t(\theta)$ [$\tilde{g}_t$: ∇_θ of log not here]

[ECL]

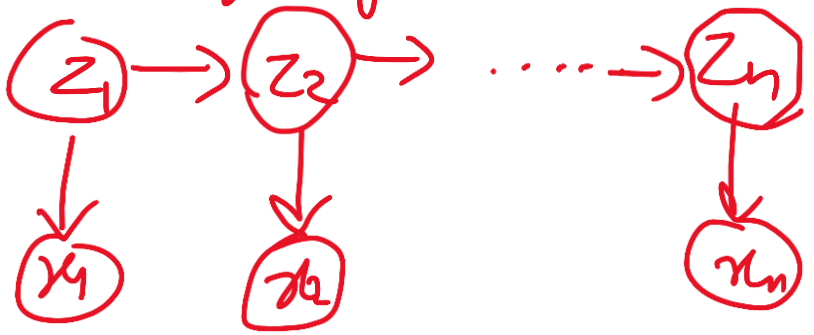
Brief aside: iid *vs.* structured data

If iid:

$$P(\mathbf{x};\theta) = \prod_{n=1}^{N} P(x_n;\theta) = \prod_n \sum_{z_n} P(x_n, z_n;\theta)$$



MLE (EM algo.) also works if data is not iid, e.g. if data is structured:



EM algo. – really generic version (non-*iid* or structured data)

- Versatile and wide-spread approach for
  - (approximate) learning of parameters θ in not just GMM, but also other LVMs; and so
  - enjoys many appns. in clustering, bioinformatics & beyond

- EM algo. – generic version pseudocode:

repeat:

- Starting at an initial θ_0, repeat until convergence for $t = 1, 2, \ldots$:
- *E-Step*: For each $x \in D$, compute the posterior $p(z \mid x; \theta_t)$.
- *M-Step*: Compute new weights via

$$\theta_{t+1} = \arg\max_{\theta} \sum_{x \in D} \mathbb{E}_{z \sim p(z|x;\theta_t)} \log p(x, z; \theta).$$

$\vec{x}$: $p(\vec{z} \mid \vec{x}; \theta_t)$

$\theta_{t+1} = \arg\max_{\theta} E_{\vec{z} \sim p(\vec{z}|\vec{x};\theta_t)} [\log p(\vec{x}, \vec{z}; \theta)]$

Fine print: For certain models like HMM where data points $\boldsymbol{x} = \{x\}_{x \in D}$ are **not** iid, and $\boldsymbol{z} = \{z\}$ are also **not** indep., we deal with the whole dataset like so: $\theta_{t+1} = argmax_{\theta}\, E_{\boldsymbol{z} \sim p(\boldsymbol{z}|\boldsymbol{x};\theta_t)} \log p(\boldsymbol{x}, \boldsymbol{z}; \theta)$.

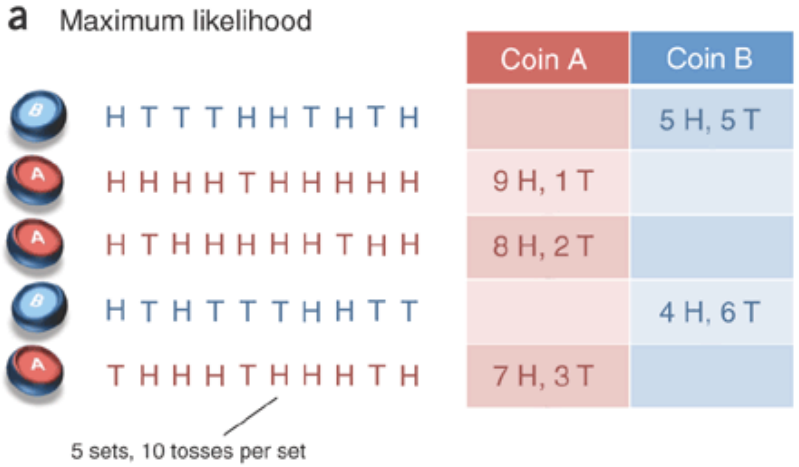
[ECL]

Outline of EM Algo. & Appns.

- Expectation-Maximization (EM) Algorithm (Algo.)
  - Background: Mixture Model (MM) or Latent Variable Model (LVM)
  - Generic algorithm description
  - Algorithm analysis (correctness)

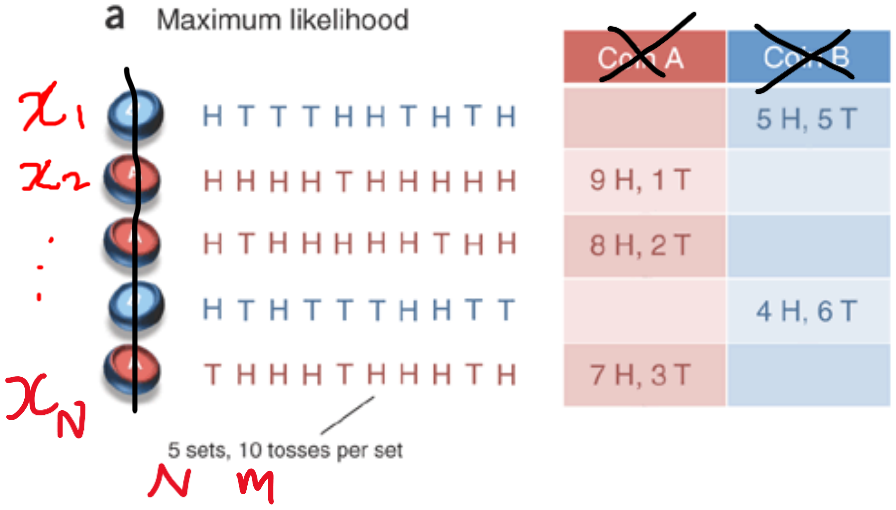
- **Bioinformatics Applications (Appns.) of EM Algo.**
  - **Application 0: Bernoulli mixture of two coins (toy/warmup appn.)**
  - Application 1: EM meets motif-finding in DNA sequences
  - Application 2: Soft k-means clustering of gene expression data (revisit appn.)
  - Application 3: Genes/isoforms' expression quantitation (deconvolution)
  - Application 4: Haplotype estimation from genotype data
  - .
  - .
  - .

Bernoulli mixture (of two coins): generative model



[DB08]

Bernoulli mixture (of two coins): inference

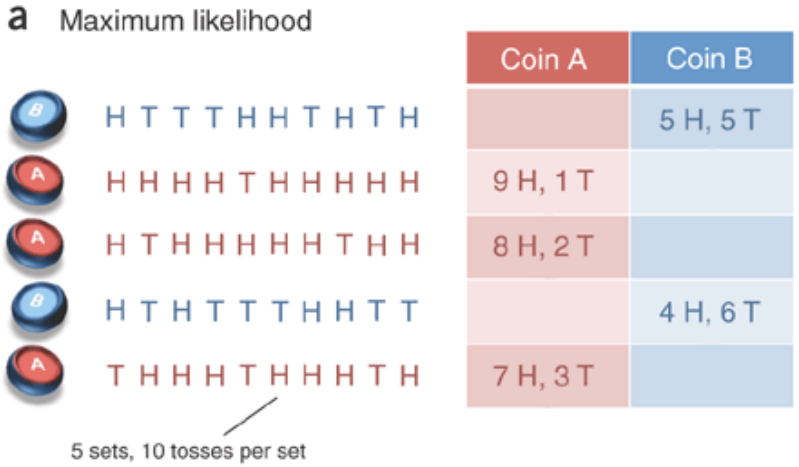


x_n = # of heads in n^{th} set

m = # of tosses per set

[DB08]

Bernoulli mixture (of two coins): parameter-learning

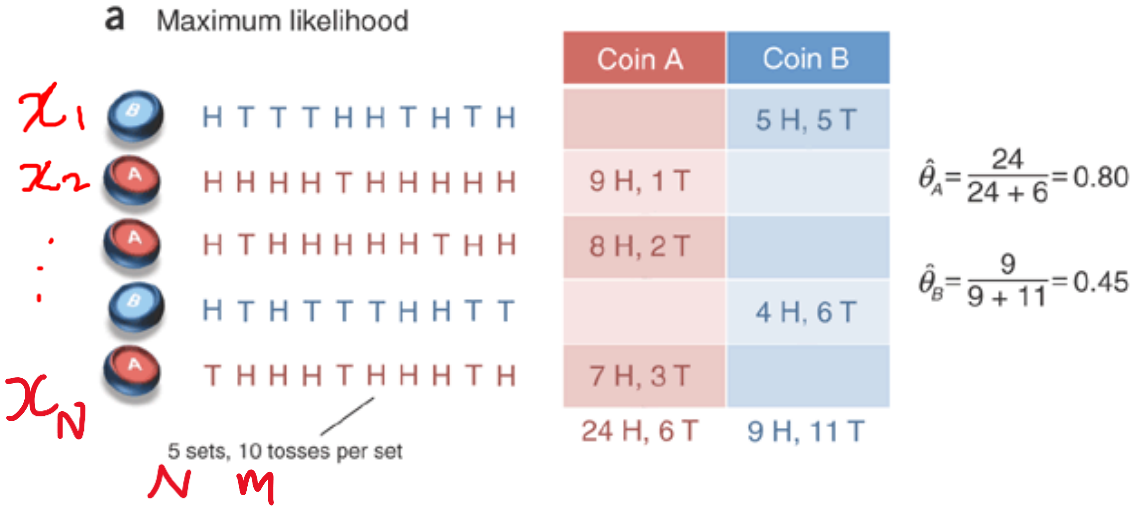


MLE from (i) complete data (warmup), and

(ii) observed data (using EM algorithm)

[DB08]

Bernoulli mixture (of two coins): MLE from complete data (observed data $\boldsymbol{x}$, and labels $\boldsymbol{z}$)

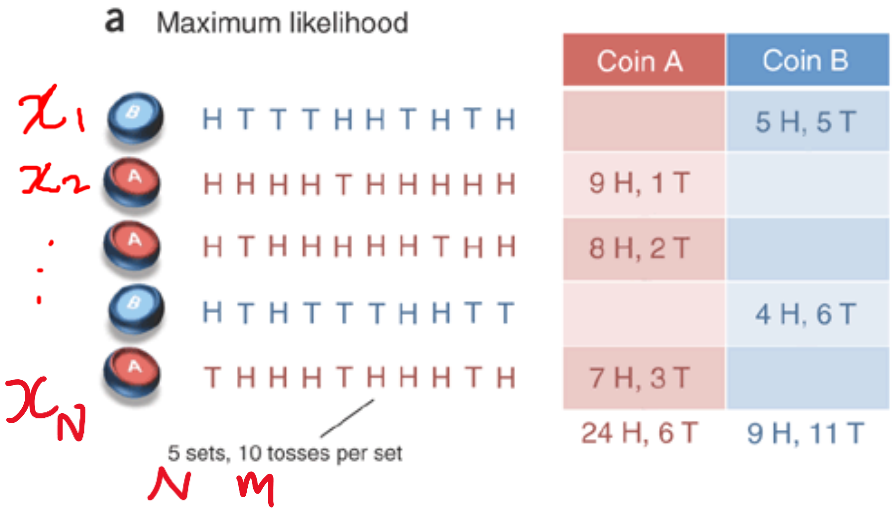


$$\mathcal{L}(\theta_1, \theta_2 | \vec{x}, \vec{z}) = \prod_{n=1}^{N} \theta_{z_n}^{x_n} (1-\theta)_{z_n}^{m-x_n} \cdot \left(\frac{1}{2}\right)$$

$$\hat{\theta}_k = \frac{\sum_{n=1}^{N} r_k(n)\, x_n}{R_k \cdot M}$$

[DB08]

Bernoulli mixture (of two coins): MLE from observed data (obs. $\boldsymbol{x}$, but labels $\boldsymbol{z}$ **hidden**) – calls for EM algo.



$$\mathcal{L}(\theta_1, \theta_2 | \vec{x}, \cancel{\vec{z}}) = \prod_{n=1}^{N} \sum_{z_n} \theta_{z_n}^{x_n} (1-\theta)_{z_n}^{m-x_n} \left(\frac{1}{2}\right)$$

[DB08]

Recap: EM algo. – generic version (*iid* data)

- Versatile and wide-spread approach for
  - (approximate) learning of parameters θ in not just GMM, but also other LVMs; and so
  - enjoys many appns. in clustering, bioinformatics & beyond

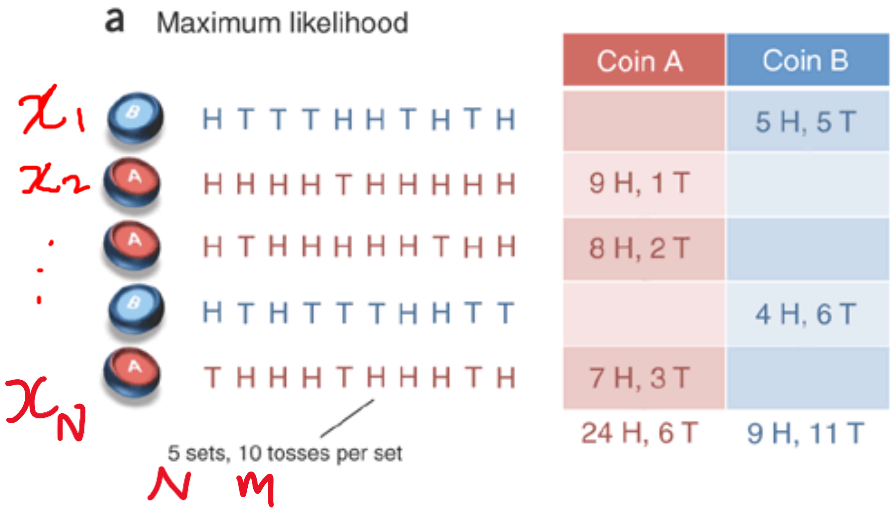
- EM algo. – generic version pseudocode:

  - Starting at an initial θ_0, repeat until convergence for $t = 1, 2, \ldots$:
  - *E-Step*: For each $x \in D$, compute the posterior $p(z \mid x; \theta_t)$.
  - *M-Step*: Compute new weights via

$$\theta_{t+1} = \arg\max_{\theta} \sum_{x \in D} \mathbb{E}_{z \sim p(z|x;\theta_t)} \log p(x, z; \theta).$$

[ECL]

Bernoulli mixture (of two coins): M-step optimization: MLE from **hallucinated/expected** complete data



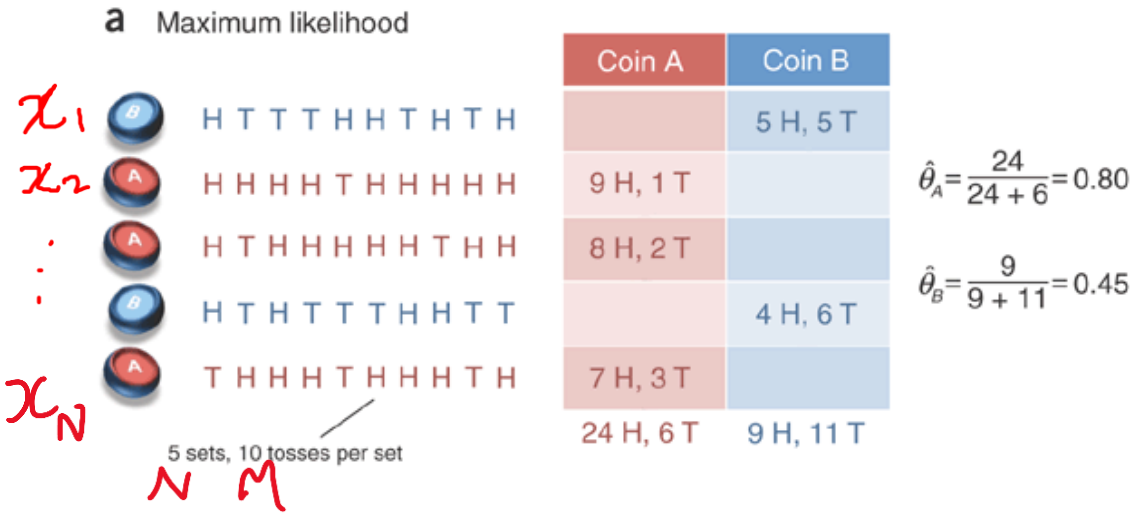
Hallucinated/expected complete data := obs. data $\boldsymbol{x}$ + hidden labels $\boldsymbol{z}$ "drawn/hallucinated"

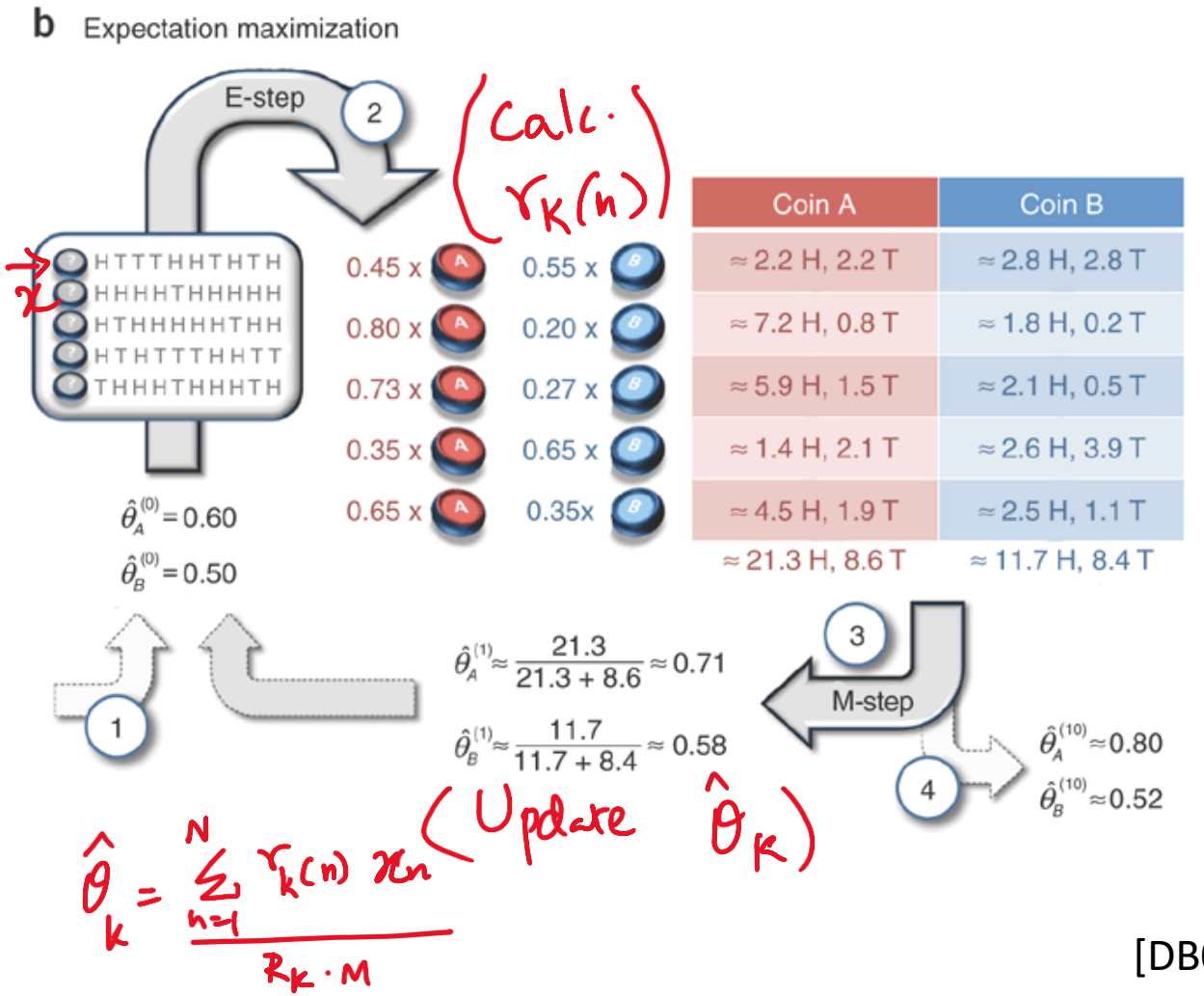
from $P(\boldsymbol{z} \mid \boldsymbol{x}; \theta^t)$

$$\hat{\theta}_k = \frac{\sum_{n=1}^{N} r_{k(n)} x_n}{R_k \cdot M}$$

[DB08]

Bernoulli mixture (of two coins): EM Algo.





$$\hat{\theta}_k = \frac{\sum_{n=1}^{N} r_k(n)\, x_n}{r_k \cdot M}$$

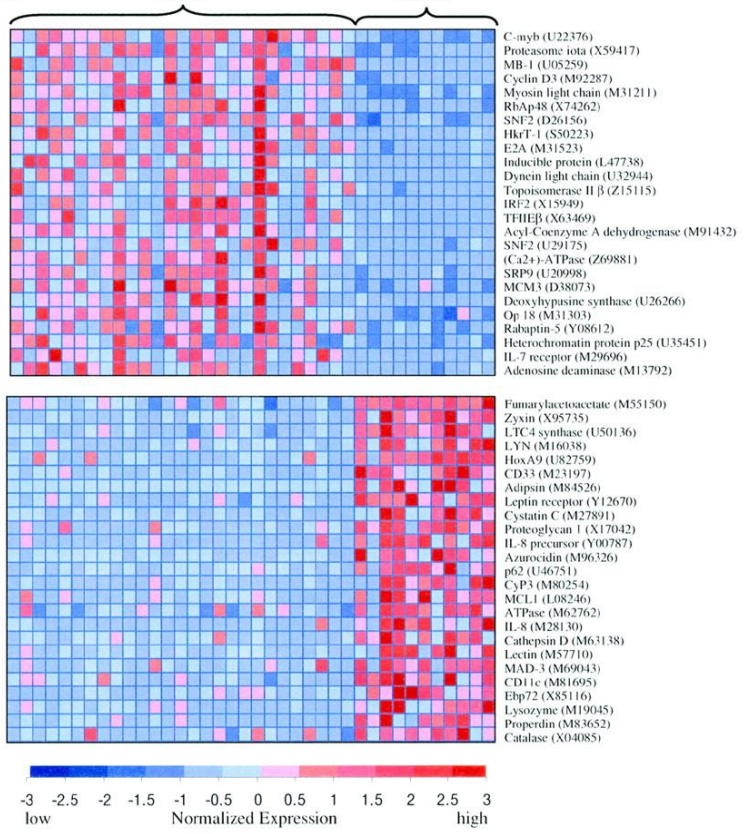
[DB08]

Outline of EM Algo. & Appns.

- Expectation-Maximization (EM) Algorithm (Algo.)
  - Background: Mixture Model (MM) or Latent Variable Model (LVM)
  - Generic algorithm description
  - Algorithm analysis (correctness)

- **Bioinformatics Applications (Appns.) of EM Algo.**
  - Application 0: Bernoulli mixture of two coins (toy/warmup appn.)
  - Application 1: EM meets motif-finding in DNA sequences
  - **Application 2: Soft k-means clustering of gene expression data (revisit appn.)**
  - Application 3: Genes/isoforms' expression quantitation (deconvolution)
  - Application 4: Haplotype estimation from genotype data
  - .
  - .
  - .

Cancer example: Can we group cancer samples into different (sub)types?



Golub et al. Science 1999

Recall: MLE for one 1D Gaussian (closed-form soln.) – *How to change it to learn mixture of Gaussians?*

- Log likelihood:

$$\ln P\left(\left\{x^{(n)}\right\}_{n=1}^{N} \mid \mu, \sigma\right) = -N \ln(\sqrt{2\pi}\sigma) - \sum_{n} \left(x^{(n)} - \mu\right)^2 / (2\sigma^2)$$

- MLE estimates:

$$\hat{\mu} = \sum_{n=1}^{N} \frac{x^{(n)}}{N}, \quad \hat{\sigma}_N^2 = \frac{\sum_{n=1}^{N} \left(x^{(n)} - \hat{\mu}\right)^2}{N}$$

Recall: MLE for one 1D Gaussian (closed-form soln.) – *How to change it to learn mixture of Gaussians?*

- Log likelihood:

$$\ln P\left(\left\{x^{(n)}\right\}_{n=1}^{N} \mid \mu, \sigma\right) = -N \ln(\sqrt{2\pi}\sigma) - \sum_{n} \left(x^{(n)} - \mu\right)^2 / (2\sigma^2)$$

θ^t = current params. $r_k^{(n)} = P(z^{(n)} = k \mid x = x^{(n)}; \theta^t)$

↓

$\theta^{t+1} = ?.?.$

- MLE estimates:

$$\hat{\mu}_N^{(k)} = \sum_{n=1}^{N} \frac{r_k^{(n)} x^{(n)}}{N r^{(k)}}, \quad \hat{\sigma}_N^{2\,(k)} = \frac{\sum_{n=1}^{N} r_k^{(n)} \left(x^{(n)} - \hat{\mu}\right)^2}{N r^{(k)}}$$

(wted. sample mean & variance)

EM-algorithm in Action: Parameter-learning of a Mixture of 1D Gaussians (aka univariate GMM)

Calculate:

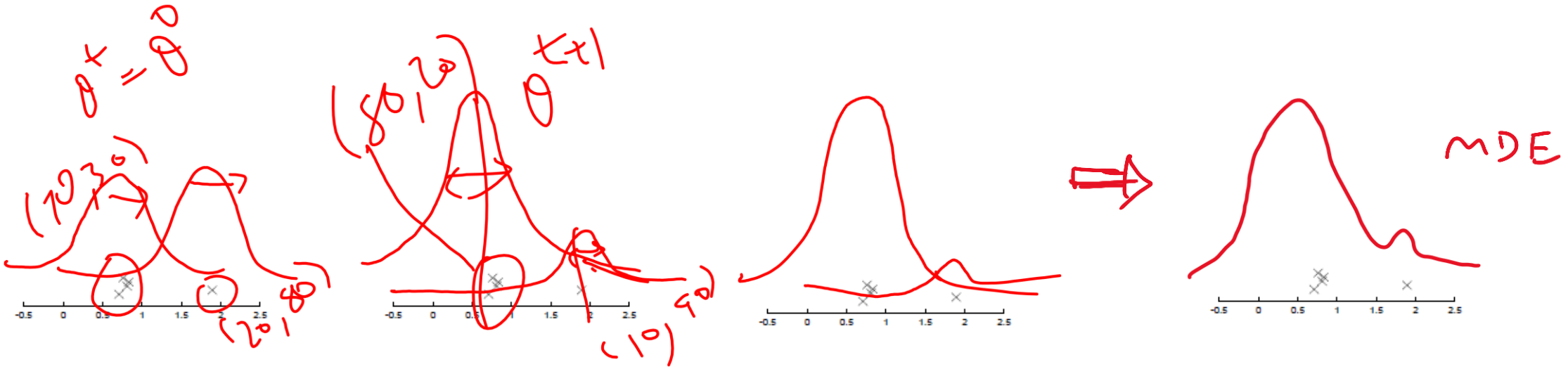
$$r_k(n) = p(z=k \mid \theta^t, x) = \frac{\pi_k N(x \mid \mu_k, \sigma_k^2)}{\sum_{k'=1}^{K} \pi_{k'} N(x \mid \mu_{k'}, \sigma_{k'}^2)}$$

E-step (Assign)

$$\hat{\mu}_k = \sum_{n=1}^{N} \frac{r_k(n) x_n}{R_k}, \quad \hat{\sigma}_k^2 = \sum_{n=1}^{N} \frac{r_k(n)}{R_k} (x_n - \hat{\mu}_k)^2$$

$$\pi_k = \frac{R_k}{\sum_{k'} R_{k'}} = \frac{R_k}{N} \qquad (\theta^{t+1})$$

M-step (Update)



[ECL]

EM-algorithm in Action: Parameter-learning of a Mixture of 1D Gaussians (aka univariate GMM)

Calculate:

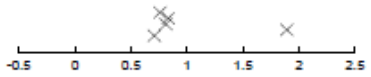
$$r_k(n) = p(z=k \mid \theta^t, x) = \frac{N(x \mid \mu_k, \sigma_k^2)}{\sum_{k'=1}^{K} N(x \mid \mu_{k'}, \sigma_{k'}^2)}$$

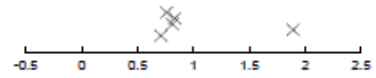
E-step (Assign)

$$\hat{\mu}_k = \sum_{n=1}^{N} \frac{r_k(n) x_n}{R_k} ;$$

(θ_{t+1})

M-step (Update)





[ECL]

EM algo. – GMM version
(aka) **Vanilla Soft K-means**

Initialization: Set K means $\{m^{(k)}\}$ to random values.

Assignment:

(E-step)

$$r_k^{(n)} = \frac{\exp\left(-\beta\, d(\mathbf{m}^{(k)}, \mathbf{x}^{(n)})\right)}{\sum_{k'} \exp\left(-\beta\, d(\mathbf{m}^{(k')}, \mathbf{x}^{(n)})\right)}.$$

(here $\beta = 1/(2\sigma^2)$)

Prove: Lt $r_k^{(n)S} \simeq r_k^{(n)h}$ as $\beta \to \infty$ (Hard kmeans)

Update:

(M-step)

$$\mathbf{m}^{(k)} = \frac{\sum_n r_k^{(n)} \mathbf{x}^{(n)}}{R^{(k)}}$$

where $R^{(k)}$ is the total responsibility of mean k,

$$R^{(k)} = \sum_n r_k^{(n)}.$$

[DJM]

Thank you!

Backup slides

EM algorithm Correctness (Idea in eqns)

- Jensen's inequality: RHS is called ELBO (Evidence Lower Bound).

$$\log P(x;\theta) = \log\left(\sum_z P(x,z;\theta)\right) = \log\left(\sum_z Q(z)\left[\frac{P(x,z;\theta)}{Q(z)}\right]\right) \geq \sum_z Q(z)\left[\log\left(\frac{P(x,z;\theta)}{Q(z)}\right)\right].$$

$\geq = g(z)$

$\log(E_{z\sim Q}[z']) \leq$
$= \log(E_{z'}[z'])$
$\geq E_{z'}(\log[z'])$
$= E_z(\log z')$

Note: By law of the unconsc. statistician, $E_{g(z)}[g(z)] = E_z[g(z)]$.

[DB08]

When is Jensen's ineq. an equality **for us**?

$\log p(x;\theta) \geq \sum_z Q(z) \log \frac{p(x,z;\theta)}{Q(z)} \quad \forall\, x, Q, \theta$ (ELBO)

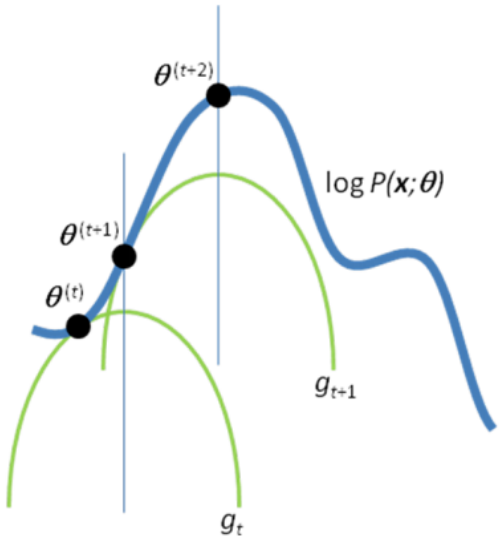
$\frac{p(x,z;\theta)}{Q(z)} = \text{const. } c$

$\Rightarrow Q(z) \propto p(x,z;\theta)$

$\sum_z Q(z) = 1$

$\Rightarrow Q(z) = \frac{p(x,z;\theta)}{\sum_z p(x,z;\theta)}$

$Q(z) = p(z|x;\theta)$



Alternate resource for EM algo.: http://cs229.stanford.edu/notes-spring2019/cs229-notes8-2.pdf