Maximum Likelihood Estimators and The Expectation Maximization Algorithm

LALEH HAGHVERDI

- Empirical Risk Minimisation
  - No assumption on the underlying data distribution
  - Aimed for a good predictor but not learning the distribution

$$\begin{aligned}
\text{Zero/one:} \quad & \ell^{(0/1)}(y,\hat{y}) = \mathbf{1}[y\hat{y} \leq 0] \\
\text{Hinge:} \quad & \ell^{(\text{hin})}(y,\hat{y}) = \max\{0, 1 - y\hat{y}\} \\
\text{Logistic:} \quad & \ell^{(\log)}(y,\hat{y}) = \frac{1}{\log 2} \log\left(1 + \exp[-y\hat{y}]\right) \\
\text{Exponential:} \quad & \ell^{(\exp)}(y,\hat{y}) = \exp[-y\hat{y}] \\
\text{Squared:} \quad & \ell^{(\text{sqr})}(y,\hat{y}) = (y - \hat{y})^2
\end{aligned}$$

Generative Models

- Assume data distribution has a specific parametric form, then learn its parameters
- Learn (prediction) on data with missing values or latent variables
- Can generate new data points
- Needs more training data

Maximum Likelihood estimation of the true probability density

- iid training set S with data points x_1,...,x_m sampled according to a probability density $\mathcal{P}$ that we assume has a certain parametric form $\mathcal{P}_\theta$
- Model paratmeters \theta
- Log-likelihood L

$$L(S;\theta) = \log\left(\prod_{i=1}^{m} \mathcal{P}_\theta(x_i)\right) = \sum_{i=1}^{m} \log(\mathcal{P}_\theta(x_i))$$

$$\underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{m} (-\log(\mathcal{P}_\theta[x_i])) = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{m} \log(\mathcal{P}_\theta[x_i])$$

- Unknown true Distribution $\mathcal{P}$

$$\ell(\theta, x) = -\log(\mathcal{P}_\theta[x])$$

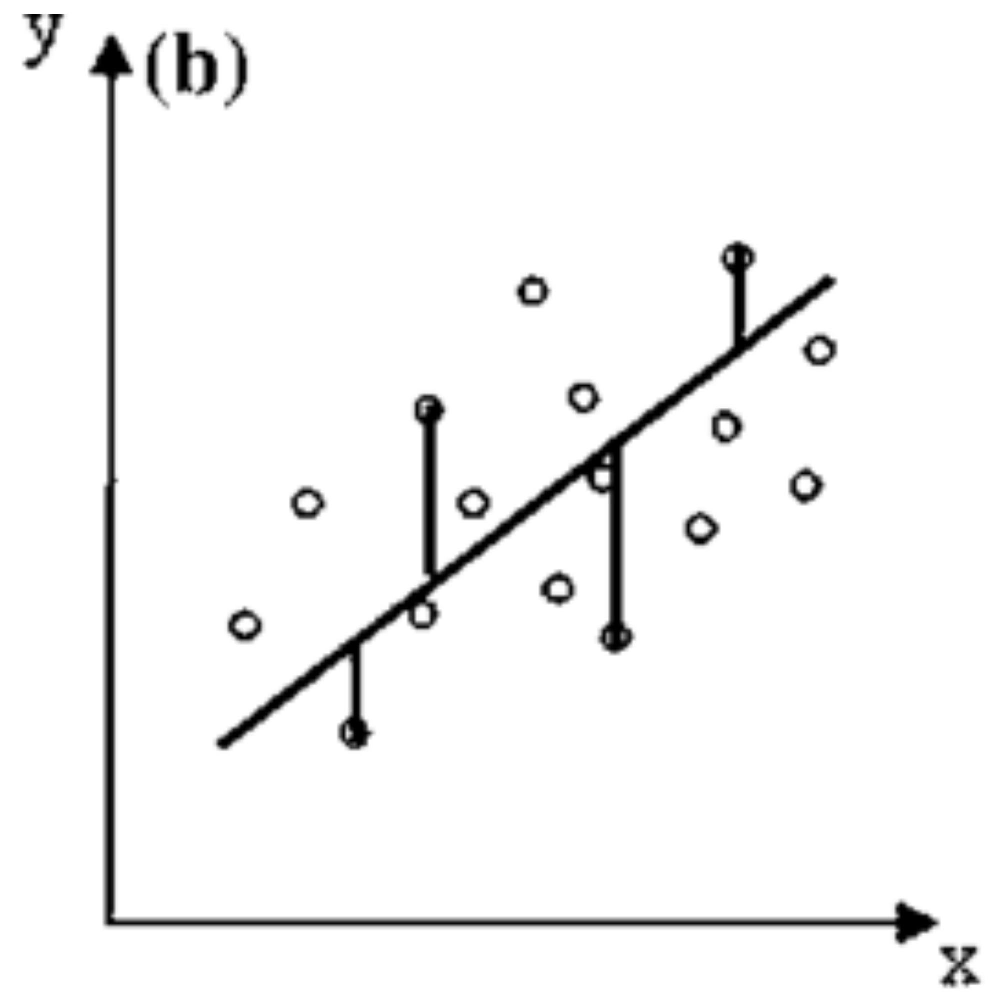
$$\mathbb{E}_x[\ell(\theta, x)] = -\sum_x \mathcal{P}[x] \log(\mathcal{P}_\theta[x])$$

$$= \underbrace{\sum_x \mathcal{P}[x] \log\left(\frac{\mathcal{P}[x]}{\mathcal{P}_\theta[x]}\right)}_{\text{Relative Entropy = KL (P\_theta | P)}} + \underbrace{\sum_x \mathcal{P}[x] \log\left(\frac{1}{\mathcal{P}[x]}\right)}_{H(\mathcal{P})}$$

- Minimal risk solution when P_theta = P
- Suboptimality measure: KL divergence

Least squares as a Maximum Likelihood Estimator

Dive into deep learning.
Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (Book)



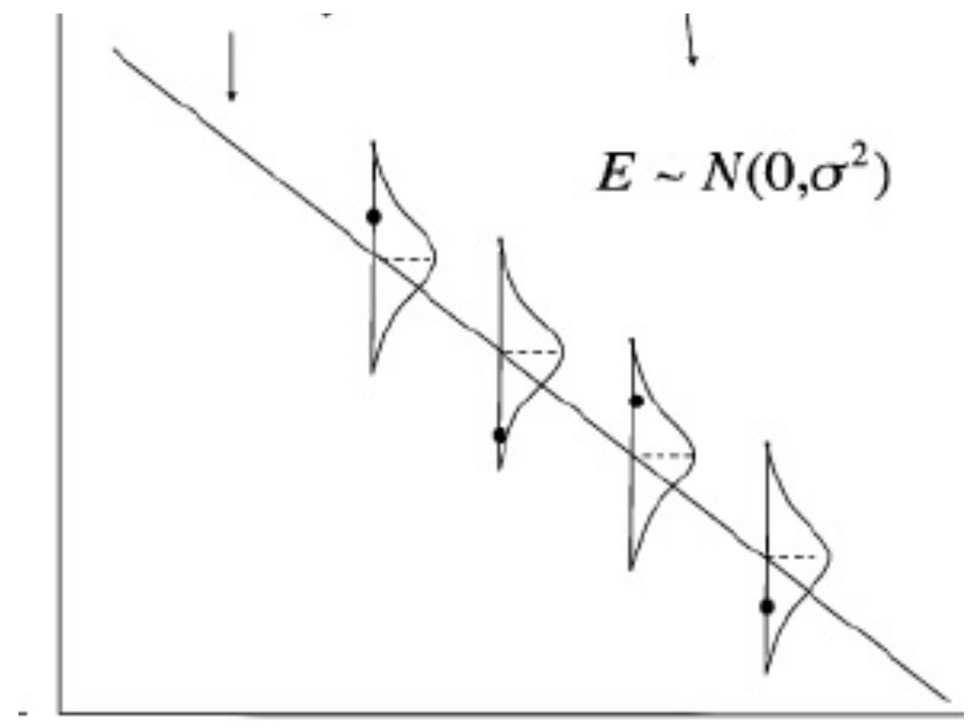
$$y = \mathbf{w}^\top \mathbf{x} + b + \epsilon \text{ where } \epsilon \sim \mathcal{N}(0, \sigma^2). \tag{3.1.12}$$

Thus, we can now write out the *likelihood* of seeing a particular y for a given $\mathbf{x}$ via

$$p(y|\mathbf{x}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y - \mathbf{w}^\top \mathbf{x} - b)^2\right). \tag{3.1.13}$$

Now, according to the *maximum likelihood principle*, the best values of b and $\mathbf{w}$ are those that maximize the *likelihood* of the entire dataset:

$$P(Y \mid X) = \prod_{i=1}^{n} p(y^{(i)}|\mathbf{x}^{(i)}). \tag{3.1.14}$$



Estimators chosen according to the *maximum likelihood principle* are called *Maximum Likelihood Estimators* (MLE). While, maximizing the product of many exponential functions, might look difficult, we can simplify things significantly, without changing the objective, by maximizing the log of the likelihood instead. For historical reasons, optimizations are more often expressed as minimization rather than maximization. So, without changing anything we can minimize the *Negative Log-Likelihood (NLL)* $-\log p(\mathbf{y}|\mathbf{X})$. Working out the math gives us:

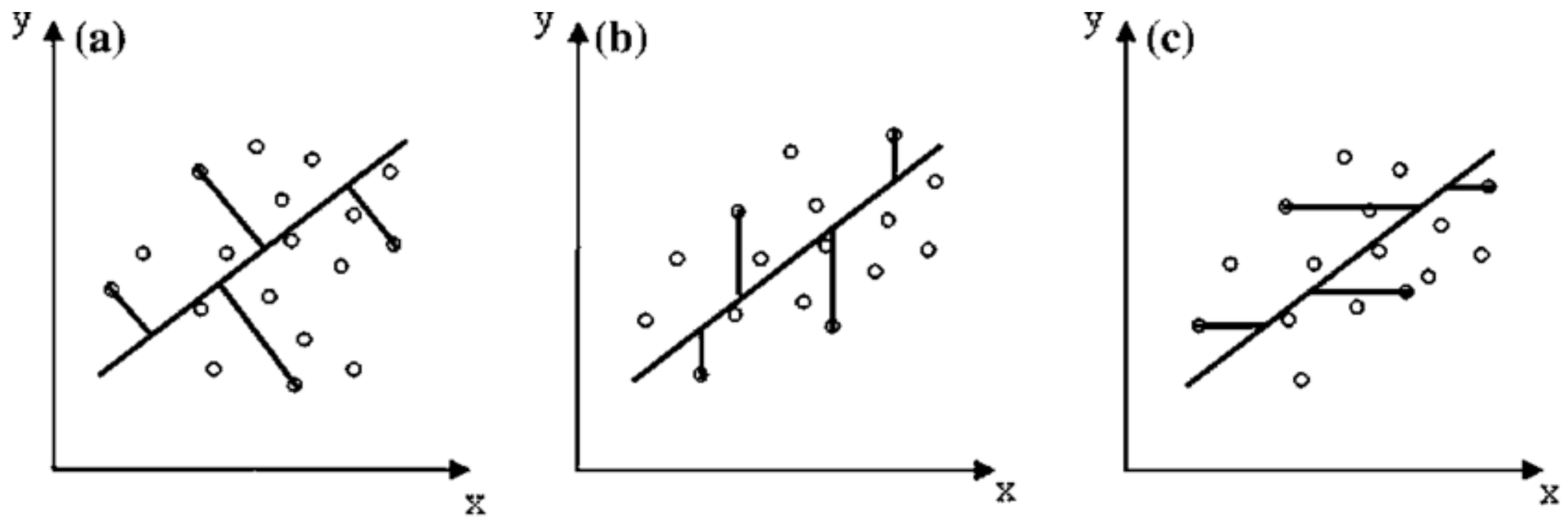
$$-\log p(\mathbf{y}|\mathbf{X}) = \sum_{i=1}^{n} \frac{1}{2} \log(2\pi\sigma^2) + \frac{1}{2\sigma^2}\left(y^{(i)} - \mathbf{w}^\top \mathbf{x}^{(i)} - b\right)^2. \tag{3.1.15}$$

Uncertainty in both X, Y —> PCA

$$y = \mathbf{w}^T \hat{x} + b + \epsilon$$

$$\epsilon \sim \mathcal{N}(0, \sigma_y^2) \quad \equiv \quad \hat{y} \sim \mathcal{N}(y, \sigma_y^2)$$

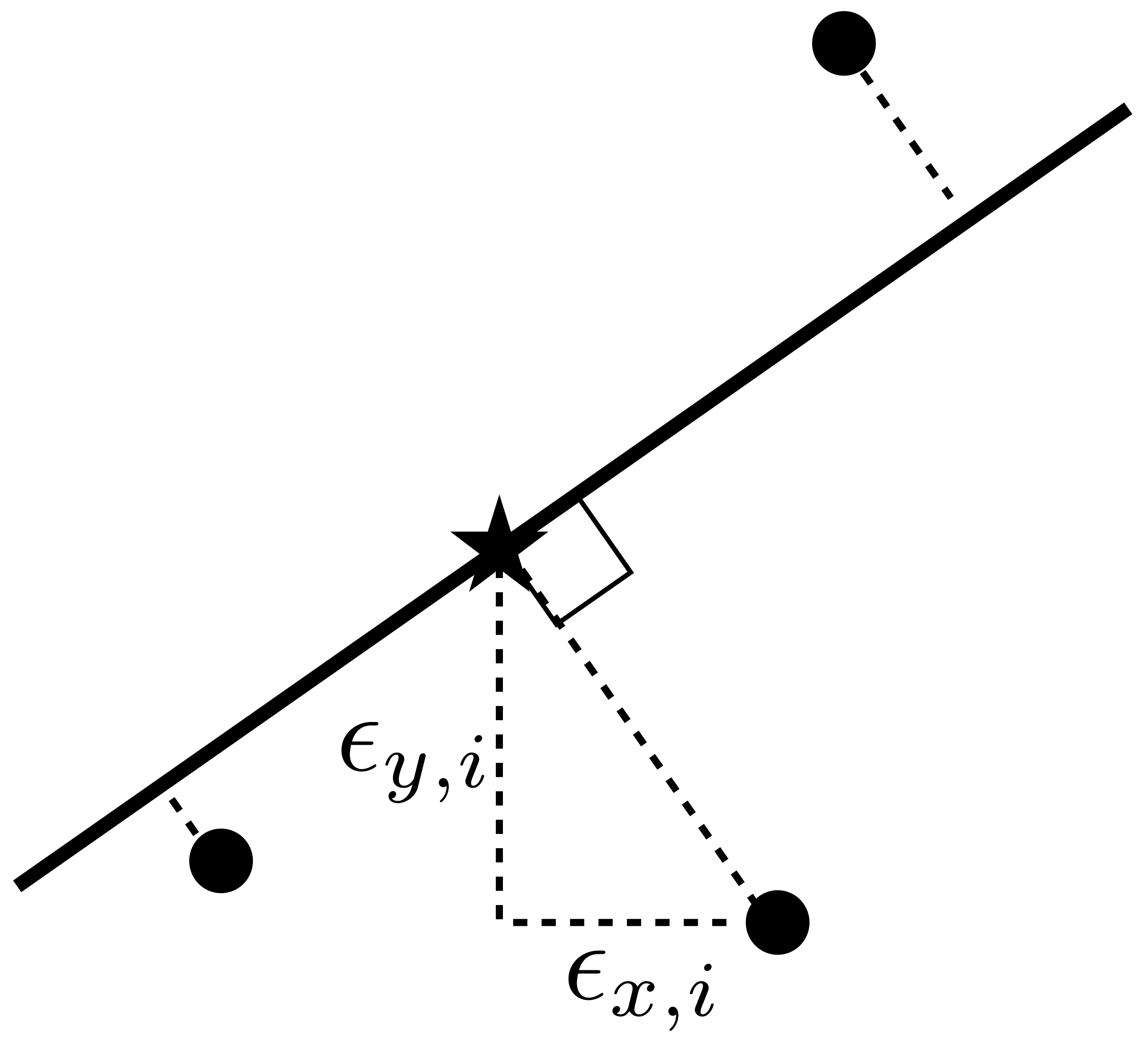
$$\hat{x} \sim \mathcal{N}(x, \sigma_x^2)$$



$$p(Y, X) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{\epsilon_{y,i}^2}{2\sigma_y^2}\right) \frac{1}{\sqrt{2\pi\sigma_x^2}} \exp\left(-\frac{\epsilon_{x,i}^2}{2\sigma_x^2}\right)$$

$$-logL = \sum_{i=1}^{N} \frac{1}{2} \log\left(2\pi\sigma_y^2\right) + \frac{1}{2} log(2\pi\sigma_x^2) + \frac{\epsilon_{y,i}^2}{2\sigma_y^2} + \frac{\epsilon_{x,i}^2}{2\sigma_x^2}$$

$$\sigma_y = \sigma_x \rightarrow$$



(This is actually the
assumption when doing PCA)

Maximum likelihood formulation of regression

- Log-Likelihood maximisation: more flexible, e.g., include a regularisation term (e.g. sparsity with Lasso regression) on the linear transformation matrix W, or use with other (non-Gaussian) noise models as in generalised linear models (glm)
- Can be used with non-isotropic variance i.e., different priors for the error distribution (error covariance matrix instead of identity, etc.) over the data dimensions. Good for multi-modal data where each modality comes from a different distribution and noise!
- Numerical EM optimisation

Exercises:

a) Can you express the Optimal Transport loss function (Session 3 Dynamical Inference) to a maximum likelihood estimation? For simplicity only consider the drift and entropy terms (with mass conservation assumption).

b) Can any ERM problem be translated to a maximum likelihood estimation?

Log-likelihood maximisation in presence of incomplete data (or hidden variables)

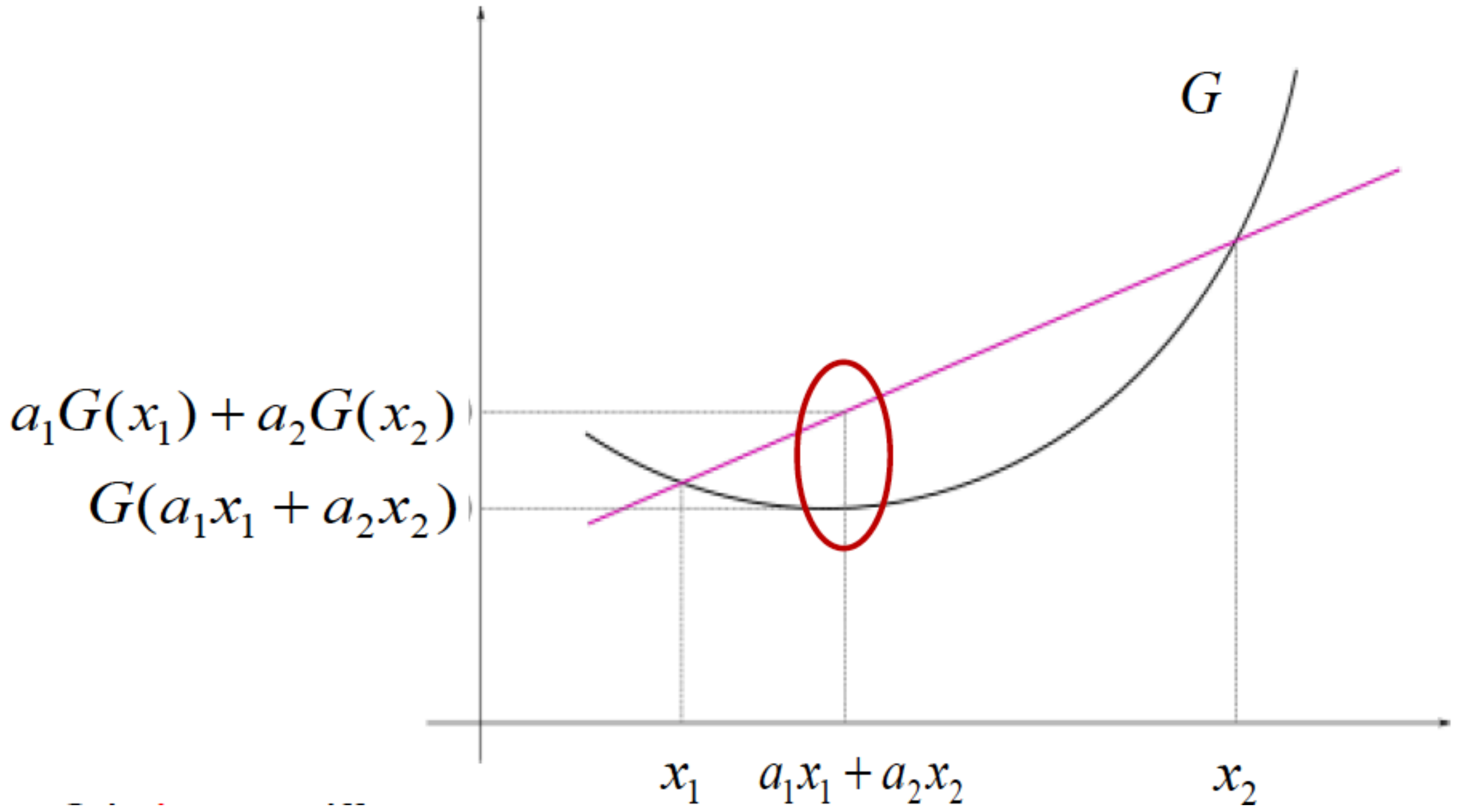
- y_i probabilities are not independent like x_i's, but sum up to 1 —> summation inside the \log
- Computationally intractable summation inside the \log

$$
\begin{aligned}
L(\boldsymbol{\theta}) &= \log \prod_{i=1}^{m} \mathcal{P}_{\boldsymbol{\theta}}[X = \mathbf{x}_i] \\
&= \sum_{i=1}^{m} \log \mathcal{P}_{\boldsymbol{\theta}}[X = \mathbf{x}_i] \\
&= \sum_{i=1}^{m} \log \left(\sum_{y=1}^{k} \mathcal{P}_{\boldsymbol{\theta}}[X = \mathbf{x}_i, Y = y] \right)
\end{aligned}
$$

$$
\underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, L(\boldsymbol{\theta}) = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_{i=1}^{m} \log \left(\sum_{y=1}^{k} \mathcal{P}_{\boldsymbol{\theta}}[X = \mathbf{x}_i, Y = y] \right)
$$

- Use Jensen inequality and convexness of the -log function to change the maximisation to a tractable lower bound maximisation problem

Jensen's inequality for convex functions



$$G(a_1x_1+a_2x_2)\le a_1G(x_1)+a_2G(x_2)$$

$$G\left(\sum_i a_i x_i\right)\le \sum_i a_i G(x_i) \qquad a_i \ge 0\ ,\ \sum_i a_i = 1$$

Expectation Maximisation (EM) algorithm

- Matrix Q [m,k] : probability that data point i \in 1:m assumes latent variable y in 1:k

$$L(\theta) \geq \; F(Q, \boldsymbol{\theta}) = \sum_{i=1}^{m} \sum_{y=1}^{k} Q_{i,y} \log \left(\mathcal{P}_{\boldsymbol{\theta}}[X = \mathbf{x}_i, Y = y] \right)$$

- If Q was known we can find the optimal \theta (Maximisation step)

$$\operatorname*{argmax}_{\boldsymbol{\theta}} F(Q, \boldsymbol{\theta})$$

$$\boldsymbol{\theta}^{(t+1)} = \operatorname*{argmax}_{\boldsymbol{\theta}} F(Q^{(t+1)}, \boldsymbol{\theta})$$

- If \theta was known, we could estimate Q (Expectation step)

$$Q_{i,y}^{(t+1)} = \mathcal{P}_{\boldsymbol{\theta}^{(t)}}[Y = y | X = \mathbf{x}_i]$$

- Initialise with random \theta_0 and Q_0
- Iterate over step 1-2 until convergence

EM as an alternating maximisation problem

$$G(Q, \boldsymbol{\theta}) = F(Q, \boldsymbol{\theta}) - \sum_{i=1}^{m} \sum_{y=1}^{k} Q_{i,y} \log(Q_{i,y})$$

$$\mathbb{Q} = \left\{ Q \in [0,1]^{m,k} : \forall i, \ \sum_{y=1}^{k} Q_{i,y} = 1 \right\}$$

LEMMA 24.2 *The EM procedure can be rewritten as:*

$$\begin{aligned} Q^{(t+1)} &= \operatorname*{argmax}_{Q \in \mathbb{Q}} G(Q, \boldsymbol{\theta}^{(t)}) \\ \boldsymbol{\theta}^{(t+1)} &= \operatorname*{argmax}_{\boldsymbol{\theta}} G(Q^{(t+1)}, \boldsymbol{\theta}) \ . \end{aligned}$$

Furthermore, $G(Q^{(t+1)}, \boldsymbol{\theta}^{(t)}) = L(\boldsymbol{\theta}^{(t)})$.

THEOREM 24.3 *The EM procedure never decreases the log-likelihood; namely, for all t,*

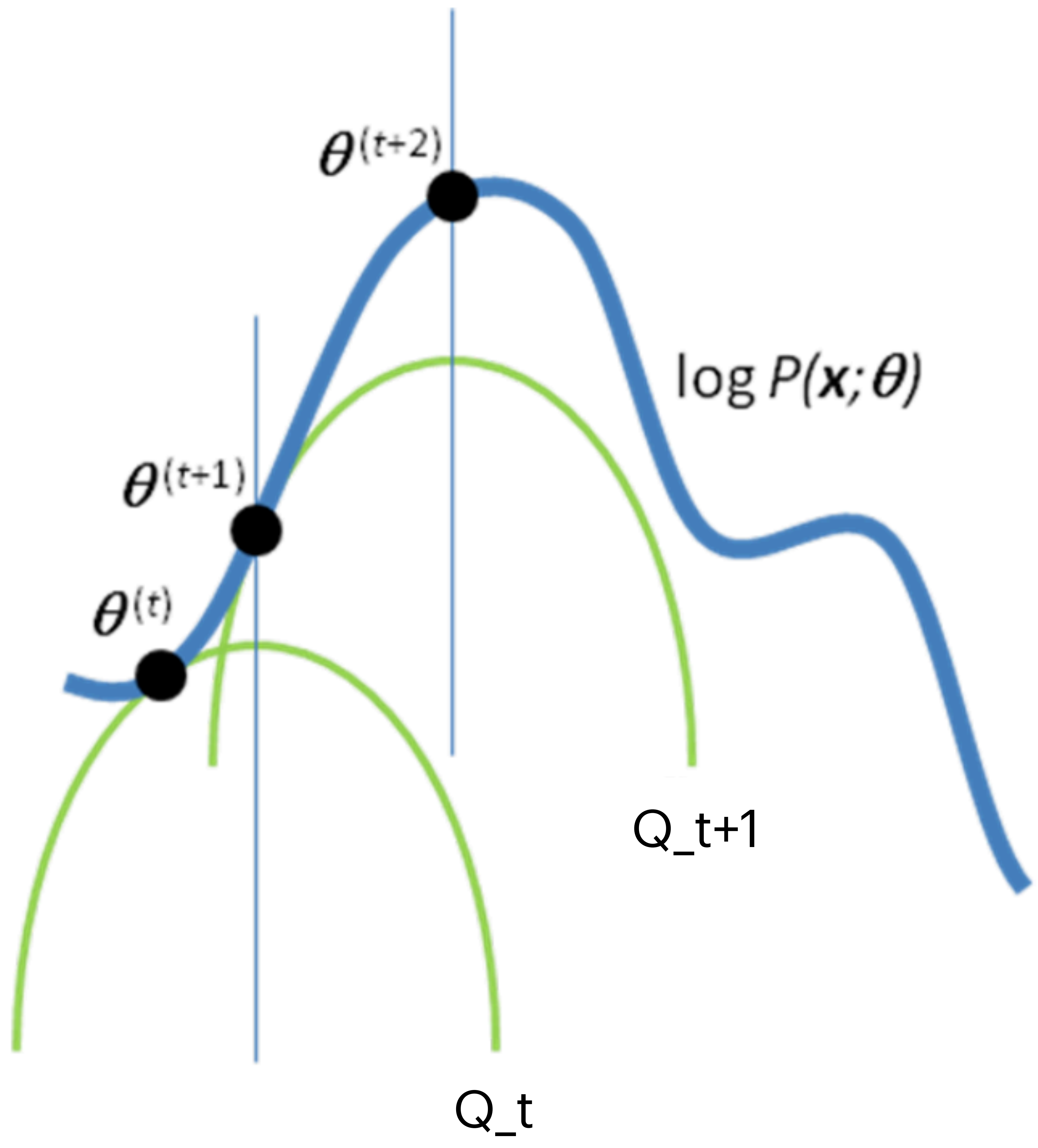
$$L(\boldsymbol{\theta}^{(t+1)}) \geq L(\boldsymbol{\theta}^{(t)}).$$

Proof By the lemma we have

$$L(\boldsymbol{\theta}^{(t+1)}) = G(Q^{(t+2)}, \boldsymbol{\theta}^{(t+1)}) \geq G(Q^{(t+1)}, \boldsymbol{\theta}^{(t)}) = L(\boldsymbol{\theta}^{(t)}).$$

EM summary

- Jensen inequality —> EM as a lower bound maximisation
- EM never decreases the log-likelihood
- Finds the local maximum, no guarantee for global solution or uniqueness
- EM can be used to learn a model (predictor) as well as hidden parameters from incomplete data (Generative)



EM, generative and variational models in single-cell omics

- (Non-negative) Matrix Factorisation
- Variational models, probabilistic PCA (on data with missing values)
- Variational auto-encoders
- Unlike PCA, in matrix factorization there is no hard constraint of orthonormality on the inferred factors, although the assumed distribution in embedding space encourages independence of factors

$$x \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

Probabilistic PCA

(Our new PCA) notation according to
Tipping M. E. & Bishop M., 1999

$$\mathbf{t} = \mathbf{W}\mathbf{x} + \boldsymbol{\mu} + \boldsymbol{\epsilon}$$

$$x \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

$$\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^{\mathbf{2}}\mathbf{I})$$

$$t \sim \mathcal{N}(\mu, \mathbf{W}\mathbf{W}^{\mathbf{T}} + \sigma^{\mathbf{2}}\mathbf{I})$$

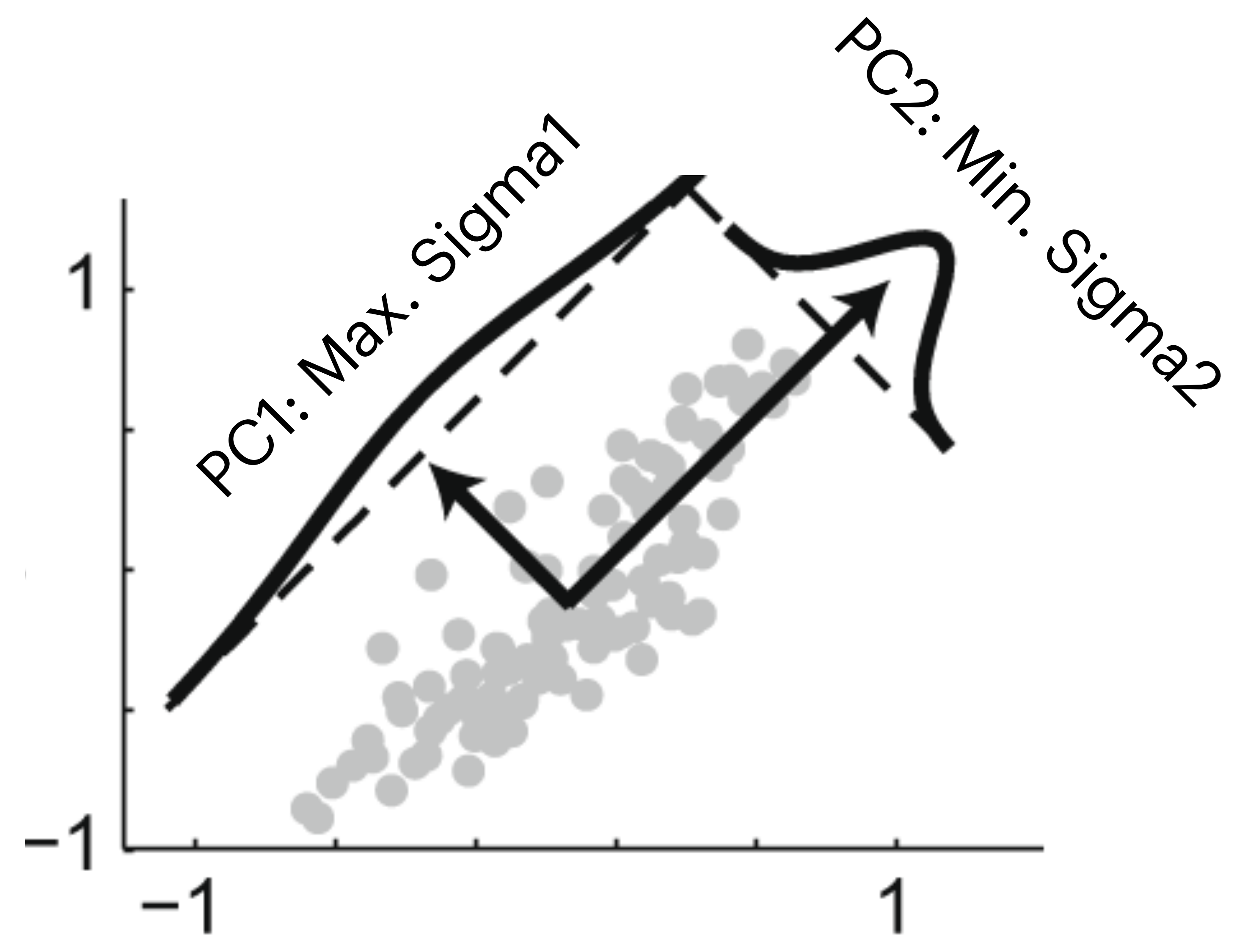
Embedding x Data t

$$\mathbf{x} = \mathbf{W}^T\mathbf{t}$$

$$\operatorname{argmax}_W(\sum_k W^T \underbrace{\mathbf{t}\mathbf{t}^T}_{\mathbf{x}\mathbf{x}^T} W) = \operatorname{argmax}_W(\operatorname{tr}(W^T\mathbf{t}\mathbf{t}^T W))$$

$$\operatorname{argmax}_W(\underbrace{\sum_{k=1}^{K}\sigma_k^2}_{\text{Explained variance}}) \equiv \operatorname{argmin}_W(\underbrace{\sum_{k=K+1}^{d}\sigma_k^2}_{\text{Sum of errors\^{}2}})$$

Because: $\sigma^2 = \sigma_1^2 + \sigma_2^2 + ... + \sigma_d^2$



Probabilistic PCA, EM algorithm

Tipping M. E. & Bishop M. J. R. Statist. Soc. B (1999)

Log-Likelihood:

$$\mathcal{L}_C = \sum_{n=1}^{N} \ln\{p(\mathbf{t}_n, \mathbf{x}_n)\}, \tag{22}$$

where, in PPCA, from the definitions in Section 3.1,

$$p(\mathbf{t}_n, \mathbf{x}_n) = (2\pi\sigma^2)^{-d/2} \exp\left(-\frac{\|\mathbf{t}_n - \mathbf{W}\mathbf{x}_n - \boldsymbol{\mu}\|^2}{2\sigma^2}\right)(2\pi)^{-q/2} \exp\left(-\frac{\|\mathbf{x}_n\|^2}{2}\right). \tag{23}$$

In the E-step, we take the expectation of $\mathcal{L}_C$ with respect to the distributions $p(\mathbf{x}_n|\mathbf{t}_n, \mathbf{W}, \sigma^2)$:

$$\langle\mathcal{L}_C\rangle = -\sum_{n=1}^{N}\left\{\frac{d}{2}\ln(\sigma^2) + \frac{1}{2}\operatorname{tr}(\langle\mathbf{x}_n\mathbf{x}_n^{\mathrm{T}}\rangle) + \frac{1}{2\sigma^2}(\mathbf{t}_n - \boldsymbol{\mu})^{\mathrm{T}}(\mathbf{t}_n - \boldsymbol{\mu}) - \frac{1}{\sigma^2}\langle\mathbf{x}_n\rangle^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}(\mathbf{t}_n - \boldsymbol{\mu}) + \frac{1}{2\sigma^2}\operatorname{tr}(\mathbf{W}^{\mathrm{T}}\mathbf{W}\langle\mathbf{x}_n\mathbf{x}_n^{\mathrm{T}}\rangle)\right\}, \tag{24}$$

Use the data t plus random initialisation for W and sigma i.e., p(x|t,W,sigma2) to calculate the **optimal x** given by:

$$\langle\mathbf{x}_n\rangle = \mathbf{M}^{-1}\mathbf{W}^{\mathrm{T}}(\mathbf{t}_n - \boldsymbol{\mu}), \qquad \mathbf{M} = \mathbf{W}^{\mathrm{T}}\mathbf{W} + \sigma^2\mathbf{I}$$

$$\langle\mathbf{x}_n\mathbf{x}_n^{\mathrm{T}}\rangle = \sigma^2\mathbf{M}^{-1} + \langle\mathbf{x}_n\rangle\langle\mathbf{x}_n\rangle^{\mathrm{T}},$$

…Probabilistic PCA, EM algorithm

Use the estimated x to **update W and sigma** (and missing values in t)

In the M-step, $\langle \mathcal{L}_C \rangle$ is maximized with respect to $\mathbf{W}$ and σ^2 giving new parameter estimates

$$\widetilde{\mathbf{W}} = \left\{ \sum_{n=1}^{N} (\mathbf{t}_n - \boldsymbol{\mu}) \langle \mathbf{x}_n \rangle^{\mathrm{T}} \right\} \left(\sum_{n=1}^{N} \langle \mathbf{x}_n \mathbf{x}_n^{\mathrm{T}} \rangle \right)^{-1}, \tag{27}$$

$$\tilde{\sigma}^2 = \frac{1}{Nd} \sum_{n=1}^{N} \{ \|\mathbf{t}_n - \boldsymbol{\mu}\|^2 - 2\langle \mathbf{x}_n \rangle^{\mathrm{T}} \widetilde{\mathbf{W}}^{\mathrm{T}} (\mathbf{t}_n - \boldsymbol{\mu}) + \mathrm{tr}(\langle \mathbf{x}_n \mathbf{x}_n^{\mathrm{T}} \rangle \widetilde{\mathbf{W}}^{\mathrm{T}} \widetilde{\mathbf{W}}) \}. \tag{28}$$

To maximize the likelihood then, the sufficient statistics of the conditional distributions are calculated from equations (25) and (26), after which revised estimates for the parameters are obtained from equations (27) and (28). These four equations are iterated in sequence until the algorithm is judged to have converged.

Exercises:

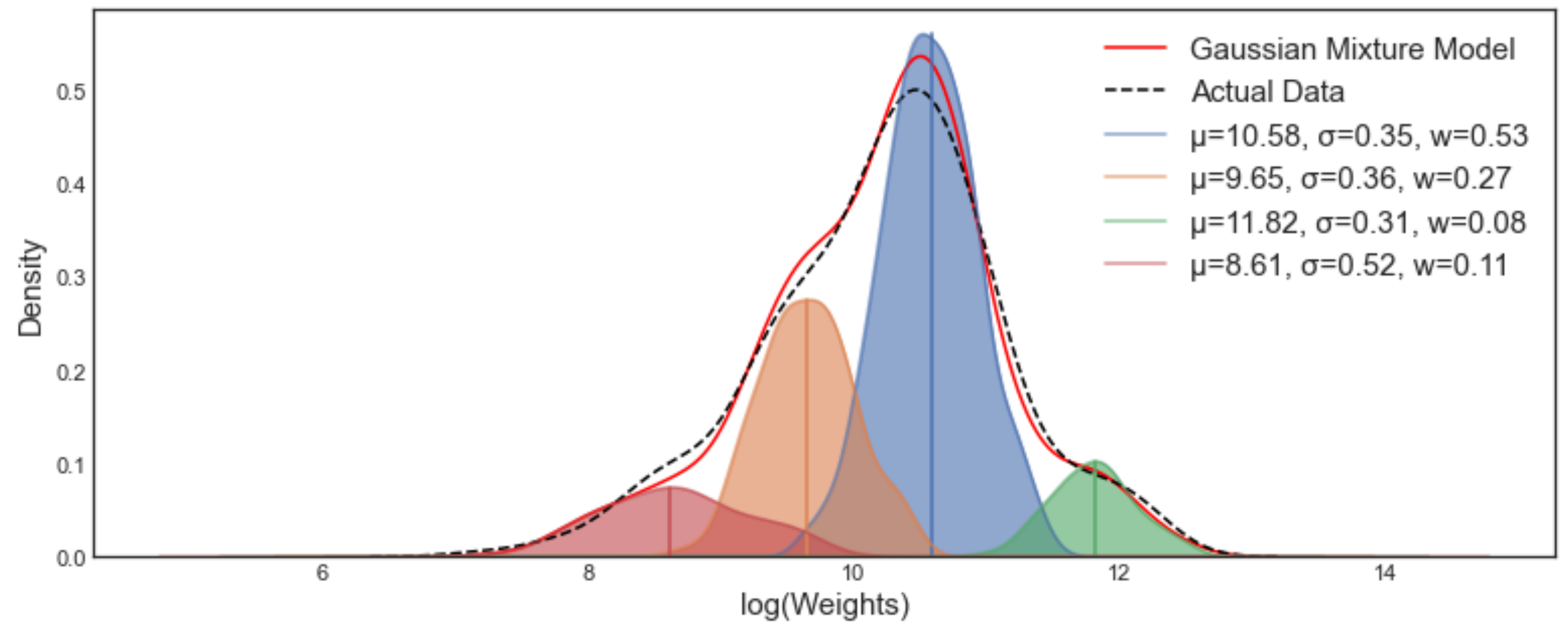
a) Apply a Probabilistic PCA package in R or Python on a dataset and compare the embedding space with standard PCA

b) What tuneable parameters does the PPCA function have?

c) Randomly delete 20% of your data matrix enters and apply PPCA on it again. Compared with the full data embedding

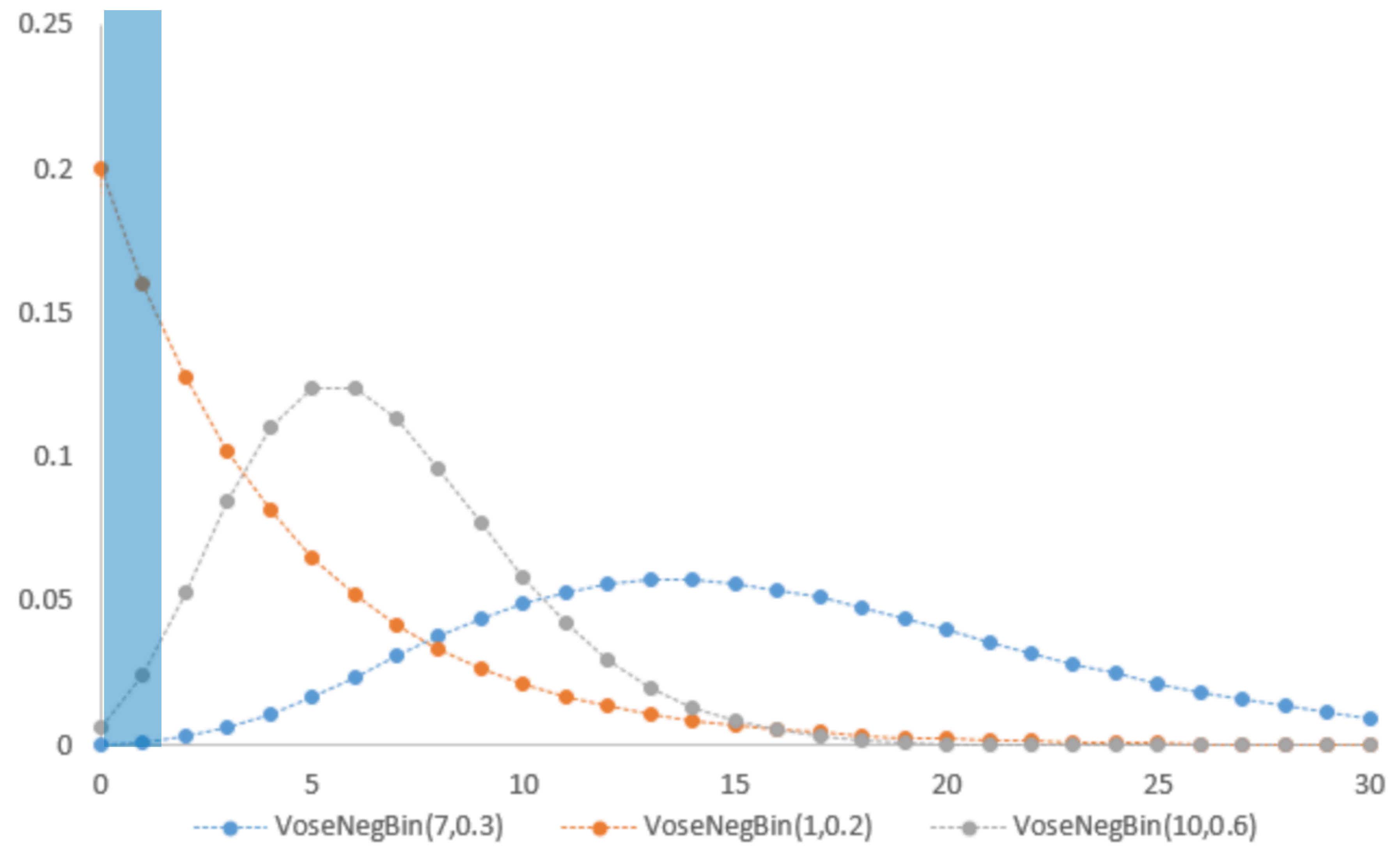
Zero-inflated NB (ZINB) model for heterogeneous cell populations

- Mechanistic Poisson and Gamma-Poisson (~NB) process (gene expression x sampling)
- Gaussian mixture —> multi-modal $\sigma^2 = \mu$
- Poisson mixture —> NB $\sigma^2 = \mu + \frac{\mu^2}{\phi}$
- NB mixture —> NB



- How about including cell populations which completely do not express that gene? —> Zero inflated model is necessary:

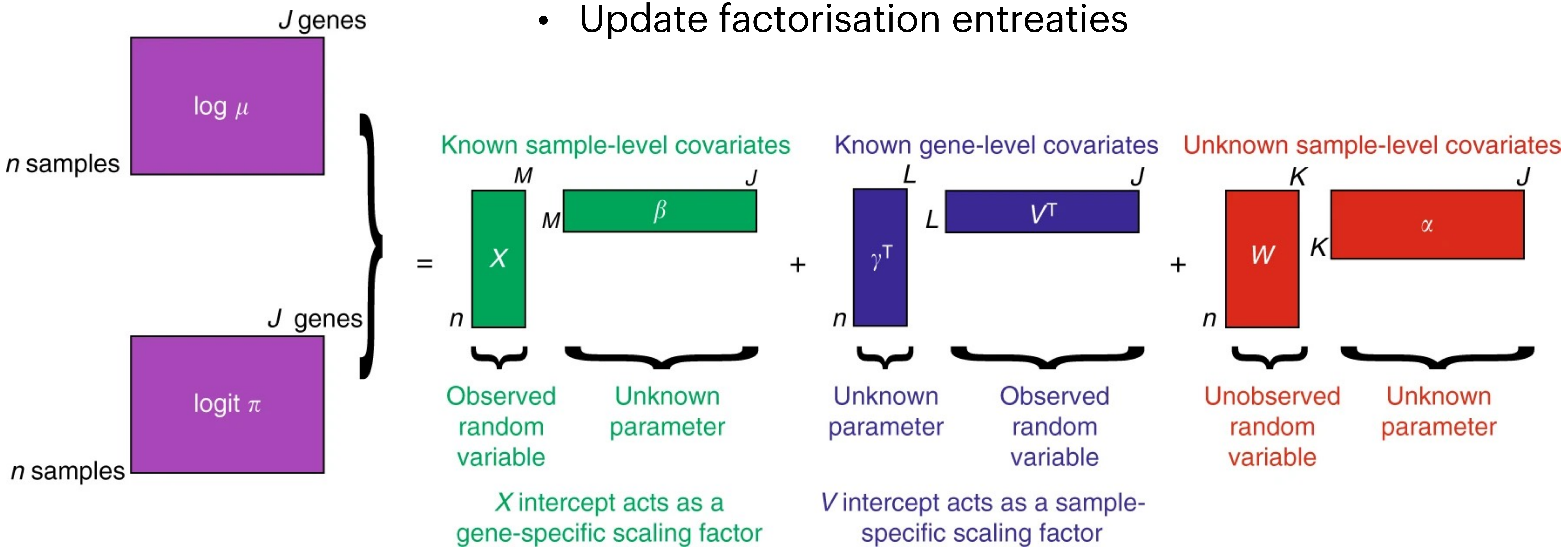
$$\Pr(x = x_0) = (1 - \pi)\delta(x_0) + \pi \ \mathrm{NB}(x_0|\mu, \phi)$$



ZINB-WaVE matrix factorisation into known and unknown components

Risso, D., Perraudeau, F., Gribkova, S. *et al.* A general and flexible method for signal extraction from single-cell RNA-seq data. *Nat Commun* (2018).

- Partly observed factors
- Update gene expression parameters \pi, \mu, \phi
- Update factorisation entreaties


J genes
log μ
n samples
J genes
logit π
n samples
=
Known sample-level covariates
M
J
M
β
X
n
Observed random variable
Unknown parameter
X intercept acts as a gene-specific scaling factor
+
Known gene-level covariates
L
J
L
V^T
γ^T
n
Unknown parameter
Observed random variable
V intercept acts as a sample-specific scaling factor
+
Unknown sample-level covariates
K
J
K
α
W
n
Unobserved random variable
Unknown parameter

Thank you for your attention!