Variational Autoencoders course

1 Generative models

- A model for the probability distribution of a data x (random variable?)
- generative model : A parametric model : $p(x, \theta)$ that "fits" the distribution of $x \in X$
- our job is to estimate θ given $x \in X$.
- eg: **given you have a die** which can be parameterized by a multinomial distribution which is a generative model
- intuitively, ideally, θ assigns probability based on how likely it is to actually sample elements x from the source distribution X.

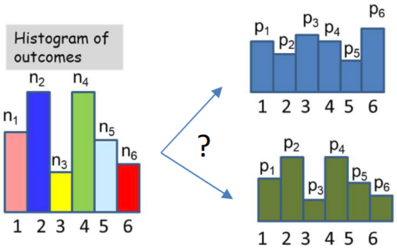
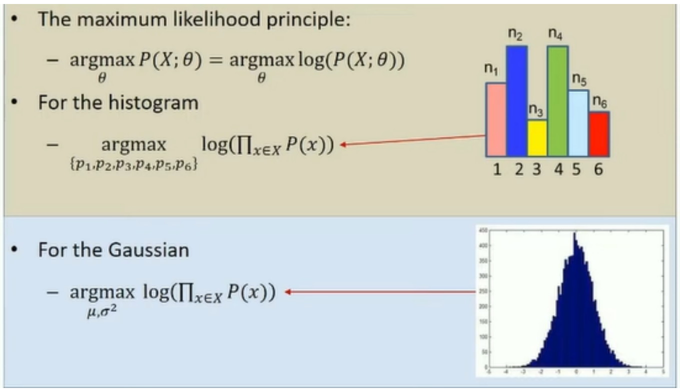


Figure 1: choosing appropriate $p(x; \theta)$ that frequency of samples drawn

- assumptions while making assigning probability distributions in this way, (choosing distributions that match sampling frequency history)
  - **the world is a boring place principle**: distribution of sampled set is typical. "odds of seeing something unusual is very low". amounts to choosing a model that **maximizes the liklihood**

2 Maximum Liklihood Estimation

- as show in Figure 2. we choose the parameter θ that maximizes the probability of the generated distribution of samples. which is finding the (product)joint probability over each sample's assigned probability (by the model). (alternatively log, because it is monotonic) (parameters are data dependent)



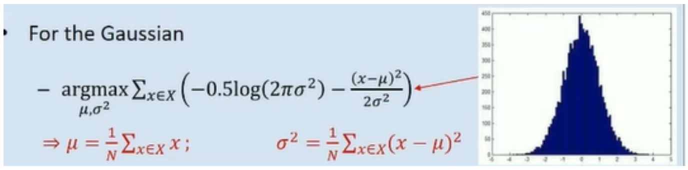
- The maximum likelihood principle:
  - $\underset{\theta}{\text{argmax}}\, P(X;\theta) = \underset{\theta}{\text{argmax}} \log(P(X;\theta))$
- For the histogram
  - $\underset{\{p_1,p_2,p_3,p_4,p_5,p_6\}}{\text{argmax}} \log(\prod_{x \in X} P(x))$
- For the Gaussian
  - $\underset{\mu,\sigma^2}{\text{argmax}} \log(\prod_{x \in X} P(x))$

Figure 2: Maximum likelihood — 10:00 Vaes part 1

- Notice how higher probabilities/(density) assigned to more frequent occurrences of a sample type/(region) maximizes the **Maximum Likelihood**
- using Log probabilities :

$$\underset{\theta}{argmax} \prod_{x \,\in\, X} p(x;\theta)$$

$$\Rightarrow \underset{\theta}{argmax} \sum_{x \,\in\, X} \log P(x; \theta`)$$



For the Gaussian

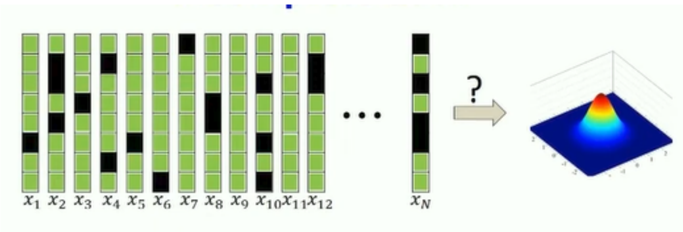
– $\underset{\mu,\sigma^2}{\text{argmax}} \sum_{x\in X} \left(-0.5\log(2\pi\sigma^2) - \frac{(x-\mu)^2}{2\sigma^2}\right)$

$\Rightarrow \mu = \frac{1}{N}\sum_{x\in X} x \,;$ $\quad \sigma^2 = \frac{1}{N}\sum_{x\in X}(x-\mu)^2$

Figure 3: MLE for Gaussians with $\log p(x;\theta)$ — 10:39 Vaes part 1

- An example for gaussian parameter estimation using MLE is shown above in Figure 3. turns out means maximize MLEs for Gaussians.

- Sometimes the data provided may be incomplete, and parameters cannot be estimated directly using MLE.
- types of missing data:
  - **data has missing components**:



x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 x_{10} x_{11} x_{12}
x_N
?

Figure 4: estimating gaussian data where the samples have missing components — 15:20 vaes part 1

    * in each data instance some components are missing like shown in Figure 3.
    * in the lecture, at 15:20, sir talks about a real-life example from that happens to netflix.
    * the world is a boring place assumption states that only the observed data (non-missing components) is modellable using MLE.
    * solution : **MARGINALIZING OUT THE MISSING COMPONENTS** : MLE on observed data along can be written as follows :

$$\underset{x\,\in\,X}{argmax}\; log(P(O)) = \underset{\mu\;,\;\sigma^2}{argmax} \sum_{o\in O} \log \int_{-\infty}^{\infty} P(o,m)dm$$

    * **breaking down the equation**
      - for a given $o \in O$: $log \int_{-\infty}^{\infty} P(o,m)dm$: you are looking at a given column (a single sample in Figure 4.). We are trying to estimate $p(o;\theta)$ $o \in O$ when some components are missing m : $log \int_{-\infty}^{\infty} P(o,m)dm$ does that. ($\sum$ of all joints $(P(o,m;\theta))$ involving the undesired random (m) variable throws it out $P(o;\theta)$)
      - The rest is the same as regular MLE.
    * To sum up : when you cannot use $p(x)$ directly in MLE because components are missing $o \cup m = x$. You have to consider only observed. you find and use $p(o)$ by marginalizing m out as shown above.
    * stud q : if you have a loot of missing components, you would be integrating over a lot of variables which will make $p(o) = 1$, making $log(p(o)) = 0$ posssibly bad estimate.
    * question : everything in negative? $log(x)$: $x \in (0,1)$ is -ve. : dw, it works, you are maximizing a negative quantity, 0 is better than any negative quantity.
    * Bad problem to solve (log of an integral) So what to do???

  - **Missing information about the model**

    - Sometimes the distribution is complex, you may not have information like dimensionality, complexity(order), etc.
    - An example of this sort that was discussed is **Gaussian mixture models** where you have to estimate both i) μs and σs of each gaussian, and the mixture proportions (weights of each gaussian) 24:18 Vaes part 1.
    - for background: the generation process consists of choosing a gaussian with $p(k)$ for a gaussian k, and drawing a sample from that gaussian.
    - problems arise when you can sample, but **you don't know which gaussian you sampled from.**
    - consider gaussian indices k, and samples (observations) o problem becomes:

      $$\underset{(\mu_k,\sigma_k^2)\forall k}{argmax} \sum_{o \in O} log(P(o))$$

      marginalizing k out :

      $$p(o) = \sum_k P(k, \mathcal{N}_k) \qquad (\text{possible abuse of } \mathcal{N}_k)$$

      $$p(O) = \underset{(\mu_k,\sigma_k^2)\forall k}{argmax} \sum_{o \in O} log \sum_k p(k)\mathcal{N}(o; \mu_k, \sigma_k^2)$$

    - notice how : **When doing MLE, if you have any sort of missing data, that you will need to marginalize out, you usually end up with log(integral) Bad for some reason, "defies direct optimization (ASK 5550)".**

  - Formalizing the problem generally: When you try to estimate parameters of a model given some missing data $m \cup u = x$. You have to use only the present/observed [?]data forcing you to **Marginalize** out m. This problem results in the below situation:

    $$\hat{\theta} = \underset{\theta}{argmax} \sum_o log \int_{-\infty}^{\infty} p(m, o; \theta)$$

  - $log \int_{-\infty}^{\infty} p(m, o; \theta)$ **IS BAD FOR OPTIMZATION** why?
  - **SIDE QUESTION 1**
  - prof says gradient descent can be used for this problem

3 Expectation Maximization

- Objective is to get a more tractable way to solve:

  $$\underset{m}{logp}(m, o, \theta)$$

- approach : **Solve a tractable approximation**
- Enter : "Auxilliary function" - guaranteed lower bound on the log likelihood, which **touches the target** function at some desired point θ_i

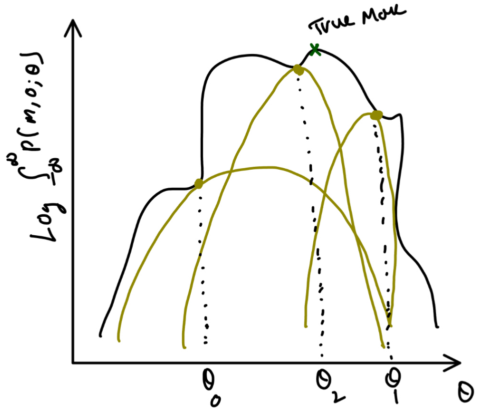


Figure 5: Auxilliary function (yellow) initialization to touch the target function at different points $\theta_0, \theta_1, \theta_2$

- Defining such a flexible lower bound allows for Hill Climbing Solutions.
- **Hill Climbing solution (as in Figure 6):**

  1. Let target function be T. Initialize an auxiliary function A_0 touching some point T_{max_i} with parameter θ_0
  2. find maximum of A_0, (point A_{max_0}) and not corresponding θ (θ_1).
  3. Choose A_2 such that it touches T at θ_1. We know that $T_{max_1} > A_{max_0} > T_{max_0}$.
  4. repeat steps 2 and 3
  5. convergence will happen what $A_{max_i} = T_{max_i}$

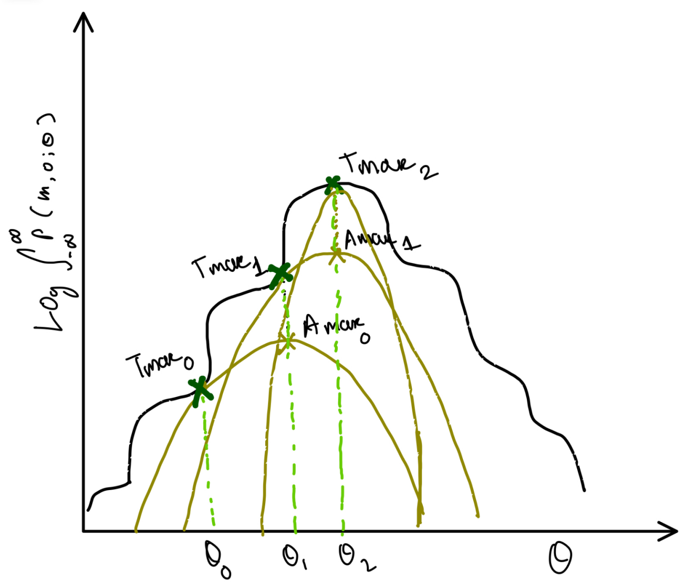


Figure 6: Hill climbing solution for the log integral using the auxiliary .

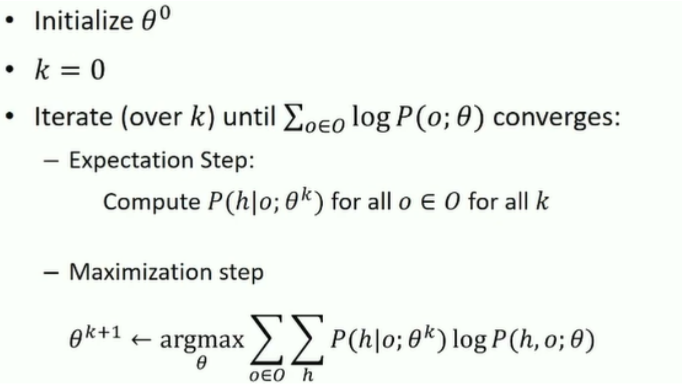
- This kind of Auxilliary function is called an **Emperical Lower Bound (ELBO) AKA VARIATIONAL LOWER BOUND**
- To recall the target function that is intractable is :

$$p(O;\theta) = \sum_{o \in O} log \int_{-\infty}^{\infty} p(m, o, \theta)$$

- The Auxilliary function, or the Emperical Lower Bound (ELBO) can be derived to be:

$$J(\theta,\bar{\theta}) = \sum_{o\in O}\sum_{m} p(m|o;\bar{\theta})logP(m,o,\theta) - \sum_{o\in O}\sum_{m} p(m|o;\bar{\theta})logP(m|o;\bar{\theta})$$

- **SIDE QUESTION 2**
- **Breaking down the equation**
  - "**Aposteriori** probability of the missing variables given observed" : $\sum_{o\in O}\sum_{m} p(m|o;\bar{\theta})logP(m|o;\bar{\theta})$
  - "**log likelihood of the joint** probability of missing and observed variables". $\sum_{o\in O}\sum_{m} p(m|o;\bar{\theta})logP(m,o,\theta)$
  - how is the above term log likelihood? shouldn't it be the same term inside and outside log?, what are you missing?
- **Formal expectation maximization algorithm UNCLEAR, GO THROUGH IT PROPERLY AGAIN —– 38:00 - vaes part 1, and read f23 slides properly, everything is there in detail**



- Initialize θ^0
- $k = 0$
- Iterate (over k) until $\sum_{o\in O} \log P(o;\theta)$ converges:
  - Expectation Step:

    Compute $P(h|o;\theta^k)$ for all $o \in O$ for all k
  - Maximization step

    $\theta^{k+1} \leftarrow \underset{\theta}{\text{argmax}} \sum_{o\in O}\sum_{h} P(h|o;\theta^k)\log P(h,o;\theta)$

Figure 7: Formal EM

- θ^k is $\bar{\theta}$: somehow the target function should evaluate to the ELBO at each θ^k (As the ELBO supposedly "touches" the target at θ^k) **check that**.
- **(verify each point)** Background to understand this : **refer to 34:00 vaes part 1 for a toy example**:

  1. Having theta, fully characterizes a distribution. you can compute, joints, and conditionals
  2. you **fill in missing data** using the distribution characterized by current θ
  3. You update you θ based on what θ should ideally be.
  4. you repeat steps 2 and 3 til convergence when updates don't result in changes.

- Problem: datapoints x are draw from a continuous distribution, the ELBO boils down from :

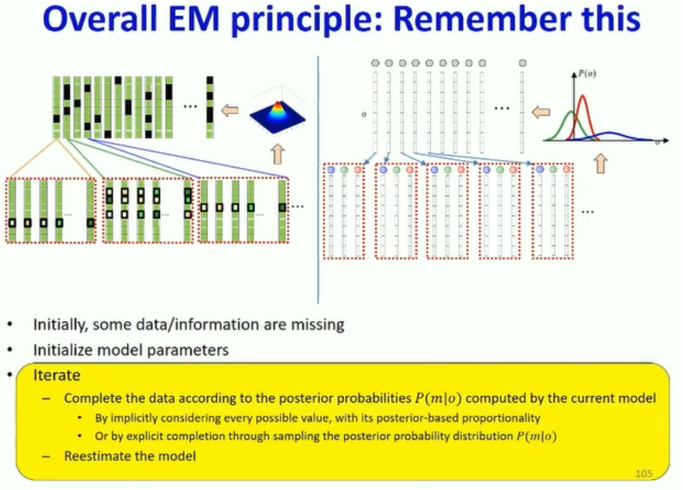
$$J(\theta,\bar{\theta}) = \sum_{o\in O}\sum_{m} p(m|o;\bar{\theta})logP(m,o,\theta) - \sum_{o\in O}\sum_{m} p(m|o;\bar{\theta})logP(m|o;\bar{\theta}) \qquad \text{ELBO}$$

to

$$J(\theta,\bar{\theta}) = \sum_{o\in O}\int_{-\infty}^{\infty} p(m|o;\bar{\theta})logP(m,o,\theta) - \sum_{o\in O}\int_{-\infty}^{\infty} p(m|o;\bar{\theta})logP(m|o;\bar{\theta}) \qquad \text{ELBO}$$

- Solution : to make up for large sums / integrals, you can you the values from a sample set.

  item : is this provably convergent ? "when you have the complete data, the estimate is a Maximum Likelihood estimate (increasing likelihood of the data). When you are sampling from the aposteriori estimate, you choose samples that increases the likelihood of the data (likelihood monotonically increases)"



Overall EM principle: Remember this

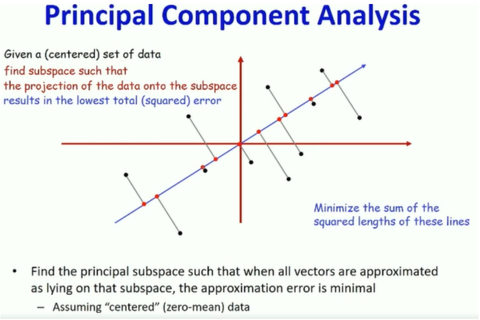
- Initially, some data/information are missing
- Initialize model parameters
- Iterate
  - Complete the data according to the posterior probabilities $P(m|o)$ computed by the current model
    - By implicitly considering every possible value, with its posterior-based proportionality
    - Or by explicit completion through sampling the posterior probability distribution $P(m|o)$
  - Reestimate the model

105

Figure 8: EM principal summary — 51:51 VAEs part 1

4 PCA (Principal Component Analysis)

- Finds a **Prinicpal Subspace** s.t. if all vectors are approximated as some scaling/point on this principal subspace that minimizes error.



Principal Component Analysis

Given a (centered) set of data
find subspace such that
the projection of the data onto the subspace
results in the lowest total (squared) error

Minimize the sum of the squared lengths of these lines

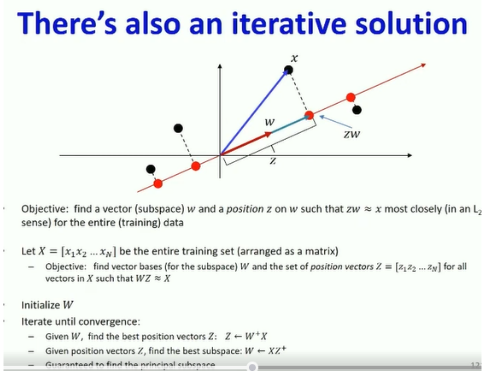
- Find the principal subspace such that when all vectors are approximated as lying on that subspace, the approximation error is minimal
  - Assuming "centered" (zero-mean) data

Figure 9: PCA background

- objective :

$$\underset{W}{Min}\ ||X - wz||^2$$

- breaking down the objective:
  - z are the coordinates (approximations) pf points on the subspace
  - x are the original datapoints.
  - W are the bases of the principal subspace



There's also an iterative solution

- Objective: find a vector (subspace) w and a *position* z on w such that $zw \approx x$ most closely (in an L_2 sense) for the entire (training) data
- Let $X = [x_1 x_2 \dots x_N]$ be the entire training set (arranged as a matrix)
  - Objective: find vector bases (for the subspace) W and the set of *position vectors* $Z = [z_1 z_2 \dots z_N]$ for all vectors in X such that $WZ \approx X$
- Initialize W
- Iterate until convergence:
  - Given W, find the best position vectors Z: $Z \leftarrow W^+ X$
  - Given position vectors Z, find the best subspace: $W \leftarrow XZ^+$
  - Guaranteed to find the principal subspace

Figure 10: Iterative solution to pca

- **PCA can be interpreted as an autoencoder!**

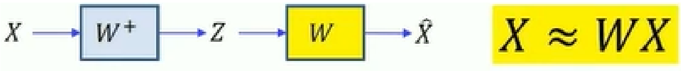


Figure 11: PCA as an autoencoder

- **W is not unique! as shown beloe**. To make it unique, we can impose condition that Z has 0 mean and unit variance

$$X = (W\ A)\ (A^{-1}\ Z) \qquad \text{for any A}$$

- More specifically, A space (here the principal subspace) can be spanned by different more than 1 unique set of basis vectors (here W). We simply (conveniently) choose the basis (W) with 0 mean and unit variance.

- EM for PCA:

  1. Initialize random W^+ (W^+ is a linear transform $W : x \to z$) (encoder)
  2. find z
  3. estimate W (W is a linear transform $W : z \to x$) (decoder)
  4. repeat steps 2, 3 till convergence

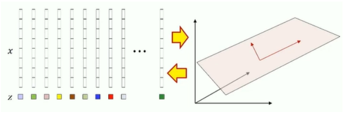
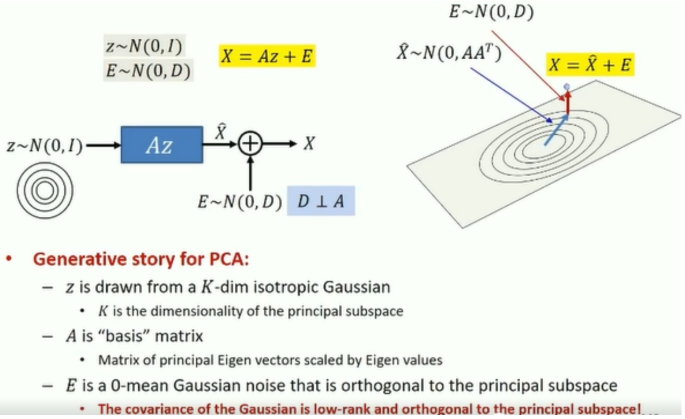


Figure 12: EM viz for pca

- EM is used to approximate probability distributions (generative models), so what distribution is being learned?
- the decoder W (yellow in Figure 12) is a generative model, for which z is a gaussian prior. (distributed with 0 mean and unit variance)

- **PCA generative story: The decoder makes you take a i) gaussian step along the plane (Az), and ii) a perpendicular step off the plane (error E)**



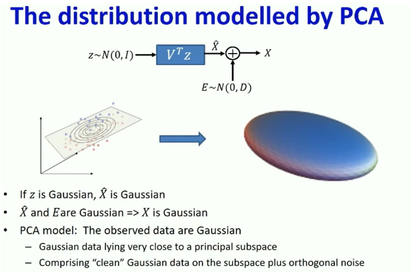
- **Generative story for PCA:**
  - z is drawn from a K-dim isotropic Gaussian
    - K is the dimensionality of the principal subspace
  - A is "basis" matrix
    - Matrix of principal Eigen vectors scaled by Eigen values
  - E is a 0-mean Gaussian noise that is orthogonal to the principal subspace
    - The covariance of the Gaussian is low-rank and orthogonal to the principal subspace!

Figure 13: Generative story of PCA

- $Az = \hat{X} \sim \mathcal{N}(\prime, \mathcal{A}\mathcal{A}^T)$, where $\hat{X}$ is the projections of result of taking the gaussian step on the principal subspace.

- Overarching Assumption of PCA : **The original data, in it's ambient space LIES CLOSE TO A LINEAR SUBSPACE ("close" as in has gaussian noise)**

The distribution modelled by PCA



- If z is Gaussian, $\hat{X}$ is Gaussian
- $\hat{X}$ and $E$are Gaussian => X is Gaussian
- PCA model: The observed data are Gaussian
  - Gaussian data lying very close to a principal subspace
  - Comprising "clean" Gaussian data on the subspace plus orthogonal noise

Figure 14: Assumption of the distribution of data in the ambient space.

- Taking a step back : **Is noise always orthogonal to data (principal subspace)? (NO!** justification in 1:19:20 vaes part 1)
- to recall, considering the generative view of PCA, it assumes noise is orthogonal to the principal subspace (parameterized by a gaussian whose covariance is Low rank and is orthogonal to W as mentioned in Figure 13.)

- An extension of PCA would be to parameterize this noise E in the second step of the decoding process using a $\mathcal{N}(0, D_{new})$, where D_{new} **IS FULL RANK!! no longer orthogonal to the principal subspace (W)** leading to **LINEAR GAUSSIAN MODELS**.

- Note, the assumption on D_{new} or D_{lg} (for linear gaussian). is that it is **UNCORRELATED (Diagonal Covariance Matrix)**.

END OF VAEs PART 1

---

5 Linear Gaussian Models

6 side questions

1. In MLE problems with missing data how do you calculate the joint $p(m, o; \theta)$?

ANSWER : Looking back at Figure 4. each sample is a column, and it has missing components m, observable components o, $(m \cup o = x)$. at an arbitrary time step i you will have an estimate for the parameters θ of the target generative model $P(x; \theta)$ or $P(m, o; \theta)$.

Having an estimate of θ fully characterizes your pre-optimized generative model, which you can sample from.

each sample will have **missing and observed** components, and if you sample some $n \to \infty$ samples, you can find the probability densities over these samples by looking at the frequency between ranges of the generated samples. **this gives you the JOINT DISTRIBUTION** $p(m, o; \theta)$

Once you have the joint distribution, if you wanna find conditionals $P(m|o; \theta)$, look at observations that have the same the values you want in the locations you want ("observed"). summing over these individual joints will let you marginalize out the "missing" variables, and will give you $p(o; \theta)$

2. How is the ELBO function of the generic MLE with missing data easier to solve than the original target function.

$$p(O; \theta) = \sum_{o \in O} log \int_{-\infty}^{\infty} p(m, o, \theta) \qquad \text{Target}$$

$$J(\theta, \bar{\theta}) = \sum_{o \in O} \sum_{m} p(m|o; \bar{\theta}) log P(m, o, \theta) - \sum_{o \in O} \sum_{m} p(m|o; \bar{\theta}) log P(m|o; \bar{\theta}) \qquad \text{ELBO}$$