Assorted Topics (4)

CS772A: Probabilistic Machine Learning

Piyush Rai

Plan today

- Calibration (contd)
- Bayesian nonparametric methods

Notion of Calibration for Classification Models

- Desirable: Predictions with confidence $\mu \in (0,1)$ are correct $(100 \times \mu)\%$ of the time
- Assume a model f that predicts softmax vector $f(x_n) = [a_{n1}, a_{n2}, \dots, a_{nC}]$ such that

Predicted label

$$\hat{y}_n = \text{argmax}_{c=\{1,2,\dots,C\}}\, a_{nc}$$

Probability of the predicted label (confidence of f for this prediction)

$$\hat{a}_n = \max_{c=\{1,2,\dots,C\}}\, a_{nc}$$

$$\text{acc}(B_b) = \frac{1}{|B_b|}\sum_{n \in B_b} \mathbb{I}(\hat{y}_n = y_n)$$

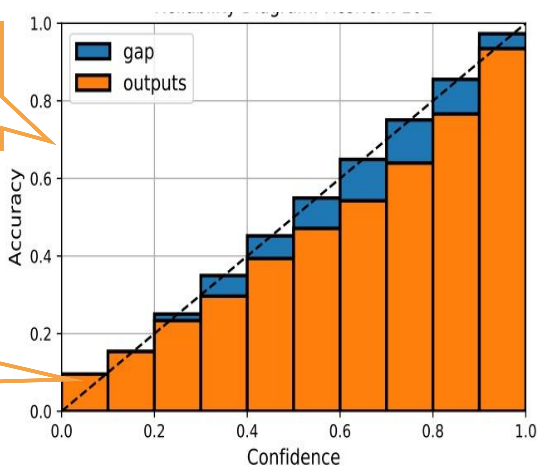
$$\text{conf}(B_b) = \frac{1}{|B_b|}\sum_{n \in B_b} \hat{a}_n$$

- Below is a typical plot of accuracy vs confidence of f on some validation set
  - To get the plot, we usually split the predictions into B bins

Using $B = 10$ equal-width bins

It's just one simple way; other ways also possible to construct the bins

Such plots are known as reliability diagrams



Should be small for a well-calibrated model

Expected Calibration Error

$$\text{ECE}(f) = \sum_{b=1}^{B} \frac{|\mathcal{B}_b|}{B} \left|\text{acc}(\mathcal{B}_b) - \text{conf}(\mathcal{B}_b)\right|$$

Model's accuracy on examples falling into the b^{th} bin

Model's confidence on these examples

Some Methods for Calibration

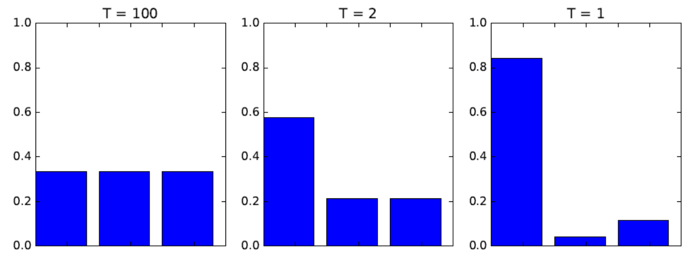
Parameters of the trained model are kept frozen in this process

- Method 1: Calibrate an already trained model in a post-hoc manner, e.g.,
  - Requires learning to scale the logits produced by the model, e.g.,

The scaling parameters are learned by minimizing the loss on some validation set

$$\text{softmax}(z_1, z_2, \ldots, z_C) \Longrightarrow \text{softmax}\left(w_1 z_1 + b_1, w_2 z_2 + b_2, \ldots, w_C z_C + b_c\right)$$

$$\text{softmax}(z_1, z_2, \ldots, z_C) \Longrightarrow \text{softmax}\left(\frac{z_1}{T}, \frac{z_2}{T}, \ldots, \frac{z_C}{T}\right)$$



- Method 2: Change the training procedure, e.g.,
  - Add a regularizer which avoids overconfident predictions

Maximize the likelihood

Maximize the entropy of the predictive distribution to reduce overconfidence

$$\mathcal{L} = \sum_{i=1}^{N} \log p(y_i | x_i, w) + \mathbb{H}[\log p(y_i | x_i, w)]$$

  - Trained with smoothed labels instead of one-hot labels

$$[0, 0, 1, 0] \Longrightarrow [0.05, 0.05, 0.85, 0.05]$$

Proper Scoring Rules and Calibration

- Assume a predictive distribution $p_\theta(y|x)$
- Define score of p_θ on an example $(x, y) \sim p^*(x, y) = p^*(x)p^*(y|x)$ as $s(p_\theta, (x, y))$

True data distribution

- The expected score of p_θ will be $s(p_\theta, p^*) = \int p^*(x)p^*(y|x)s\big(p_\theta, (x, y)\big)dydx$

- A scoring rule is said to be a "proper scoring rule" if $s(p_\theta, p^*) \le s(p^*, p^*)$
- The log-likelihood $s\big(p_\theta, (x, y)\big) = \log p_\theta(y|x)$ is a proper scoring rule because

$$S(p_{\boldsymbol{\theta}}, p^*) = \mathbb{E}_{p^*(\boldsymbol{x})p^*(y|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(y|\boldsymbol{x})\right] \le \mathbb{E}_{p^*(\boldsymbol{x})p^*(y|\boldsymbol{x})}\left[\log p^*(y|\boldsymbol{x})\right]$$

Holds because of Gibbs inequality – entropy less than or equal to cross-entropy

- Optimizing a proper scoring rule (e.g., loglik) should do the "right thing"

If using such loss functions, the model will try to match true probabilities and be well-calibrated

But doesn't happen in practice due to optimization related issues, training set characteristics, etc

- Another proper scoring rule is the Brier score (lower is better)

$$S(p_{\boldsymbol{\theta}}, (y, \boldsymbol{x})) \triangleq \frac{1}{C}\sum_{c=1}^{C}(p_{\boldsymbol{\theta}}(y = c|\boldsymbol{x}) - \mathbb{I}(y = c))^2$$

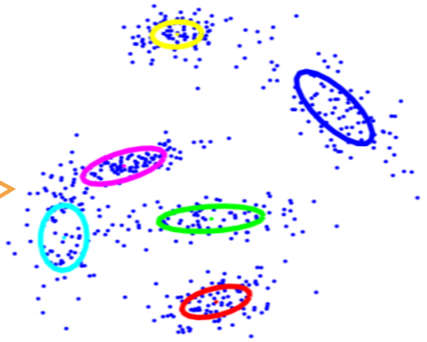
Can use Brier score (and also NLL) as a way to measure calibration of a model

Squared error of predictive distribution as compared to one-hot vector

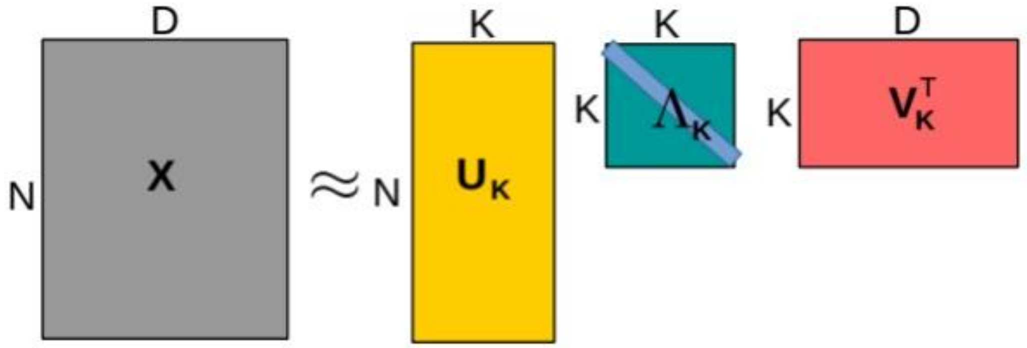
Nonparametric Bayesian methods

- Nonparametric Bayesian (NPBayes) model don't have a pre-defined number of parameters
- The model size (complexity) can grow with data
  - Have already seen Gaussian Process which is an example of an NPBayes model
- NPBayes models exist for other problems as well, e.g.,
  - Clustering/mixture models
  - Matrix factorization

Can we have the number of clusters as unbounded and learn it from data?



Can we have rank of factorization as unbounded and learn it from data?

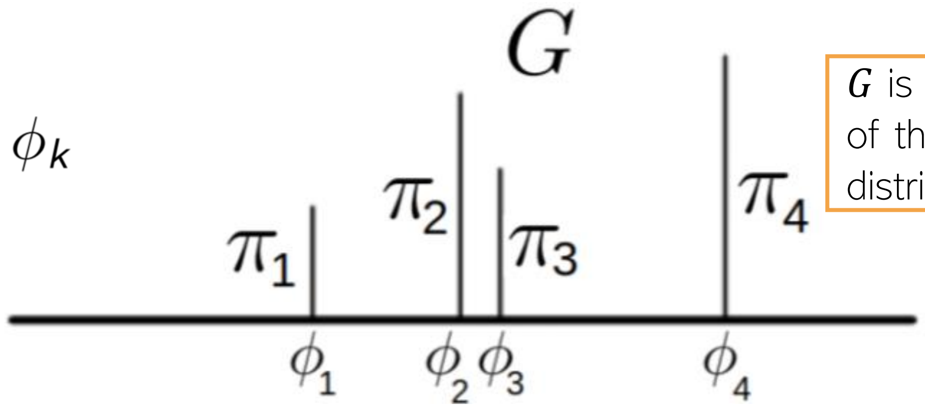


- We will see two examples of NPBayes models for
  - Mixture modeling
  - SVD-style matrix factorization

"Infinite" Mixture Models

- Consider a finite mixture model with K components with params $(\mu_k, \Sigma_k)_{k=1}^{K}$

$$(\mu_k, \Sigma_k) = \phi_k$$



G is a representation of this mixture distribution

Defined by K locations or "atoms" with parameters $\{\phi_k\}_{k=1}^{K}$ with respective selection probabilities $\{\pi_k\}_{k=1}^{K}$

$$G = \sum_{k=1}^{K} \pi_k \delta_{\phi_k}$$

- In the finite case, we can assume $\boldsymbol{\pi} = [\pi_1, \dots, \pi_K]$ and $\boldsymbol{\pi} \sim \text{Dirichlet}\left(\frac{\alpha}{K}, \dots, \frac{\alpha}{K}\right)$
- We can make it a nonparametric model by making $\boldsymbol{\pi}$ an infinite-dimensional vector

In practice, only a finite of these will have nonzero values, and others will shrink to very small (or zero), as we will see

$$\pi_1, \pi_2, \pi_3, \dots, \qquad \sum_{k=1}^{\infty} \pi_k = 1$$

Indeed. Called a "Dirichlet Process"

Related: "Stick-breaking Process"

- How to construct such a vector? Is there an infinite dimensional Dirichlet distribution?

Stick-Breaking Process (Sethuraman'94)

SBP gives us a way to construct infinite dimensional Dirichlet distribution known as the "Dirichlet Process"

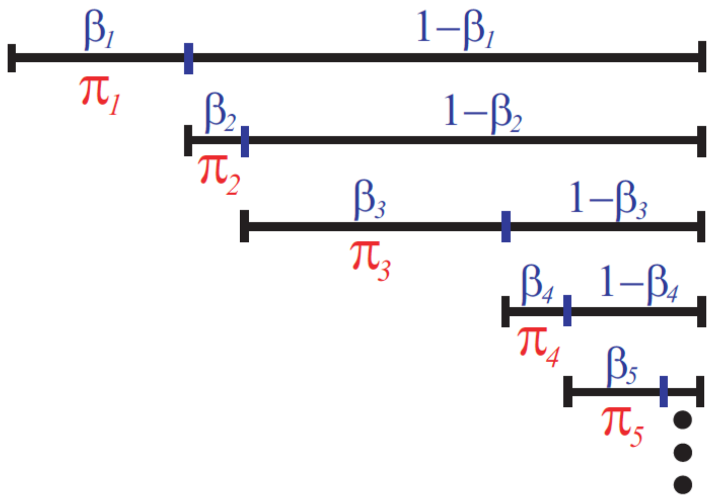
A similar SBP construction to generate a set probability values $\pi_1, \pi_2, \dots, \pi_k$ which don't sum to one but have the property that π_k shrinks to zero as k gets larger and larger. It is known as "Beta Process"

- Recursively break a length 1 stick into two pieces
- Assume breaking point in each round is drawn from a Beta distribution

$$\beta_k \sim \text{Beta}(1, \alpha) \qquad k = 1, \dots, \infty$$

$$\pi_1 = \beta_1$$

$$\pi_k = \beta_k \prod_{\ell=1}^{k-1} (1 - \beta_\ell) \qquad k = 2, \dots, \infty$$



- Can show that $\sum_{k=1}^{\infty} \pi_k - 1 \rightarrow 0$ which is what we want
- We can now have a "nonparametric/infinite" mixture distribution $G = \sum_{k=1}^{\infty} \pi_k \, \delta_{\phi_k}$
- "Location/atoms" ϕ_k can be drawn from a "base" distr G_0, say NIW if $\phi_k = (\mu_k, \Sigma_k)$
- We basically replaced the Dirichlet prior on $\boldsymbol{\pi}$ by a Stick-Breaking Process (SBP) prior

An Aside: Infinite Dimensional Dirichlet

- Drawing from an infinite-dim Dirichlet would give an infinite-dim prob. vector

$$\boldsymbol{\pi} = [\pi_1, \pi_2, \pi_3, \dots]$$

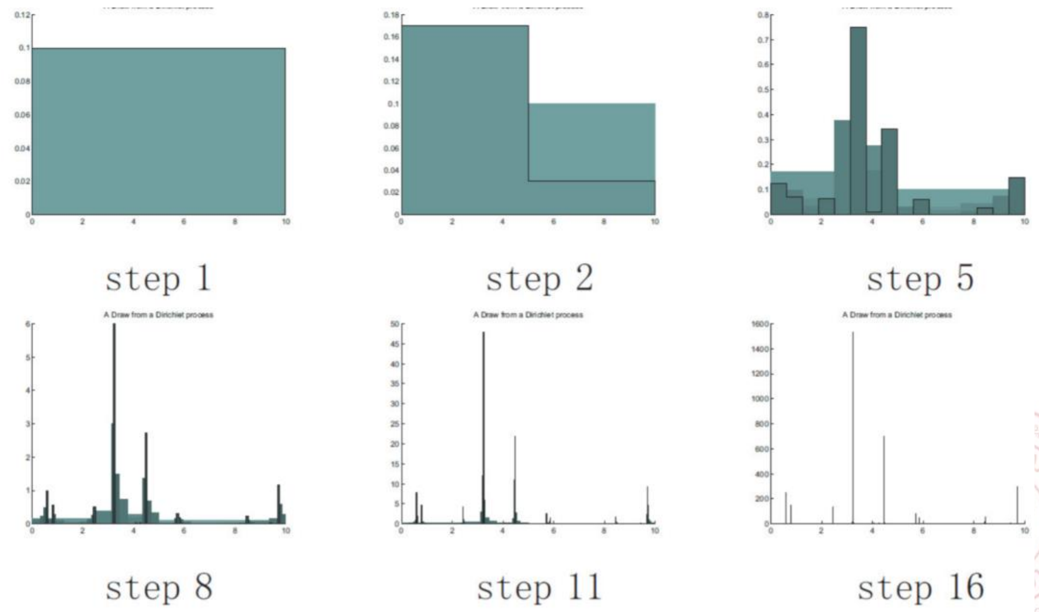
- We can construct this vector to have very few entries as nonzero
- Consider recursively drawing from a Dirichlet as defined below

$$\begin{array}{rcl} 1 & \sim & \text{Dirichlet}(\alpha) \\ (\pi_1, \pi_2) & \sim & \text{Dirichlet}(\alpha/2, \alpha/2) \\ (\pi_1\pi_{11}, \pi_1\pi_{12}, \pi_2\pi_{21}, \pi_2\pi_{22}) & \sim & \text{Dirichlet}(\alpha/4, \alpha/4, \alpha/4, \alpha/4) \end{array}$$

As the concentration parameter gets smaller and smaller, the split of values in LHS get more and more skewed

Therefore, after doing the above a few times, the π vector will only have a very few entries as nonzero and in the infinite-sized π, there will only be a finite many nonzero entries, and rest will be zero

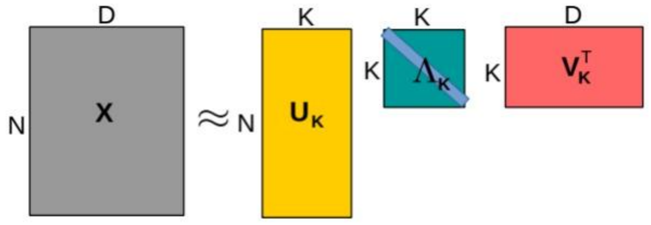
This is basically what happens in the case of Dirichlet Process / Stick-Breaking Process



"Infinite" SVD based matrix factorization

- Consider the SVD-style probabilistic model with an *a priori* unbounded K

$$\mathbf{X} = \sum_{k=1}^{\infty} \lambda_k \boldsymbol{u}_k \boldsymbol{v}_k^\top$$



- Consider the following prior on each "singular values" λ_k

$$\begin{aligned} \lambda_k &\sim \mathcal{N}(0, \tau_k^{-1}) \\ \tau_k &= \prod_{\ell=1}^{k} \delta_\ell \\ \delta_\ell &\sim \text{Gamma}(\alpha, 1) \quad \text{where } \alpha > 1 \end{aligned}$$

Precision keeps on getting larger and larger as k grows (thus variance keeps getting small and smaller)

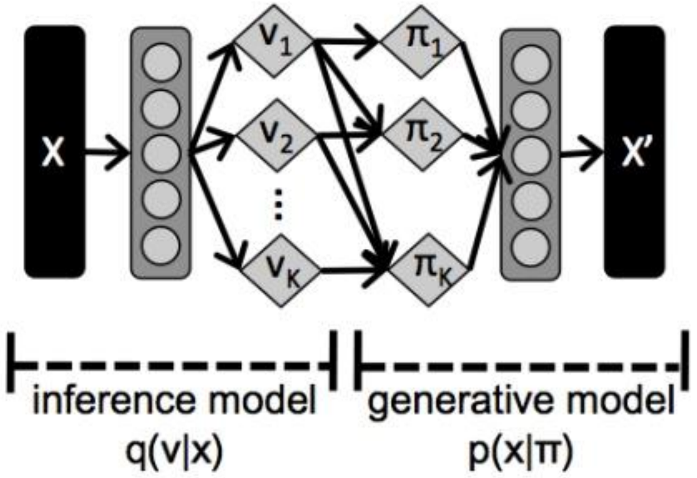
Called "multiplicative gamma process"

Thus $\mathbb{E}[\delta_\ell] = \alpha$ (greater than 1 in expectation)

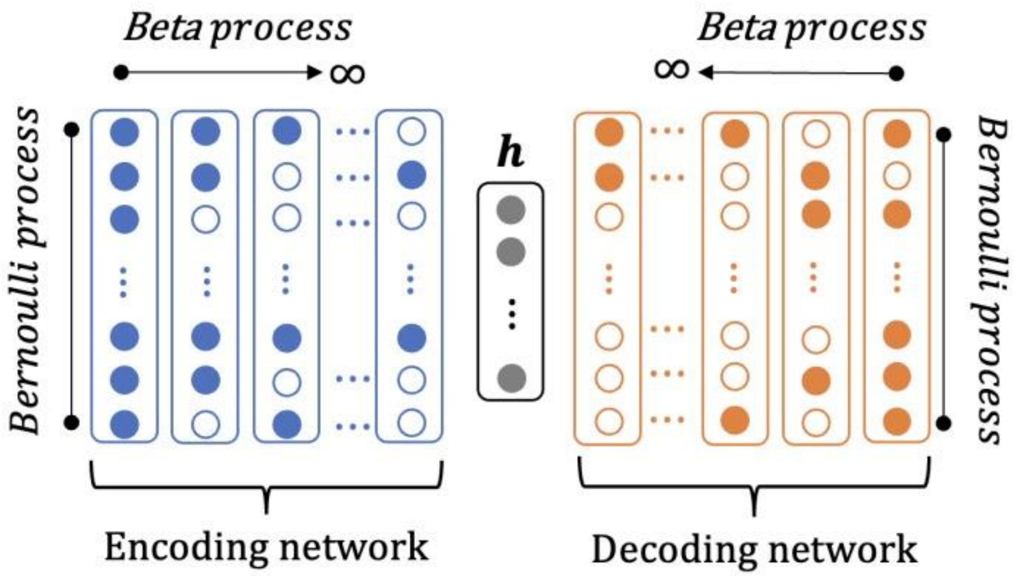
- In practice we can set K to be a sufficiently very large
  - Due to the shrinkage property, only a finite many λ_k will be nonzero
  - The nonzero λ_k's will dictate the effective K

Some recent applications of NPBayes in Deep Learning

- Can use NPBayes approaches to learn the “right” size of a deep network
- Example: Width and depth of a deep neural network in VAEs



Stick-breaking VAE# (ICLR, 2017)



AdaVAE* (NeurIPS, 2023)

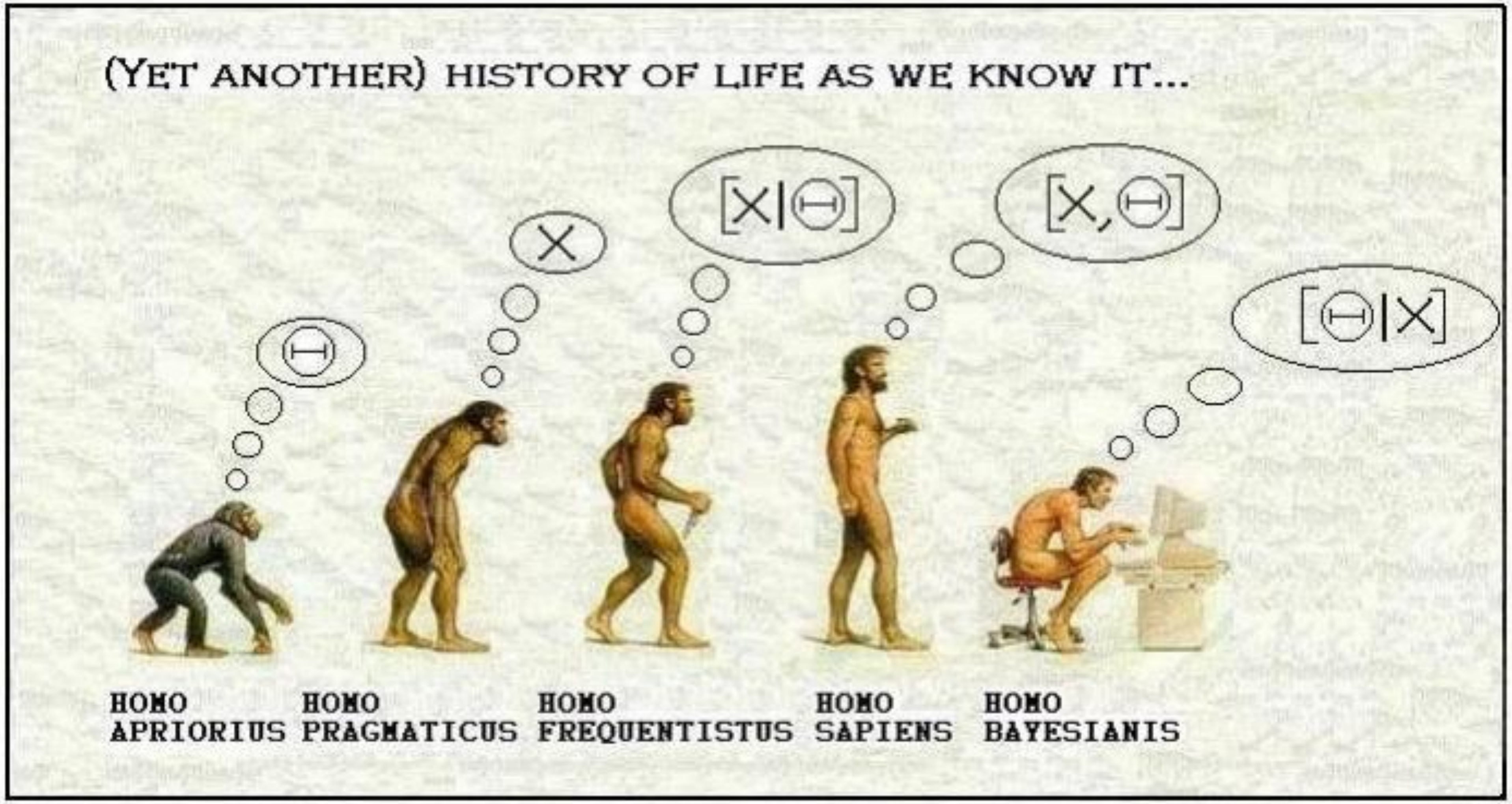
#Stick-Breaking Variational Autoencoders (ICLR, 2017)

*AdaVAE: Bayesian Structural Adaptation for Variational Autoencoders

Conclusion

- Probabilistic modeling provides a natural way to think about models of data
- Many benefits as compared to non-probabilistic approaches
  - Easier to model and leverage uncertainty in data/parameters
  - Principle of marginalization while making prediction
  - Easier to encode prior knowledge about the problem (via prior/likelihood distributions)
  - Easier to handle missing data (by marginalizing it out if possible, or by treating as latent variable)
  - Easier to build complex models can be neatly combining/extending simpler probabilistic models
  - Easier to learn the "right model" (hyperparameter estimation, nonparametric Bayesian models)
- Bayesian approaches as well as single model based uncertainty
- Uncertainty is important but proper calibration of uncertainty is also important
- Fast-moving field, lots of recent advances on new models and inference methods

Thank You!


(YET ANOTHER) HISTORY OF LIFE AS WE KNOW IT...
[Θ]
[X]
[X|Θ]
[X,Θ]
[Θ|X]
HOMO APRIORIUS
HOMO PRAGMATICUS
HOMO FREQUENTISTUS
HOMO SAPIENS
HOMO BAYESIANIS