

# 딥러닝 에스프레소

## 딥러닝을 위한 베이지안 통계 Day1

### Introduction, Probability Theory

2020  
멀티캠퍼스

박수철

v1.0

# 소개



- 박수철
- Independent Researcher/Developer
- scspark@gmail.com  
[github.com/scspark20](https://github.com/scspark20)
- 모두의연구소 - RubatoLab  
가우디오 - AI Scientist  
멀티캠퍼스 - Generative Models  
Deep Learning Espresso
- 연구 분야  
Speech Synthesis  
Generative Models  
Bayesian Reasoning

# 다루는 내용

# 학습 목표

- Bayesian model을 통해 결정에 대한 불확실성(uncertainty)를 구할 수 있다.
- Bayesian model의 핵심이 되는 prior, likelihood, posterior의 개념을 이해할 수 있다.
- Bayesian model를 deep learning에 적용하기 위해서 필요한 variational inference의 개념을 이해할 수 있다.
- Bayesian을 접목한 deep learning model들을 이해하고 활용할 수 있다.

# 다루는 내용

## ● Bayesian Linear Regression

Linear Regression의 Bayesian 버전입니다. Bayesian이라고 하면 model의 parameter를 고정된 값이 아닌 random variable로 여기는 것입니다. 이를 통해 predictive distribution을 얻고 예측에 대한 불확실성(uncertainty)을 표현할 수 있습니다.

## ● GMM (Gaussian Mixture Models)

데이터를 여러 Gaussian distribution들로 clustering 합니다. 데이터마다 주어진 cluster component를 latent variable이라 여기면 visible data로부터 latent data를 inference하는 문제로 생각할 수 있습니다. Gaussian mixture로 주어진 모델이 데이터를 잘 표현하도록 marginal likelihood를 최대화하는 방식으로 트레이닝이 이루어지며 이를 위해 EM 알고리즘이 등장합니다.

## ● Probabilistic PCA

Continuous latent space 상의 prior에서 뽑은 샘플이 data space로 linear transform을 통해 맵핑되고 이를 mean으로 구성된 Gaussian distribution이 data distribution을 나타내도록 모델링하는 generative model입니다. GMM과 같이 marginal likelihood를 최대화하는 방식으로 트레이닝 되고, close form이나 EM 알고리즘을 통해 solution을 얻을 수 있습니다.

## ● Auto-Encoders

Deep Learning의 대표적인 모델로 dimension reduction, feature extraction을 수행합니다. Encoder, Decoder 구조를 통해 데이터를 압축하고 다시 복원합니다. 이 과정에서 얻은 latent variable을 데이터의 중요한 feature가 담긴 정보로 여길 수 있습니다.

# 다루는 내용

## ● VAE (Variational Auto-Encoders)

Deep learning에서 generative model로써 중요한 위치를 차지하고 있는 모델이며 control 가능한 latent variable을 얻는데 목적이 있습니다. Auto-Encoders와 같이 Encoder, Decoder 구조를 가지고 있으며, 이에 더해 latent variable의 분포를 Gaussian 등으로 제약시키는 특징을 가지고 있습니다. GMM, Probabilistic PCA와 같이 marginal likelihood를 최대화 하는 것을 목적으로 하지만 계산의 intractability 때문에 variational inference가 적용되고 ELBO를 최대화하는 것으로 대체됩니다.

## ● BBB (Bayes by Backprop.)

Bayesian을 deep learning에 접목시켜 regression과 classification 문제를 해결합니다. VAE가 latent variable의 posterior를 얻기 위해 variational inference를 사용하였다면 BBB에서는 모델의 weight의 posterior를 얻기 위해 variational inference를 사용합니다. Reparameterization trick 등 VAE와 기본적으로 접근 방식이 비슷합니다.

## ● MC Dropout (Dropout as a Bayesian Approximation)

Neural networks에 dropout을 포함시켜 Bayesian neural networks처럼 작동하도록 합니다. Weight의 posterior를 구하기 위해 variational inference를 사용하는 번거로움을 해결할 수 있습니다. Bayesian neural networks와 마찬가지로 결과에 대한 uncertainty를 측정할 수 있습니다.

## ● GAN (Generative Adversarial Networks)

VAE와 함께 deep learning에서 대표적인 generative models 중 하나입니다. Generator와 discriminator를 서로 경쟁 관계에 놓아 좋은 결과를 얻을 수 있도록 트레이닝 합니다. VAE와 달리 매우 선명한 결과를 생성하는 것이 특징입니다. Bayesian 모델은 아니지만 data distribution과 model distribution, KL-divergence 등의 개념이 적용됩니다.

# 선행 지식

# 선행 지식

- Deep learning 기초

- CNN, RNN, Fully-connected Layers

- Backpropagation

- Stochastic Gradient Descent

- Deep learning frameworks

- Tensorflow, Pytorch, Keras

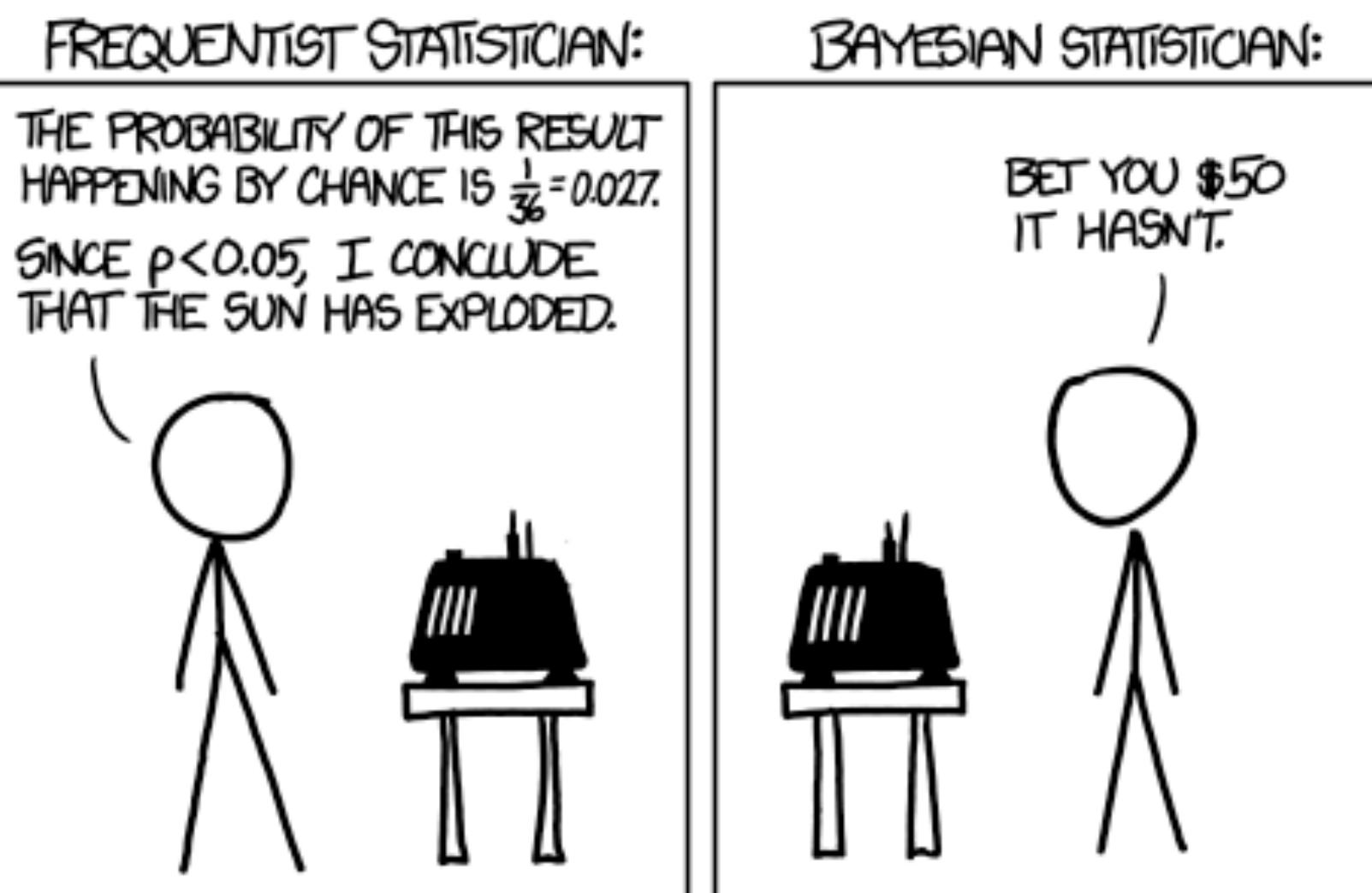
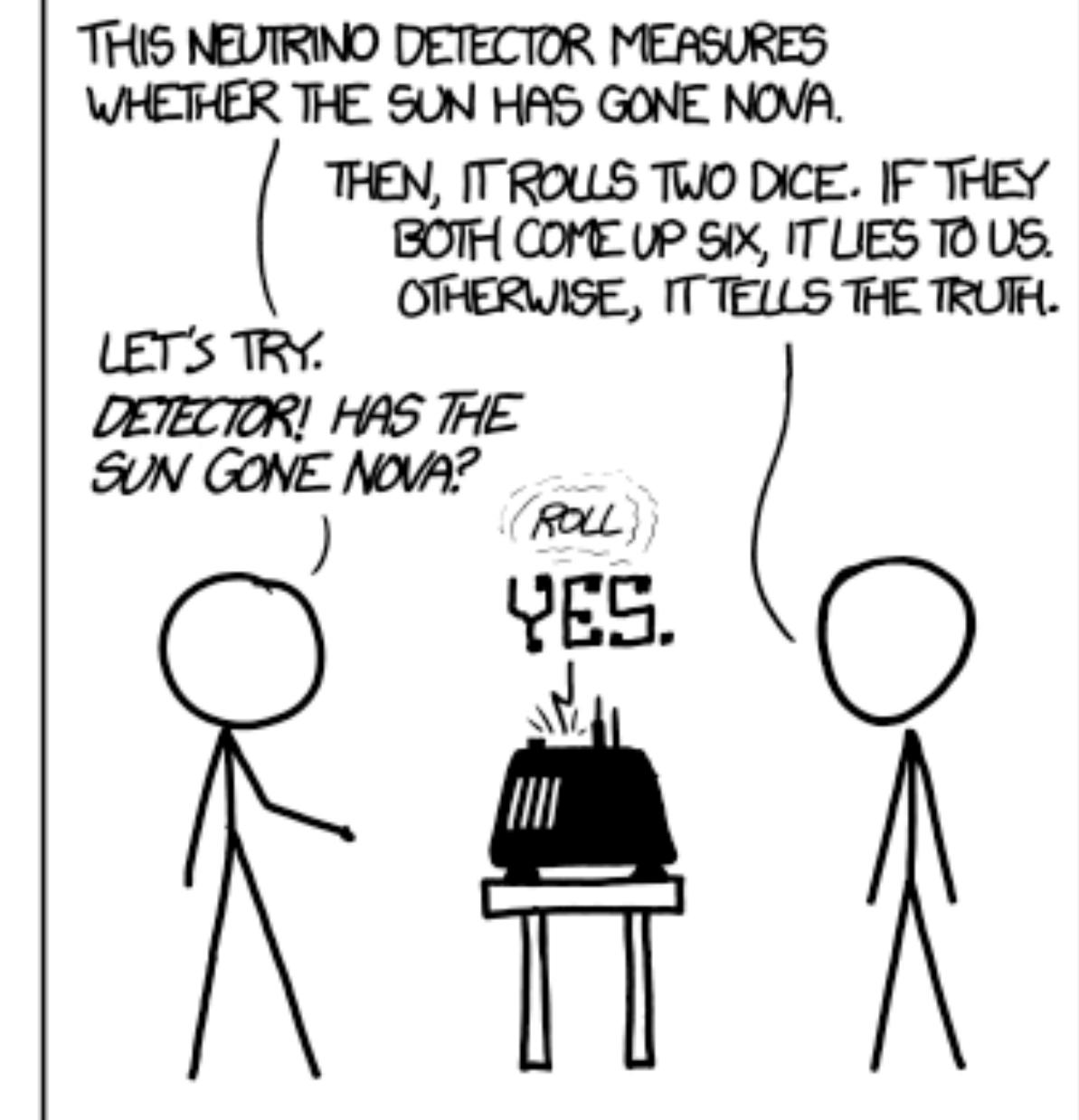
- 기초 수학 과목

- Calculus, Linear Algebra, Probability and Statistics

Prior

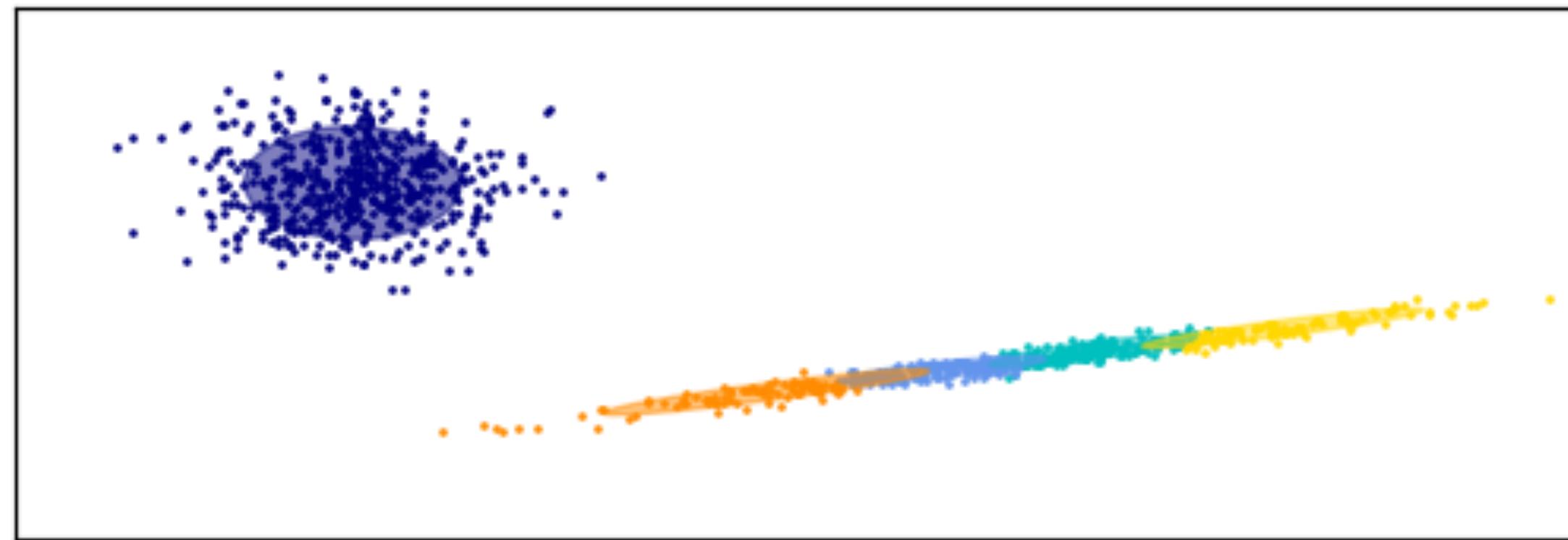
# Prior

DID THE SUN JUST EXPLODE?  
(IT'S NIGHT, SO WE'RE NOT SURE.)

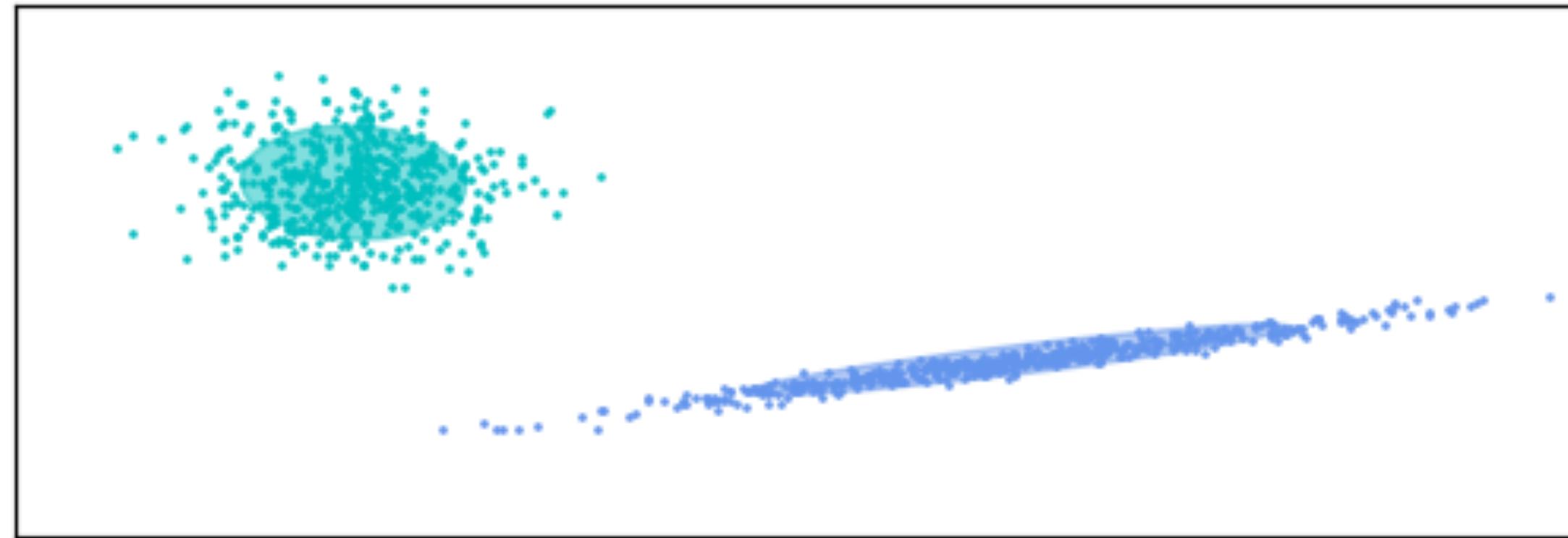


# Prior GMM vs. Variational GMM

Gaussian Mixture



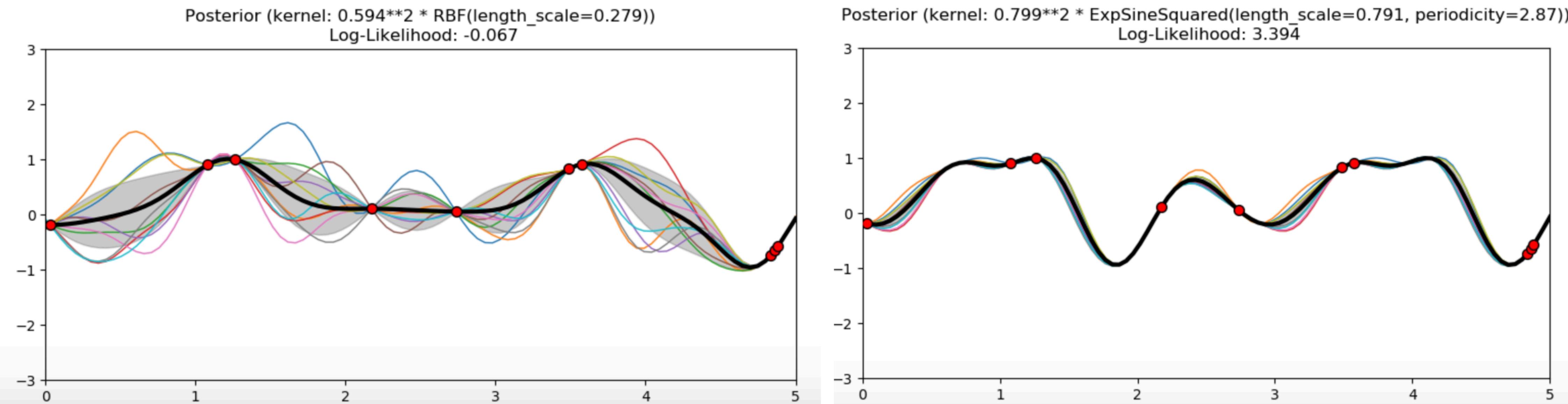
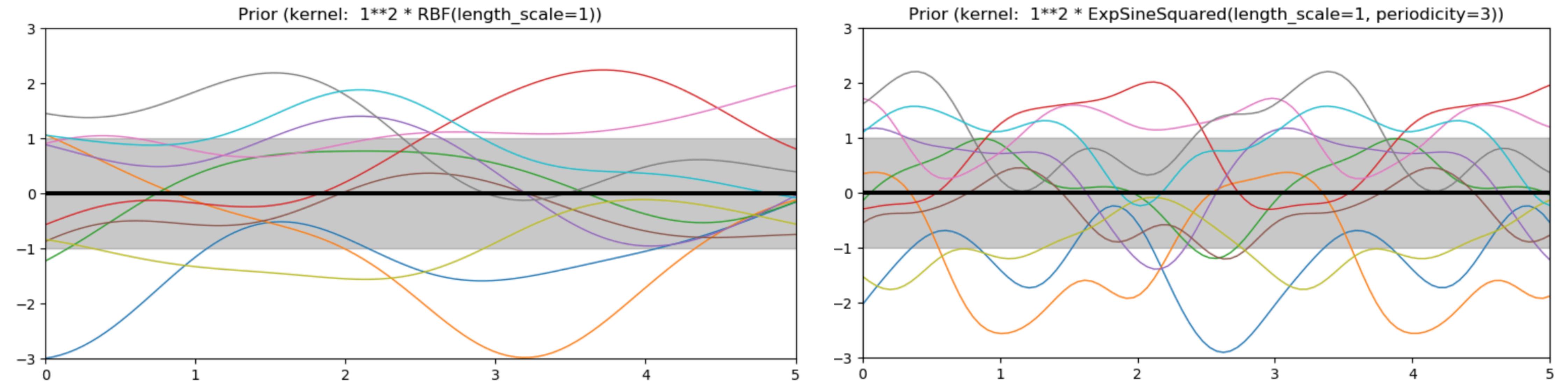
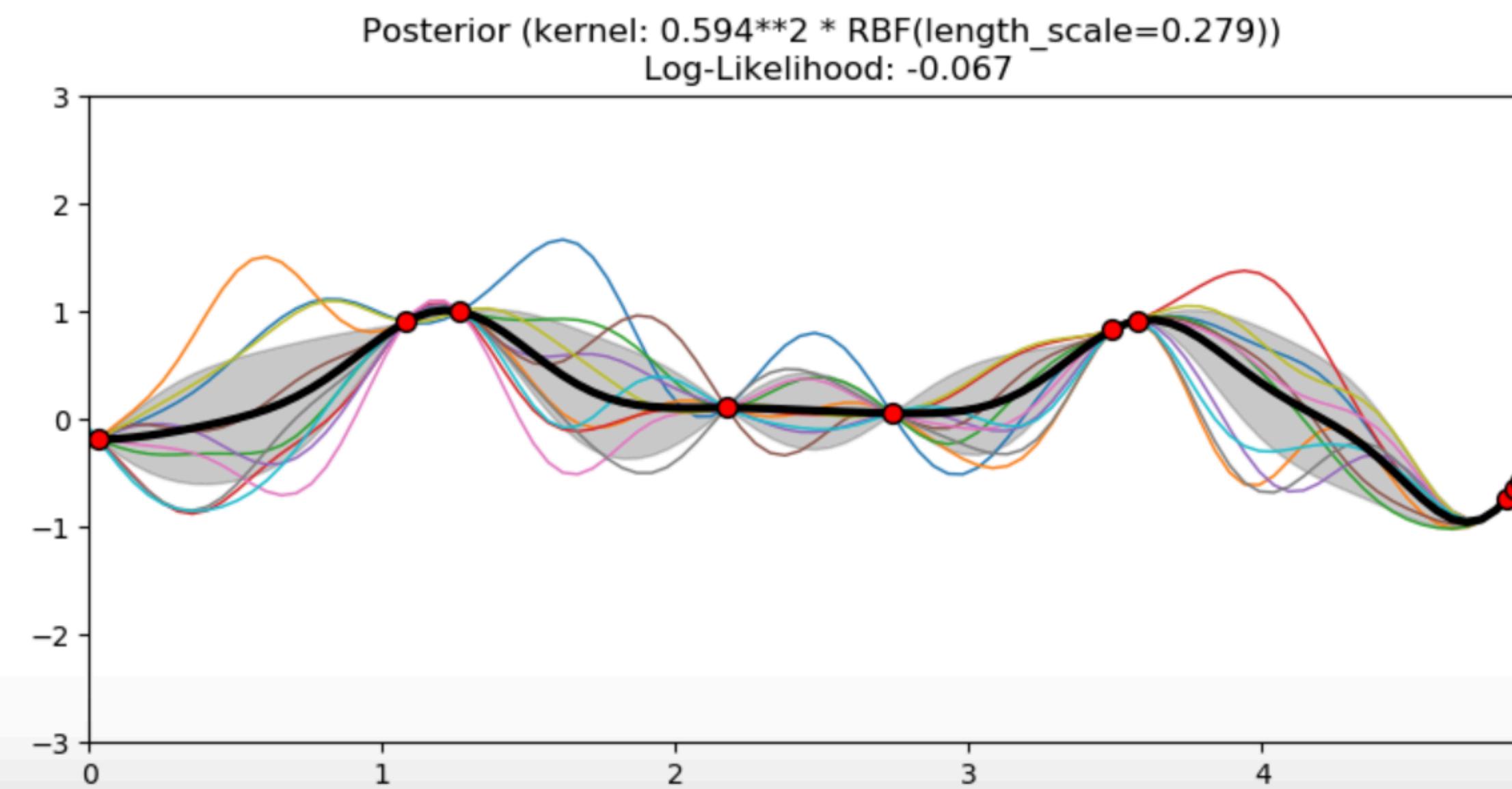
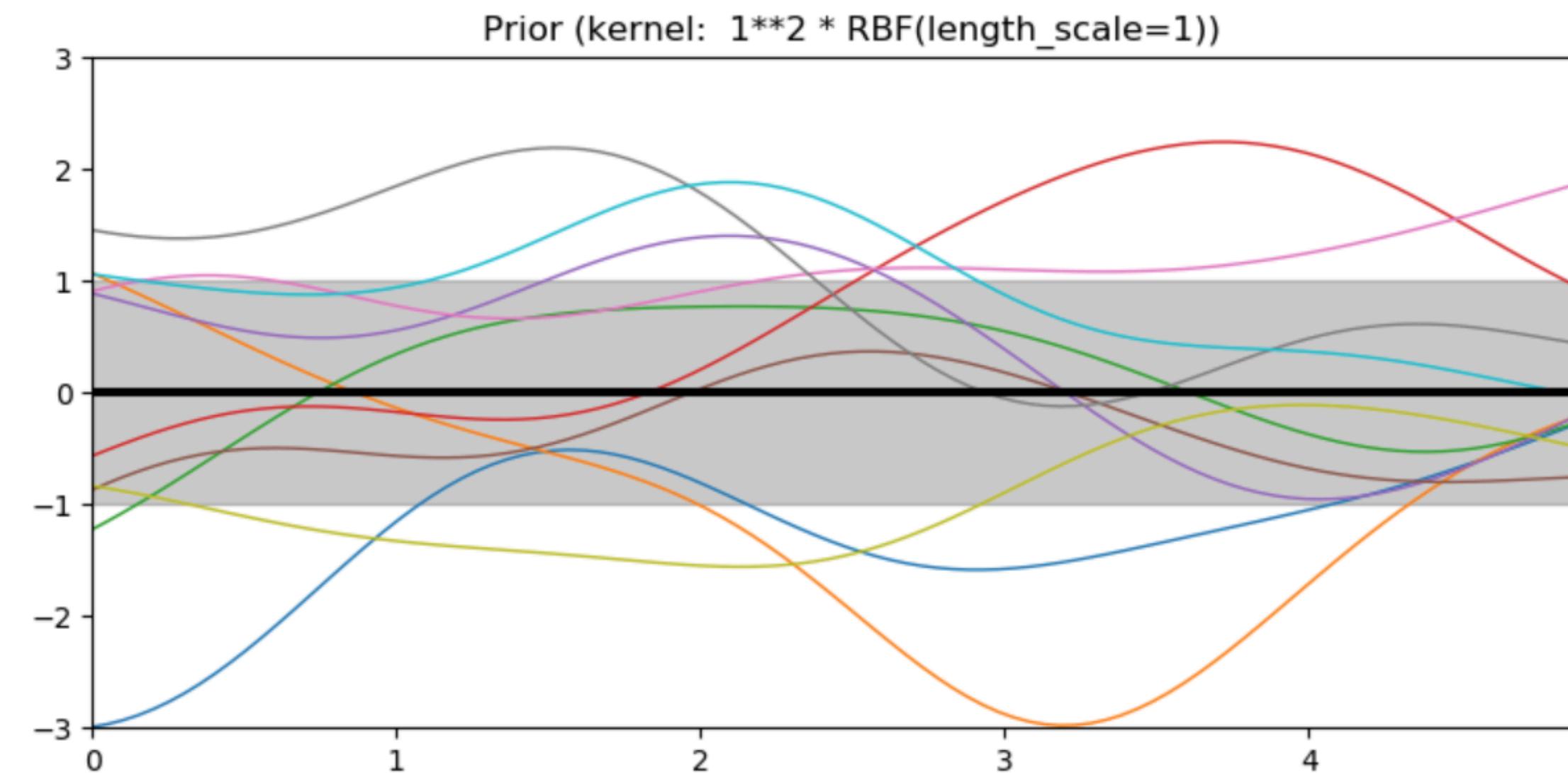
Bayesian Gaussian Mixture with a Dirichlet process prior



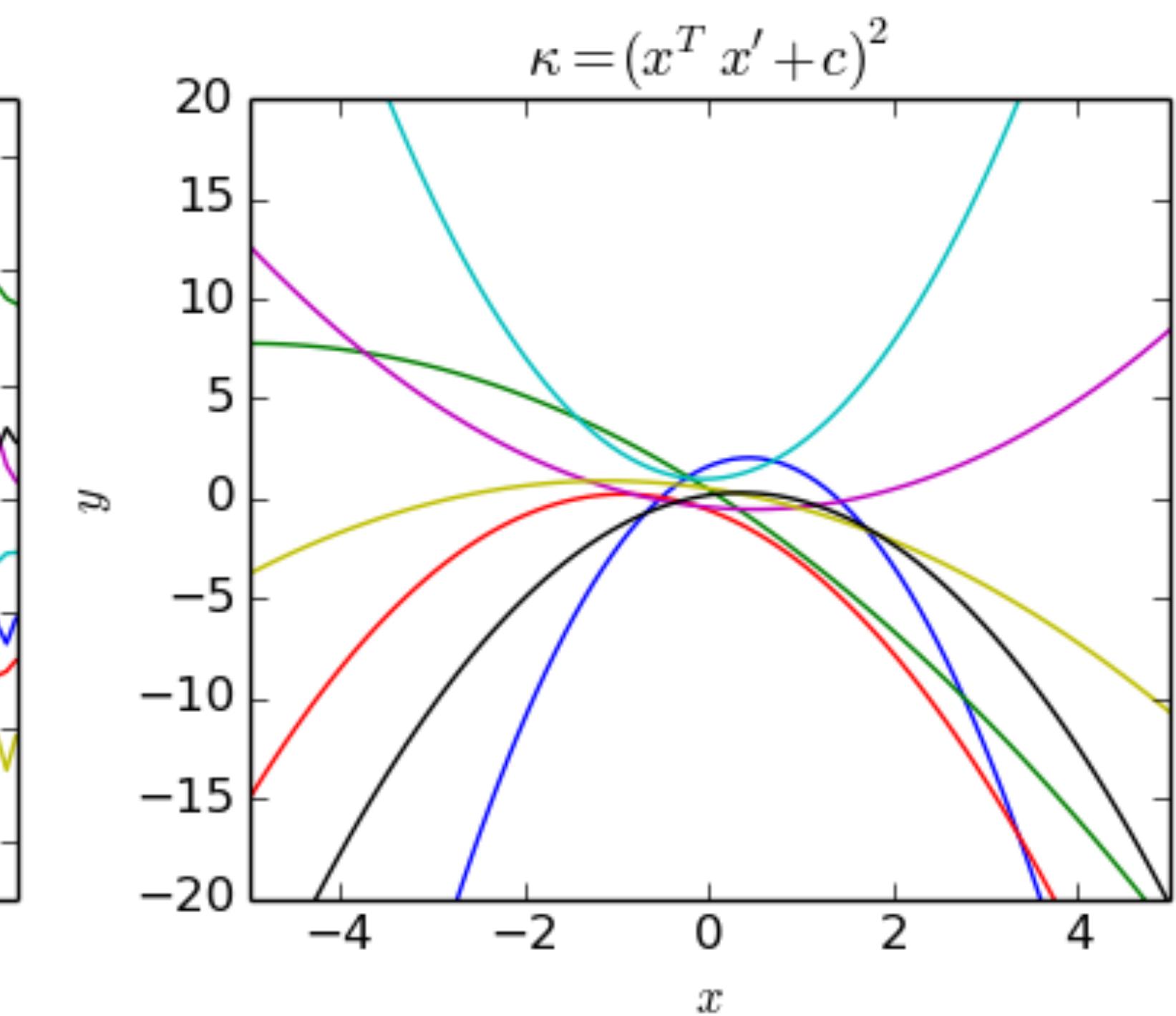
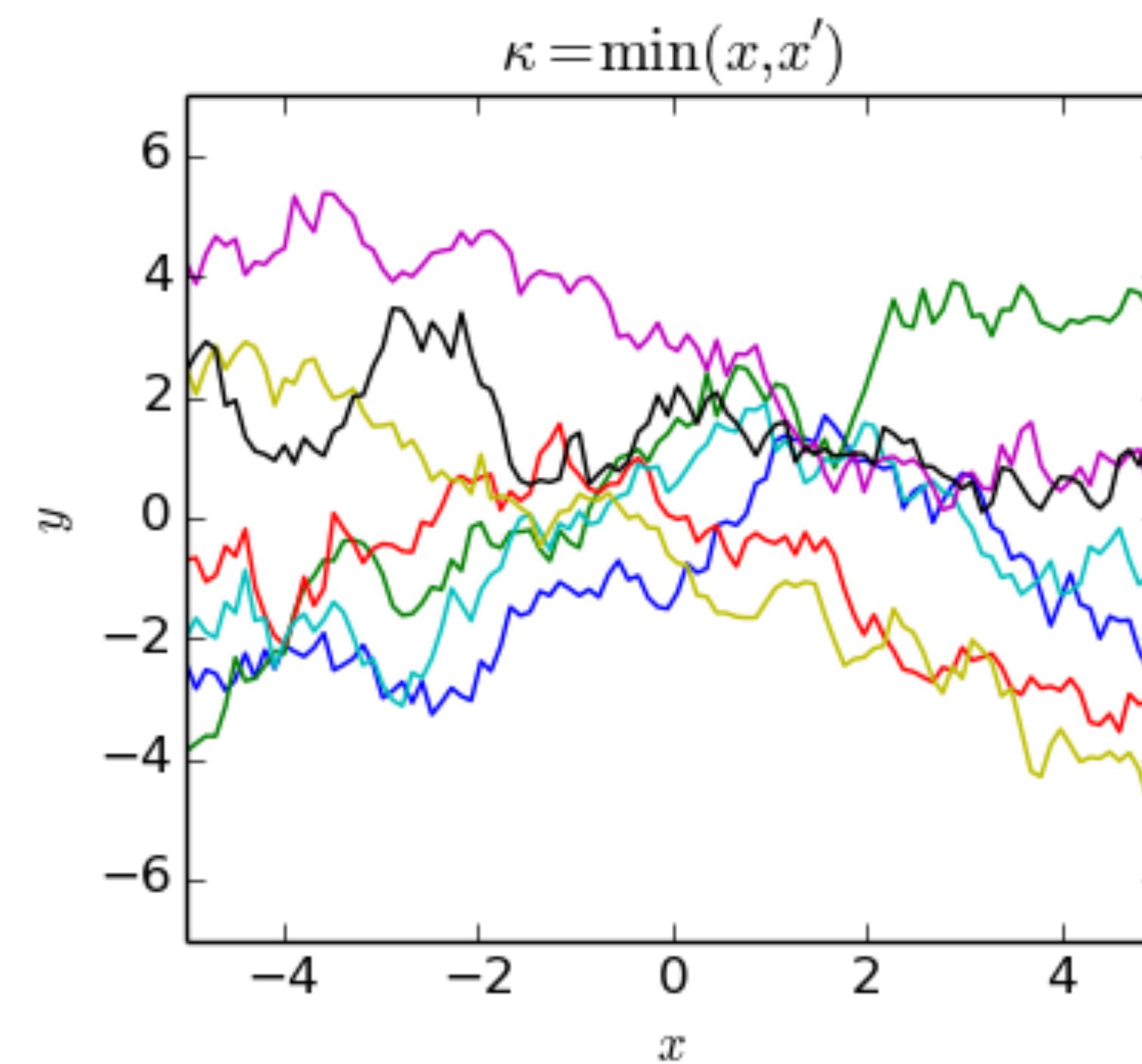
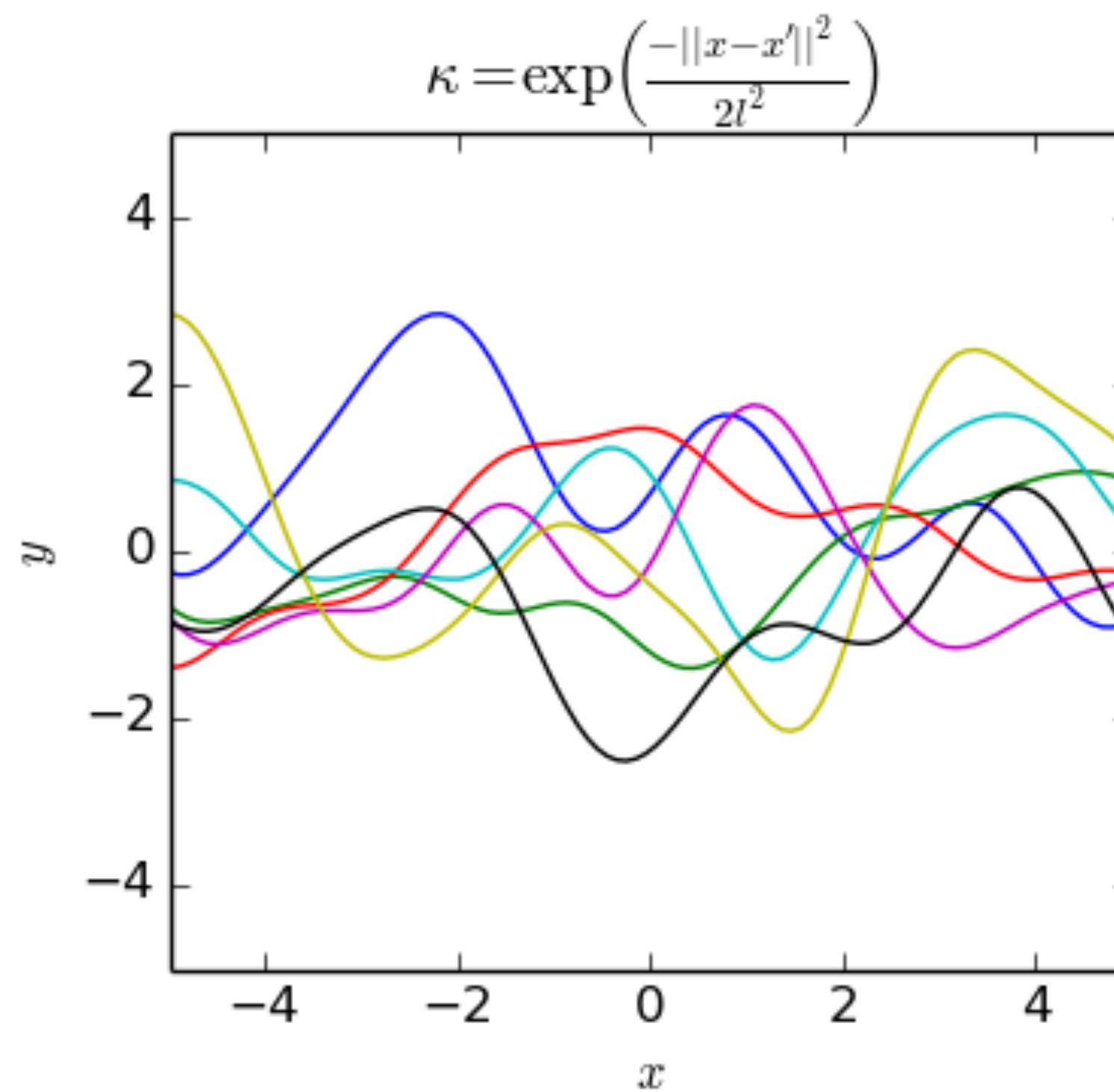
# Prior GMM vs. Variational GMM



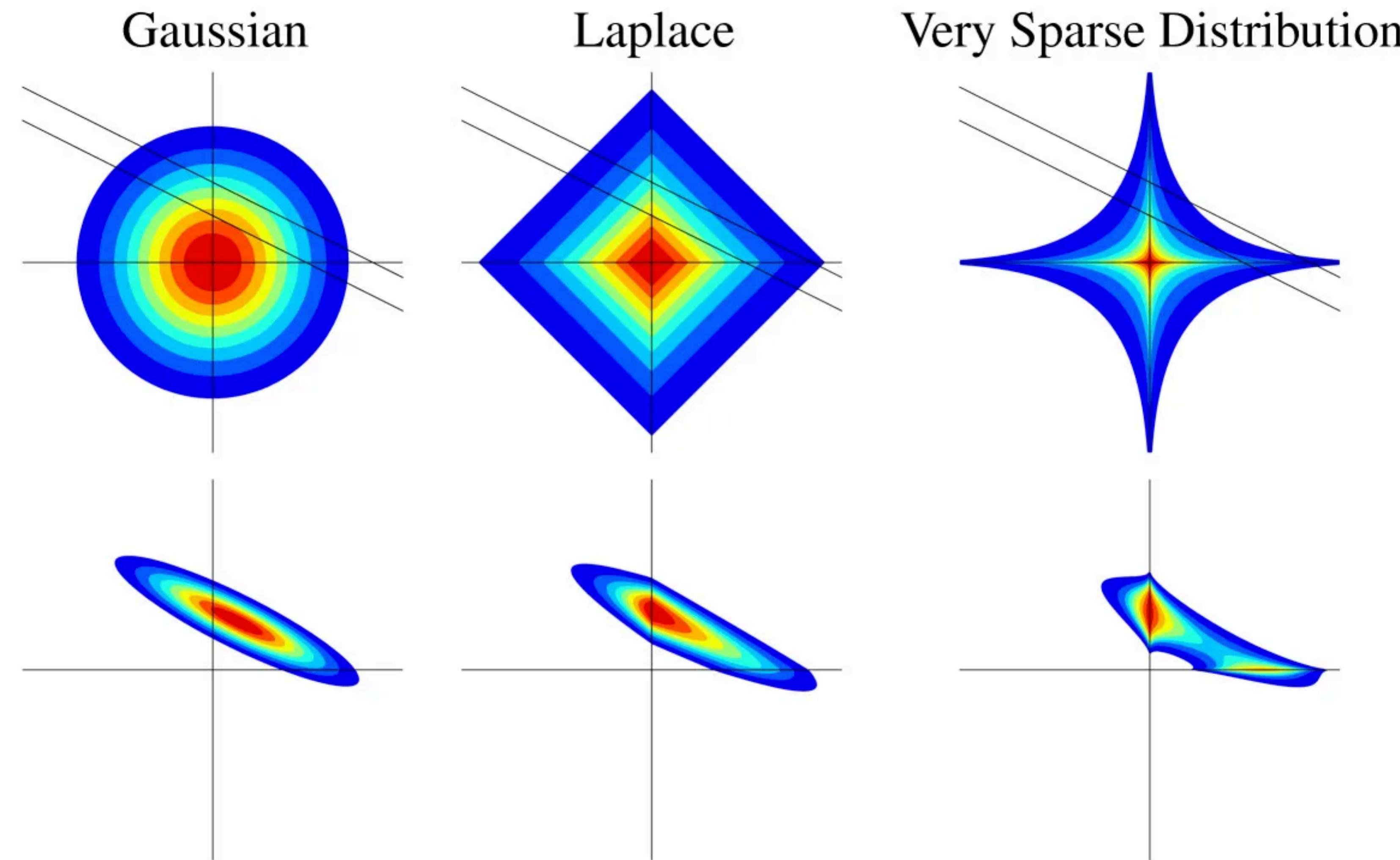
# Prior Gaussian Process



# Prior Gaussian Process

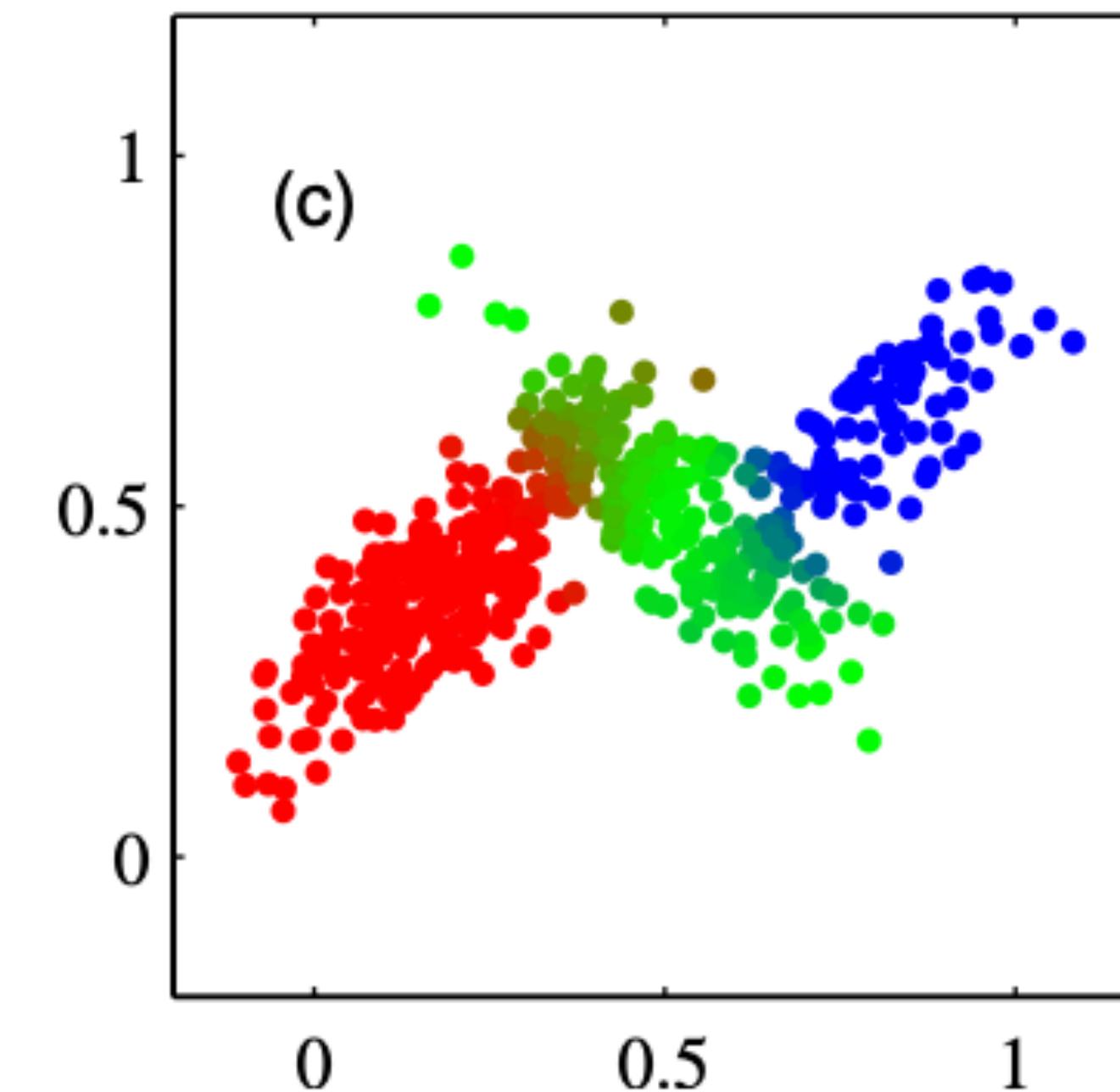
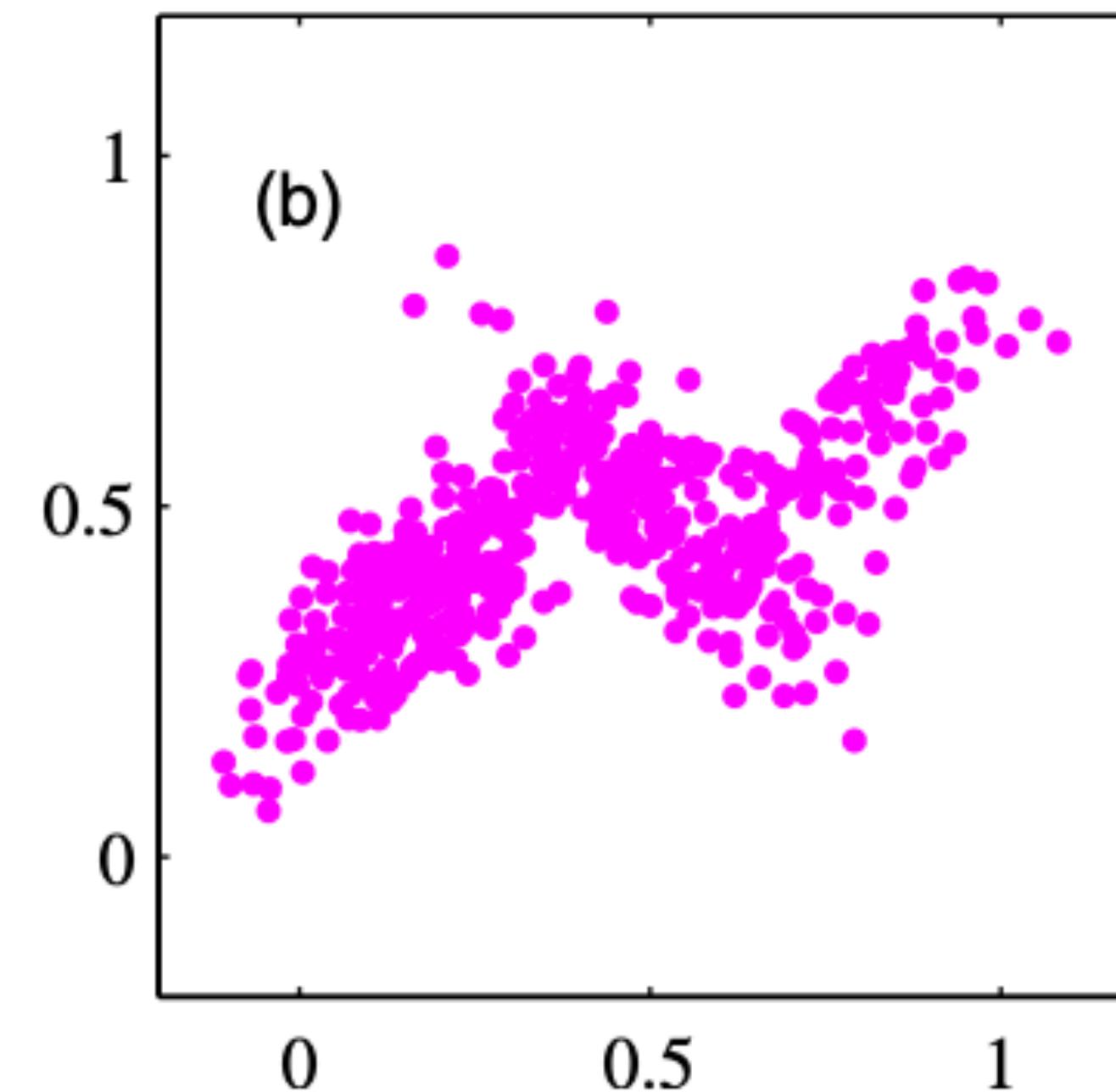
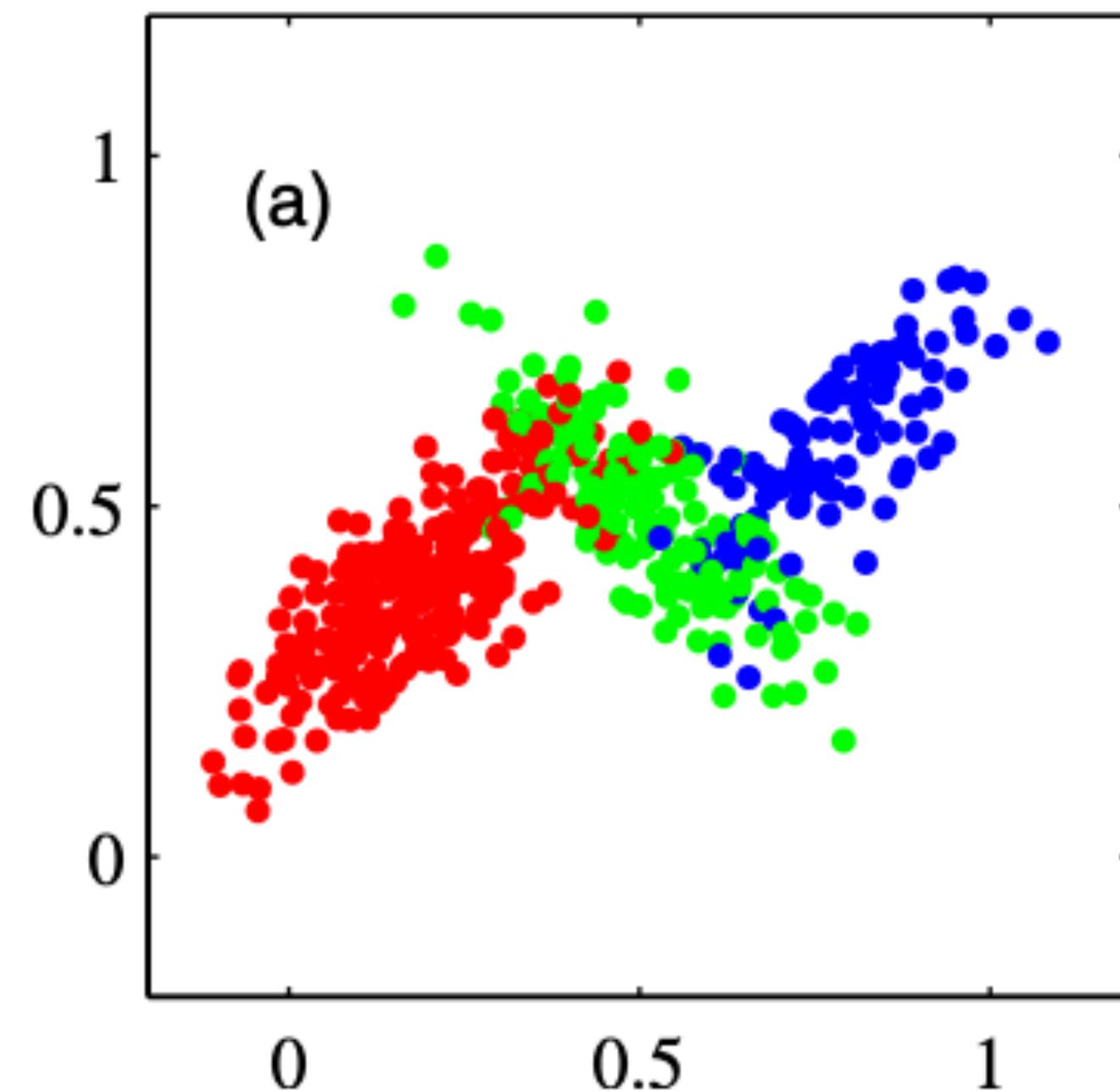


# Prior Gaussian, Laplace, Sparse Distribution



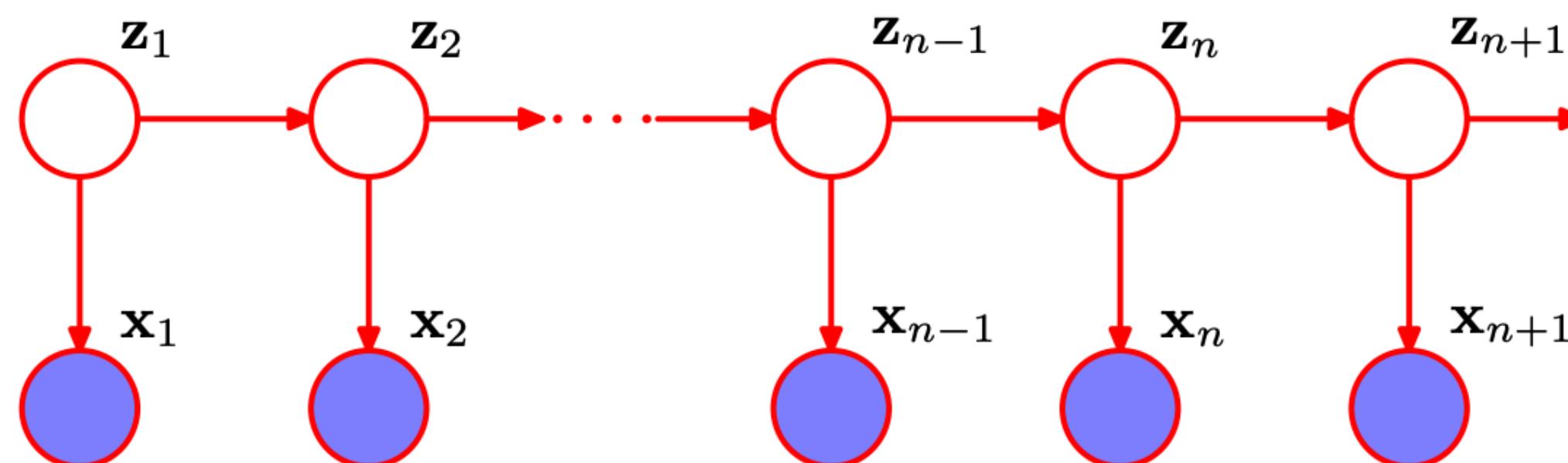
# Hidden Variables

# Hidden Variables Gaussian Mixture Models

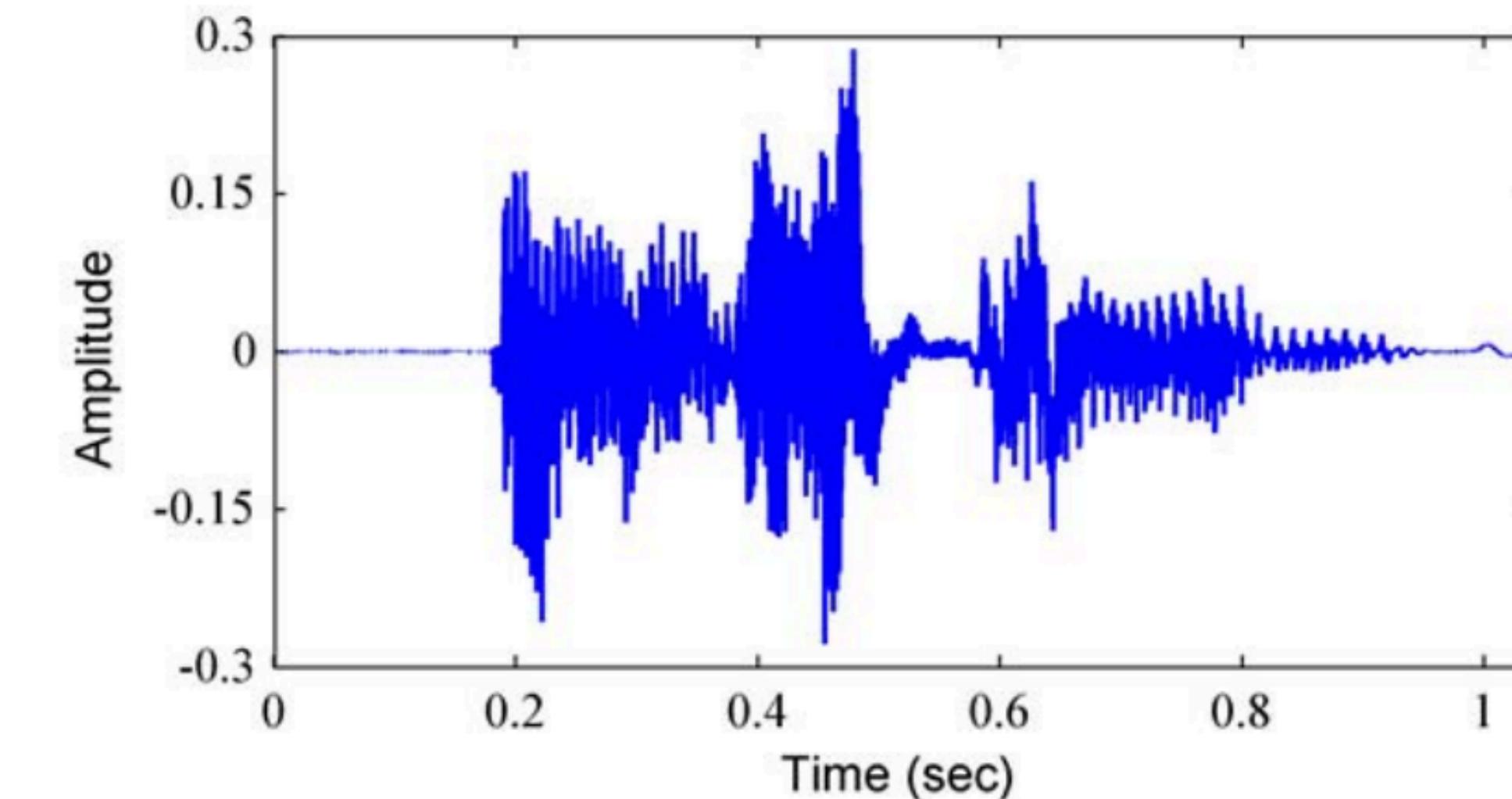


# Hidden Variables Hidden Markov Chains

Directed graphs representation of the HMM

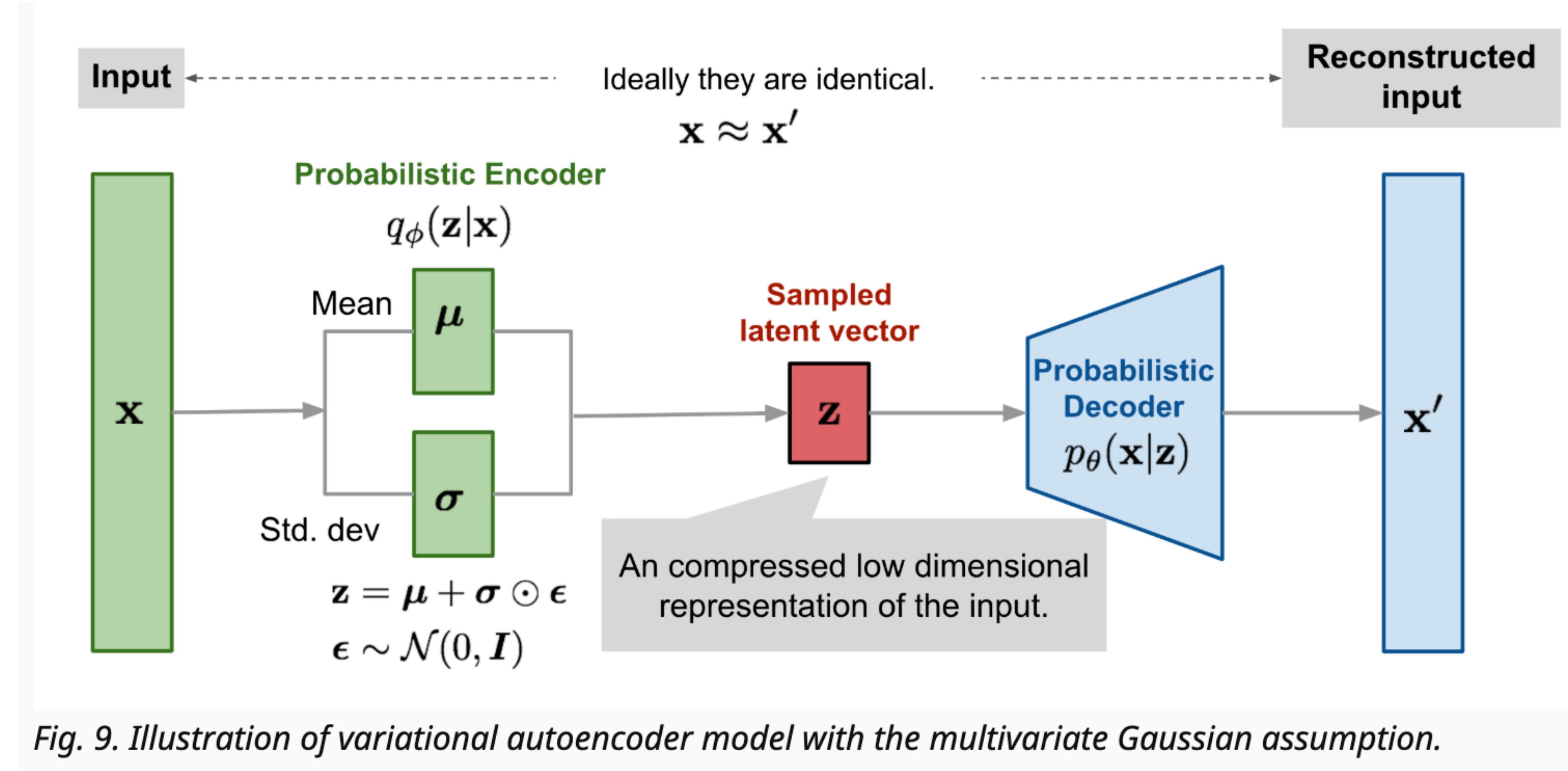


Speech recognition using the HMM



| b |    ey |    z |    th |    ih |    er |    em |  
|         Bayes' |              Theorem |

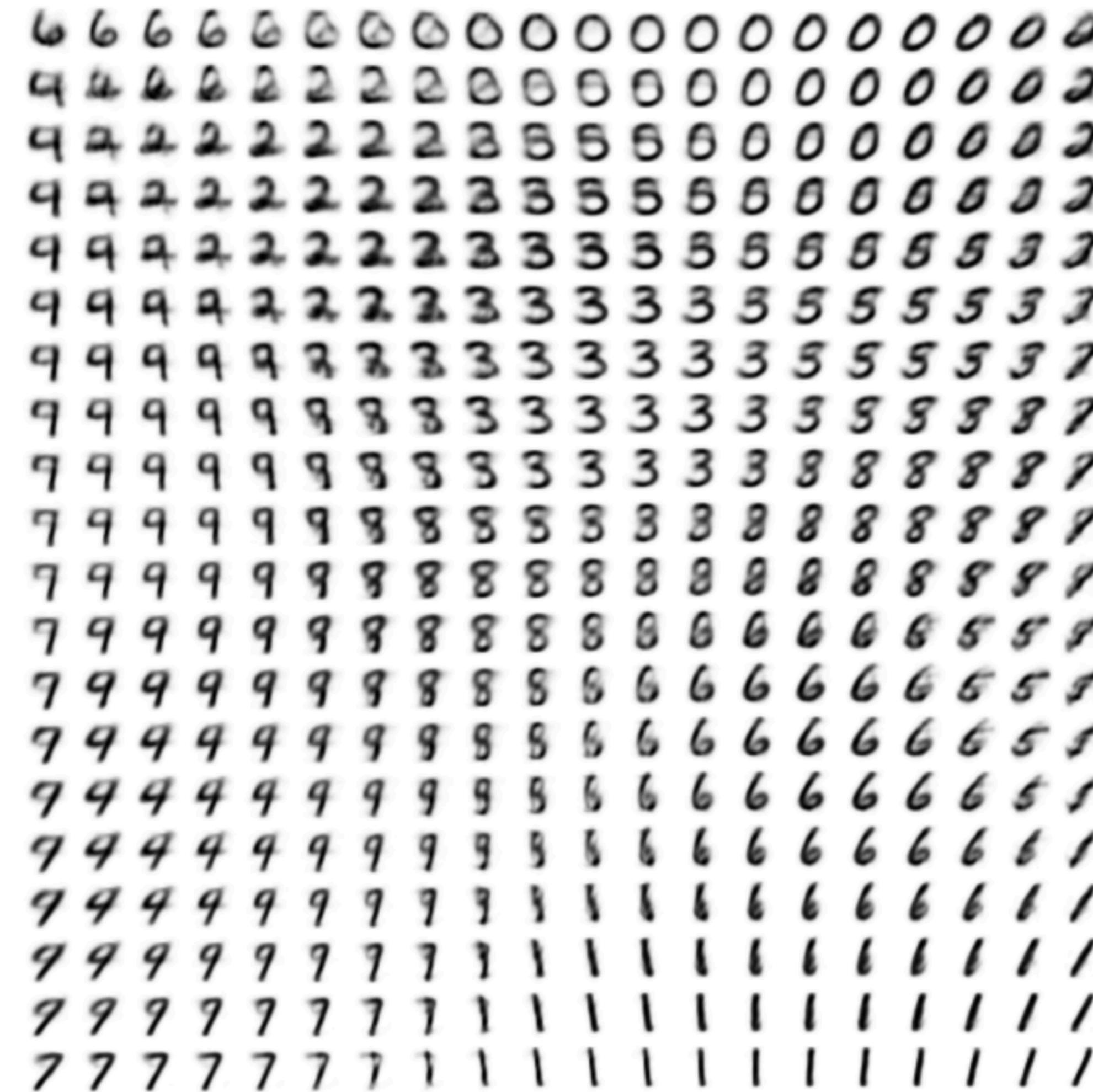
# Hidden Variables Variational Auto-Encoders



# Hidden Variables Variational Auto-Encoders



(a) Learned Frey Face manifold



(b) Learned MNIST manifold

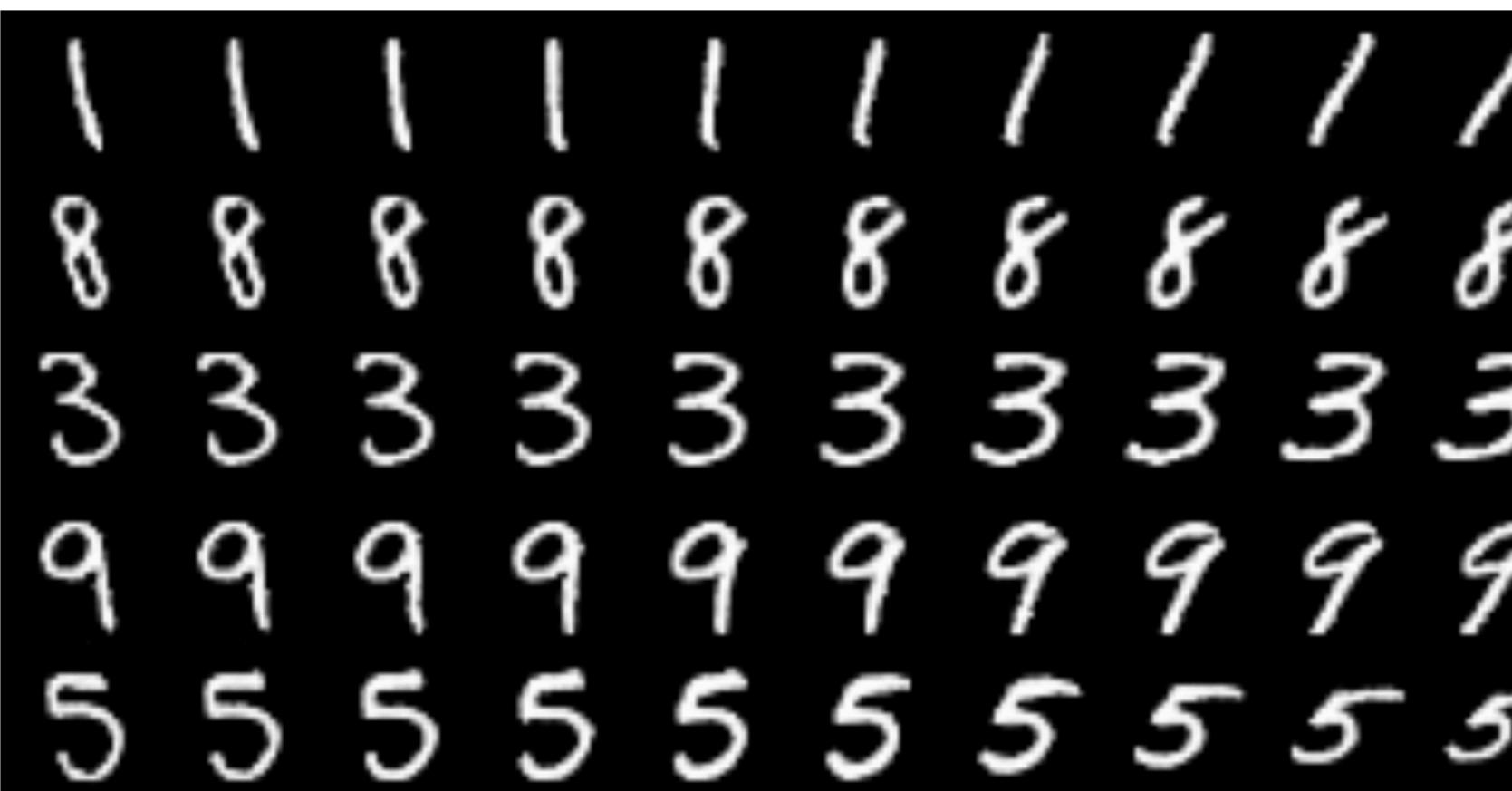
# Hidden Variables InfoGAN



(a) Varying  $c_1$  on InfoGAN (Digit type)



(b) Varying  $c_1$  on regular GAN (No clear meaning)



(c) Varying  $c_2$  from  $-2$  to  $2$  on InfoGAN (Rotation)



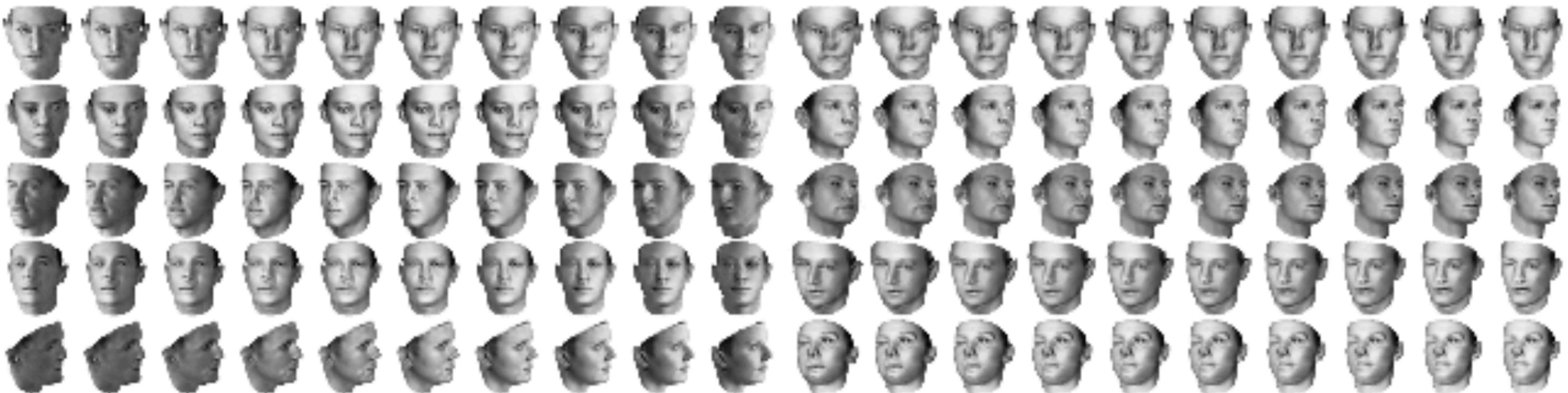
(d) Varying  $c_3$  from  $-2$  to  $2$  on InfoGAN (Width)

# Hidden Variables InfoGAN



(a) Azimuth (pose)

(b) Elevation



(c) Lighting

(d) Wide or Narrow

# Hidden Variables InfoGAN

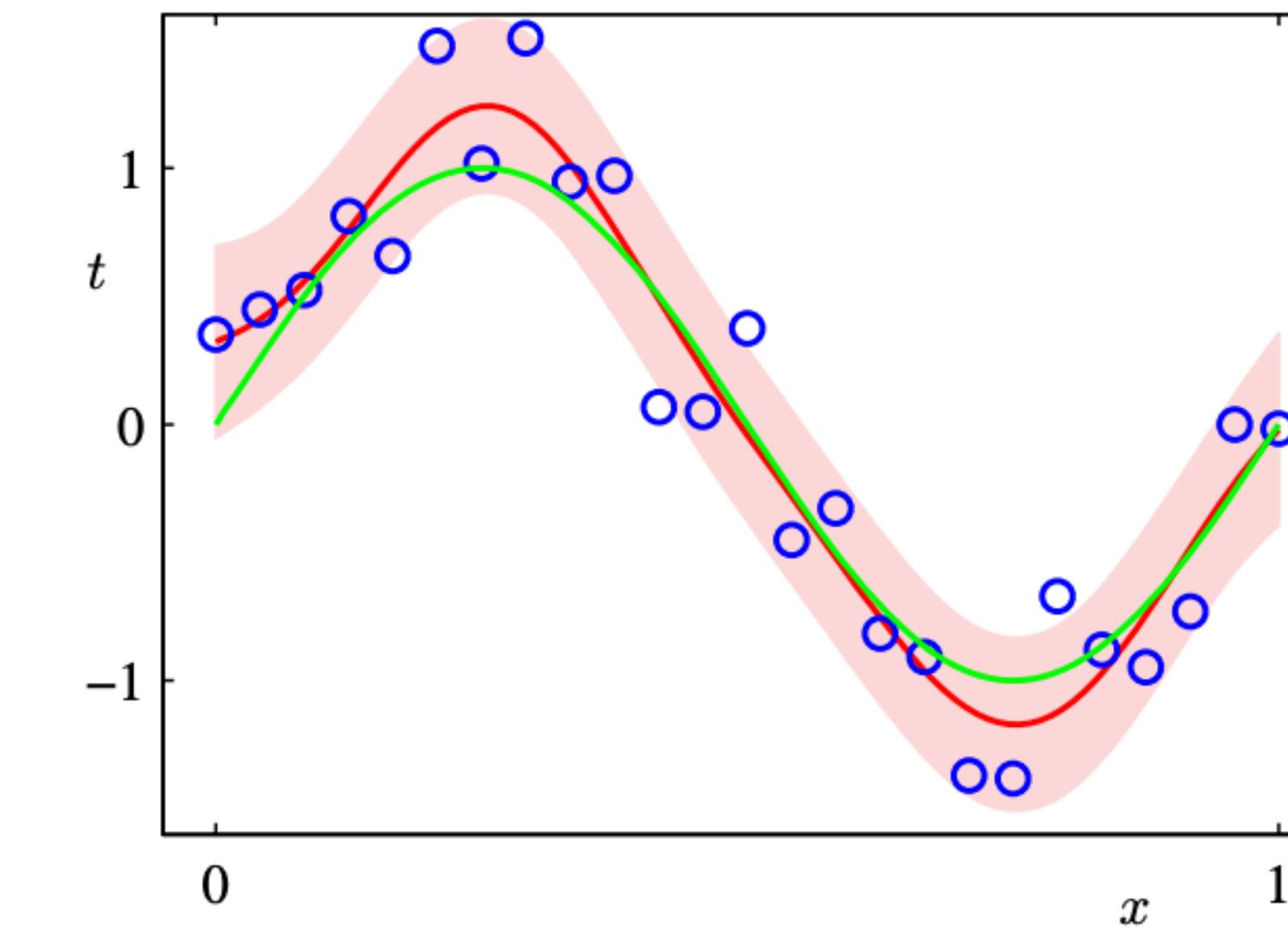
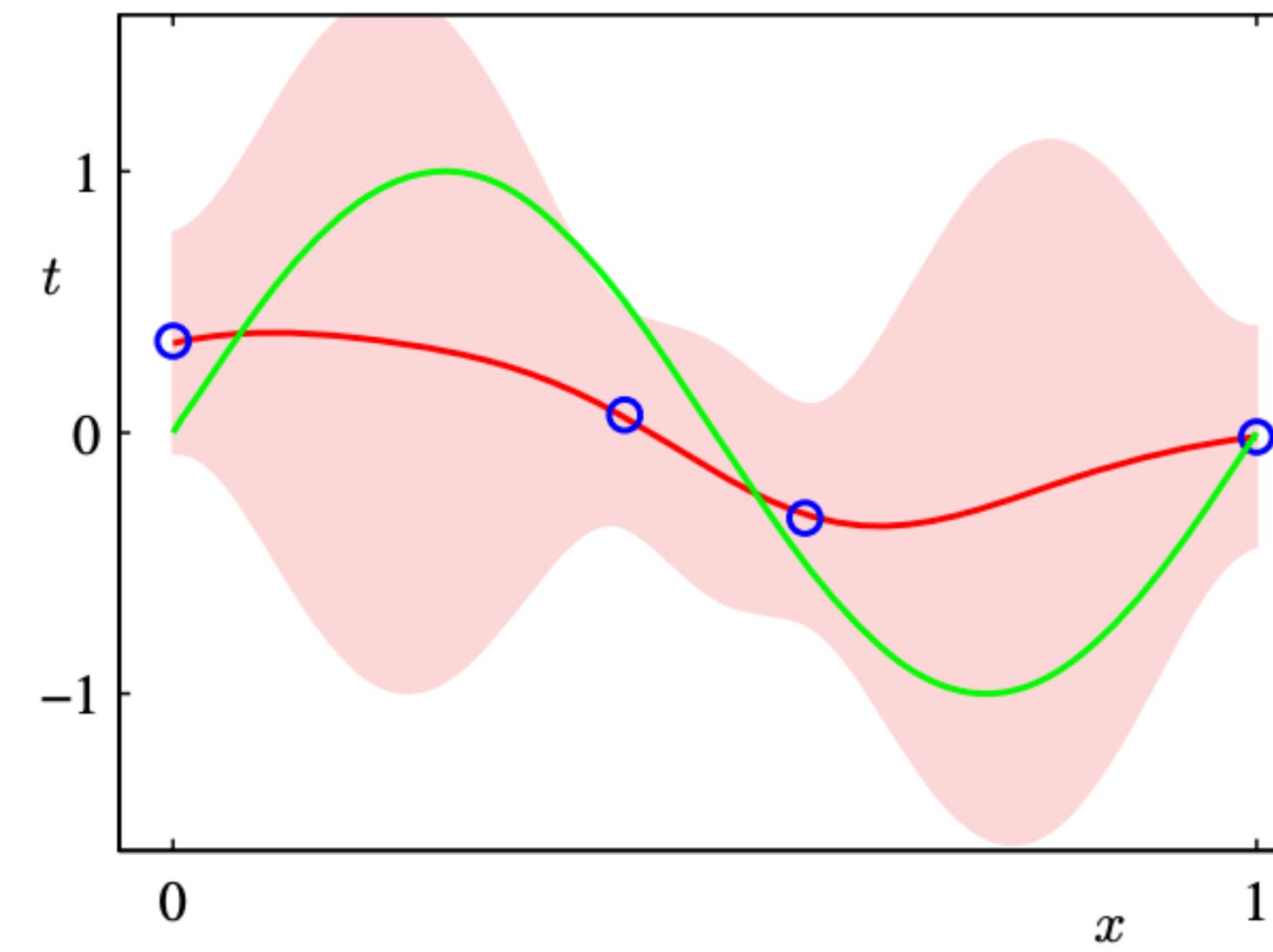
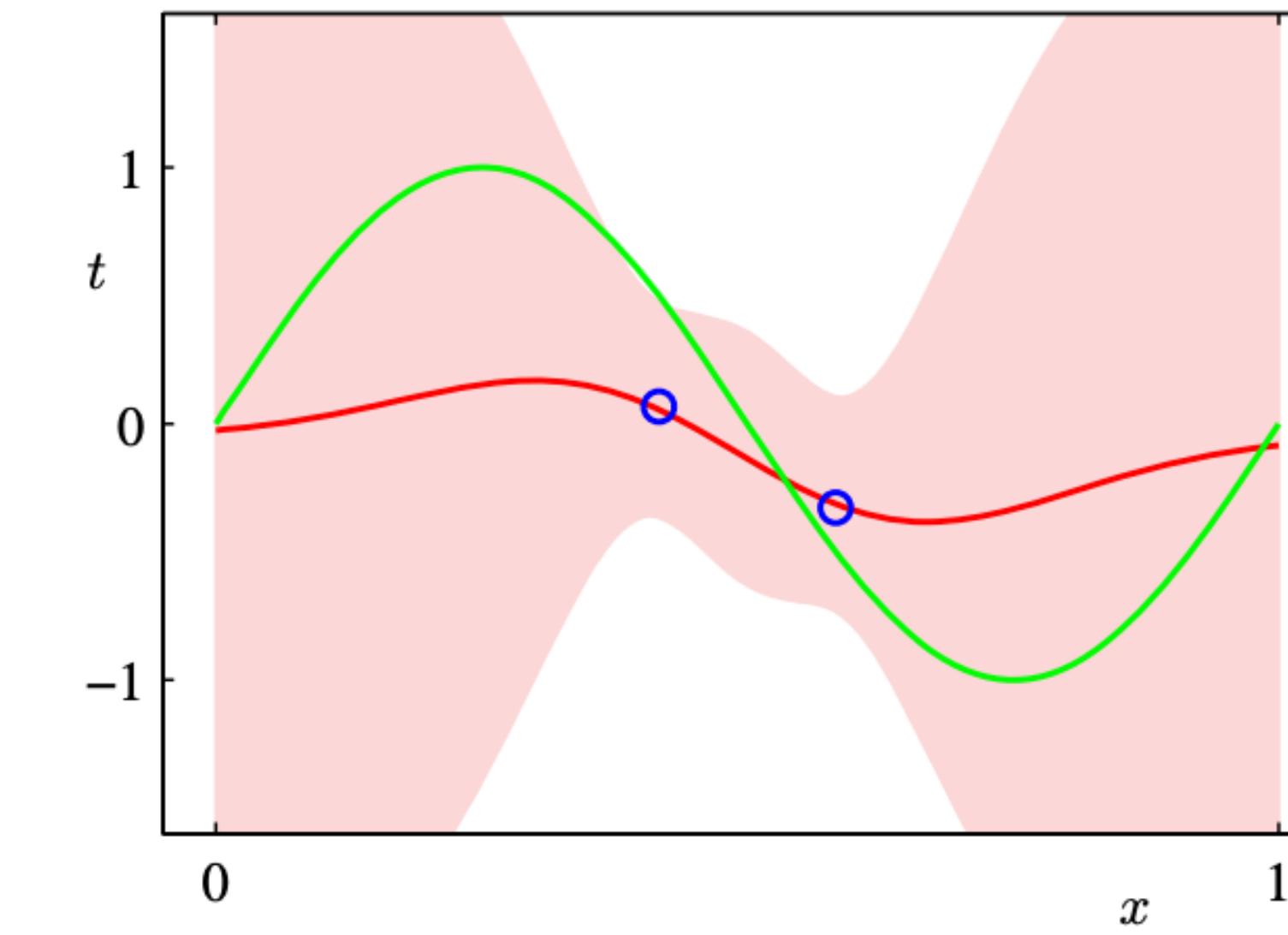
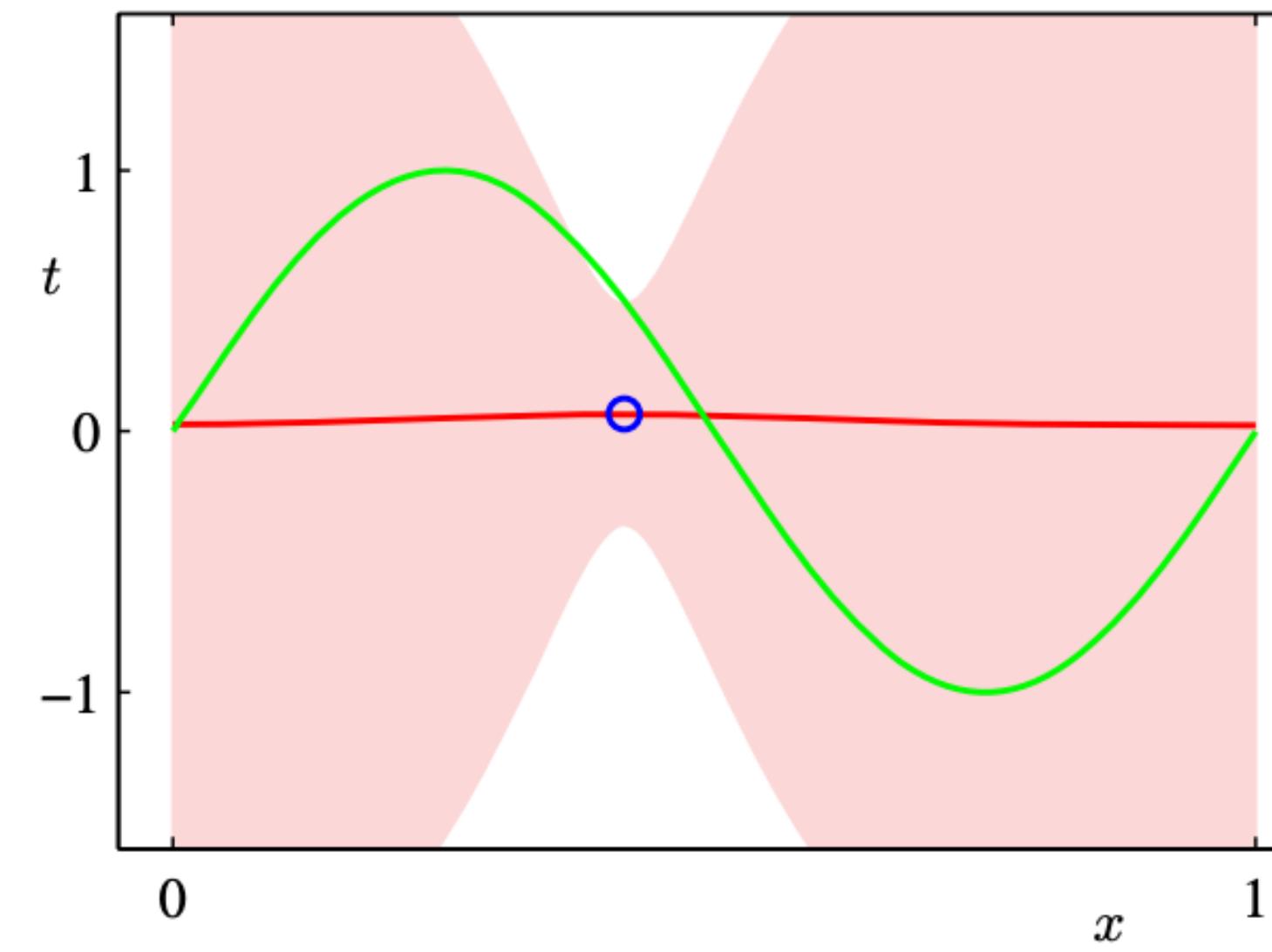


(a) Rotation

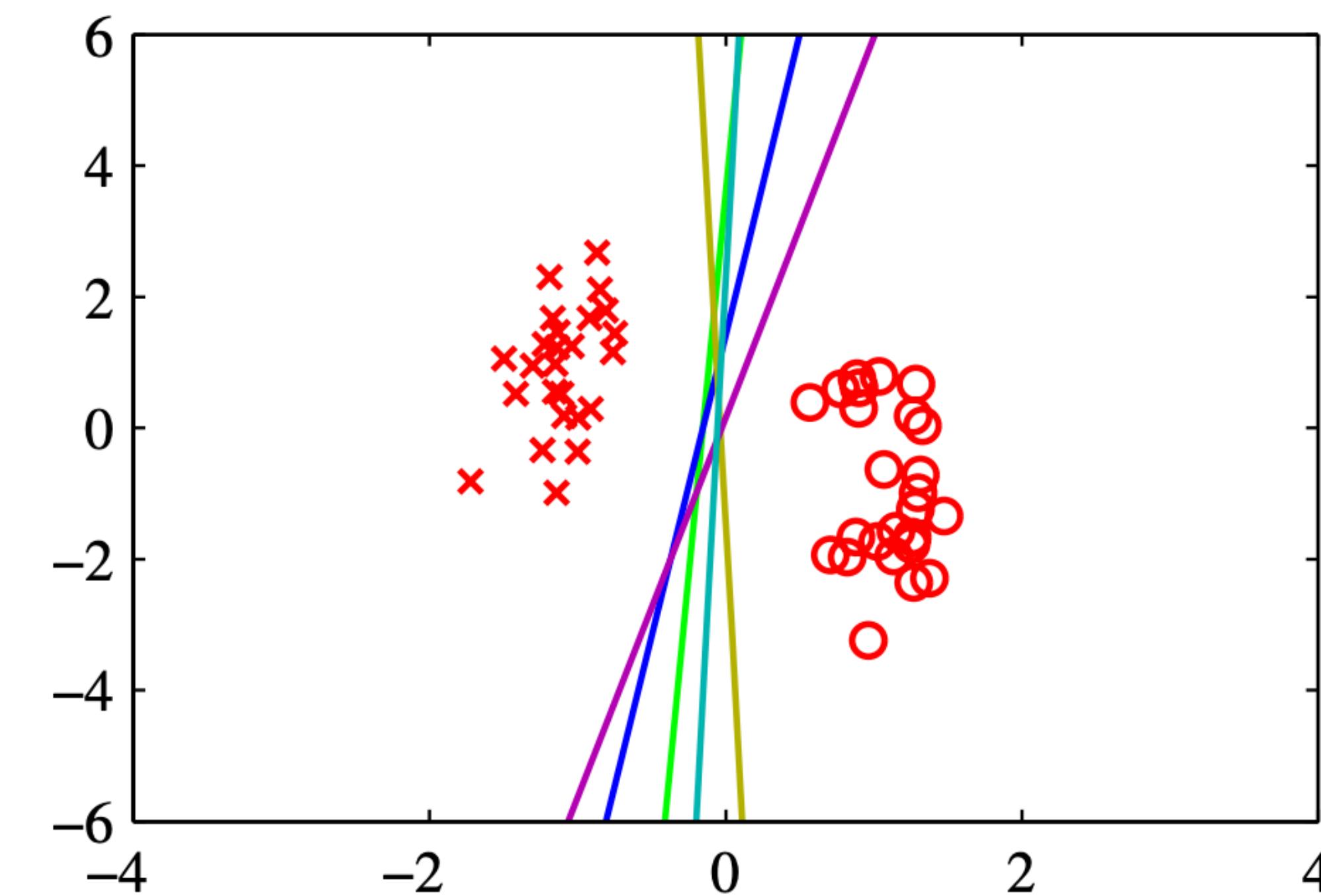
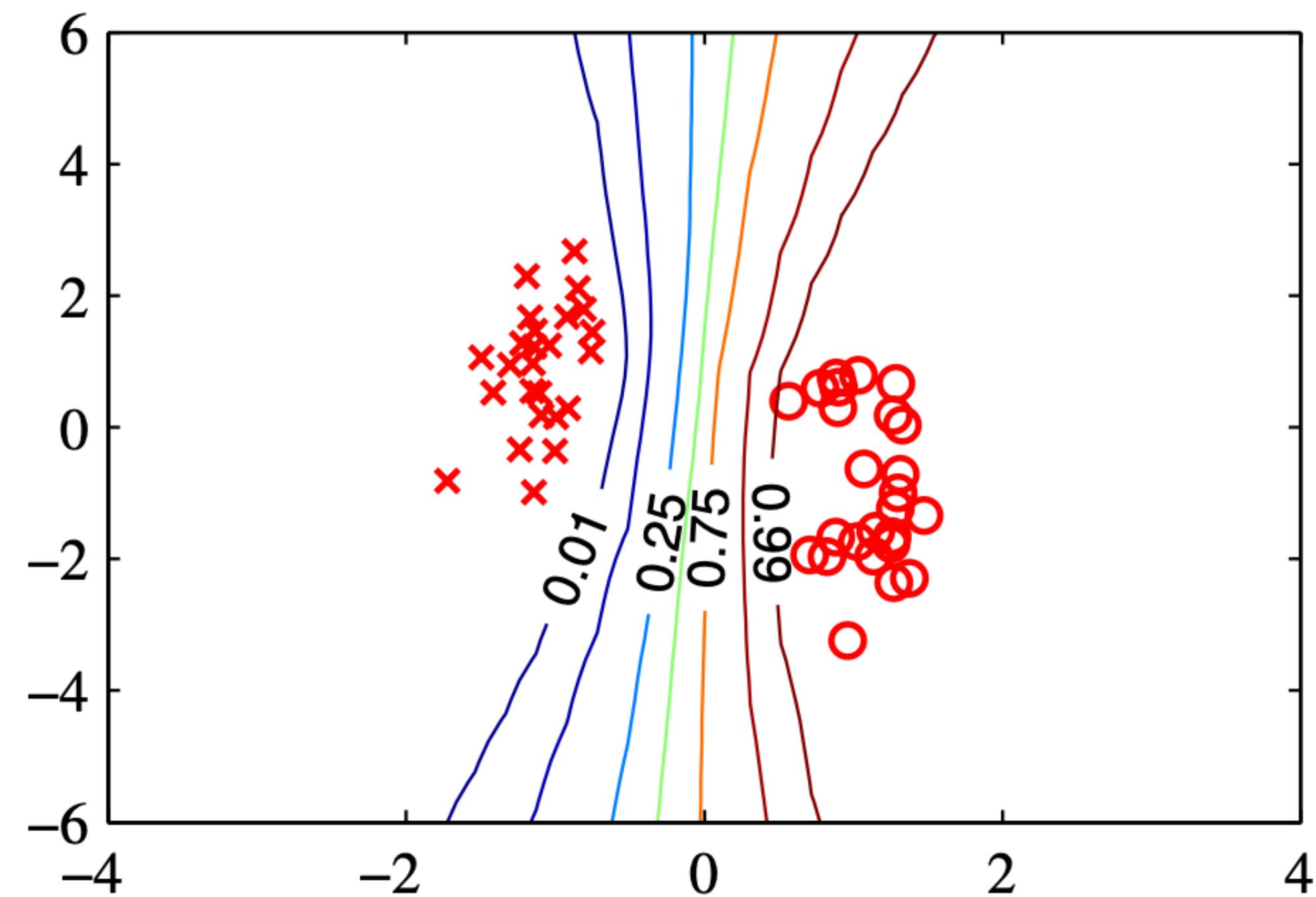
(b) Width

# Predictive Distribution

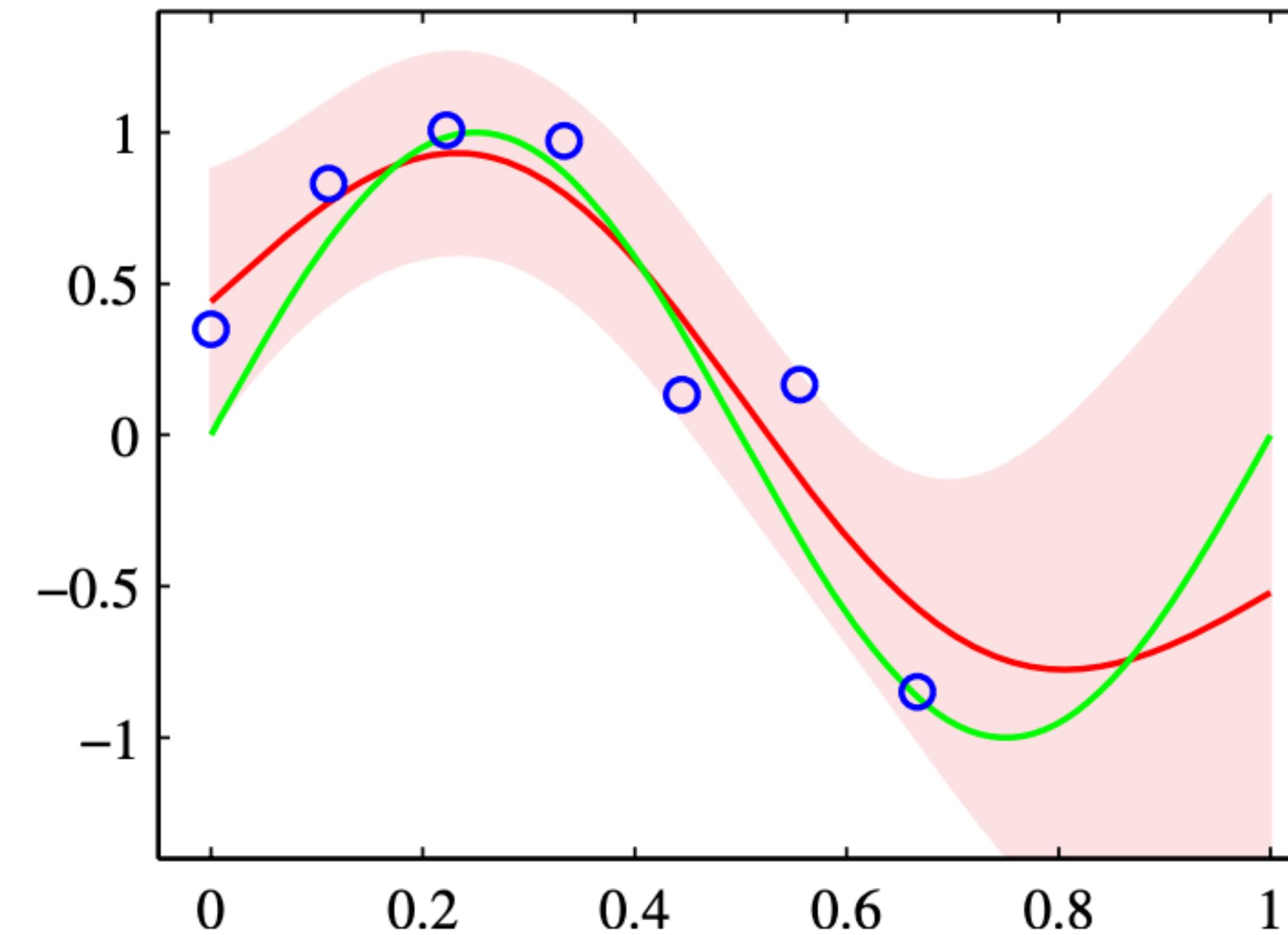
# Predictive Distribution Bayesian Linear Regression



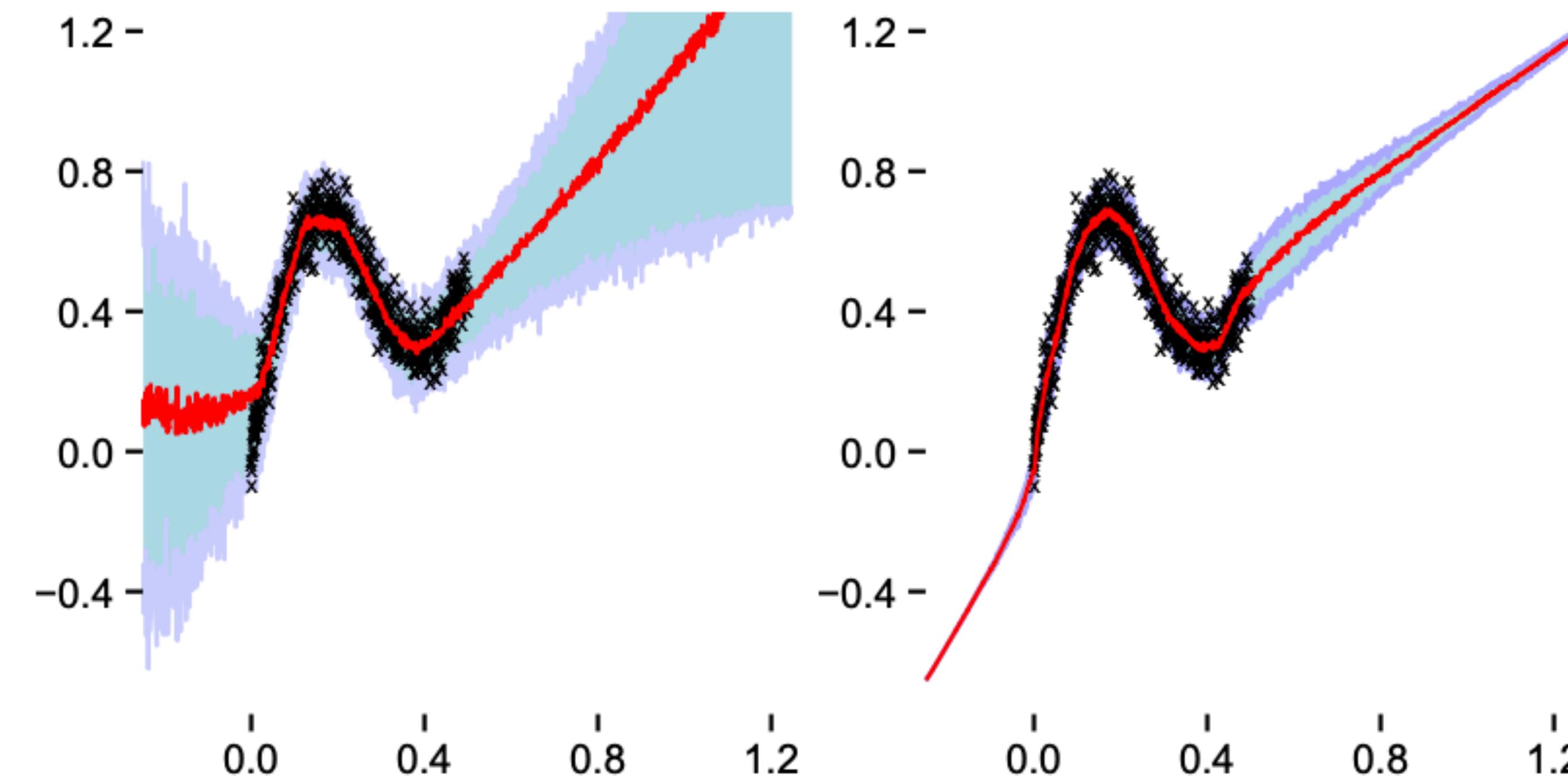
# Predictive Distribution Bayesian Logistic Regression



# Predictive Distribution Gaussian Process



# Predictive Distribution Bayes By Backdrop.



*Figure 5.* Regression of noisy data with interquartile ranges. Black crosses are training samples. Red lines are median predictions. Blue/purple region is interquartile range. Left: Bayes by Back-prop neural network, Right: standard neural network.

# Predictive Distribution

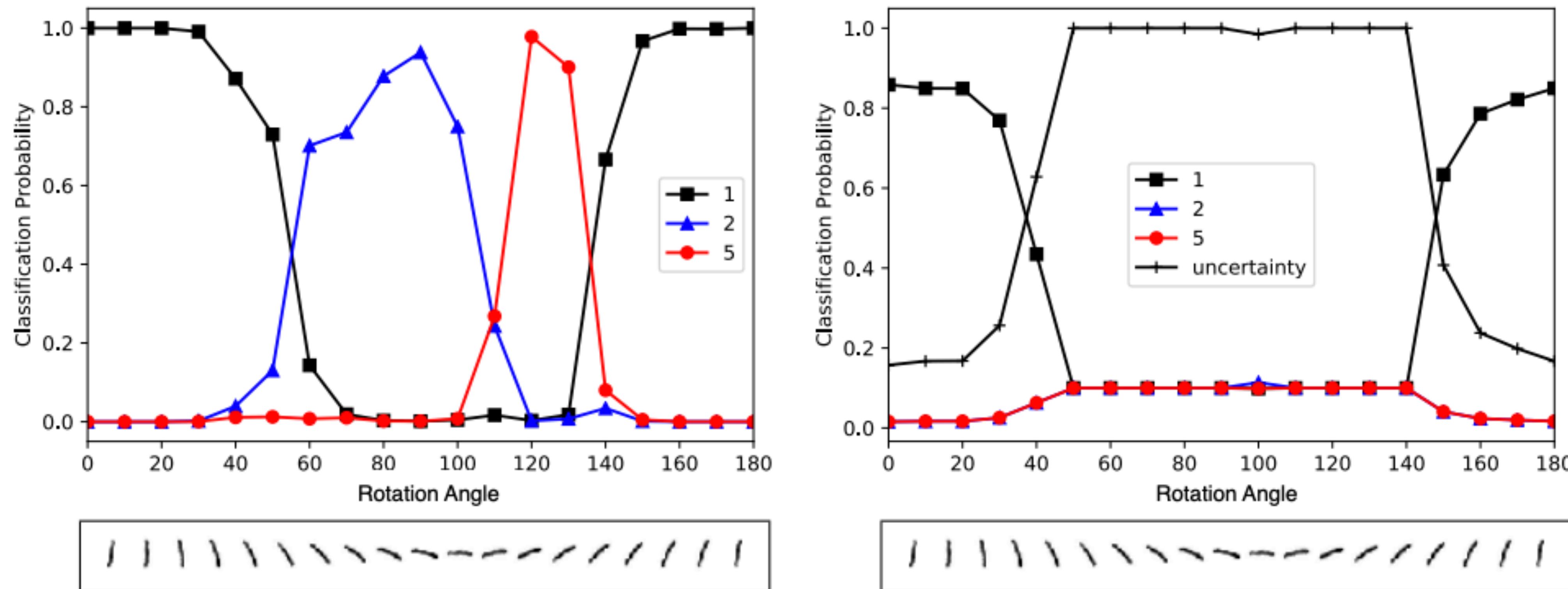
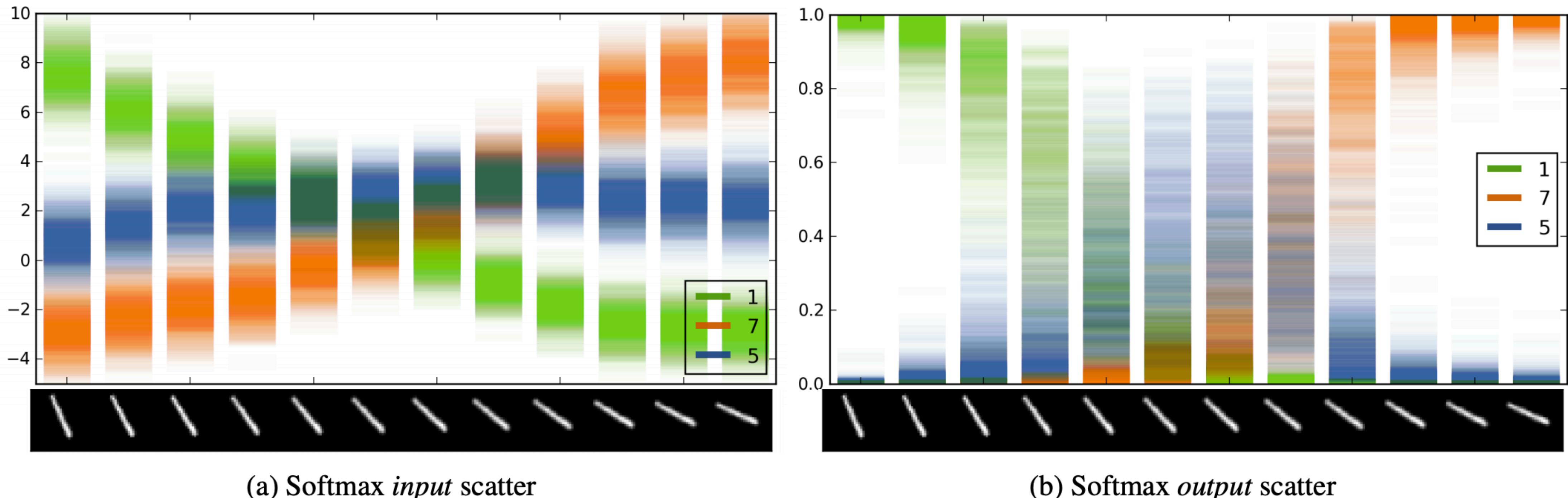


Figure 1: Classification of the rotated digit 1 (at bottom) at different angles between 0 and 180 degrees. **Left:** The classification probability is calculated using the *softmax* function. **Right:** The classification probability and uncertainty are calculated using the proposed method.

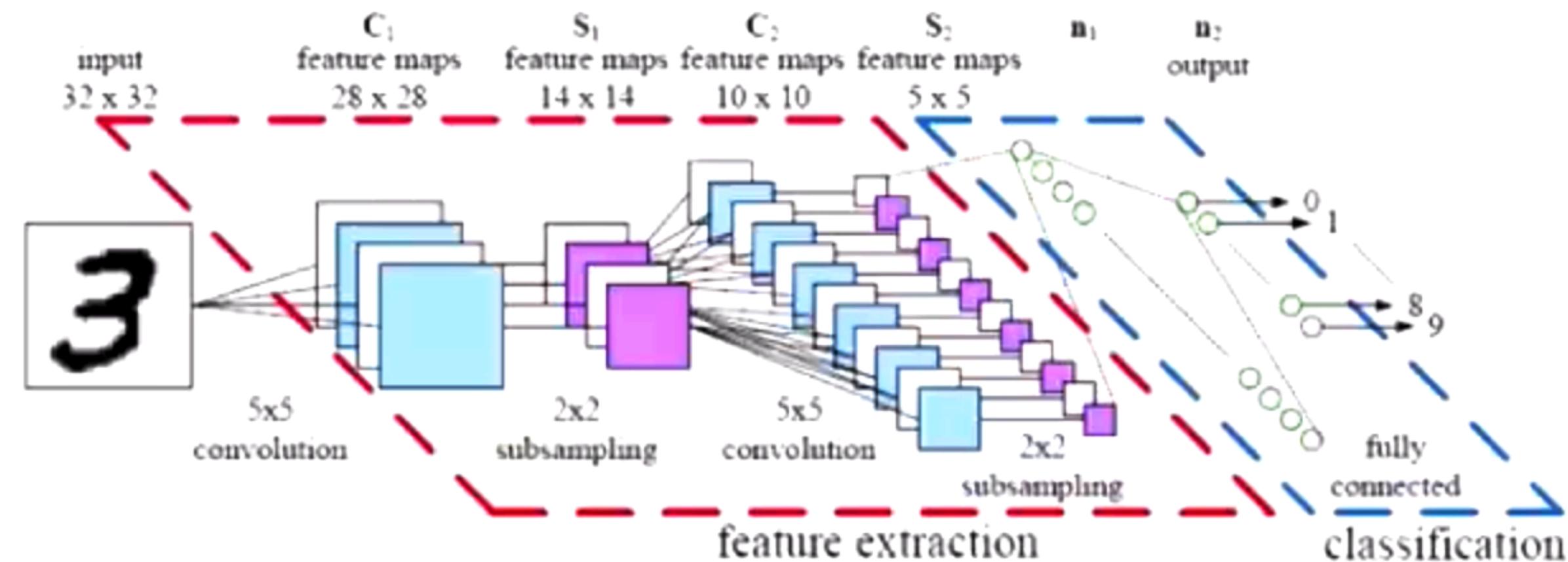
# Predictive Distribution



**Figure 4. A scatter of 100 forward passes of the softmax input and output for dropout LeNet.** On the  $X$  axis is a rotated image of the digit 1. The input is classified as digit 5 for images 6-7, even though model uncertainty is extremely large (best viewed in colour).

# Bayesian Deep-Learning

# Pros and Cons of Deep Learning



A framework for constructing flexible **models**

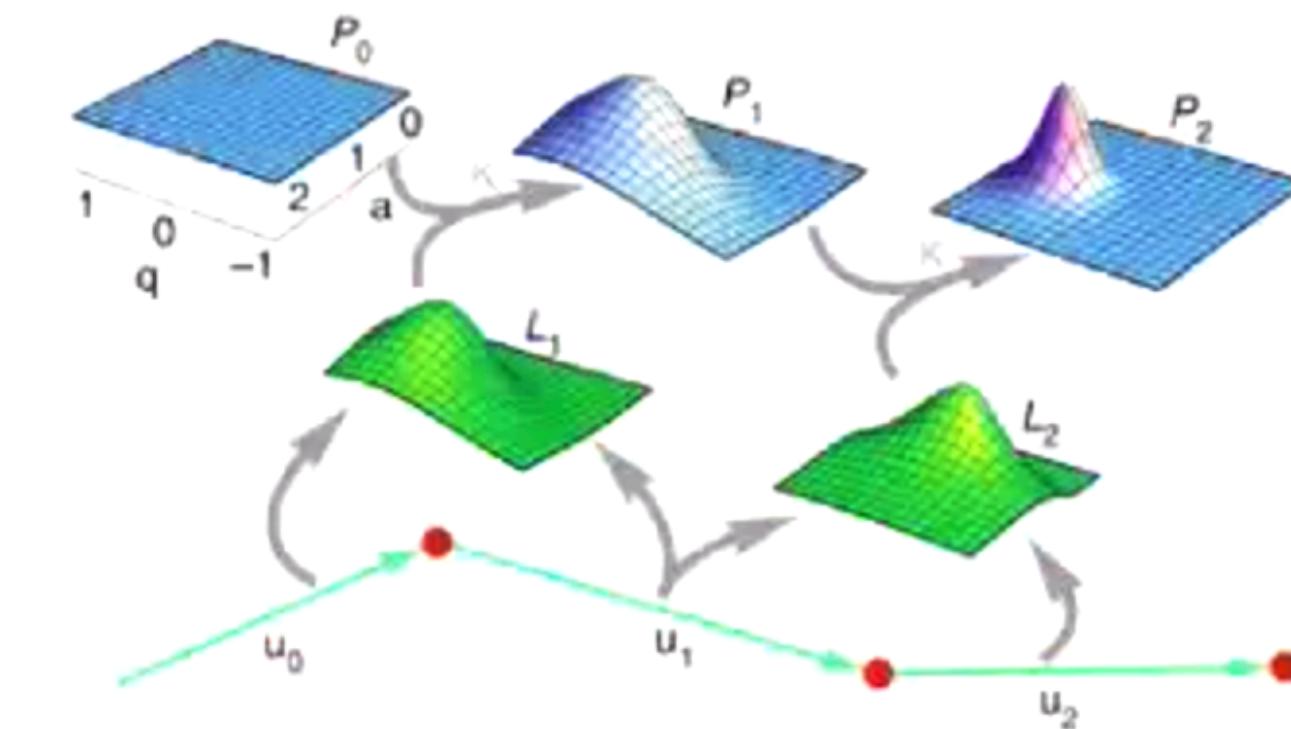
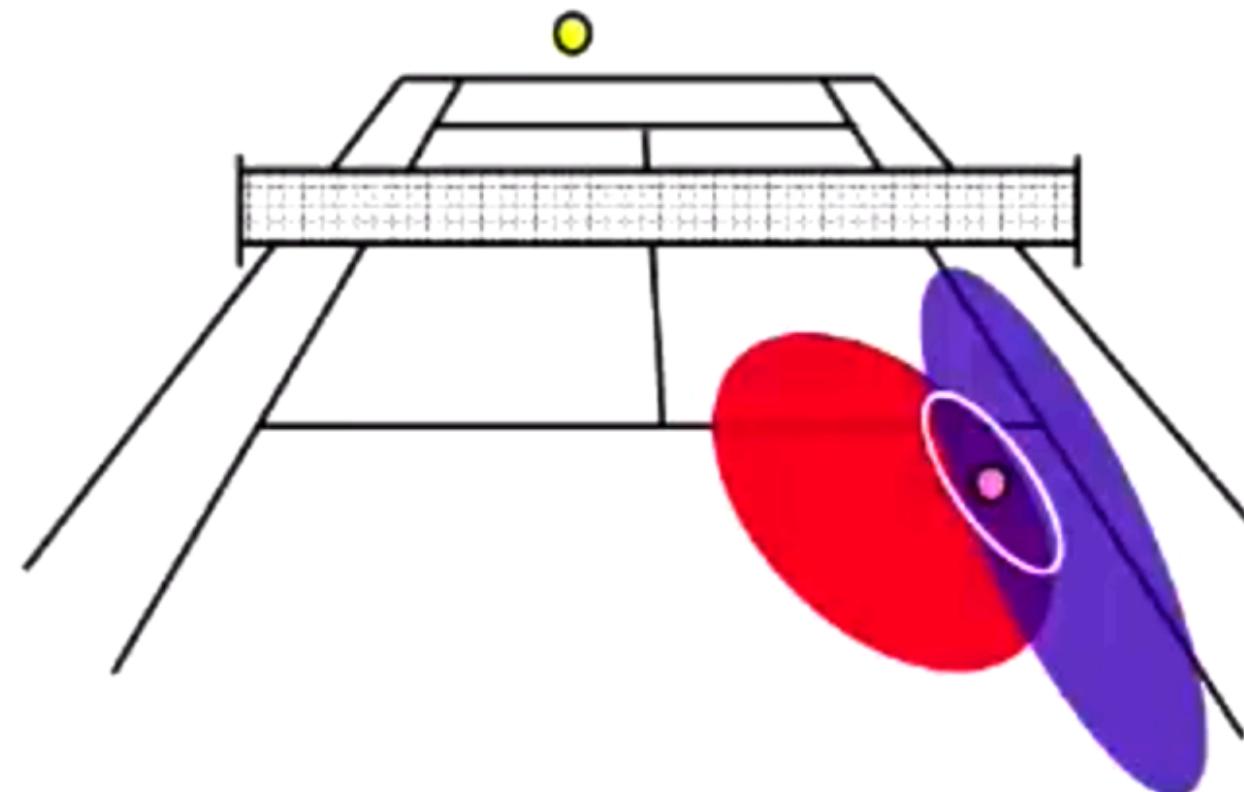
- + Rich non-linear models for classification and sequence prediction.
- + Scalable learning using stochastic approximations and conceptually simple.
- + Easily composable with other gradient-based methods
- Only point estimates
- Hard to score models, do model selection and complexity penalisation.

# Limitation of Deep Learning

Neural networks and deep learning systems give amazing performance on many benchmark tasks, but they are generally:

- ▶ very **data hungry** (e.g. often millions of examples)
- ▶ very **compute-intensive** to train and deploy (cloud GPU resources)
- ▶ poor at representing **uncertainty**
- ▶ **easily fooled** by adversarial examples
- ▶ **finicky to optimise**: non-convex + choice of architecture, learning procedure, initialisation, etc, require expert knowledge and experimentation
- ▶ uninterpretable **black-boxes**, lacking in transparency, difficult to trust

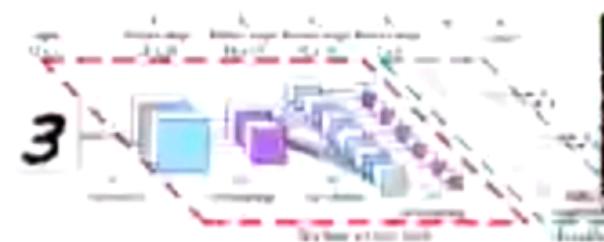
# Pros and Cons of Bayesian Reasoning



A framework for **inference and decision making**

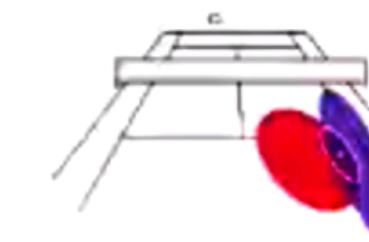
- + Unified framework for model building, inference, prediction and decision making
- + Explicit accounting for uncertainty and variability of outcomes
- + Robust to overfitting; tools for model selection and composition.
- Mainly conjugate and linear models
- Intractable inference leading to expensive computation or long simulation times.

# Bayesian Deep Learning



## Deep Learning

- + Rich non-linear models for classification and sequence prediction.
- + Scalable learning using stochastic approximation and conceptually simple.
- + Easily composable with other gradient-based methods
- Only point estimates
- Hard to score models, do selection and complexity penalisation.

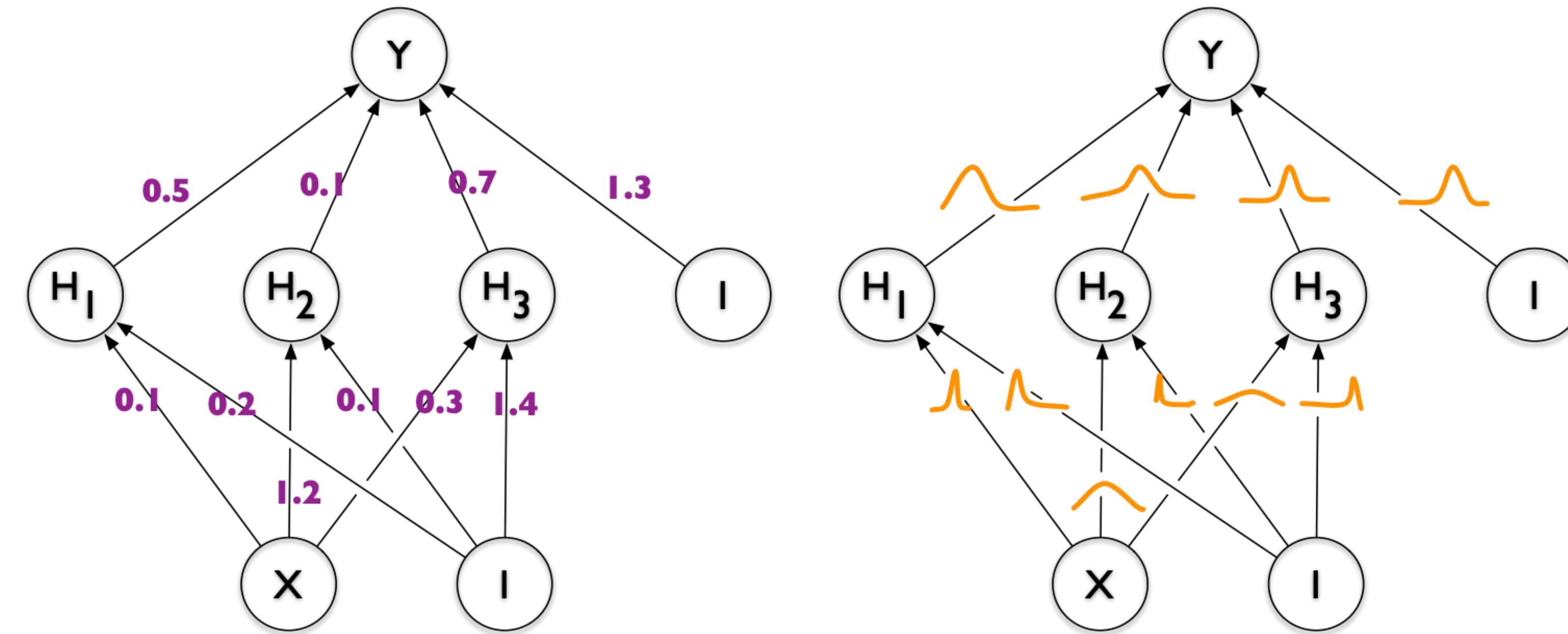


## Bayesian Reasoning

- Many conjugate and linear models
- Potentially intractable inference, computationally expensive or long simulation time.
- + Unified framework for model building, inference, prediction and decision making
- + Explicit accounting for uncertainty and variability of outcomes
- + Robust to overfitting; tools for model selection and composition.

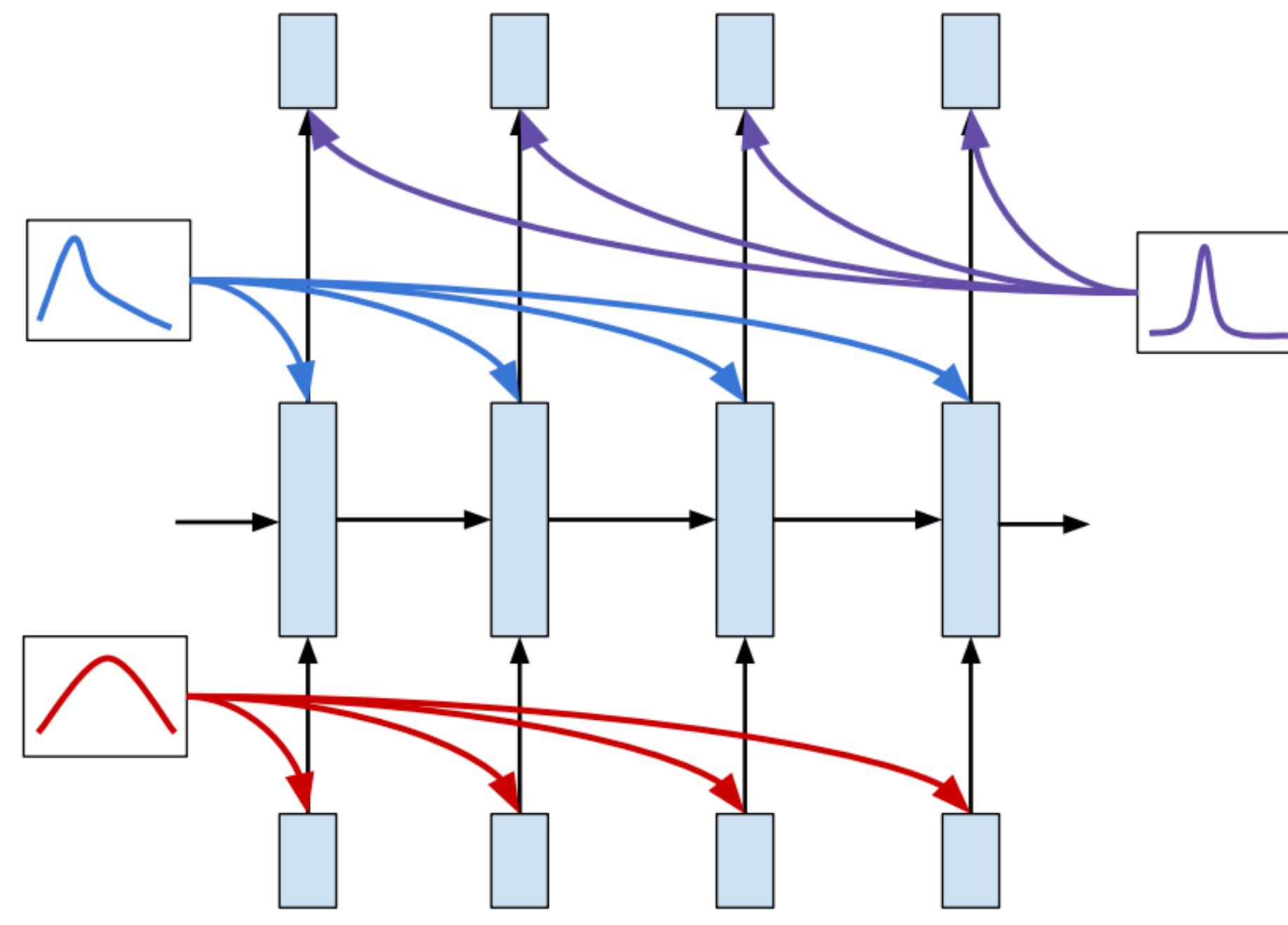
*Natural to marry these approaches.*

# Bayesian Deep Learning



*Figure 1.* Left: each weight has a fixed value, as provided by classical backpropagation. Right: each weight is assigned a distribution, as provided by Bayes by Backprop.

# Bayesian RNN



## Algorithm: Bayes by Backprop for RNNs

Sample  $\epsilon \sim \mathcal{N}(0, I)$ ,  $\epsilon \in \mathbb{R}^d$ , and set network parameters to  $\theta = \mu + \sigma\epsilon$ .

Sample a minibatch of truncated sequences  $(x, y)$ . Do forward and backward propagation as normal, and let  $g$  be the gradient w.r.t  $\theta$ .

Let  $g_\theta^{KL}, g_\mu^{KL}, g_\sigma^{KL}$  be the gradients of  $\log \mathcal{N}(\theta | \mu, \sigma^2) - \log p(\theta)$  w.r.t.  $\theta, \mu$  and  $\sigma$  respectively.

Update  $\mu$  using the gradient  $\frac{g + \frac{1}{C}g_\theta^{KL}}{B} + \frac{g_\mu^{KL}}{BC}$ .

Update  $\sigma$  using the gradient  $\left( \frac{g + \frac{1}{C}g_\theta^{KL}}{B} \right) \epsilon + \frac{g_\sigma^{KL}}{BC}$ .

Figure 1: Illustration (left) and Algorithm (right) of Bayes by Backprop applied to an RNN.

# Bayesian RNN

Table 1: Word-level perplexity on the Penn Treebank language modelling task (lower is better), where DE indicates that Dynamic Evaluation was used.

Model (medium)	Val	Test	Val (DE)	Test (DE)
LSTM (Zaremba et al., 2014)	120.7	114.5	-	-
LSTM dropout (Zaremba et al., 2014)	86.2	82.1	79.7	77.1
Variational LSTM (tied weights) (Gal & Ghahramani, 2016)	81.8	79.7	-	-
Variational LSTM (tied weights, MS) (Gal & Ghahramani, 2016)	-	79.0	-	-
Bayesian RNN (BRNN)	78.8	75.5	73.4	70.7
BRNN w/ Posterior Sharpening	$\leq 77.8$	$\leq 74.8$	$\leq 72.6$	$\leq 69.8$

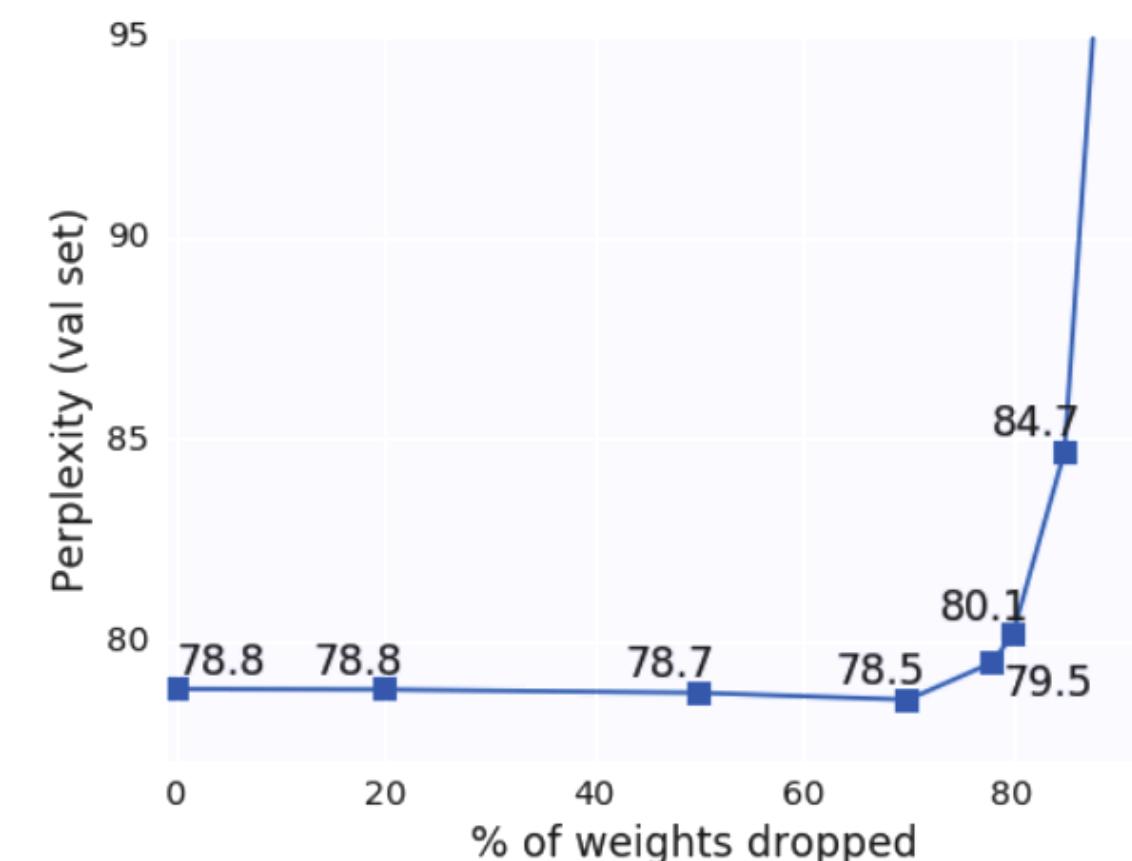


Figure 2: Weight pruning experiment. No significant loss on performance is observed until pruning more than 80% of weights.

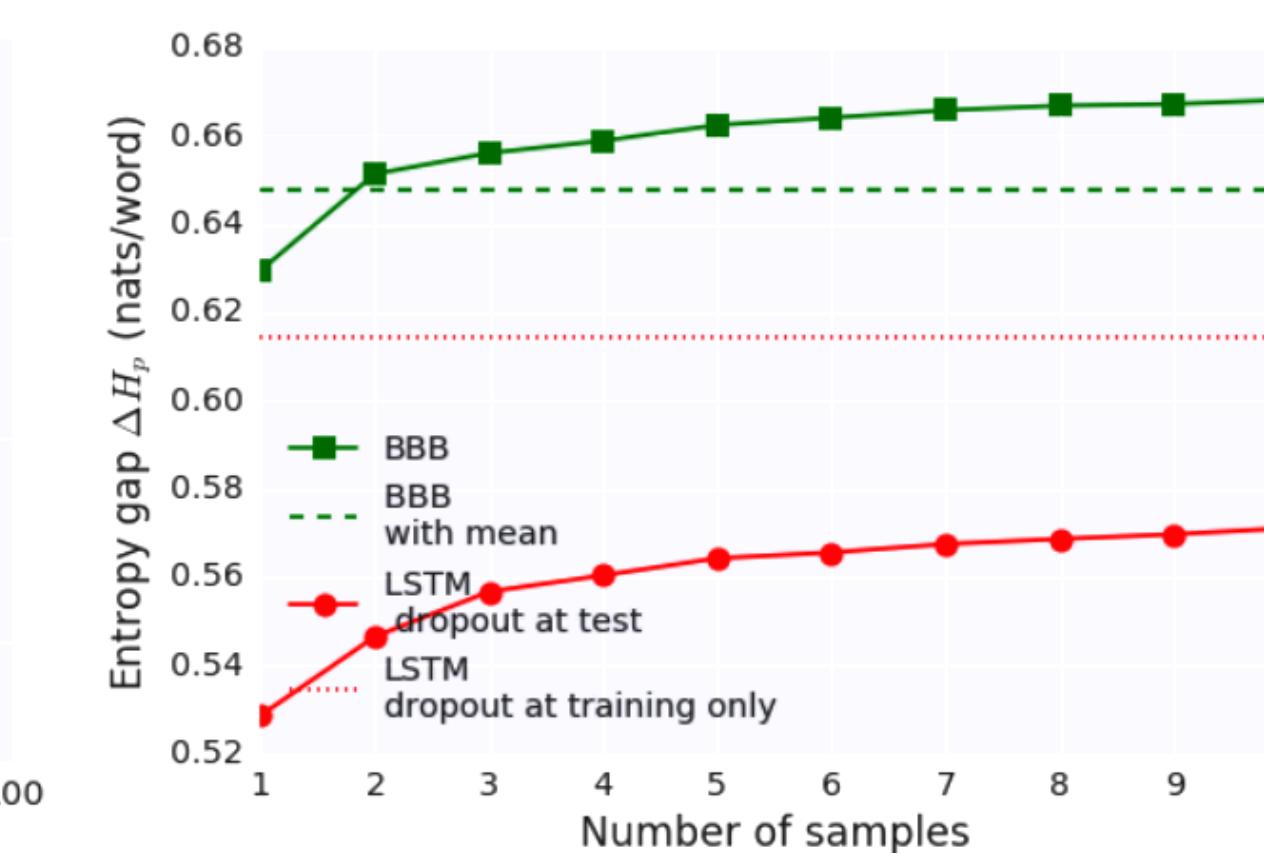


Figure 3: Entropy gap  $\Delta H_p$  (Eq. (20)) between reversed and regular Penn Treebank test sets × number of samples.

# Bayesian RNN

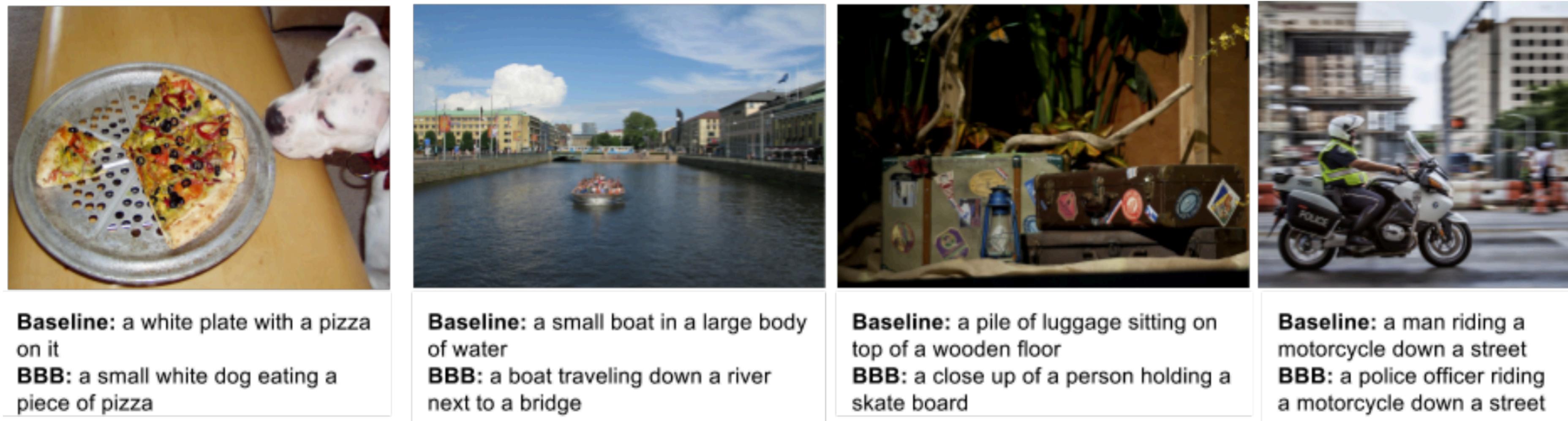


Figure 4: Image captioning results on MSCOCO development set.

We used the MSCOCO (Lin et al., 2014) data set and report perplexity, BLUE-4, and CIDEr scores on compared to the Show and Tell model (Vinyals et al., 2016), which was the winning entry of the captioning challenge in 2015<sup>3</sup>. The results are:

Model	Perplexity	BLUE-4	CIDEr
Show and Tell	8.3	28.8	89.8
Bayes RNN	<b>8.1</b>	<b>30.2</b>	<b>96.0</b>

# Bayesian CNN

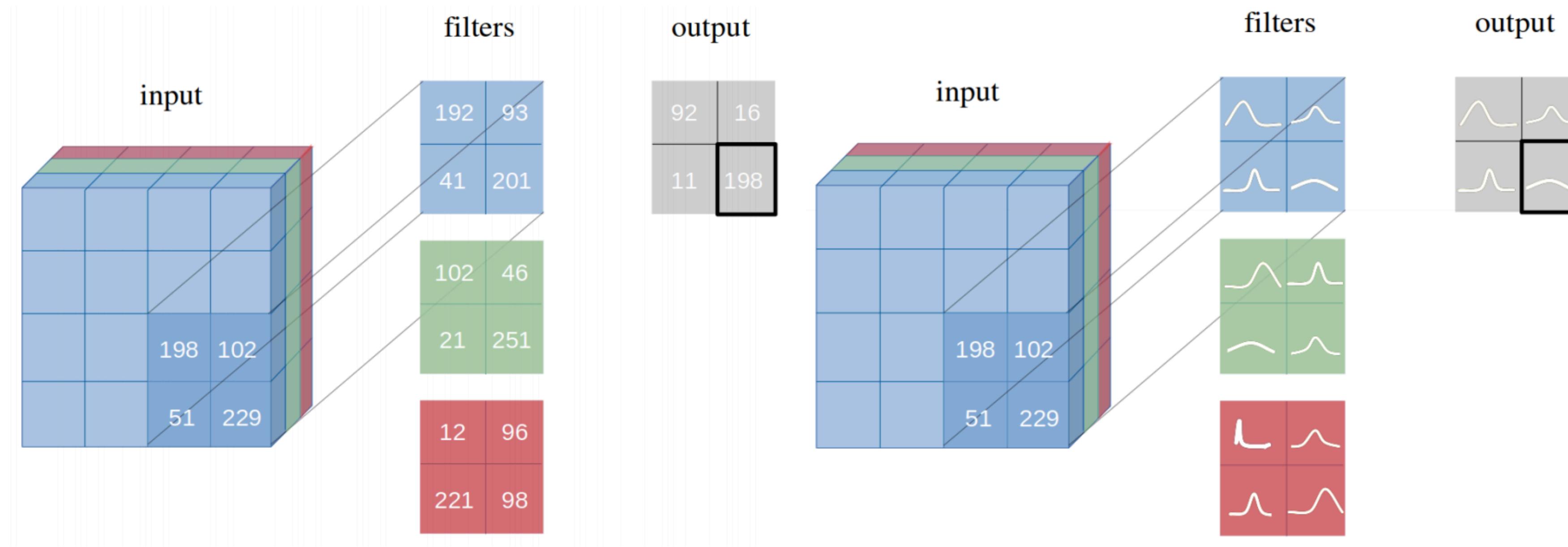


Figure 4: Input image with exemplary pixel values, filters, and corresponding output with point-estimates (top) and probability distributions (bottom) over weights.[55]

# Bayesian CNN

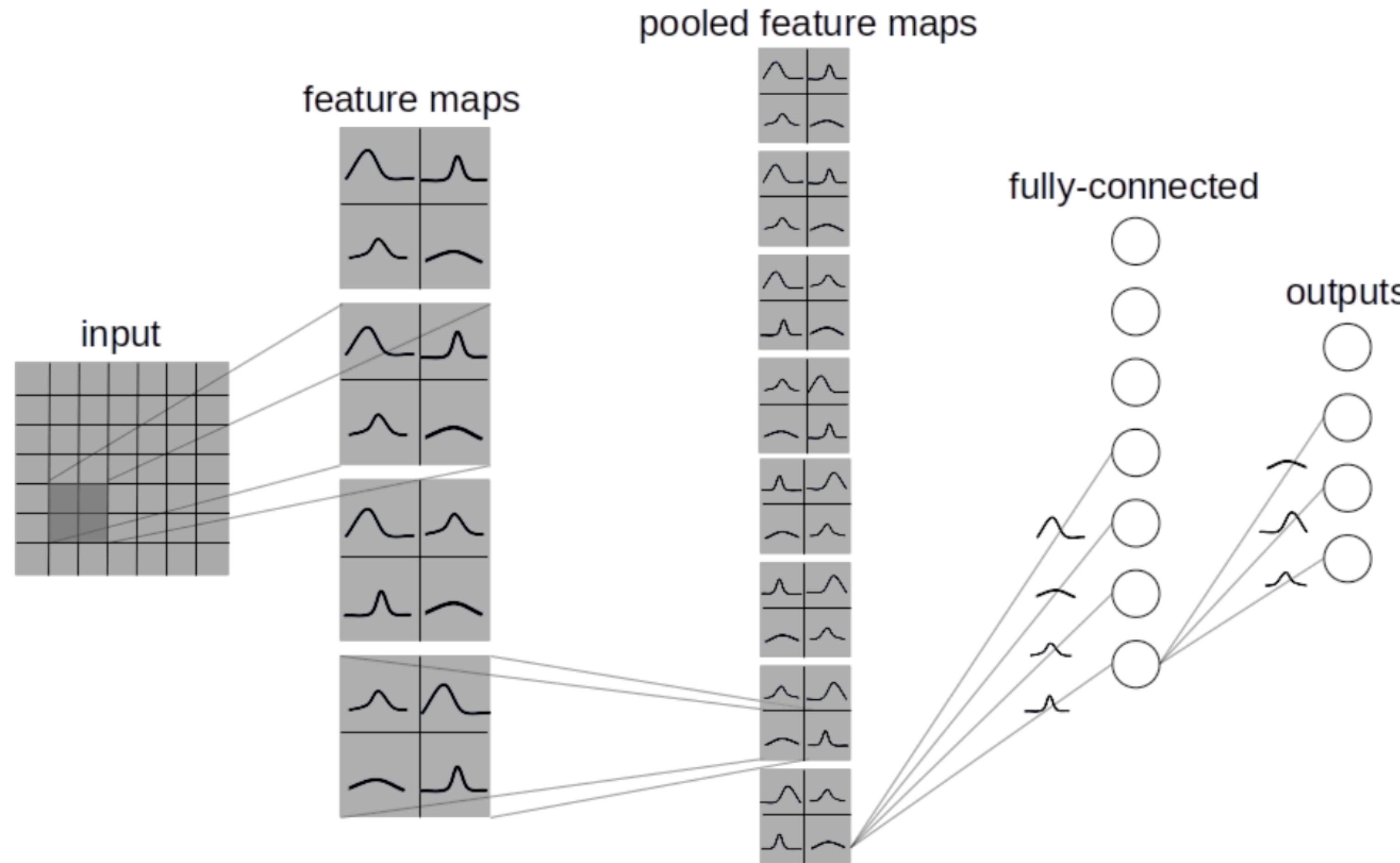


Figure 5: Fully Bayesian perspective of an exemplary CNN. Weights in filters of convolutional layers, and weights in fully-connected layers have the form of a probability distribution. [55]

# Bayesian CNN

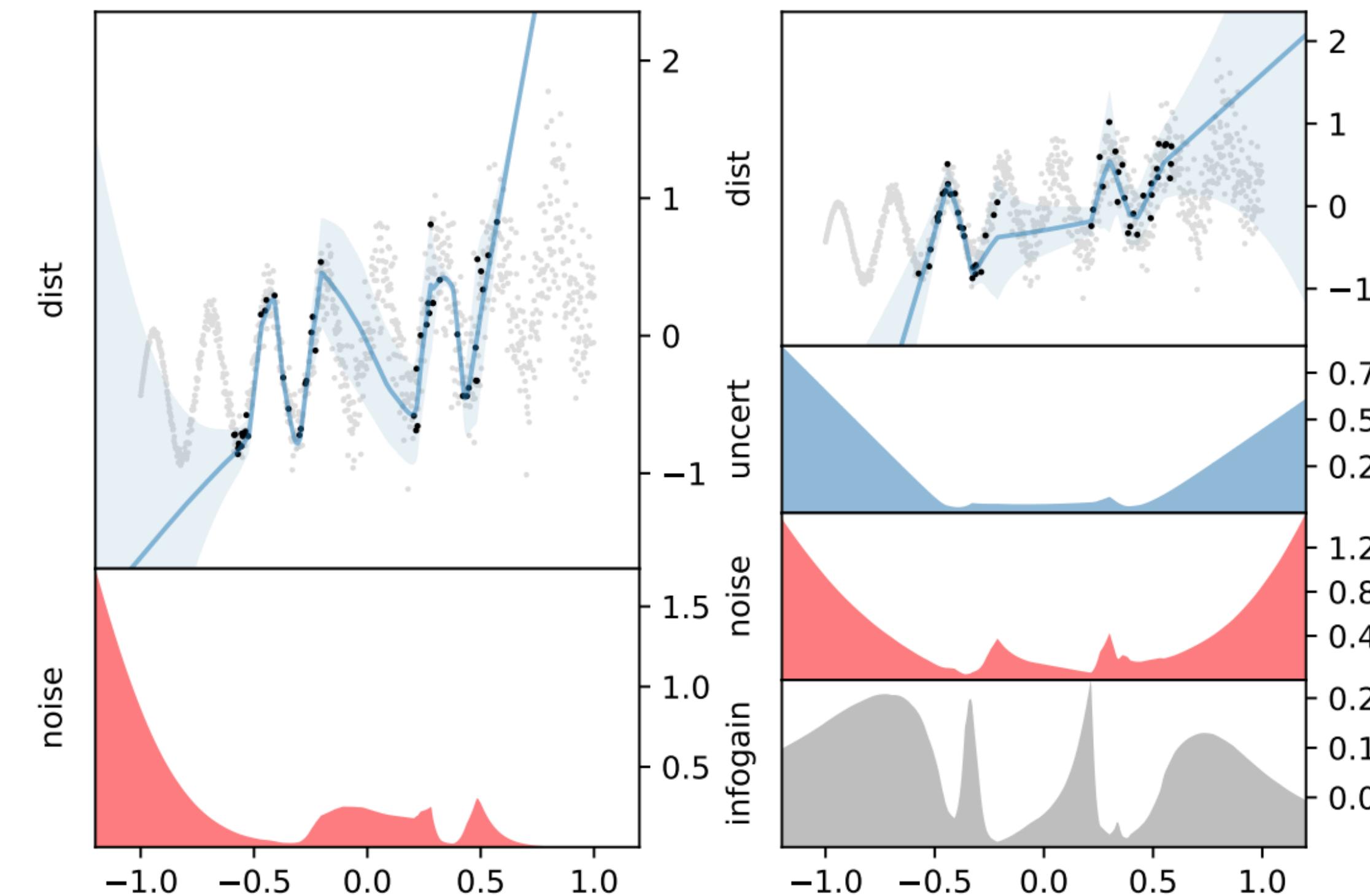


Figure 6: Predictive distributions is estimated for a low-dimensional active learning task. The predictive distributions are visualized as mean and two standard deviations shaded. ■ shows the epistemic uncertainty and ■ shows the aleatoric noise. Data points are shown in ■. **(Left)** A deterministic network conflates uncertainty as part of the noise and is overconfident outside of the data distribution. **(Right)** A variational Bayesian neural network with standard normal prior represents uncertainty and noise separately but is overconfident outside of the training distribution as defined by [22]

# Bayesian CNN

- BayesCNN for Image Super Resolution



Figure 12: Sample image in Low Resolution image space taken randomly from BSD 300 [46] dataset.



Figure 13: Generated Super Resolution Image scaled to 40 percent to fit

# When is the probabilistic approach essential?

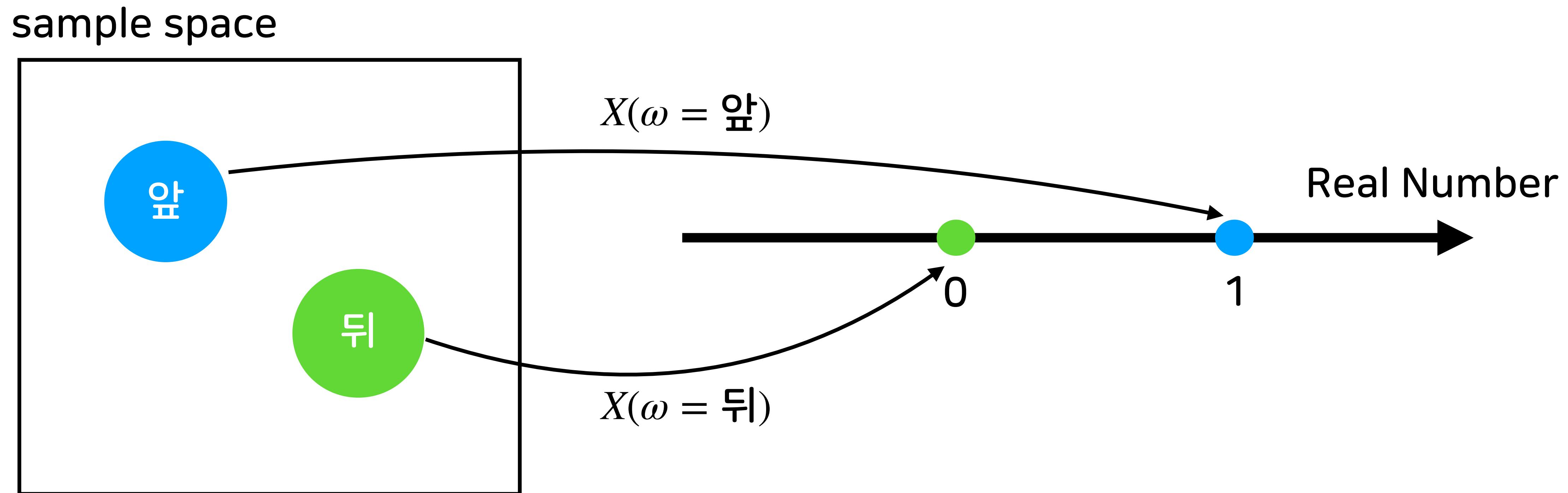
Many aspects of learning and intelligence depend crucially on the careful probabilistic representation of *uncertainty*:

- ▶ Forecasting
- ▶ Decision making
- ▶ Learning from limited, noisy, and missing data
- ▶ Learning complex personalised models
- ▶ Data compression
- ▶ Automating scientific modelling, discovery, and experiment design

# Random Variables & Probability Distributions

# Random Variable

- 확률 변수 (Random Variable)는 무작위적으로 다른 값을 가질 수 있는 변수를 나타냅니다.
- 좀 더 엄밀하게는 sample space 내의 예측할 수 없는 각 사건들을 실수값에 대응시키는 함수로 생각할 수 있습니다.



# Probability Distribution

- 확률 분포 (probability distribution)는 random variable이 가질 수 있는 값들의 가능성을 나타냅니다.
- Probability distribution은 discrete random variable에 대한 Probability Mass Function (PMF)와 continuous random variable에 대한 Probability Density Function(PDF) 두 종류가 있습니다.

# Bernoulli-정의

- 베르누이 분포 (Bernoulli distribution)은 0 또는 1 두가지 값을 가지는 random variable의 확률 분포입니다.
- Random variable이 1 값을 가질 확률을 나타내는  $\mu$ 를 parameter로 가집니다.
- Bernoulli distribution의 PMF는 다음과 같은 식으로 표현됩니다.
  - $$p(x|\mu) = Ber(x|\mu) = \begin{cases} \mu, & \text{if } x = 1 \\ 1 - \mu, & \text{if } x = 0 \end{cases}$$
  - 또는 다음과 같이 표현 할 수도 있습니다.

$$p(x|\mu) = Ber(x|\mu) = \mu^x(1 - \mu)^{(1-x)}$$

# Bernoulli-최적화

- Dataset  $X = \{x_1, x_2, \dots, x_N\}$ 이 주어진 경우, likelihood function을 다음과 같이 표현할 수 있습니다.

$$p(X|\mu) = \prod_{n=1}^N p(x_n|\mu) = \prod_{n=1}^N \mu^{x_n}(1-\mu)^{(1-x_n)}$$

- 빈도주의적 관점에서 위 식을 최대화 하는 parameter인  $\mu$ 를 구할 수 있습니다. 또는 단조 증가 함수인 log함수를 likelihood에 적용하여 최대화 할 수도 있습니다.

$$\log p(X|\mu) = \sum_{n=1}^N \log p(x_n|\mu) = \sum_{n=1}^N \{x_n \log \mu + (1-x_n) \log (1-\mu)\}$$

# Bernoulli-최적화

- Data의 likelihood 또는 log likelihood를 최대화하는 최적화 방법을 Maximum Likelihood라고 합니다.
- Maximum Likelihood를 통해 얻은 파라메터  $\mu$ 의 값은 다음과 같습니다.
- 
- 이는 전체 데이터 중 1값을 갖는 데이터의 비율과 같습니다.

$$\mu_{ML} = \frac{1}{N} \sum_{n=1}^N x_n$$

# Binomial-정의

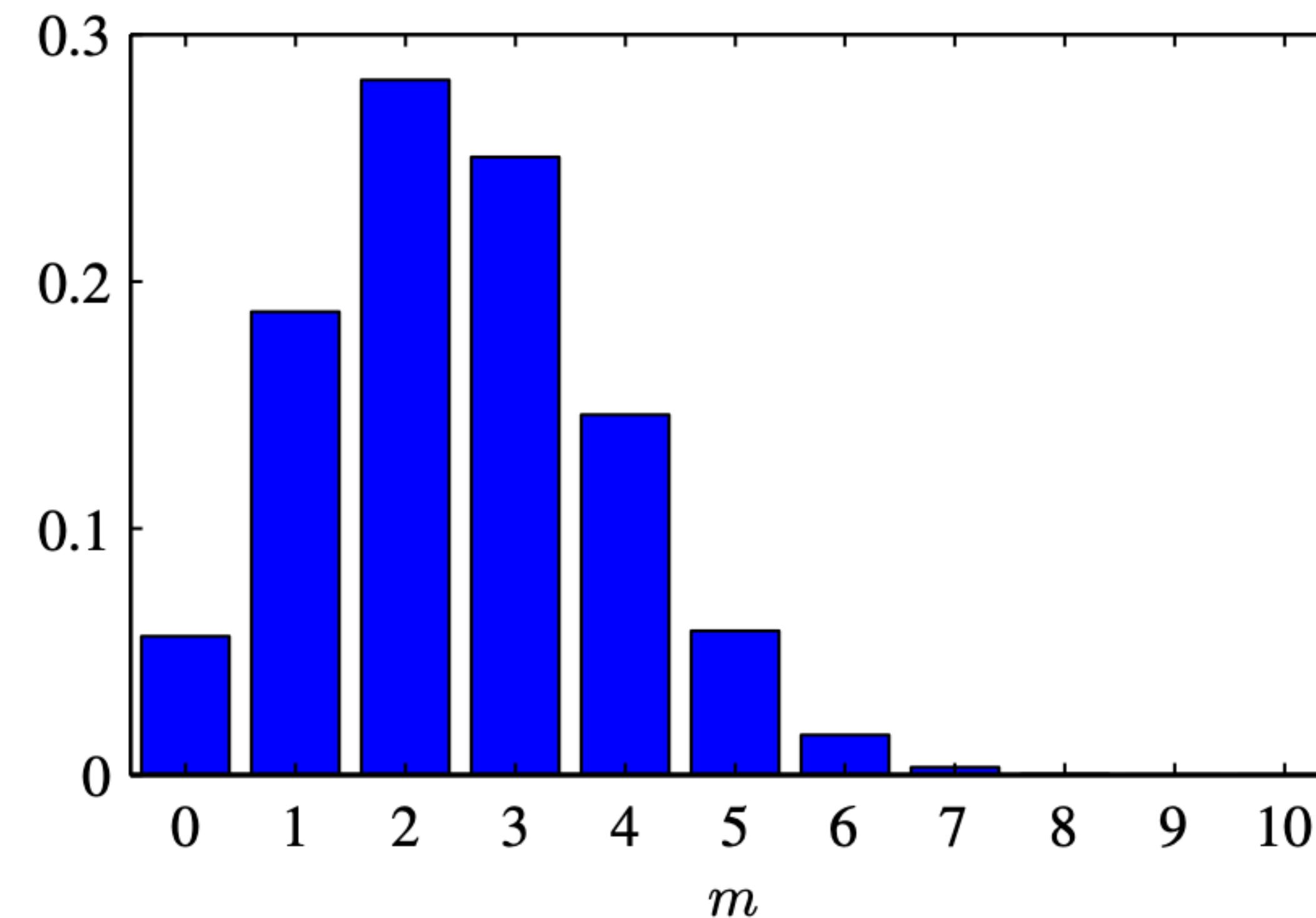
- Binomial distribution은 베르누이 시행(Bernoulli trials)을  $N$ 번 독립적으로 했을 때 얻을 수 있는 1의 갯수에 대한 확률 분포입니다.
- 총 시행 횟수  $N$ 과 1 값이 발생할 확률  $\mu$ 를 파라메터로 갖습니다.
- Binomial distribution의 PMF는 다음과 같이 정의할 수 있습니다.

$$Bin(m | N, \mu) = \binom{N}{m} \mu^m (1 - \mu)^{N-m}$$

$$\text{where } \binom{N}{m} \equiv \frac{N!}{(N - m)!m!}$$

# Binomial

**Figure 2.1** Histogram plot of the binomial distribution (2.9) as a function of  $m$  for  $N = 10$  and  $\mu = 0.25$ .



# Categorical-정의

- Categorical distribution은  $K$ 개의 discrete 값을 가질 수 있는 random variable에 대한 probability distribution입니다.
- Parameter로 각 카테고리에 대한 확률 값을 나타내는 벡터  $\mu = (\mu_1, \mu_2, \dots, \mu_K)$ 를 갖습니다.
- Random variable이 갖는 값은 one-hot 벡터로 나타낼 수 있습니다. 예를 들어 6개의 카테고리가 있고 random variable이 3번째 카테고리 값을 갖는다면 다음과 같이 나타낼 수 있습니다.

$$\mathbf{x} = (0, 0, 1, 0, 0, 0)^T$$

- Categorical distribution의 PMF는 다음과 같이 표현할 수 있습니다.

$$p(\mathbf{x} | \mu) = \prod_{k=1}^K \mu_k^{x_k}$$

# Categorical-최적화

- Dataset  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ 이 주어진 경우, likelihood function을 다음과 같이 표현할 수 있습니다.

- 

$$p(\mathbf{X} | \boldsymbol{\mu}) = \prod_{n=1}^N \prod_{k=1}^K \mu_k^{x_{nk}} = \prod_{k=1}^K \mu_k^{(\sum_n x_{nk})} = \prod_{k=1}^K \mu_k^{N_k}$$

- 빈도주의적 관점에서 위 식을 최대화 하는 파라메터인  $\boldsymbol{\mu}$ 를 구할 수 있습니다. 또는 단조 증가 함수인 log함수를 likelihood에 적용하여 최대화 할 수도 있습니다.

- 

$$\log p(\mathbf{X} | \boldsymbol{\mu}) = \sum_{k=1}^K N_k \log \mu_k$$

# Categorical-최적화

- 단,  $\mu_k$ 값들의 합이 1이 되어야 하는 조건을 걸기 위해, Lagrange multiplier를 이용하여 다음 식을 최대화할 수 있습니다.

- 

$$\log p(\mathbf{X} | \boldsymbol{\mu}) = \sum_{k=1}^K N_k \log \mu_k + \lambda \left( \sum_{k=1}^K \mu_k - 1 \right)$$

- 위 식을 최대화하는  $\mu_k$ 는 다음과 같습니다.

- 

$$\mu_k = \frac{N_k}{N}$$

- 이는 전체 데이터 중 k번째 카테고리를 갖는 데이터의 비율과 같습니다.

# Categorical-최적화

- 증명

$$\frac{\partial \log p(\mathbf{X} | \boldsymbol{\mu})}{\partial \mu_k} = \frac{N_k}{\mu_k} + \lambda, \frac{\partial \log p(\mathbf{X} | \boldsymbol{\mu})}{\partial \lambda} = \sum_{k=1}^K \mu_k - 1$$

- 위 두 편미분 값을 0으로 두면,

$$\frac{N_k}{\mu_k} + \lambda = 0, \sum_{k=1}^K \mu_k - 1 = 0$$

- 위 두식을 정리하면,

$$\lambda = -N, \mu_k = \frac{N_k}{N}$$

# Multinomial-정의

- Multinomial distribution은  $K$ 개의 다른 카테고리를 가질 수 있는 random variable에서  $N$ 번 독립적으로 값을 얻었을 때, 각 카테고리가  $N_k$ 번씩 선택될 확률에 대한 분포입니다.
- Parameter로 각 카테고리에 대한 확률 값  $\mu = (\mu_1, \mu_2, \dots, \mu_K)$ 과 시행 횟수  $N$ 을 갖습니다.
- Multinomial distribution의 PMF는 다음과 같이 표현할 수 있습니다.

$$Mult(N_1, N_2, \dots, N_K | \mu, N) = \binom{N}{N_1 N_2 \cdots N_K} \prod_{k=1}^K \mu_k^{N_k}$$

$$\text{where } \binom{N}{N_1 N_2 \cdots N_K} \equiv \frac{N!}{N_1! N_2! \cdots N_K!}$$

# Gaussian-정의

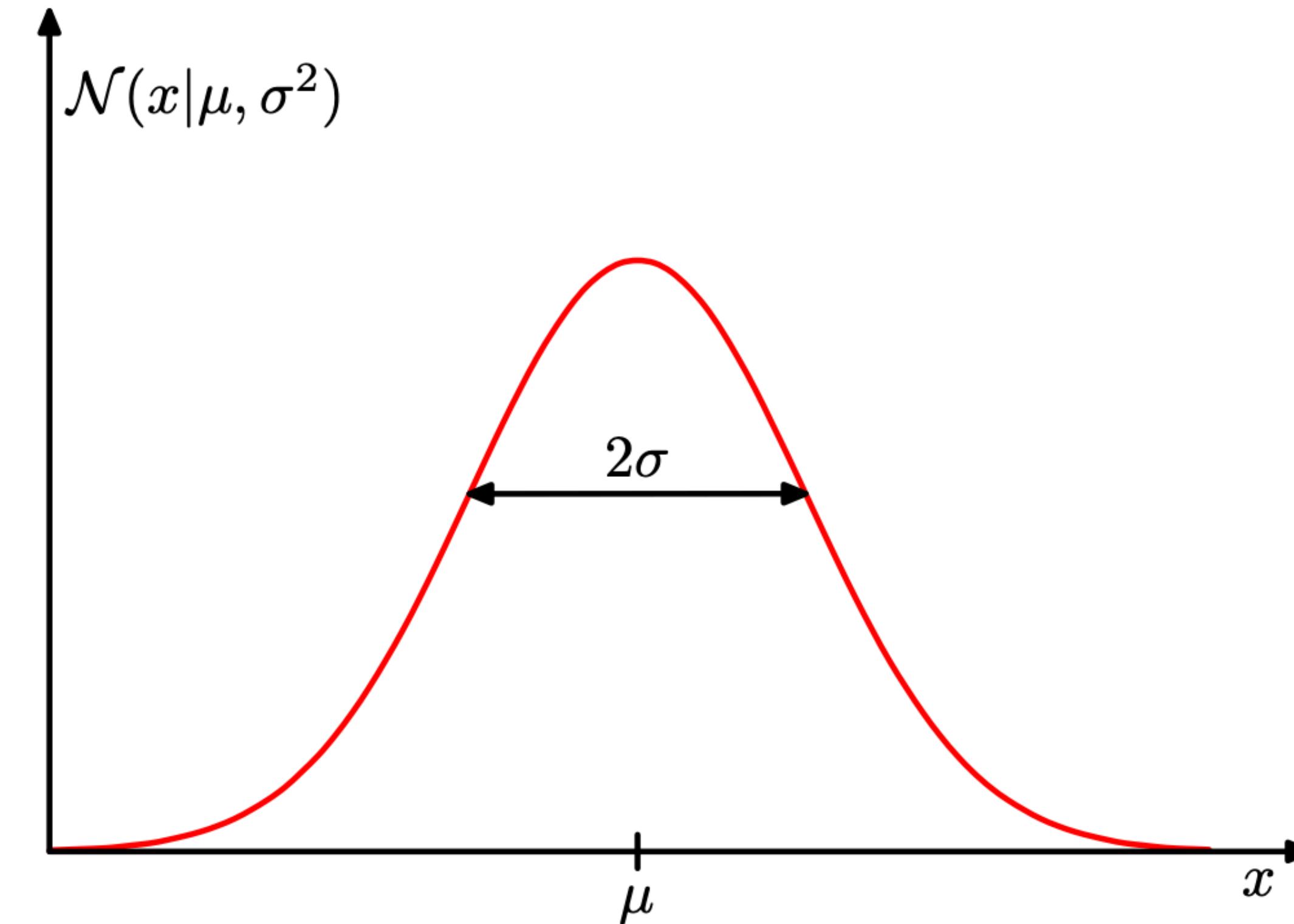
- Gaussian Distribution의 PDF(Probability Distribution Function)은 다음과 같습니다.

$$\mathcal{N}(x | \mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp \left\{ -\frac{1}{2\sigma^2}(x - \mu)^2 \right\}$$

- Parameter로 mean값을 나타내는  $\mu$ 와, 분산을 나타내는  $\sigma^2$ 을 갖습니다.

# Gaussian

**Figure 1.13** Plot of the univariate Gaussian showing the mean  $\mu$  and the standard deviation  $\sigma$ .



# Gaussian-최적화

- Gaussian Distribution에서 독립적으로 샘플링한 데이터셋  $X = \{x_1, \dots, x_N\}^T$ 에 대해 log likelihood function은 다음과 같이 쓸 수 있습니다.

$$\ln p(X | \mu, \sigma^2) = -\frac{N}{2} \ln 2\pi - \frac{N}{2} \ln \sigma^2 - \frac{1}{2\sigma^2} \sum_{n=1}^N (x_n - \mu)^2$$

- Frequentist 관점에서 위 식을 최대화하는  $\mu$ 와  $\Sigma$ 는 다음과 같습니다.

$$\mu_{\text{ML}} = \frac{1}{N} \sum_{n=1}^N x_n$$
$$\sigma_{\text{ML}}^2 = \frac{1}{N} \sum_{n=1}^N (x_n - \mu_{\text{ML}})^2$$

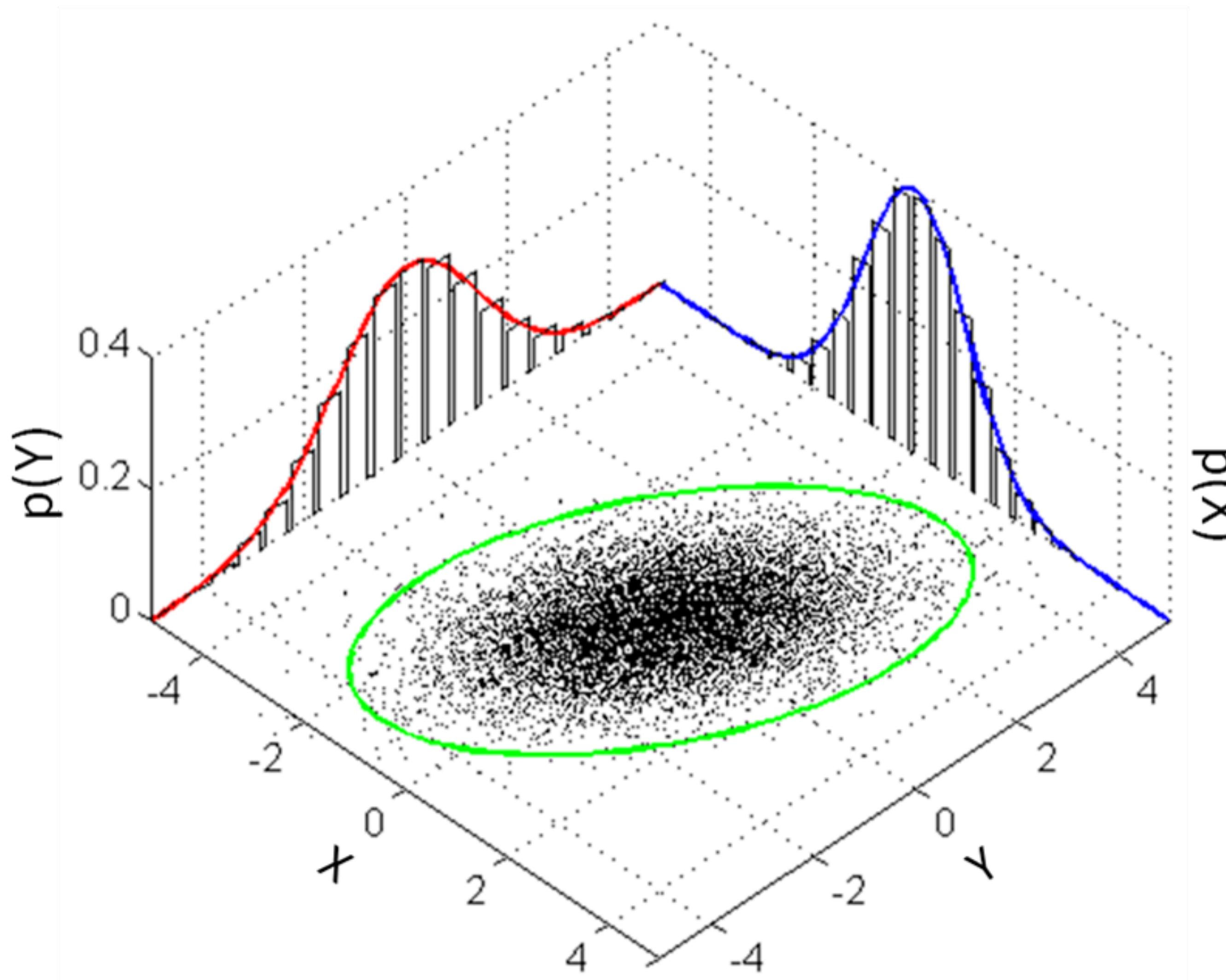
# Multivariate Gaussian-정의

- 데이터를 나타내는  $\mathbf{x}$  가  $D$  차원일 벡터일 때, Multivariate Gaussian Distribution의 PDF(Probability Distribution Function)은 다음과 같습니다.

$$\mathcal{N}(\mathbf{x} | \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right\}$$

- Parameter로 mean값을 나타내는  $\boldsymbol{\mu}$ 와, covariance matrix인  $\boldsymbol{\Sigma}$ 을 갖습니다.

# Bivariate Gaussian distribution



# Multivariate Gaussian-최적화

- Multivariate Gaussian Distribution에서 독립적으로 샘플링한 데이터셋  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}^T$ 에 대해 log likelihood function은 다음과 같이 쓸 수 있습니다.

$$\ln p(\mathbf{X} | \boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\frac{ND}{2} \ln(2\pi) - \frac{N}{2} \ln |\boldsymbol{\Sigma}| - \frac{1}{2} \sum_{n=1}^N (\mathbf{x}_n - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_n - \boldsymbol{\mu})$$

- Frequentist 관점에서 위 식을 최대화하는  $\boldsymbol{\mu}$ 와  $\boldsymbol{\Sigma}$ 는 다음과 같습니다.

$$\boldsymbol{\mu}_{\text{ML}} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n$$
$$\boldsymbol{\Sigma}_{\text{ML}} = \frac{1}{N} \sum_{n=1}^N (\mathbf{x}_n - \boldsymbol{\mu}_{\text{ML}}) (\mathbf{x}_n - \boldsymbol{\mu}_{\text{ML}})^T$$

# Probability Theory

# Joint & Marginal Probability Distribution

$P(x = x, y = y)$	$y_1$	$y_2$	$y_3$	$P(x = x)$
$x_1$	3/20	5/20	4/20	12/20
$x_2$	2/20	3/20	3/20	8/20
$P(y = y)$	5/20	8/20	7/20	20/20

Joint prob. distribution

$$P(x = x, y = y)$$

Marginal probability distribution

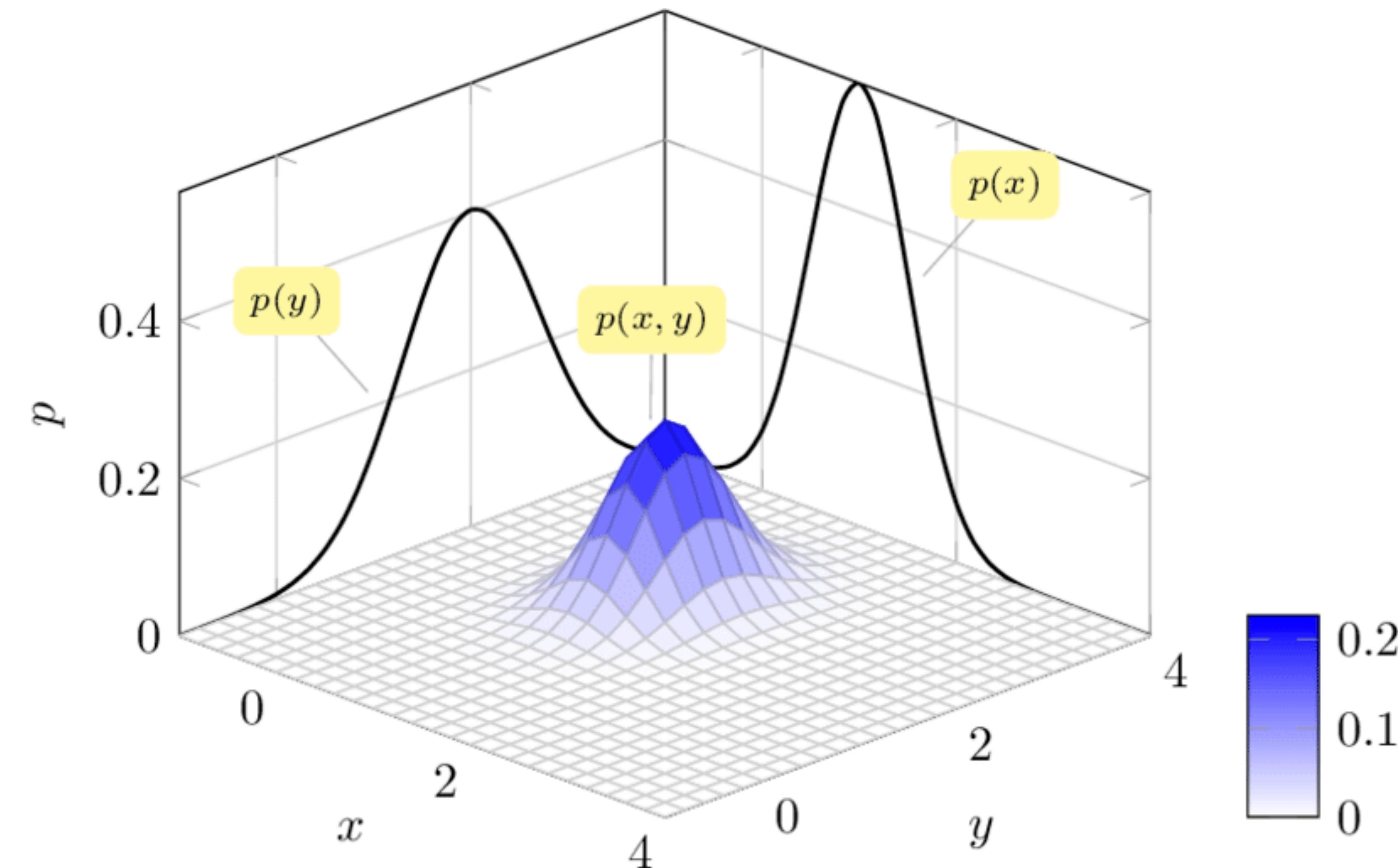
$$\forall x \in X, P(x = x) = \sum_y P(x = x, y = y)$$

Continuous case

$$p(x) = \int p(x, y) dy$$

$$\forall y \in Y, P(y = y) = \sum_x P(x = x, y = y)$$

# Joint & Marginal Probability Distribution



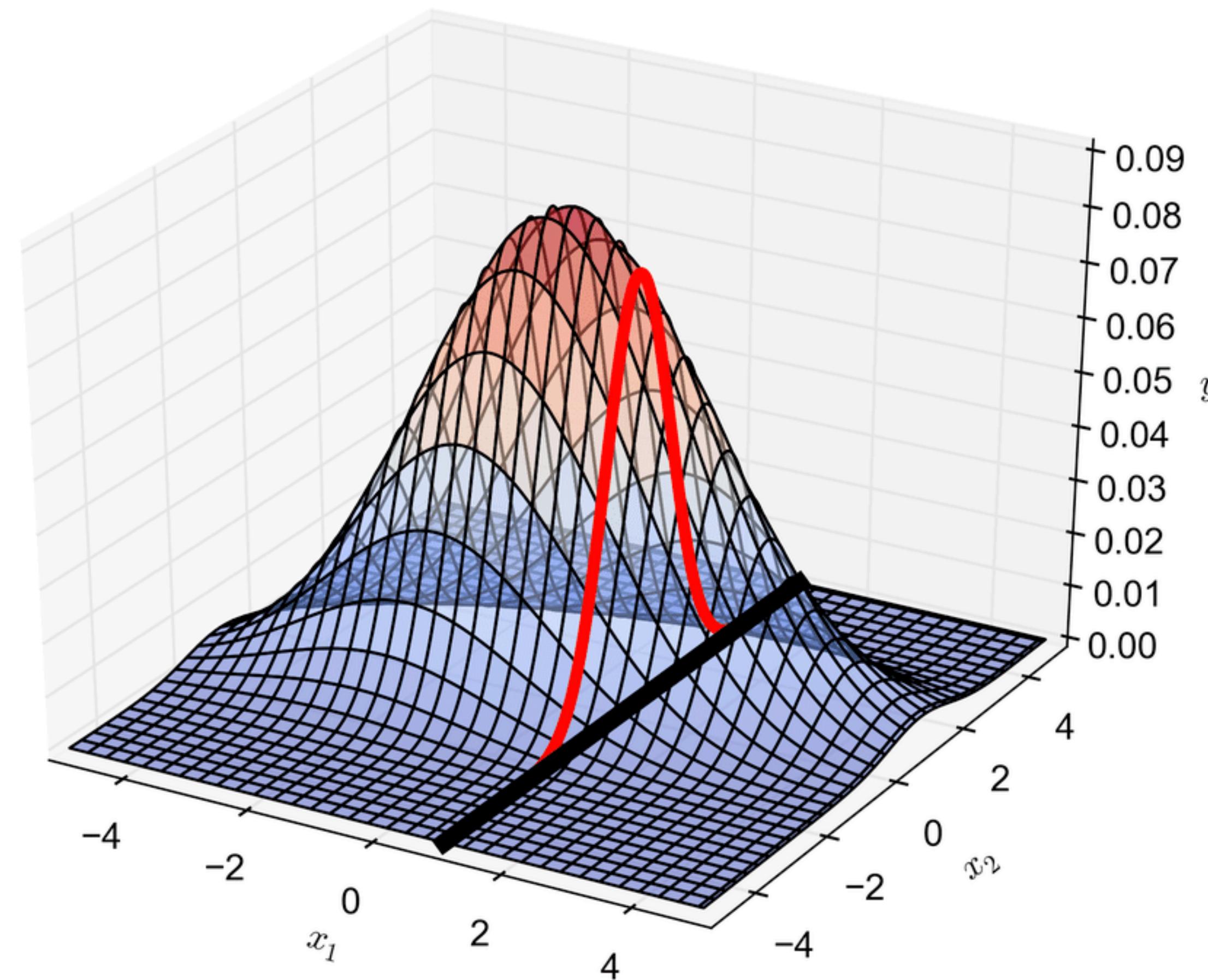
# Conditional Probability Distribution

$P(x = x, y = y)$	$y_1$	$y_2$	$y_3$	$P(x = x)$
$x_1$	3/20	5/20	4/20	12/20
$x_2$	2/20	3/20	3/20	8/20
$P(y = y)$	5/20	8/20	7/20	20/20

Conditional probability

$$P(y = y|x = x) = \frac{P(y = y, x = x)}{P(x = x)}, \quad \text{when } P(x = x) > 0$$

# Conditional Probability Distribution



# Expectation

- Discrete probability distribution  $p(x)$ 에 대한 function  $f(x)$ 의 기댓값(expectation)은 다음과 같이 계산됩니다.



$$\mathbb{E}[f] = \sum_x p(x)f(x)$$

- 또는  $p(x)$ 가 continuous인 경우 다음과 같습니다.



$$\mathbb{E}[f] = \int p(x)f(x)dx$$

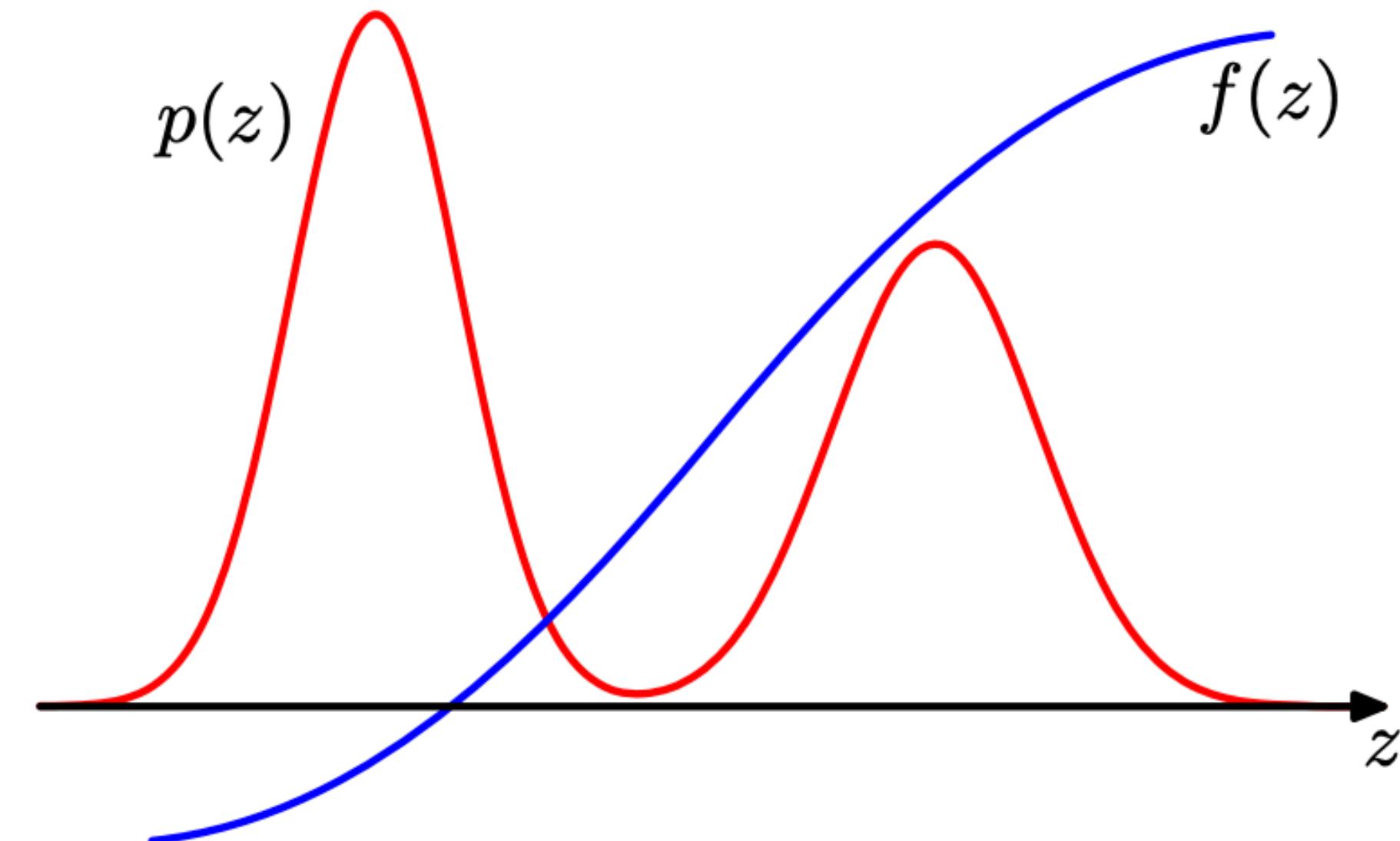
- Expectation은 다음과 같이 sampling을 통한 근사(approximation)으로 계산할 수도 있습니다.



$$\mathbb{E}[f] \simeq \frac{1}{N} \sum_{n=1}^N f(x_n)$$

# Expectation

**Figure 11.1** Schematic illustration of a function  $f(z)$  whose expectation is to be evaluated with respect to a distribution  $p(z)$ .



# Exponential Family

# Exponential Family

- Exponential family에 속하는 distribution들은 다음과 같은 형태로 전개할 수 있습니다.

$$p(\mathbf{x} | \boldsymbol{\eta}) = h(\mathbf{x})g(\boldsymbol{\eta})\exp \left\{ \boldsymbol{\eta}^T \mathbf{u}(\mathbf{x}) \right\}$$

- $\boldsymbol{\eta}$ 는 natural parameters라 부르며  $g(\boldsymbol{\eta})$ 는 적분 또는 summation시 1이 되게 하는 coefficient 역할을 합니다.

$$g(\boldsymbol{\eta}) \int h(\mathbf{x})\exp \left\{ \boldsymbol{\eta}^T \mathbf{u}(\mathbf{x}) \right\} d\mathbf{x} = 1$$

- 다음과 같은 많은 distribution이 exponential family에 속합니다.

## Examples of exponential family distributions [ edit ]

Exponential families include many of the most common distributions. Among many others, exponential families includes the following:

- [normal](#)
- [exponential](#)
- [gamma](#)
- [chi-squared](#)
- [beta](#)
- [Dirichlet](#)
- [Bernoulli](#)
- [categorical](#)
- [Poisson](#)
- [Wishart](#)
- [inverse Wishart](#)
- [geometric](#)

# Exponential Family

- I.I.D. Dataset  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ 에 대한 likelihood는 다음과 같이 나타낼 수 있습니다.

- 

$$p(\mathbf{X} | \boldsymbol{\eta}) = \left( \prod_{n=1}^N h(\mathbf{x}_n) \right) g(\boldsymbol{\eta})^N \exp \left\{ \boldsymbol{\eta}^T \sum_{n=1}^N \mathbf{u}(\mathbf{x}_n) \right\}$$

- likelihood를 최대화하는 natural parameter  $\boldsymbol{\eta}$ 을 구하기 위해  $\boldsymbol{\eta}$ 에 대한 그래디언트를 0으로 놓고 식을 전개하면 다음과 같습니다.

- 

$$-\nabla \ln g(\boldsymbol{\eta}_{\text{ML}}) = \frac{1}{N} \sum_{n=1}^N \mathbf{u}(\mathbf{x}_n)$$

- Maximum likelihood 방식으로 parameter를 구하기 위해 필요한 정보가  $\sum_{n=1}^N \mathbf{u}(\mathbf{x}_n)$ 에 모두 담겨 있으므로 이를 sufficient statistic라 부릅니다.

# Exponential Family Bernoulli

- Bernoulli의 PMF는 다음과 같이 exponential family 형식으로 전개할 수 있습니다.

- 

$$\begin{aligned} p(x|\mu) &= \text{Bern}(x|\mu) = \mu^x(1-\mu)^{1-x} \\ &= \exp\{x \ln \mu + (1-x) \ln(1-\mu)\} \\ &= (1-\mu) \exp\left\{\ln\left(\frac{\mu}{1-\mu}\right)x\right\} \end{aligned}$$

- 이는 natural parameter  $\eta = \ln\left(\frac{\mu}{1-\mu}\right)$ 가 주어진 식으로 다음과 같이 쓸 수 있습니다.

- 

$$p(x|\eta) = \sigma(-\eta) \exp(\eta x), \text{ where } \sigma(\eta) = \frac{1}{1 + \exp(-\eta)}$$

- 앞에서 밝힌 function들  $u(x), h(x), g(\eta)$ 는 다음과 같이 됩니다.

- 

$$\begin{aligned} u(x) &= x \\ h(x) &= 1 \\ g(\eta) &= \sigma(-\eta) \end{aligned}$$

# Exponential Family Categorical

- Categorical의 PMF는 다음과 같이 exponential family 형식으로 전개할 수 있습니다.

- 

$$p(\mathbf{x} | \boldsymbol{\mu}) = \prod_{k=1}^M \mu_k^{x_k} = \exp \left\{ \sum_{k=1}^M x_k \ln \mu_k \right\}, \text{ where } \mathbf{x} = (x_1, \dots, x_N)^T \text{ and } \boldsymbol{\mu} = (\mu_1, \dots, \mu_N)^T$$

- 이는 natural parameter  $\boldsymbol{\eta} = (\eta_1, \dots, \eta_M)^T$  가 주어진 식으로 다음과 같이 쓸 수 있습니다.

- 

$$p(\mathbf{x} | \boldsymbol{\eta}) = \sigma(-\boldsymbol{\eta}) \exp(\boldsymbol{\eta} \mathbf{x})$$

- 앞에서 밝힌 function들  $u(x), h(x), g(\eta)$ 은 다음과 같이 됩니다.

- 

$$\mathbf{u}(\mathbf{x}) = \mathbf{x}$$

$$h(\mathbf{x}) = 1$$

$$g(\boldsymbol{\eta}) = 1$$

# Exponential Family Gaussian

- Gaussian의 PDF는 다음과 같이 exponential family 형식으로 전개할 수 있습니다.

- 

$$\begin{aligned} p(x|\mu, \sigma^2) &= \frac{1}{(2\pi\sigma^2)^{1/2}} \exp \left\{ -\frac{1}{2\sigma^2}(x-\mu)^2 \right\} \\ &= \frac{1}{(2\pi\sigma^2)^{1/2}} \exp \left\{ -\frac{1}{2\sigma^2}x^2 + \frac{\mu}{\sigma^2}x - \frac{1}{2\sigma^2}\mu^2 \right\} \end{aligned}$$

- 앞에서 밝힌 natural parameter  $\eta$ 와 function들  $u(x), h(x), g(\eta)$ 은 다음과 같이 됩니다.

- 

$$\boldsymbol{\eta} = \begin{pmatrix} \mu/\sigma^2 \\ -1/2\sigma^2 \end{pmatrix}$$

$$\mathbf{u}(x) = \begin{pmatrix} x \\ x^2 \end{pmatrix}$$

$$h(\mathbf{x}) = (2\pi)^{-1/2}$$

$$g(\boldsymbol{\eta}) = (-2\eta_2)^{1/2} \exp \left( \frac{\eta_1^2}{4\eta_2} \right)$$

# Bayesian Statistics

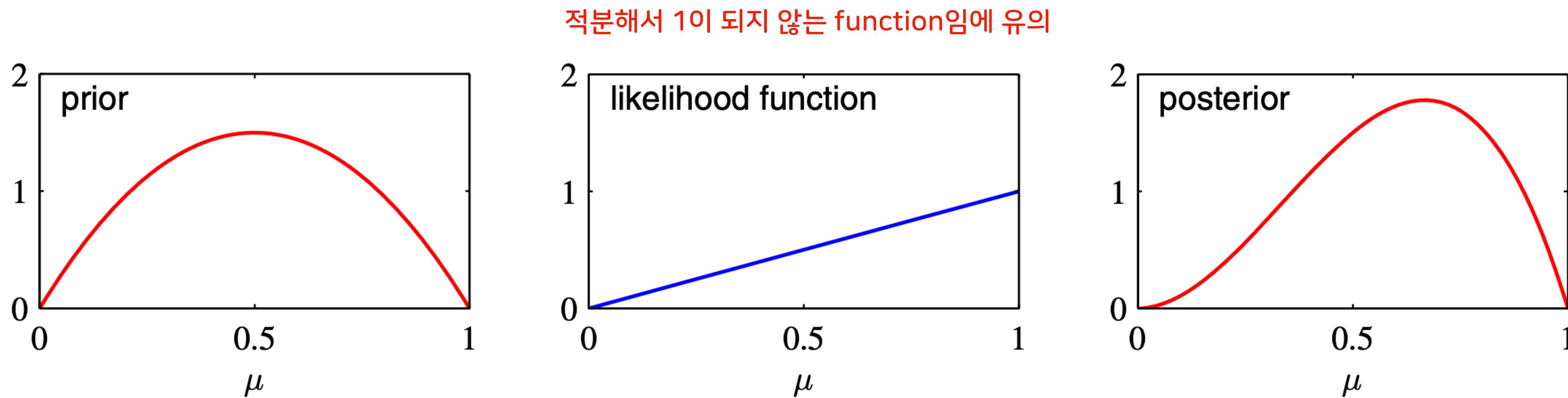
# Prior

- 베이지안에서는 Parameter나 갖고 있지 않은 hidden variable  $z$ 를 고정된 값이 아닌 random variable로 여기고 이에 대한 distribution을 prior로 설정합니다.
- Parameter  $\theta$ 에 대한 prior로  $p(\theta)$ 와 같이 표기하며, hidden variable에  $z$ 에 대한 prior로  $p(z)$ 와 같이 표기합니다.
- 데이터가 주어지기 이전이라는 의미에서 사전 분포(prior distribution)라고 칭합니다.
- Prior는 데이터와 무관하게 자유롭게 정할 수 있지만 likelihood function과 곱했을 때 같은 꼴의 수식을 가질 수 있도록 하는 conjugate prior를 주로 사용합니다.
- Bernoulli와 binomial에 대한 conjugate prior로 beta distribution이 있으며, categorical과 multinomial에 대한 conjugate prior로 Dirichlet distribution이 있습니다.
- Gaussian에 대한 conjugate prior는 그대로 Gaussian을 사용합니다.

# Likelihood

- Likelihood는 parameter이나 z-variable가 달라짐에 따라 주어진 데이터가 일어날 가능성에 대한 함수입니다.
- Parameter  $\theta$ 가 달라짐에 따라 주어진 데이터  $x$ 의 likelihood는  $\mathcal{L}(\theta | x)$  또는  $p(x | \theta)$ 로 표기하며, hidden variable  $z$ 가 달라짐에 따라 주어진 데이터  $x$ 의 likelihood는  $\mathcal{L}(z | x)$  또는  $p(x | z)$ 로 표기합니다.
- Frequentist 관점에서는 이 likelihood function을 최대화하는 parameter를 구합니다.
- data space에서의 함수가 아니라 parameter space에서의 함수입니다.
- parameter가 주어져 있고 데이터를 변수로 하는 conditional distribution과 PMF 또는 PDF식을 공유하므로 이에 유의해야 합니다.
- PMF와 PDF를 공유하되 parameter의 관점에서 바라보기 때문에 적분해서 1이 되지 않을 수 있으므로 distribution이 아닌 function입니다.

# Likelihood-Bernoulli



**Figure 2.3** Illustration of one step of sequential Bayesian inference. The prior is given by a beta distribution with parameters  $a = 2$ ,  $b = 2$ , and the likelihood function, given by (2.9) with  $N = m = 1$ , corresponds to a single observation of  $x = 1$ , so that the posterior is given by a beta distribution with parameters  $a = 3$ ,  $b = 2$ .

$$Bin(m | N, \mu) = \binom{N}{m} \mu^m (1 - \mu)^{N-m} = \binom{1}{1} \mu^1 (1 - \mu)^0 = \mu$$

$\mu$ 가 변수가 됨

# Posterior

- Posterior는 데이터가 주어져 있을 때 parameter나 z-variable의 distribution입니다.
- Bayesian에서 주로 구하려는 대상이 posterior이며, parameter의 posterior의 경우 predictive distribution을 구하는데 사용됩니다(Bayesian regression).
- z-variable의 posterior의 경우 데이터를 좀 더 낮은 차원이나 간단한 형태로 맵핑하는데 사용됩니다(GMM, PPCA, VAE).

# Bayes' Rule

- 베이지안 통계, 머신러닝의 근간을 이루는 식으로 prior, likelihood와 posterior의 관계를 나타냅니다.
- 데이터  $\mathbf{x}$ 가 주어졌을 때 parameter  $\theta$ 의 posterior는 다음과 같이 prior와 likelihood의 곱에 비례하는 형태로 나타낼 수 있습니다.

$$p(\theta | \mathbf{x}) = \frac{p(\theta)p(\mathbf{x} | \theta)}{p(\mathbf{x})}$$

- $p(\mathbf{x})$ 는 evidence라고 부르며, prior와 likelihood의 곱에 나누어 확률분포(적분해서 1)가 되도록 normalizing constant 역할을 합니다.

$$p(\mathbf{x}) = \sum_{\theta} p(\theta)p(\mathbf{x} | \theta), \theta \text{가 discrete한 경우}$$

$$p(\mathbf{x}) = \int p(\theta)p(\mathbf{x} | \theta)d\theta, \theta \text{가 continuous한 경우}$$

# Beta Distribution

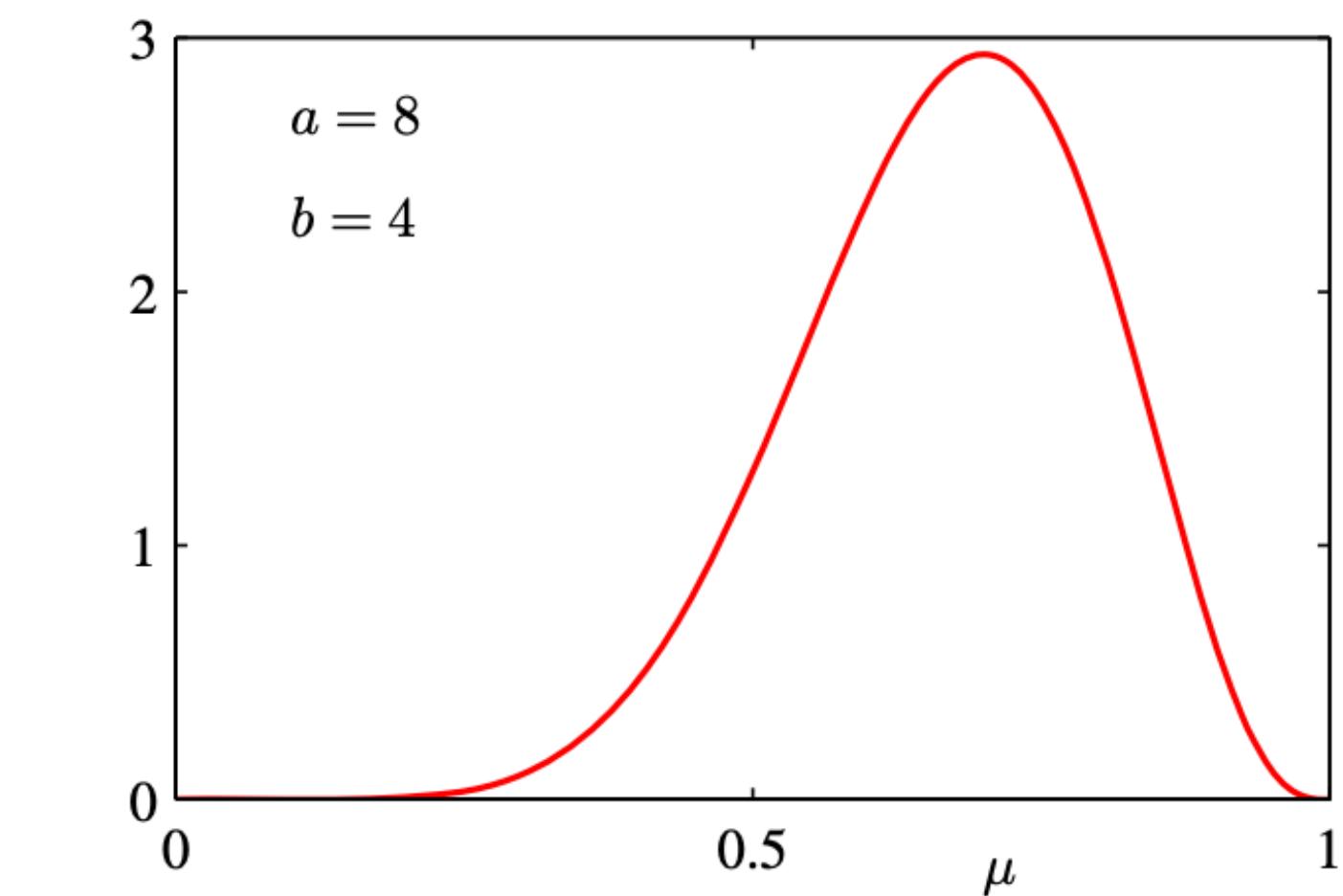
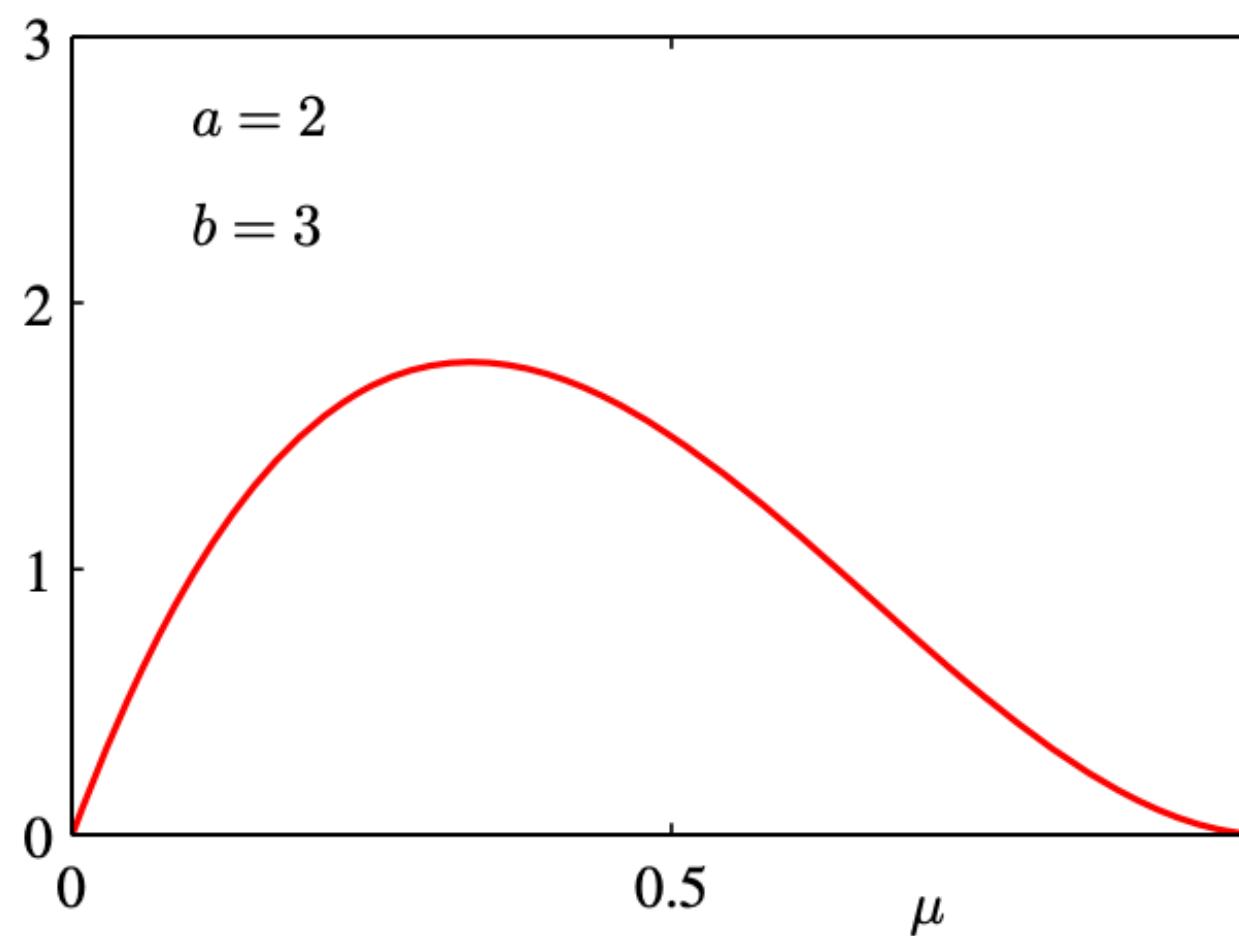
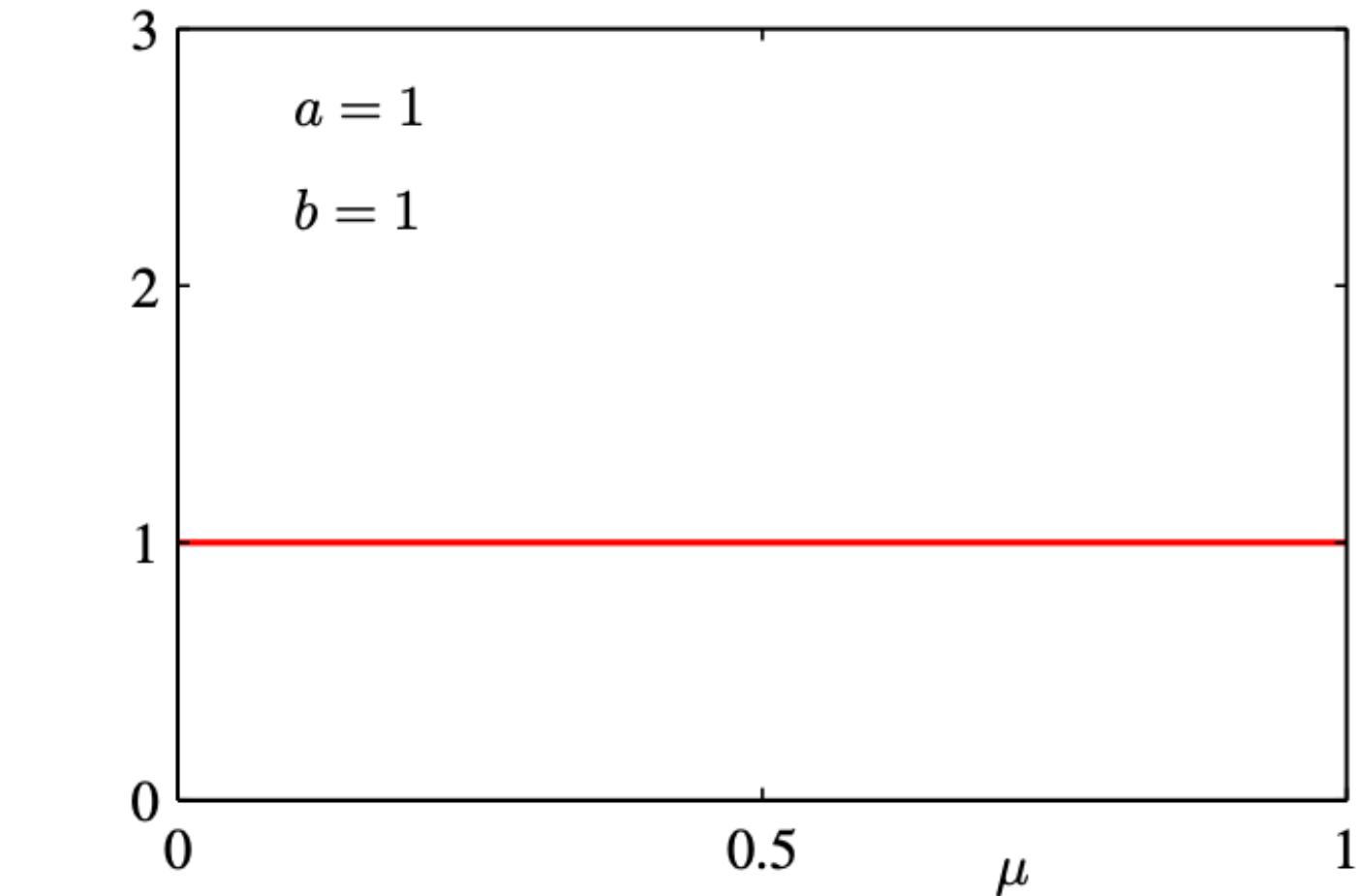
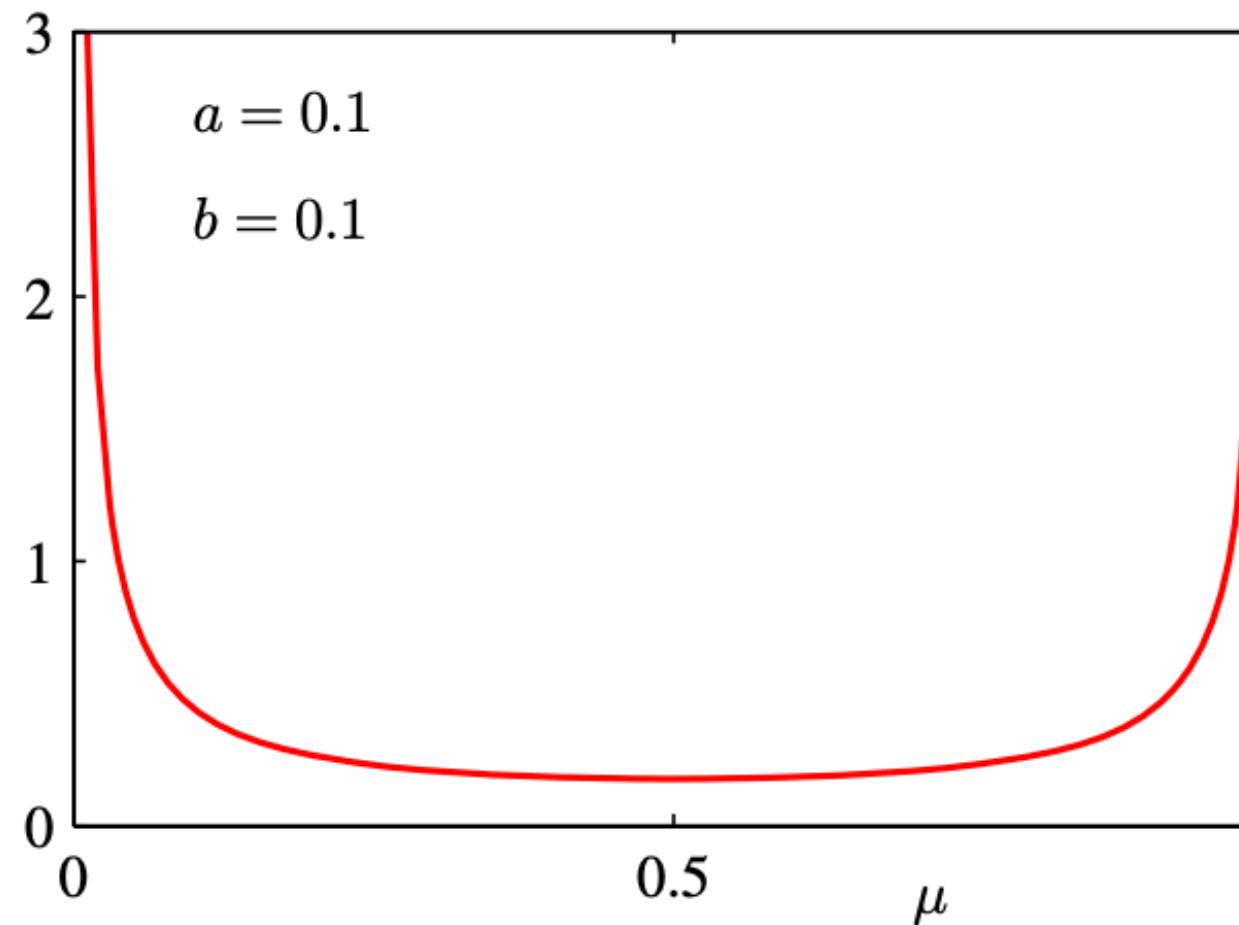
- Beta distribution은 Bernoulli distribution과 binomial distribution의 conjugate prior로 사용됩니다.
- Beta distribution의 parameter는  $a$ 와  $b$ 로 구성되며 PDF는 다음과 같습니다.
  - $$\text{Beta}(\mu | a, b) = \frac{\Gamma(a + b)}{\Gamma(a)\Gamma(b)} \mu^{a-1} (1 - \mu)^{b-1}$$
where  $\Gamma(x)$  is the gamma function
- Beta distribution은 discrete distribution에 대한 prior이지만 parameter에 대한 분포이므로 continuous distribution입니다.
- Gamma function이 들어가 있는  $\frac{\Gamma(a + b)}{\Gamma(a)\Gamma(b)}$  부분은 normalizing constant 역할을 하며 shape을 정하는데 기억하지 않습니다.

# Beta Distribution

- Beta distribution의 mean과 variance는 다음과 같습니다. Bernoulli와 conjugate관계를 가지면서 평균이  $\frac{a}{a+b}$ 가 되도록 만든 distribution이라 생각할 수 있습니다.

$$\mathbb{E}[\mu] = \frac{a}{a+b}$$

$$\text{var}[\mu] = \frac{ab}{(a+b)^2(a+b+1)}$$



# Beta Distribution

- Prior인 beta distribution와 likelihood인 binomial likelihood function를 이용하여 posterior를 구할 수 있습니다.
- 이 때, beta distribution은 conjugate prior이므로 posterior는 다시 beta distribution 꼴이 됩니다.

$$\begin{aligned} \text{Beta}(\mu | a, b) \cdot \text{Bin}(m | N, \mu) &= \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \mu^{a-1} (1-\mu)^{b-1} \cdot \binom{N}{m} \mu^m (1-\mu)^{N-m} \\ &\propto \mu^{a+m-1} (1-\mu)^{b+l-1} \\ &\propto \frac{\Gamma(a+m+b)}{\Gamma(a+m)\Gamma(b+l)} \mu^{a+m-1} (1-\mu)^{b+l-1} \\ &= \text{Beta}(\mu | a+m, b+l), \text{ where } l = N - m \end{aligned}$$

# Beta Distribution

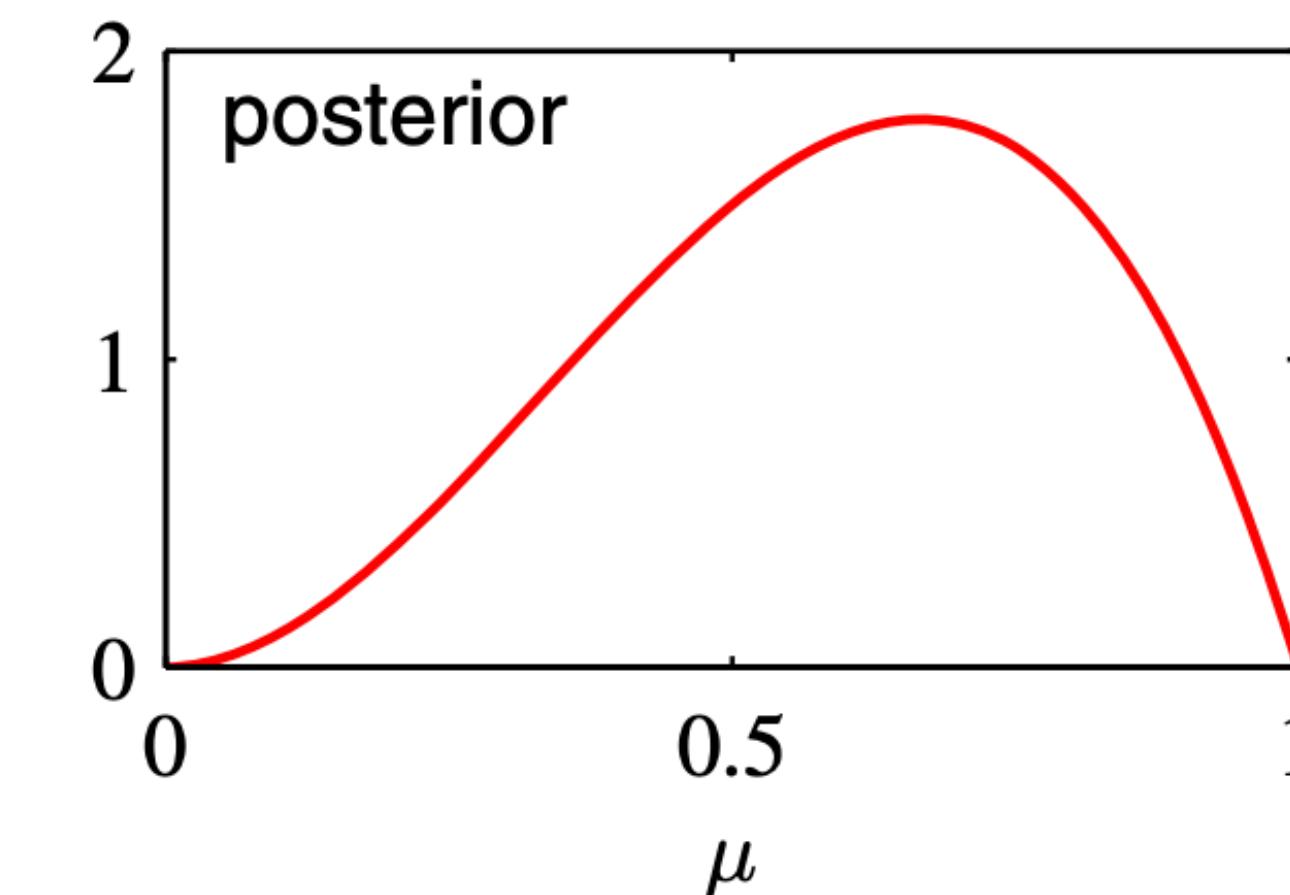
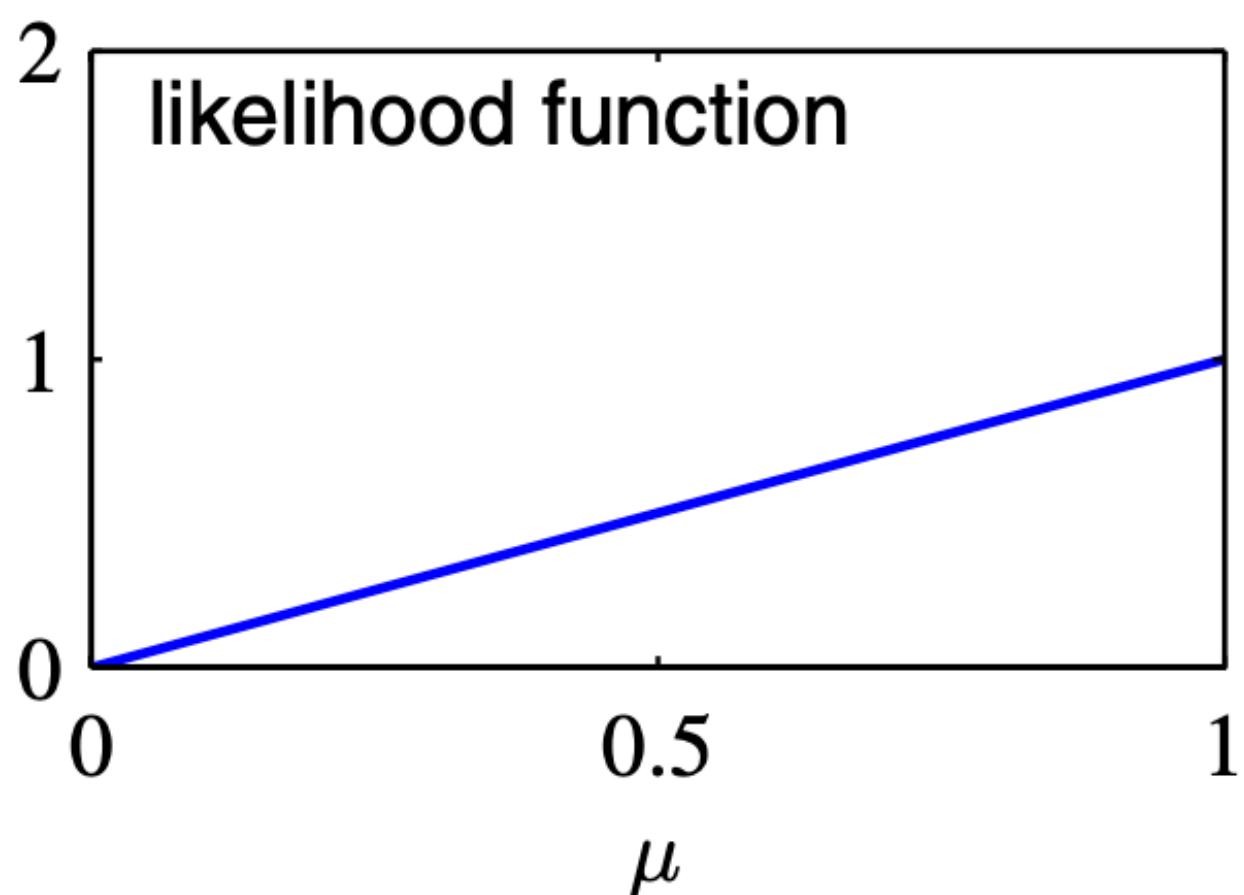
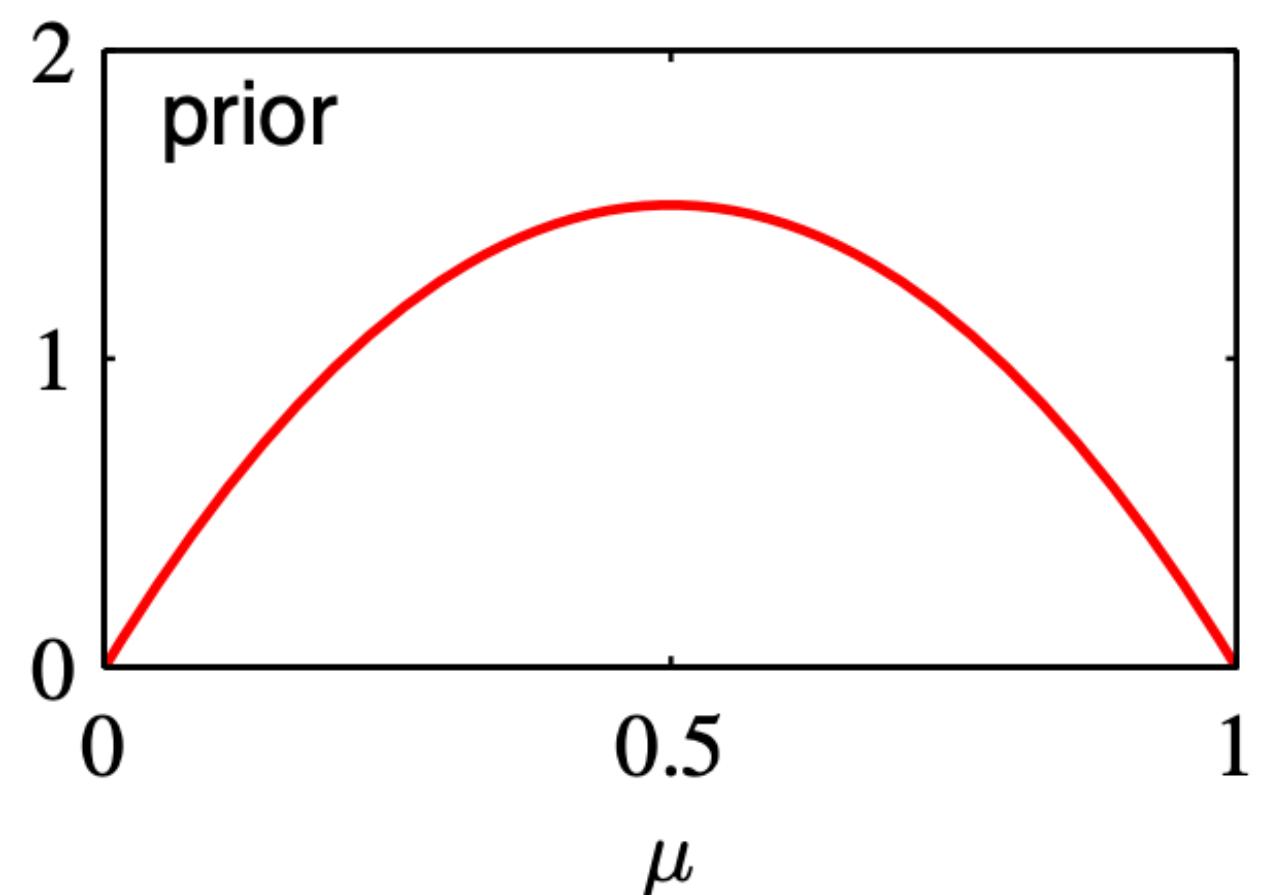
$$\text{Beta}(\mu | 2, 2)$$

$\times$

$$\text{Bin}(1 | 1, \mu)$$

$\propto$

$$\text{Beta}(\mu | 3, 2)$$



**Figure 2.3** Illustration of one step of sequential Bayesian inference. The prior is given by a beta distribution with parameters  $a = 2$ ,  $b = 2$ , and the likelihood function, given by (2.9) with  $N = m = 1$ , corresponds to a single observation of  $x = 1$ , so that the posterior is given by a beta distribution with parameters  $a = 3$ ,  $b = 2$ .

# Dirichlet Distribution

- Dirichlet distribution은 categorical distribution과 multinomial distribution의 conjugate prior로 사용됩니다.
- Dirichlet distribution의 parameter는  $\alpha = (\alpha_1, \dots, \alpha_K)^T$ 로 구성되며 PDF는 다음과 같습니다.

$$\text{Dir}(\mu | \alpha) = \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_1) \cdots \Gamma(\alpha_K)} \prod_{k=1}^K \mu_k^{\alpha_k - 1}$$

where  $\Gamma(x)$  is the gamma function and  $\alpha_0 = \sum_{k=1}^K \alpha_k$

- Dirichlet distribution은 discrete distribution에 대한 prior이지만 parameter에 대한 분포이므로 continuous distribution입니다.

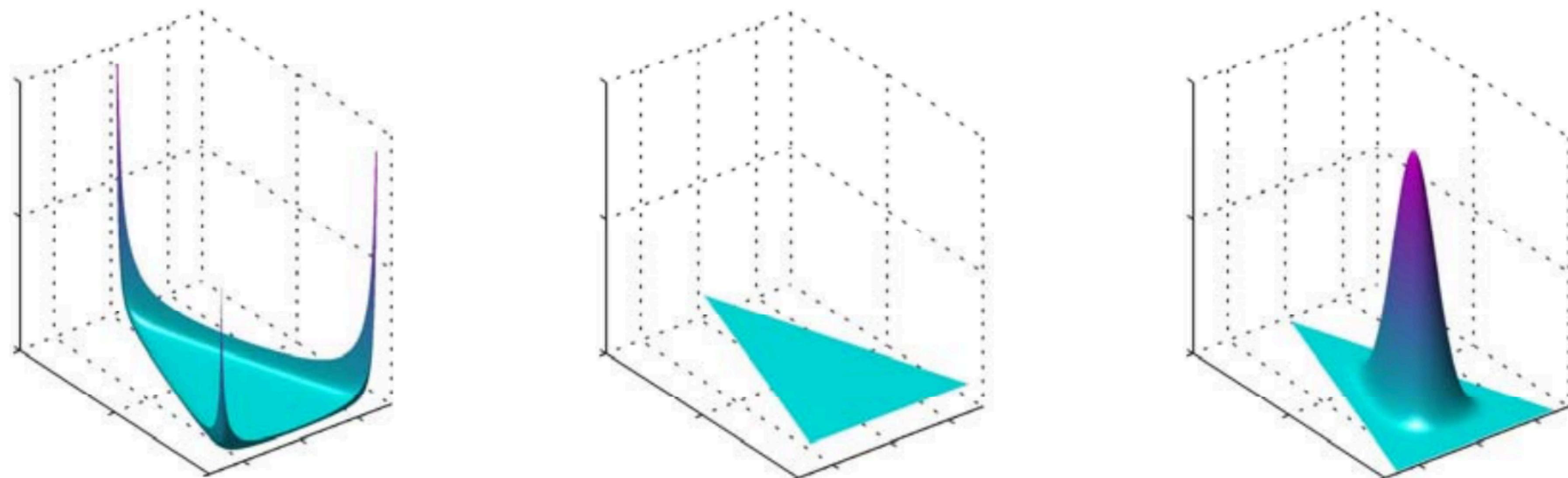
# Dirichlet Distribution

- Prior인 Dirichlet distribution에 likelihood인 multinomial likelihood function을 곱하여 posterior를 구할 수 있습니다.
- 이 때, Dirichlet distribution은 conjugate prior이므로 posterior는 다시 Dirichlet distribution 꼴이 됩니다.

●

$$\begin{aligned} \text{Dir}(\boldsymbol{\mu} | \boldsymbol{\alpha}) \cdot \text{Mult}(\mathbf{m} | \boldsymbol{\mu}, N) &= \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_1) \cdots \Gamma(\alpha_K)} \prod_{k=1}^K \mu_k^{\alpha_k-1} \binom{N}{m_1 m_2 \cdots m_K} \prod_{k=1}^K \mu_k^{m_k} \\ &\propto \prod_{k=1}^K \mu_k^{\alpha_k + m_k - 1} \\ &\propto \frac{\Gamma(\alpha_0 + m_0)}{\Gamma(\alpha_1 + m_1) \cdots \Gamma(\alpha_K + m_K)} \prod_{k=1}^K \mu_k^{\alpha_k + m_k - 1} \\ &= \text{Dir}(\boldsymbol{\mu} | \boldsymbol{\alpha} + \mathbf{m}), \text{ where } \mathbf{m} = (m_1, \dots, m_K) \end{aligned}$$

# Dirichlet Distribution



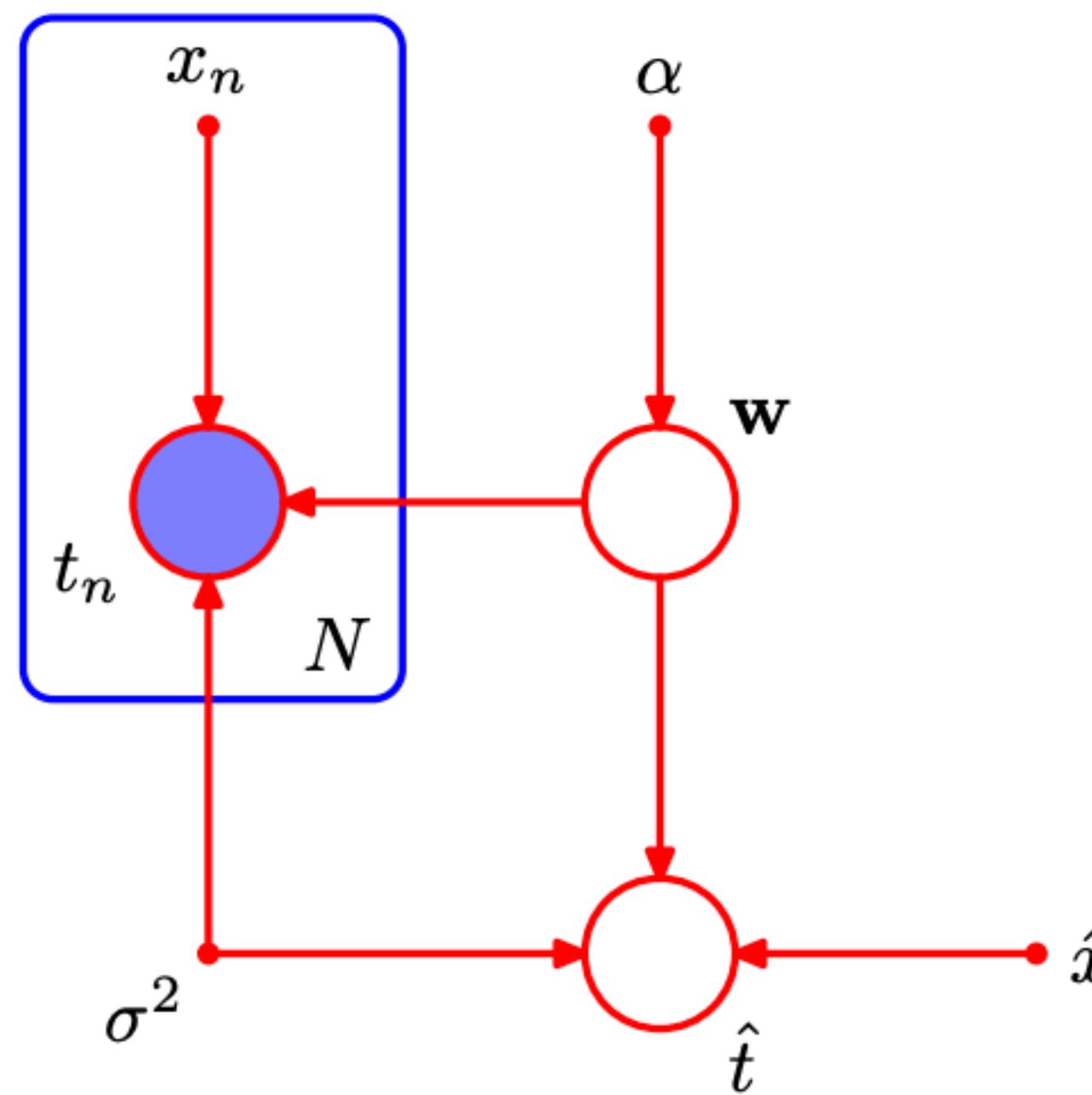
**Figure 2.5** Plots of the Dirichlet distribution over three variables, where the two horizontal axes are coordinates in the plane of the simplex and the vertical axis corresponds to the value of the density. Here  $\{\alpha_k\} = 0.1$  on the left plot,  $\{\alpha_k\} = 1$  in the centre plot, and  $\{\alpha_k\} = 10$  in the right plot.

# Graphical Representation

# Graphical Representation의 예

- Linear Regression

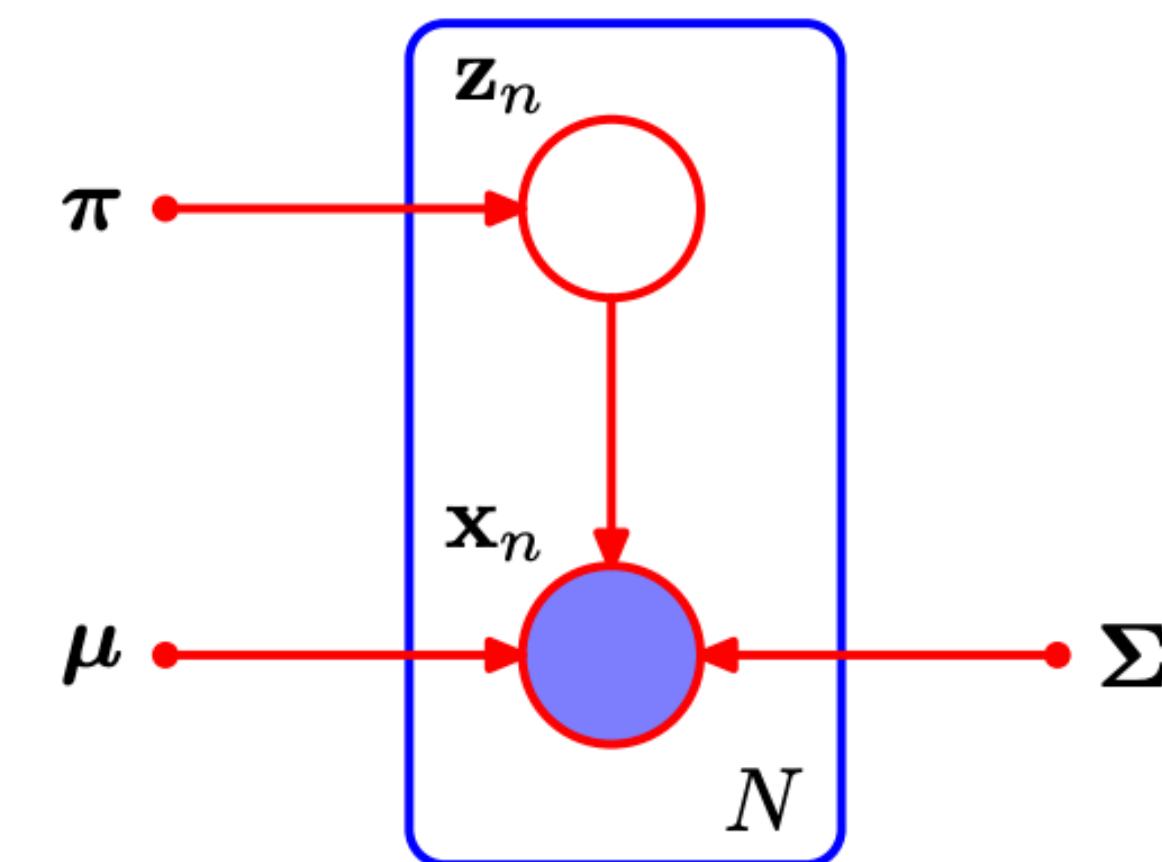
**Figure 8.7** The polynomial regression model, corresponding to Figure 8.6, showing also a new input value  $\hat{x}$  together with the corresponding model prediction  $\hat{t}$ .



# Graphical Representation의 예

- Gaussian Mixture Models

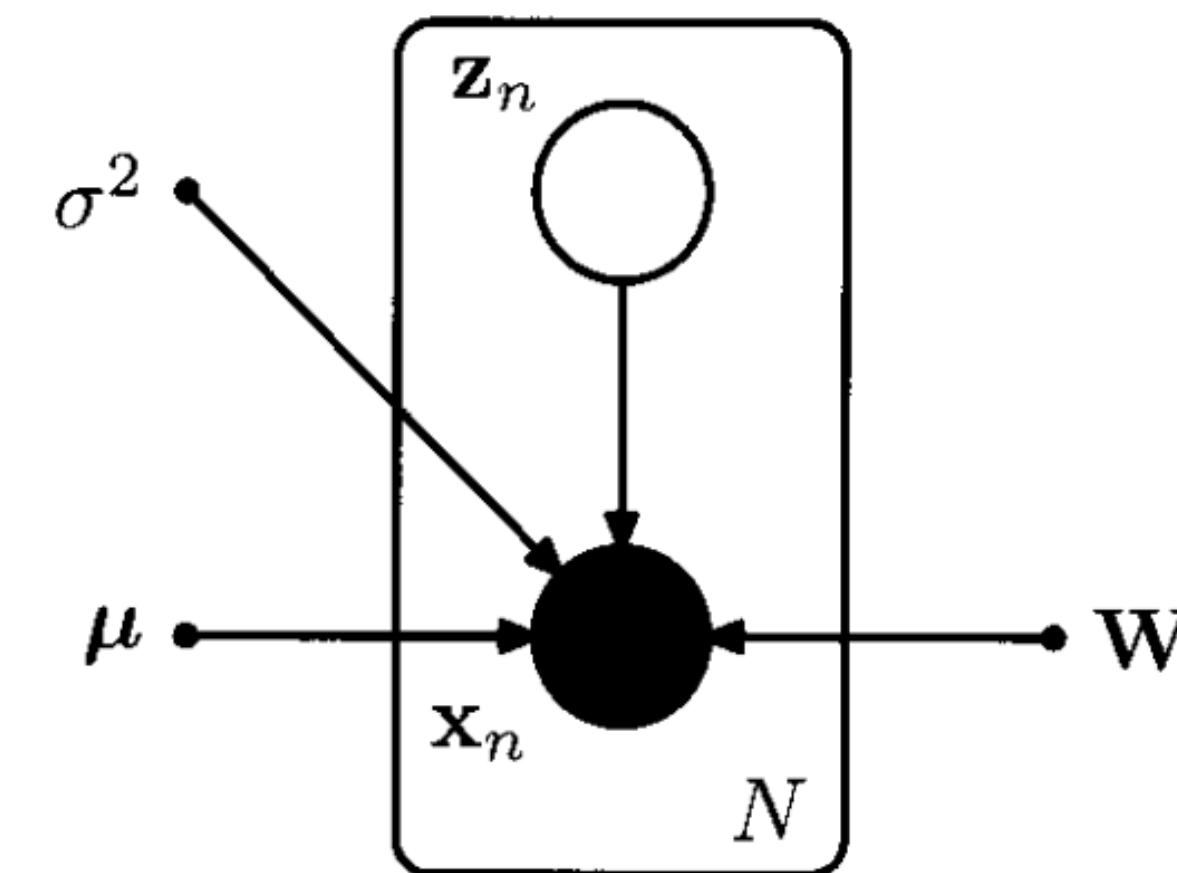
**Figure 9.6** Graphical representation of a Gaussian mixture model for a set of  $N$  i.i.d. data points  $\{\mathbf{x}_n\}$ , with corresponding latent points  $\{\mathbf{z}_n\}$ , where  $n = 1, \dots, N$ .



# Graphical Representation의 예

- Probabilistic PCA

**Figure 12.10** The probabilistic PCA model for a data set of  $N$  observations of  $\mathbf{x}$  can be expressed as a directed graph in which each observation  $\mathbf{x}_n$  is associated with a value  $z_n$  of the latent variable.



# Graphical Representation의 예

## ● Variational Auto-Encoders

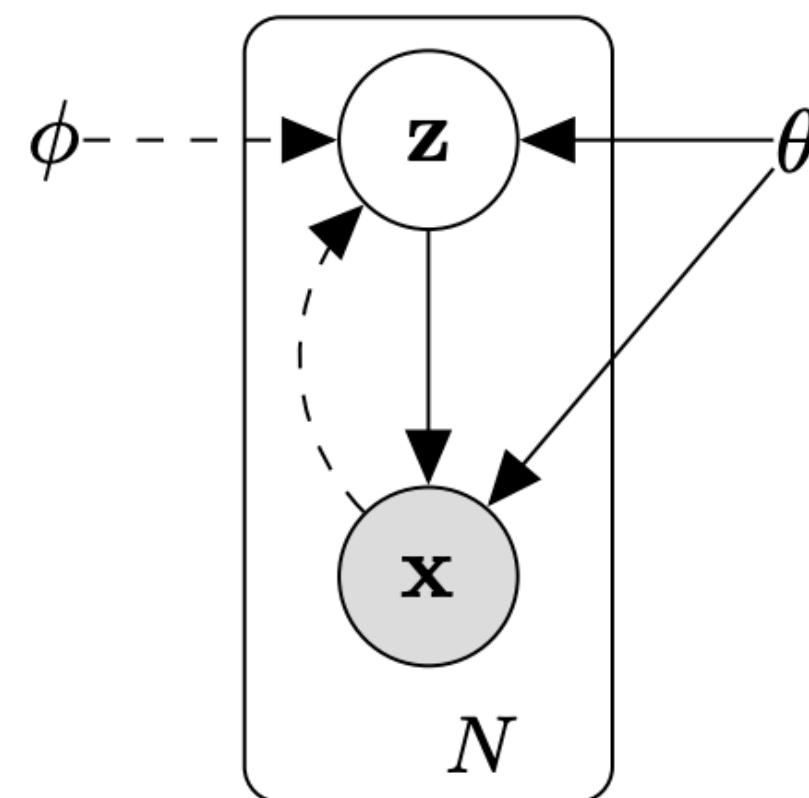
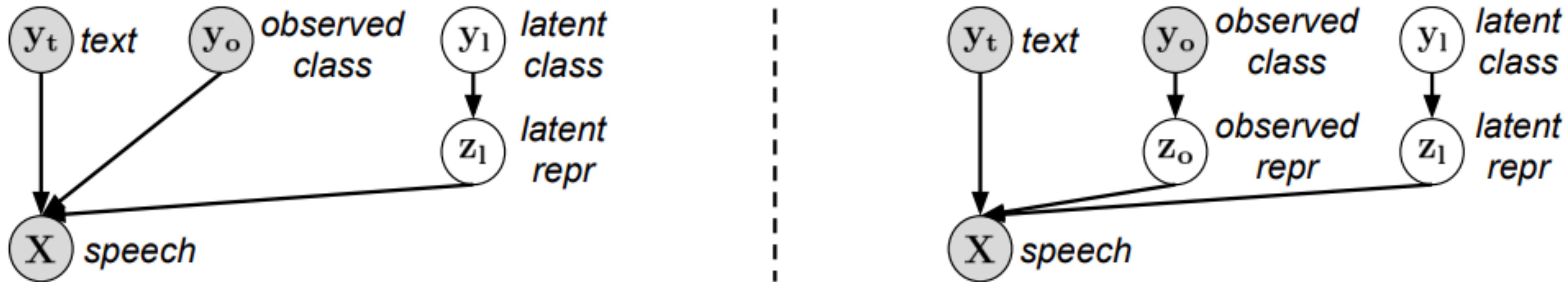


Figure 1: The type of directed graphical model under consideration. Solid lines denote the generative model  $p_\theta(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z})$ , dashed lines denote the variational approximation  $q_\phi(\mathbf{z}|\mathbf{x})$  to the intractable posterior  $p_\theta(\mathbf{z}|\mathbf{x})$ . The variational parameters  $\phi$  are learned jointly with the generative model parameters  $\theta$ .

# Graphical Representation의 예

- Controllable speech synthesis (Tacotron)

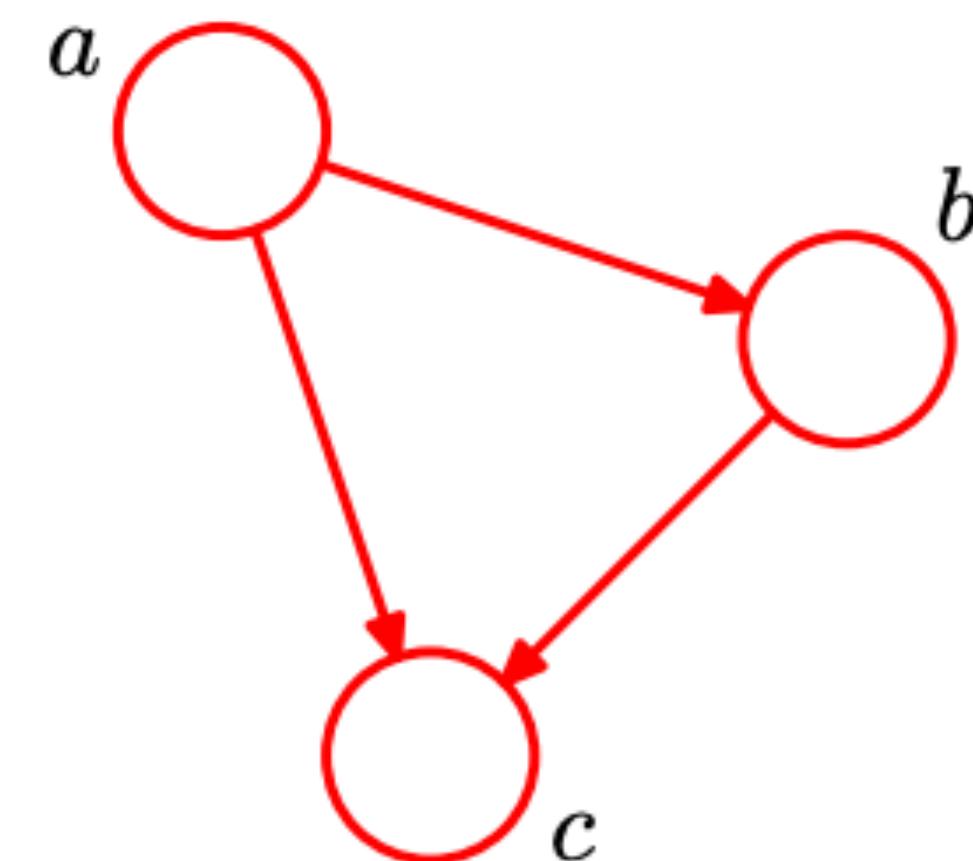


# Bayesian Network

- Joint distribution  $p(a, b, c)$ 을 chain rule에 의해 다음과 같이 나눌 수 있습니다.
- $$p(a, b, c) = p(a)p(b | a)p(c | a, b)$$
- 위와 같이 joint distribution을 conditional distribution들의 곱으로 표현한 것을 directed graph로 나타낼 수 있습니다.

**Figure 8.1**

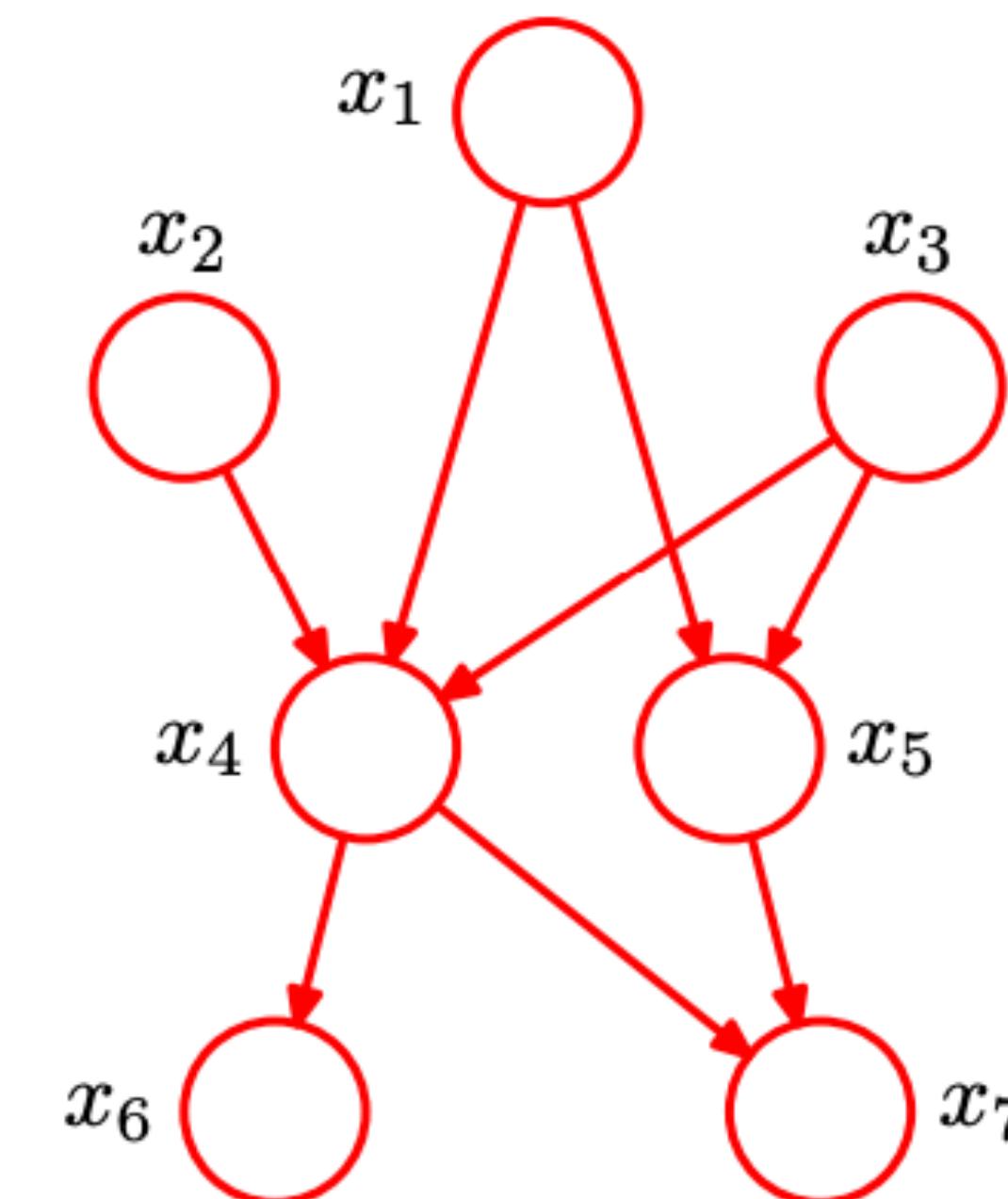
A directed graphical model representing the joint probability distribution over three variables  $a$ ,  $b$ , and  $c$ , corresponding to the decomposition on the right-hand side of (8.2).



# Bayesian Network

- 이러한 graphical representation을 통해 random variable 간의 dependency를 직관적으로 나타낼 수 있습니다.
- $$p(x_1)p(x_2)p(x_3)p(x_4 | x_1, x_2, x_3)p(x_5 | x_1, x_3)p(x_6 | x_4)p(x_7 | x_4, x_5)$$

**Figure 8.2** Example of a directed acyclic graph describing the joint distribution over variables  $x_1, \dots, x_7$ . The corresponding decomposition of the joint distribution is given by (8.4).



# Conditional Independence

- Random variable  $a, b, c$ 에 대해 다음 등식이 성립하면  $c$ 가 주어졌을 때  $a$ 는  $b$ 에 조건부 독립이라고 말합니다. ( $a$  is conditionally independent of  $b$  given  $c$ .)

●

$$p(a | b, c) = p(a | c)$$

- 이는 다음 등식이 성립함과 동치입니다.

●

$$p(a, b | c) = p(a | c)p(b | c)$$

- $a$ 가  $b$ 에 conditionally independent하다는 것을 다음과 같이 간단히 표현할 수 있습니다.

●

$$a \perp\!\!\!\perp b | c$$

# tail-to-tail, common parent

- 다음과 같이 joint distribution을 나눌 수 있는 경우를 생각해봅시다.



$$p(a, b, c) = p(c)p(a | c)p(b | c)$$

- Random variable  $c$ 가 condition으로 주어지지 않은 경우  $a$ 와  $b$ 의 joint distribution은 다음과 같이 전개할 수 있습니다.



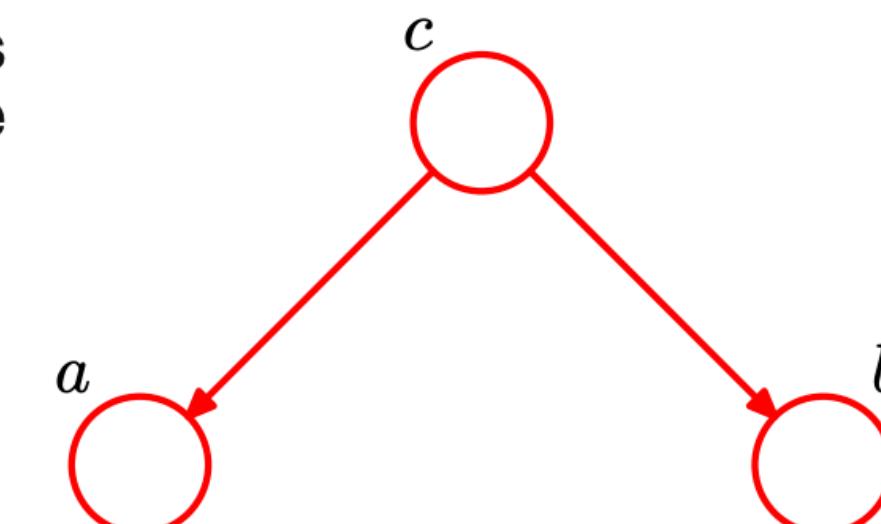
$$p(a, b) = \sum_c p(a | c)p(b | c)p(c)$$

- 위 식을 더 전개하여  $p(a)p(b)$ 를 만들 수 없으므로  $a$ 와  $b$ 는 independent하지 않으며, 다음과 같이 기호로 표기할 수 있습니다.



$$a \not\perp\!\!\!\perp b | \emptyset$$

**Figure 8.15** The first of three examples of graphs over three variables  $a$ ,  $b$ , and  $c$  used to discuss conditional independence properties of directed graphical models.



# tail-to-tail, common parent

- Random variable  $c$ 가 condition으로 주어진 경우  $a$ 와  $b$ 의 joint distribution은 다음과 같이 전개할 수 있습니다.



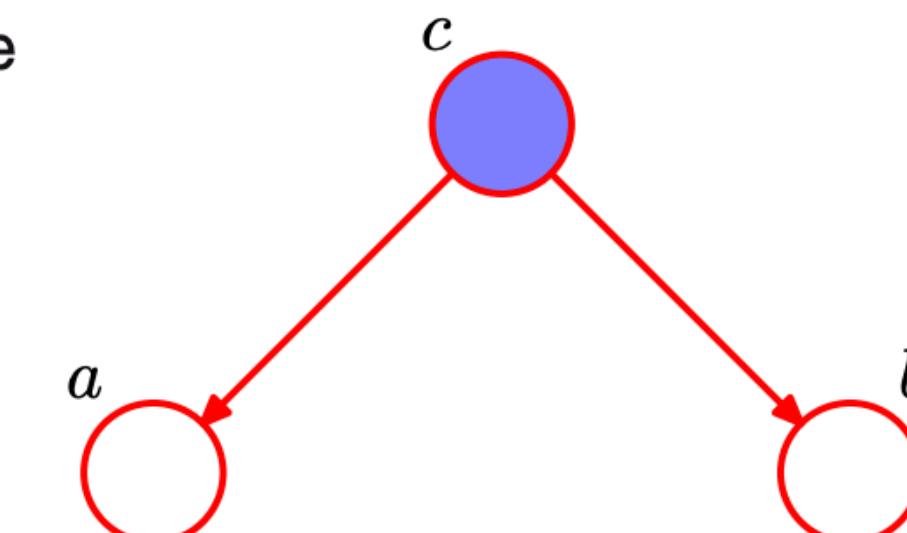
$$p(a, b | c) = \frac{p(a, b, c)}{p(c)} = p(a | c)p(b | c)$$

- 따라서  $c$ 가 condition으로 주어졌을 때  $a$ 와  $b$ 가 independent하다고 다음과 같이 기호로 표기할 수 있습니다.



$$a \perp\!\!\!\perp b | c$$

**Figure 8.16** As in Figure 8.15 but where we have conditioned on the value of variable  $c$ .



# head-to-tail, cascading

- 또는 다음과 같이 joint distribution을 나눌 수 있는 경우를 생각해봅시다.

- 

$$p(a, b, c) = p(a)p(c | a)p(b | c)$$

- Random variable  $c$ 가 condition으로 주어지지 않은 경우  $a$ 와  $b$ 의 joint distribution을 다음과 같이 전개할 수 있습니다.

- 

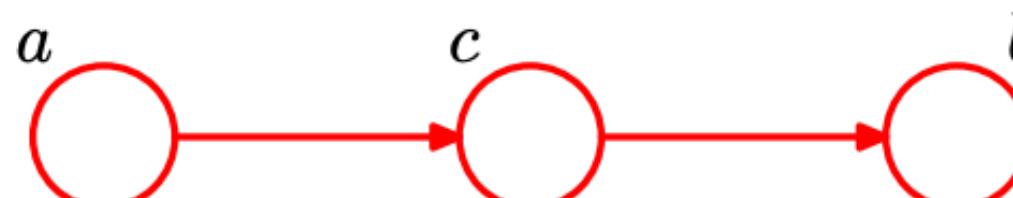
$$p(a, b) = p(a) \sum_c p(c | a)p(b | c) = p(a)p(b | a)$$

- 위 결과에 따라  $a$ 와  $b$ 는 independent하지 않으며, 다음과 같이 기호로 표기할 수 있습니다.

- 

$$a \not\perp\!\!\!\perp b | \emptyset$$

**Figure 8.17** The second of our three examples of 3-node graphs used to motivate the conditional independence framework for directed graphical models.



# head-to-tail, cascading

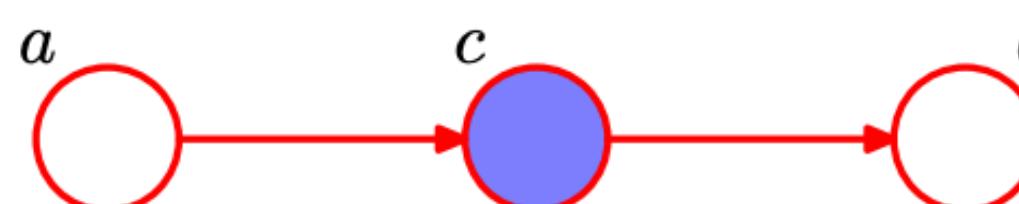
- Random variable  $c$ 가 condition으로 주어지지 않은 경우  $a$ 와  $b$ 의 joint distribution을 다음과 같이 전개할 수 있습니다.

- $$\begin{aligned} p(a, b | c) &= \frac{p(a, b, c)}{p(c)} \\ &= \frac{p(a)p(c | a)p(b | c)}{p(c)} \\ &= p(a | c)p(b | c) \end{aligned}$$

- 위 결과에 따라  $a$ 와  $b$ 는 conditionally independent하며, 다음과 같이 기호로 표기할 수 있습니다.

- $$a \perp\!\!\!\perp b | c$$

**Figure 8.18** As in Figure 8.17 but now conditioning on node  $c$ .



# head-to-head, V-structure

- 또는 다음과 같이 joint distribution을 나눌 수 있는 경우를 생각해봅시다.

- 

$$p(a, b, c) = p(a)p(b)p(c | a, b)$$

- Random variable  $c$ 가 condition으로 주어지지 않은 경우  $a$ 와  $b$ 의 joint distribution을 다음과 같이 전개할 수 있습니다.

- 

$$p(a, b) = p(a)p(b)$$

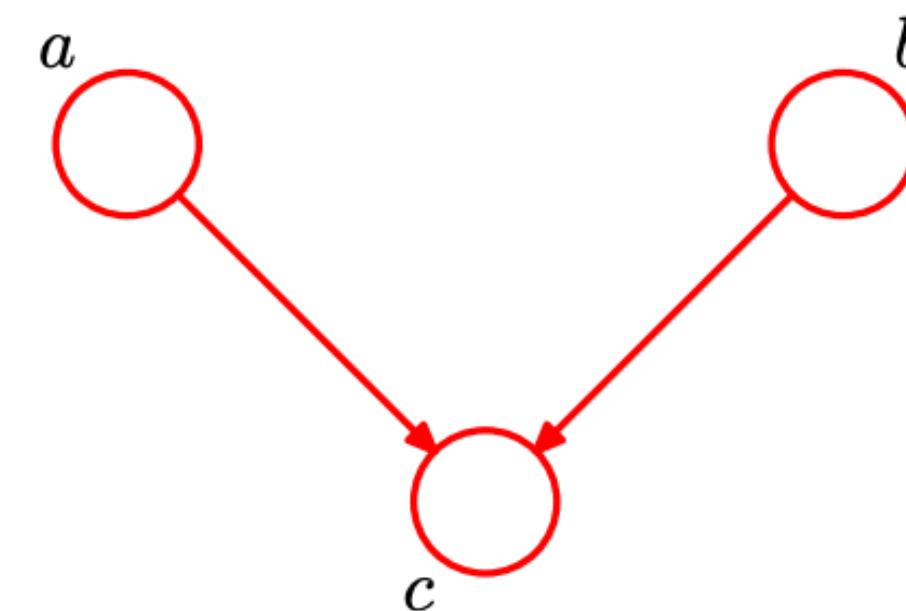
- 위 결과에 따라  $a$ 와  $b$ 는 independent하며, 다음과 같이 기호로 표기할 수 있습니다.

- 

$$a \perp\!\!\!\perp b | \emptyset$$

**Figure 8.19**

The last of our three examples of 3-node graphs used to explore conditional independence properties in graphical models. This graph has rather different properties from the two previous examples.



# head-to-head, V-structure

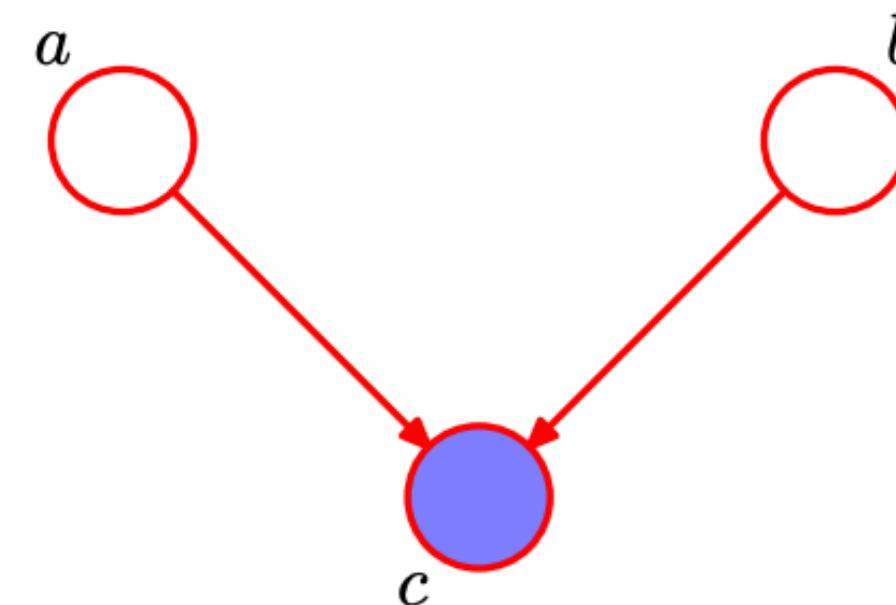
- Random variable  $c$ 가 condition으로 주어지지 않은 경우  $a$ 와  $b$ 의 joint distribution을 다음과 같이 전개할 수 있습니다.

- $$\begin{aligned} p(a, b | c) &= \frac{p(a, b, c)}{p(c)} \\ &= \frac{p(a)p(b)p(c | a, b)}{p(c)} \end{aligned}$$

- 위 결과에 따라  $a$ 와  $b$ 는 conditionally independent하지 않으며, 다음과 같이 기호로 표기할 수 있습니다.

- $$a \perp\!\!\!\perp b | c$$

**Figure 8.20** As in Figure 8.19 but conditioning on the value of node  $c$ . In this graph, the act of conditioning induces a dependence between  $a$  and  $b$ .



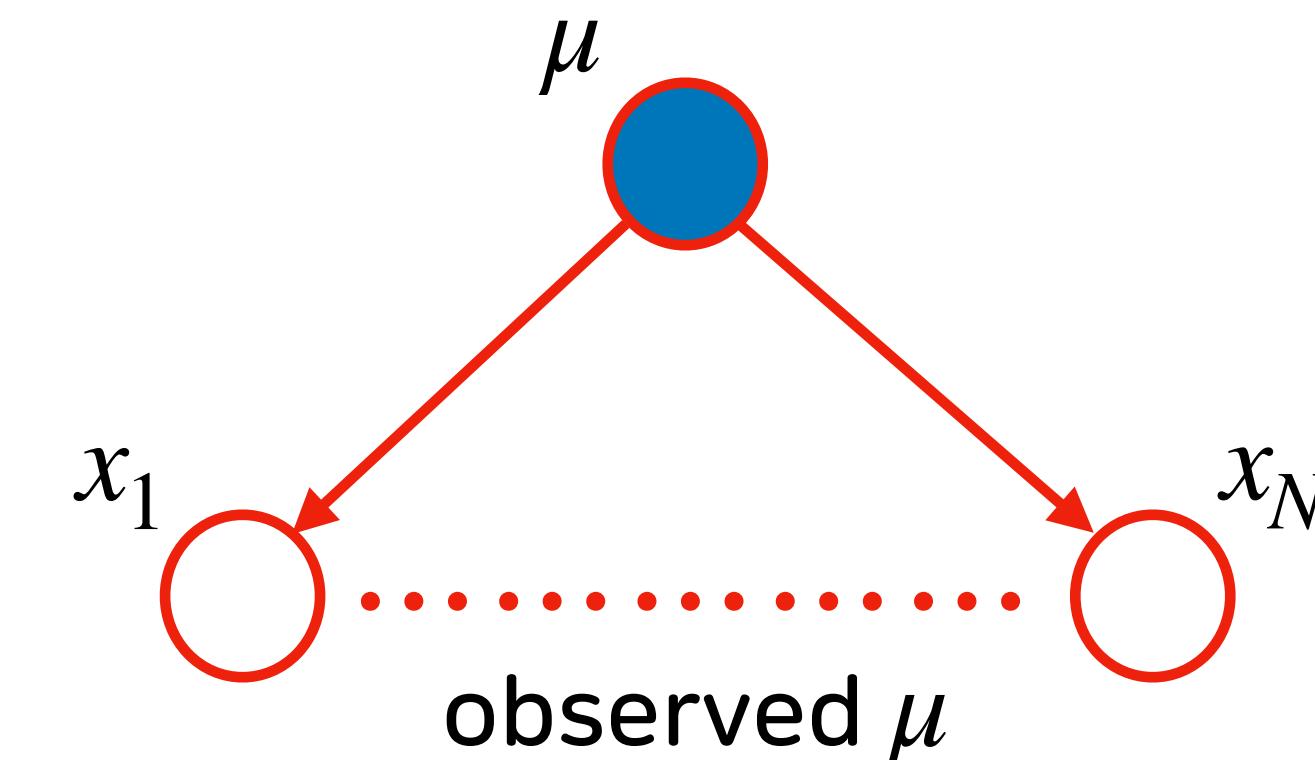
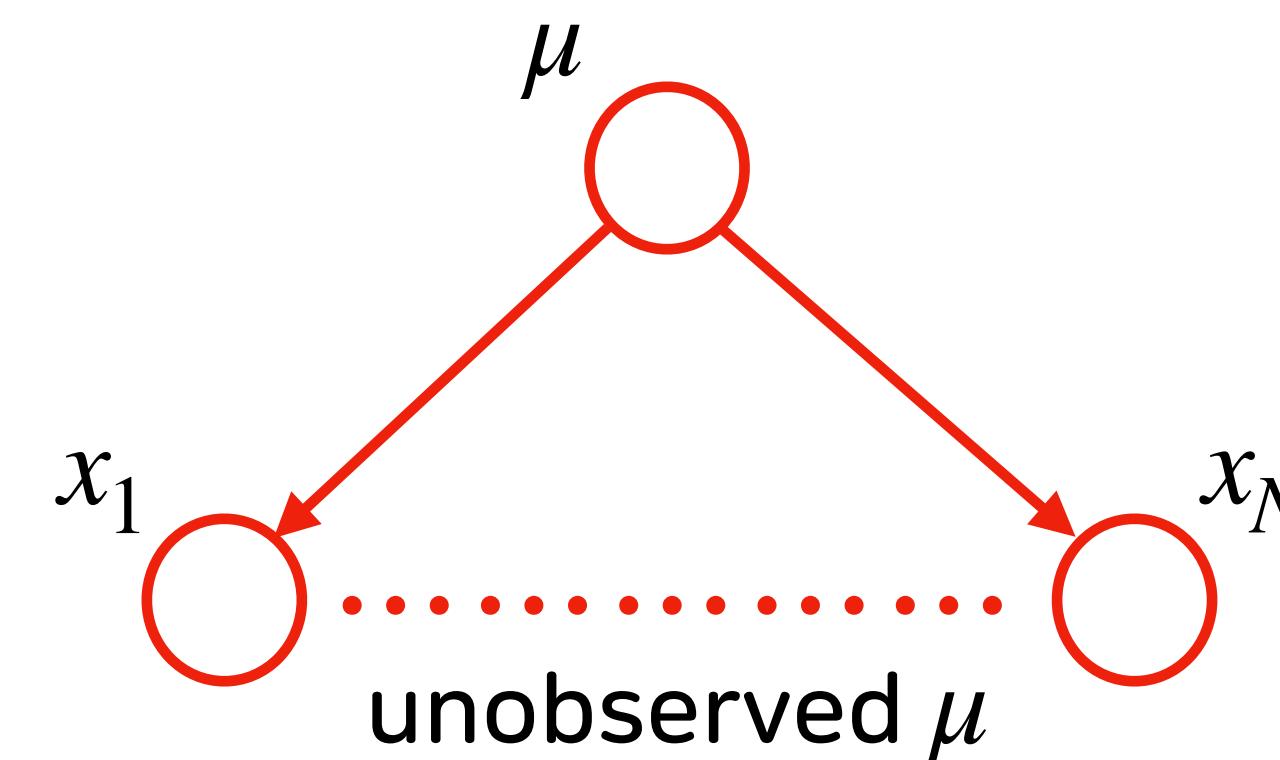
# The samples of The Gaussian Distribution

- Mean이 random variable  $\mu$ 인 univariate Gaussian distribution에서 샘플링된 데이터들  $\mathcal{D} = \{x_1, \dots, x_N\}$ 을 생각해봅시다.
- 이는 graphical representation으로 tail-to-tail (common parent) 관계에 있다고 할 수 있습니다.
- Mean  $\mu$ 가 condition으로 주어졌을 때 (observed) 각 데이터  $x_1, \dots, x_N$ 들은 independent합니다.

$$p(\mathcal{D} | \mu) = \prod_{n=1}^N p(x_n | \mu)$$

- 그러나  $\mu$ 가 condition으로 주어지지 않으면 (not observed) independent하지 않습니다.

$$p(\mathcal{D}) = \int_{-\infty}^{\infty} p(\mathcal{D} | \mu)p(\mu)d\mu \neq \prod_{n=1}^N p(x_n)$$

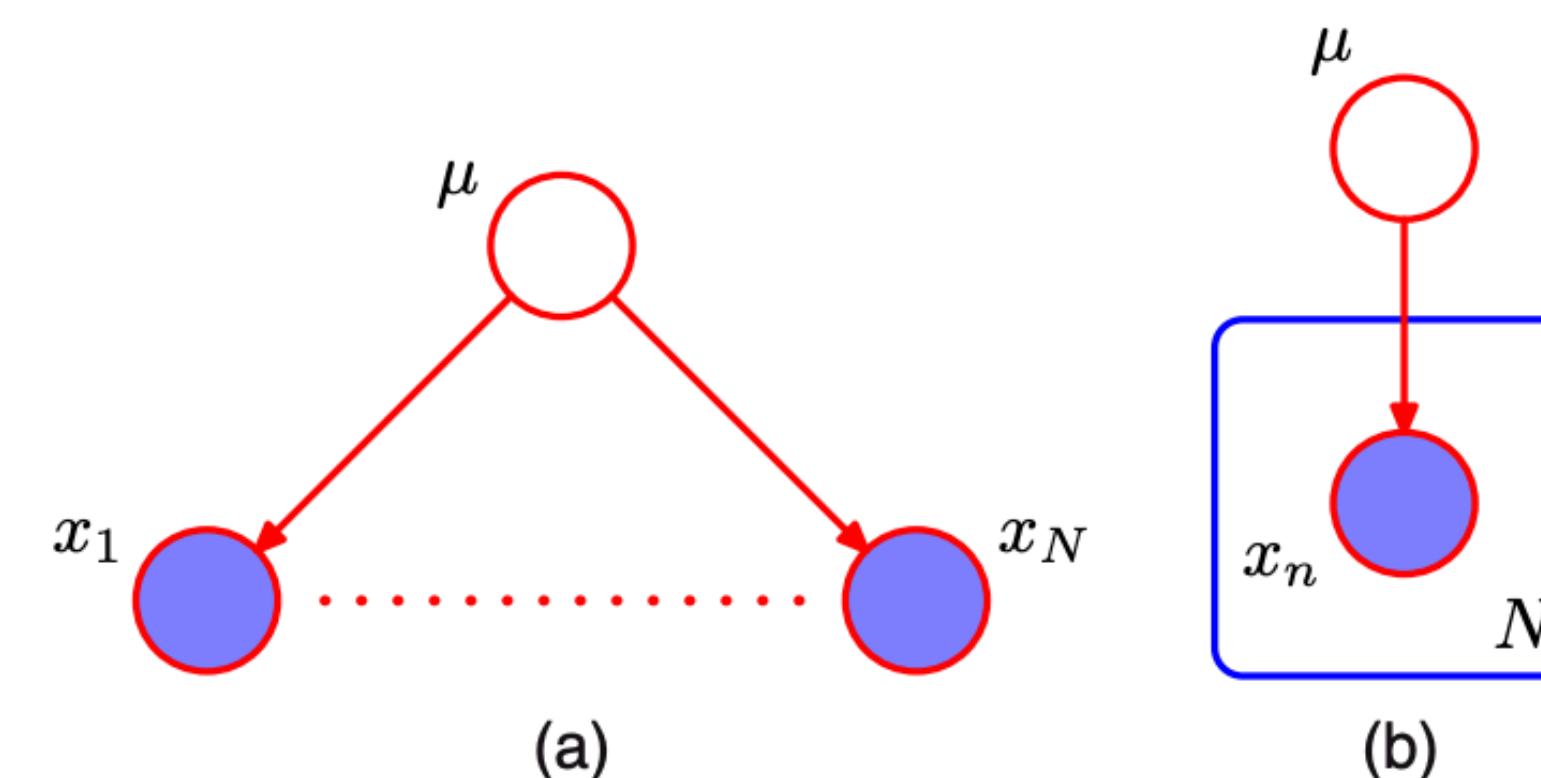


# The samples of The Gaussian Distribution

- 많은 문제에서 보통 데이터들이 주어져 있고, 파라메터를 추정하는 일을 합니다.
- 이 경우에는 샘플링된 데이터들  $\mathcal{D} = \{x_1, \dots, x_N\}$ 에 의해서 random variable인 mean  $\mu$ 를 추정해볼 수 있습니다.

$$p(\mu | \mathcal{D}) = \frac{p(\mu)p(\mathcal{D} | \mu)}{p(\mathcal{D})}$$

**Figure 8.23** (a) Directed graph corresponding to the problem of inferring the mean  $\mu$  of a univariate Gaussian distribution from observations  $x_1, \dots, x_N$ . (b) The same graph drawn using the plate notation.



# Linear Regression

# Linear Basis Function Models

- $D$ 차원의 데이터  $\mathbf{x} = (x_1, \dots, x_D)^T$ 가 주어지고 이에 대한 weight  $\mathbf{w} = (w_0, w_1, \dots, w_D)^T$ 가 주어졌을 때 linear combination의 output  $y$ 를 다음과 같이 나타낼 수 있습니다.

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + \dots + w_D x_D$$

- 또는 데이터  $\mathbf{x}$ 를 임의의 basis function  $\phi : \mathbb{R}^D \mapsto \mathbb{R}^M$ 에 의해 매핑시키고 weight  $\mathbf{w} = (w_0, w_1, \dots, w_M)^T$ 을 적용해볼 수 있습니다.

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x})$$

- $w_0$ 은 bias라 부르며,  $\phi_0(\mathbf{x}) = 1$ 로 정의하여 다음과 같이 위 식을 간단히 나타낼 수 있습니다.

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}), \text{ where } \mathbf{w} = (w_0, \dots, w_{M-1})^T \text{ and } \boldsymbol{\phi} = (\phi_0, \dots, \phi_{M-1})^T$$

# Maximum Likelihood and Least Squares

- Input values  $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_D)^T$ 에 대한 target values  $\mathbf{t} = (t_1, \dots, t_D)^T$ 가 dataset으로 주어졌을 때 다음과 같이 deterministic 함수와 노이즈를 포함한 식을 세워볼 수 있습니다.



$$t = y(\mathbf{x}, \mathbf{w}) + \epsilon$$

- 노이즈 값  $\epsilon$ 이 Gaussian distribution을 따른다 할 때 variance의 역수 precision을  $\beta$ 로 두어 datapoint 하나에 대한 likelihood를 표현할 수 있습니다.



$$p(t | \mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t | y(\mathbf{x}, \mathbf{w}), \beta^{-1})$$

# Maximum Likelihood and Least Squares

- Dataset 안의 각 datapoint들이 각각 independent한 sample이라는 가정하에 dataset 전체의 likelihood를 나타낼 수 있습니다.



$$p(\mathbf{t} | \mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^N \mathcal{N}\left(t_n | \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}\right)$$

- 최적화 단계에서 계산의 편의를 위해 log를 써워 다시 쓰면 다음과 같습니다. 이를 log-likelihood라고 부릅니다.



$$\ln p(\mathbf{t} | \mathbf{w}, \beta) = \sum_{n=1}^N \ln \mathcal{N}\left(t_n | \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}\right)$$

$$= \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi) - \beta E_D(\mathbf{w})$$

$$, \text{ where } E_D(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \left\{ t_n - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n) \right\}^2$$

# Maximum Likelihood and Least Squares

- Log-likelihood를 최대화 하는  $w$ 를 찾기 위해 gradient를 구합니다.

- $\nabla \ln p(t | w, \beta) = \sum_{n=1}^N \left\{ t_n - w^T \phi(x_n) \right\} \phi(x_n)^T$

- Gradient를 0으로 만드는  $w$ 를 구합니다.

- $0 = \sum_{n=1}^N t_n \phi(x_n)^T - w^T \left( \sum_{n=1}^N \phi(x_n) \phi(x_n)^T \right)$

# Maximum Likelihood and Least Squares

- Likelihood를 최대화 하는  $w$ 를 행렬을 통해서 나타내면 다음과 같습니다.

$$\mathbf{w}_{\text{ML}} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{t}$$
$$\Phi = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}$$

- 같은 방법으로 likelihood를 최대화하는  $\beta$ 를 구하면 다음과 같습니다.

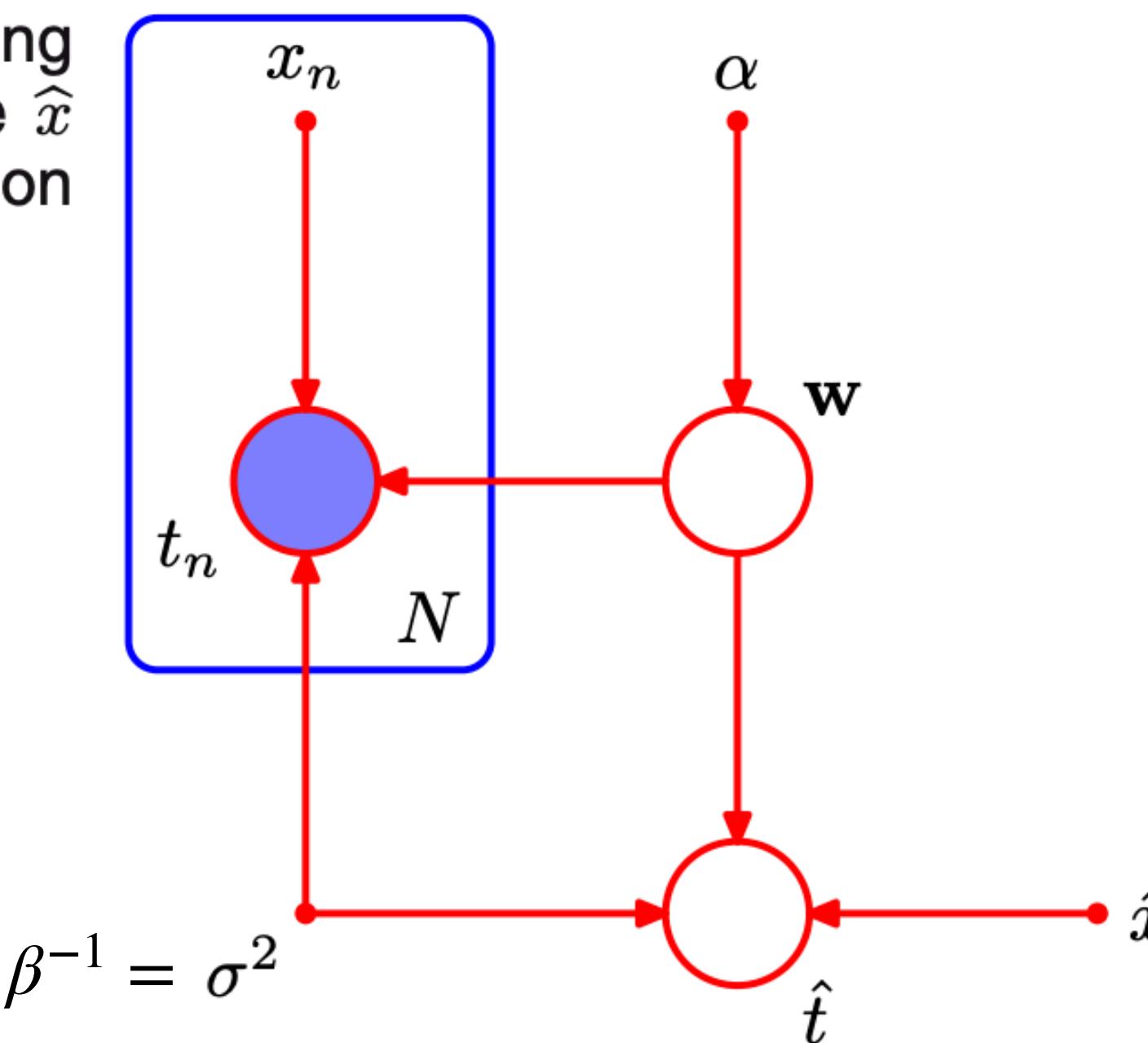
$$\frac{1}{\beta_{\text{ML}}} = \frac{1}{N} \sum_{n=1}^N \left\{ t_n - \mathbf{w}_{\text{ML}}^T \boldsymbol{\phi}(\mathbf{x}_n) \right\}^2$$

# Bayesian Linear Regression

# Bayesian Linear Regression

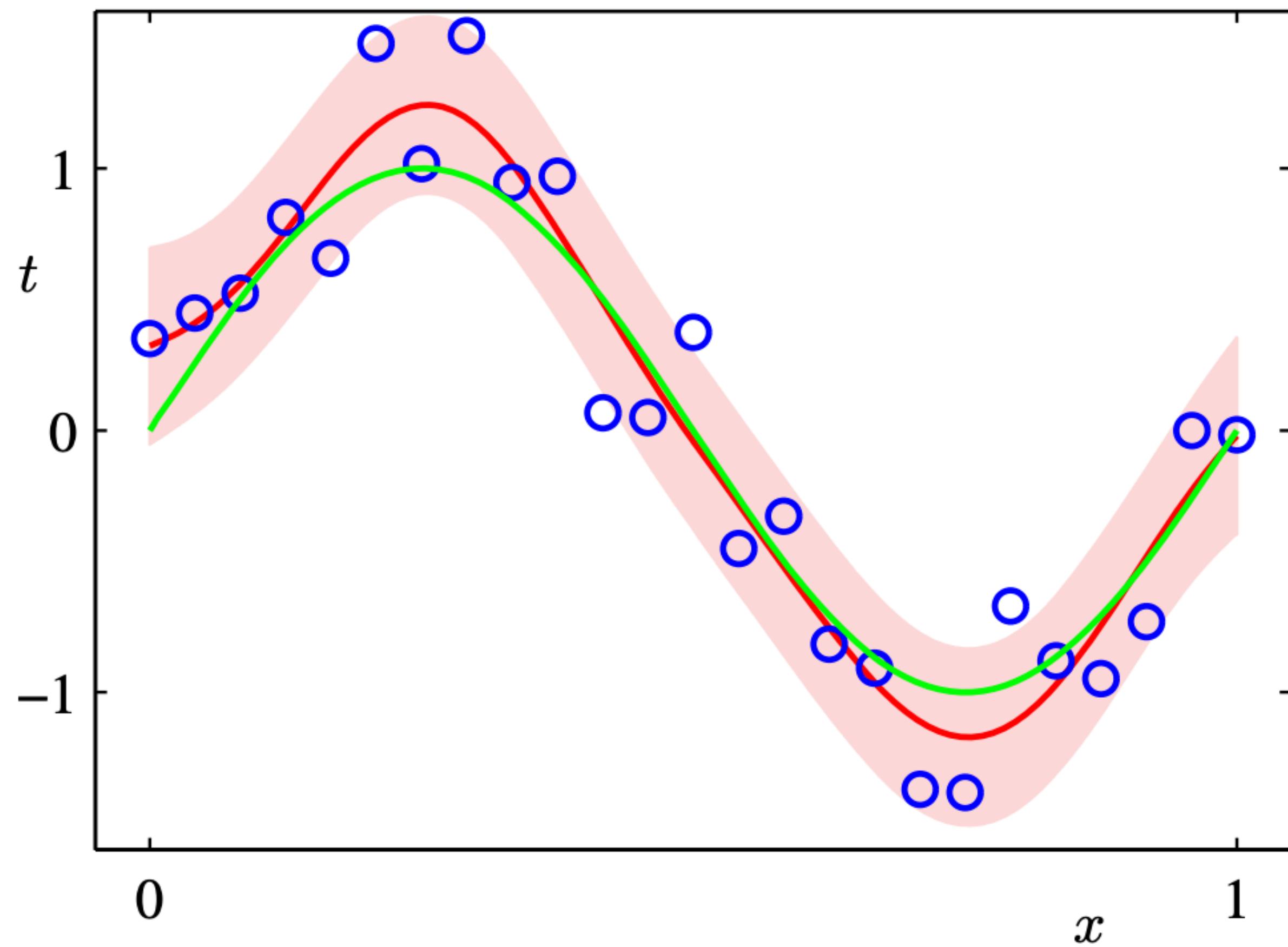
- Bayesian linear regression에서는 weight  $w$ 를 고정된 값으로 두지 않고 random variable로 설정합니다.
- Weight의 prior를 설정하면 데이터가 들어왔을 때 likelihood function과 곱하여 posterior를 구할 수 있습니다.
- 이렇게 구한 posterior를 통해 새로운 입력 샘플이 들어왔을 때 출력의 predictive distribution을 구할 수 있습니다.

**Figure 8.7** The polynomial regression model, corresponding to Figure 8.6, showing also a new input value  $\hat{x}$  together with the corresponding model prediction  $\hat{t}$ .



# Bayesian Linear Regression

- Predictive distribution



# Bayesian Linear Regression

- 기존 머신러닝에서는 linear regression, Gaussian process, probabilistic PCA, linear dynamical system 등 linear Gaussian model을 이용한 많은 모델들이 있습니다.
- Linear Gaussian model을 이용하면 marginal distribution이나 posterior distribution을 구하기가 용이하기 때문입니다.
- 다음 공식들은 random variable  $y$ 가 random variable  $x$ 의 linear transform을 mean으로 하는 Gaussian distribution일 때, marginal distribution과 posterior를 closed form으로 계산한 식입니다.
- 공식을 외울 필요는 없지만 model이 linear Gaussian인 경우 다음과 같이 유용하게 전개됨을 알고 있어야 합니다.

# Bayesian Linear Regression

## Marginal and Conditional Gaussians

Given a marginal Gaussian distribution for  $\mathbf{x}$  and a conditional Gaussian distribution for  $\mathbf{y}$  given  $\mathbf{x}$  in the form

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}) \quad (2.113)$$

$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\mathbf{x} + \mathbf{b}, \mathbf{L}^{-1}) \quad (2.114)$$

the marginal distribution of  $\mathbf{y}$  and the conditional distribution of  $\mathbf{x}$  given  $\mathbf{y}$  are given by

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{L}^{-1} + \mathbf{A}\boldsymbol{\Lambda}^{-1}\mathbf{A}^T) \quad (2.115)$$

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\Sigma}\{\mathbf{A}^T\mathbf{L}(\mathbf{y} - \mathbf{b}) + \boldsymbol{\Lambda}\boldsymbol{\mu}\}, \boldsymbol{\Sigma}) \quad (2.116)$$

where

$$\boldsymbol{\Sigma} = (\boldsymbol{\Lambda} + \mathbf{A}^T\mathbf{L}\mathbf{A})^{-1}. \quad (2.117)$$

# Bayesian Linear Regression Parameter Distribution

- Weight  $\mathbf{w} = (w_0, w_1, \dots, w_D)^T$  가 random variable이라 하고 prior로 mean  $\mathbf{m}_0$  과 covariance matrix  $\mathbf{S}_0$ 인 gaussian distribution을 갖는다 가정합니다.

●

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w} | \mathbf{m}_0, \mathbf{S}_0)$$

- 데이터  $\mathbf{t} = (t_1, \dots, t_D)^T$ 가 주어진 상황에서 posterior는 식(2.116)에 의해 다음과 같이 유도됩니다.

●

$$p(\mathbf{w} | \mathbf{t}) = \mathcal{N}(\mathbf{w} | \mathbf{m}_N, \mathbf{S}_N)$$

where, 
$$\mathbf{m}_N = \mathbf{S}_N (\mathbf{S}_0^{-1} \mathbf{m}_0 + \beta \Phi^T \mathbf{t})$$
$$\mathbf{S}_N^{-1} = \mathbf{S}_0^{-1} + \beta \Phi^T \Phi$$

# Bayesian Linear Regression Parameter Distribution

- prior가 다음과 같이 zero mean과 diagonal covariance matrix를 갖는 경우를 생각해봅시다.

- 

$$p(\mathbf{w} | \alpha) = \mathcal{N}(\mathbf{w} | \mathbf{0}, \alpha^{-1}\mathbf{I})$$

- 이 경우 posterior를 좀 더 간단한 식으로 유도할 수 있습니다.

- 

$$p(\mathbf{w} | \mathbf{t}) = \mathcal{N}(\mathbf{w} | \mathbf{m}_N, \mathbf{S}_N)$$

$$\text{where, } \begin{aligned} \mathbf{m}_N &= \beta \mathbf{S}_N \boldsymbol{\Phi}^T \mathbf{t} \\ \mathbf{S}_N^{-1} &= \alpha \mathbf{I} + \beta \boldsymbol{\Phi}^T \boldsymbol{\Phi} \end{aligned}$$

- 앞에서 얻은 posterior에 log를 씌우고 weight  $\mathbf{w}$ 에 무관한 항들을 constant로 두면 다음 식과 같이 됩니다.

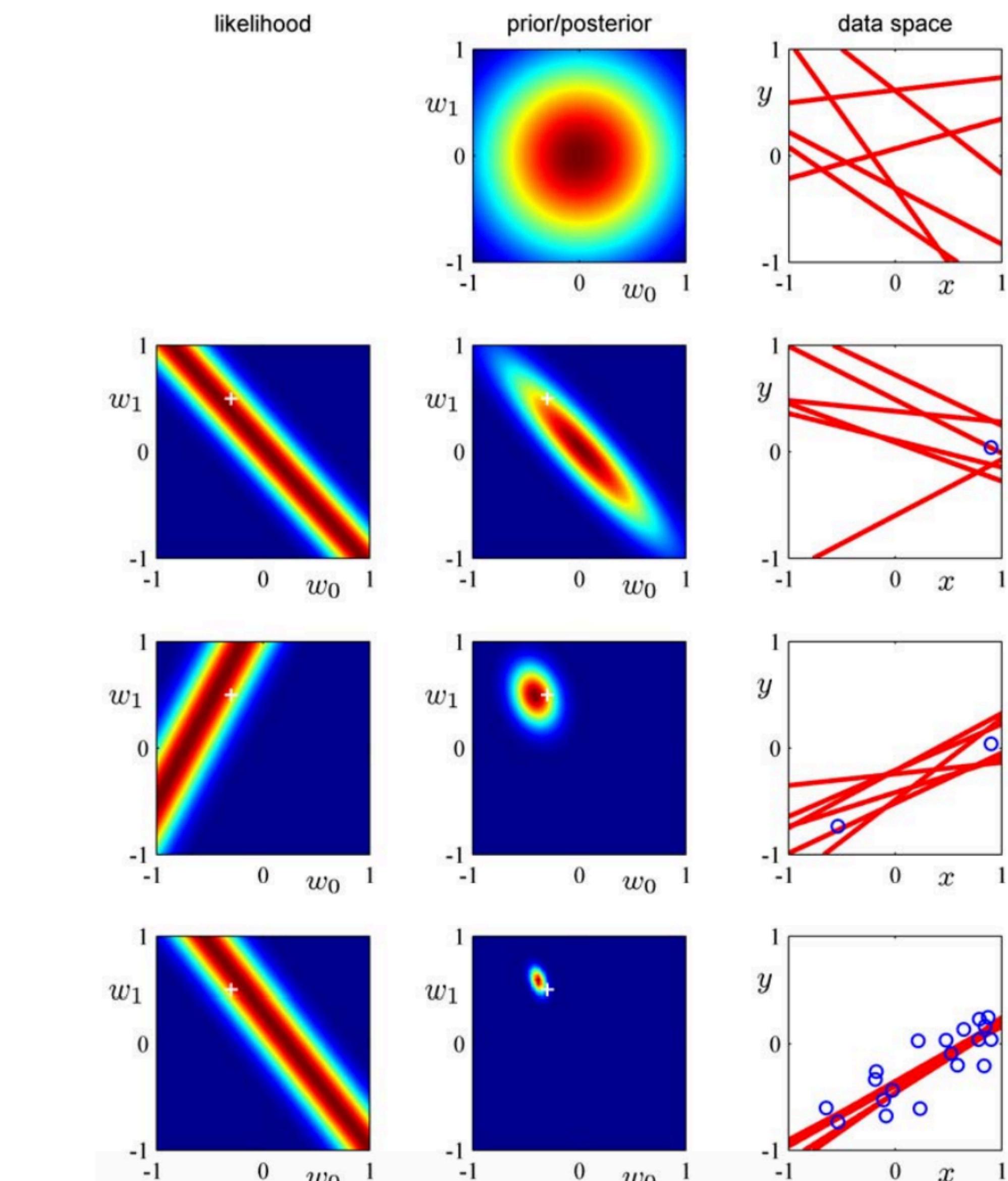
- 

$$\ln p(\mathbf{w} | \mathbf{t}) = -\frac{\beta}{2} \sum_{n=1}^N \left\{ t_n - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n) \right\}^2 - \frac{\alpha}{2} \mathbf{w}^T \mathbf{w} + \text{const.}$$

- Posterior distribution을 최대화하는 것은 L2 regularization이 추가된 sum-of-squares error를 최소화하는 것으로 생각할 수 있습니다.

# Bayesian Linear Regression Parameter Distribution

- 가장 위 줄은 데이터가 주어지지 않았을 때 prior의 분포와 prior에서 샘플링한 weight가 그리는 함수들을 나타냅니다.
- 점차 데이터가 주어짐에 따라 (data space에서의 작은 파란 원) likelihood function과 prior를 곱하여 posterior distribution을 업데이트합니다.
- 이에 따라 점점 posterior의 범위는 좁아지며 posterior로부터 샘플링한 weight가 그리는 data space상의 함수들은 데이터 주위를 지나는 것을 관찰할 수 있습니다.



# Bayesian Linear Regression Predictive Distribution

- Posterior를 이용해서 새로운 데이터  $x$ 가 주어졌을 때의 prediction  $t$ 의 distribution을 다음과 같이 구할 수 있습니다.



$$p(t | \mathbf{t}, \alpha, \beta) = \int p(t | \mathbf{w}, \beta) p(\mathbf{w} | \mathbf{t}, \alpha, \beta) d\mathbf{w}$$

- 직관적 이해를 돋기 위해 이 수식을 expectation식으로 나타내면 다음과 같습니다.



$$p(t | \mathbf{t}, \alpha, \beta) = \mathbb{E}_{p(\mathbf{w} | \mathbf{t}, \alpha, \beta)} [p(t | \mathbf{w}, \beta)]$$

- 또는 샘플링에 의한 근사로 나타내면 다음과 같습니다.



$$p(t | \mathbf{t}, \alpha, \beta) = \sum_{l=1}^L p(t | \mathbf{w}^l, \beta), \mathbf{w}^l \text{ is a sample from } p(\mathbf{w} | \mathbf{t}, \alpha, \beta)$$

# Bayesian Linear Regression Predictive Distribution

- Predictive distribution의 계산은 식 (2.115)에 의해 다음과 같이 유도됩니다.

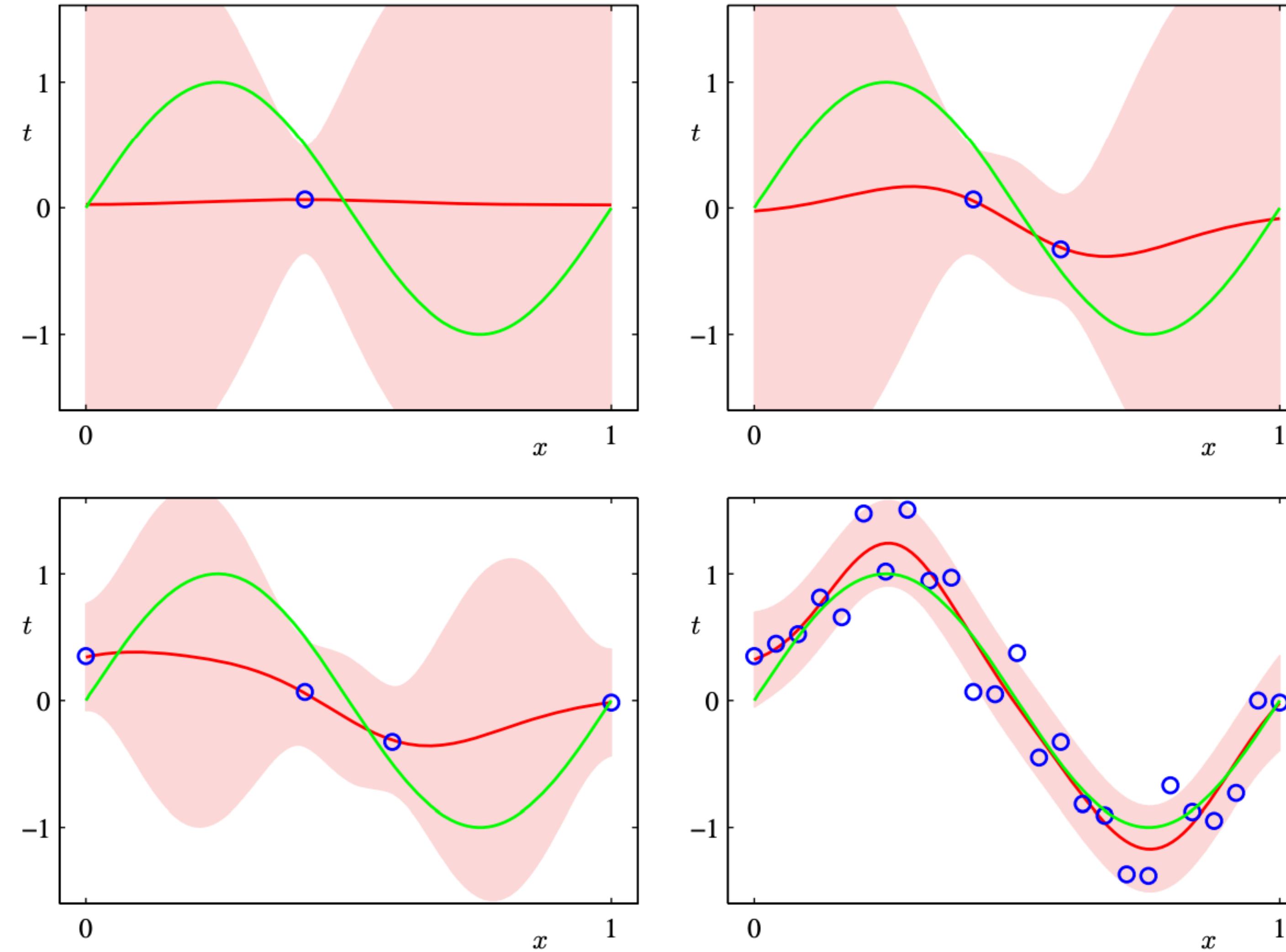
- 

$$p(t | \mathbf{x}, \mathbf{t}, \alpha, \beta) = \mathcal{N} \left( t | \mathbf{m}_N^T \boldsymbol{\phi}(\mathbf{x}), \sigma_N^2(\mathbf{x}) \right)$$

$$\text{where } \sigma_N^2(\mathbf{x}) = \frac{1}{\beta} + \boldsymbol{\phi}(\mathbf{x})^T \mathbf{S}_N \boldsymbol{\phi}(\mathbf{x})$$

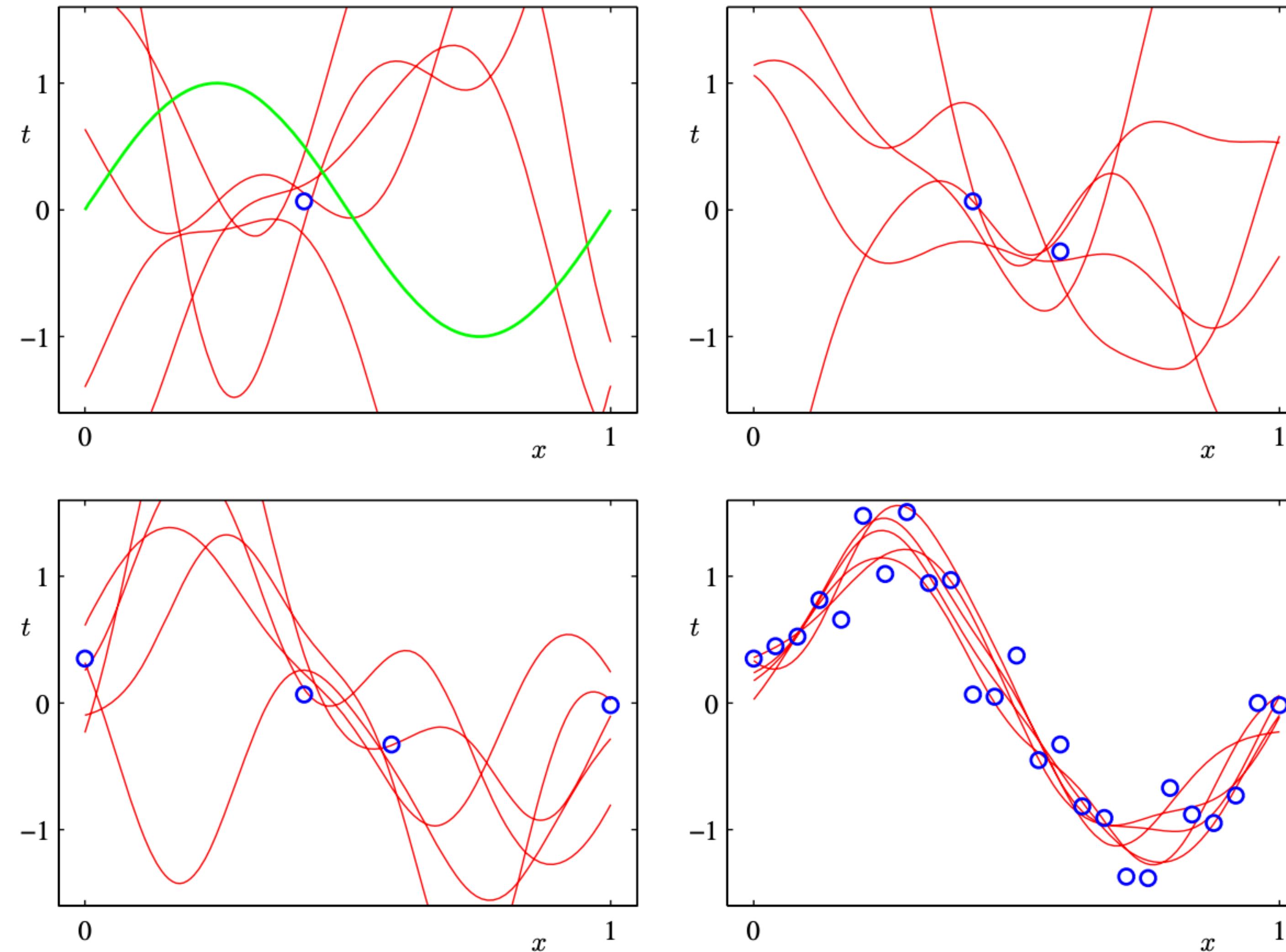
- $\sigma_N^2(\mathbf{x})$ 의 첫번째 항은 데이터 자체의 노이즈를 말하고, 두번째 항은 parameter  $w$ 와 연관되어지는 모델의 불확실성(uncertainty)를 나타냅니다.

# Bayesian Linear Regression Predictive Distribution



**Figure 3.8** Examples of the predictive distribution (3.58) for a model consisting of 9 Gaussian basis functions of the form (3.4) using the synthetic sinusoidal data set of Section 1.1. See the text for a detailed discussion.

# Bayesian Linear Regression Predictive Distribution



**Figure 3.9** Plots of the function  $y(x, w)$  using samples from the posterior distributions over  $w$  corresponding to the plots in Figure 3.8.

# Bayesian Linear Regression Marginal Likelihood

- 모델의 parameter인  $\alpha$ 와  $\beta$ 를 정하기 위해 Maximum Likelihood 방법을 사용합니다.
  - 이를 위해  $\alpha$ 와  $\beta$ 가 condition으로 주어졌을 때 트레이닝 데이터의 likelihood를 구해야 합니다.
  - weight  $w$ 가 고정된 값이 아니라 distribution으로 주어져 있으므로 다음과 같이  $w$ 에 대해 적분하여 marginal likelihood를 구합니다.

$$p(\mathbf{t} \mid \boldsymbol{\alpha}, \boldsymbol{\beta}) = \int p(\mathbf{t} \mid \mathbf{w}, \boldsymbol{\beta}) p(\mathbf{w} \mid \boldsymbol{\alpha}) d\mathbf{w}$$

- 위 식은 expectation식으로 다음과 같이 쓸 수 있습니다.

$$p(\mathbf{t} \mid \boldsymbol{\alpha}, \boldsymbol{\beta}) = \mathbb{E}_{p(\mathbf{w} \mid \boldsymbol{\alpha})}[p(\mathbf{t} \mid \mathbf{w}, \boldsymbol{\beta})]$$

- 위 식은 prior에서 샘플링한 weight로 구한 dataset의 likelihood의 기댓값이 됩니다.

# Bayesian Linear Regression Marginal Likelihood

- 앞서 weight  $w$ 와 data precision  $\beta$ 가 주어졌을 때 dataset  $t$ 의 likelihood  $p(t | w, \beta)$ 와, weight distribution의 precision  $\alpha$ 가 주어졌을 때  $w$ 의  $p(w | \alpha)$ 는 다음과 같이 정했습니다.

●

$$p(t | w, \beta) = \prod_{n=1}^N \mathcal{N}\left(t_n | w^T \Phi(x_n), \beta^{-1}\right), p(w | \alpha) = \mathcal{N}(w | 0, \alpha^{-1}I)$$

- 위 식을 대입해 (2.115)를 이용하여 dataset  $t$ 의 marginal likelihood를 구하면 다음과 같습니다.

●

$$p(t | \alpha, \beta) = \left(\frac{\beta}{2\pi}\right)^{N/2} \left(\frac{\alpha}{2\pi}\right)^{M/2} \int \exp\{-E(w)\} dw$$

$$E(w) = \beta E_D(w) + \alpha E_W(w)$$

where

$$= \frac{\beta}{2} \underbrace{\|t - \Phi w\|^2}_{\text{data error}} + \frac{\alpha}{2} \underbrace{w^T w}_{\text{weight penalty}}$$

# Bayesian Linear Regression Marginal Likelihood

- 적분식을 계산하고 log marginal likelihood를 구하면 다음과 같이 유도됩니다.

- 

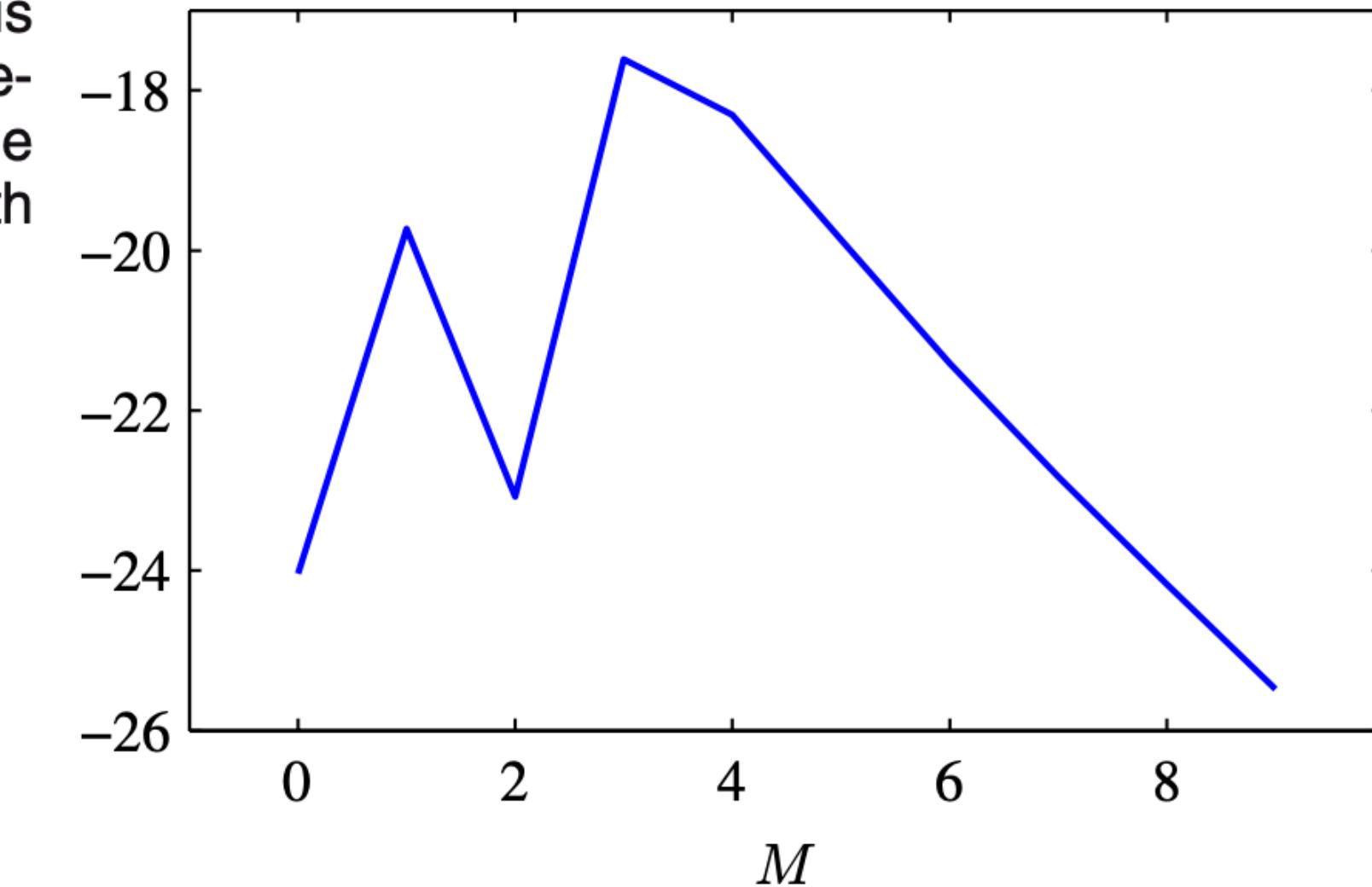
$$\ln p(\mathbf{t} | \alpha, \beta) = \frac{M}{2} \ln \alpha + \frac{N}{2} \ln \beta - E(\mathbf{m}_N) - \frac{1}{2} \ln |\mathbf{A}| - \frac{N}{2} \ln(2\pi)$$

$$\text{where } E(\mathbf{m}_N) = \frac{\beta}{2} \left\| \mathbf{t} - \Phi \mathbf{m}_N \right\|^2 + \frac{\alpha}{2} \mathbf{m}_N^T \mathbf{m}_N \text{ and } \mathbf{A} = \alpha \mathbf{I} + \beta \Phi^T \Phi$$

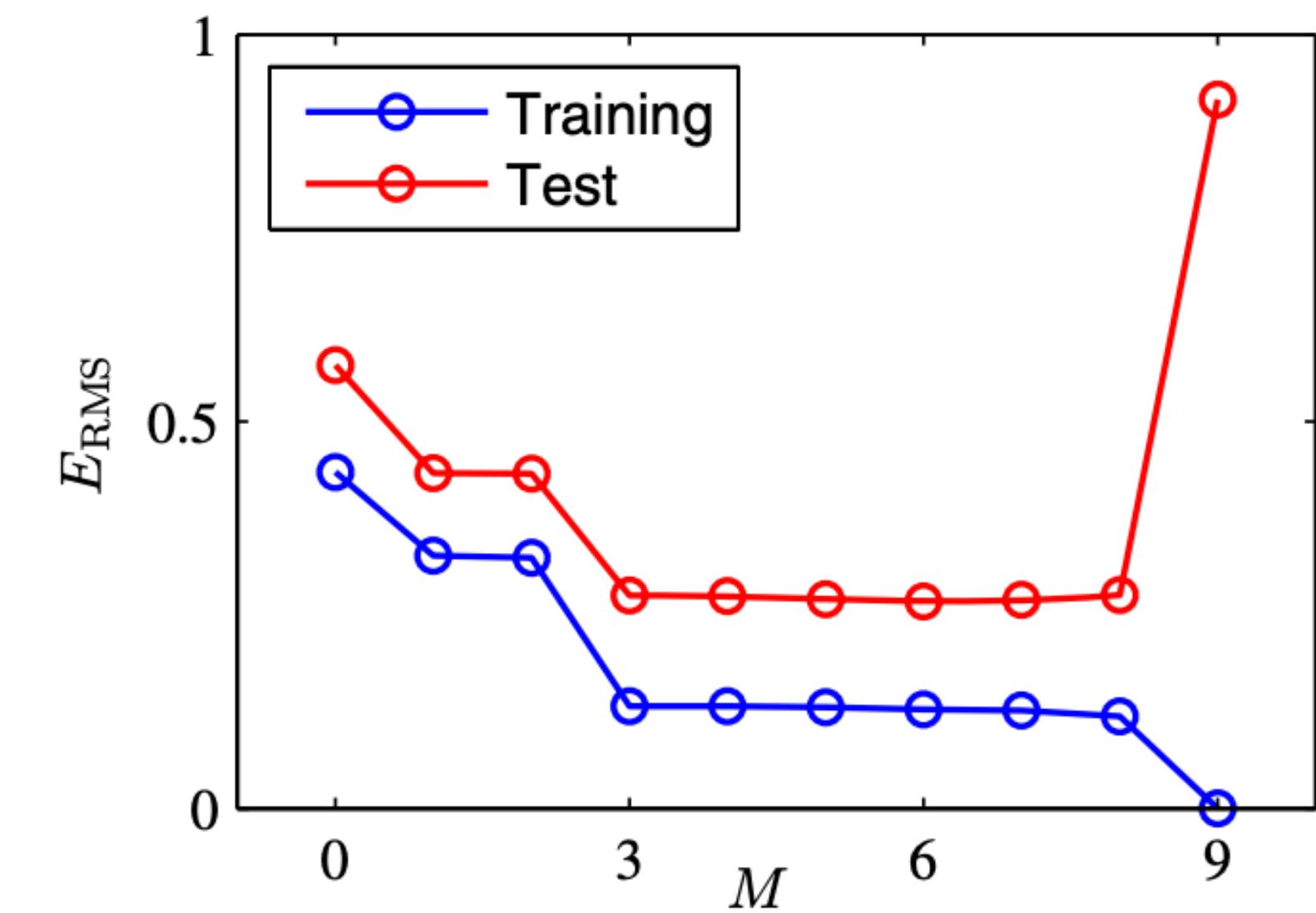
# Bayesian Linear Regression Marginal Likelihood

- Marginal Likelihood 값을 계산하면 데이터에 적합한 polynomial의 차수 즉, 모델 복잡도를 알 수 있습니다.
- 이는 차수가 증가함에 따라 과적합(over-fitting)되어 RMS error가 줄어드는 least square를 이용한 regression방식과는 다른 모습입니다.

**Figure 3.14** Plot of the model evidence versus the order  $M$ , for the polynomial regression model, showing that the evidence favours the model with  $M = 3$ .



**Figure 1.5** Graphs of the root-mean-square error, defined by (1.3), evaluated on the training set and on an independent test set for various values of  $M$ .



# Bayesian Linear Regression Maximum Marginal Likelihood

- 앞서 구한 marginal likelihood 식을 최대화 하는  $\alpha$ 와  $\beta$ 를 구하면 다음과 같습니다.

- 

$$\alpha = \frac{\gamma}{\mathbf{m}_N^T \mathbf{m}_N}$$

$$\frac{1}{\beta} = \frac{1}{N - \gamma} \sum_{n=1}^N \left\{ t_n - \mathbf{m}_N^T \boldsymbol{\phi}(\mathbf{x}_n) \right\}^2$$

$$\gamma = \sum_i \frac{\lambda_i}{\alpha + \lambda_i}, \text{ where } \lambda_i \text{ are eigenvalues of } \beta \boldsymbol{\Phi}^T \boldsymbol{\Phi}$$

- $\alpha$ 와  $\beta$ 는 closed form 으로 구해지지 않으며 초기화 후 iteration을 통해 구할 수 있습니다.

- iteration시  $\alpha$ 와  $\beta$ 뿐 아니라, 이를 얻기 위해 필요한 weight posterior의 mean  $\mathbf{m}_N$ , covariance matrix  $\mathbf{S}_N$ 도 함께 변경됩니다.

# Fully Bayesian Linear Regression

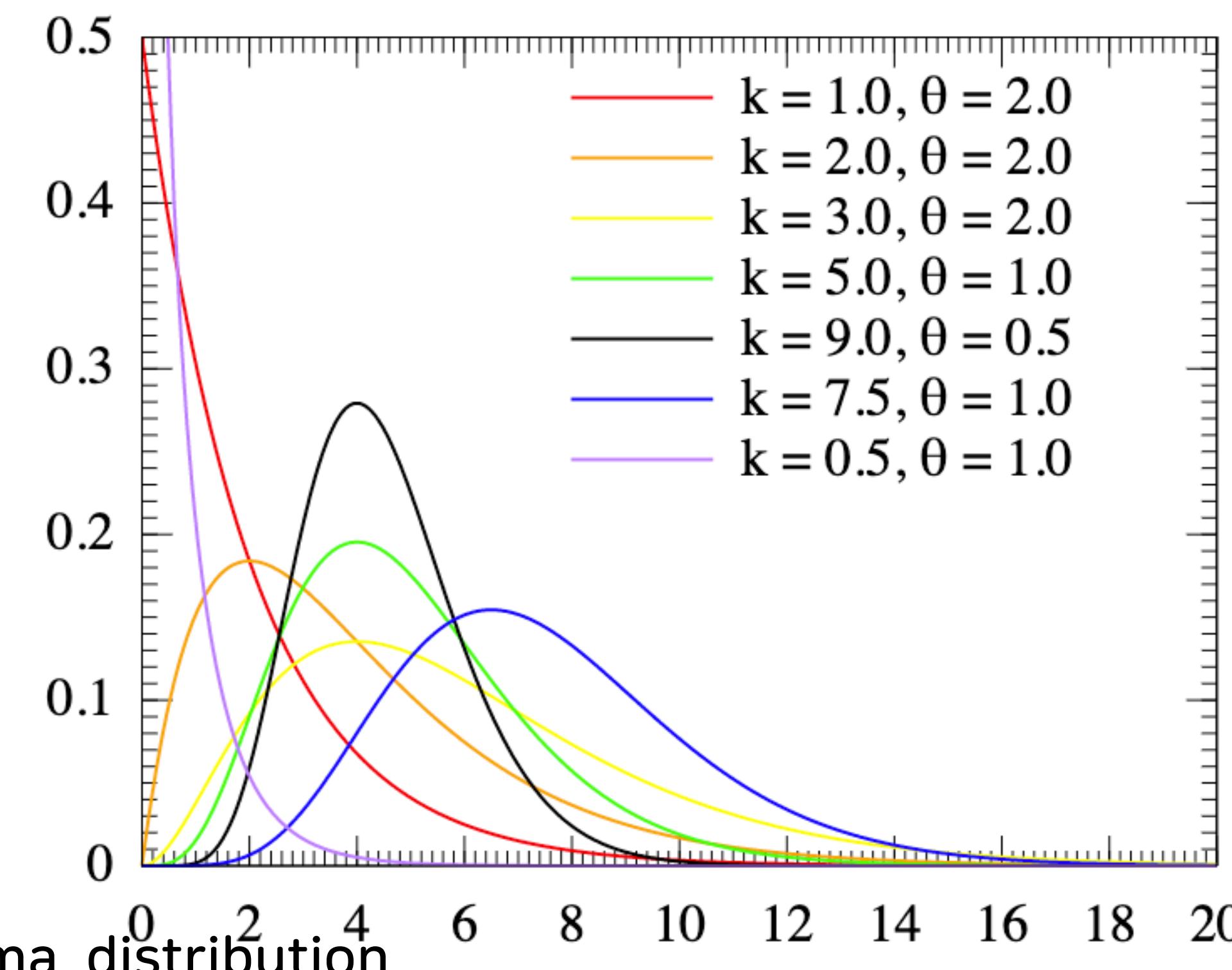
- 앞선 Bayesian linear regression 모델은 weight 분포의 파라메터 precision  $\alpha$ 와 data의 precision  $\beta$ 를 고정된 값으로 두었습니다.
- 이보다 한걸음 더 나아가  $\alpha$ 와  $\beta$ 값 또한 고정시키지 않고 확률분포로 정하여 더욱 유연한 모델을 만들어 볼 수 있습니다.
- Scikit-learn의 `sklearn.linear_model.BayesianRidge` class에서 이 모델을 구현하고 있습니다.
- $\alpha$ 와  $\beta$ 값은 precision을 나타내며 이러한 종류의 값은 conjugate prior로 gamma distribution을 갖습니다.

# Fully Bayesian Linear Regression (gamma distribution)

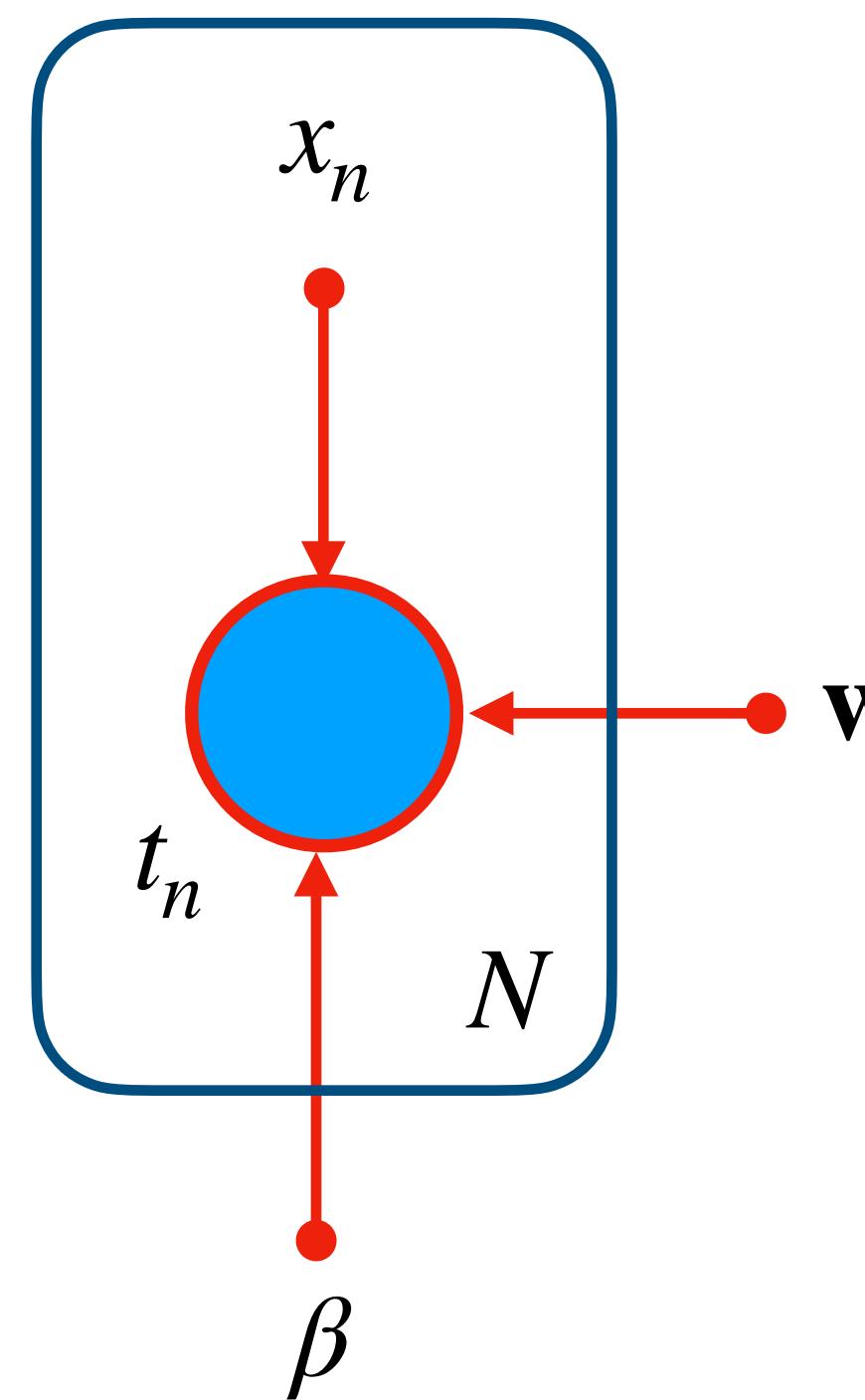
- Gamma distribution은 variance나 precision에 대한 conjugate prior입니다.

$$p(x) = \frac{1}{\Gamma(k)\theta^k} x^{k-1} e^{-\frac{x}{\theta}}$$

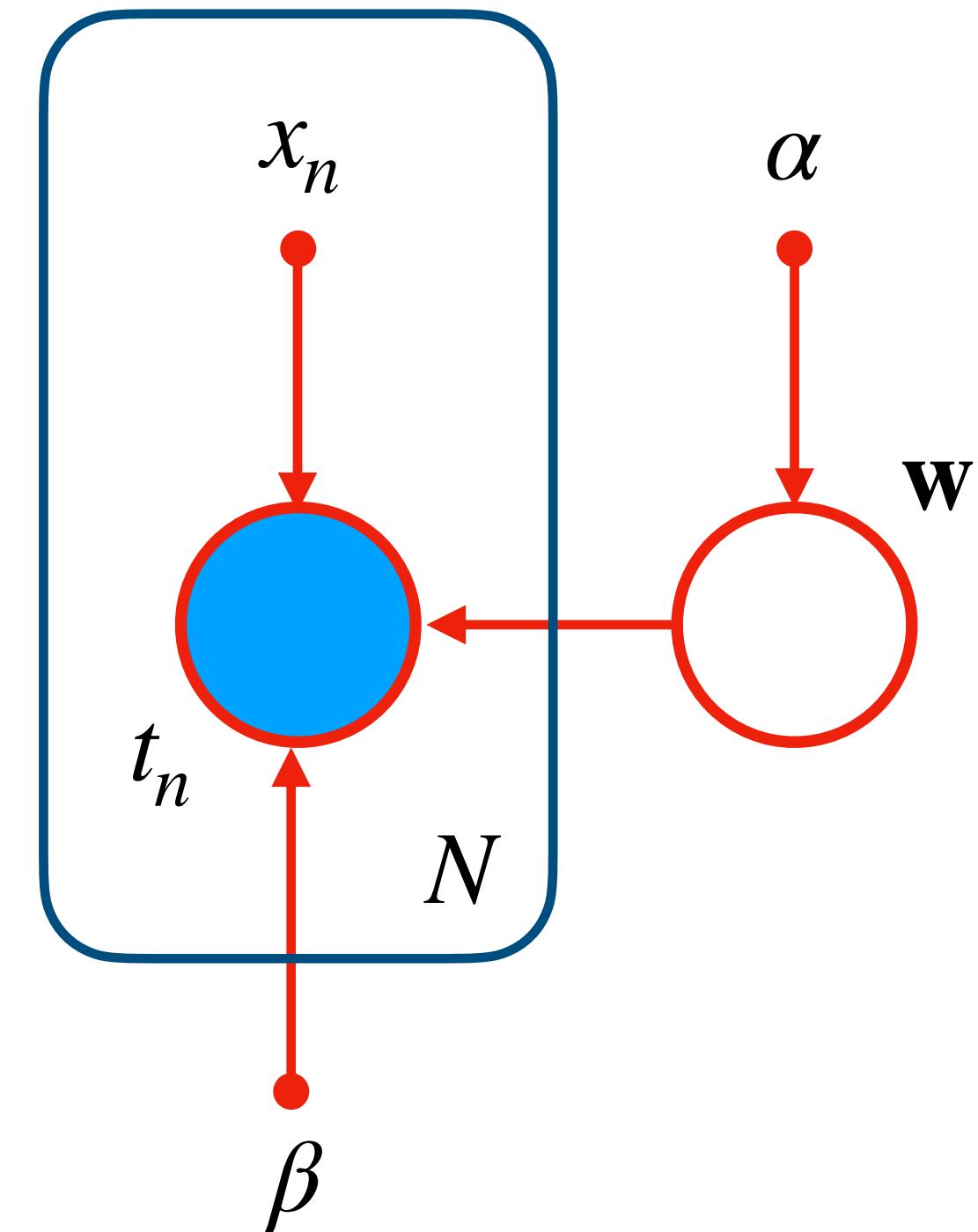
- parameter로는 shape을 결정하는  $k$ 와 scale을 결정하는  $\theta$ 가 있습니다.



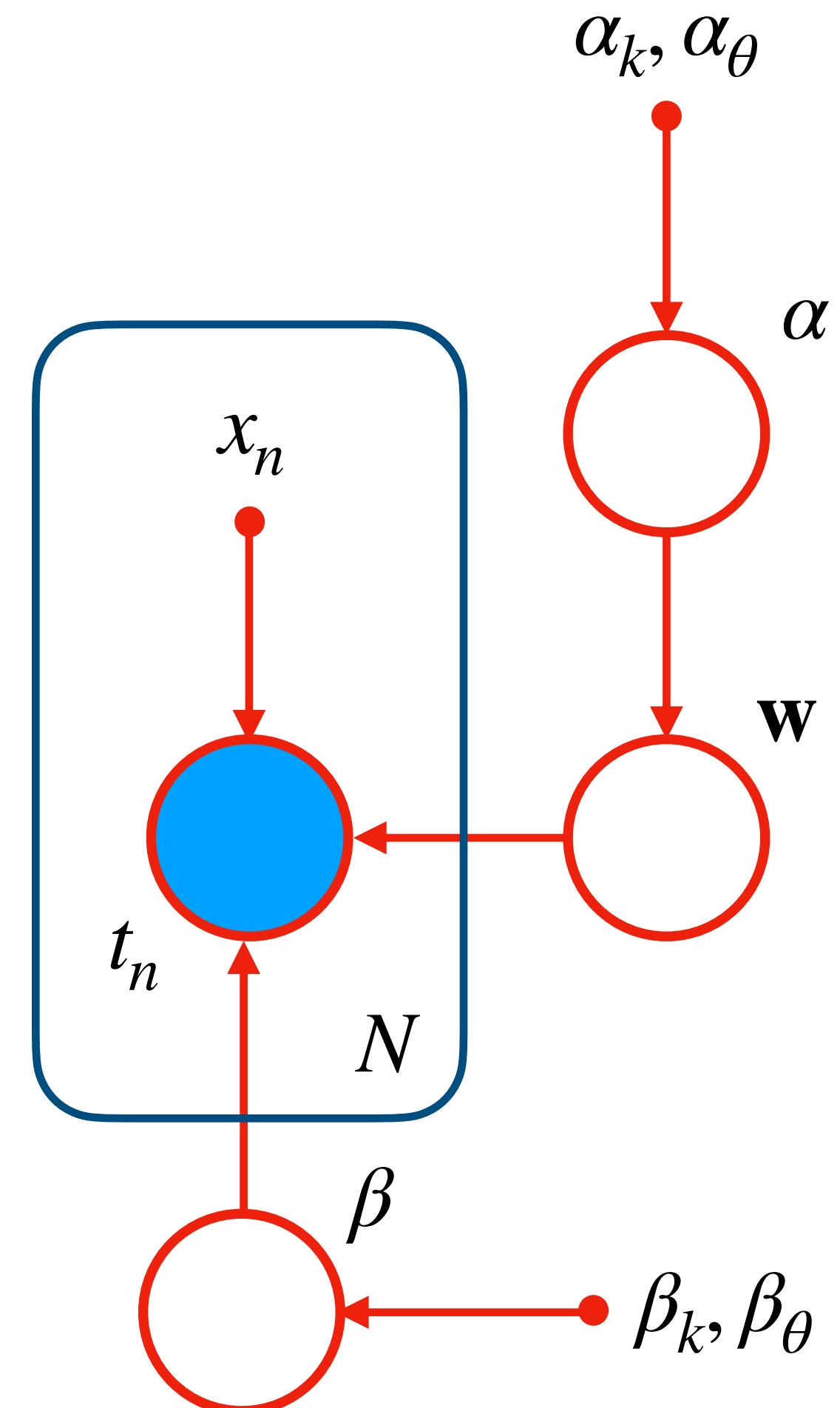
# Linear Regression, Bayesian, Fully Bayesian



Linear Regression  
(point estimation)



Bayesian Linear Regression  
(empirical Bayes)



Fully-Bayesian  
Linear Regression