# Intro. to Bayesian methods

MIAN HUANG

# Preliminary

## Rules of probability

Sum rule: $$p(x) = \int p(x, y)\, dy$$

Product rule: $$p(x, y) = p(y|x)p(x)$$

Bayes' rule: $$p(y|x) = \frac{p(x,y)}{p(x)} = \frac{p(x|y)p(y)}{p(x)}$$

Apply them on parameters.

# Bayesian Regression

## Probabilistic model for regression

Likelihood: $p(y \mid \mathbf{x}, \mathbf{w}) = \mathcal{N}(y;\ f(\mathbf{x}; \mathbf{w}),\ \sigma_y^2)$

Any regression model to fit

Variance/noise

Negative Log Likelihood:

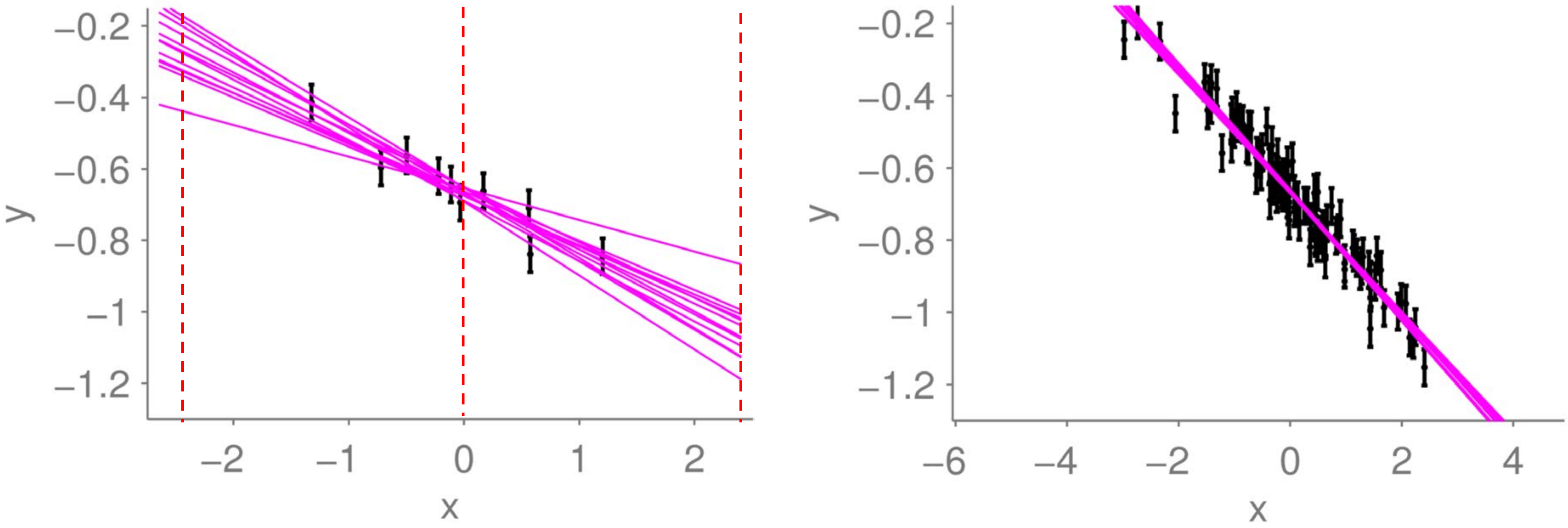
$$-\log p(\mathbf{y} \mid X, \mathbf{w}) \overset{\text{i.i.d.}}{=} -\sum_n \log p(y^{(n)} \mid \mathbf{x}^{(n)}, \mathbf{w})$$

$$= \frac{1}{2\sigma_y^2} \sum_{n=1}^{N} \underbrace{\boxed{\left[ (y^{(n)} - f(\mathbf{x}^{(n)}; \mathbf{w}))^2 \right]}}_{\text{Square error}} + \frac{N}{2} \log(2\pi\sigma_y^2)$$

**MLE w.r.t. w == *min*(square error)**

# Bayesian Regression

## Uncertainty of models



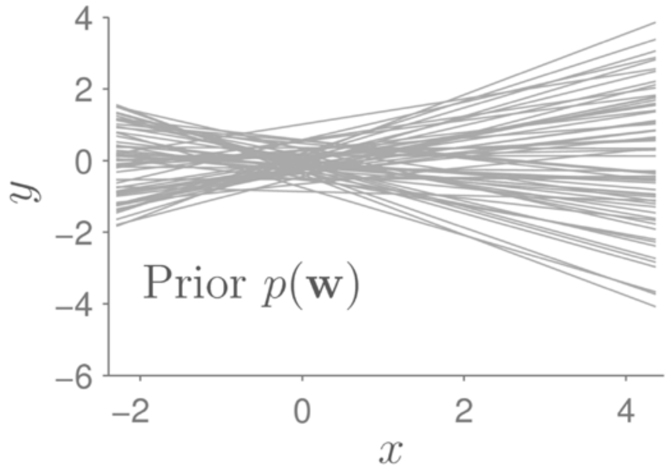
# Bayesian Regression

## Reasoning model parameters

Model: e.g. $f(x;\mathbf{w}) = w_1 x + w_2$

The Prior: e.g. $p(\mathbf{w}) = \mathcal{N}(\mathbf{w};\, \mathbf{0}, 0.4^2\mathbb{I})$

Data: $\mathcal{D} = \{\mathbf{x}^{(n)}, y^{(n)}\}$




**The Posterior Update:** Bayes' rule

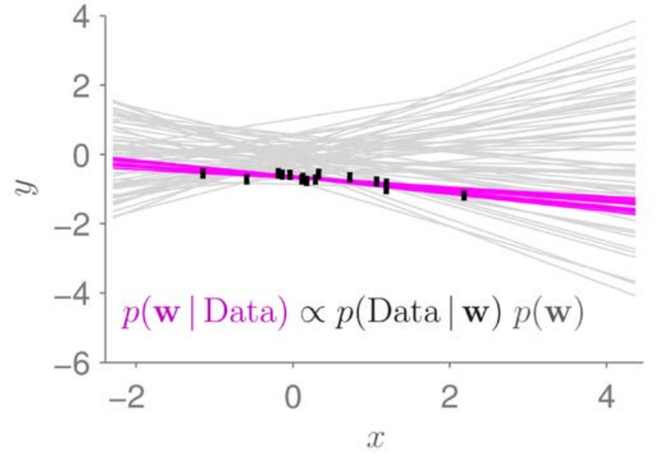
Likelihood

$$p(\mathbf{w}\,|\,\mathcal{D}) = p(\mathbf{w}\,|\,\mathbf{y}, X) = \frac{p(\mathbf{y}\,|\,\mathbf{w}, X)\,p(\mathbf{w})}{p(\mathbf{y}\,|\,X)} \propto p(\mathbf{y}\,|\,\mathbf{w}, X)\,p(\mathbf{w})$$

**MAP =** ***max*****(likelihood + prior) =** ***min*****(square error + regularization)**

e.g., Gaussian

e.g., L2



# Example: Likelihood vs. Posterior

Underlying Bernoulli model:

$$p(x;\theta) = \theta^x (1-\theta)^{1-x} \quad \theta = \frac{1}{3}$$

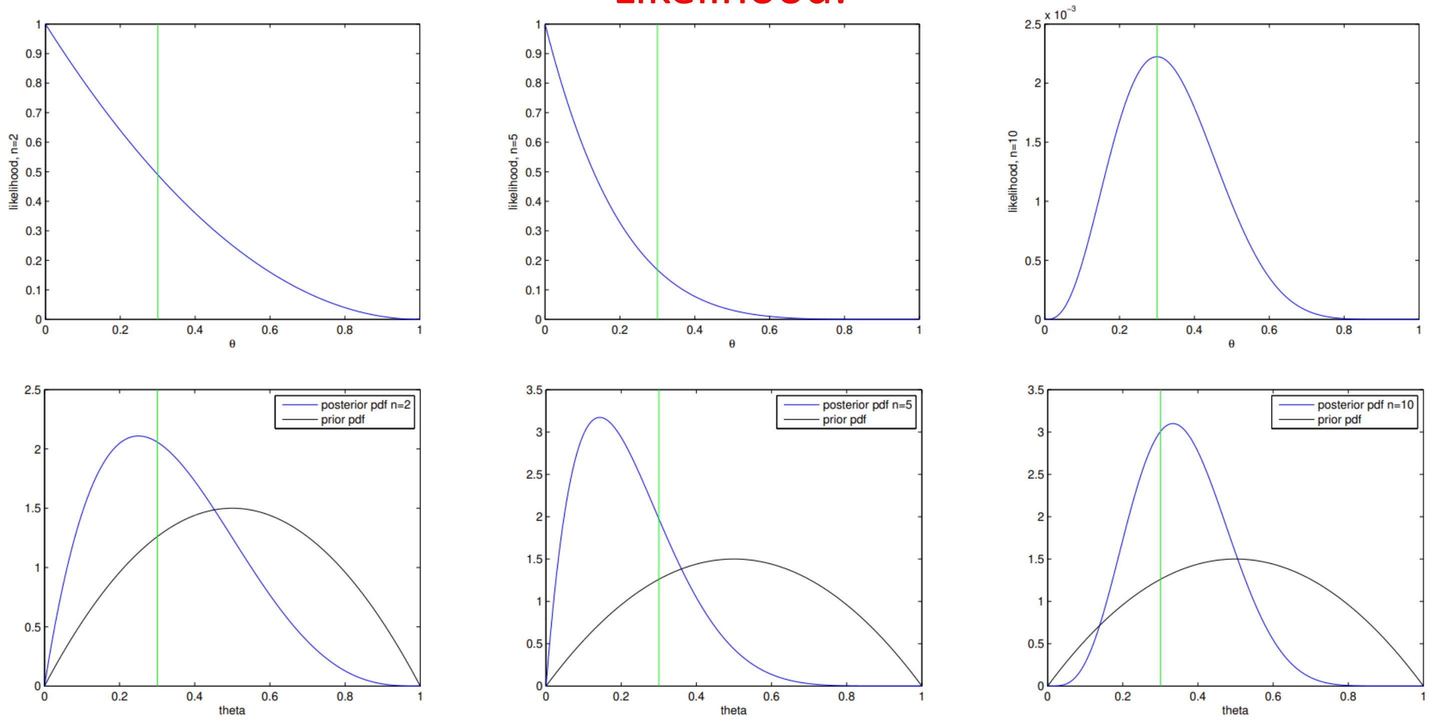
MLE estimate:

$$\hat{\theta} = \frac{n_{x=1}}{n}$$

$$\mathcal{D} = (0,0,0,0,0,0,0,1,1,1,\ldots)$$

**Pull towards the prior**

Likelihood:




(a) $n = 2$ observations (b) $n = 5$ observations (c) $n = 10$ observations

Posterior:

# Bayesian Prediction

## Prediction

Latent variable (e.g. model params)

Prediction of y: $$p(y \,|\, \mathbf{x}, \mathcal{D}) = \int p(y, \mathbf{w} \,|\, \mathbf{x}, \mathcal{D}) \, \mathrm{d}\mathbf{w}$$ Sum rule

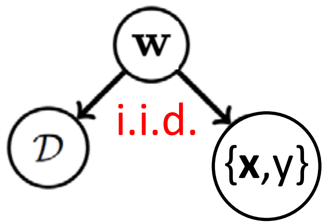
$$= \int p(y \,|\, \mathbf{x}, \mathbf{w}) \, p(\mathbf{w} \,|\, \mathcal{D}) \, \mathrm{d}\mathbf{w}$$ Product rule

Likelihood Posterior of params

$$= \mathbb{E}_{p(\mathbf{w} \,|\, \mathcal{D})} [P(y \,|\, \mathbf{x}, \mathbf{w})]$$

**Likelihood weighted by posterior**

Visualisation as a DAG:




However, often **intractable** unless integral in closed form (e.g., Gaussian)

**Approximate** (e.g., Laplace, VI, MCMC)

# Bayesian Prediction

## Decision making

Actual output estimate

Square loss

Cost function: $c = \mathbb{E}_{p(y \mid \mathbf{x}, \mathcal{D})}[L(y, \hat{y})]$

e.g. $L(y, \hat{y}) = (y - \hat{y})^2$

$$\frac{\partial c}{\partial \hat{y}} = \mathbb{E}_{p(y \mid \mathbf{x}, \mathcal{D})}[-2(y - \hat{y})] = -2(\mathbb{E}_{p(y \mid \mathbf{x}, \mathcal{D})}[y] - \hat{y})$$

$$\hat{y} = \mathbb{E}_{p(y \mid \mathbf{x}, \mathcal{D})}[y]$$

**Prediction posterior mean**

Multiple decisions:

Separates **modelling data** from the application-specific **loss function**

# Bayesian Model Choice

## Cross-validation → Marginal likelihood

**Marginal** likelihood (sum up $\mathbf{w}$ ):

Model (e.g., hyper-params, linear regression/NN)

$$p(\mathbf{y}\,|\,X,\mathcal{M}) = \int \underbrace{p(\mathbf{y},\mathbf{w}\,|\,X,\mathcal{M})}_{\boxed{\text{Sum rule}}}\,\mathrm{d}\mathbf{w} = \int \underbrace{p(\mathbf{y}\,|\,X,\mathbf{w},\mathcal{M})}_{\text{Likelihood}}\,\underbrace{p(\mathbf{w}\,|\,\mathcal{M})}_{\text{Prior}}\,\mathrm{d}\mathbf{w}. \qquad \text{(For all } \mathbf{w}\text{)}$$

Product rule

(1) To score each model chosen, no need for held-out set.

(2) Can explain overfitting.

# Bayesian Optimization

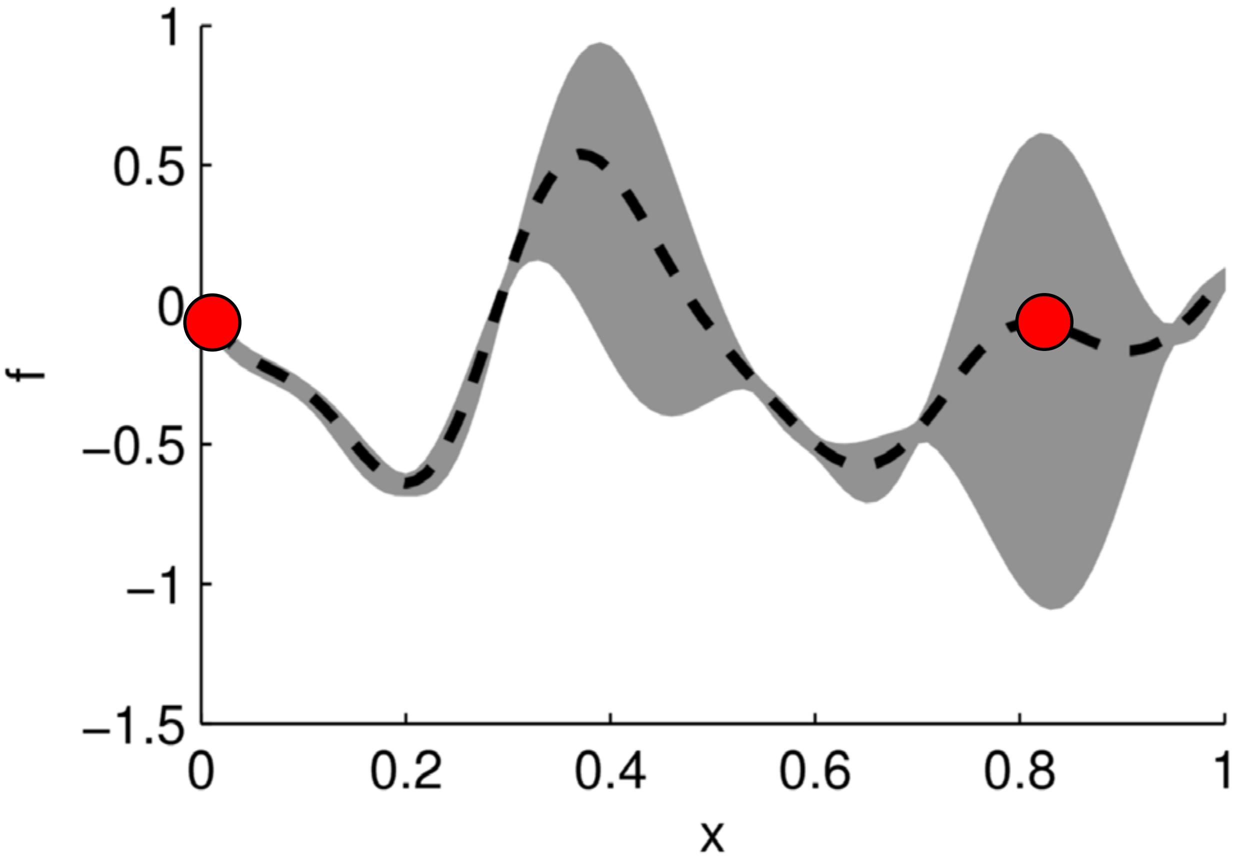
## Use uncertainty to gather data

Parameter-free

Gradient-free

e.g., hyper-params optimization

Few data, Efficient



# Bayesian Optimization

The only weak assumption

## Surrogate model: Gaussian Process (GP)

Infinite dimensional

(If no domain knowledge)

Kernel (positive definite)

GP prior: $p(\mathbf{f}) = \mathcal{N}(\mathbf{f}; \mathbf{0}, K)$

where $f_i = f(\mathbf{x}^{(i)})$ and $K_{ij} = k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$

[1] Rule of conditional Gaussian

$$p(\mathbf{f}, \mathbf{g}) = \mathcal{N}\left(\begin{bmatrix}\mathbf{f}\\ \mathbf{g}\end{bmatrix}; \begin{bmatrix}\mathbf{a}\\ \mathbf{b}\end{bmatrix}, \begin{bmatrix}A & C\\ C^\top & B\end{bmatrix}\right)$$

$$p(\mathbf{f} \mid \mathbf{g}) = \mathcal{N}(\mathbf{f};\ \mathbf{a} + CB^{-1}(\mathbf{g} - \mathbf{b}),\ A - CB^{-1}C^\top)$$

noise

Observation/Data: $y_i \sim \mathcal{N}(f_i, \sigma_y^2)$

Joint: $$p\left(\begin{bmatrix}\mathbf{y}\\ \mathbf{f}_*\end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{y}\\ \mathbf{f}_*\end{bmatrix};\ \mathbf{0}, \begin{bmatrix}K(X,X) + \sigma_y^2\mathbb{I} & K(X, X_*)\\ K(X_*, X) & K(X_*, X_*)\end{bmatrix}\right)$$

Point to predict

<u>Marginalize **y** [1]:</u>

Posterior/Prediction: $p(\mathbf{f}_* \mid \mathbf{y}) = \mathcal{N}(f;\ m,\ s^2)$

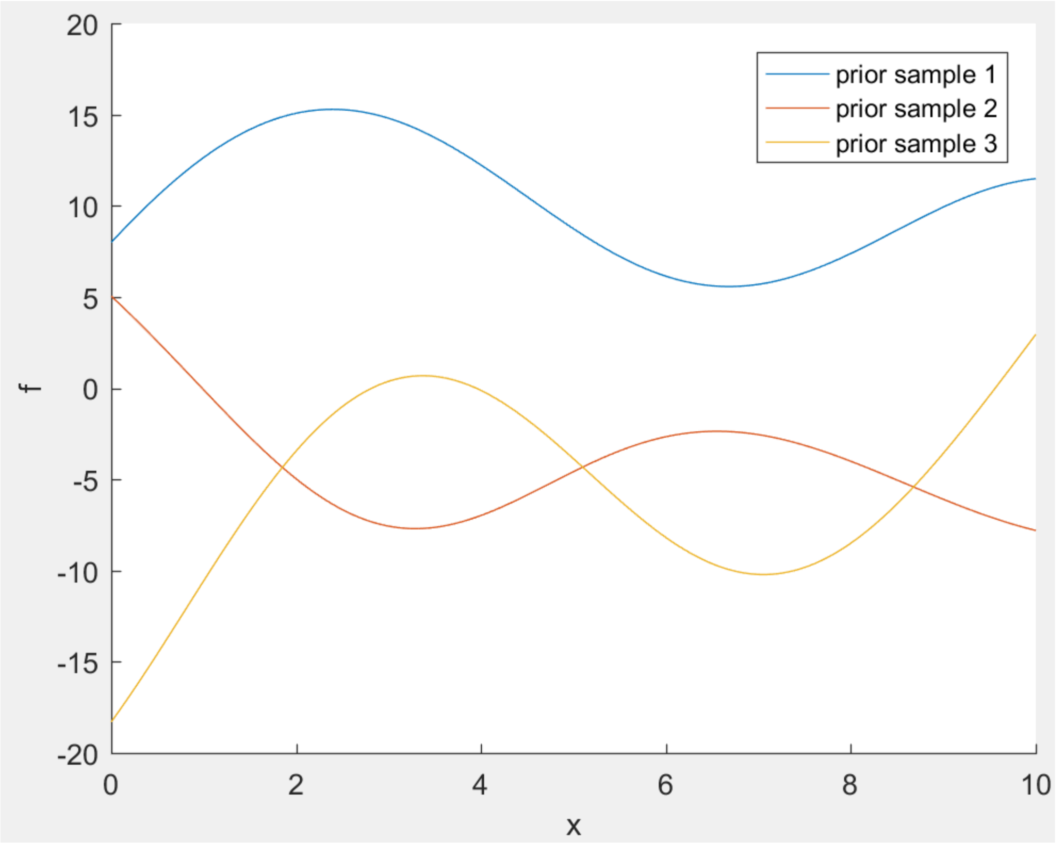
$$M = K + \sigma_y^2\mathbb{I},$$
$$m = \mathbf{k}^{(*)\top} M^{-1}\mathbf{y},$$
$$s^2 = k(\mathbf{x}^{(*)}, \mathbf{x}^{(*)}) - \mathbf{k}^{(*)\top} M^{-1}\mathbf{k}^{(*)}$$

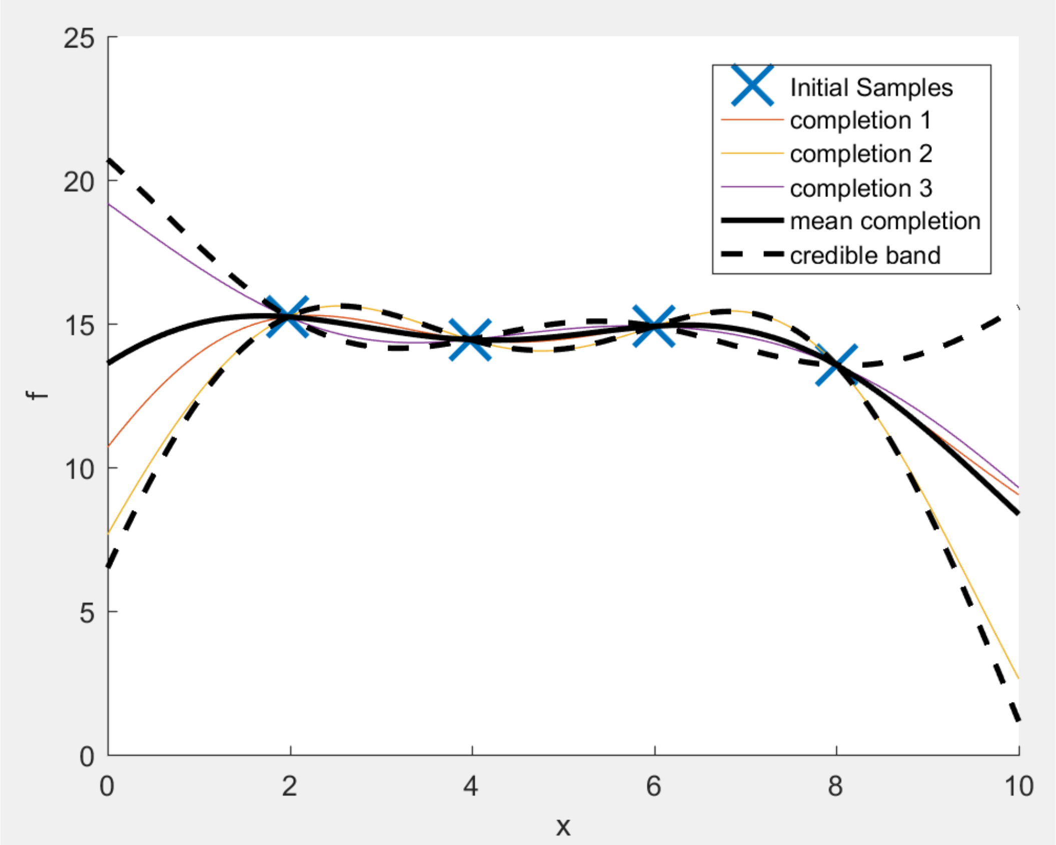
1. No dependence to y
2. Positive

**Know how certain from when**

# Bayesian Optimization

## GP Visualization



# Bayesian Optimization

## Acquisition function

**Objective:** $\boldsymbol{x}^*_{next} = argmax_{\boldsymbol{x}^*}(a(\boldsymbol{x}^*))$

Exploitation-exploration trade-off

**e.g., Upper confidence bound (UCB):** $a(\boldsymbol{x}^*) = \mu^* + \kappa\sigma^*$

Probability of improvement (PI)

Expected improvement (EI)

……

Reference:
*Jasper Snoek et.al. ''Practical bayesian optimization of machine learning algorithms.'' NIPS (2012)*

# Bayesian Optimization

## GP:

**Pros:**

Only a weak assumption

Can model expensive function (few data): e.g., optimize hyper-params of NN

Report uncertainty

**Cons:**

Scale poorly with large datasets:

(1) $M^{-1}$ needs $O(N^3)$ complexity

(2) $K$ needs $O(N^2)$ memory

# Dealing with intractable integral

## Recall: Bayesian Prediction

Likelihood

Prediction of y: $p(y \mid \mathbf{x}, \mathcal{D}) = \int p(y, \mathbf{w} \mid \mathbf{x}, \mathcal{D}) \, \mathrm{d}\mathbf{w} = \int p(y \mid \mathbf{x}, \mathbf{w}) \boxed{p(\mathbf{w} \mid \mathcal{D})} \, \mathrm{d}\mathbf{w}$

Posterior of params **(intractable)**

1. Approximate directly (Laplace approx., Variational method, etc.)
2. Monte-Carlo estimate

# Dealing with intractable integral

## 1. Approximate directly: Variational method

A family of possible distributions (e.g., Gaussian, Exponential, NN, etc.):

$$q(\mathbf{w}; \alpha)$$

$D_{KL}$: a measure of the discrepancy between two distributions

$$D_{\mathrm{KL}}(p \,||\, q) = \int p(\mathbf{z}) \log \frac{p(\mathbf{z})}{q(\mathbf{z})} \, \mathrm{d}\mathbf{z}$$

# Dealing with intractable integral

## 1. Approximate directly: Variational method

Inference → **Optimization**:

$$D_{\mathrm{KL}}(q(\mathbf{w};\alpha)\,||\,p(\mathbf{w}\,|\,\mathcal{D})) = \int q(\mathbf{w};\alpha)\log\frac{q(\mathbf{w};\alpha)}{p(\mathbf{w}\,|\,\mathcal{D})}\,\mathrm{d}\mathbf{w}$$
$$= -\int q(\mathbf{w};\alpha)\log p(\mathbf{w}\,|\,\mathcal{D})\,\mathrm{d}\mathbf{w} + \int q(\mathbf{w};\alpha)\log q(\mathbf{w};\alpha)\,\mathrm{d}\mathbf{w}$$

$$\boxed{p(\mathbf{w}\,|\,\mathcal{D}) = \frac{p(\mathcal{D}\,|\,\mathbf{w})\,p(\mathbf{w})}{p(\mathcal{D})}}$$

Neg. Entropy

Cross-entropy

$$D_{\mathrm{KL}}(q\,||\,p) = \boxed{\mathrm{E}_q[\log q(\mathbf{w})] - \mathrm{E}_q[\log p(\mathcal{D}\,|\,\mathbf{w})] - \mathrm{E}_q[\log p(\mathbf{w})]} + \log p(\mathcal{D}) \geq 0$$

Lower bound (e.g., SVI, Black-box VI)

More reference:

*Blei, David M. et.al. "Variational Inference: A Review for Statisticians." JASA (2017)*

# Dealing with intractable integral

## 2. Monte-Carlo Estimate: MCMC

Prediction of y:
$$\begin{aligned} P(y \mid \mathbf{x}, \mathcal{D}) &= \int P(y \mid \mathbf{x}, \mathbf{w})\, p(\mathbf{w} \mid \mathcal{D})\, \mathrm{d}\mathbf{w} \\ &= \mathbb{E}_{p(\mathbf{w} \mid \mathcal{D})}\left[P(y \mid \mathbf{x}, \mathbf{w})\right] \\ &\approx \frac{1}{S} \sum_{s=1}^{S} P(y \mid \mathbf{x}, \mathbf{w}^{(s)}), \quad \mathbf{w}^{(s)} \sim p(\mathbf{w} \mid \mathcal{D}) \end{aligned}$$

**Metropolis-Hastings:** Generate samples from $p(\mathbf{w} \mid \mathcal{D})$ using random walk.

**Require**:
Proposal distribution parameterized by previous state: e.g., $q\left(\boldsymbol{w}; \boldsymbol{w}^{(t-1)}\right) = N(\boldsymbol{w}; \boldsymbol{w}^{(t-1)}, \epsilon^2)$
Function proportional to $p(\mathbf{w} \mid \mathcal{D})$

# Dealing with intractable integral

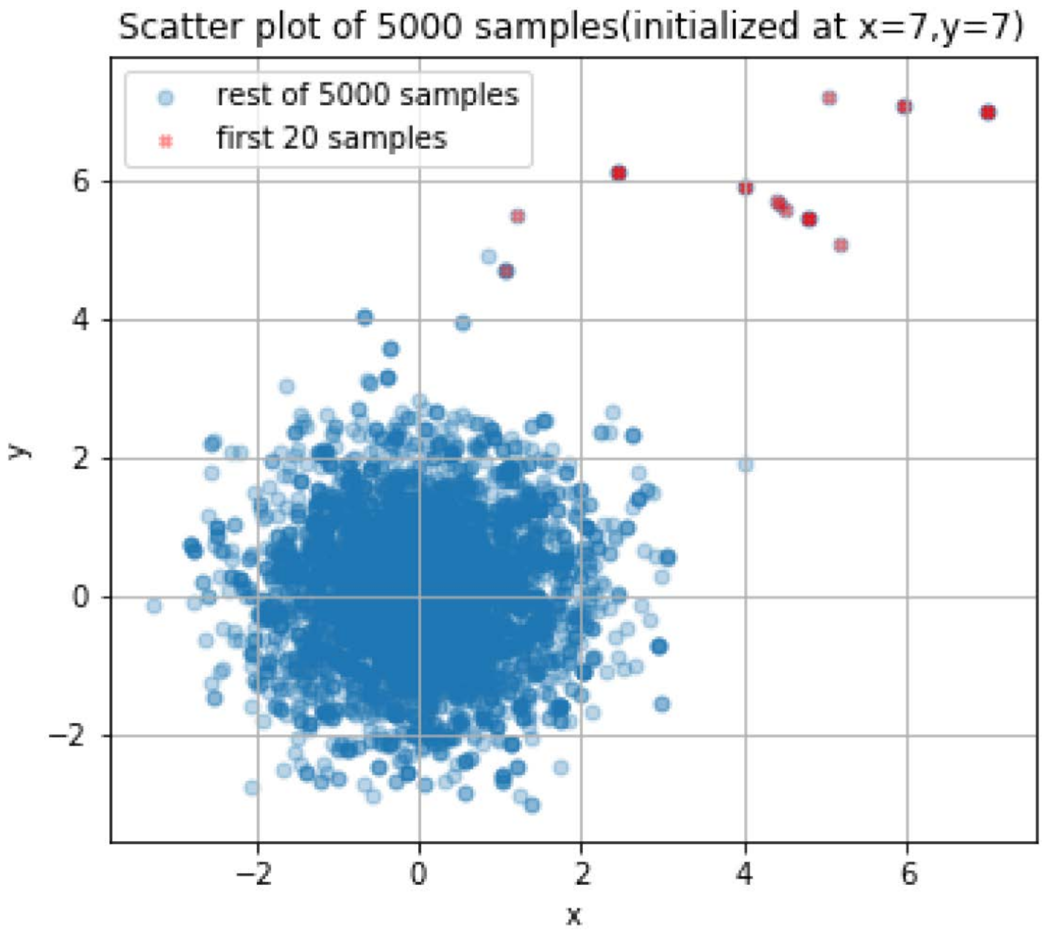
## 2. Monte-Carlo Estimate: MCMC

Example:

$$p(x, y) = \mathcal{N}(x; 0, 1)\mathcal{N}(y; 0, 1)$$

Initialize at $(x = 7, y = 7)$

Need a warm-up (burn-in) period



# Application

## Recall: Bayesian GAN

Normal GAN:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

Bayesian GAN:

$$\boxed{p(\theta_d|\mathbf{z},\mathbf{X},\theta_g)} \propto \prod_{i=1}^{n_d} D(\mathbf{x}^{(i)};\theta_d) \times \prod_{i=1}^{n_g}(1 - D(G(\mathbf{z}^{(i)};\theta_g);\theta_d)) \times \boxed{p(\theta_d|\alpha_d)}$$

posterior

prior

$$\boxed{p(\theta_g|\mathbf{z},\theta_d)} \propto \left(\prod_{i=1}^{n_g} D(G(\mathbf{z}^{(i)};\theta_g);\theta_d)\right) \boxed{p(\theta_g|\alpha_g)}$$

# Summary

Bayesian is a theory and a framework

Bayesian does not fit, but reports uncertainty

Incorporate with deep neural nets