Gaussian Processes

Ali Haghiaghtgoo

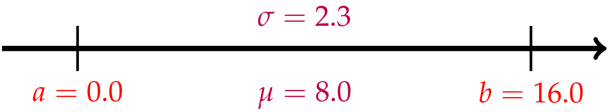
Shahid Beheshti University

May 23, 2023

Contents

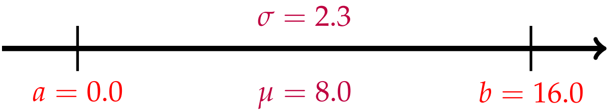
Conents

Why Gaussian is important?



- Exponential
- Gaussian
- Uniform

Why Gaussian is important?



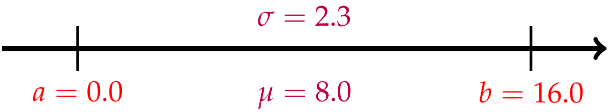
- Exponential
- Gaussian
- Uniform

Max Entropy Priciple

The maximum entropy principle is a statistical inference technique that selects the most unbiased probability distribution that satisfies a set of constraints.

$$S = -\int dx\, p(x) \log p(x)$$

Why Gaussian is important?



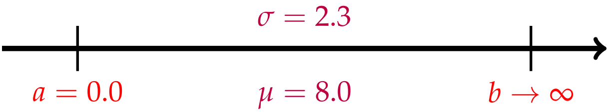
- Exponential
- Gaussian
- Uniform

Max Entropy Priciple

The maximum entropy principle is a statistical inference technique that selects the most unbiased probability distribution that satisfies a set of constraints.

$$S = -\int dx\, p(x) \log p(x)$$

Why Gaussain in important?



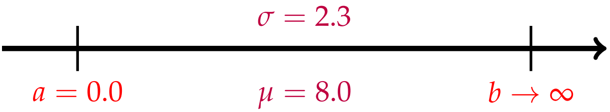
- Exponential
- Gaussian
- Uniform

Max Entropy Priciple

The maximum entropy principle is a statistical inference technique that selects the most unbiased probability distribution that satisfies a set of constraints.

$$S = -\int dx\, p(x) \log p(x)$$

Why Gaussain in important?



- Exponential
- Gaussian
- Uniform

Max Entropy Priciple

The maximum entropy principle is a statistical inference technique that selects the most unbiased probability distribution that satisfies a set of constraints.

$$S = -\int dx\, p(x) \log p(x)$$

Why Gaussain in important?

$\sigma = 2.3$

$\mu = 8.0$

- Exponential
- Gaussian
- Uniform

Max Entropy Priciple

The maximum entropy principle is a statistical inference technique that selects the most unbiased probability distribution that satisfies a set of constraints.

$$S = -\int dx\, p(x) \log p(x)$$

Why Gaussain in important?

$$\sigma = 2.3$$

$$\mu = 8.0$$

- Exponential
- Gaussian
- Uniform

Max Entropy Priciple

The maximum entropy principle is a statistical inference technique that selects the most unbiased probability distribution that satisfies a set of constraints.

$$S = -\int dx\, p(x) \log p(x)$$

Why Gaussian is important?

- Central Limit Theorem
- A good criteria to check dependency

$$\mathcal{A}, \mathcal{B} \sim \mathcal{N}(\mu, \Sigma)$$

- Least Square Loss

$$\sum (y_i - f(x_i))^2$$
$$y = f(x) + \epsilon; \qquad \epsilon \sim \mathcal{N}(\mu, \Sigma)$$

Definition

- **Mean Vector** $\mu \in \mathbb{R}^d$
- **Cov Matrix** $\Sigma \in \mathbb{R}^{d \times d}$

$$p(s|\mu,\Sigma) = (2\pi)^{-\frac{d}{2}}\Sigma^{-\frac{1}{2}} exp\left(-\frac{1}{2}(s-\mu)^T\Sigma^{-1}(s-\mu)\right) \qquad (1)$$

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_d \end{pmatrix} \qquad \mathcal{C} = \begin{pmatrix} 1 & \mathcal{C}(s_1,s_2) & \cdots & \mathcal{C}(s_1,s_d) \\ \mathcal{C}(s_2,s_1) & 1 & & \vdots \\ \vdots & & \ddots & \\ \mathcal{C}(s_d,s_1) & \cdots & & \sigma_d^2 \end{pmatrix}$$

Definition

- **Mean Vector** $\mu \in \mathbb{R}^d$
- **Cov Matrix** $\Sigma \in \mathbb{R}^{d \times d}$

$$p(s|\mu,\Sigma) = (2\pi)^{-\frac{d}{2}}\Sigma^{-\frac{1}{2}} exp\left(-\frac{1}{2}(s-\mu)^T\Sigma^{-1}(s-\mu)\right) \quad (1)$$

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_d \end{pmatrix} \quad \Sigma = \begin{pmatrix} \sigma_1^2 & \mathcal{C}(s_1,s_2)\sigma_1\sigma_2 & \cdots & \mathcal{C}(s_1,s_d)\sigma_1\sigma_d \\ \mathcal{C}(s_2,s_1)\sigma_2\sigma_1 & \sigma_2^2 & & \vdots \\ \vdots & & \ddots & \\ \mathcal{C}(s_d,s_1)\sigma_d\sigma_1 & \cdots & & \sigma_d^2 \end{pmatrix}$$

Properties

- modeling random noise (CLT)
- Convenient for Analytical Manipulation

$$x = \begin{pmatrix} x_A \\ x_B \end{pmatrix} \qquad \mu = \begin{pmatrix} \mu_A \\ \mu_B \end{pmatrix} \qquad \Sigma = \begin{pmatrix} \Sigma_{AA} & \Sigma_{AB} \\ \Sigma_{BA} & \Sigma_{BB} \end{pmatrix}$$

- **Marginalization**

$$p(x_A) = \int_{x_B} p(x_A, x_B; \mu, \Sigma) dx_B \qquad x_A \sim \mathcal{N}(\mu_A, \Sigma_{AA})$$

$$p(x_B) = \int_{x_A} p(x_A, x_B; \mu, \Sigma) dx_A \qquad x_B \sim \mathcal{N}(\mu_B, \Sigma_{BB})$$

Properties

- **Conditioning**

$$p(x_A|x_B) = \frac{p(x_A, x_B; \mu, \Sigma)}{\int_{x_A} p(x_A, x_B; \mu, \Sigma) dx_A}$$

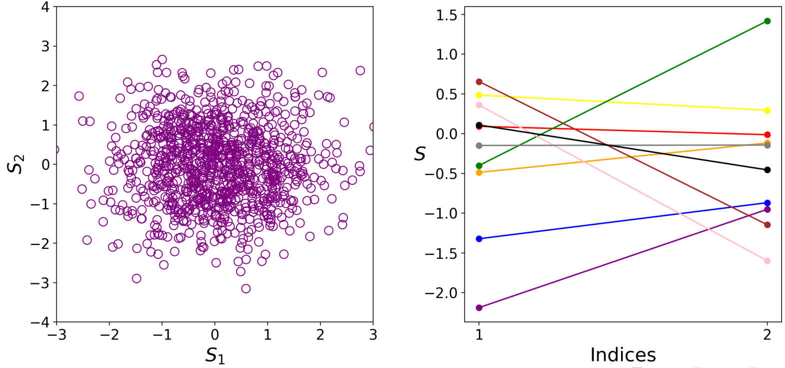
$$x_A|x_B \sim \mathcal{N}(\mu_A + \Sigma_{AB}\Sigma_{BB}^{-1}(x_B - \mu_B), \Sigma_{AA} - \Sigma_{AB}\Sigma_{BB}^{-1}\Sigma_{BA})$$

$$p(x_B|x_A) = \frac{p(x_B, x_A; \mu, \Sigma)}{\int_{x_B} p(x_B, x_A; \mu, \Sigma) dx_B}$$

$$x_B|x_A \sim \mathcal{N}(\mu_B + \Sigma_{BA}\Sigma_{AA}^{-1}(x_A - \mu_A), \Sigma_{BB} - \Sigma_{BA}\Sigma_{AA}^{-1}\Sigma_{AB})$$

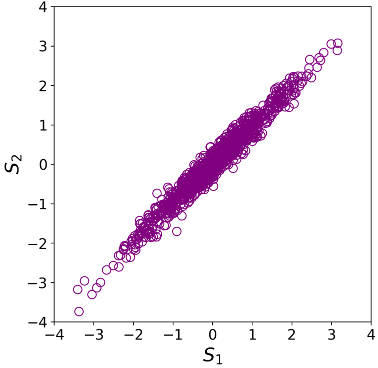
Visualization

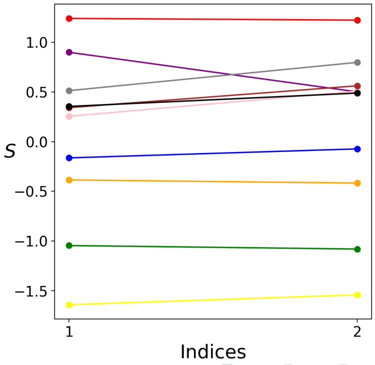
$$Cov(i,j) = \begin{cases} 1.00 & i = j \\ 0.00 & i \neq j \end{cases} \qquad (2)$$



Visualization

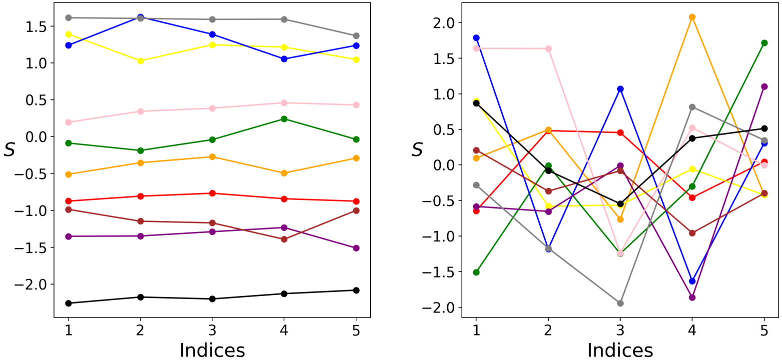
$$Cov(i,j) = \begin{cases} 1.00 & i = j \\ 0.98 & i \neq j \end{cases}$$





Visualization

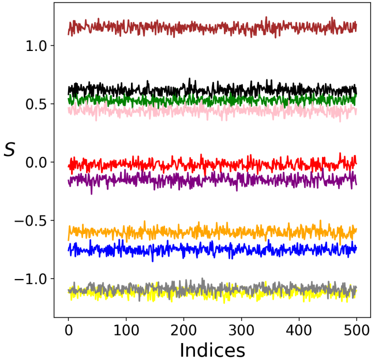
$$Cov(i,j) = \begin{cases} 1.00 & i = j \\ 0.98 & i \neq j \end{cases} \qquad Cov(i,j) = \begin{cases} 1.00 & i = j \\ 0.00 & i \neq j \end{cases}$$

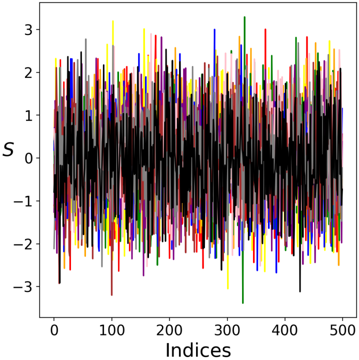


Visualization

$$Cov(i,j) = \begin{cases} 1.00 & i = j \\ 0.98 & i \neq j \end{cases}$$

$$Cov(i,j) = \begin{cases} 1.00 & i = j \\ 0.00 & i \neq j \end{cases}$$

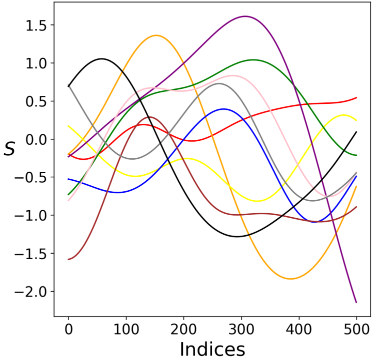


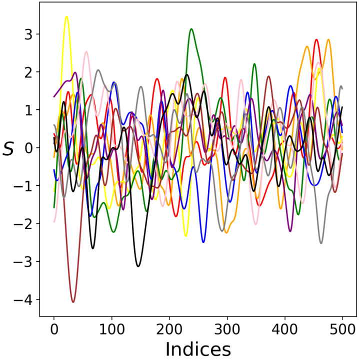


Visualization (RBF)

$$Cov(x, x') = exp\left(-\frac{1}{2}\frac{(x-x')^2}{\gamma^2}\right)$$

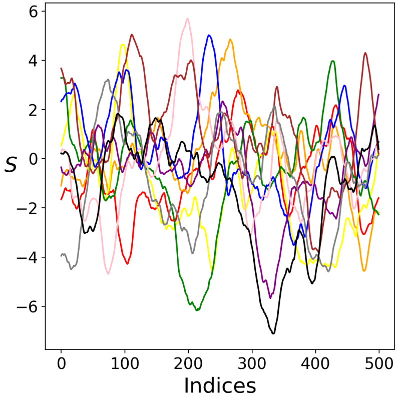
$\gamma = 100$ $\gamma = 10$





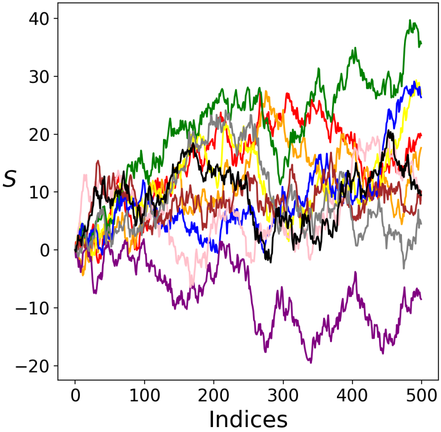
Visualization (Matern $\frac{3}{2}$)

$$Cov(x,x') = \sigma^2 \left(1 + \frac{\sqrt{3}|x-x'|}{\rho}\right) exp\left(-\frac{\sqrt{3}}{\rho}|x-x'|\right)$$



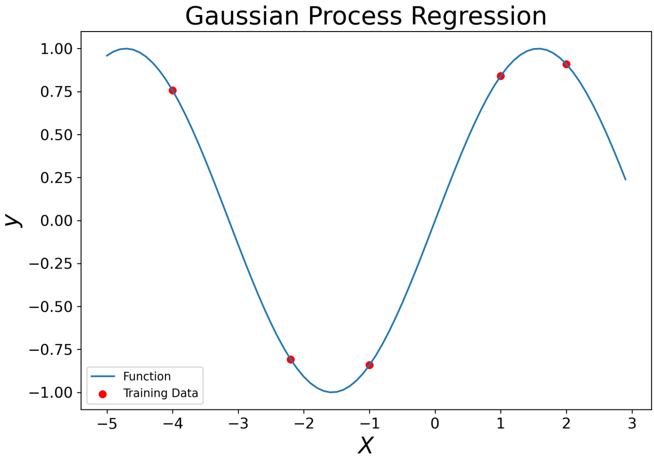
Visualization (Brownian Motion)

$$Cov(x, x') = min(x, x')$$



Conents

Gaussian Process Regression



Bayesian Linear Regression

$$y = f(x) + \epsilon; \qquad \epsilon \in \mathcal{N}(0, \sigma^2)$$
$$f(x) = W^T \phi(x)$$

Weight Space View:

- Prior

$$P(W)$$

- Posterior

$$P(W|X, y) = \frac{P(y|W, X)\ P(W)}{\int P(y, W|X)\, dW}$$

Bayesian Linear Regression

$$y = f(x) + \epsilon; \qquad \epsilon \in \mathcal{N}(0, \sigma^2)$$

$$f(x) = W^T \phi(x)$$

unknown

Weight Space View:

- Prior

Gaussian

$$P(W)$$

- Posterior

Gaussian

$$P(W|X,y) = \frac{P(y|W,X)\ P(W)}{\int P(y,W|X)\,dW}$$

Bayesian Linear Regression

$$y = f(x) + \epsilon; \qquad \epsilon \in \mathcal{N}(0, \sigma^2)$$

$$f(x) = W^T \phi(x)$$

Weight Space View:

- Prior

$$P(W)$$

- Posterior

$$P(W|X,y) = \frac{P(y|W,X)\ P(W)}{\int P(y,W|X)\, dW}$$

Bayesian Linear Regression

$$y = f(x) + \epsilon; \qquad \epsilon \in \mathcal{N}(0, \sigma^2)$$
$$f(x) = W^T \phi(x)$$

Function Space View:

- Prior

$$P(f(x^*)) = \int_W P(f|W, x^*)P(W)dW$$

- Posterior

$$P(f(x^*)|X, y) = \int_W P(f|W, x^*)P(W|X, y)dW$$

Bayesian Linear Regression

$$y = f(x) + \epsilon; \qquad \epsilon \in \mathcal{N}(0, \sigma^2)$$

$$f(x) = W^T \phi(x)$$

unknown

Function Space View:

- Prior

Deterministic

$$P(f(x^*)) = \int_W P(f|W, x^*) P(W) dW$$

Gaussian   Gaussian

- Posterior

$$P(f(x^*)|X, y) = \int_W P(f|W, x^*) P(W|X, y) dW$$

Conents

GP

- Based on Function View

$$\forall x^*, \exists\ N(\mu, \Sigma) \text{ at } f(x^*) \text{ with correlation through } W$$

- Distribution Over All $f(x^*)$ $(P(f(x^*)))$

GP

- Based on Function View

$$\forall x^*, \exists\ N(\mu, \Sigma) \text{ at } f(x^*) \text{ with correlation through } W$$

- Distribution Over All $f(x^*)$ $(P(f(x^*)))$

Gaussian Process

$$f(x) \sim GP(\mu(x), \mathcal{K}(x, x'))$$

$$\mu(x) = \langle f(x) \rangle_{samples}$$
$$\mathcal{K}(x, x') = \Big\langle (f(x) - \mu(x))\ (f(x') - \mu(x')) \Big\rangle_{samples}$$

Mean Function μ

- Setting

$$f(x) = W^T\phi(x) \qquad W \sim \mathcal{N}(0, \tau^{-1}\mathbb{1})$$

$$\begin{aligned} \mu(x) &= \langle f(x) \rangle \\ &= \langle W^T\phi(x) \rangle \\ &= 0 \end{aligned}$$

Kernel Covariance Function $\mathcal{K}$

- Setting

$$f(x) = W^T \phi(x) \qquad W \sim \mathcal{N}(0, \tau^{-1}\mathbb{1})$$

Kernel Covariance Function $\mathcal{K}$

- Setting

$$f(x) = W^T \phi(x) \qquad W \sim \mathcal{N}(0, \tau^{-1}\mathbb{1})$$

$$\begin{aligned}\mathcal{K}(x, x') &= \langle f(x) f(x') \rangle \\ &= \phi(x)^T \langle W W^T \rangle \phi(x') \\ &= \frac{\phi(x)^T \phi(x')}{\tau}\end{aligned}$$

GPR Vs BLR

GPR $\cong$ Kernelized Bayesian Regression

Bayesian Linear Regression

- Weight Space View
- Goal: $P(W|X, y)$
- Complexity: Cubic in # of basis functions

GPR

- Function Space View
- Goal: $P(f|X, y)$
- Complexity: Cubic in # of training points

|CS480/680 Spring 2019 Pascal Poupart/University of Waterloo

Recap: Bayesian Linear Regression

- Prior

$$P(W) = \mathcal{N}(0, \Sigma)$$

- Likelihood

$$P(y|X, W) = \mathcal{N}(W^T\Phi, \sigma^2\mathbb{1})$$

- Posterior

$$P(W|X, y) = \mathcal{N}(\sigma^{-2}(\sigma^{-2}\Phi\Phi^T + \Sigma^{-1})^{-1}\Phi y, (\sigma^{-2}\Phi\Phi^T + \Sigma^{-1})^{-1})$$

Prediction

$$P(y^*|x^*, X, y) = \mathcal{N}(\sigma^{-2}\phi(x^*)^T(\sigma^{-2}\Phi\Phi^T + \Sigma^{-1})^{-1}\Phi y, \\ \phi(x^*)^T(\sigma^{-2}\Phi\Phi^T + \Sigma^{-1})^{-1}\phi(x^*))$$

- Complexity: inversion of A is cubic in # of basis functions

|CS480/680 Spring 2019 Pascal Poupart/University of Waterloo

Recap: Gaussian Progress Regression

- Prior

$$P(f(.)) = \mathcal{N}(\mu(.), \mathcal{K}(.,.))$$

- Likelihood

$$P(y|X, f) = \mathcal{N}(f(.), \sigma^2 \mathbb{1})$$

- Posterior

$$P(f(.)|X, y) = \mathcal{N}(\bar{f}(.), \mathcal{K}'(.,.))$$

- Prediction

$$P(y^*|x^*, X, y) = \mathcal{N}(\bar{f}(x^*), \mathcal{K}'(x^*, x^*))$$

- Complexity: inversion of $K + \sigma^2 \mathbb{1}$ is cubic in # of basis functions

$$\bar{f}(.) = \mathcal{K}(., X)(\mathcal{K} + \sigma^2 \mathbb{1})^{-1} y$$

$$\mathcal{K}'(.,.) = \mathcal{K}(.,.) - \mathcal{K}(., X)(K + \sigma^2 \mathbb{1})^{-1} \mathcal{K}(X, .)$$

|CS480/680 Spring 2019 Pascal Poupart/University of Waterloo

Conents

Infinite Neural Networks

Universal Approximation Theorem: Neural networks with a single hidden layer (that contains sufficiently many hidden units) can approximate any function arbitrarily closely.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. Neural Networks, 2(5), 359-366.

The limit of an infinite single hidden layer neural network is a Gaussian Process.

Neal, R. M. (1994). Priors for infinite networks (tech. rep. no. crg-tr-94-1). University of Toronto, 415.

Bayesian Neural Network

- Neural Network with J hidden units

$$y_k = f(x;W) = \sum_{j=1}^{J} W_{kj}\mathcal{T}(\sum_i W_{ji}x_i + W_{j0}) + W_{k0}$$

- Bayesian Learning
  - Weight Space View

$$\langle w_{kj}\rangle = 0 \qquad var(w_{kj}) = \frac{\tau}{J}\ \forall j,$$
$$\langle w_{k0}\rangle = 0 \qquad var(w_{k0}) = \sigma^2\ \forall ji$$

  - Function Space View

$$J \to \infty$$
$$P(f(x)) = \mathcal{N}(f(x)|0, \tau\langle\mathcal{T}(x)\mathcal{T}(x')\rangle + \sigma^2)$$

Mean Derivation

$$
\begin{aligned}
\langle f(x) \rangle &= \sum_{j=1}^{J} \langle W_{kj} \mathcal{T}(x) \rangle + \langle W_{k0} \rangle \\
&= \sum_{j=1}^{J} \langle W_{kj} \rangle \, \langle \mathcal{T}(x) \rangle + \langle W_{k0} \rangle \\
&= \sum_{j=1}^{J} 0 \langle \mathcal{T}(x) \rangle + 0 \\
&= 0
\end{aligned}
$$

Covariance Derivation

$$
\begin{aligned}
cov(f(x), f(x')) &= \langle f(x) f(x') \rangle - \langle f(x) \rangle \langle f(x') \rangle \\
&= \langle f(x) f(x') \rangle \\
&= \langle (\sum_j W_{kj} \mathcal{T}_j(x) + W_{k0})(\sum_j W_{kj} \mathcal{T}_j(x') + W_{k0}) \rangle \\
&= \sum_{j=1}^{J} \langle W_{kj} \mathcal{T}_j(x) W_{kj} \mathcal{T}_j(x') \rangle + \langle W_{k0} W_{k0} \rangle \\
&= \sum_{j=1}^{J} \langle W_{kj}^2 \rangle \langle \mathcal{T}_j(x) \mathcal{T}_j(x') \rangle + Var(W_{k0}) \\
&= \sum_{j=1}^{J} Var(W_{kj}) \langle \mathcal{T}(x) \mathcal{T}(x') \rangle + Var(W_{k0}) \\
&= \sum_{j=1}^{J} \frac{\alpha}{J} \langle \mathcal{T}(x) \mathcal{T}(x') \rangle + \sigma^2 \\
&= \alpha \langle \mathcal{T}(x) \mathcal{T}(x') \rangle + \sigma^2
\end{aligned}
$$

- When # of hidden units $J \to \infty$, Neural net is equivalent to a GP.
- This works for:
  - Any activation function.
  - Any i.i.d prior over the weights with $mean = 0$

Credit

CS480/680 Spring 2019 Pascal Poupart/University of Waterloo

Thanks