University of BRISTOL

Machine Learning

Dual Linear Regression

Carl Henrik Ek - carlhenrik.ek@bristol.ac.uk

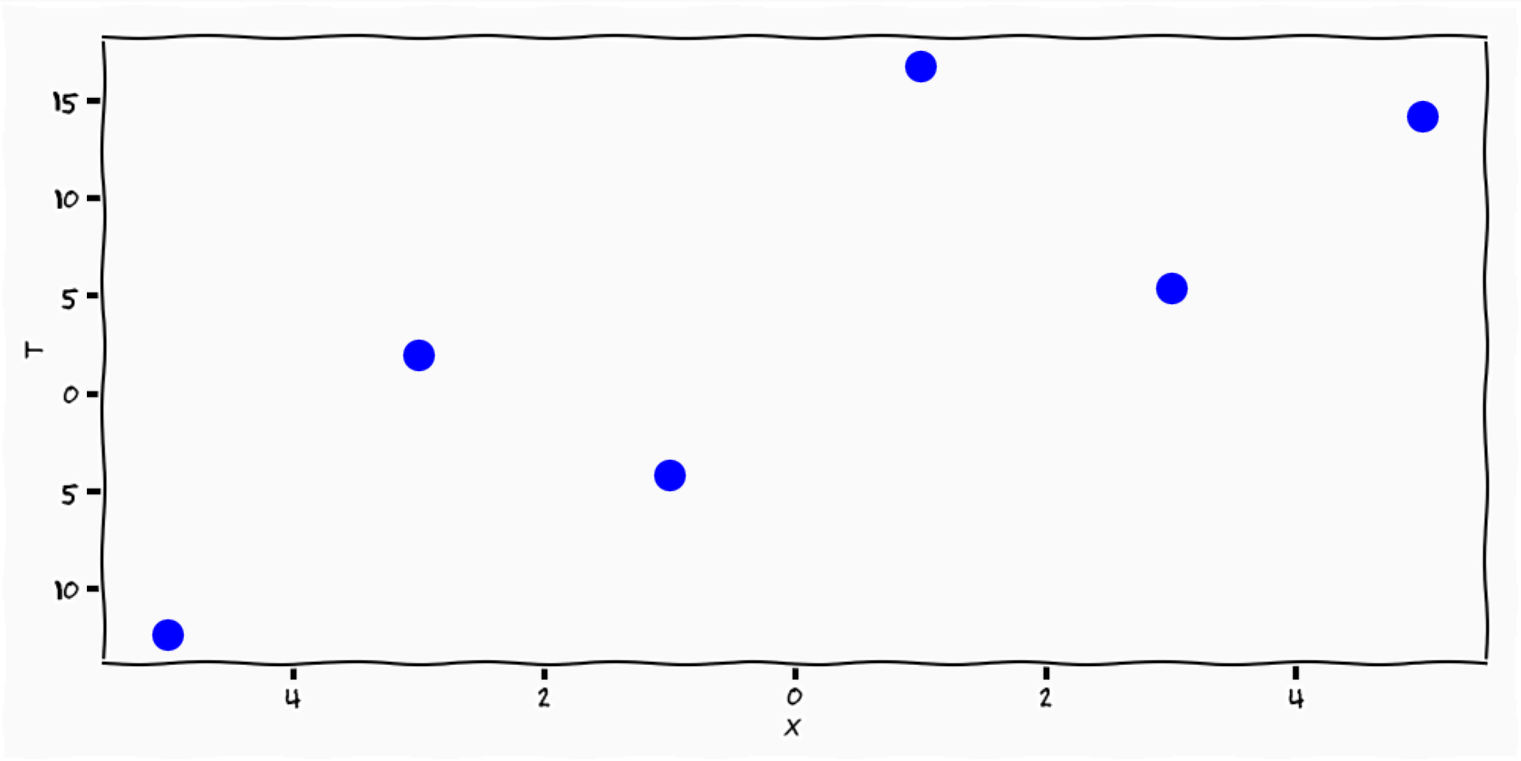
October 15, 2019

http://carlhenrik.com

Reddit

Introduction

Linear Regression [1] Ch 3.1



- Linear function in both parameters and data

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + \ldots w_D x_D = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0 = \{D = 1\} = \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x})$$

Linear Regression

One point

$$t_1 = \mathbf{w}^{\mathrm{T}}\mathbf{x} = [w_0, w_1] \cdot \begin{bmatrix} 1 \\ x_1 \end{bmatrix} = w_0 + w_1 \cdot x_1$$

Multiple points

$$\begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_N \end{bmatrix} \cdot \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

Likelihood

$$t = f(\mathbf{x}) + \epsilon$$

Likelihood

$$t = f(\mathbf{x}) + \epsilon$$

$$t - f(\mathbf{x}) = \epsilon$$

Likelihood

$$t = f(\mathbf{x}) + \epsilon$$

$$t - f(\mathbf{x}) = \epsilon$$

$$t - f(\mathbf{x}) \sim \mathcal{N}(\epsilon|0, \beta^{-1}I) = \frac{\beta}{(2\pi)^{\frac{1}{2}}} e^{-\frac{1}{2}(\epsilon-0)\beta(\epsilon-0)}$$

Likelihood

$$t = f(\mathbf{x}) + \epsilon$$

$$t - f(\mathbf{x}) = \epsilon$$

$$t - f(\mathbf{x}) \sim \mathcal{N}(\epsilon|0, \beta^{-1}I) = \frac{\beta}{(2\pi)^{\frac{1}{2}}} e^{-\frac{1}{2}(\epsilon-0)\beta(\epsilon-0)}$$

$$\Rightarrow \mathcal{N}(t - f(\mathbf{x})|0, \beta^{-1}I) \frac{\beta}{(2\pi)}^{\frac{1}{2}} e^{-\frac{1}{2}(t-f(\mathbf{x}))\beta(t-f(\mathbf{x}))}$$

Likelihood

$$t = f(\mathbf{x}) + \epsilon$$

$$t - f(\mathbf{x}) = \epsilon$$

$$t - f(\mathbf{x}) \sim \mathcal{N}(\epsilon|0, \beta^{-1}I) = \frac{\beta}{(2\pi)^{\frac{1}{2}}} e^{-\frac{1}{2}(\epsilon-0)\beta(\epsilon-0)}$$

$$\Rightarrow \mathcal{N}(t - f(\mathbf{x})|0, \beta^{-1}I) \frac{\beta}{(2\pi)}^{\frac{1}{2}} e^{-\frac{1}{2}(t-f(\mathbf{x}))\beta(t-f(\mathbf{x}))}$$

$$\Rightarrow \mathcal{N}(t - f(\mathbf{x})|0, \beta^{-1}I) = \mathcal{N}(t|f(\mathbf{x}), \beta^{-1}I)$$

Likelihood

$$t = f(\mathbf{x}) + \epsilon$$

$$t - f(\mathbf{x}) = \epsilon$$

$$t - f(\mathbf{x}) \sim \mathcal{N}(\epsilon|0, \beta^{-1}I) = \frac{\beta}{(2\pi)^{\frac{1}{2}}} e^{-\frac{1}{2}(\epsilon-0)\beta(\epsilon-0)}$$

$$\Rightarrow \mathcal{N}(t - f(\mathbf{x})|0, \beta^{-1}I) \frac{\beta}{(2\pi)}^{\frac{1}{2}} e^{-\frac{1}{2}(t-f(\mathbf{x}))\beta(t-f(\mathbf{x}))}$$

$$\Rightarrow \mathcal{N}(t - f(\mathbf{x})|0, \beta^{-1}I) = \mathcal{N}(t|f(\mathbf{x}), \beta^{-1}I)$$

$$\Rightarrow p(t|f, \mathbf{x}) = \mathcal{N}(t|f(\mathbf{x}), \beta^{-1}I)$$

Likelihood

$$t = f(\mathbf{x}) + \epsilon$$

$$t - f(\mathbf{x}) = \epsilon$$

$$t - f(\mathbf{x}) \sim \mathcal{N}(\epsilon|0, \beta^{-1}I) = \frac{\beta}{(2\pi)^{\frac{1}{2}}} e^{-\frac{1}{2}(\epsilon-0)\beta(\epsilon-0)}$$

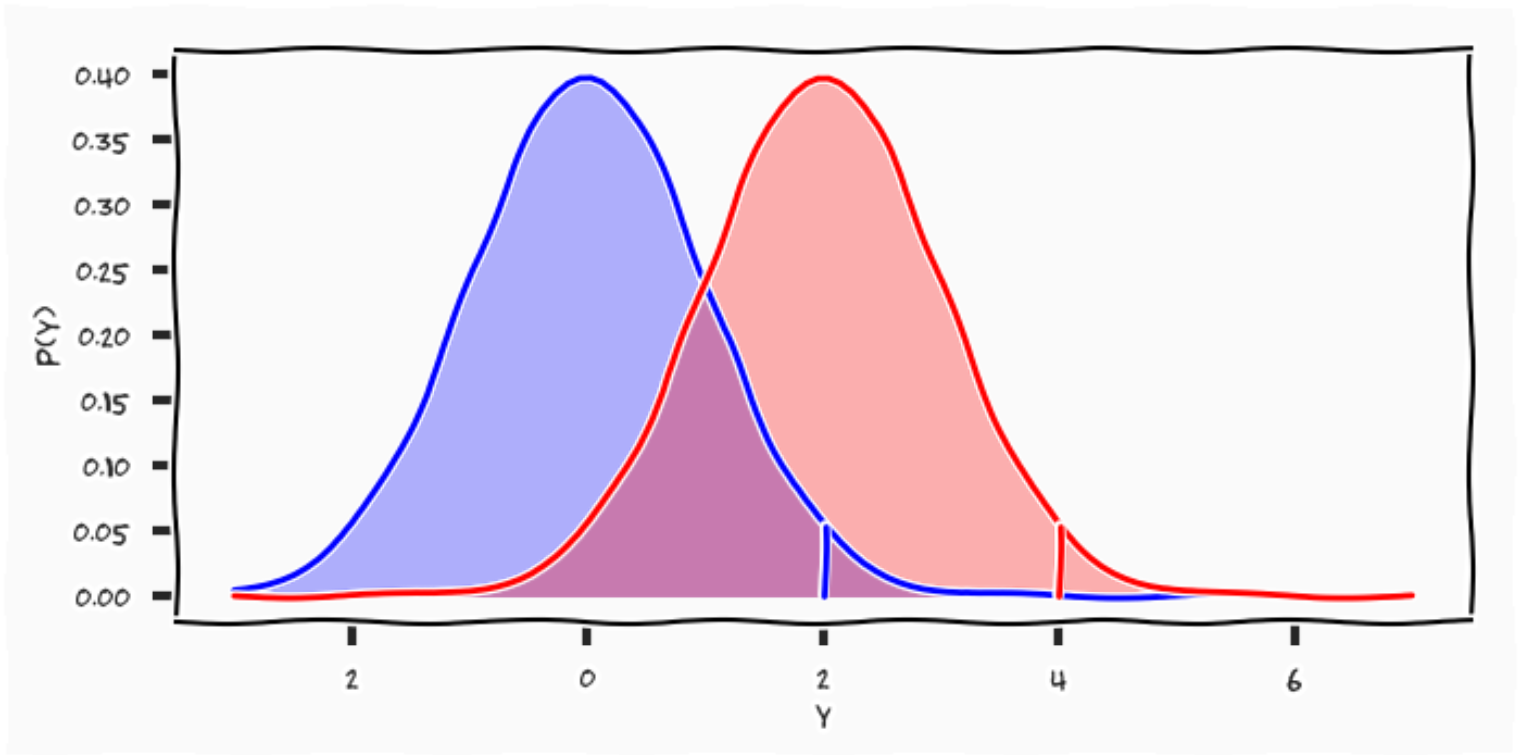
$$\Rightarrow \mathcal{N}(t - f(\mathbf{x})|0, \beta^{-1}I) \frac{\beta}{(2\pi)}^{\frac{1}{2}} e^{-\frac{1}{2}(t-f(\mathbf{x}))\beta(t-f(\mathbf{x}))}$$

$$\Rightarrow \mathcal{N}(t - f(\mathbf{x})|0, \beta^{-1}I) = \mathcal{N}(t|f(\mathbf{x}), \beta^{-1}I)$$

$$\Rightarrow p(t|f, \mathbf{x}) = \mathcal{N}(t|f(\mathbf{x}), \beta^{-1}I)$$

- By making an assumption of the noise we have reached a conditional distribution over the output given the model, i.e. the likelihood function

Likelihood



$$\mathcal{N}(y = 4 - 2|\mu = 0, 1.0) = \mathcal{N}(y = 4|\mu = 2, 1.0)$$

Linear Regression

- Model

$$t = y(\mathbf{x}, \mathbf{w}) + \epsilon = \mathbf{w}^{\mathrm{T}}\mathbf{x} + \epsilon$$
$$\epsilon \sim \mathcal{N}(0, \beta^{-1} I)$$

Linear Regression

- Model

$$t = y(\mathbf{x}, \mathbf{w}) + \epsilon = \mathbf{w}^{\mathrm{T}}\mathbf{x} + \epsilon$$
$$\epsilon \sim \mathcal{N}(0, \beta^{-1} I)$$

- Likelihood

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

Linear Regression

- Model

$$t = y(\mathbf{x}, \mathbf{w}) + \epsilon = \mathbf{w}^{\mathrm{T}}\mathbf{x} + \epsilon$$
$$\epsilon \sim \mathcal{N}(0, \beta^{-1} I)$$

- Likelihood

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

- Independence

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

Linear Regression

- Model

$$t = y(\mathbf{x}, \mathbf{w}) + \epsilon = \mathbf{w}^{\mathrm{T}}\mathbf{x} + \epsilon$$
$$\epsilon \sim \mathcal{N}(0, \beta^{-1} I)$$

- Likelihood

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

- Independence

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

- Prior (Conjugate)

$$p(\mathbf{w}|m_0, S_0) = \mathcal{N}(\mathbf{w}|m_0, S_0)$$

Posterior

- Posterior

$$p(\mathbf{w}|\mathbf{X},\mathbf{t}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N,\mathbf{S}_N)$$
$$\mathbf{m}_N = \left(\mathbf{S}_0^{-1} + \beta\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\left(S_0^{-1}\mathbf{m}_0 + \beta\mathbf{X}^{\mathrm{T}}\mathbf{t}\right)$$
$$\mathbf{S}_N = \left(\mathbf{S}_0^{-1} + \beta\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}$$

Posterior

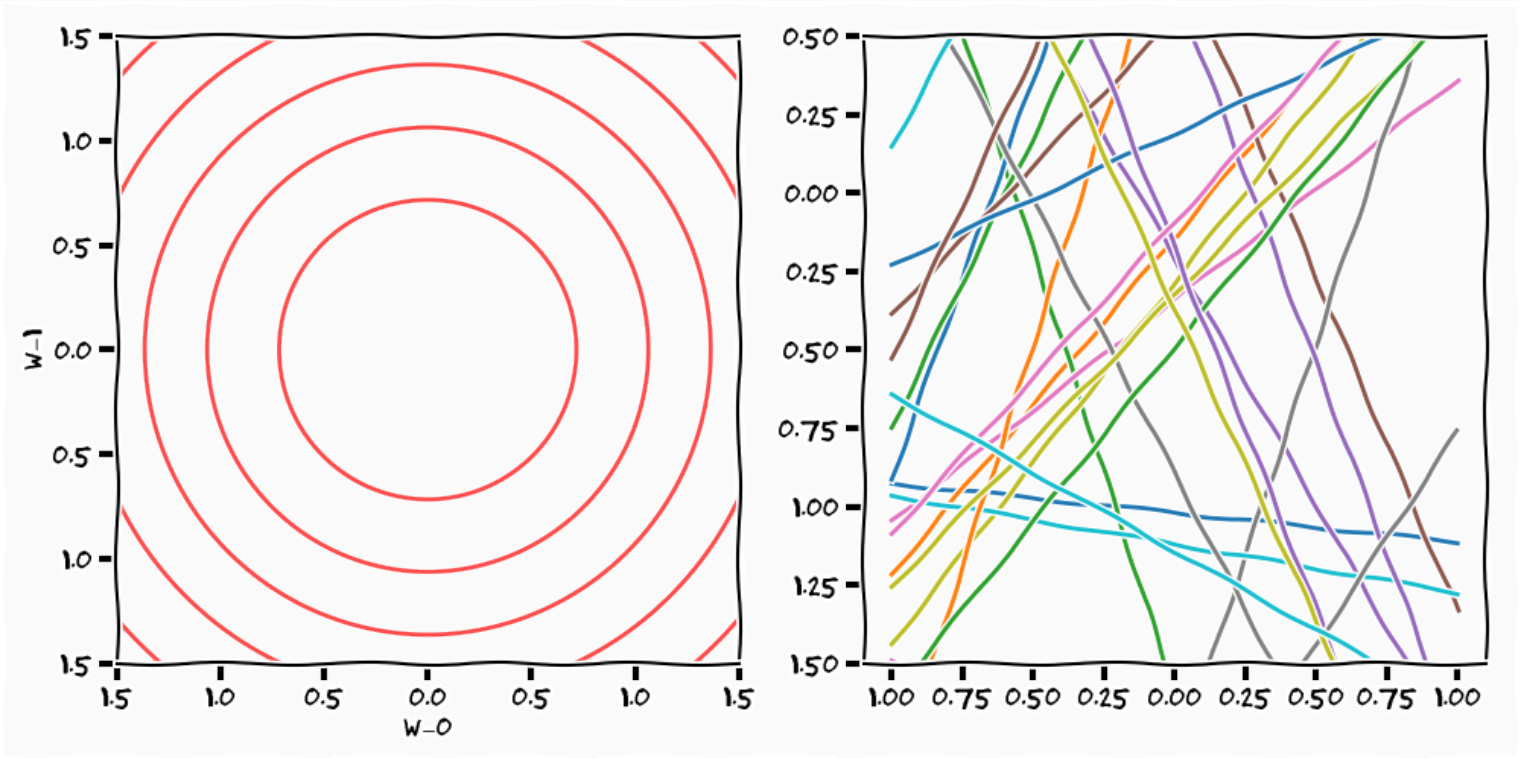
- Posterior

$$p(\mathbf{w}|\mathbf{X},\mathbf{t}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N,\mathbf{S}_N)$$
$$\mathbf{m}_N = \left(\mathbf{S}_0^{-1} + \beta\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\left(S_0^{-1}\mathbf{m}_0 + \beta\mathbf{X}^{\mathrm{T}}\mathbf{t}\right)$$
$$\mathbf{S}_N = \left(\mathbf{S}_0^{-1} + \beta\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}$$

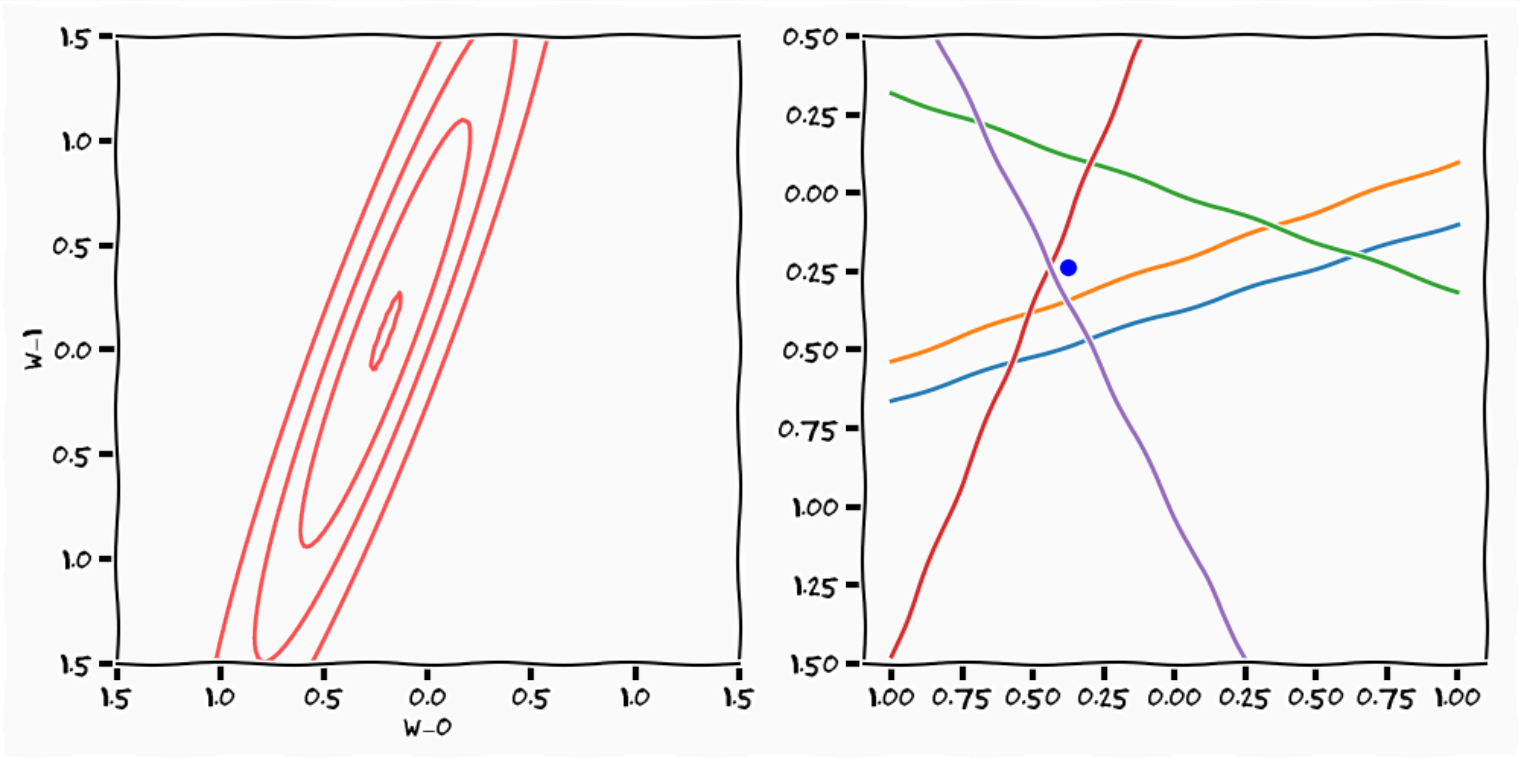
- Assumption Zero mean isotropic Gaussian

$$p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\underbrace{\mathbf{0}}_{\mathbf{m}_0},\underbrace{\alpha^{-1}\mathbf{I}}_{\mathbf{S}_0})$$

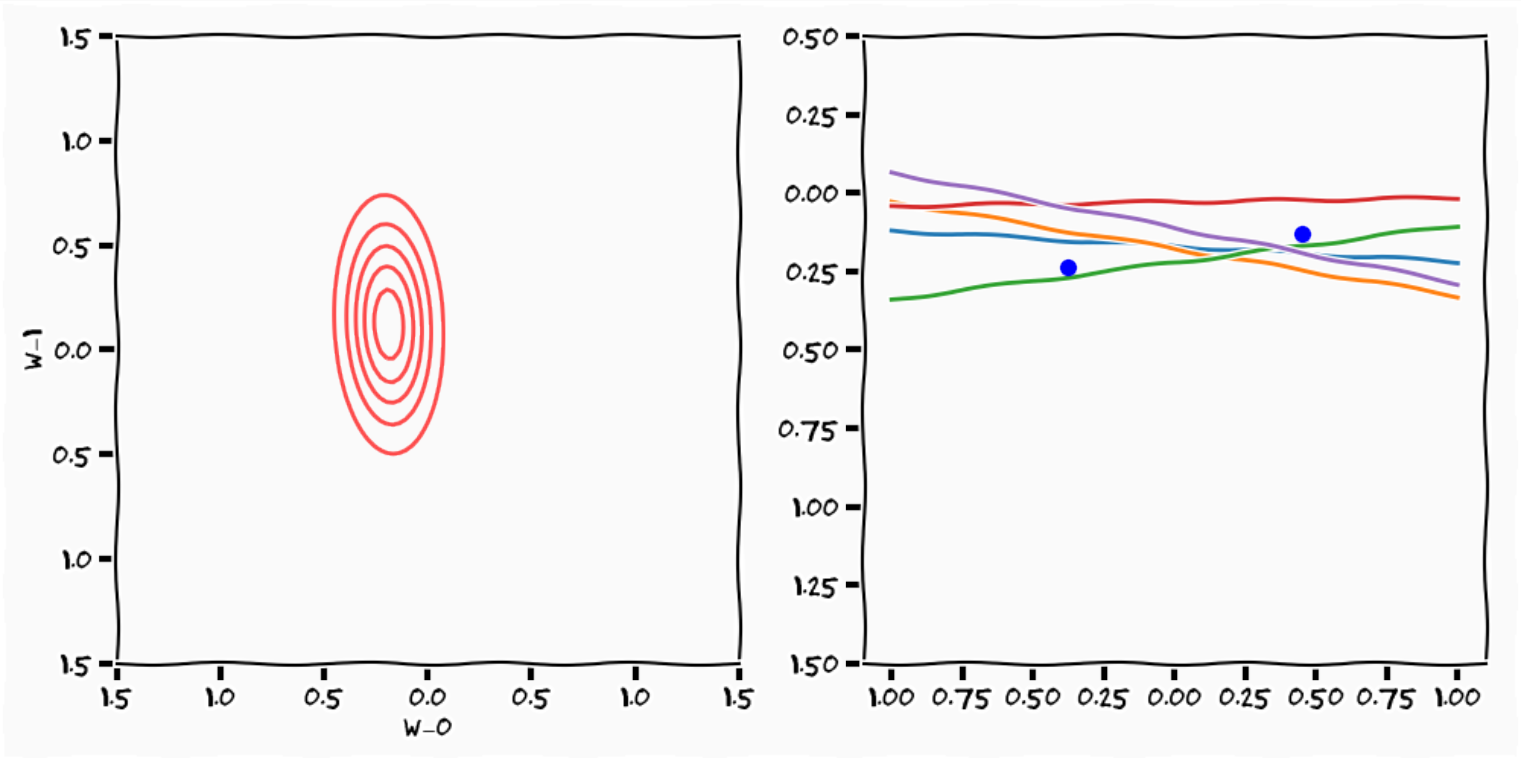
Linear Regression Example



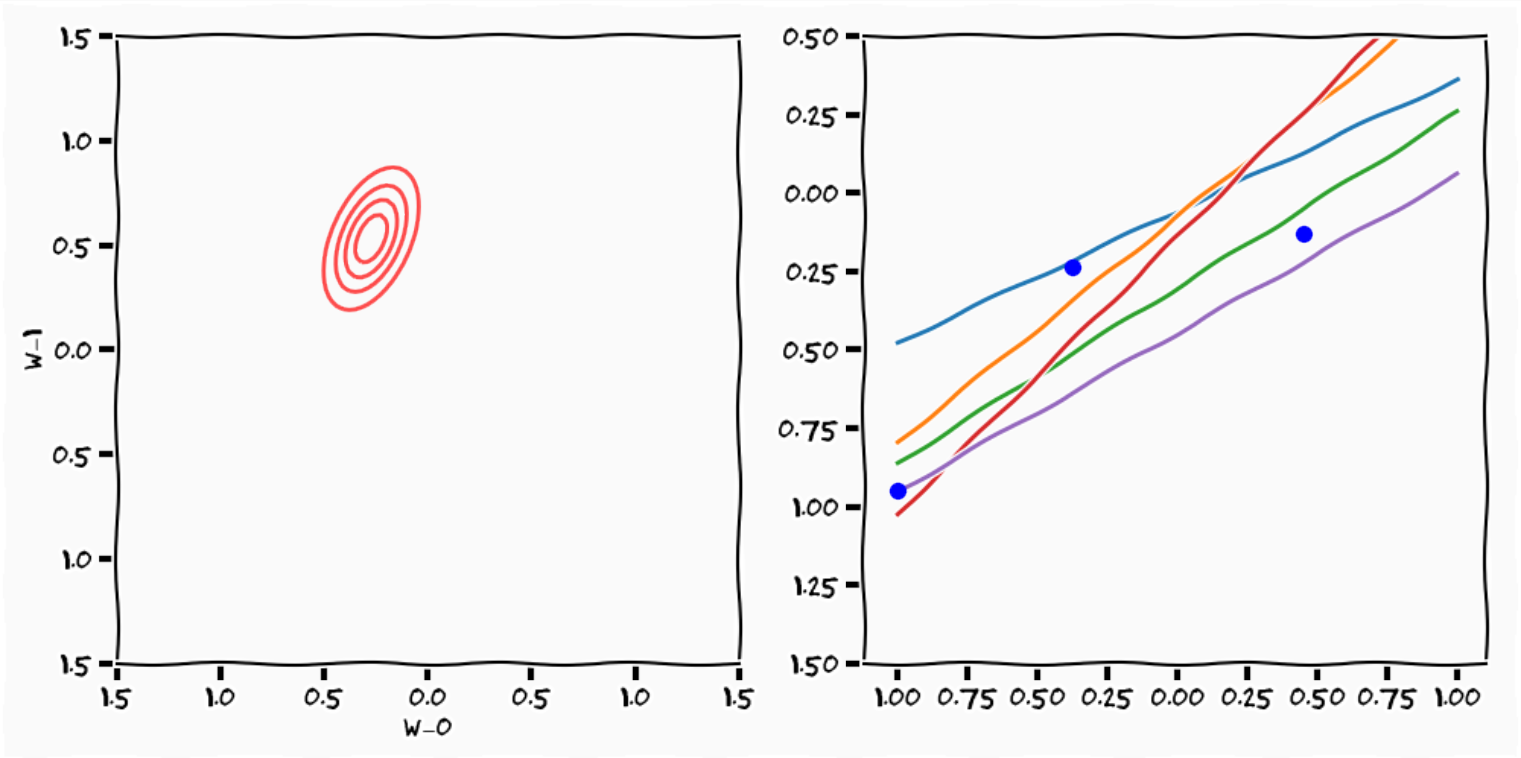
Linear Regression Example



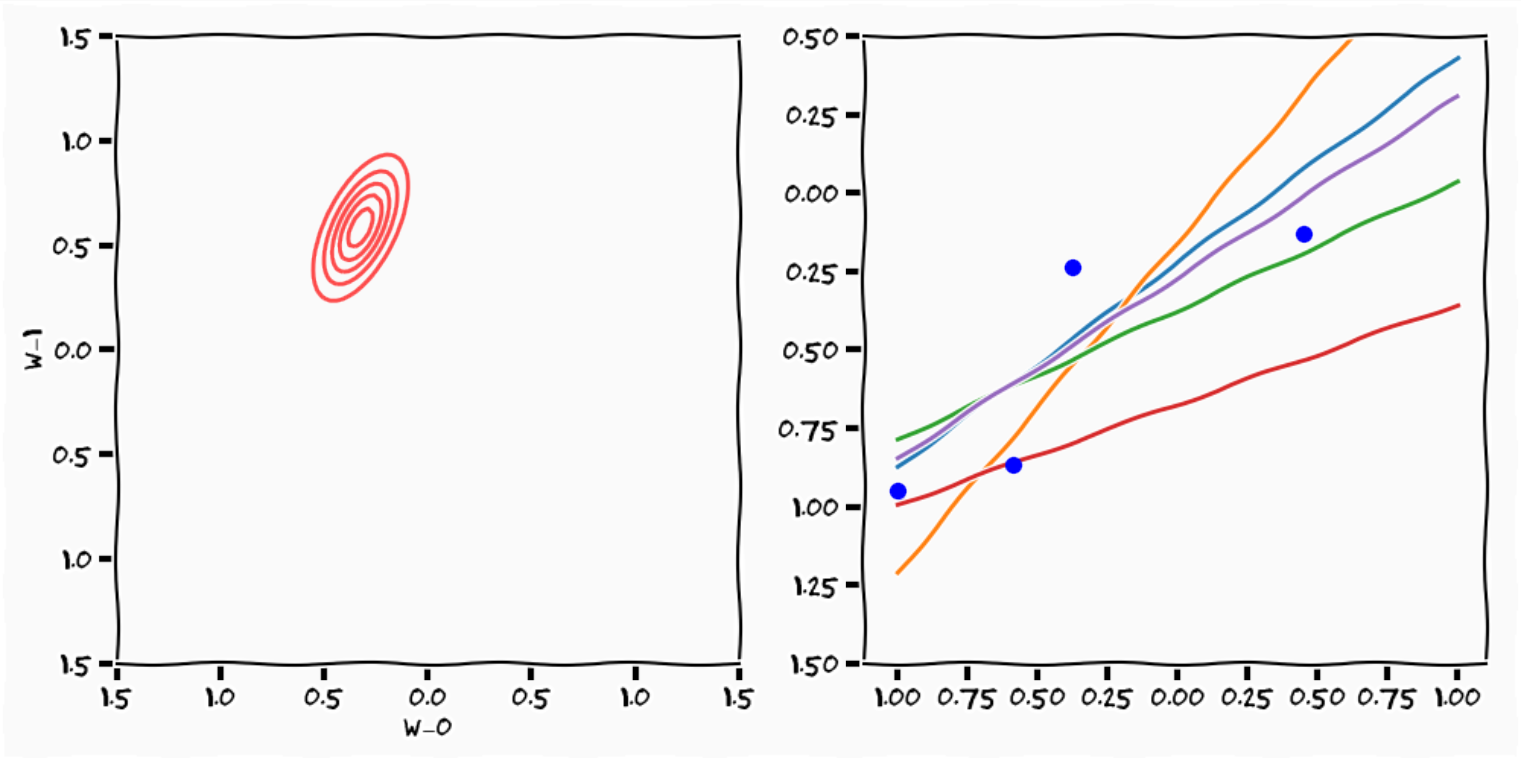
Linear Regression Example



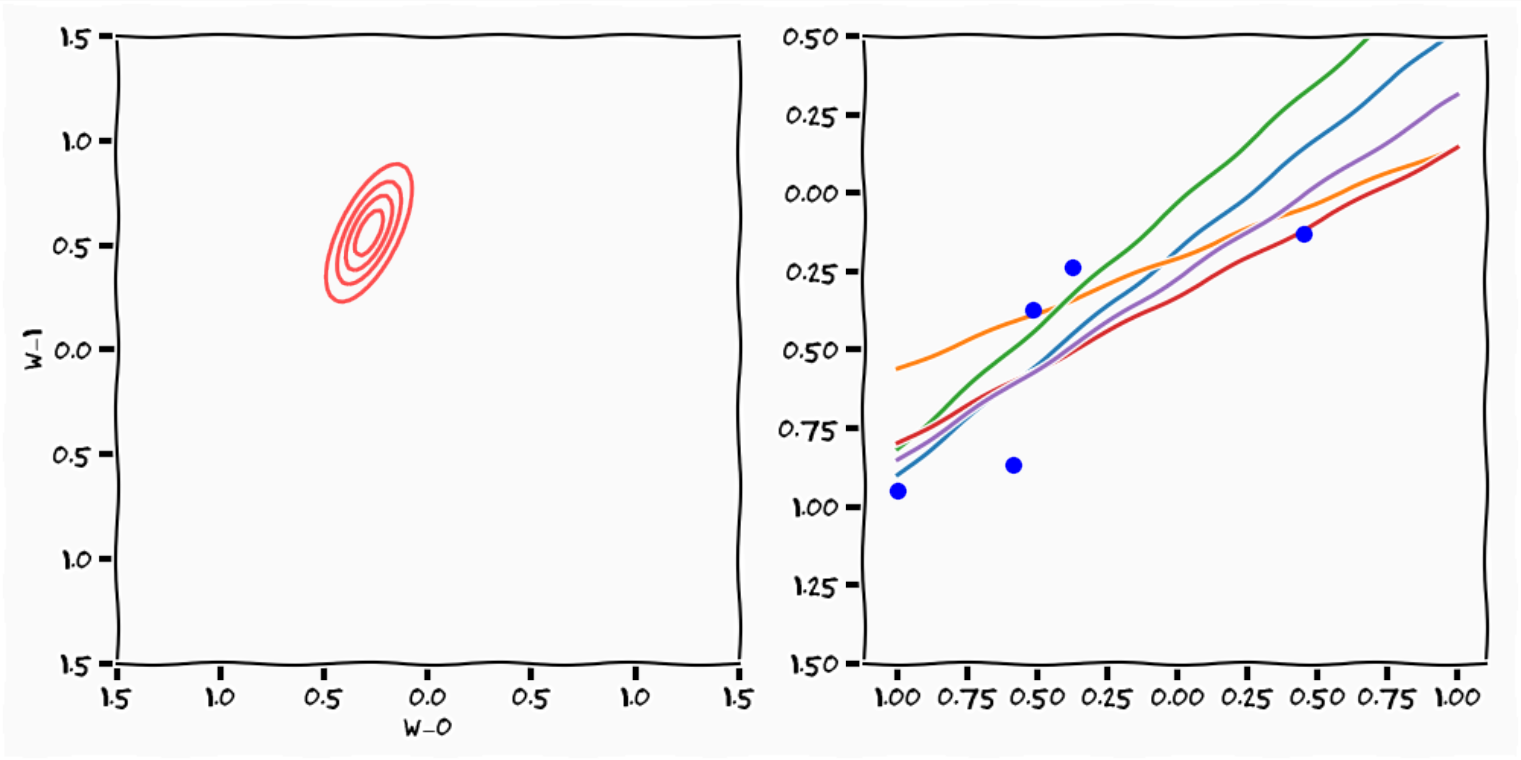
Linear Regression Example



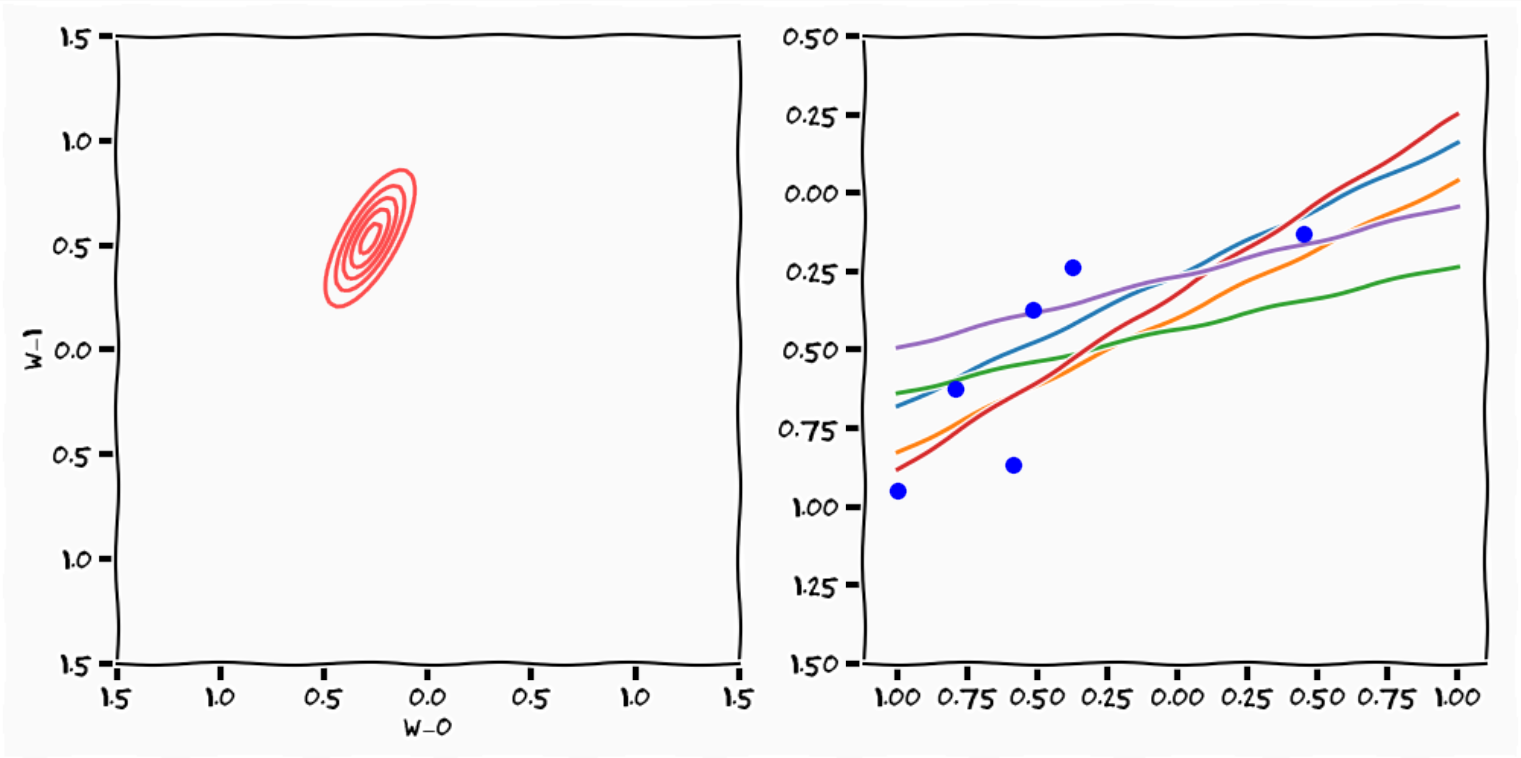
Linear Regression Example



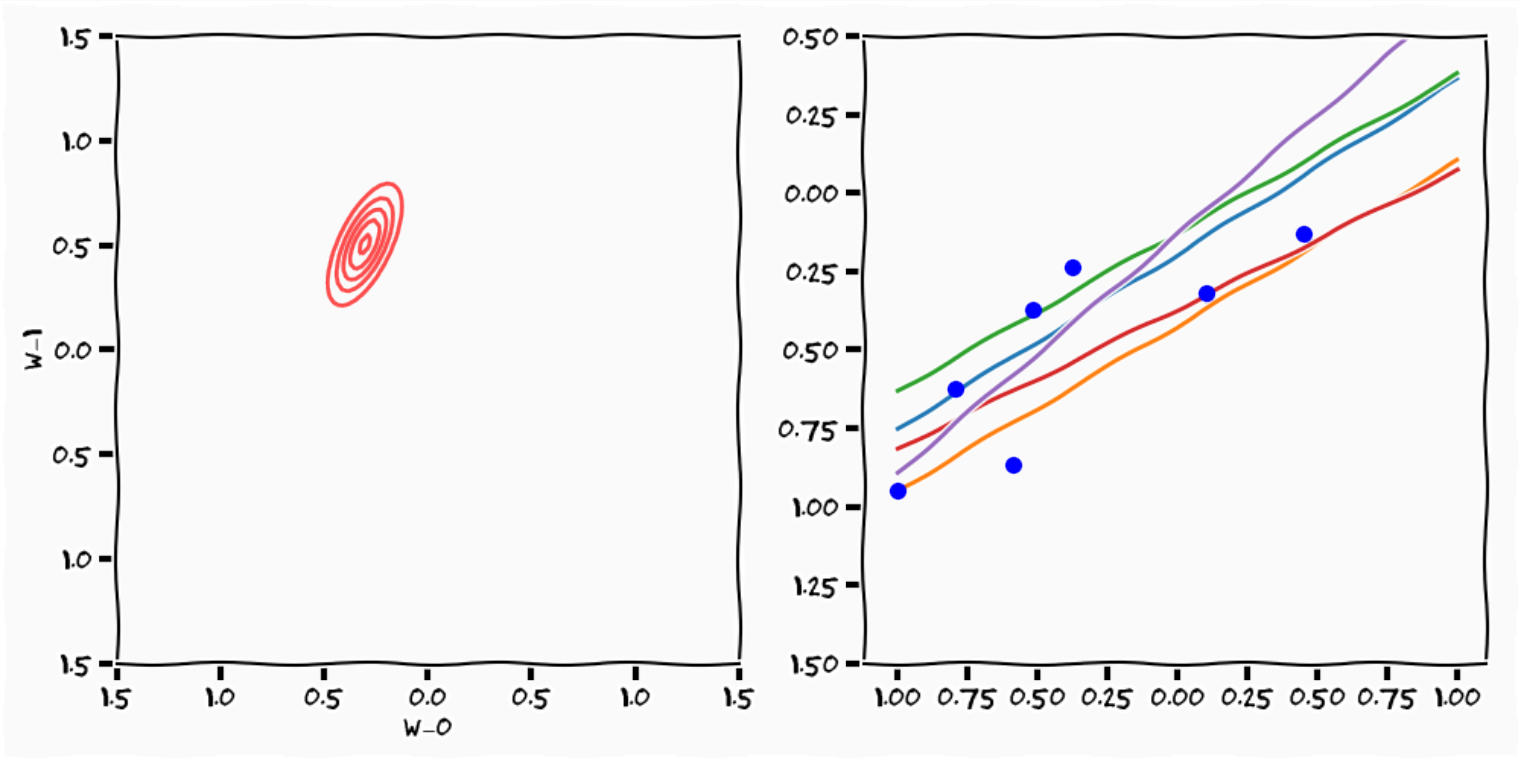
Linear Regression Example



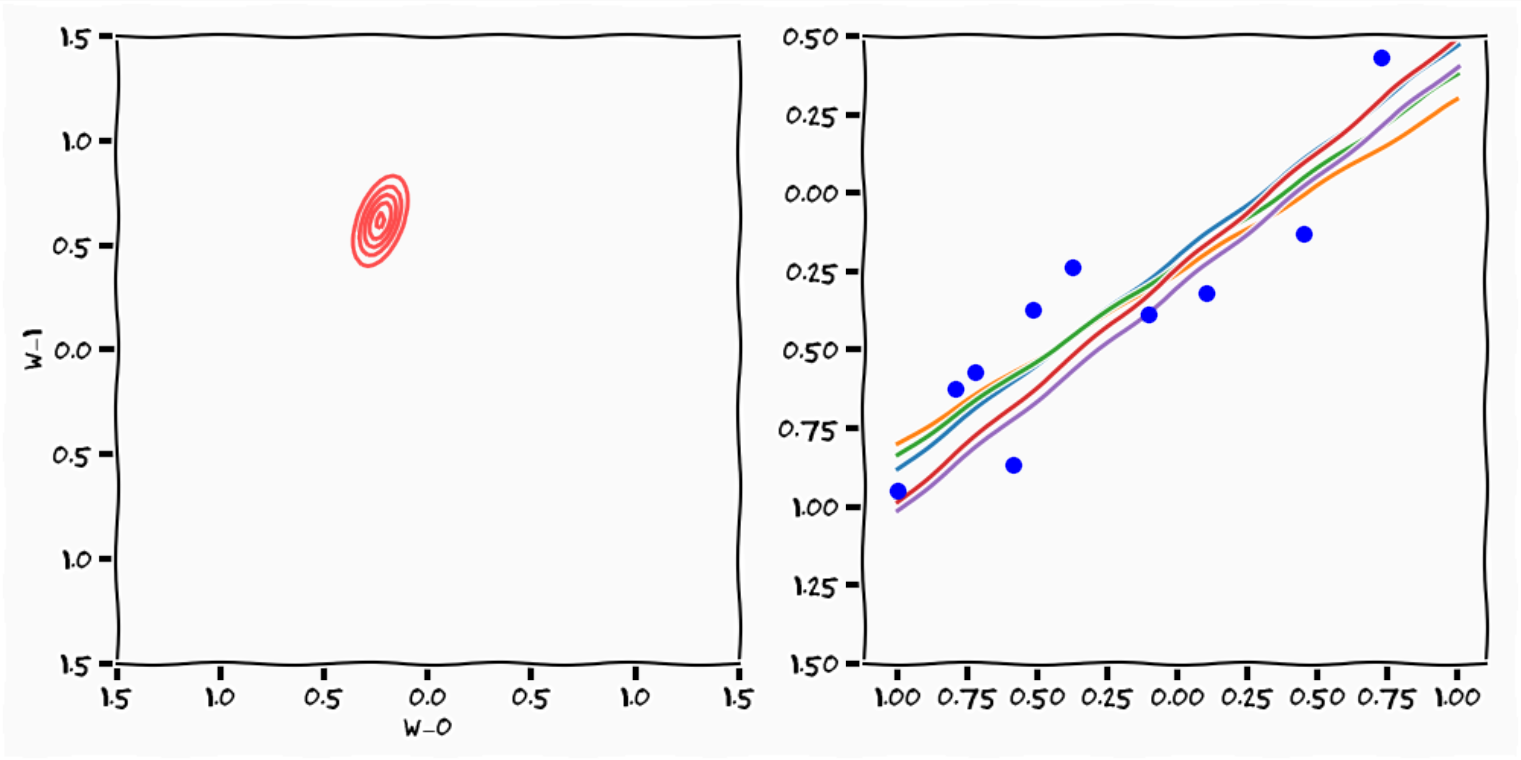
Linear Regression Example



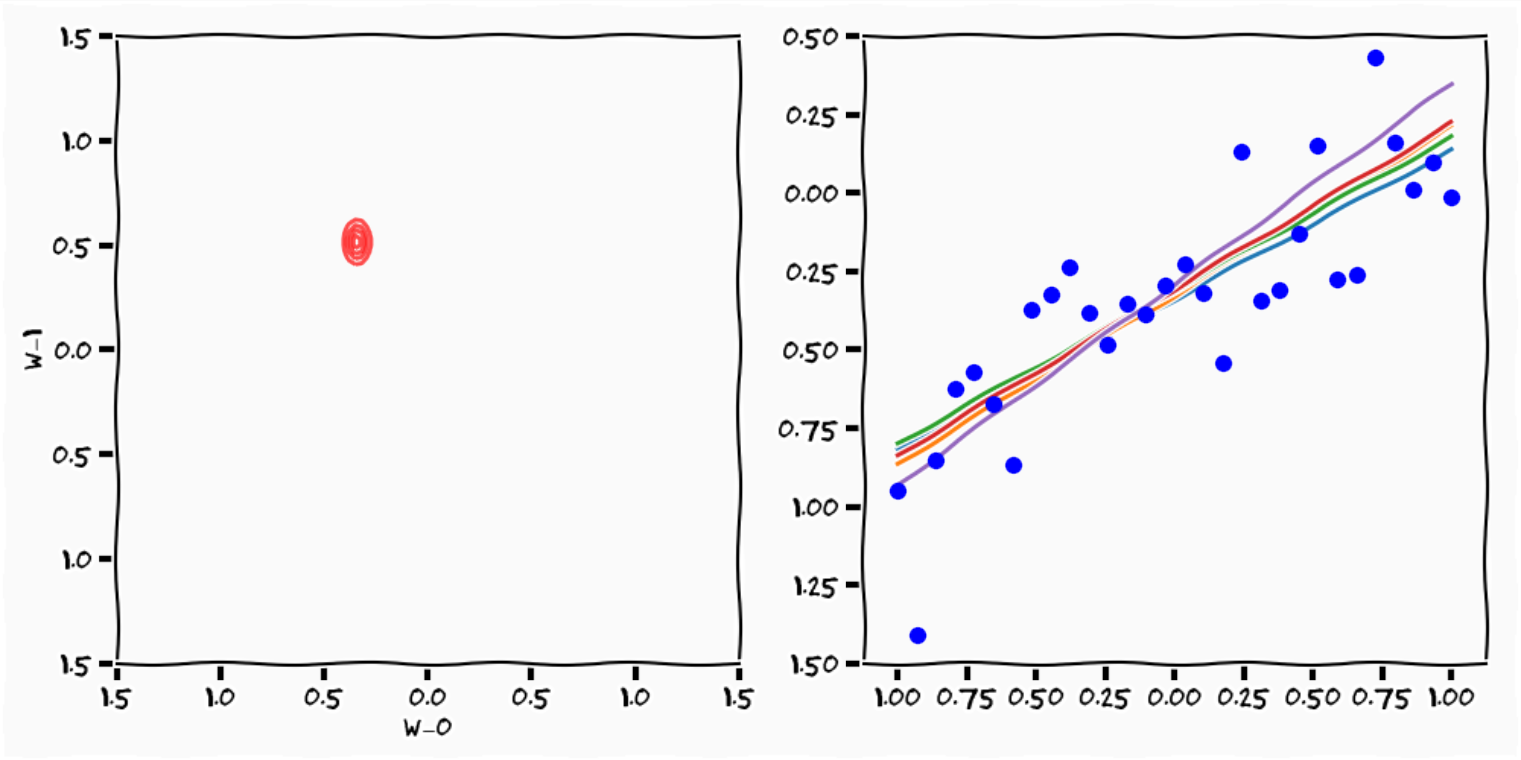
Linear Regression Example



Linear Regression Example



Linear Regression Example



Does this make sense?

Posterior Variance

$$\mathbf{S}_N = \left(\mathbf{I}\alpha + \beta\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}$$

Posterior Mean

$$\mathbf{m}_N = \left(\frac{1}{\alpha}\mathbf{I} + \beta\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\beta\mathbf{X}^{\mathrm{T}}\mathbf{t}$$

Posterior Variance

$$\mathbf{S}_N = \left(\mathbf{I}\alpha + \beta \mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}$$

$$= \left(\mathbf{I}\alpha + \beta \begin{bmatrix} \sum_i^N 1 & \sum_i x_i \\ \sum_i x_i & \sum_i x_i^2 \end{bmatrix}\right)^{-1} = \begin{bmatrix} \beta N + \alpha & \beta \sum_i x_i \\ \beta \sum_i x_i & \alpha + \beta \sum_i x_i^2 \end{bmatrix}^{-1}$$

$$= \frac{1}{(\beta N + \alpha)(\alpha + \beta \sum_i x_i^2) - (\beta \sum_i x_i)^2} \begin{bmatrix} \alpha + \beta \sum_i x_i^2 & -\beta \sum_i x_i \\ -\beta \sum_i x_i & \beta N + \alpha \end{bmatrix}$$

Posterior Variance

$$\mathbf{S}_N = \frac{1}{(\beta N + \alpha)(\alpha + \beta \sum_i x_i^2) - (\beta \sum_i x_i)^2} \begin{bmatrix} \alpha + \beta \sum_i x_i^2 & -\beta \sum_i x_i \\ -\beta \sum_i x_i & \beta N + \alpha \end{bmatrix}$$

Posterior Variance

$$\mathbf{S}_N = \frac{1}{(\beta N + \alpha)(\alpha + \beta \sum_i x_i^2) - (\beta \sum_i x_i)^2} \begin{bmatrix} \alpha + \beta \sum_i x_i^2 & -\beta \sum_i x_i \\ -\beta \sum_i x_i & \beta N + \alpha \end{bmatrix}$$

- Lets assume input is centered $\Rightarrow \sum_i x_i = 0$

$$\mathbf{S}_N = \frac{1}{(\beta N + \alpha)(\alpha + \beta \sum_i x_i^2)} \begin{bmatrix} \alpha + \beta \sum_i x_i^2 & 0 \\ 0 & \beta N + \alpha \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{\beta N + \alpha} & 0 \\ 0 & \frac{1}{\alpha + \beta \sum_i x_i^2} \end{bmatrix}$$

Posterior Mean

$$
\begin{aligned}
\mathbf{m}_N &= \left(\alpha\mathbf{I} + \beta\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\beta\mathbf{X}^{\mathrm{T}}\mathbf{t} \\
&= \beta\mathbf{S}_N \begin{bmatrix} 1 & \dots & 1 \\ x_1 & \dots & x_N \end{bmatrix} \begin{bmatrix} t_1 \\ \vdots \\ t_N \end{bmatrix} \\
&= \beta\mathbf{S}_N \begin{bmatrix} \sum_i t_i \\ \sum_i t_i x_i \end{bmatrix}
\end{aligned}
$$

Posterior Mean

$$\mathbf{m}_N = \beta \mathbf{S}_N \begin{bmatrix} \sum_i t_i \\ \sum_i t_i x_i \end{bmatrix}$$

- Lets assume input is centered $\Rightarrow \sum_i x_i = 0$

$$\mathbf{m}_N = \beta \begin{bmatrix} \frac{1}{\beta N + \alpha} & 0 \\ 0 & \frac{1}{\alpha + \beta \sum_i x_i^2} \end{bmatrix} \begin{bmatrix} \sum_i t_i \\ \sum_i t_i x_i \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\beta \sum_i t_i}{\beta N + \alpha} \\ \frac{\beta \sum_i t_i x_i}{\alpha + \beta \sum_i x_i^2} \end{bmatrix}$$

Posterior Mean Slope

$$\tilde{w}_0 = \frac{\beta \sum_i t_i}{\beta N + \alpha}$$

$$p(w_0) = \mathcal{N}(w_0|0, \frac{1}{\alpha})$$

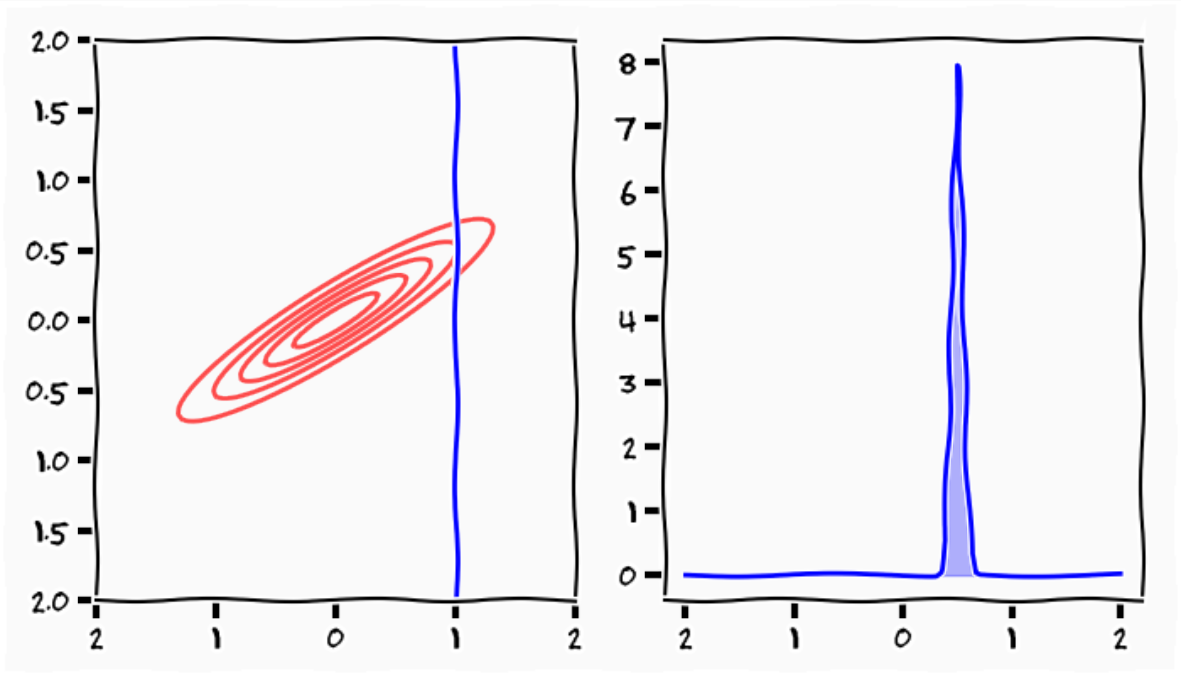
$$p(\epsilon) = \mathcal{N}(\epsilon|0, \frac{1}{\beta})$$

Prediction

$$p(t_*|\mathbf{t}, \mathbf{x}_*, \mathbf{X}, \alpha, \beta) = \int p(t_*|\mathbf{x}_*, \mathbf{w}, \beta) p(\mathbf{w}|\mathbf{t}, \mathbf{X}, \alpha, \beta) \mathrm{d}\mathbf{w}$$
$$= \mathbb{E}_{p(\mathbf{w}|\mathbf{t}, \mathbf{X}, \alpha, \beta)} \left[p(t_*|\mathbf{x}_*, \mathbf{w}, \beta) \right]$$

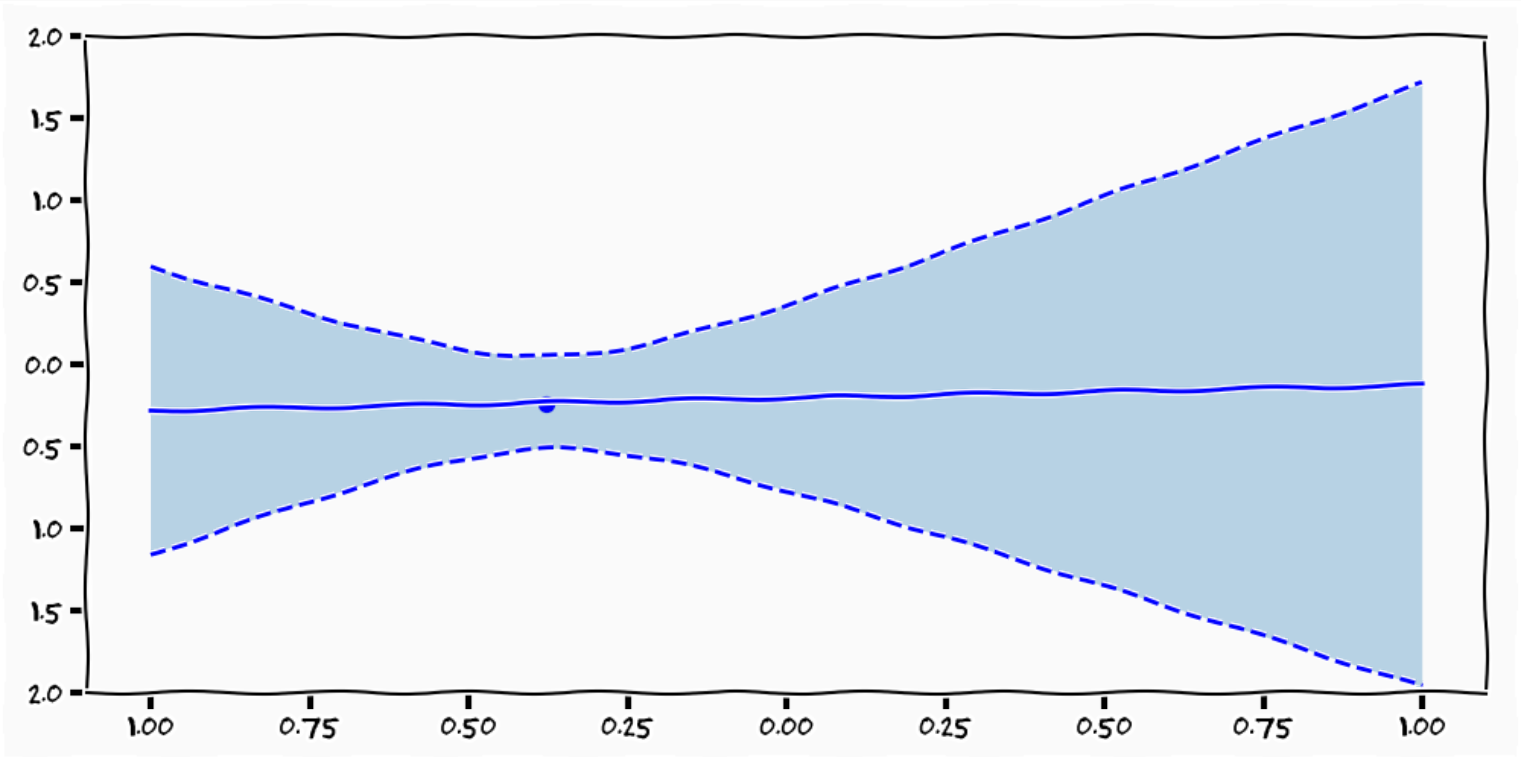
- we do not really care about $\mathbf{w}$ we care about new prediction t_* at location $\mathbf{x}_*$
- look at the marginal distribution, i.e. when we average out the weight
- integrate a Gaussian over a Gaussian $\Rightarrow$ Gaussian identities

Prediction

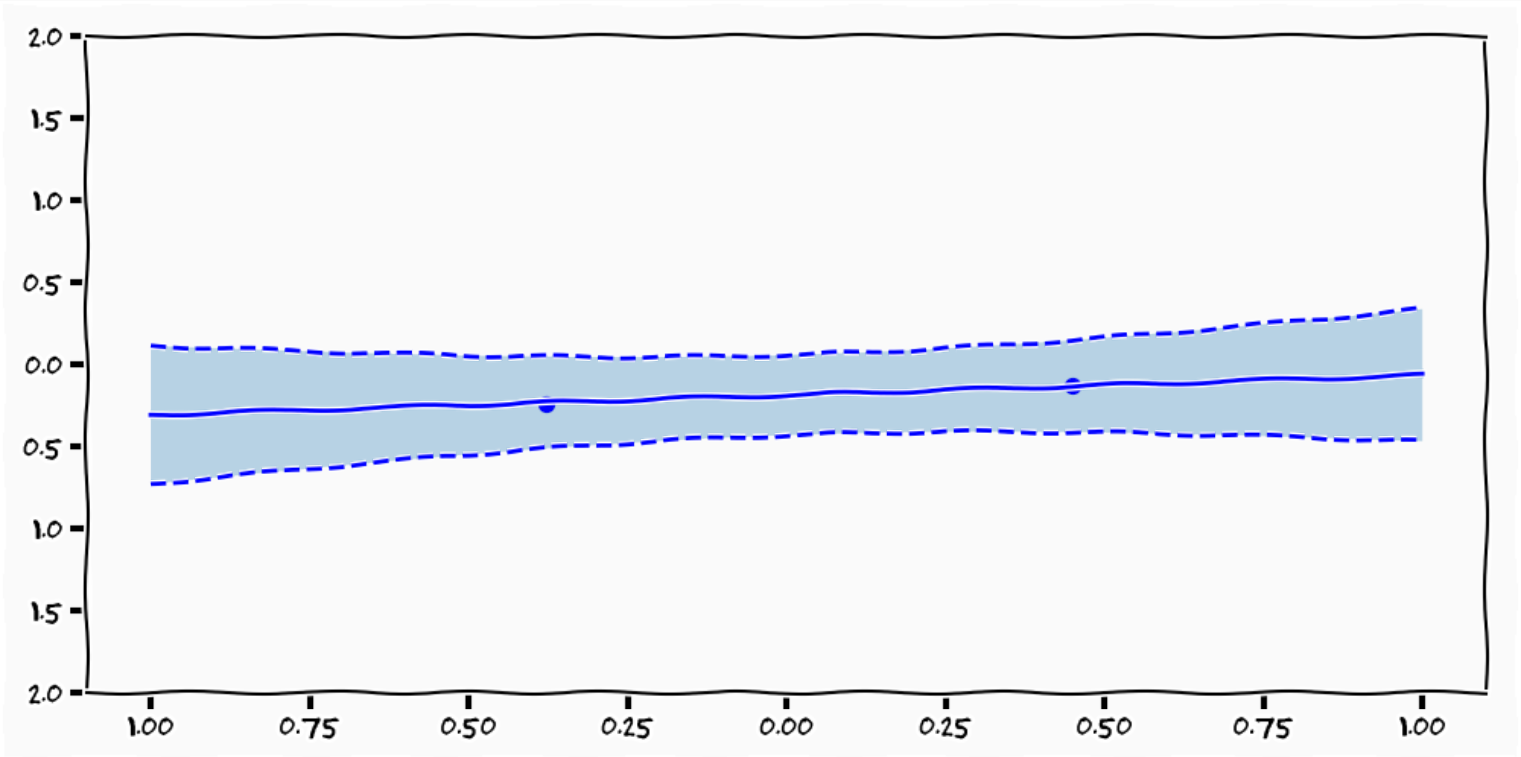


$$
\begin{aligned}
\text{"}p(t_*|\mathbf{x}_*)\text{"} &= \int p(t_*|\mathbf{x}_*, \mathbf{w}, \beta) p(\mathbf{w}|\mathbf{t}, \mathbf{X}, \alpha, \beta) \mathrm{d}\mathbf{w} \\
&= \mathcal{N}(t_*|\mathbf{m}_N^{\mathrm{T}} \phi(\mathbf{x}_*), \frac{1}{\beta} + \phi(\mathbf{x}_*)^{\mathrm{T}} \mathbf{S}_N \phi(\mathbf{x}_*))
\end{aligned}
$$

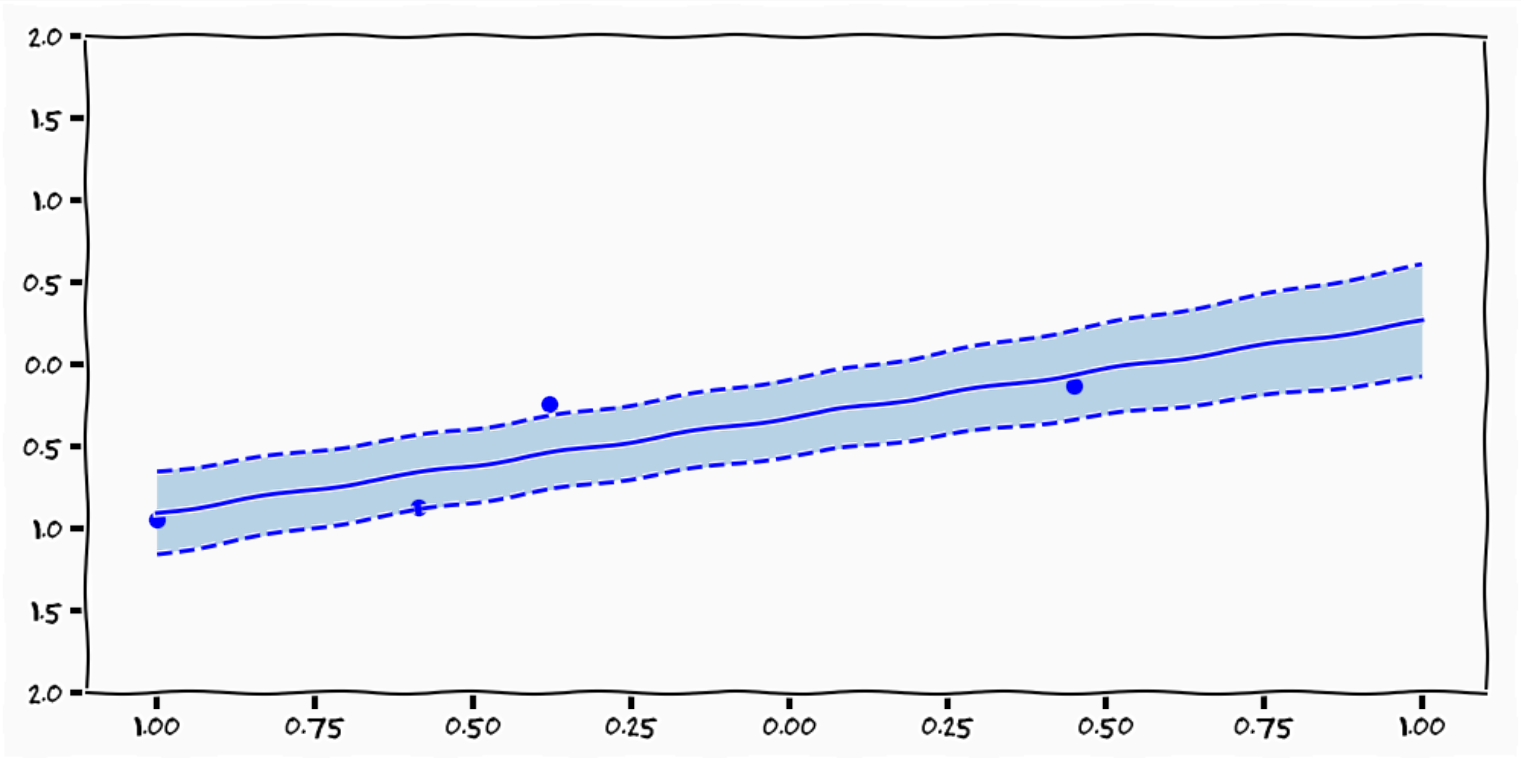
Predictive Posterior



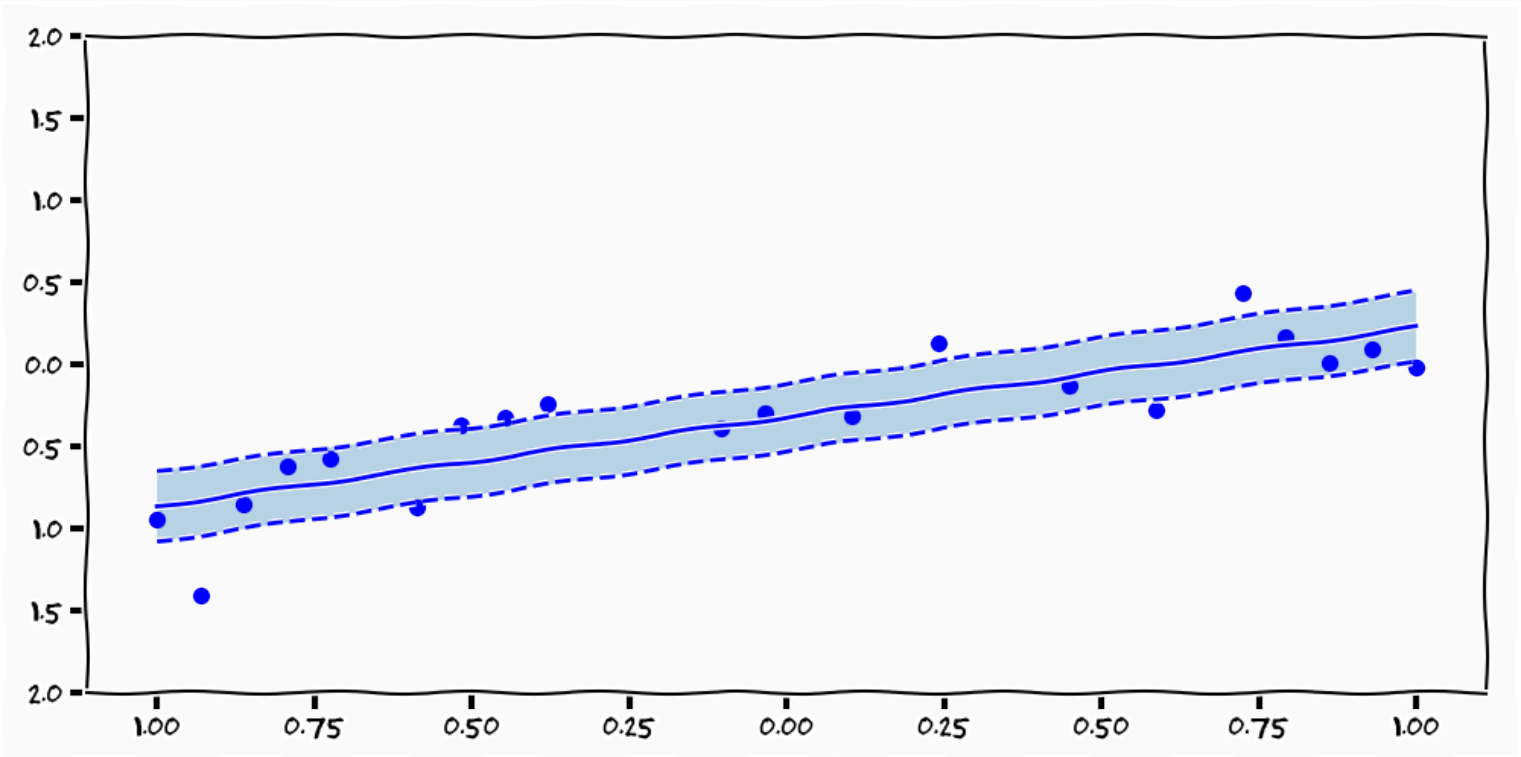
Predictive Posterior



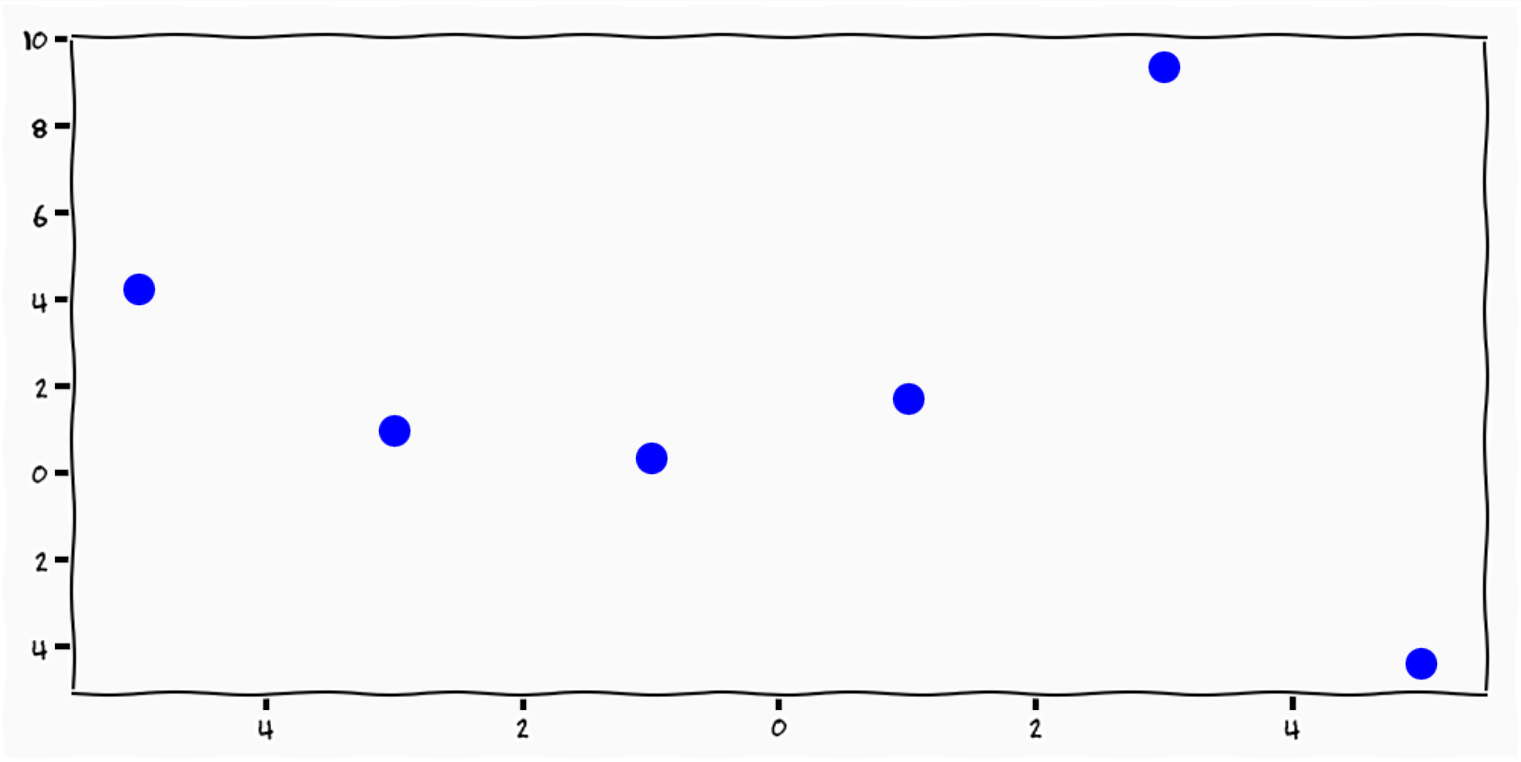
Predictive Posterior



Predictive Posterior



Linear Regression

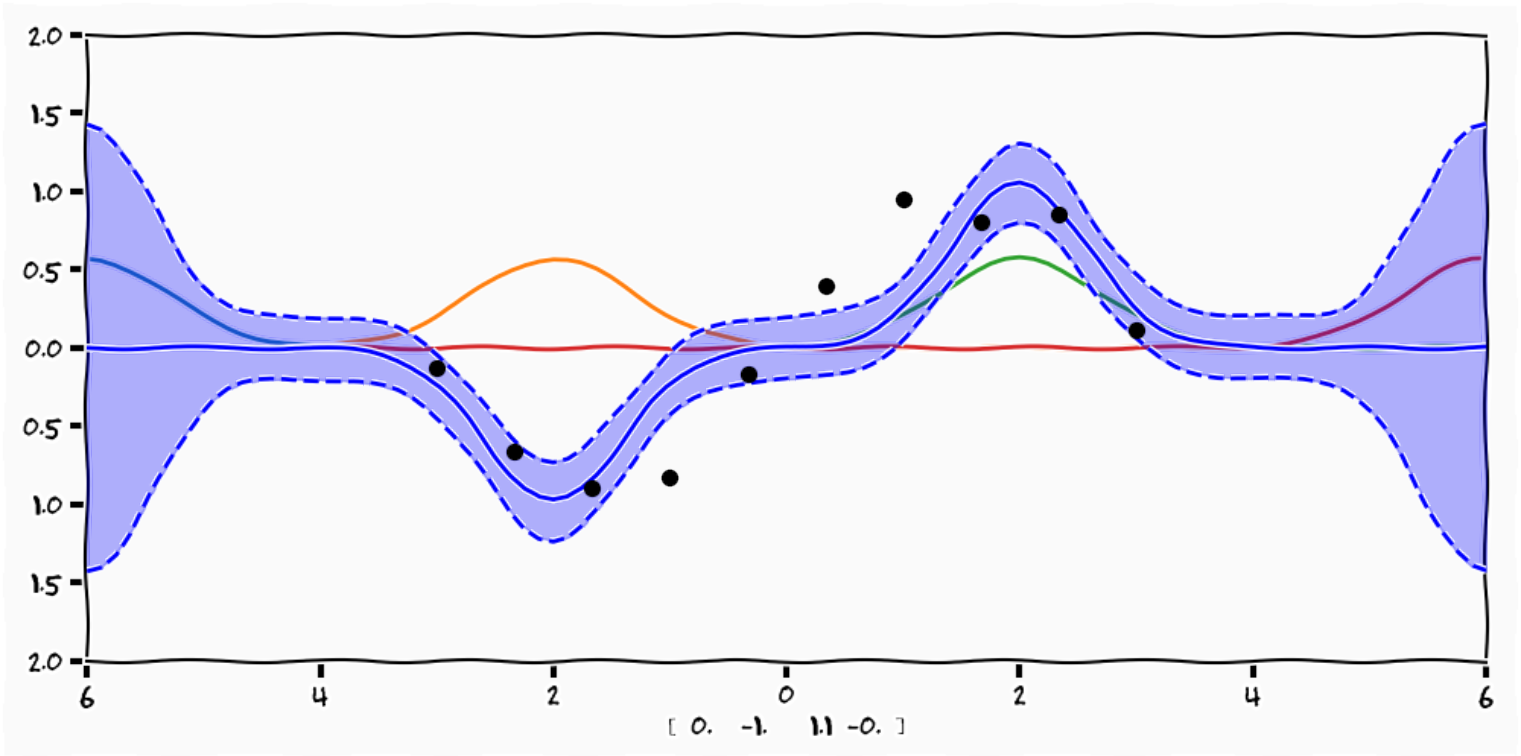


- Linear function only in parameters

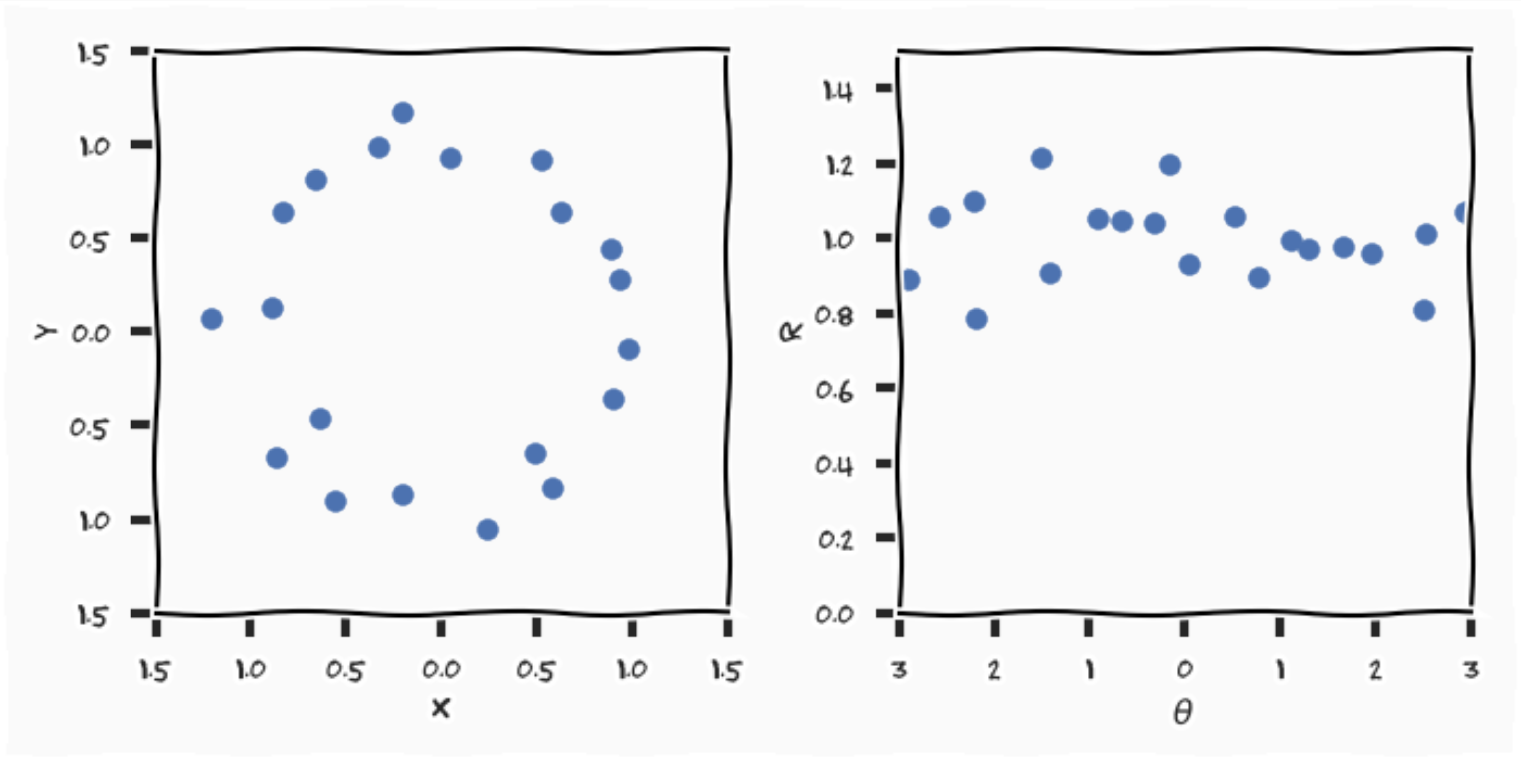
$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x}) = \{\phi_0(\mathbf{x}) = 1\} = \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x})$$

- We can choose many types of basis functions $\phi(\mathbf{x})$

Non-Linear Basis Functions



Change of Basis?



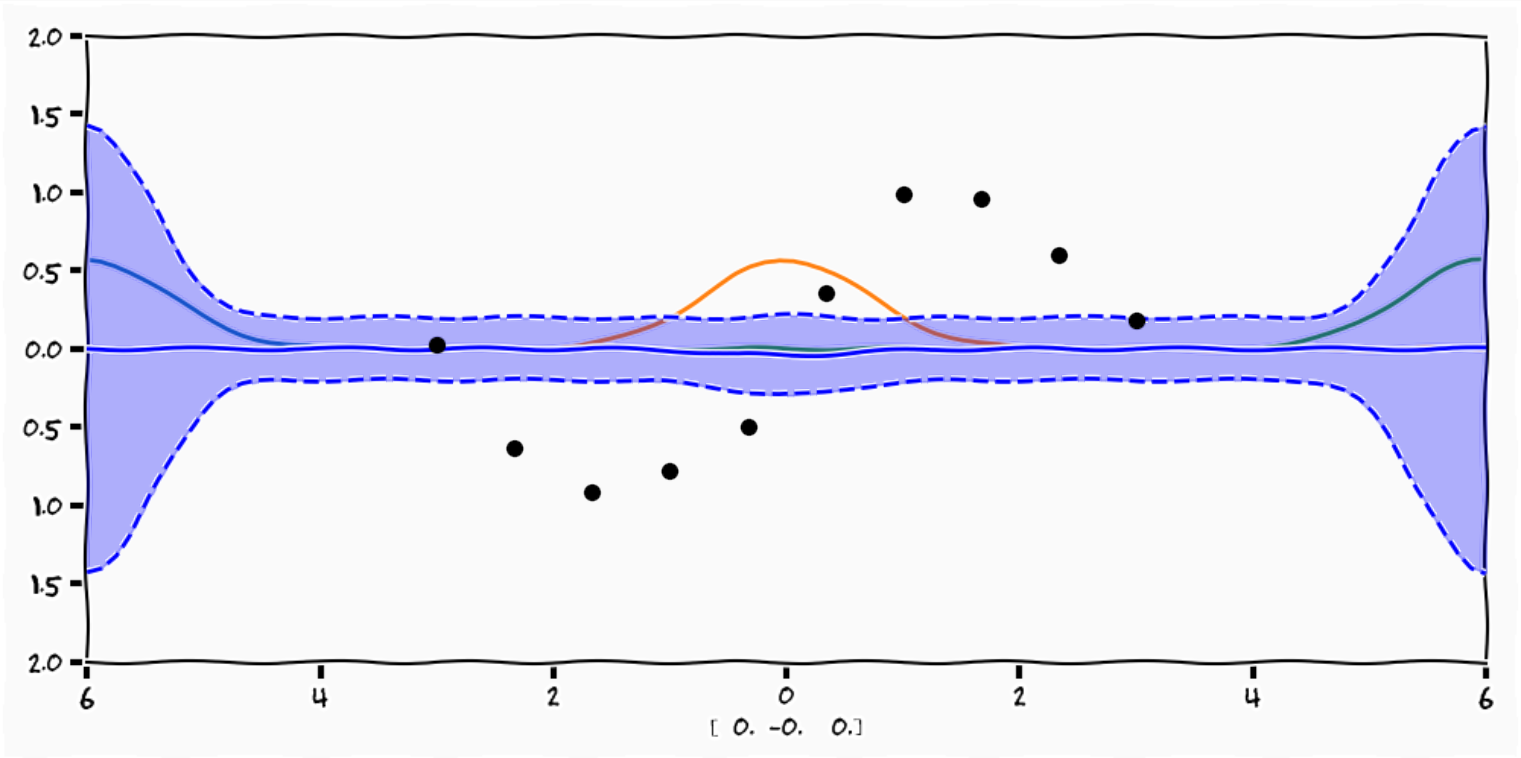
Change of Basis

$$y = \mathbf{w}^{\mathrm{T}} \phi(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \mathbf{z},$$

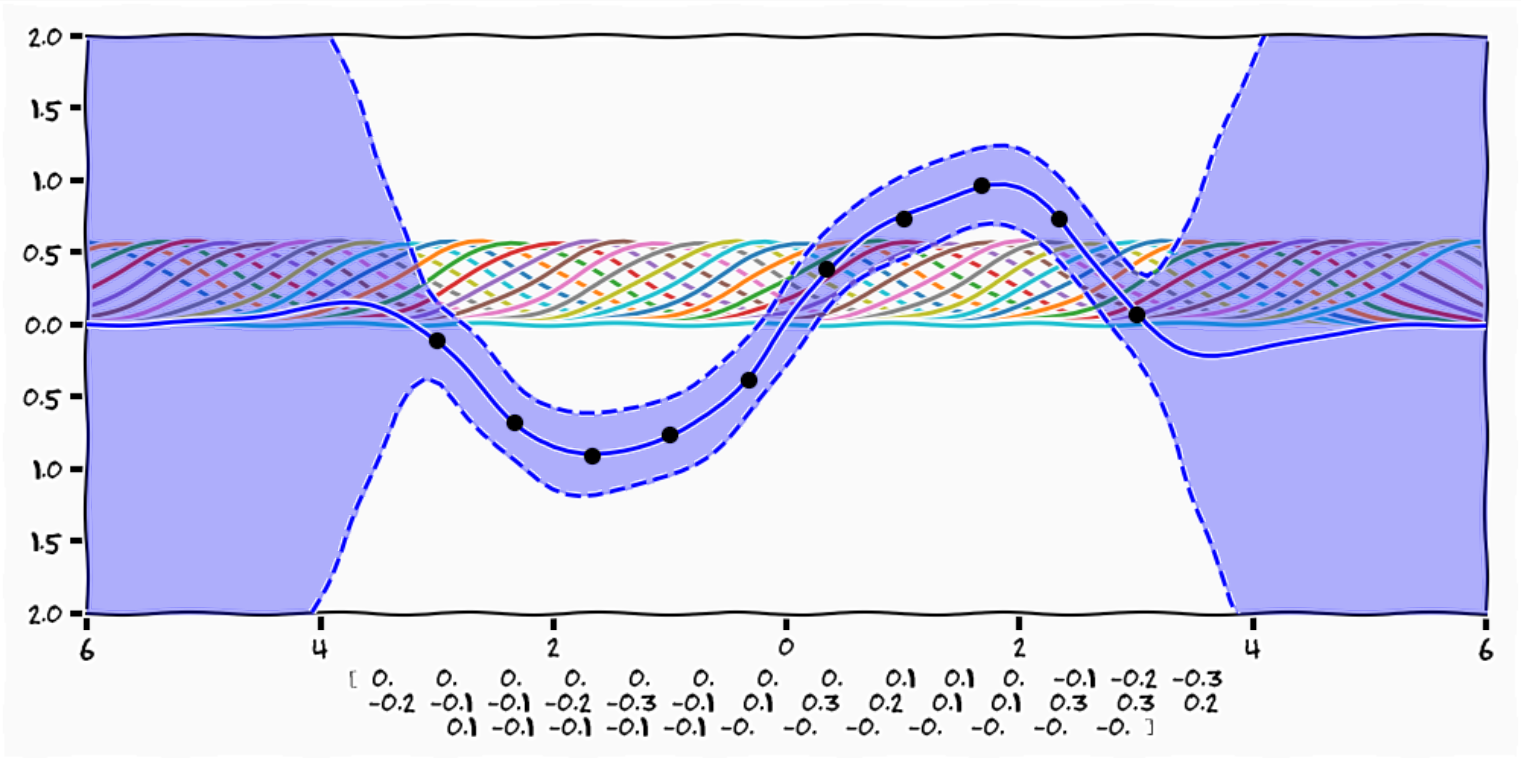
where $\phi(\cdot)$ is a mapping from,

$$\phi : \mathcal{X} \to \mathcal{Z}$$
$$\mathbf{x} \in \mathcal{X}$$
$$\mathbf{z} \in \mathcal{Z}.$$

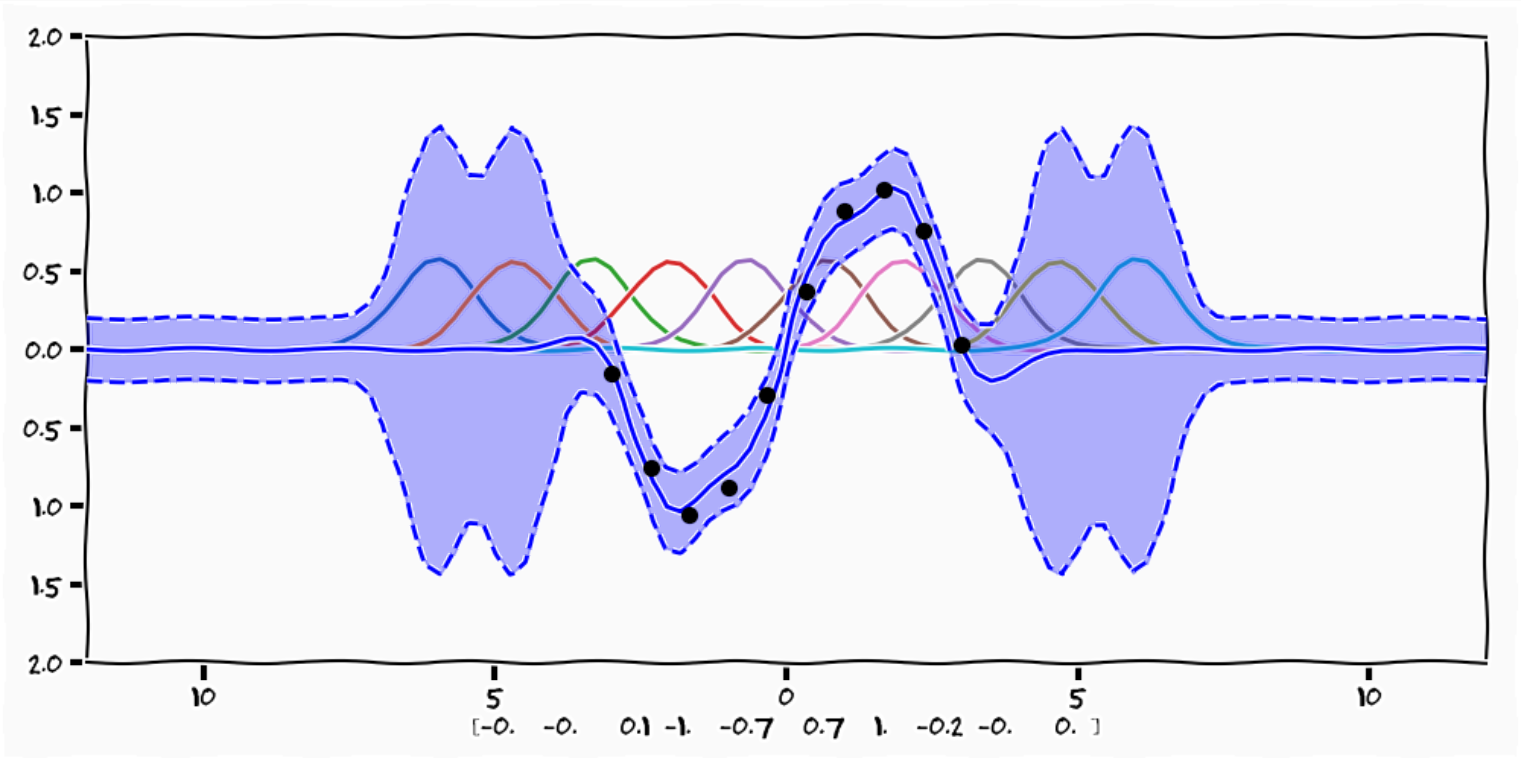
Non-Linear Basis Functions



Non-Linear Basis Functions



Non-Linear Basis Functions



Dual Linear Regression

Dual Linear Regression

$$p(\mathbf{w}|\mathbf{t},\mathbf{x}) = \frac{p(\mathbf{t}|\mathbf{w},\mathbf{x})p(\mathbf{w})}{p(\mathbf{t})}$$

$$p(\mathbf{t}|\mathbf{w},\mathbf{x}) = \prod_{n}^{N} p(t_n|\mathbf{w},\mathbf{x}) = \prod_{n}^{N} \mathcal{N}(t_n|\mathbf{w}^{\mathrm{T}}\mathbf{x}_n, \sigma^2\mathbf{I})$$

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{0}, \tau^2\mathbf{I})$$

Dual Linear Regression

$$p(\mathbf{w}|\mathbf{t}, \mathbf{x}) = \frac{p(\mathbf{t}|\mathbf{w}, \mathbf{x})p(\mathbf{w})}{p(\mathbf{t})}$$

$$p(\mathbf{t}|\mathbf{w}, \mathbf{x}) = \prod_n^N p(t_n|\mathbf{w}, \mathbf{x}) = \prod_n^N \mathcal{N}(t_n|\mathbf{w}^{\mathrm{T}}\mathbf{x}_n, \sigma^2\mathbf{I})$$

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{0}, \tau^2\mathbf{I})$$

$$p(\mathbf{w}|\mathbf{t}, \mathbf{x}) \propto p(\mathbf{t}|\mathbf{w}, \mathbf{x})p(\mathbf{w})$$

- Through conjugacy we know the form of the posterior

Dual Linear Regression

$$p(\mathbf{w}|\mathbf{t},\mathbf{x}) \propto \prod_n^N \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n - t_n)^{\mathrm{T}}(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n - y_n)} \frac{1}{\sqrt{2\pi\tau^2}} e^{-\frac{1}{2\tau^2}(\mathbf{w}^{\mathrm{T}}\mathbf{w})}$$

$$= \frac{1}{(\sqrt{2\pi\sigma^2})^N} e^{-\frac{1}{2\sigma^2}\mathrm{tr}\left((\mathbf{X}\mathbf{w}-\mathbf{t})^{\mathrm{T}}(\mathbf{X}\mathbf{w}-\mathbf{t})\right)} \frac{1}{(\sqrt{2\pi\tau^2})^N} e^{-\frac{1}{2\tau^2}(\mathbf{w}^{\mathrm{T}}\mathbf{w})}$$

- Lets maximise the above to find a point estimate (not a distribution) of $\mathbf{w}$

Dual Linear Regression [1] Ch 6.1

$$-\log p(\mathbf{w}|\mathbf{t}, \mathbf{x}) = J(\mathbf{w}) = \frac{1}{2}(\mathbf{X}\mathbf{w} - \mathbf{t})^{\mathrm{T}}(\mathbf{X}\mathbf{w} - \mathbf{t}) + \frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

- Find a stationary point in $\mathbf{w}$

Dual Linear Regression [1] Ch 6.1

$$-\log p(\mathbf{w}|\mathbf{t},\mathbf{x}) = J(\mathbf{w}) = \frac{1}{2}(\mathbf{X}\mathbf{w}-\mathbf{t})^{\mathrm{T}}(\mathbf{X}\mathbf{w}-\mathbf{t}) + \frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

$$\frac{\partial}{\partial \mathbf{w}} J(\mathbf{w}) = \frac{1}{2} 2\mathbf{X}^{\mathrm{T}}(\mathbf{X}\mathbf{w}-\mathbf{t}) + \frac{\lambda}{2} 2\mathbf{w}$$

- Find a stationary point in $\mathbf{w}$

Dual Linear Regression [1] Ch 6.1

$$-\log p(\mathbf{w}|\mathbf{t},\mathbf{x}) = J(\mathbf{w}) = \frac{1}{2}(\mathbf{X}\mathbf{w}-\mathbf{t})^{\mathrm{T}}(\mathbf{X}\mathbf{w}-\mathbf{t}) + \frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

$$\frac{\partial}{\partial \mathbf{w}} J(\mathbf{w}) = \frac{1}{2} 2\mathbf{X}^{\mathrm{T}}(\mathbf{X}\mathbf{w}-\mathbf{t}) + \frac{\lambda}{2} 2\mathbf{w}$$

$$\mathbf{w} = -\frac{1}{\lambda}\mathbf{X}^{\mathrm{T}}(\mathbf{X}\mathbf{w}-\mathbf{t})$$

- Find a stationary point in $\mathbf{w}$

Dual Linear Regression [1] Ch 6.1

$$
\begin{aligned}
-\log p(\mathbf{w}|\mathbf{t}, \mathbf{x}) = J(\mathbf{w}) &= \frac{1}{2}(\mathbf{X}\mathbf{w} - \mathbf{t})^{\mathrm{T}}(\mathbf{X}\mathbf{w} - \mathbf{t}) + \frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} \\
\frac{\partial}{\partial \mathbf{w}} J(\mathbf{w}) &= \frac{1}{2} 2\mathbf{X}^{\mathrm{T}}(\mathbf{X}\mathbf{w} - \mathbf{t}) + \frac{\lambda}{2} 2\mathbf{w} \\
\mathbf{w} &= -\frac{1}{\lambda}\mathbf{X}^{\mathrm{T}}(\mathbf{X}\mathbf{w} - \mathbf{t}) \\
&= \mathbf{X}^{\mathrm{T}}\mathbf{a} = \sum_{n}^{N} \alpha_n \mathbf{x}_n
\end{aligned}
$$

- Find a stationary point in $\mathbf{w}$

Dual Linear Regression

$$J(\mathbf{w}) = \frac{1}{2}(\mathbf{X}\mathbf{w} - \mathbf{t})^{\mathrm{T}}(\mathbf{X}\mathbf{w} - \mathbf{t}) + \frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

$$\mathbf{w} = \mathbf{X}^{\mathrm{T}}\mathbf{a}$$

- Rewrite objective in terms of $\mathbf{a}$

Dual Linear Regression

$$J(\mathbf{w}) = \frac{1}{2}(\mathbf{X}\mathbf{w} - \mathbf{t})^{\mathrm{T}}(\mathbf{X}\mathbf{w} - \mathbf{t}) + \frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

$$\mathbf{w} = \mathbf{X}^{\mathrm{T}}\mathbf{a}$$

$$J(\mathbf{a}) = \frac{1}{2}\mathbf{a}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{a} - \mathbf{a}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{t} + \frac{1}{2}\mathbf{t}^{\mathrm{T}}\mathbf{t} + \frac{\lambda}{2}\mathbf{a}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{a}$$

- Rewrite objective in terms of $\mathbf{a}$

Dual Linear Regression

$$[\mathbf{K}]_{ij} = \mathbf{x}_i^{\mathrm{T}} \mathbf{x}_j$$

$$J(\mathbf{a}) = \frac{1}{2}\mathbf{a}^{\mathrm{T}}\mathbf{K}\mathbf{K}\mathbf{a} - \mathbf{a}\mathbf{K}\mathbf{t} + \frac{1}{2}\mathbf{t}^{\mathrm{T}}\mathbf{t} + \frac{\lambda}{2}\mathbf{a}^{\mathrm{T}}\mathbf{K}\mathbf{a}$$

- $\mathbf{K}$ is a matrix with all inner-products between the data points

Dual Linear Regression

$$\alpha_n = -\frac{1}{\lambda}(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n - t_n)$$

$$\mathbf{w} = \sum_{n}^{N} \alpha_n \mathbf{x}_n = \mathbf{X}^{\mathrm{T}}\mathbf{a}$$

- Eliminate $\mathbf{w}$ and rewrite in terms of $\mathbf{a}$

Dual Linear Regression

$$\alpha_n = -\frac{1}{\lambda}(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n - t_n)$$

$$\mathbf{w} = \sum_{n}^{N} \alpha_n \mathbf{x}_n = \mathbf{X}^{\mathrm{T}}\mathbf{a}$$

$$\Rightarrow \mathbf{a} = (\mathbf{K} + \lambda \mathbf{I})^{-1}\mathbf{t}$$

- Eliminate $\mathbf{w}$ and rewrite in terms of $\mathbf{a}$

Dual Linear Regression

$$[\mathbf{K}]_{ij} = \mathbf{x}_i^{\mathrm{T}} \mathbf{x}_j$$

$$J(\mathbf{a}) = \frac{1}{2}\mathbf{a}^{\mathrm{T}}\mathbf{K}\mathbf{K}\mathbf{a} - \mathbf{a}\mathbf{K}\mathbf{t} + \frac{1}{2}\mathbf{t}^{\mathrm{T}}\mathbf{t} + \frac{\lambda}{2}\mathbf{a}^{\mathrm{T}}\mathbf{K}\mathbf{a}$$

$$\mathbf{a} = (\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{t}$$

Dual Linear Regression

$$
\begin{aligned}
[\mathbf{K}]_{ij} &= \mathbf{x}_i^{\mathrm{T}} \mathbf{x}_j \\
J(\mathbf{a}) &= \frac{1}{2}\mathbf{a}^{\mathrm{T}}\mathbf{K}\mathbf{K}\mathbf{a} - \mathbf{a}\mathbf{K}\mathbf{t} + \frac{1}{2}\mathbf{t}^{\mathrm{T}}\mathbf{t} + \frac{\lambda}{2}\mathbf{a}^{\mathrm{T}}\mathbf{K}\mathbf{a} \\
\mathbf{a} &= (\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{t} \\
\mathbf{y}(\mathbf{x}_*) &= \mathbf{w}^{\mathrm{T}}\mathbf{x}_* = \mathbf{a}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{x}_* = \mathbf{a}^{\mathrm{T}}k(\mathbf{x}, \mathbf{x}_*) = \\
&= ((\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{t})^{\mathrm{T}}k(\mathbf{x}, \mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{x})(\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{t}
\end{aligned}
$$

What have we actually done

- Linear Regression
  - See data
  - Encode relationship between variates using parameters $\mathbf{w}$
  - Make predictions using $\mathbf{w}$

What have we actually done

- Linear Regression
  - See data
  - Encode relationship between variates using parameters $\mathbf{w}$
  - Make predictions using $\mathbf{w}$
- Dual
  - See Data
  - Encode relationship between variates using variates themselves

What have we actually done

- Linear Regression
  - See data
  - Encode relationship between variates using parameters $\mathbf{w}$
  - Make predictions using $\mathbf{w}$
- Dual
  - See Data
  - Encode relationship between variates using variates themselves
  - *Model complexity depends on data*

What have we actually done

- Linear Regression
  - See data
  - Encode relationship between variates using parameters $\mathbf{w}$
  - Make predictions using $\mathbf{w}$
- Dual
  - See Data
  - Encode relationship between variates using variates themselves
  - *Model complexity depends on data*
  - Non-parametric model

Kernel Functions

$$\phi : \mathbf{x}_i \rightarrow \mathbf{f}_i$$

$$\mathbf{y}(\mathbf{x}_*) = \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_*) = \mathbf{a}^{\mathrm{T}}\phi(\mathbf{X})\phi(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{X})(\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{y}$$

$$k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^{\mathrm{T}}\phi(\mathbf{x}')$$

- we actually never need to know $\phi(\mathbf{x})$ only $\phi(\mathbf{x}_i)^{\mathrm{T}}\phi(\mathbf{x}_j)$
- functions that describes inner-products are called *kernel-functions*

Kernel Functions

$$\mathbf{x} \in \mathbb{R}^2$$

$$k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^{\mathrm{T}} \mathbf{x}_j)^2$$

- Kernel functions need to forefill certain properties and is a subclass of functions
- Can be incredibly useful, think similarity rather than location

Kernel Functions

$$\mathbf{x} \in \mathbb{R}^2$$

$$k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^{\mathrm{T}} \mathbf{x}_j)^2 = (x_{i1} x_{j1} + x_{i2} x_{j2})^2$$

- Kernel functions need to forefill certain properties and is a subclass of functions
- Can be incredibly useful, think similarity rather than location

Kernel Functions

$$\mathbf{x} \in \mathbb{R}^2$$

$$k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^{\mathrm{T}} \mathbf{x}_j)^2 = (x_{i1} x_{j1} + x_{i2} x_{j2})^2$$
$$= x_{i1}^2 x_{j1}^2 + 2 x_{i1} x_{j1} x_{i2} x_{j2} + x_{i2}^2 x_{j2}^2$$

- Kernel functions need to forefill certain properties and is a subclass of functions
- Can be incredibly useful, think similarity rather than location

Kernel Functions

$$\mathbf{x} \in \mathbb{R}^2$$

$$\begin{aligned} k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^{\mathrm{T}} \mathbf{x}_j)^2 &= (x_{i1} x_{j1} + x_{i2} x_{j2})^2 \\ &= x_{i1}^2 x_{j1}^2 + 2 x_{i1} x_{j1} x_{i2} x_{j2} + x_{i2}^2 x_{j2}^2 = \\ &= (x_{i1}^2, \sqrt{2} x_{i1} x_{i2}, x_{i2}^2)(x_{j1}^2, \sqrt{2} x_{j1} x_{j2}, x_{j2}^2)^{\mathrm{T}} \end{aligned}$$

- Kernel functions need to forefill certain properties and is a subclass of functions
- Can be incredibly useful, think similarity rather than location

Kernel Functions

$$
\begin{aligned}
& & \mathbf{x} \in \mathbb{R}^2 \\
k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^{\mathrm{T}} \mathbf{x}_j)^2 &= (x_{i1} x_{j1} + x_{i2} x_{j2})^2 \\
&= x_{i1}^2 x_{j1}^2 + 2 x_{i1} x_{j1} x_{i2} x_{j2} + x_{i2}^2 x_{j2}^2 = \\
&= (x_{i1}^2, \sqrt{2} x_{i1} x_{i2}, x_{i2}^2)(x_{j1}^2, \sqrt{2} x_{j1} x_{j2}, x_{j2}^2)^{\mathrm{T}} = \\
&= \phi(\mathbf{x}_i)^{\mathrm{T}} \phi(\mathbf{x}_j)
\end{aligned}
$$

- Kernel functions need to forefill certain properties and is a subclass of functions
- Can be incredibly useful, think similarity rather than location

Kernel Functions

$$
\begin{aligned}
\mathbf{x} &\in \mathbb{R}^2 \\
k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^{\mathrm{T}} \mathbf{x}_j)^2 &= (x_{i1} x_{j1} + x_{i2} x_{j2})^2 \\
&= x_{i1}^2 x_{j1}^2 + 2 x_{i1} x_{j1} x_{i2} x_{j2} + x_{i2}^2 x_{j2}^2 = \\
&= (x_{i1}^2, \sqrt{2} x_{i1} x_{i2}, x_{i2}^2)(x_{j1}^2, \sqrt{2} x_{j1} x_{j2}, x_{j2}^2)^{\mathrm{T}} = \\
&= \phi(\mathbf{x}_i)^{\mathrm{T}} \phi(\mathbf{x}_j) \\
\phi(\mathbf{x}) &= ((\mathbf{e}_1^{\mathrm{T}} \mathbf{x})^2, \sqrt{2} \mathbf{e}_1^{\mathrm{T}} \mathbf{x} \mathbf{e}_2^{\mathrm{T}} \mathbf{x}, (\mathbf{e}_2^{\mathrm{T}} \mathbf{x})^2)
\end{aligned}
$$

- Kernel functions need to forefill certain properties and is a subclass of functions
- Can be incredibly useful, think similarity rather than location

Kernel Functions

- Kernels allows for *implicit* feature mappings

Kernel Functions

- Kernels allows for *implicit* feature mappings
- We do NOT need to know the feature space

Kernel Functions

- Kernels allows for *implicit* feature mappings
- We do NOT need to know the feature space
- The space can have infinite dimensionality

Kernel Functions

- Kernels allows for *implicit* feature mappings
- We do NOT need to know the feature space
- The space can have infinite dimensionality
- The mapping can be non-linear but the problem is still linear!

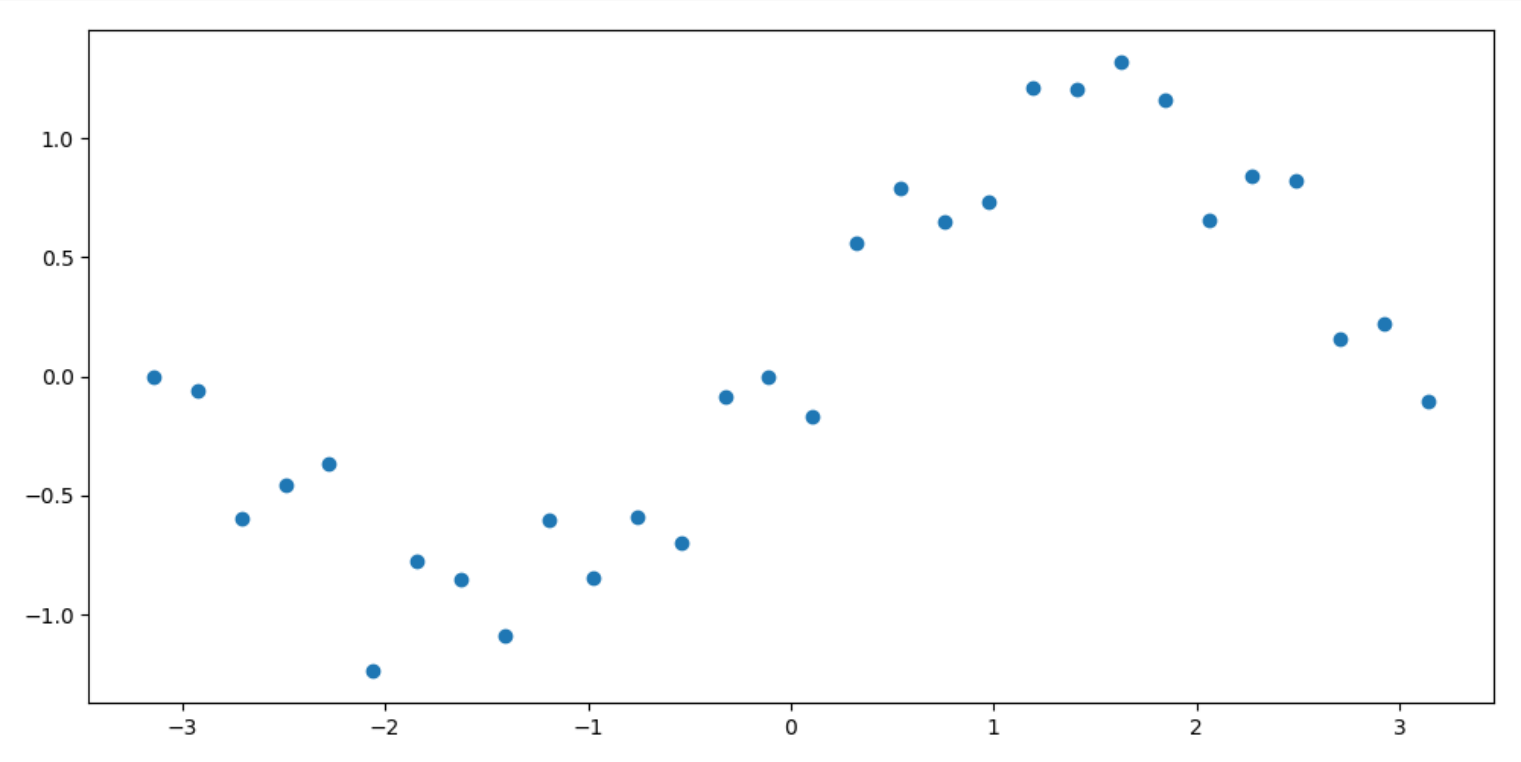
Kernel Functions

- Kernels allows for *implicit* feature mappings
- We do NOT need to know the feature space
- The space can have infinite dimensionality
- The mapping can be non-linear but the problem is still linear!
- Allows for putting weird things like, strings (DNA) in a vector space

Kernel Functions

- Kernels allows for *implicit* feature mappings
- We do NOT need to know the feature space
- The space can have infinite dimensionality
- The mapping can be non-linear but the problem is still linear!
- Allows for putting weird things like, strings (DNA) in a vector space
- More next lecture, these things are very powerful

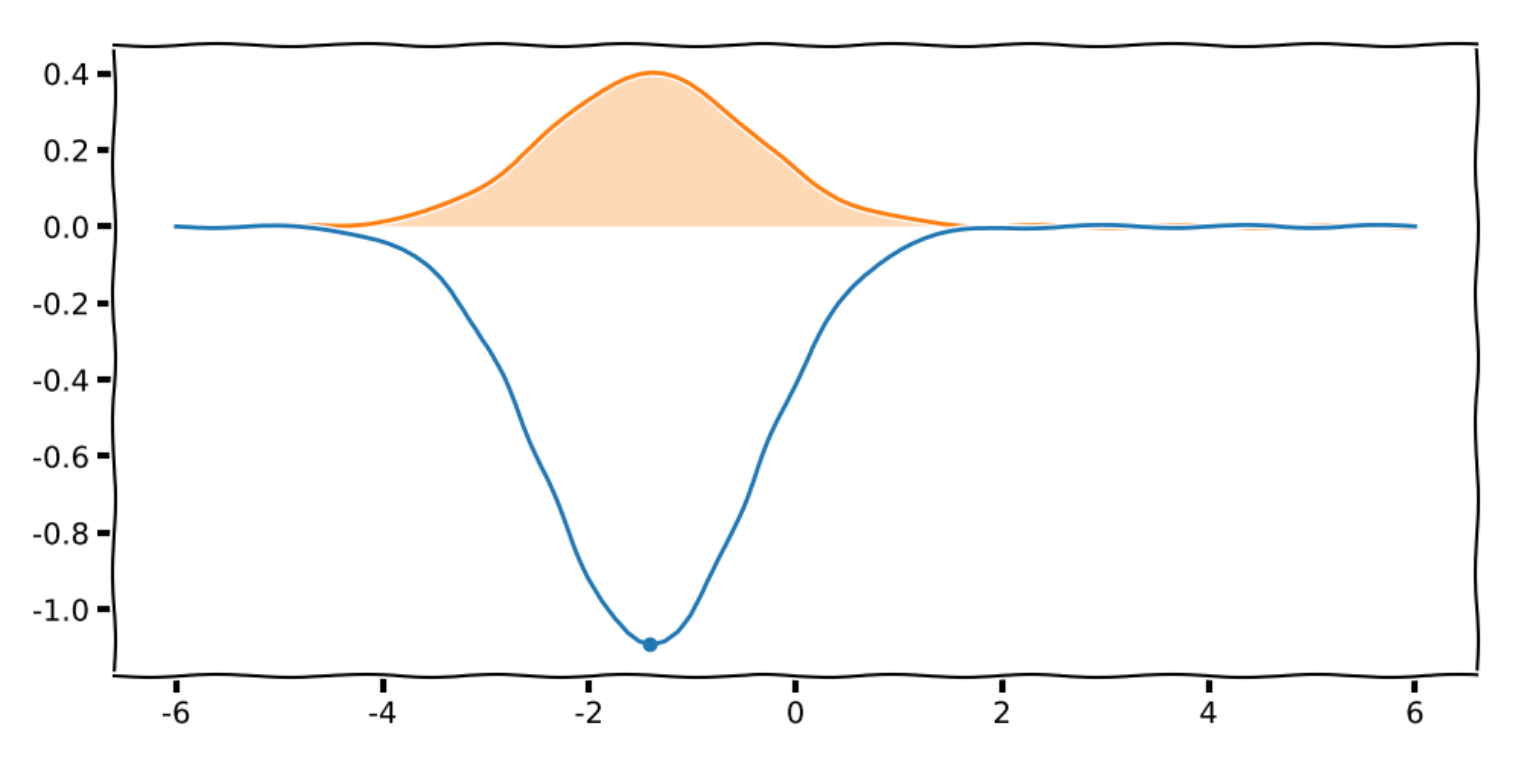
Kernel Functions



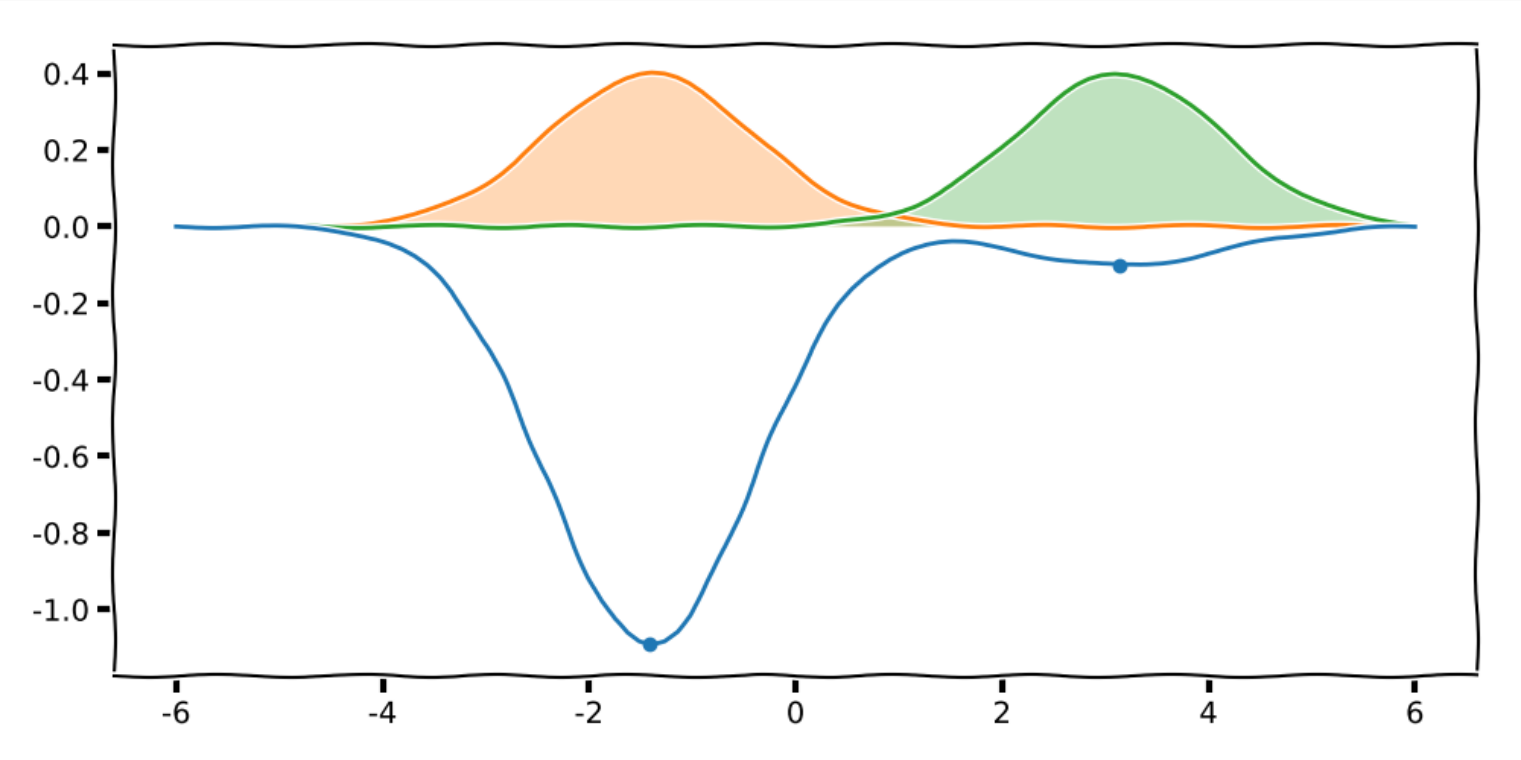
$$t = f(x) + \epsilon$$

$$k(x_i, x_j) = e^{-\frac{1}{2}\frac{(x_i - x_j)^2}{l}}$$

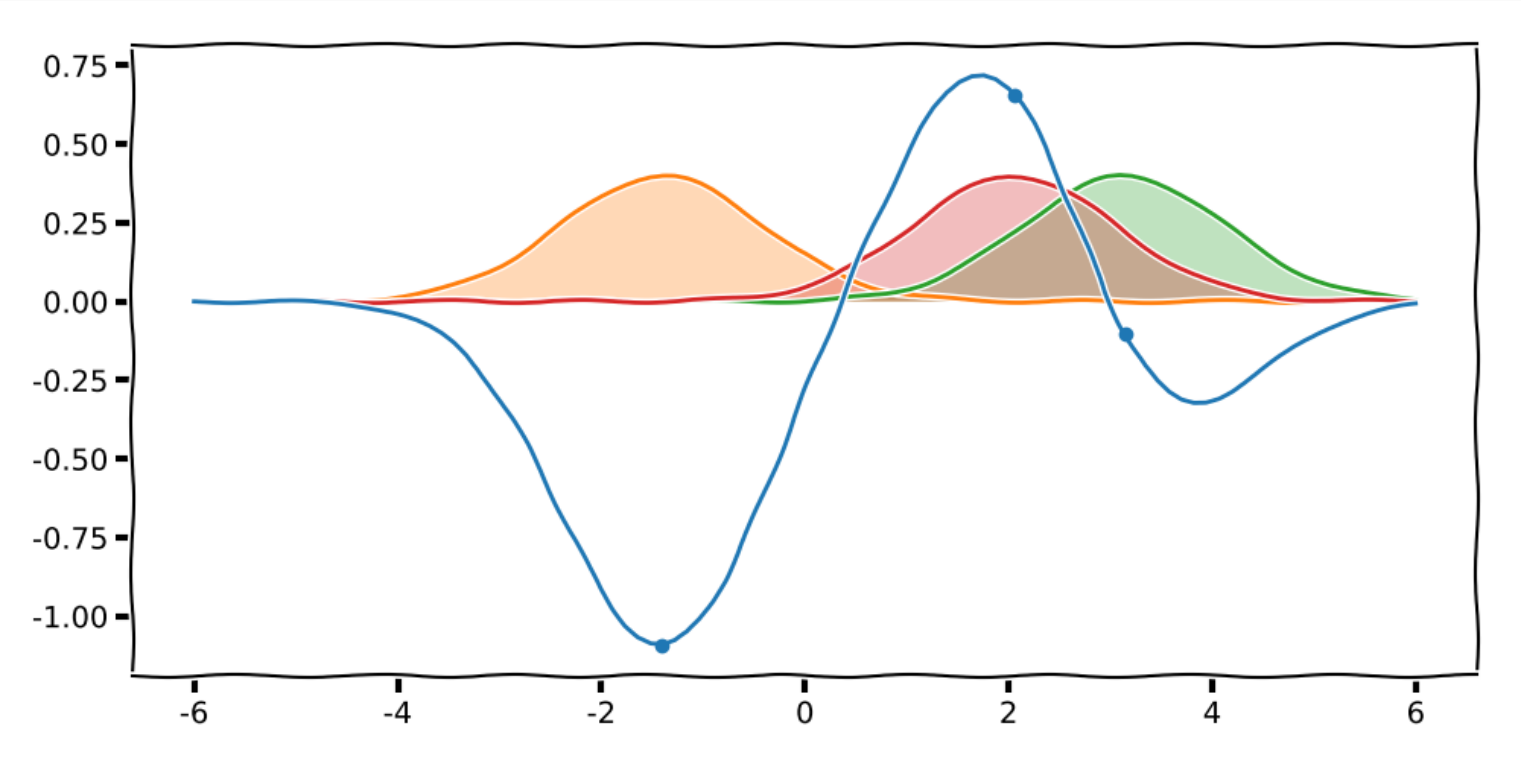
Kernel Regression



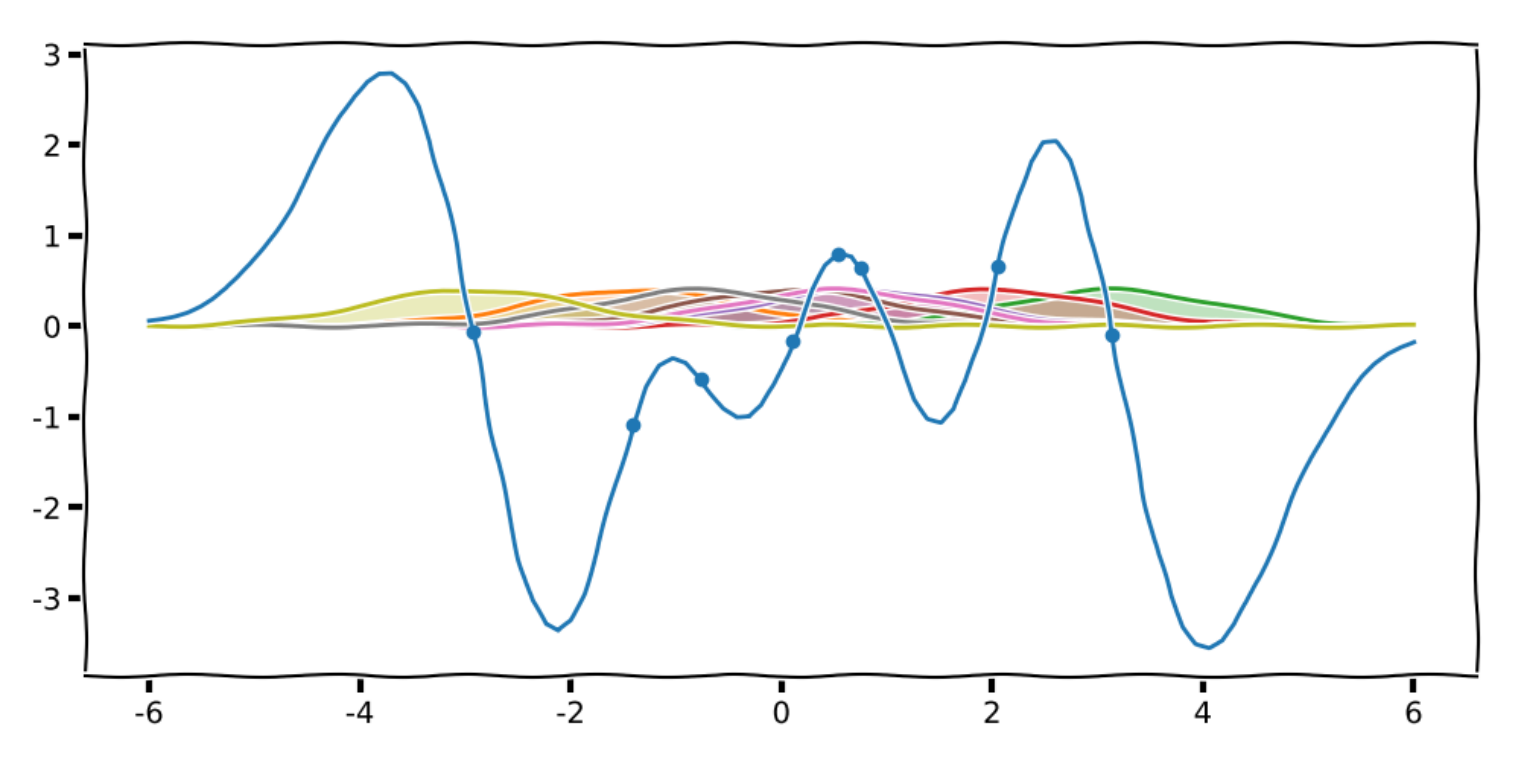
Kernel Regression



Kernel Regression

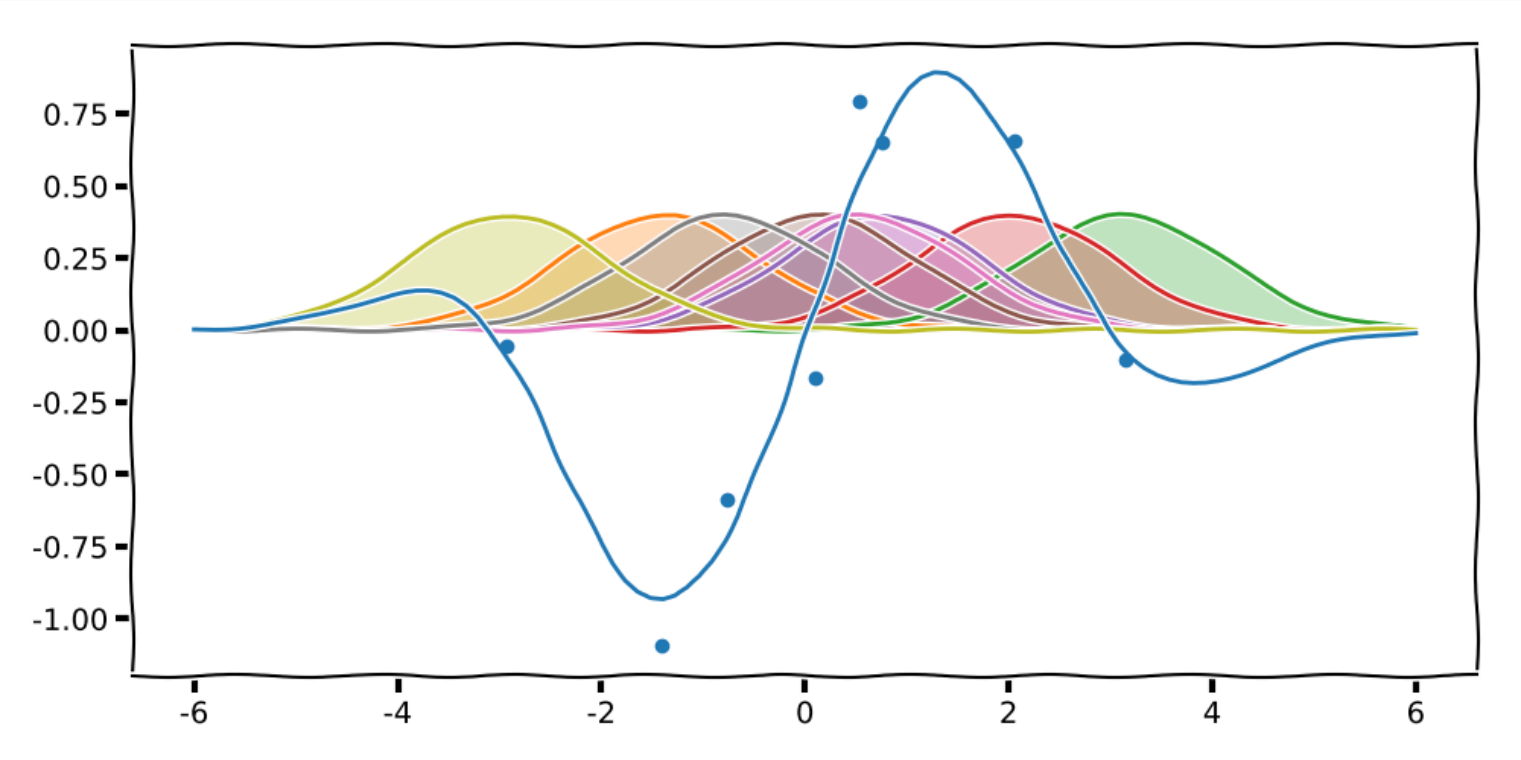


Kernel Regression

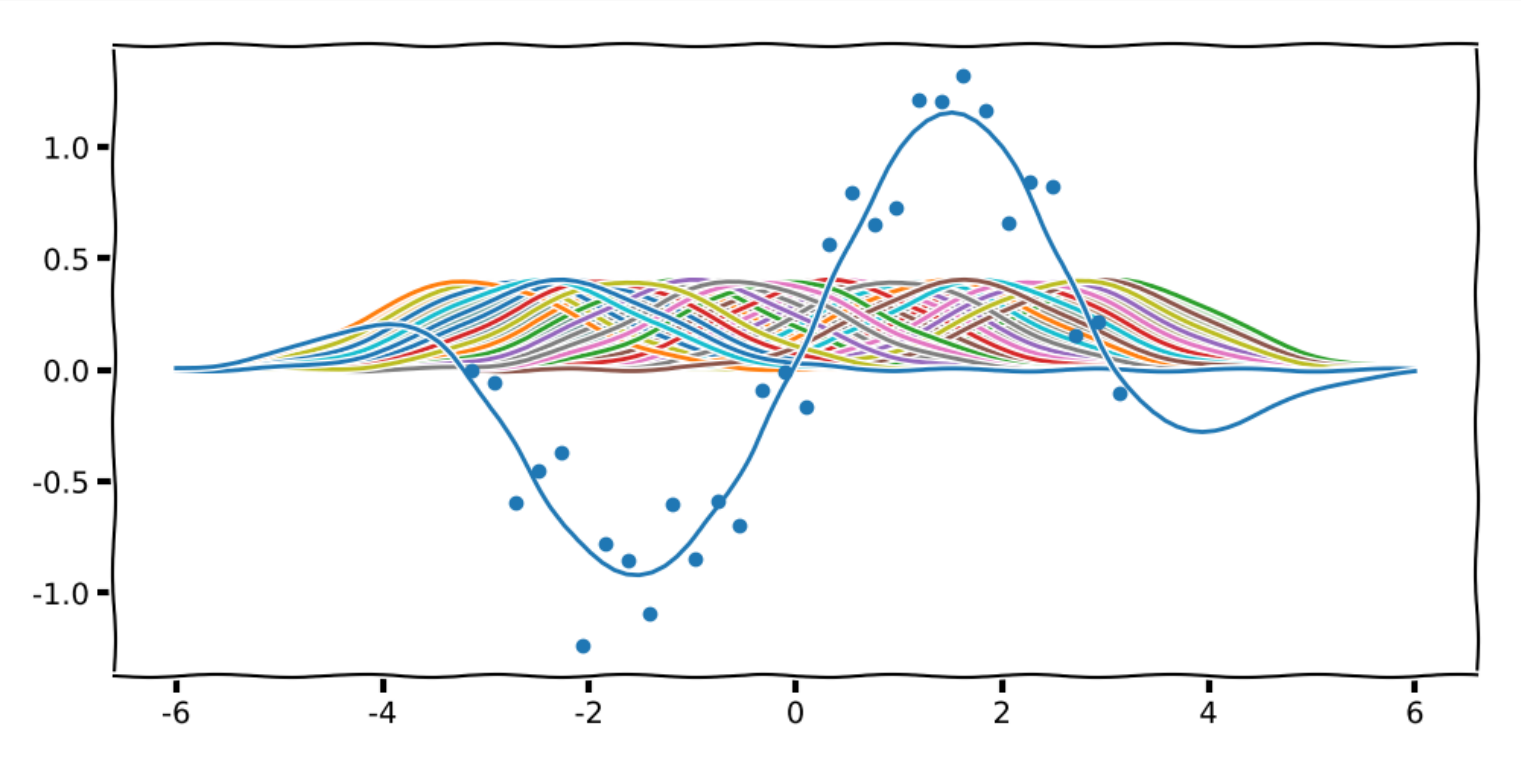


$$\mathbf{y}(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{x})(\mathbf{K} + \lambda \mathbf{I})^{-1}\mathbf{t}$$

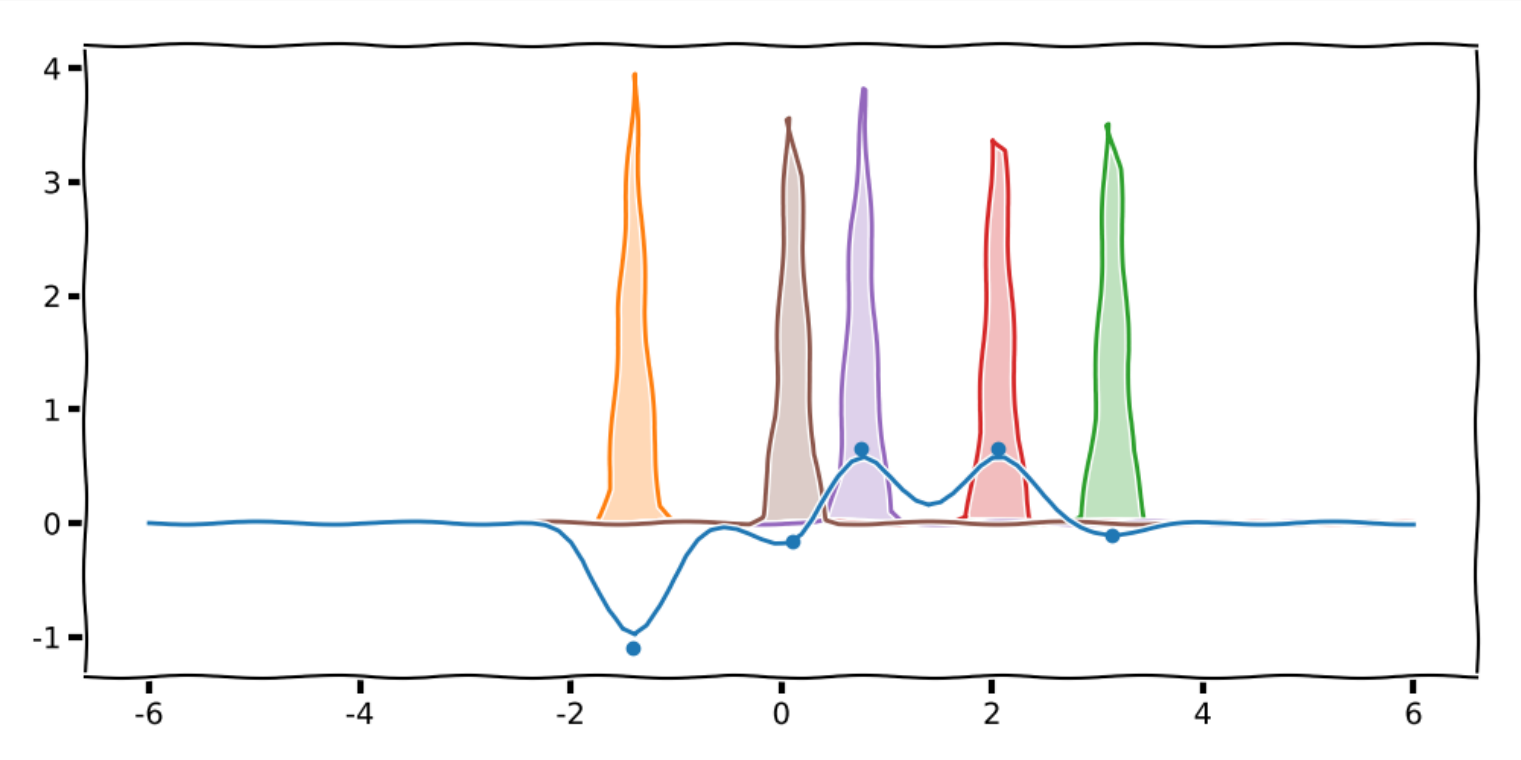
Kernel Regression



Kernel Regression

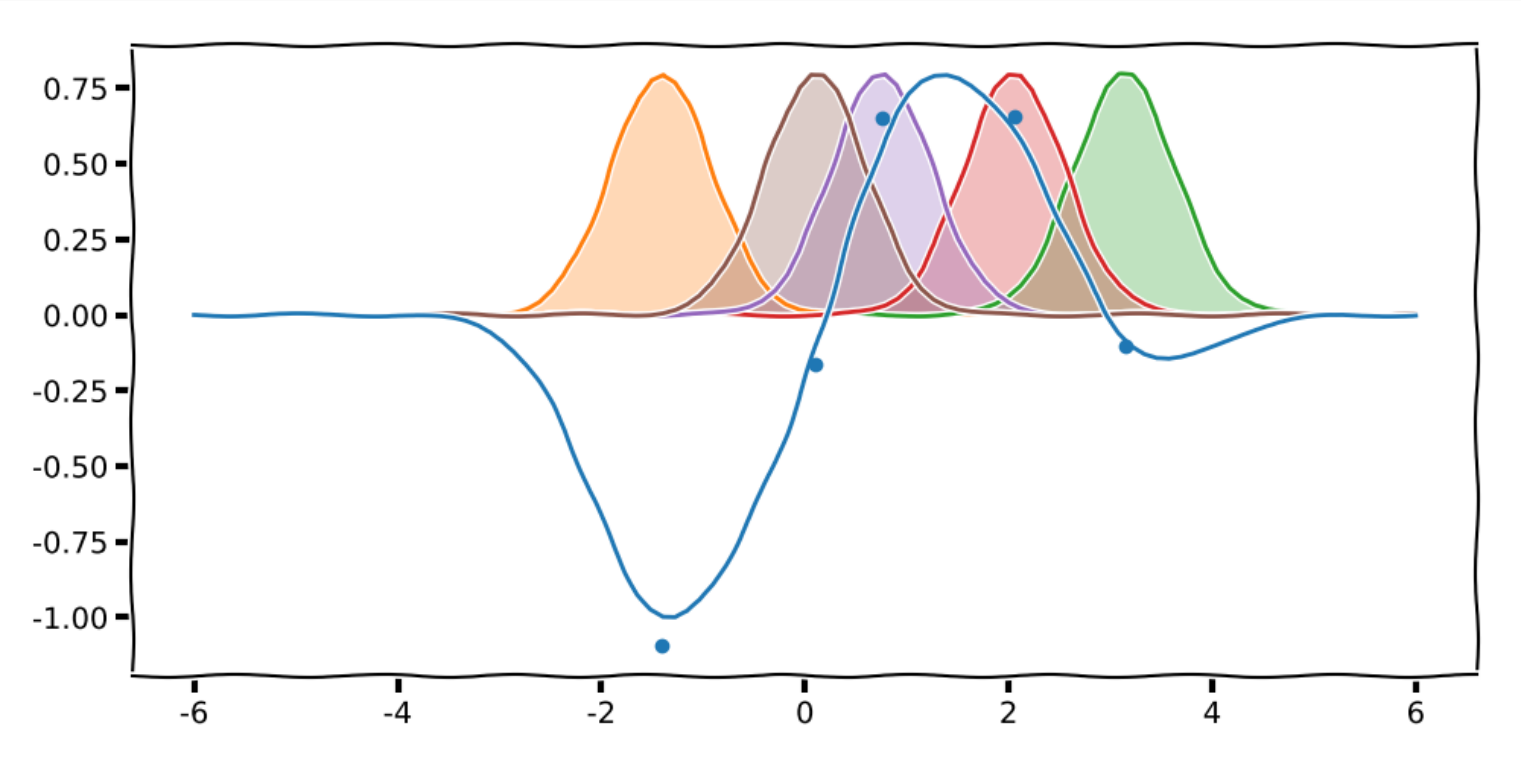


Kernel Regression

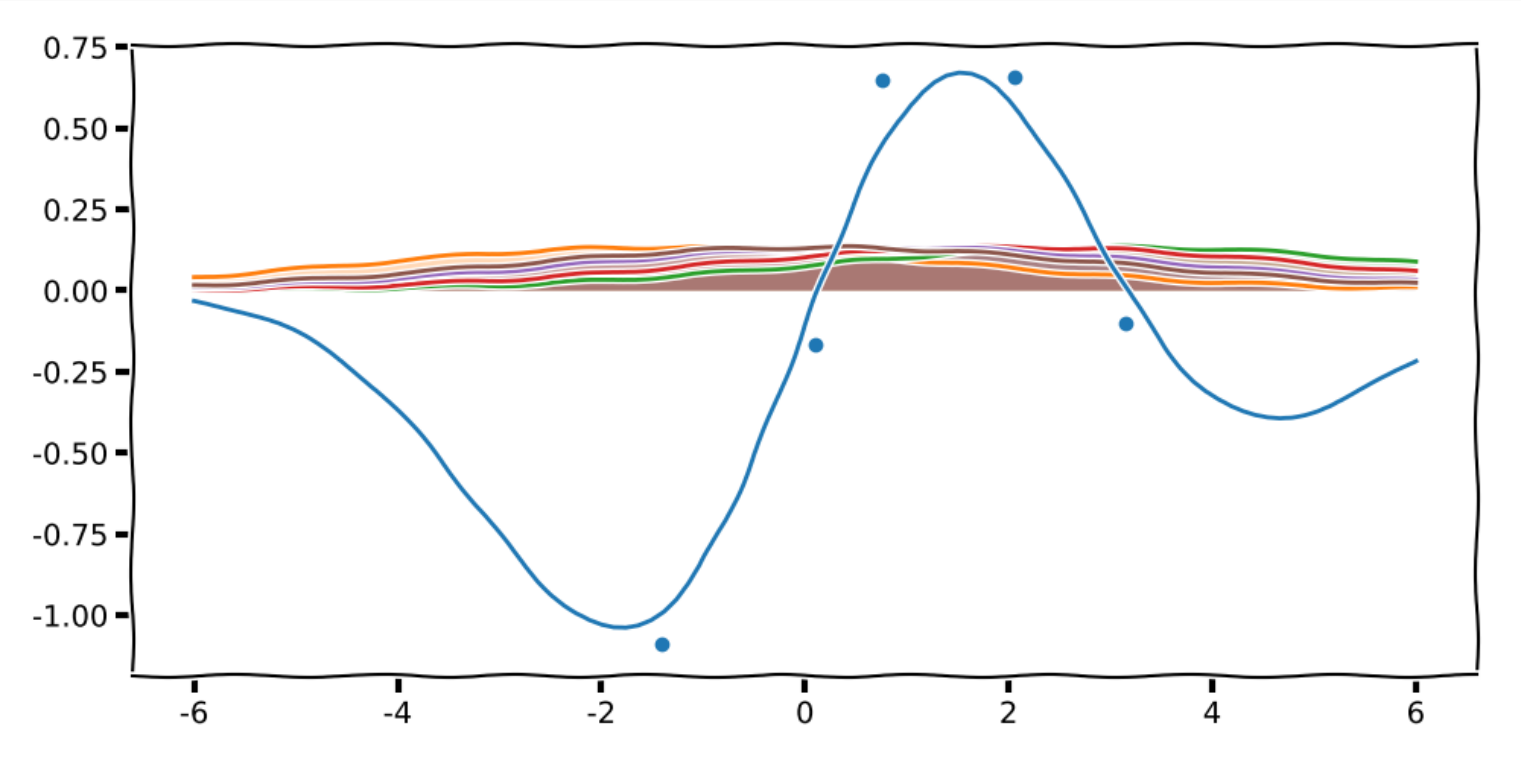


$$k(x_i, x_j) = e^{-\frac{1}{2}\frac{(x_i - x_j)^2}{l}}$$

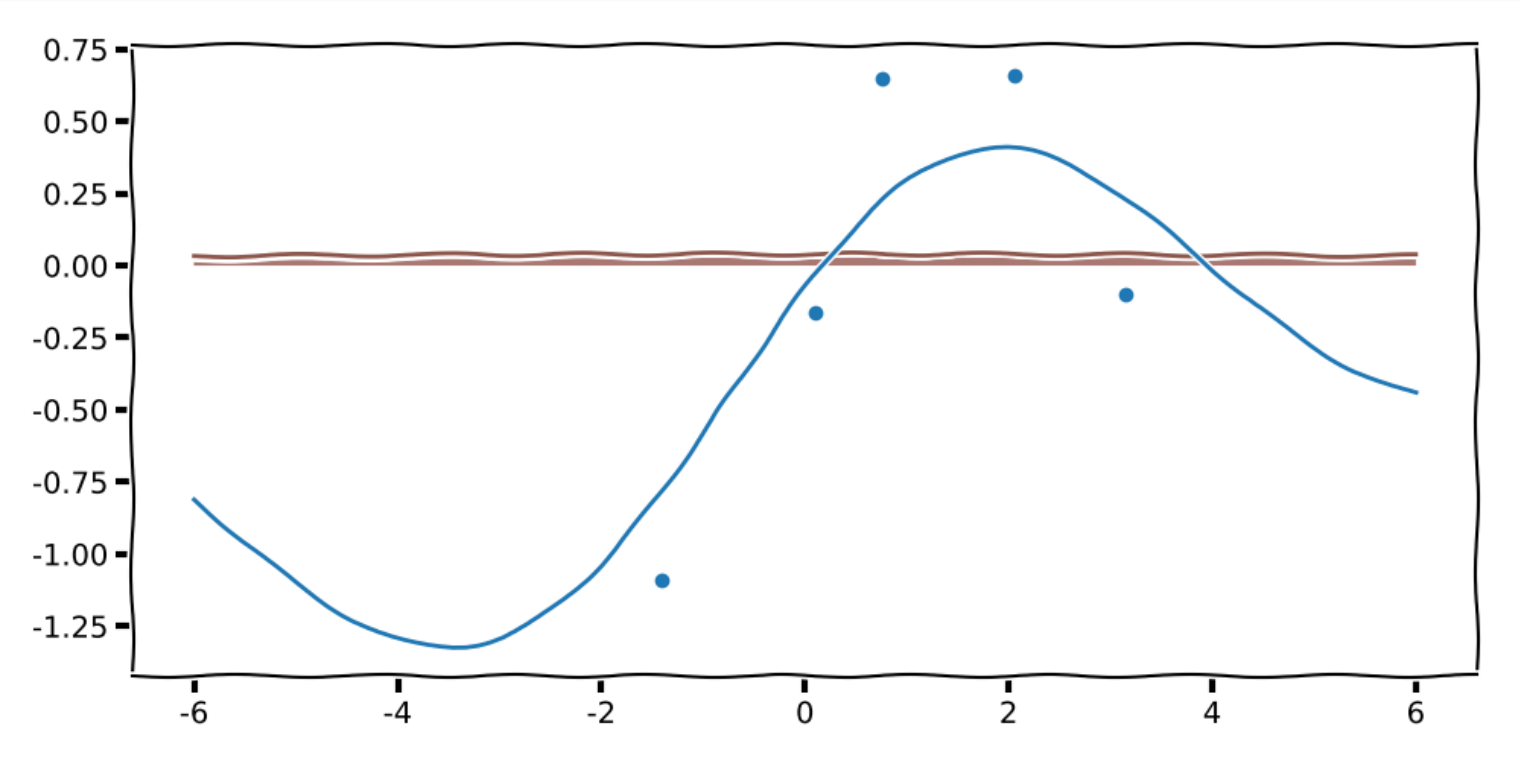
Kernel Regression



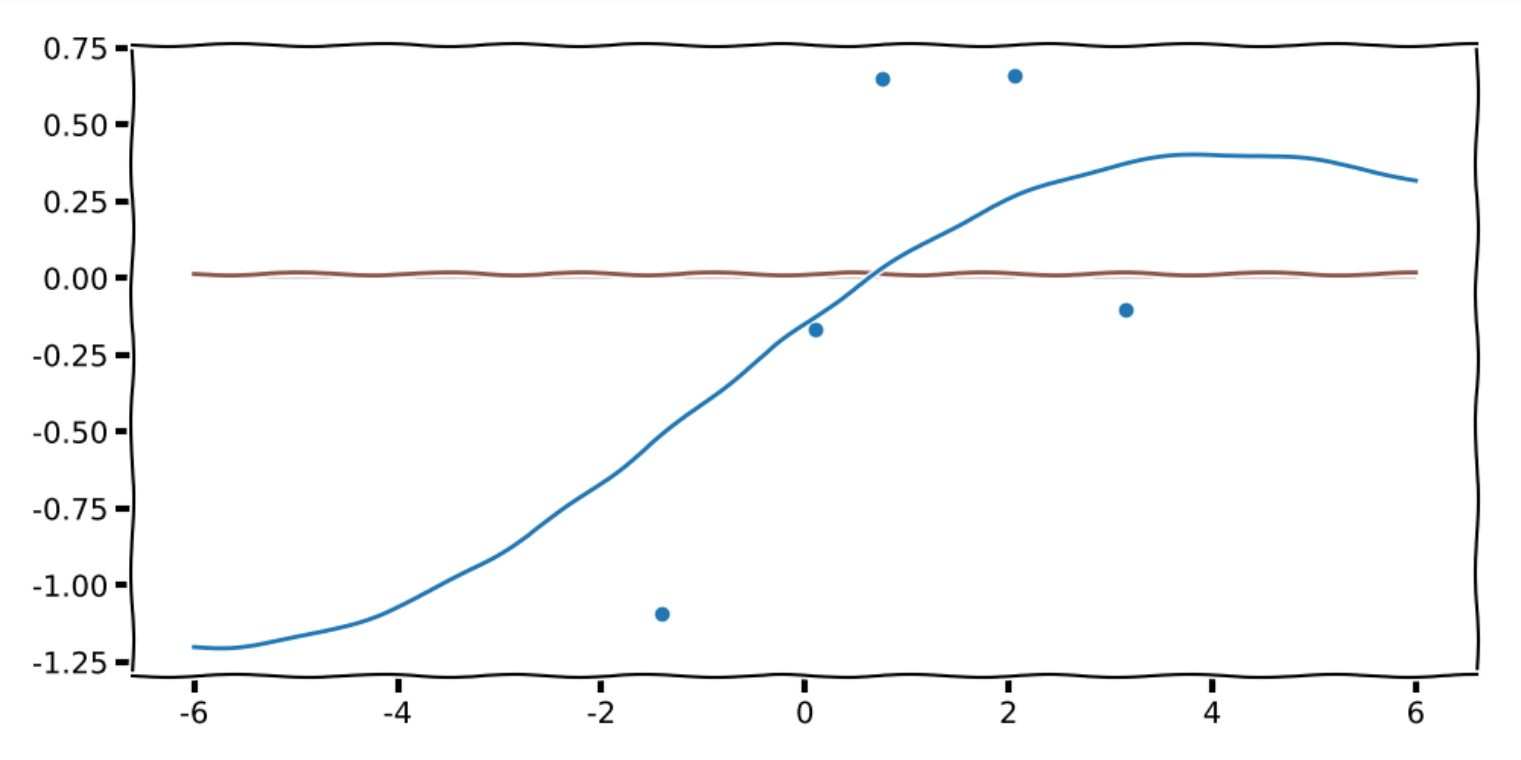
Kernel Regression



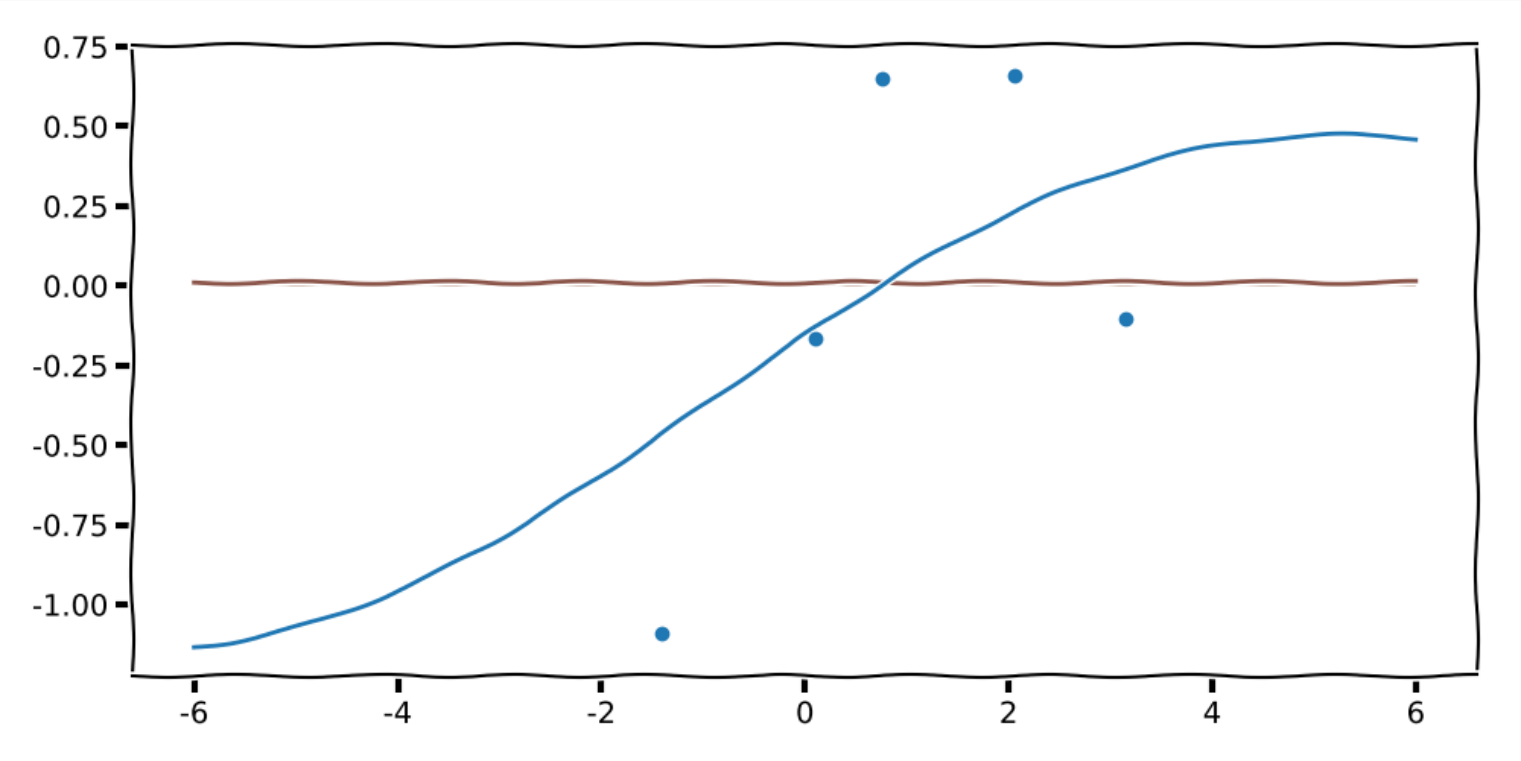
Kernel Regression



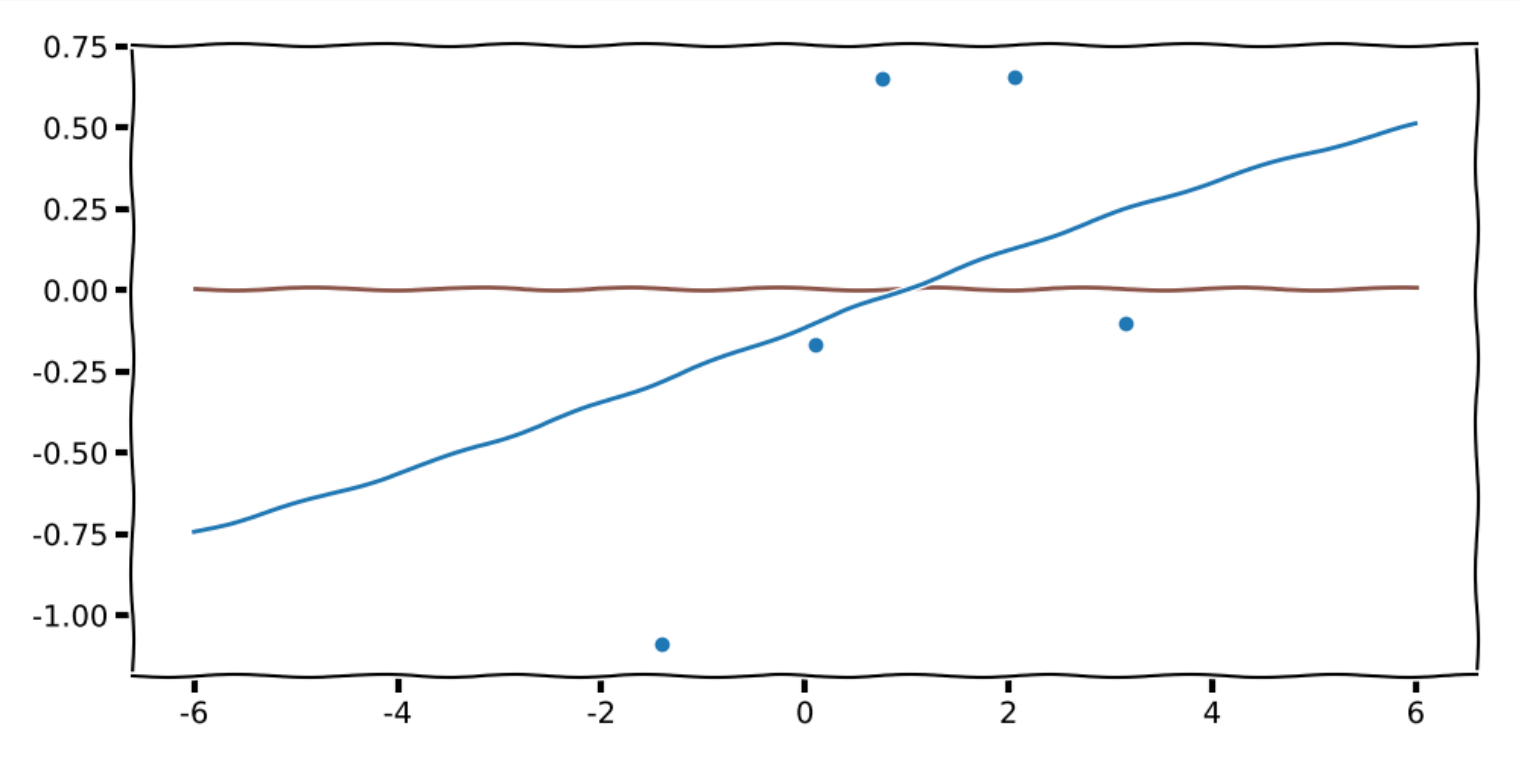
Kernel Regression



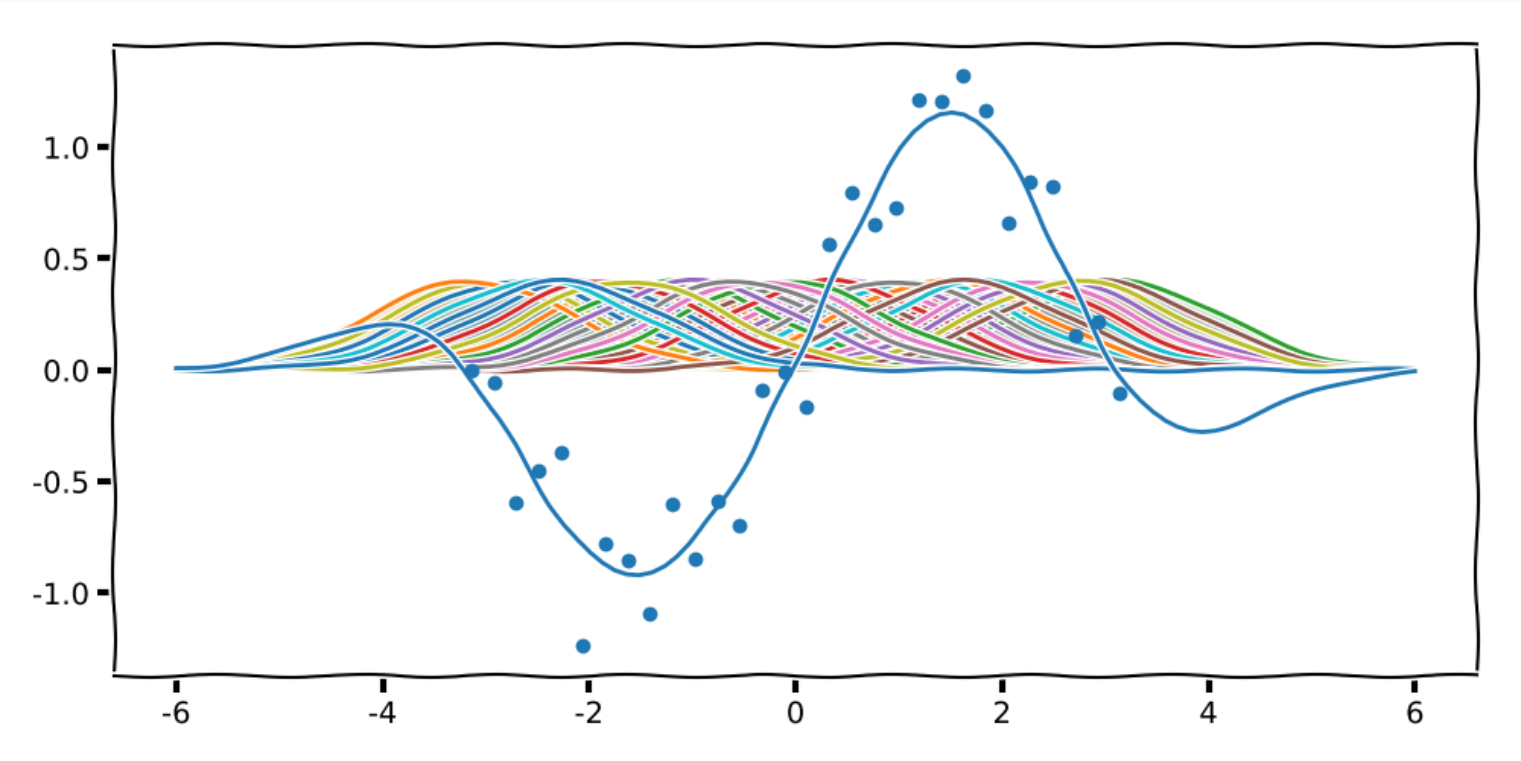
Kernel Regression



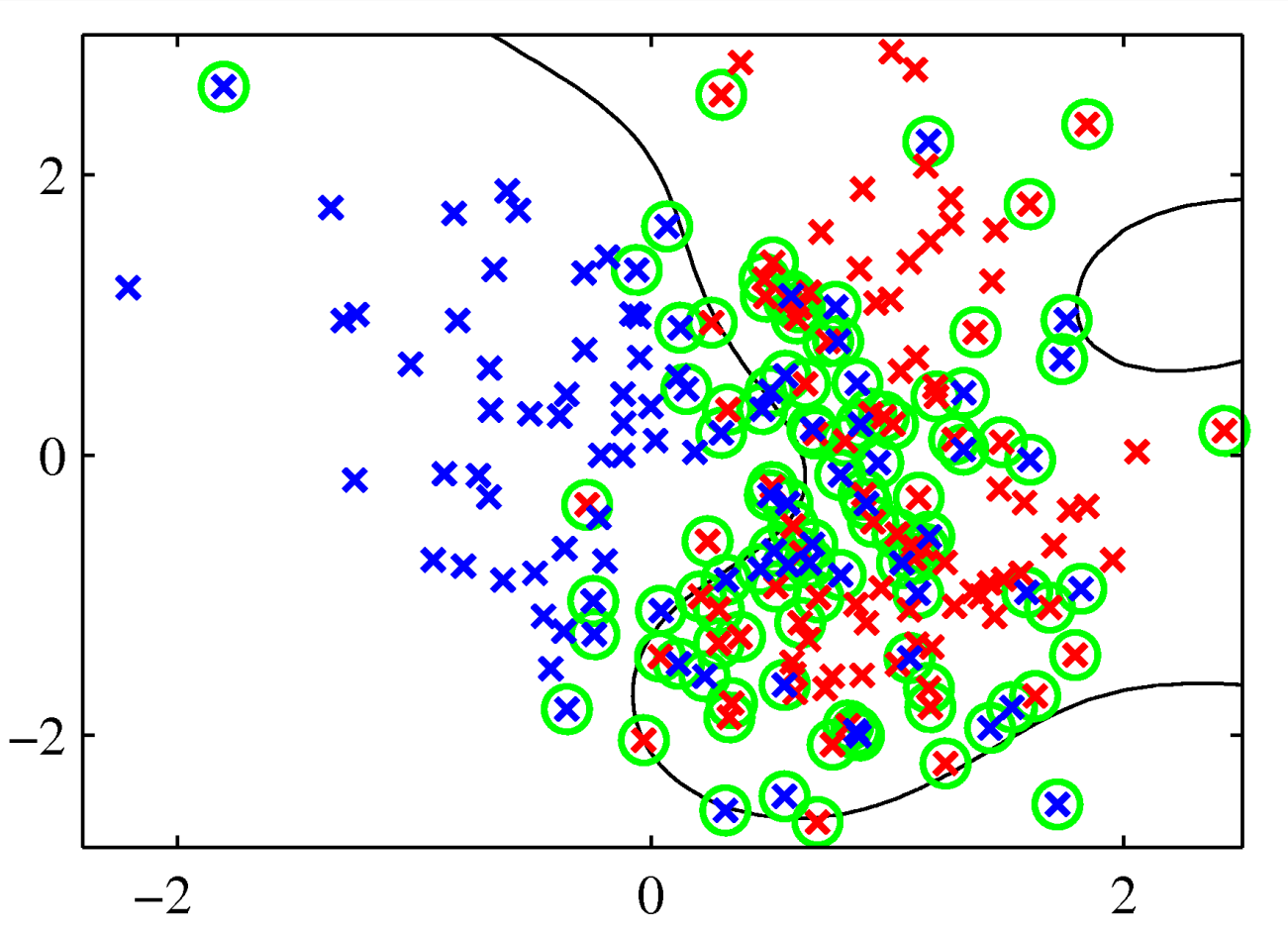
Kernel Regression



Kernel Regression



Support Vector Machines [1] Figure 7.4



Kernel Machines

- Allows us to
  - let the model complexity adapt to data
  - to put non vectorial data in a vector space
  - *problem is still linear*

Kernel Machines

- Allows us to
  - let the model complexity adapt to data
  - to put non vectorial data in a vector space
  - *problem is still linear*
- But
  - how to set kernel width
  - how to set noise assumption

Kernel Machines

- Allows us to
  - let the model complexity adapt to data
  - to put non vectorial data in a vector space
  - *problem is still linear*
- But
  - how to set kernel width
  - how to set noise assumption
- Next week we will learn these

Summary

Summary

- Repeat of the machine learning proceedure
  - assumption + data + compute $\rightarrow$ updated assumption
  - don't worry it will become clear eventually
- Non-parametrics
  - kernel regression
  - dual formulation
  - *the problem is still linear*

eof

References


Christopher M. Bishop.
Pattern Recognition and Machine Learning (Information Science and Statistics).
Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.