Lecture 5: Linear Regression by Maximum Likelihood, Maximum A Posteriori and Bayesian Learning
CS480/680 Intro to Machine Learning

2023-1-24

Pascal Poupart
David R. Cheriton School of Computer Science

UNIVERSITY OF WATERLOO

Noisy Linear Regression

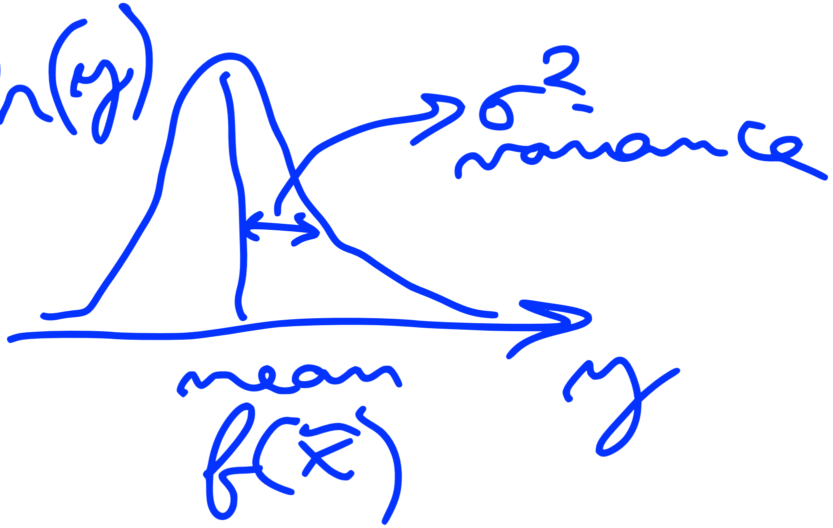
- Assume y is obtained from $\boldsymbol{x}$ by a deterministic function f that has been perturbed (i.e., noisy measurement)

$$y = f(\bar{\boldsymbol{x}}) + \epsilon$$

$$\boldsymbol{w}^T\bar{\boldsymbol{x}} \qquad N(0, \sigma^2)$$

- Gaussian noise:

dataset



$$\Pr(\boldsymbol{y}|\bar{\boldsymbol{X}}, \boldsymbol{w}, \sigma) = N\left(\boldsymbol{y}|\boldsymbol{w}^T\bar{\boldsymbol{X}}, \sigma^2\right)$$

$$= \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\left(y_n - \boldsymbol{w}^T\bar{\boldsymbol{x}}_n\right)^2}{2\sigma^2}}$$

normalization constant

mean

variance

UNIVERSITY OF WATERLOO

Maximum Likelihood

- Possible objective: find best $\boldsymbol{w}^*$ by maximizing the likelihood of the data

$$\boldsymbol{w}^* = argmax_{\boldsymbol{w}} \Pr(\boldsymbol{y}|\overline{\boldsymbol{X}}, \boldsymbol{w}, \sigma)$$

take log

$$= argmax_{\boldsymbol{w}} \prod_n e^{-\frac{\left(y_n - \boldsymbol{w}^T \overline{\boldsymbol{x}}_{\boldsymbol{n}}\right)^2}{2\sigma^2}}$$

$$= argmax_{\boldsymbol{w}} \sum_n -\frac{\left(y_n - \boldsymbol{w}^T \overline{\boldsymbol{x}}_{\boldsymbol{n}}\right)^2}{2\sigma^2}$$

$$= argmin_{\boldsymbol{w}} \sum_n \left(y_n - \boldsymbol{w}^T \overline{\boldsymbol{x}}_{\boldsymbol{n}}\right)^2$$

- We arrive at the original least square problem!

Maximum A Posteriori

- Alternative objective: find $\boldsymbol{w}^*$ with highest posterior probability
- Consider Gaussian prior: $\Pr(\boldsymbol{w}) = N(\mathbf{0}, \boldsymbol{\Sigma})$ (matrix)

$\begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$

proportional

- Posterior:

$$\Pr(\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y}) \propto \Pr(\boldsymbol{w}) \Pr(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{w})$$

$$= k e^{-\frac{\boldsymbol{w}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{w}}{2}} e^{-\frac{\sum_n \left(y_n - \boldsymbol{w}^T \boldsymbol{x}_n\right)^2}{2\sigma^2}}$$

Maximum A Posteriori

- Optimization:

$$\boldsymbol{w}^* = argmax_{\boldsymbol{w}} \Pr(\boldsymbol{w}|\bar{\boldsymbol{X}}, \boldsymbol{y})$$

Take log and drop constants

$$= argmax_{\boldsymbol{w}} \; -\sum_n \left(y_n - \boldsymbol{w}^T \bar{\boldsymbol{x}}_n\right)^2 - \boldsymbol{w}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{w}$$

$$= argmin_{\boldsymbol{w}} \sum_n \left(y_n - \boldsymbol{w}^T \bar{\boldsymbol{x}}_n\right)^2 + \boldsymbol{w}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{w}$$

$$\Sigma^{-1} = \begin{pmatrix} \lambda & & 0 \\ & \lambda & \\ 0 & & \ddots \\ & & & \lambda \end{pmatrix}$$

Let $\boldsymbol{\Sigma}^{-1} = \lambda \boldsymbol{I}$ then

$$\Sigma = \begin{pmatrix} 1/\lambda & & 0 \\ & 1/\lambda & \\ 0 & & \ddots \\ & & & 1/\lambda \end{pmatrix}$$

$$= argmin_{\boldsymbol{w}} \sum_n \left(y_n - \boldsymbol{w}^T \bar{\boldsymbol{x}}_n\right)^2 + \lambda \left|\left|\boldsymbol{w}\right|\right|_2^2$$

- We arrive at the original **regularized** least square problem!

Expected Squared Loss

- Even though we use a statistical framework, it is interesting to evaluate the expected squared loss

$$E[L] = \int_{\boldsymbol{x},y} \Pr(\boldsymbol{x}, y)\left(y - \boldsymbol{w}^T \bar{\boldsymbol{x}}\right)^2 d\boldsymbol{x}dy$$

$$= \int_{\boldsymbol{x},y} \Pr(\boldsymbol{x}, y)\left(y - f(\boldsymbol{x}) + f(\boldsymbol{x}) - \boldsymbol{w}^T \bar{\boldsymbol{x}}\right)^2 d\boldsymbol{x}dy$$

$$= \int_{\boldsymbol{x},y} \Pr(\boldsymbol{x}, y)\left[\left(y - f(\boldsymbol{x})\right)^2 + 2\underbrace{\left(y - f(\boldsymbol{x})\right)}_{\text{Expectation with respect to } y \text{ is } 0}\left(f(\boldsymbol{x}) - \boldsymbol{w}^T \bar{\boldsymbol{x}}\right) + \left(f(\boldsymbol{x}) - \boldsymbol{w}^T \bar{\boldsymbol{x}}\right)^2\right] d\boldsymbol{x}dy$$

$$E[L] = \underbrace{\int_{\boldsymbol{x},y} \Pr(\boldsymbol{x}, y)\left(y - f(\boldsymbol{x})\right)^2 d\boldsymbol{x}dy}_{\text{noise (constant)}} + \underbrace{\int_{\boldsymbol{x}} \Pr(\boldsymbol{x})\left(f(\boldsymbol{x}) - \boldsymbol{w}^T \bar{\boldsymbol{x}}\right)^2 d\boldsymbol{x}}_{\text{error (depends on } \boldsymbol{w})}$$

Expected Squared Loss

- Let's focus on the error part, which depends on $\boldsymbol{w}$

$$E_{\boldsymbol{x}}[(f(\boldsymbol{x}) - \boldsymbol{w}^T\overline{\boldsymbol{x}})^2] = \int_{\boldsymbol{x}} \Pr(\boldsymbol{x}) \left(f(\boldsymbol{x}) - \boldsymbol{w}^T\overline{\boldsymbol{x}}\right)^2 d\boldsymbol{x}$$

- But the choice of $\boldsymbol{w}$ depends on the dataset S
- Instead consider expectation with respect to S

$$E_S\left[(f(\boldsymbol{x}) - \boldsymbol{w}_S^T\overline{\boldsymbol{x}})^2\right]$$

where $\boldsymbol{w}_S$ is the weight vector obtained based on S

Bias-Variance Decomposition

- Decompose squared loss

$$E_S[(f(\boldsymbol{x}) - \boldsymbol{w}_S^T\overline{\boldsymbol{x}})^2]$$

$$= E_S[f(\boldsymbol{x}) - E_S[\boldsymbol{w}_S^T\overline{\boldsymbol{x}}] + E_S[\boldsymbol{w}_S^T\overline{\boldsymbol{x}}] - \boldsymbol{w}_S^T\overline{\boldsymbol{x}}]^2$$

$$= E_S\left[(f(\boldsymbol{x}) - E_S[\boldsymbol{w}_S^T\overline{\boldsymbol{x}}])^2 + 2(f(\boldsymbol{x}) - E_S[\boldsymbol{w}_S^T\overline{\boldsymbol{x}}])\underbrace{(E_S[\boldsymbol{w}_S^T\overline{\boldsymbol{x}}] - \boldsymbol{w}_S^T\overline{\boldsymbol{x}})}_{\text{Expectation is 0}} + (E_S[\boldsymbol{w}_S^T\overline{\boldsymbol{x}}] - \boldsymbol{w}_S^T\overline{\boldsymbol{x}})^2\right]$$

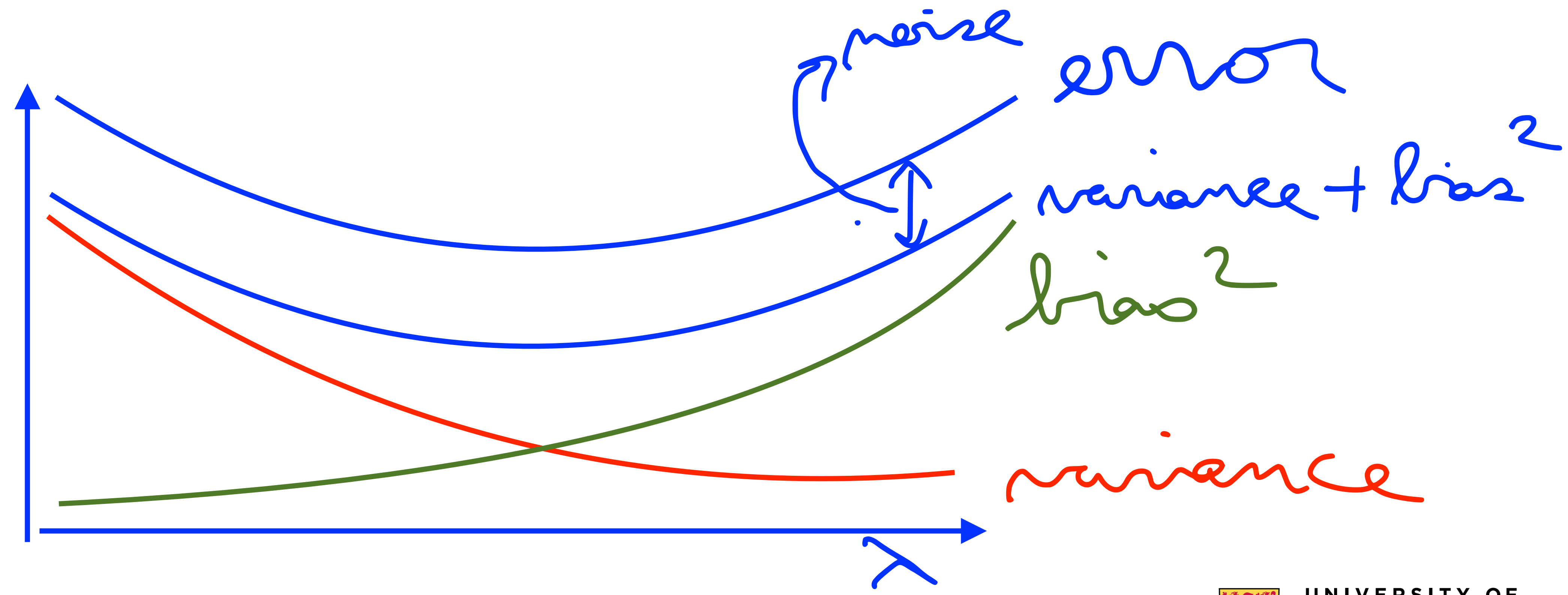
$$= \underbrace{(f(\boldsymbol{x}) - E_S[\boldsymbol{w}_S^T\overline{\boldsymbol{x}}])^2}_{\text{bias}^2} + \underbrace{E_S\left[(E_S[\boldsymbol{w}_S^T\overline{\boldsymbol{x}}] - \boldsymbol{w}_S^T\overline{\boldsymbol{x}})^2\right]}_{\text{variance}}$$

Bias-Variance Decomposition

- Hence:

$$E[loss] = (bias)^2 + variance + noise$$

- Picture:



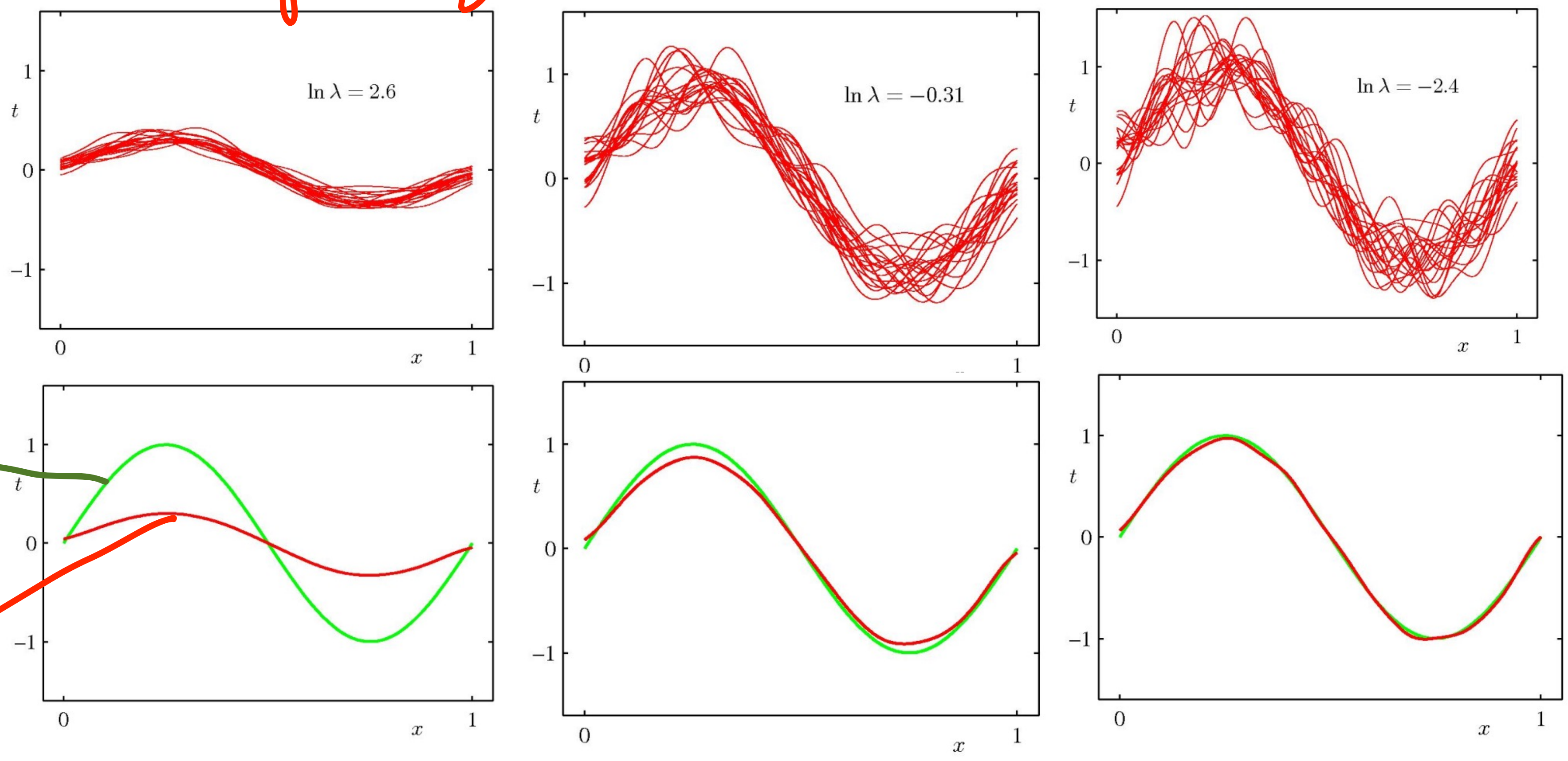
UNIVERSITY OF WATERLOO

Bias-Variance Decomposition

- Example

underfitting

overfitting



variance

bias

f(x)
true fn

estimate
$E(w_S^T x)$

Bayesian Linear Regression

- We don't know if $\boldsymbol{w}^*$ is the true underlying $\boldsymbol{w}$
- Instead of making predictions according to $\boldsymbol{w}^*$, compute the weighted average prediction according to $\Pr(\boldsymbol{w}|\overline{\boldsymbol{X}}, \boldsymbol{y})$

$$\Pr(\boldsymbol{w}|\overline{\boldsymbol{X}}, \boldsymbol{y}) = k e^{-\frac{\boldsymbol{w}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{w}}{2}} e^{-\frac{\sum_n \left(y_n - \boldsymbol{w}^T \overline{\boldsymbol{x}}_n\right)^2}{2\sigma^2}}$$

$$= k e^{-\frac{1}{2}(\boldsymbol{w}-\overline{\boldsymbol{w}})^T \boldsymbol{A} (\boldsymbol{w}-\overline{\boldsymbol{w}})} = N(\overline{\boldsymbol{w}}, \boldsymbol{A}^{-1})$$

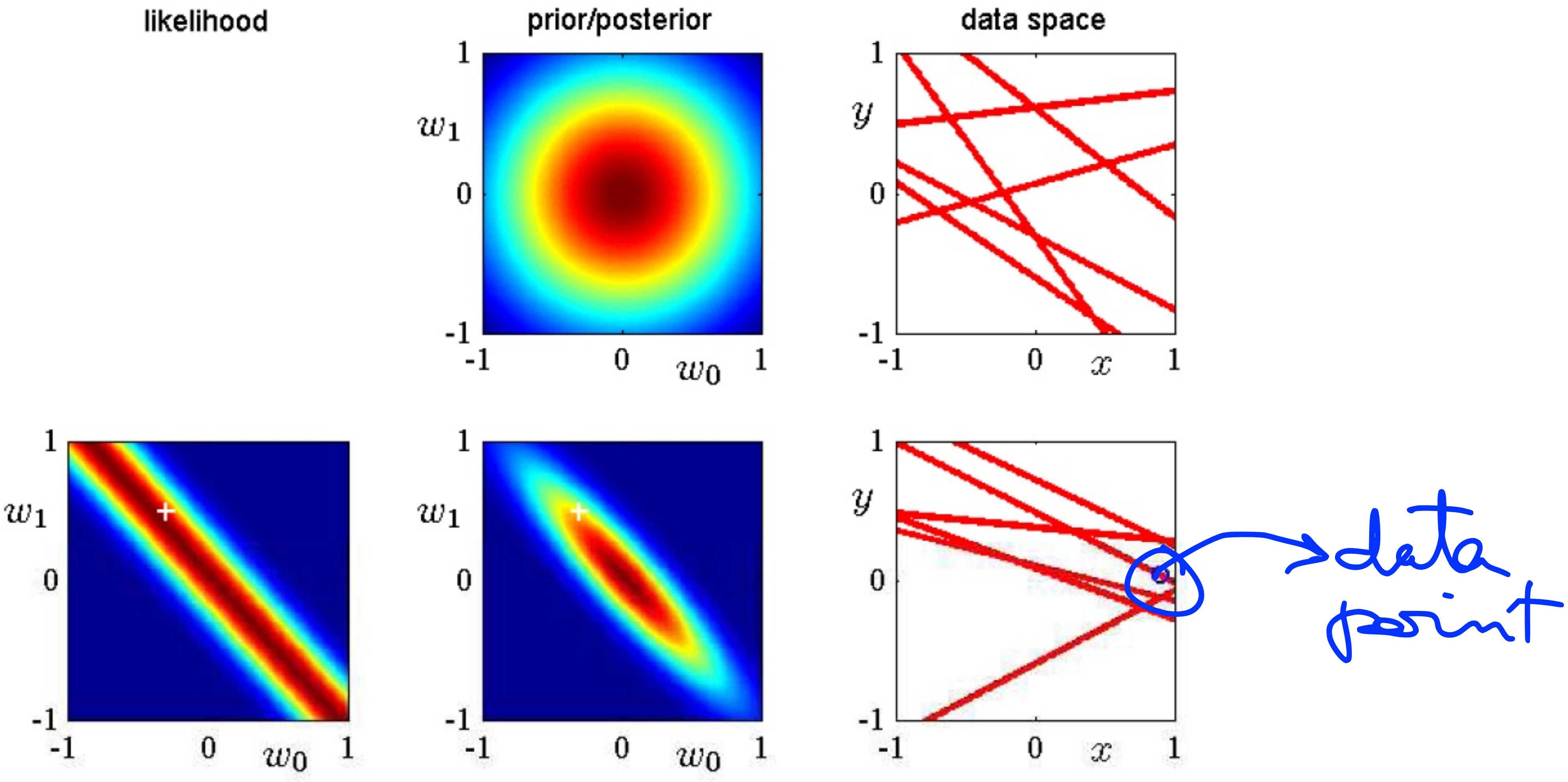
mean

covariance

where $\overline{\boldsymbol{w}} = \sigma^{-2} \boldsymbol{A}^{-1} \overline{\boldsymbol{X}} \boldsymbol{y}$

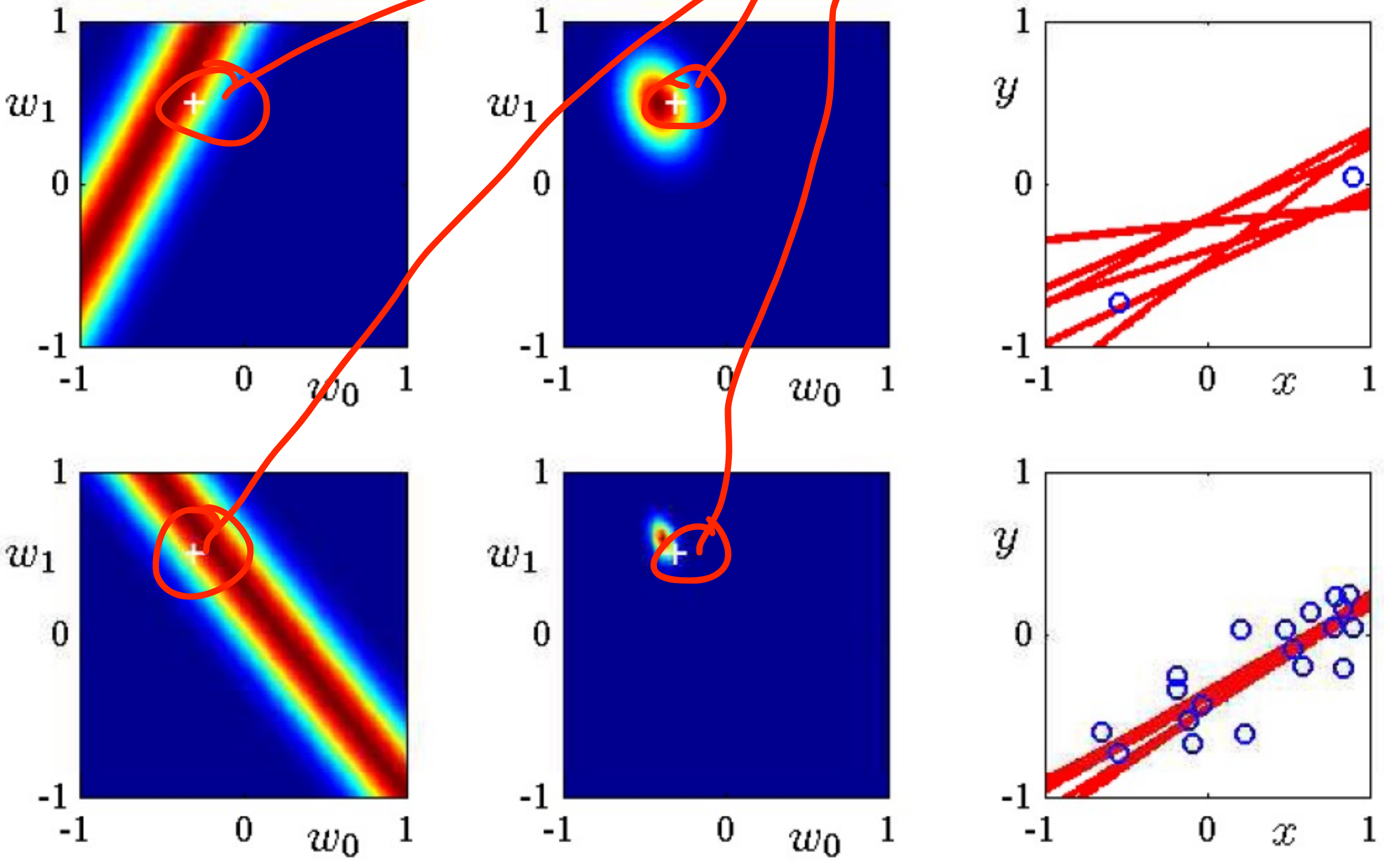
$$\boldsymbol{A} = \sigma^{-2} \overline{\boldsymbol{X}}\,\overline{\boldsymbol{X}}^T + \boldsymbol{\Sigma}^{-1}$$

Bayesian Learning



Bayesian Learning

true weights



Bayesian Prediction

- Let $\boldsymbol{x}_*$ be the input for which we want a prediction and y_* be the corresponding prediction

query point

$$\Pr(y_*|\overline{\boldsymbol{x}}_*, \overline{\boldsymbol{X}}, \boldsymbol{y}) = \int_{\boldsymbol{w}} \Pr(y_*|\overline{\boldsymbol{x}}_*, \boldsymbol{w}) \Pr(\boldsymbol{w}|\overline{\boldsymbol{X}}, y)\, d\boldsymbol{w}$$

$$= k \int_{\boldsymbol{w}} e^{-\frac{\left(y_* - \overline{\boldsymbol{x}}_*^T \boldsymbol{w}\right)^2}{2\sigma^2}} e^{-\frac{1}{2}(\boldsymbol{w}-\overline{\boldsymbol{w}})^T \boldsymbol{A}(\boldsymbol{w}-\overline{\boldsymbol{w}})}\, d\boldsymbol{w}$$

$$= N(\underbrace{\sigma^{-2}\overline{\boldsymbol{x}}_*^T \boldsymbol{A}^{-1} \overline{\boldsymbol{X}} \boldsymbol{y}}_{\text{mean}}, \underbrace{\sigma^2 + \overline{\boldsymbol{x}}_*^T \boldsymbol{A}^{-1} \overline{\boldsymbol{x}}_*}_{\text{variance}})$$