

Introduction to Machine Learning

Brown University CSCI 1950-F, Spring 2011
Prof. Erik Sudderth

Lecture 8: Linear Regression

Many figures courtesy Kevin Murphy's textbook,
Machine Learning: A Probabilistic Perspective

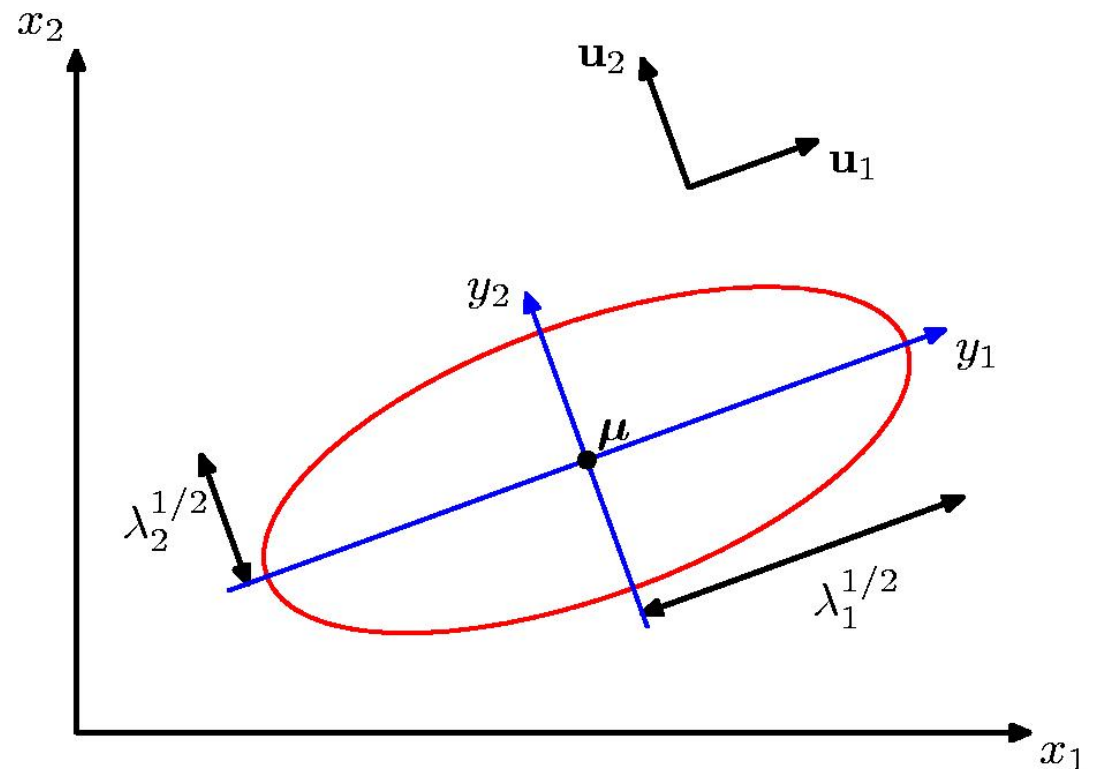
Geometry of the Multivariate Gaussian

$$\Delta^2 = (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})$$

$$\boldsymbol{\Sigma}^{-1} = \sum_{i=1}^D \frac{1}{\lambda_i} \mathbf{u}_i \mathbf{u}_i^T$$

$$\Delta^2 = \sum_{i=1}^D \frac{y_i^2}{\lambda_i}$$

$$y_i = \mathbf{u}_i^T (\mathbf{x} - \boldsymbol{\mu})$$



*Slides adapted from Bishop's "Pattern Recognition and Machine Learning"
Notation differs from Murphy's "Machine Learning: A Probabilistic Perspective"*

Partitioned Gaussian Distributions

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{pmatrix} \quad \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{pmatrix} \quad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{pmatrix}$$

$$\boldsymbol{\Lambda} \equiv \boldsymbol{\Sigma}^{-1} \quad \boldsymbol{\Lambda} = \begin{pmatrix} \boldsymbol{\Lambda}_{aa} & \boldsymbol{\Lambda}_{ab} \\ \boldsymbol{\Lambda}_{ba} & \boldsymbol{\Lambda}_{bb} \end{pmatrix}$$

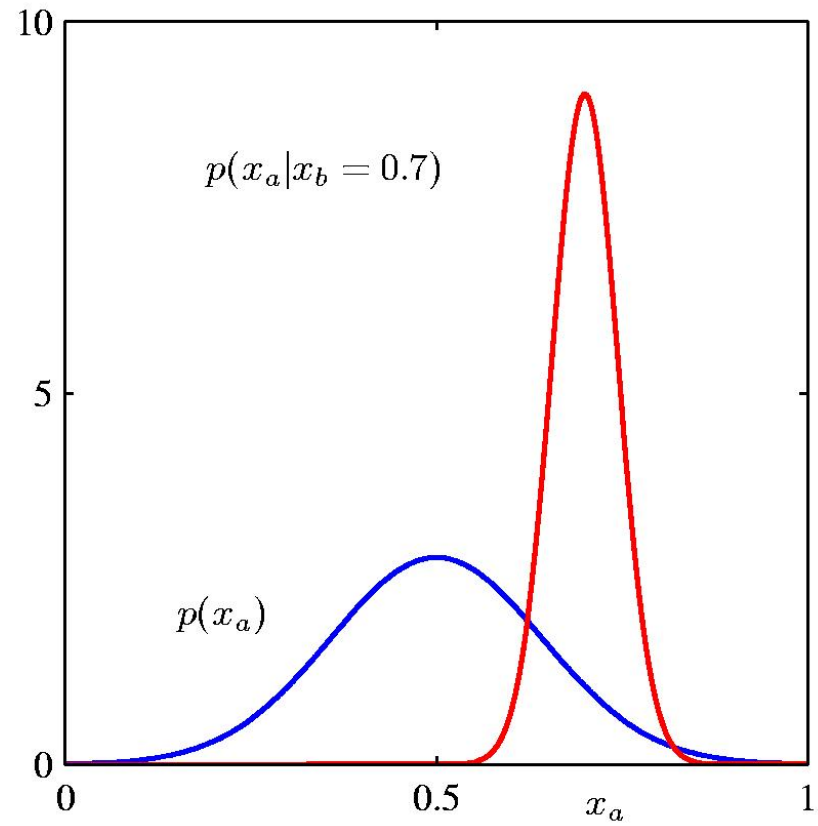
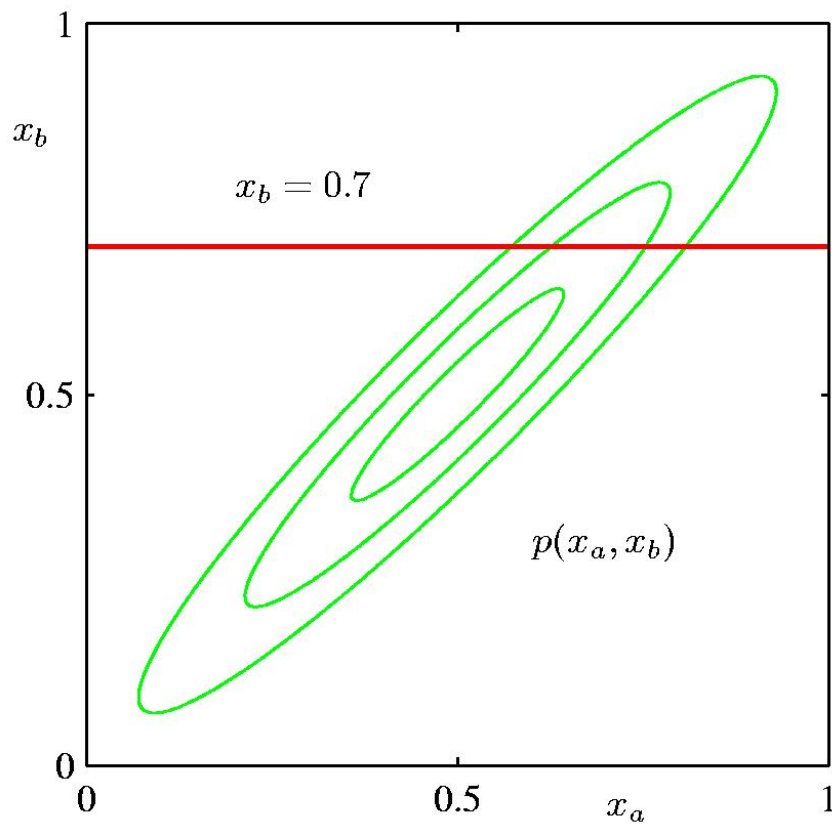
Partitioned Conditionals and Marginals

$$p(\mathbf{x}_a|\mathbf{x}_b) = \mathcal{N}(\mathbf{x}_a|\boldsymbol{\mu}_{a|b}, \boldsymbol{\Sigma}_{a|b})$$

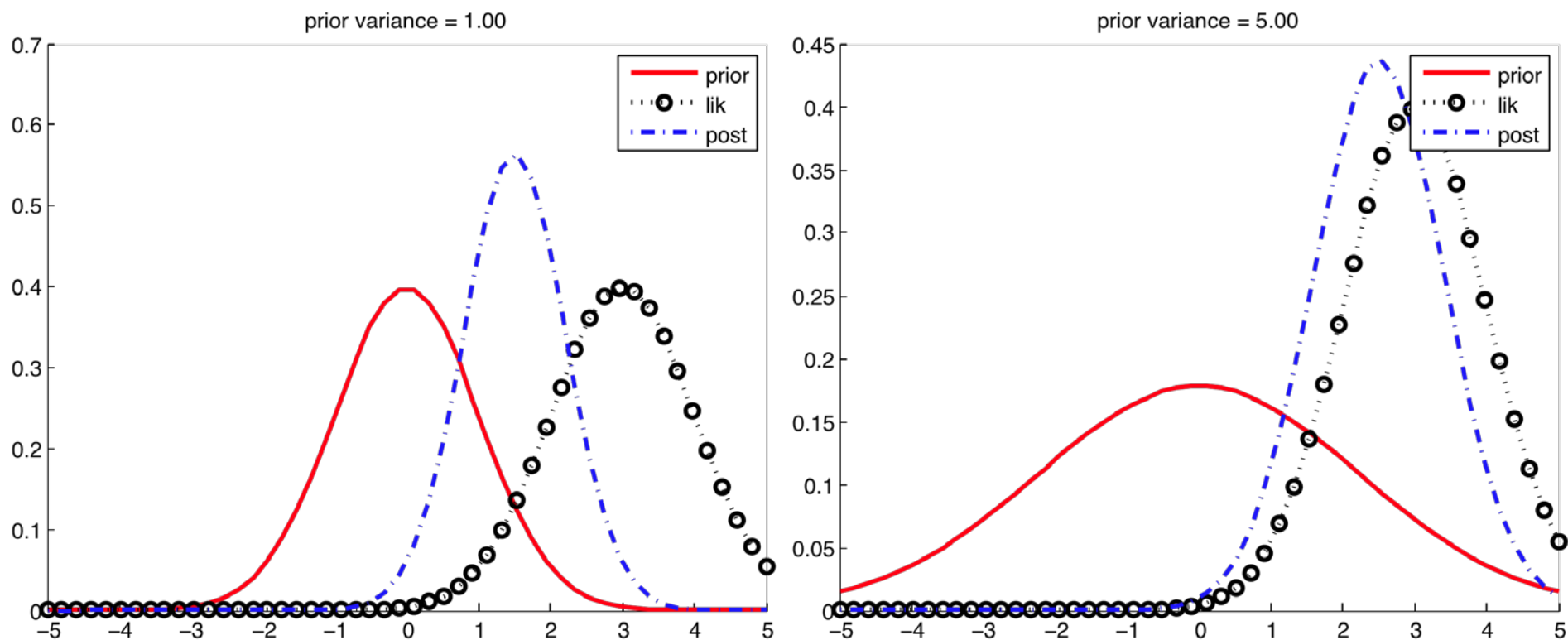
$$\begin{aligned}\boldsymbol{\Sigma}_{a|b} &= \boldsymbol{\Lambda}_{aa}^{-1} = \boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba} \\ \boldsymbol{\mu}_{a|b} &= \boldsymbol{\Sigma}_{a|b} \{ \boldsymbol{\Lambda}_{aa}\boldsymbol{\mu}_a - \boldsymbol{\Lambda}_{ab}(\mathbf{x}_b - \boldsymbol{\mu}_b) \} \\ &= \boldsymbol{\mu}_a - \boldsymbol{\Lambda}_{aa}^{-1}\boldsymbol{\Lambda}_{ab}(\mathbf{x}_b - \boldsymbol{\mu}_b) \\ &= \boldsymbol{\mu}_a + \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}(\mathbf{x}_b - \boldsymbol{\mu}_b)\end{aligned}$$

$$\begin{aligned}p(\mathbf{x}_a) &= \int p(\mathbf{x}_a, \mathbf{x}_b) \mathrm{d}\mathbf{x}_b \\ &= \mathcal{N}(\mathbf{x}_a|\boldsymbol{\mu}_a, \boldsymbol{\Sigma}_{aa})\end{aligned}$$

Partitioned Conditionals and Marginals



Impact of Prior Covariance



$$p(x|y) = \mathcal{N}(x|\mu_N, \lambda_N^{-1})$$

$$p(x) = \mathcal{N}(x|\mu_0, \lambda_0^{-1})$$

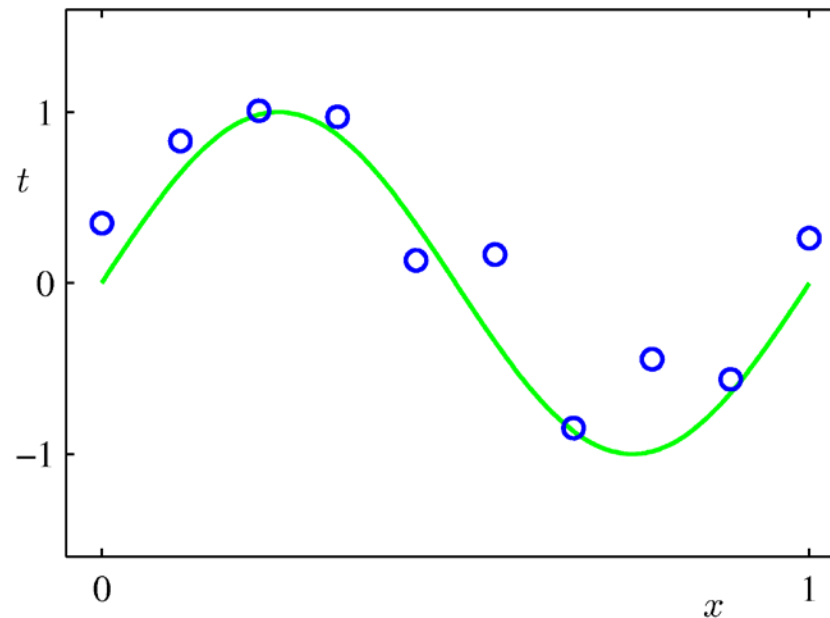
$$\lambda_N = \lambda_0 + N\lambda_y$$

$$p(y_i|x) = \mathcal{N}(y_i|x, \lambda_y^{-1})$$

$$\mu_N = \frac{N\lambda_y \bar{y} + \lambda_0 \mu_0}{\lambda_N} = \frac{N\lambda_y}{N\lambda_y + \lambda_0} \bar{y} + \frac{\lambda_0}{N\lambda_y + \lambda_0} \mu_0$$

Linear Basis Function Models (1)

- Example: Polynomial Curve Fitting



$$y(x, \mathbf{w}) = w_0 + w_1x + w_2x^2 + \dots + w_Mx^M = \sum_{j=0}^M w_jx^j$$

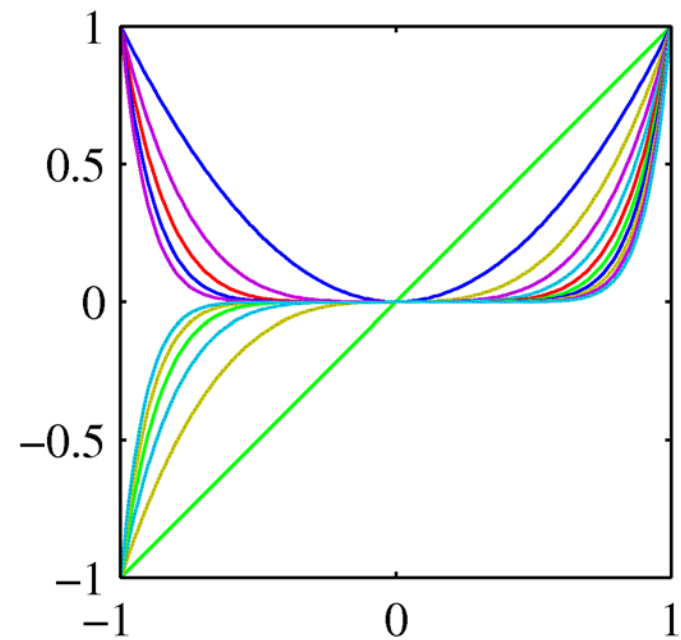
*Slides adapted from Bishop's "Pattern Recognition and Machine Learning"
Notation differs from Murphy's "Machine Learning: A Probabilistic Perspective"*

Linear Basis Function Models (2)

- Polynomial basis functions:

$$\phi_j(x) = x^j.$$

- These are global; a small change in x affect all basis functions.

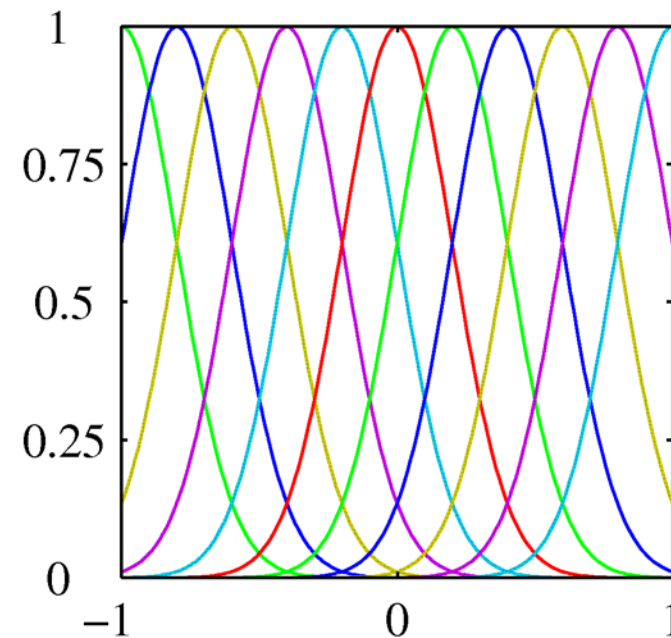


Linear Basis Function Models (3)

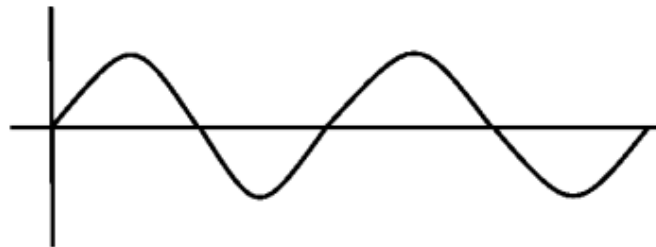
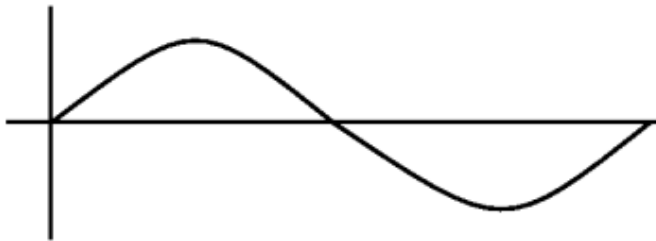
- Gaussian basis functions:

$$\phi_j(x) = \exp \left\{ -\frac{(x - \mu_j)^2}{2s^2} \right\}$$

- These are local; a small change in x only affect nearby basis functions. Parameters control location and scale (width).

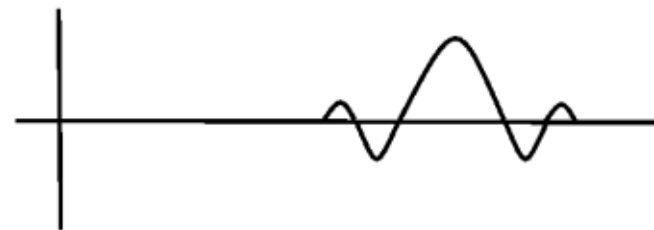
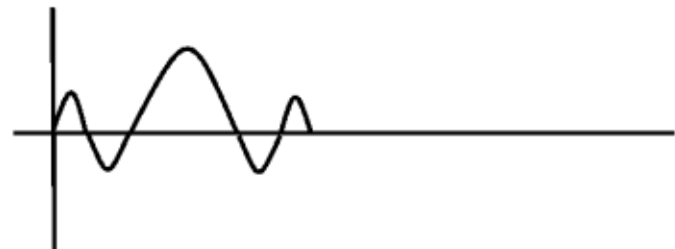
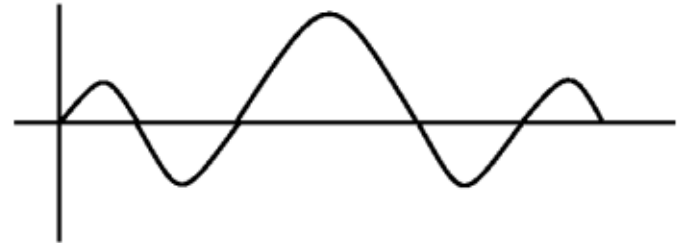


Linear Basis Function Models (4)



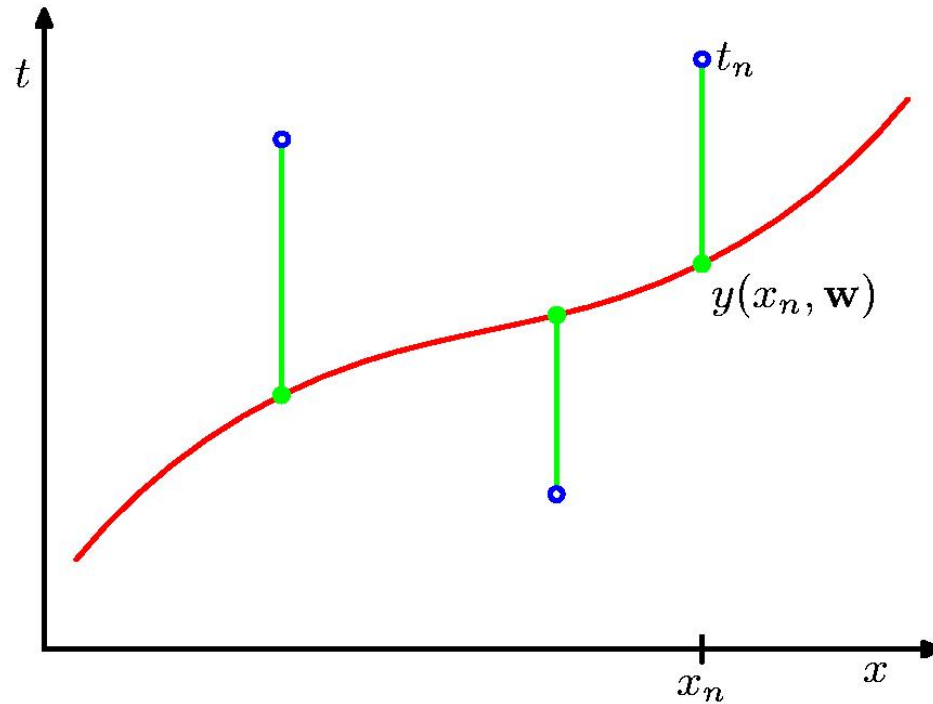
•
•
•

Fourier Basis



Wavelet Basis

Sum-of-Squares Error Function



$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{y(x_n, \mathbf{w}) - t_n\}^2$$

Geometry of Least Squares

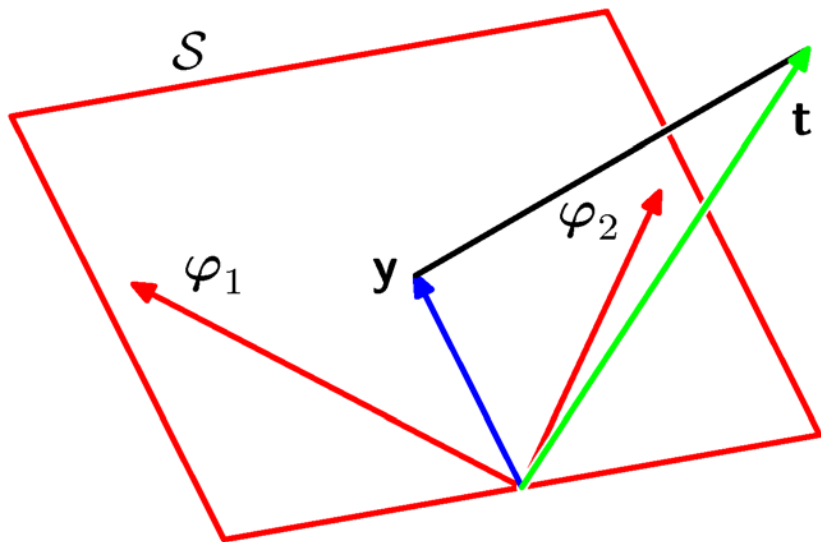
- Consider

$$\mathbf{y} = \Phi \mathbf{w}_{\text{ML}} = [\varphi_1, \dots, \varphi_M] \mathbf{w}_{\text{ML}}.$$

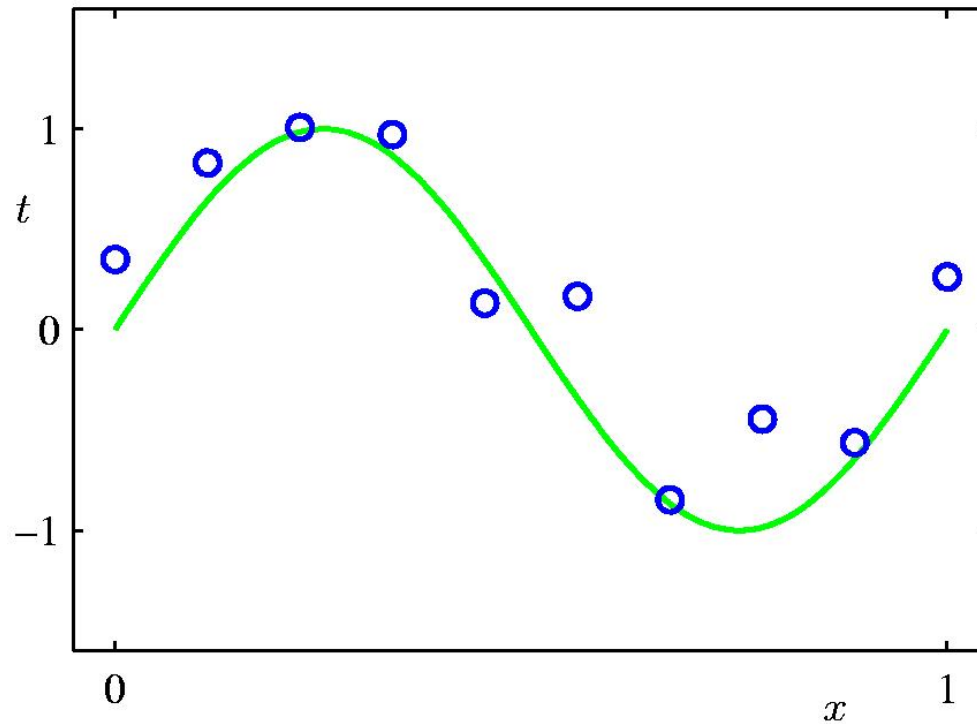
$$\mathbf{y} \in \mathcal{S} \subseteq \mathcal{T} \quad \mathbf{t} \in \mathcal{T}$$

↑ ↑
N-dimensional
M-dimensional

- $\mathcal{S}$ is spanned by $\varphi_1, \dots, \varphi_M$.
- $\mathbf{w}_{\text{ML}}$ minimizes the distance between $\mathbf{t}$ and its orthogonal projection on $\mathcal{S}$, i.e. $\mathbf{y}$.

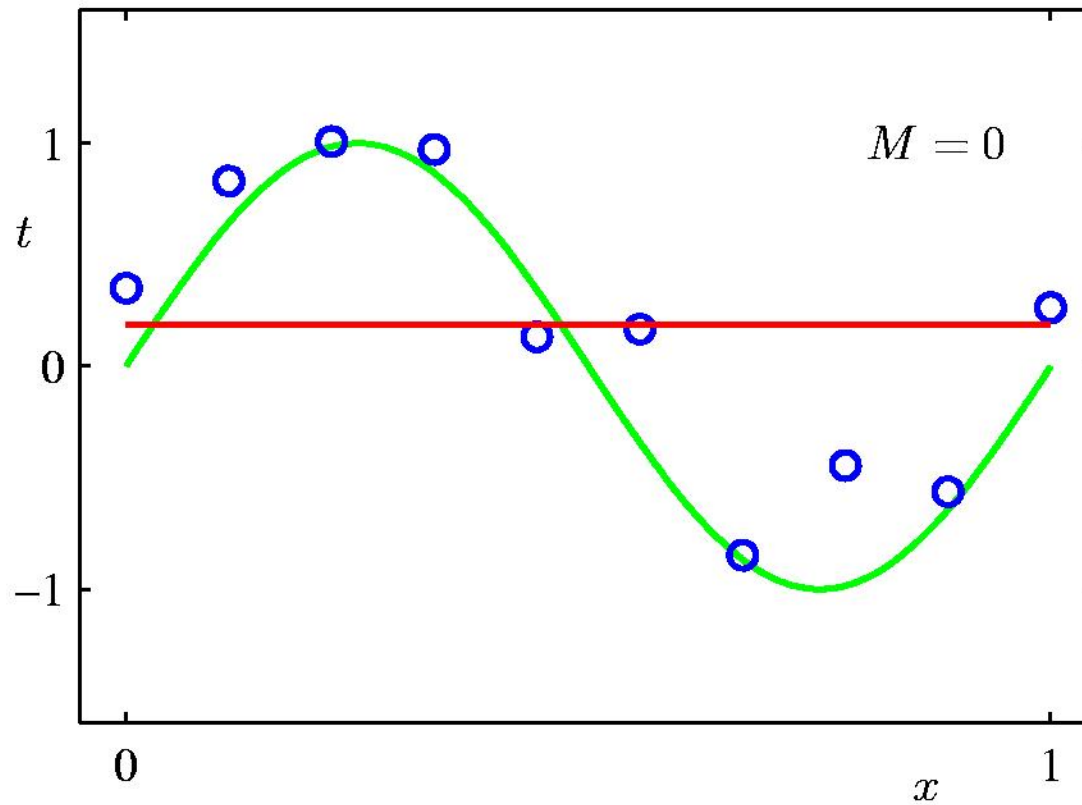


Polynomial Curve Fitting

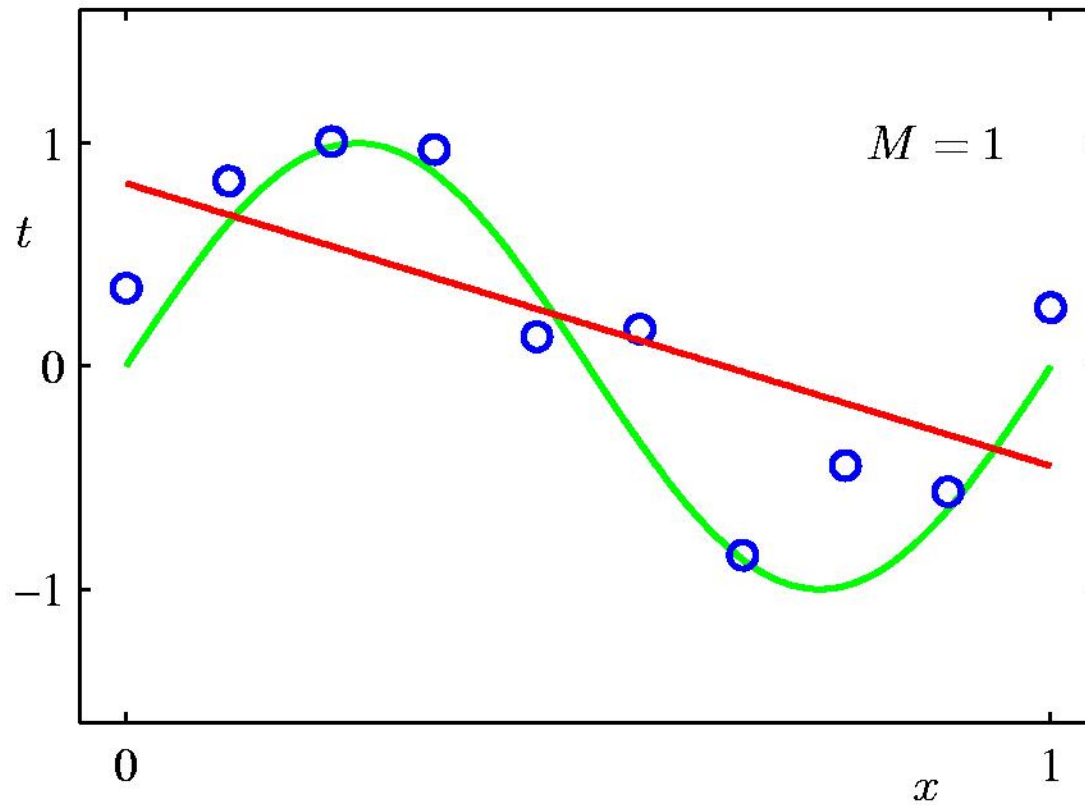


$$y(x, \mathbf{w}) = w_0 + w_1x + w_2x^2 + \dots + w_Mx^M = \sum_{j=0}^M w_jx^j$$

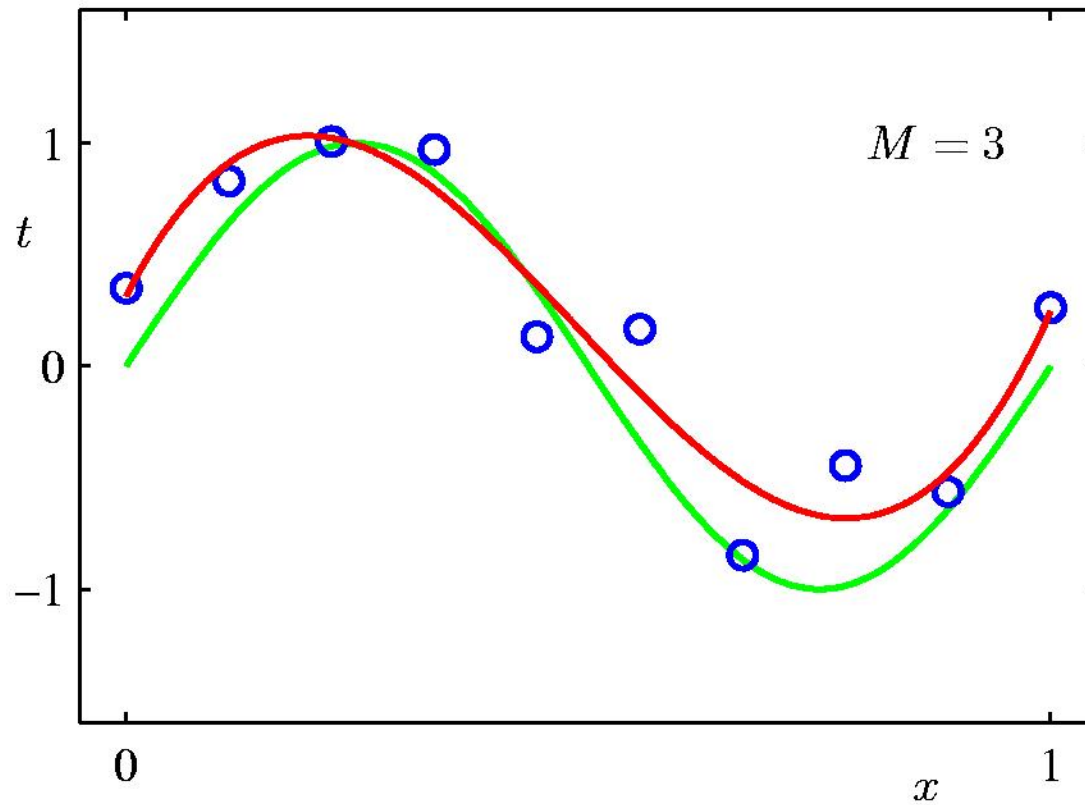
0th Order Polynomial



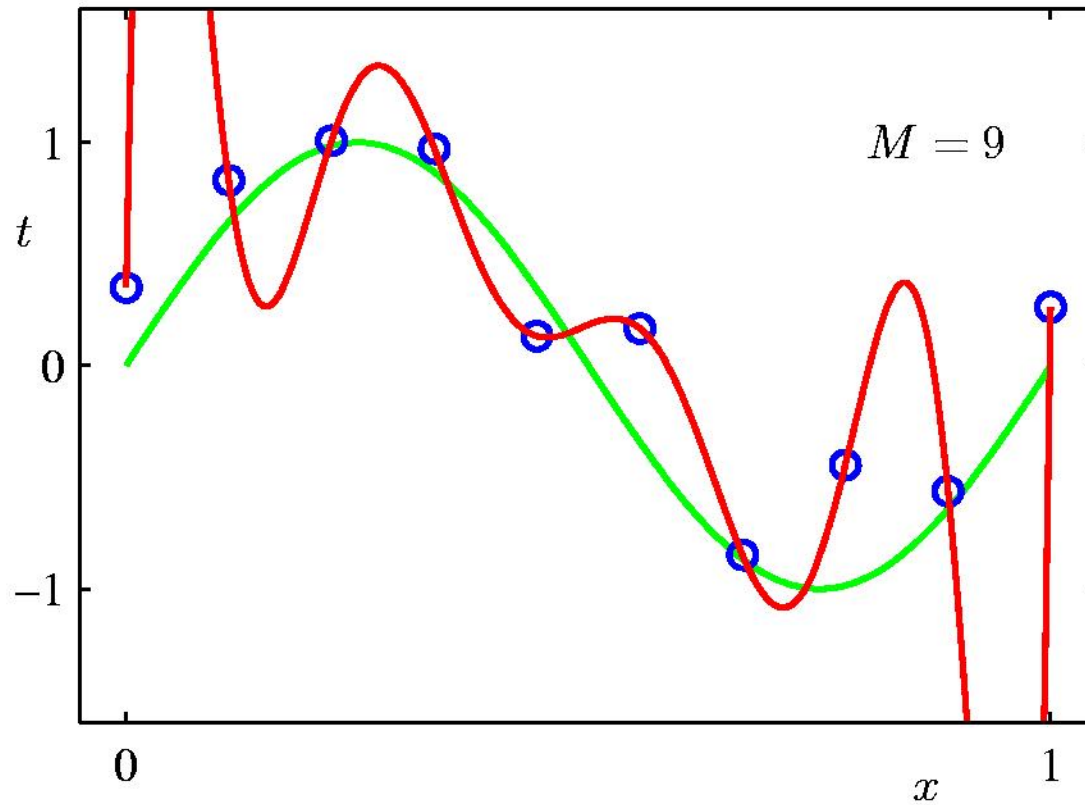
1st Order Polynomial



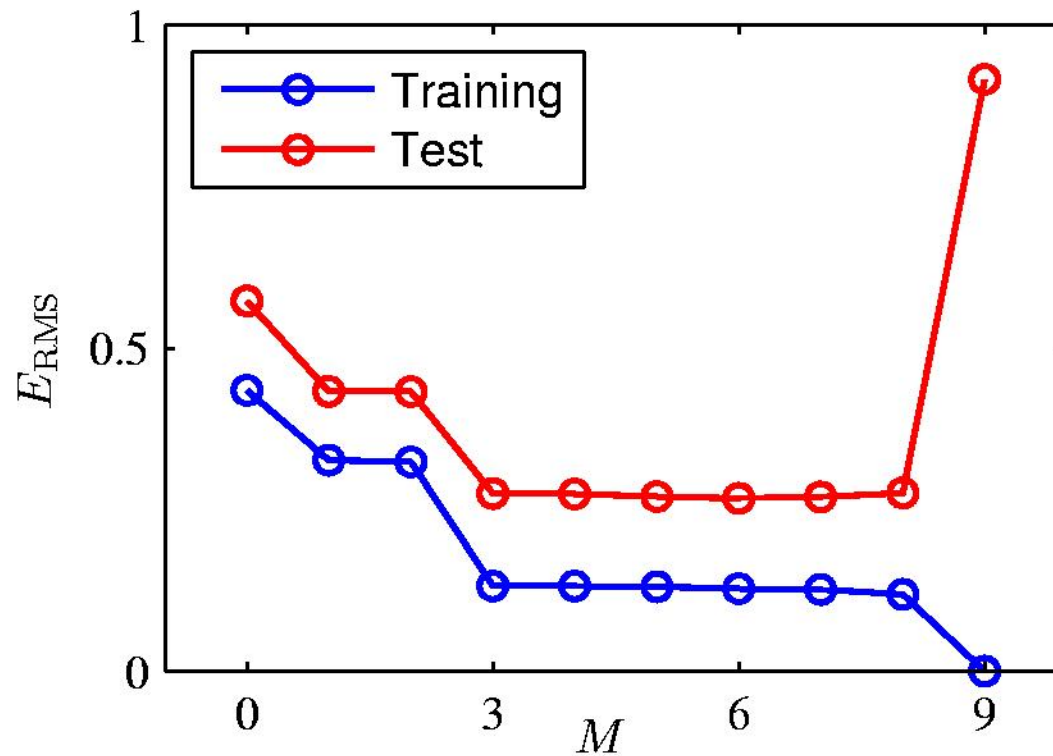
3rd Order Polynomial



9th Order Polynomial



Over-fitting



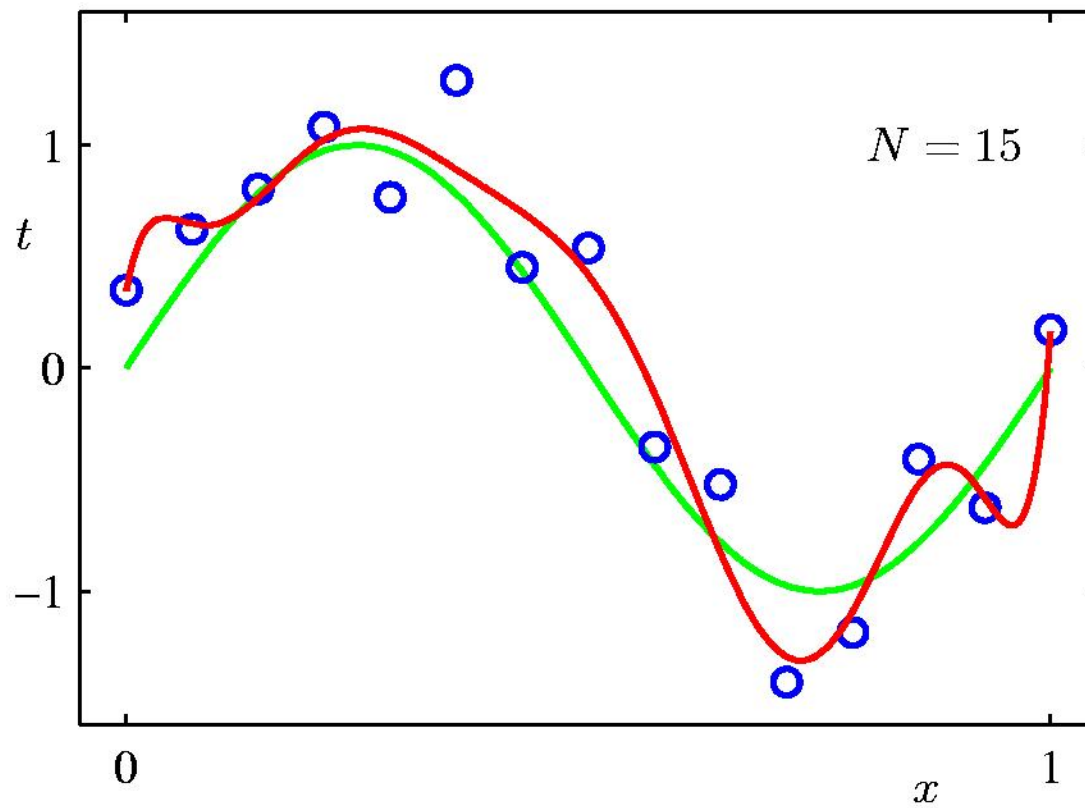
Root-Mean-Square (RMS) Error: $E_{\text{RMS}} = \sqrt{2E(\mathbf{w}^*)/N}$

Polynomial Coefficients

	$M = 0$	$M = 1$	$M = 3$	$M = 9$
w_0^*	0.19	0.82	0.31	0.35
w_1^*		-1.27	7.99	232.37
w_2^*			-25.43	-5321.83
w_3^*			17.37	48568.31
w_4^*				-231639.30
w_5^*				640042.26
w_6^*				-1061800.52
w_7^*				1042400.18
w_8^*				-557682.99
w_9^*				125201.43

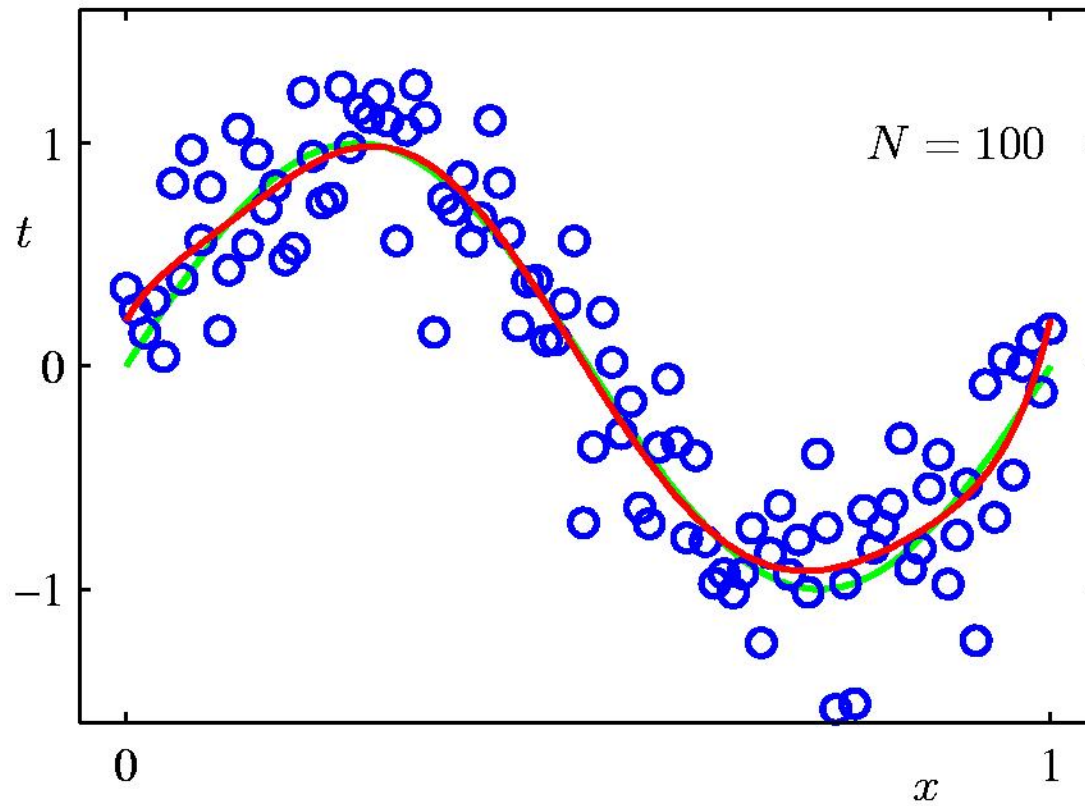
Data Set Size: $N = 15$

9th Order Polynomial



Data Set Size: $N = 100$

9th Order Polynomial



Regularized Least Squares (1)

- Consider the error function:

$$E_D(\mathbf{w}) + \lambda E_W(\mathbf{w})$$

Data term + Regularization term

- With the sum-of-squares error function and a quadratic regularizer, we get

$$\frac{1}{2} \sum_{n=1}^N \{t_n - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n)\}^2 + \frac{\lambda}{2} \mathbf{w}^T \mathbf{w}$$

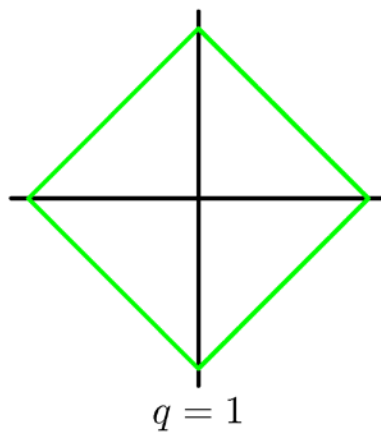
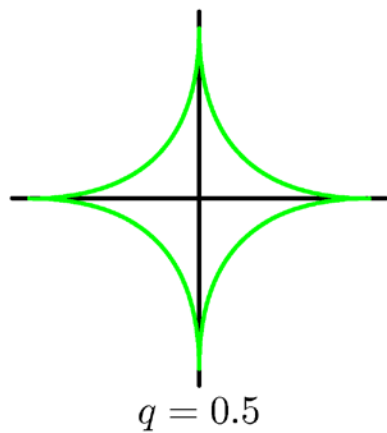
- which is minimized by

$$\mathbf{w} = \left(\lambda \mathbf{I} + \boldsymbol{\Phi}^T \boldsymbol{\Phi} \right)^{-1} \boldsymbol{\Phi}^T \mathbf{t}.$$

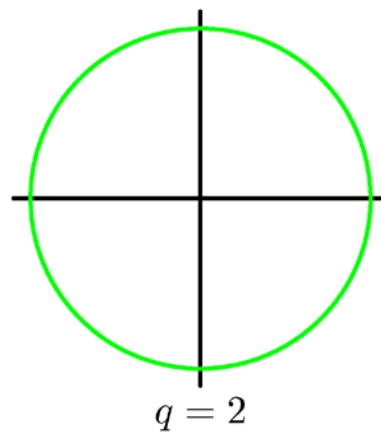
Regularized Least Squares (2)

- With a more general regularizer, we have

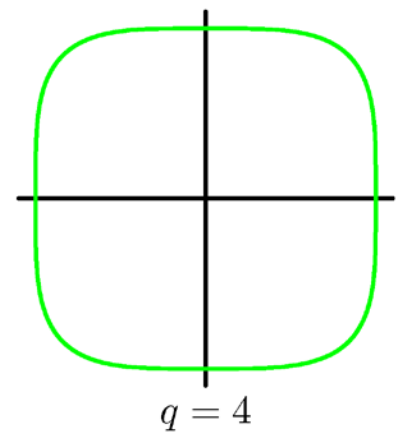
$$\frac{1}{2} \sum_{n=1}^N \{t_n - \mathbf{w}^T \phi(\mathbf{x}_n)\}^2 + \frac{\lambda}{2} \sum_{j=1}^M |w_j|^q$$



Lasso

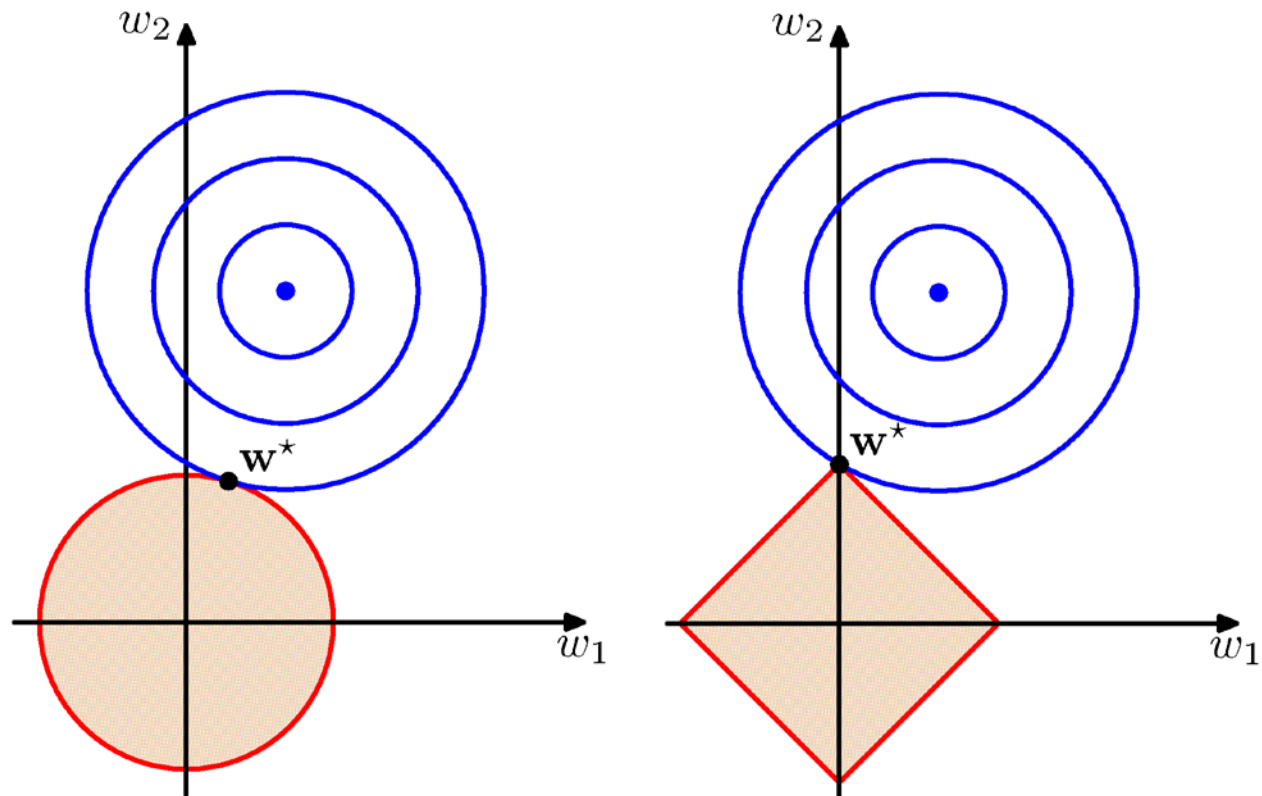


Quadratic

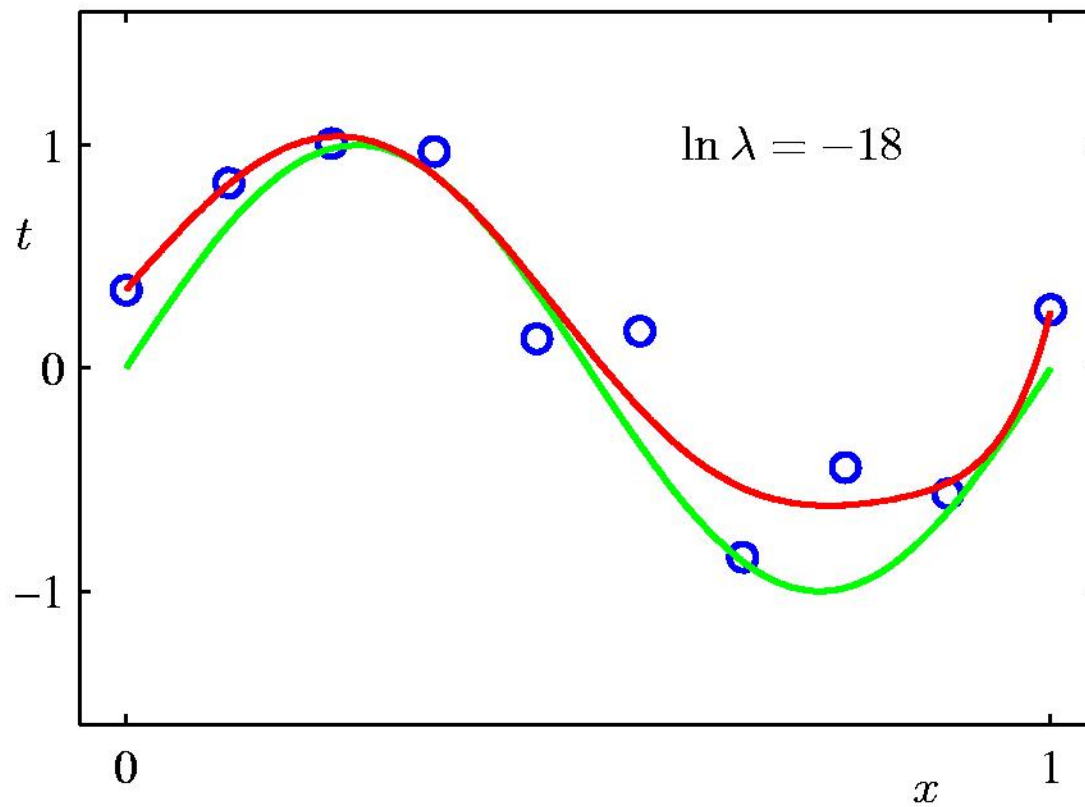


Regularized Least Squares (3)

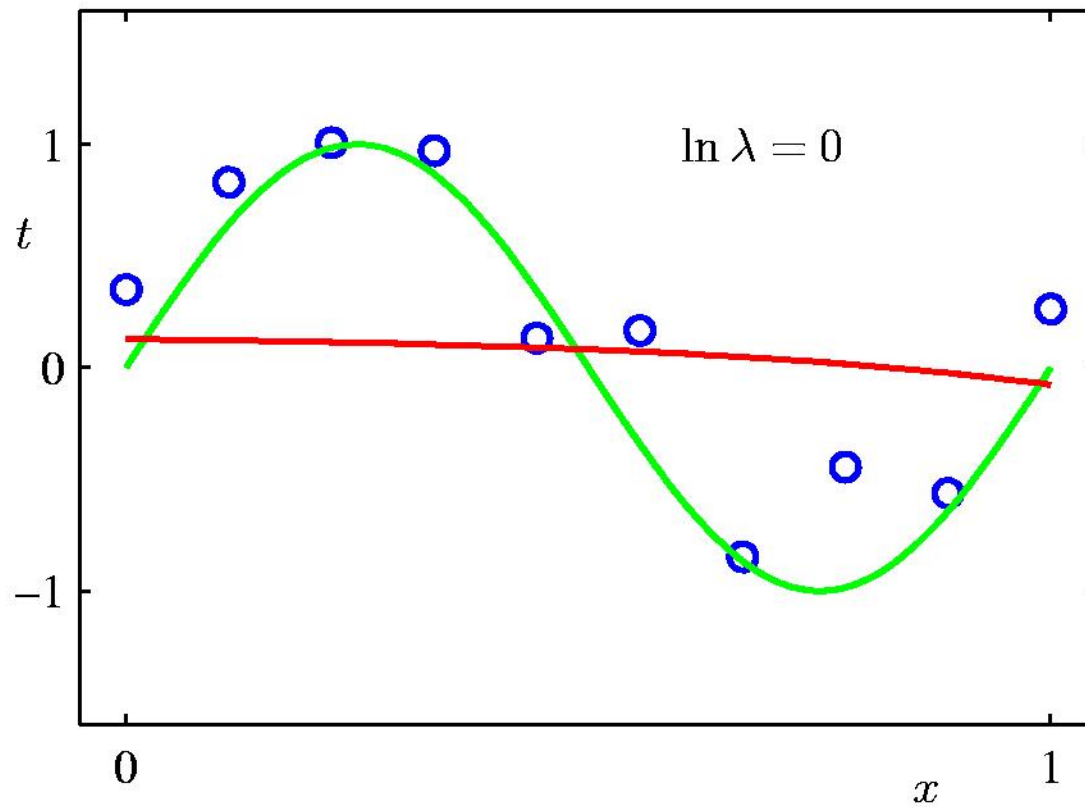
- Lasso tends to generate sparser solutions than a quadratic regularizer.



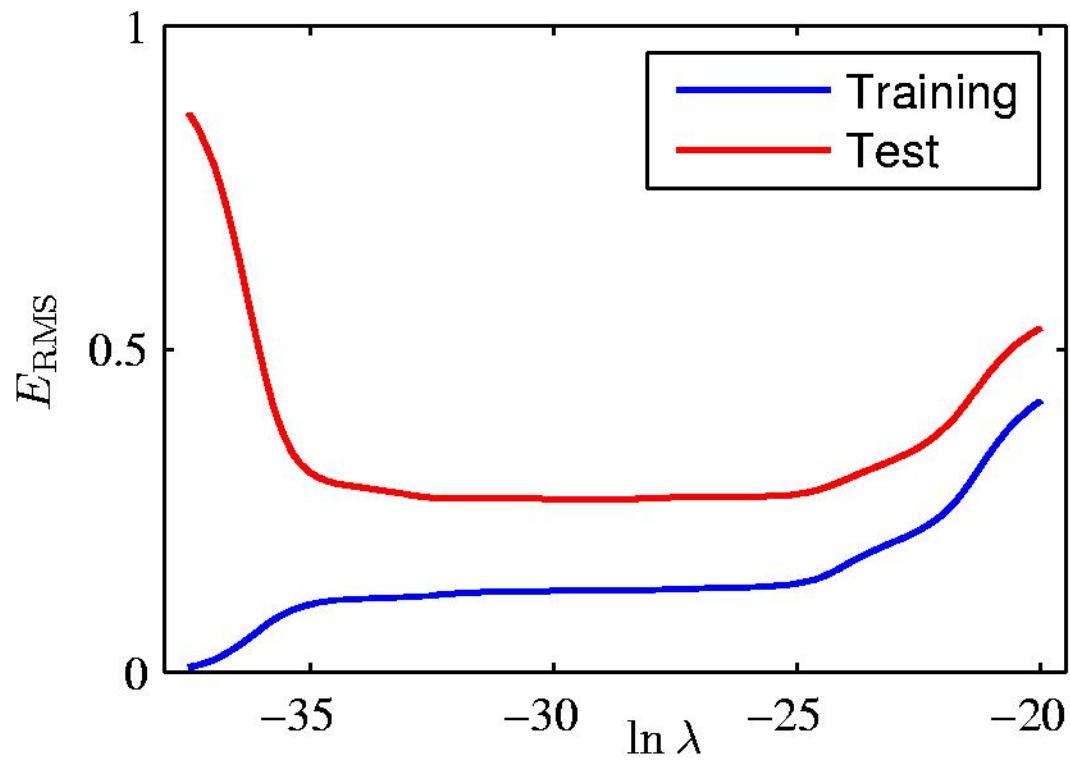
Regularization: $\ln \lambda = -18$



Regularization: $\ln \lambda = 0$



Regularization: E_{RMS} vs. $\ln \lambda$



Polynomial Coefficients

	$\ln \lambda = -\infty$	$\ln \lambda = -18$	$\ln \lambda = 0$
w_0^*	0.35	0.35	0.13
w_1^*	232.37	4.74	-0.05
w_2^*	-5321.83	-0.77	-0.06
w_3^*	48568.31	-31.97	-0.05
w_4^*	-231639.30	-3.89	-0.03
w_5^*	640042.26	55.28	-0.02
w_6^*	-1061800.52	41.32	-0.01
w_7^*	1042400.18	-45.95	-0.00
w_8^*	-557682.99	-91.53	0.00
w_9^*	125201.43	72.68	0.01

Bayesian Linear Regression (1)

- Define a conjugate prior over $\mathbf{w}$

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

- Combining this with the likelihood function and using results for marginal and conditional Gaussian distributions, gives the posterior

- where

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

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

$$\mathbf{S}_N^{-1} = \mathbf{S}_0^{-1} + \beta \Phi^T \Phi.$$

Bayesian Linear Regression (2)

- A common choice for the prior is

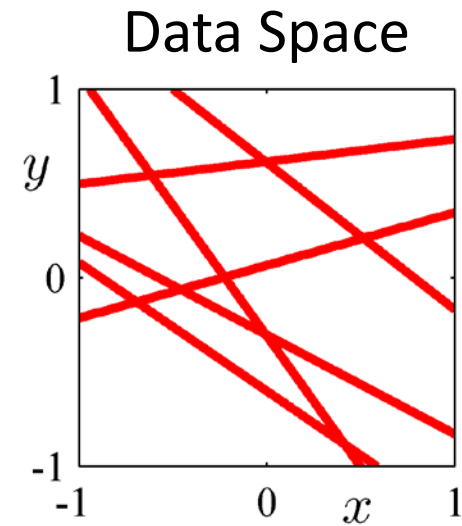
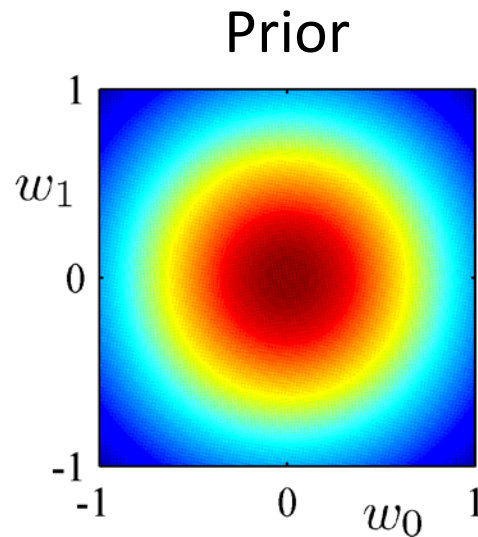
- for which $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I})$

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

- Next we consider an example ...

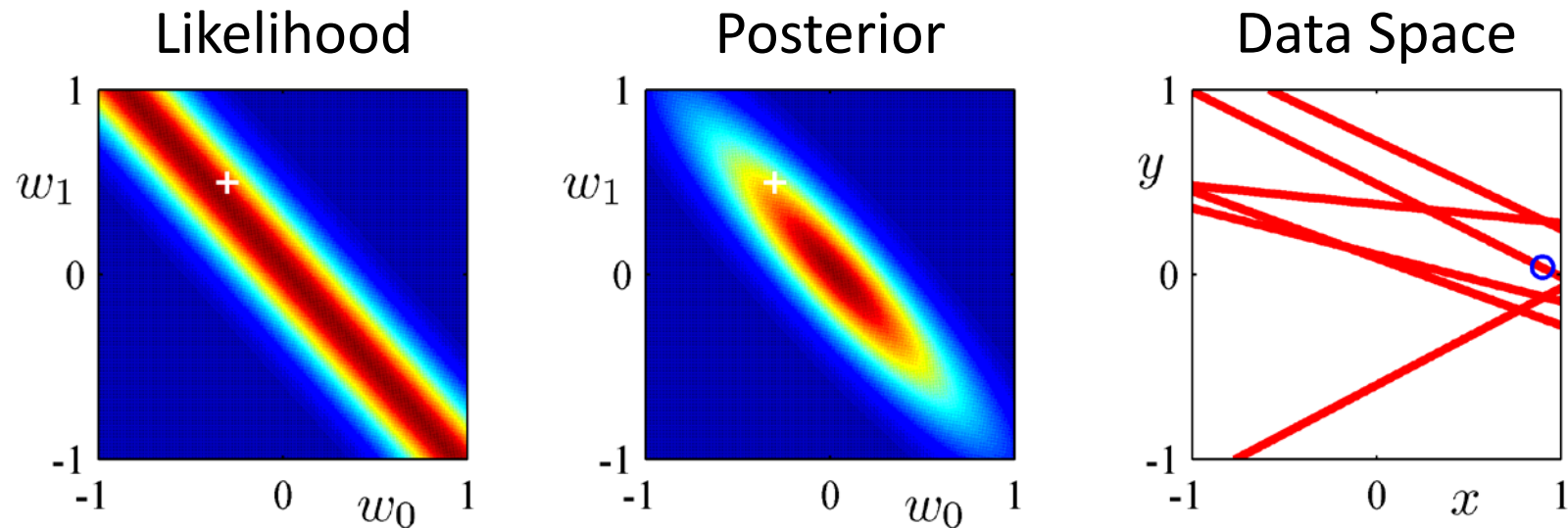
Bayesian Linear Regression (3)

0 data points observed



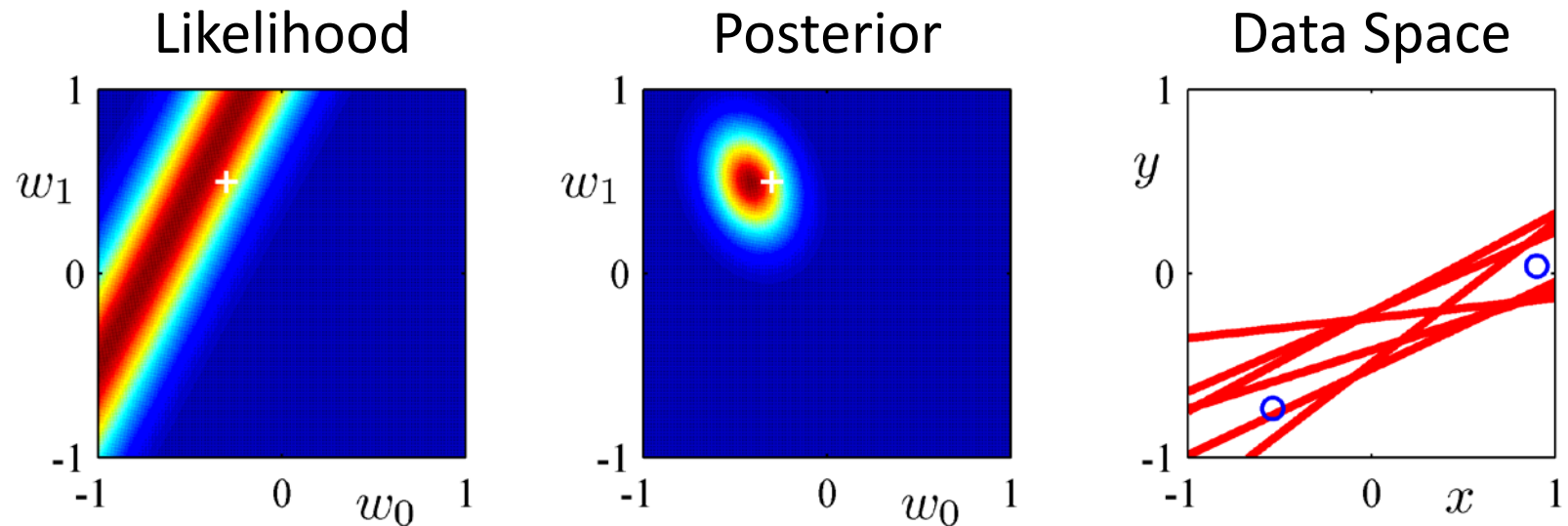
Bayesian Linear Regression (4)

1 data point observed



Bayesian Linear Regression (5)

2 data points observed



Bayesian Linear Regression (6)

20 data points observed

