Introduction to Machine Learning

Brown University CSCI 1950-F, Spring 2012
Prof. Erik Sudderth

Lecture 9:
Bayesian Prediction & Linear Regression
Logistic & Probit Regression
Gaussian Discriminant Analysis

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

Generative or Discriminative?

$y \longrightarrow$ output (discrete class, continuous response, …)

$x \longrightarrow$ input features to be used for classification, regression, …

$\theta \longrightarrow$ parameters to be learned, pick model from family

Generative Models

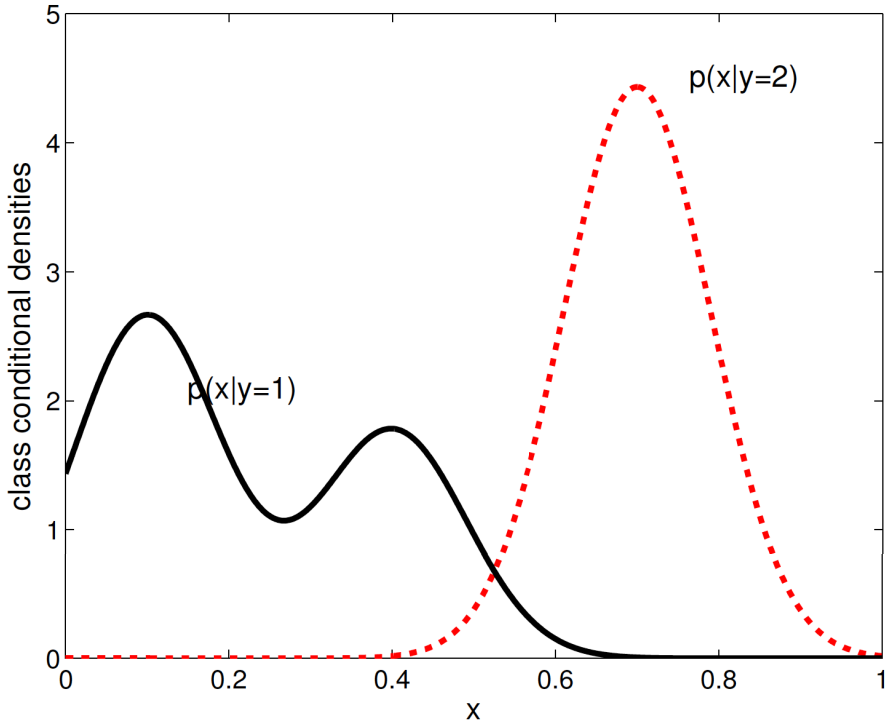
- *Training:* Learn prior and likelihood: $p(y \mid \theta), p(x \mid y, \theta)$
- *Test:* Posterior from Bayes' rule:

$$p(y \mid x) \propto p(y \mid \theta) p(x \mid y, \theta)$$

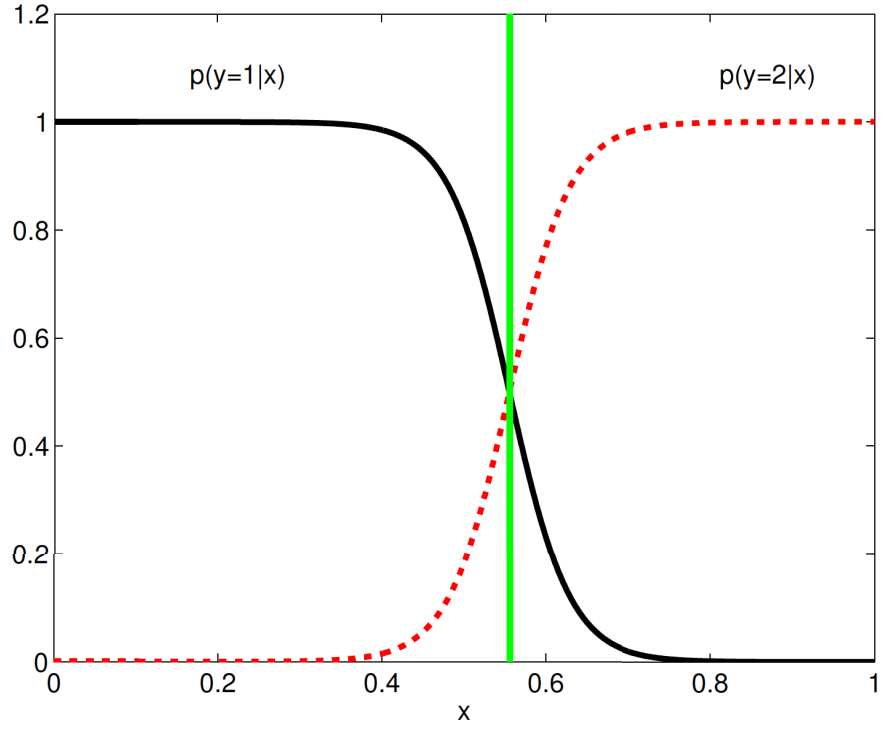
Discriminative or Conditional Models

- *Training:* Learn posterior: $p(y \mid x, \theta)$
- *Test:* Apply posterior:
- *Con:* Easier to incorporate domain knowledge generatively
- *Con:* Cannot handle missing features, no model of $p(x)$
- *Pro:* No need to design an accurate model of $p(x)$

Discrimination can be Simpler



Likelihood functions



Posterior distributions

All optimal binary decision rules depend on likelihood ratio:

$$\frac{p(y=1 \mid x)}{p(y=0 \mid x)} = \frac{p(y=1)p(x \mid y=1)}{p(y=0)p(x \mid y=0)}$$

Discriminative ***Generative***

Discriminative Bayesian Learning

Posterior Predictive Distribution

$$p(y_{\text{test}} \mid x_{\text{test}}, y_{\text{train}}, x_{\text{train}}) = \int_{\Theta} p(y_{\text{test}} \mid x_{\text{test}}, \theta) p(\theta \mid y_{\text{train}}, x_{\text{train}}) \, d\theta$$

But can we compute this integral?

Maximum a Posteriori (MAP) Estimation

$$p(y_{\text{test}} \mid x_{\text{test}}, y_{\text{train}}, x_{\text{train}}) \approx p(y_{\text{test}} \mid x_{\text{test}}, \hat{\theta})$$

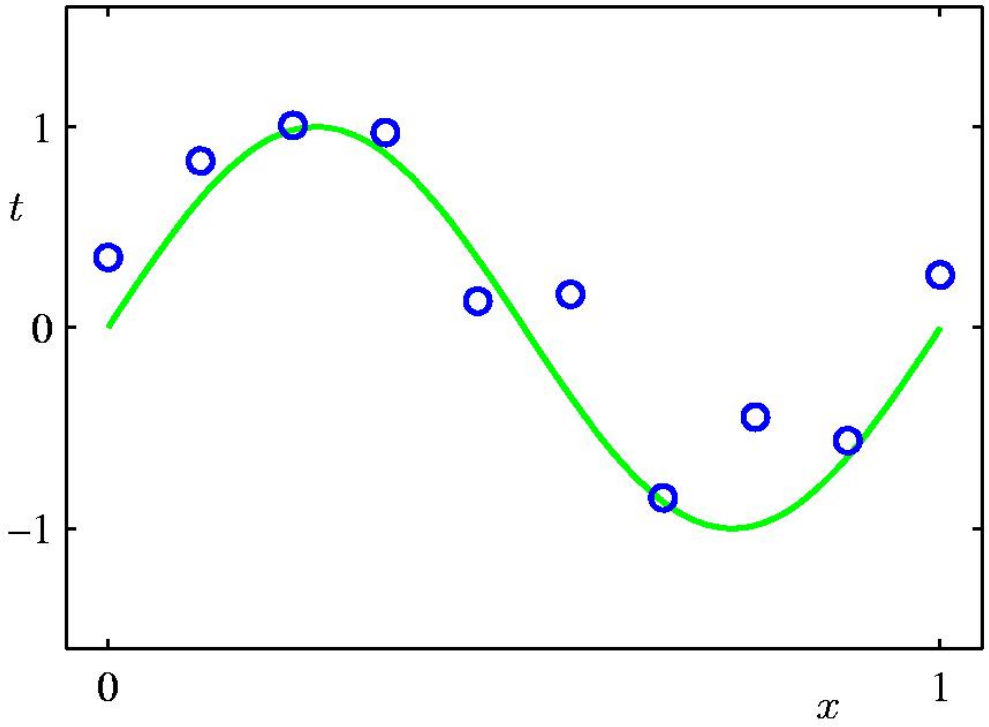
$$\hat{\theta} = \arg\max_{\theta} \; \log p(\theta) + \sum_{i} \log p(y_i \mid x_i, \theta)$$

Maximum Likelihood (ML) Estimation

$$p(y_{\text{test}} \mid x_{\text{test}}, y_{\text{train}}, x_{\text{train}}) \approx p(y_{\text{test}} \mid x_{\text{test}}, \hat{\theta})$$

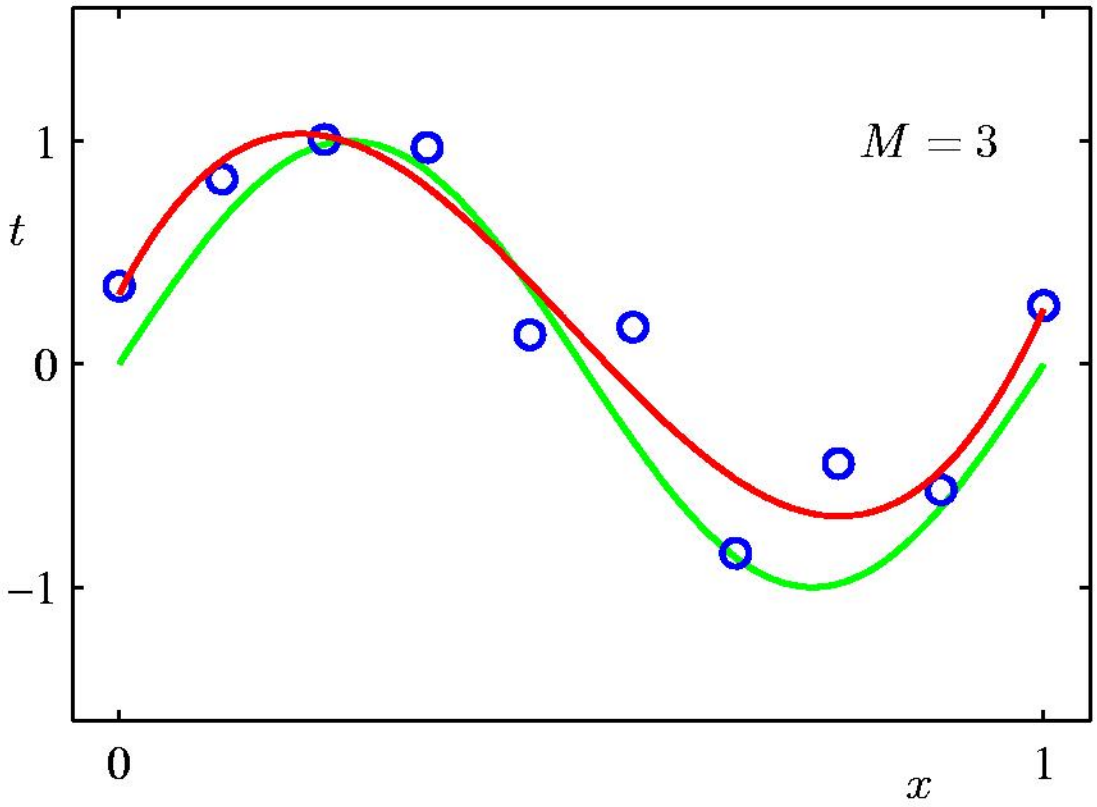
$$\hat{\theta} = \arg\max_{\theta} \sum_{i} \log p(y_i \mid x_i, \theta)$$

Polynomial Regression



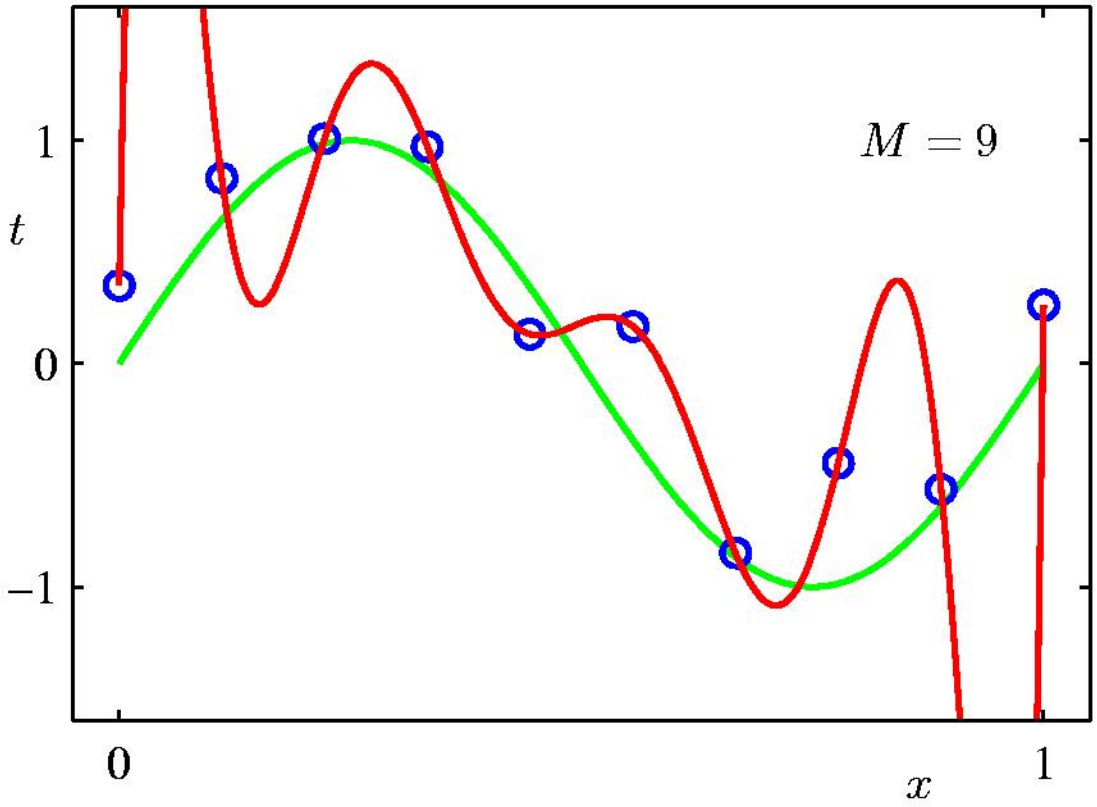
$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

ML Estimation: N=10, M=3



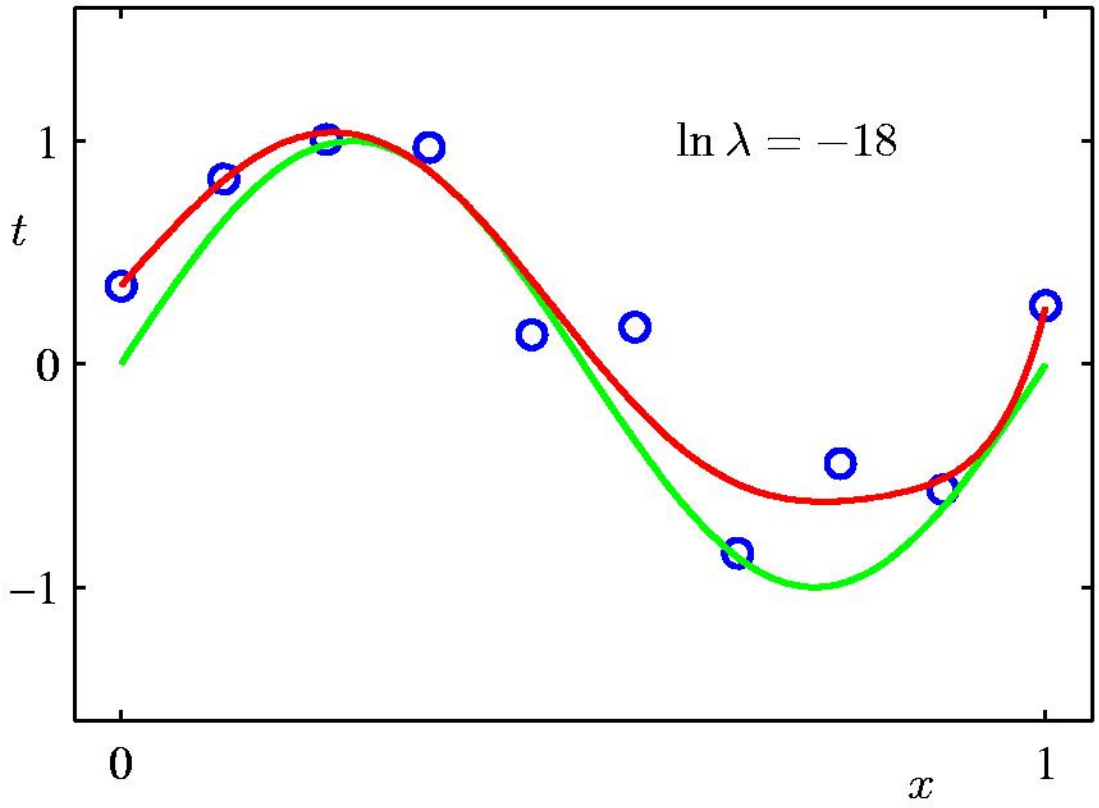
$$\hat{w} = (\Phi^T \Phi)^{-1} \Phi^T y$$

ML Estimation: N=10, M=9



$$\hat{w} = (\Phi^T \Phi)^{-1} \Phi^T y$$

MAP Estimation: N=10, M=9



$$\hat{w} = (\Phi^T \Phi + \lambda I)^{-1} \Phi^T y$$

Posterior Predictive Distribution

- Posterior distribution given training data is Gaussian:

$$p(w \mid y_{\text{train}}, x_{\text{train}}) = \mathcal{N}(w \mid m_N, S_N)$$

- Predictive distribution is then also Gaussian:

$$p(y_{\text{test}} \mid x_{\text{test}}, y_{\text{train}}, x_{\text{train}}) =$$

$$= \int \mathcal{N}(y_{\text{test}} \mid \phi(x_{\text{test}})^T w, \beta^{-1}) \mathcal{N}(w \mid m_N, S_N)\, dw$$

$$= \mathcal{N}(y_{\text{test}} \mid \phi(x_{\text{test}})^T m_N, \beta^{-1} + \phi(x_{\text{test}})^T S_N \phi(x_{\text{test}}))$$

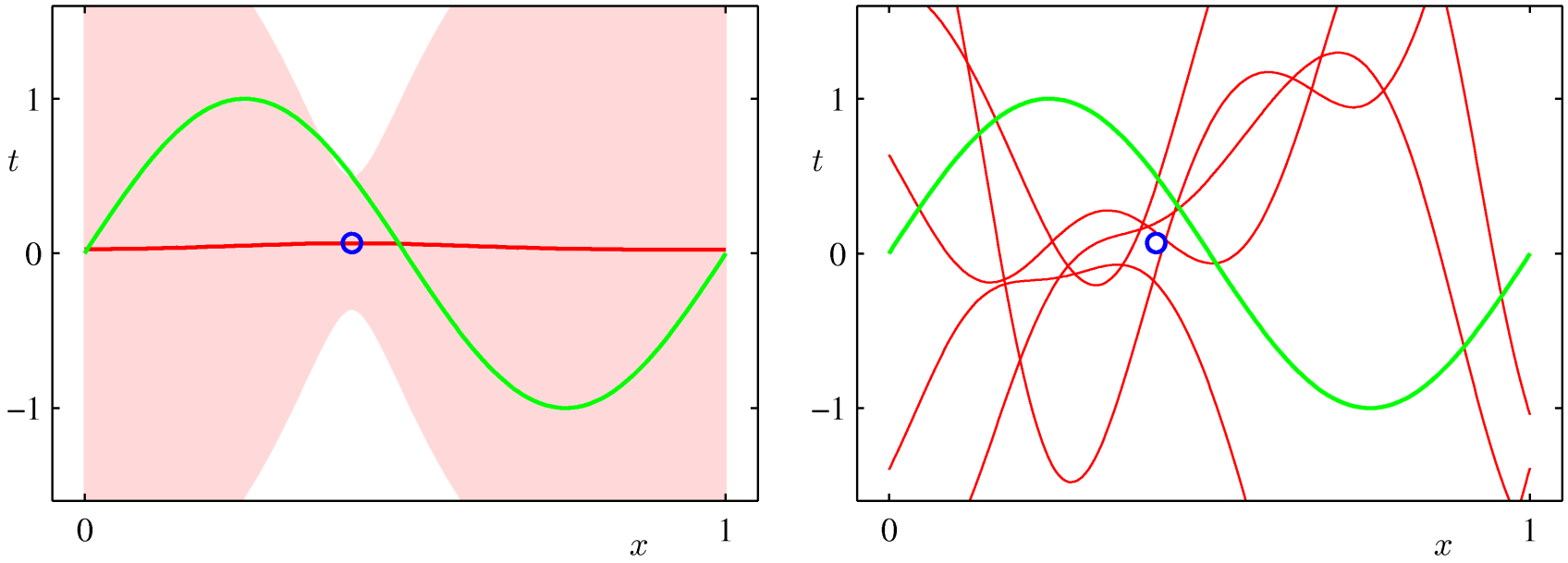
- How does this prediction relate to the MAP estimate?

$$\hat{w} = \arg\max_w \; p(w \mid y_{\text{train}}, x_{\text{train}}) = m_N$$

$$p(y_{\text{test}} \mid x_{\text{test}}, \hat{w}) = \mathcal{N}(y_{\text{test}} \mid \phi(x_{\text{test}})^T m_N, \beta^{-1})$$

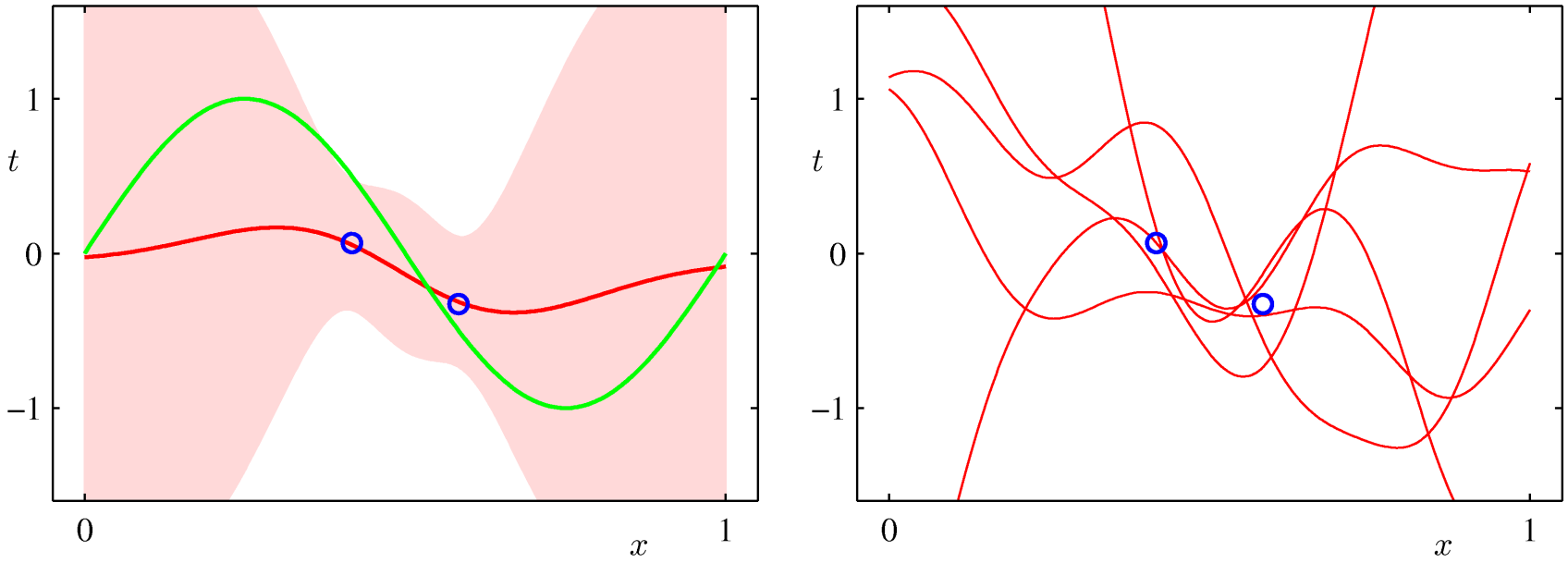
Predictive Distribution: N=1

- Example: Sinusoidal data, 9 Gaussian basis functions, 1 data point



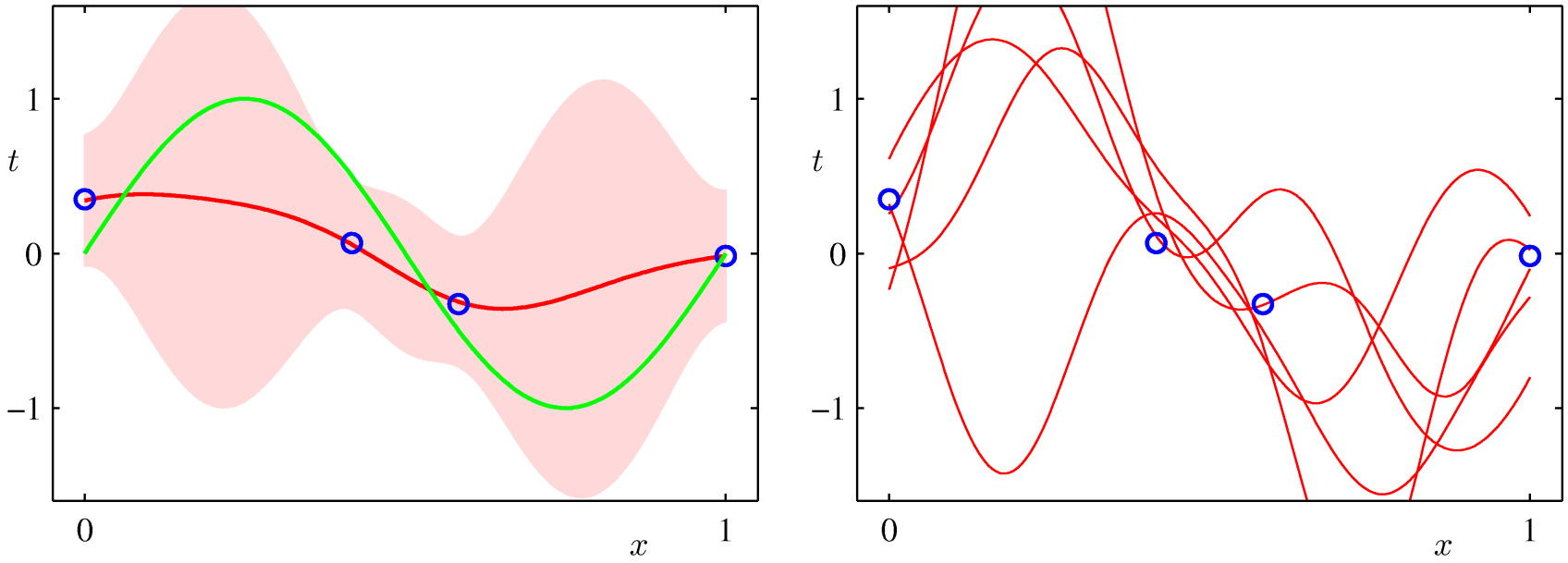
Predictive Distribution: N=2

- Example: Sinusoidal data, 9 Gaussian basis functions, 2 data points



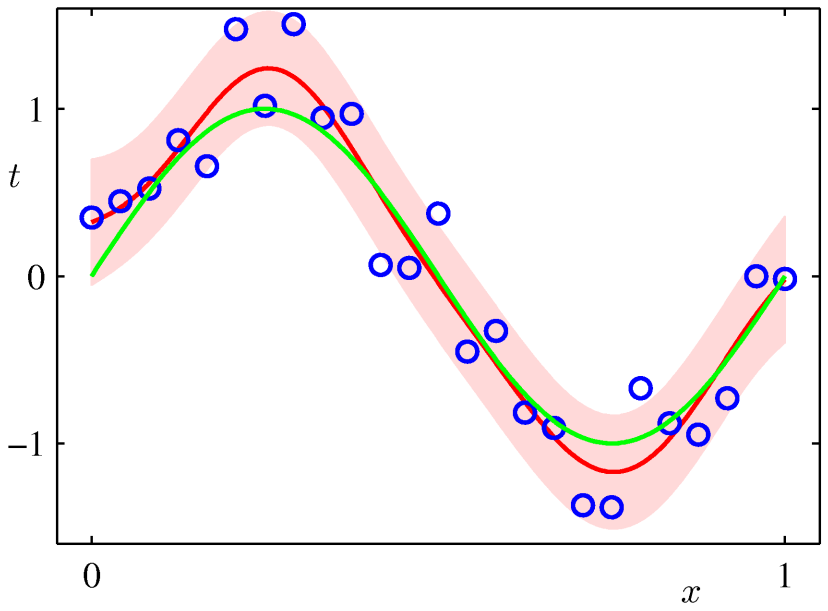
Predictive Distribution: N=4

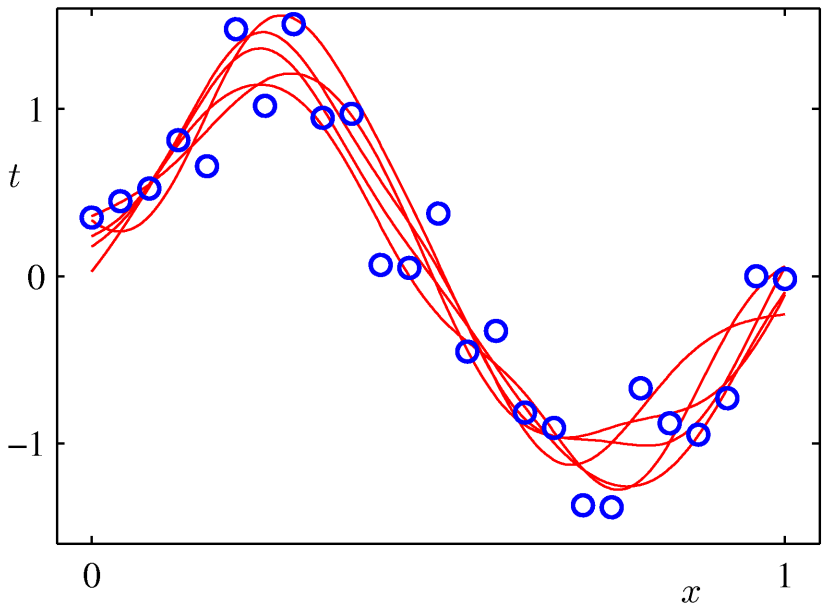
- Example: Sinusoidal data, 9 Gaussian basis functions, 4 data points



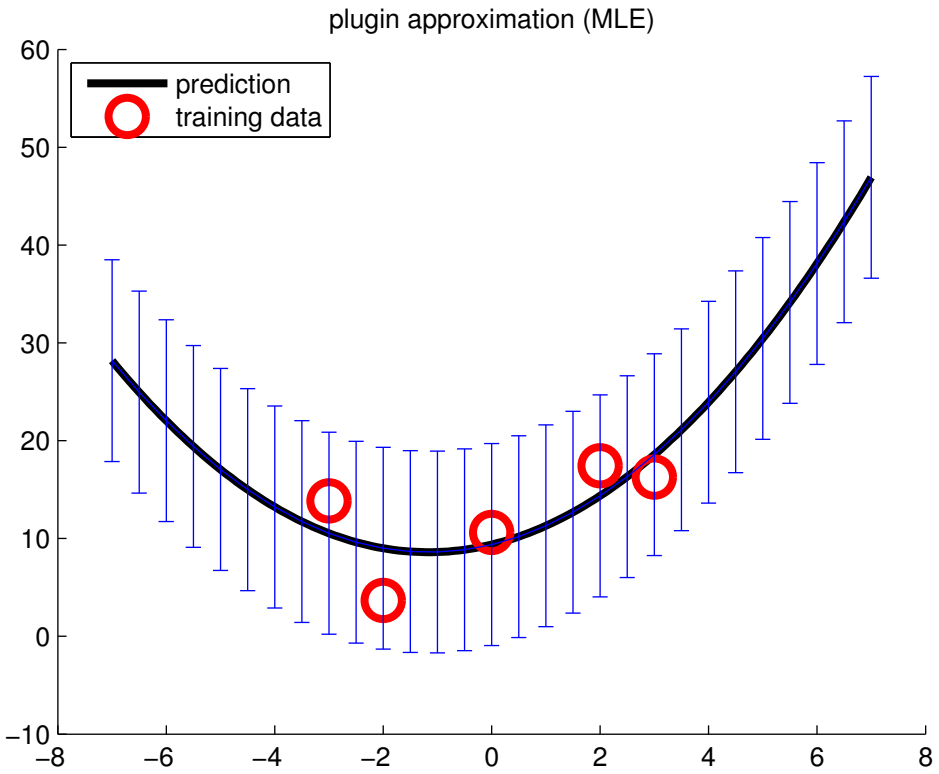
Predictive Distribution: N=25

- Example: Sinusoidal data, 9 Gaussian basis functions, 25 data points

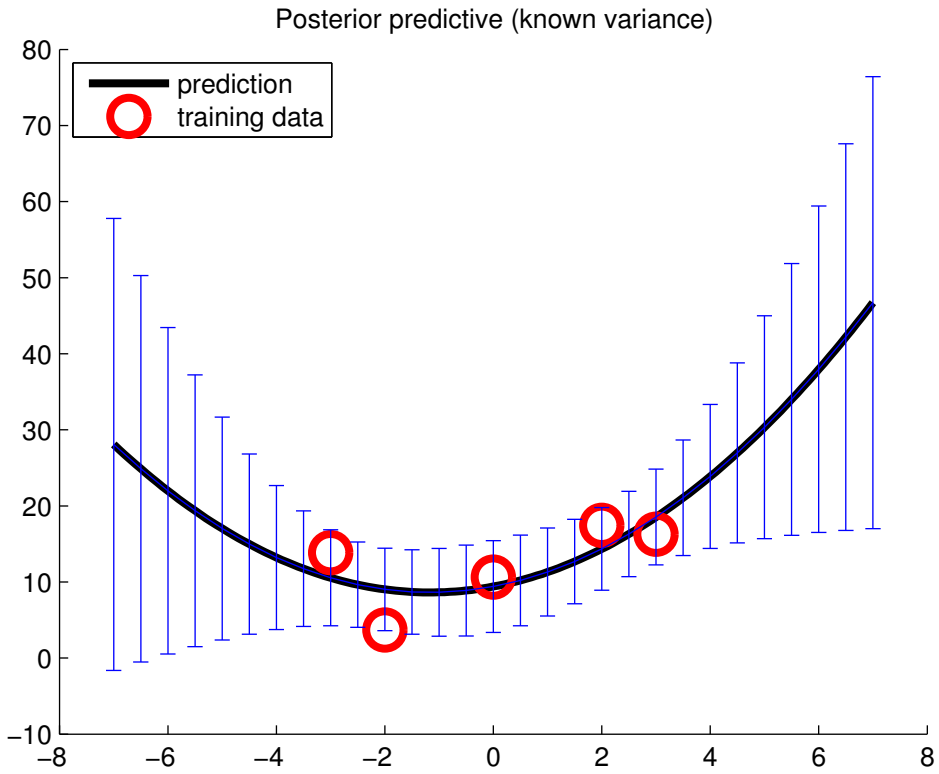




Estimation vs. Predictive Distributions



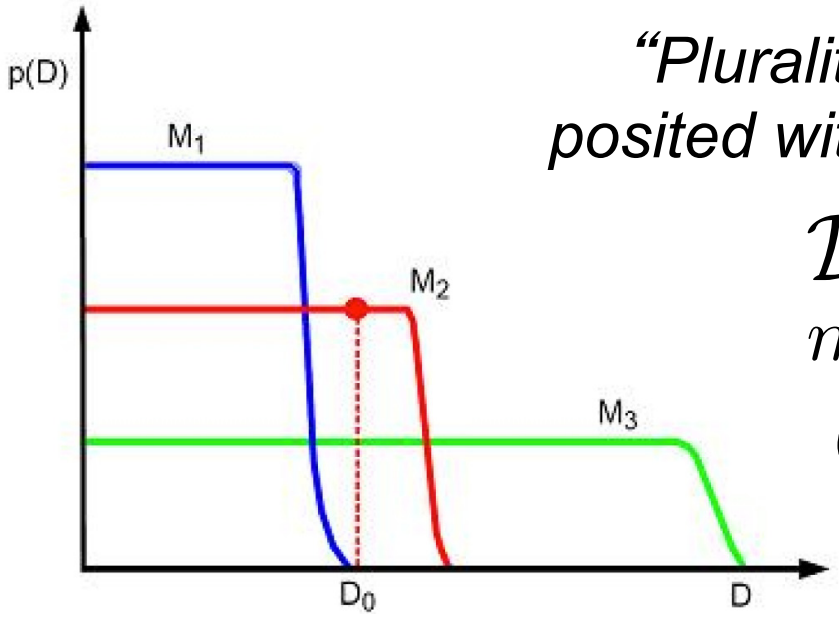
MAP Estimate, Uniform Prior



Gaussian Predictive Distribution

$$\mathcal{N}(y_{\text{test}} \mid \phi(x_{\text{test}})^T m_N, \beta^{-1} + \phi(x_{\text{test}})^T S_N \phi(x_{\text{test}}))$$

Bayesian Ockham's Razor



"Plurality must never be posited without necessity."

$\mathcal{D}$ → *data*

m → *model*

θ → *parameters*

William of Ockham

$$p(\mathcal{D}|m) = \int p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta}|m)d\boldsymbol{\theta}$$

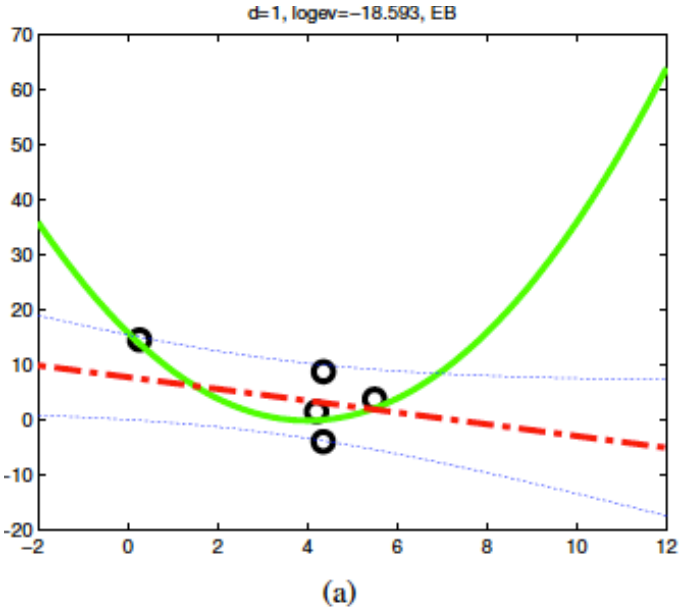
$$p(m|\mathcal{D}) = \frac{p(\mathcal{D}|m)p(m)}{\sum_{m\in\mathcal{M}} p(m,\mathcal{D})}$$

$$p(\mathcal{D} \mid m) \approx \frac{1}{S}\sum_{s=1}^{S} p(\mathcal{D} \mid \theta^{(s)})$$

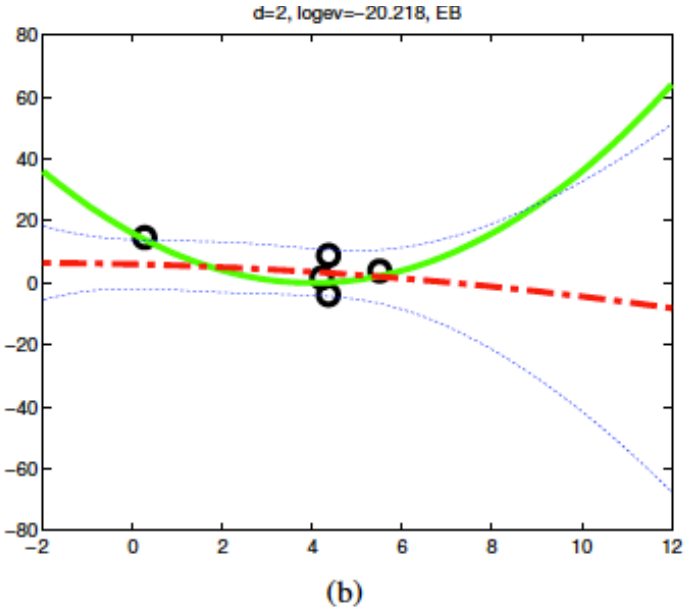
$$\theta^{(s)} \sim p(\theta \mid m)$$

Even with uniform $p(m)$, marginal likelihood provides a model selection bias

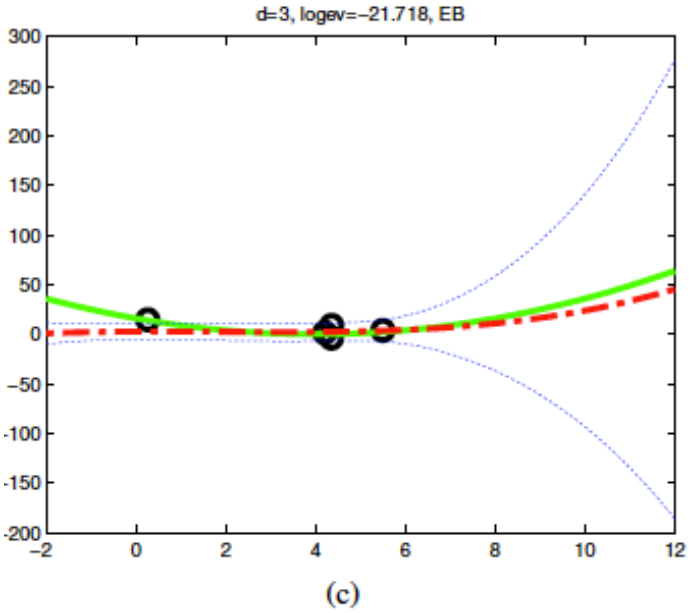
Marginal Data Likelihood: N=5



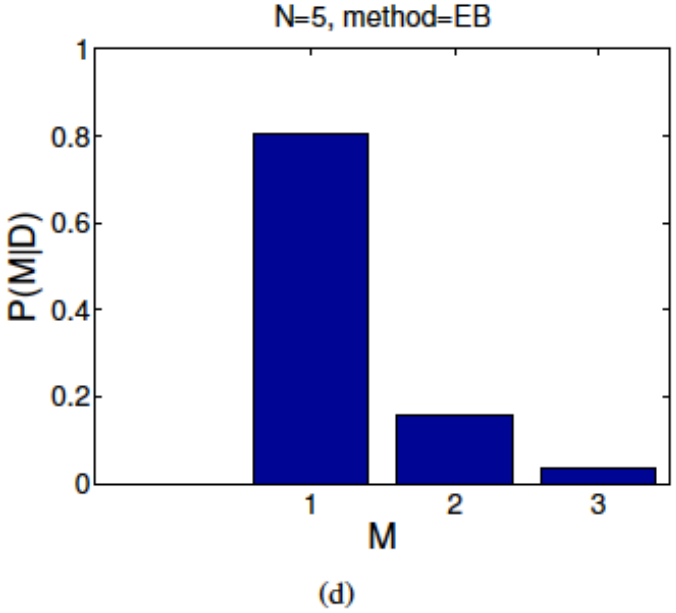
(a)



(b)

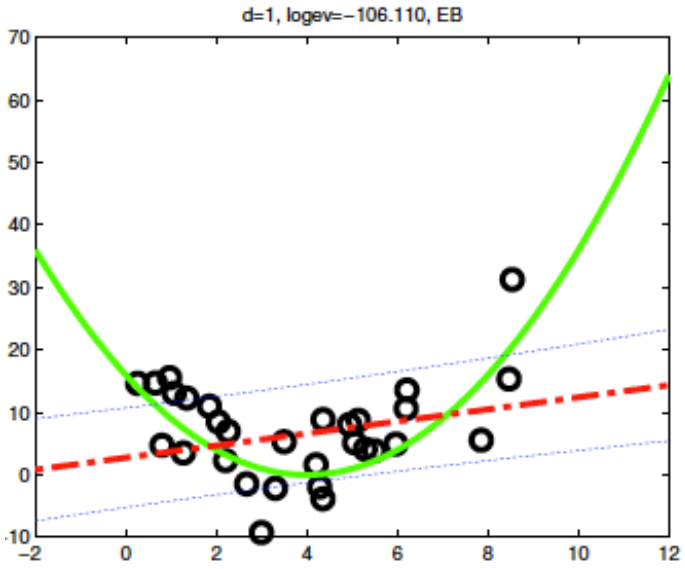


(c)

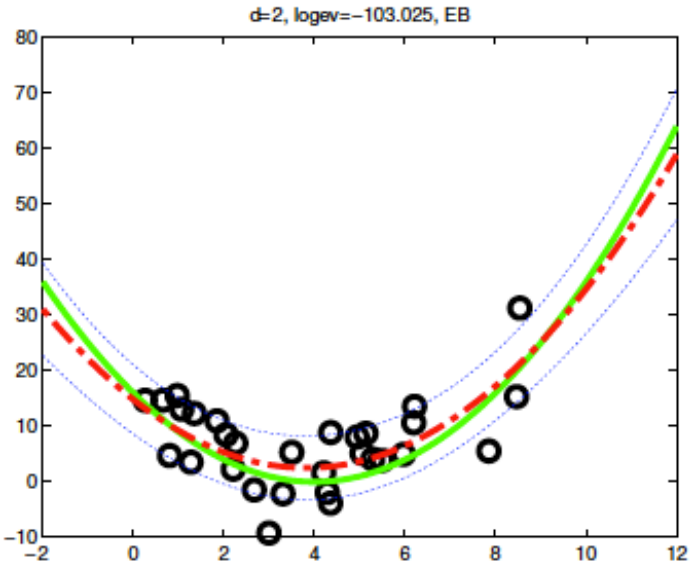


(d)

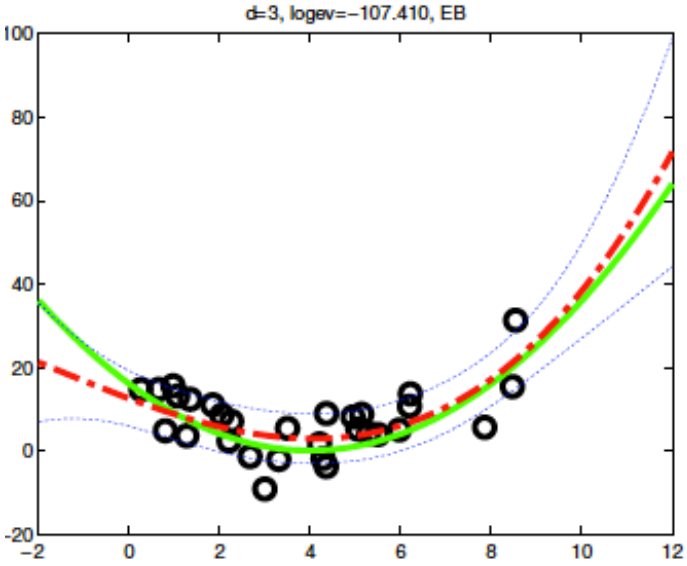
Marginal Data Likelihood: N=30



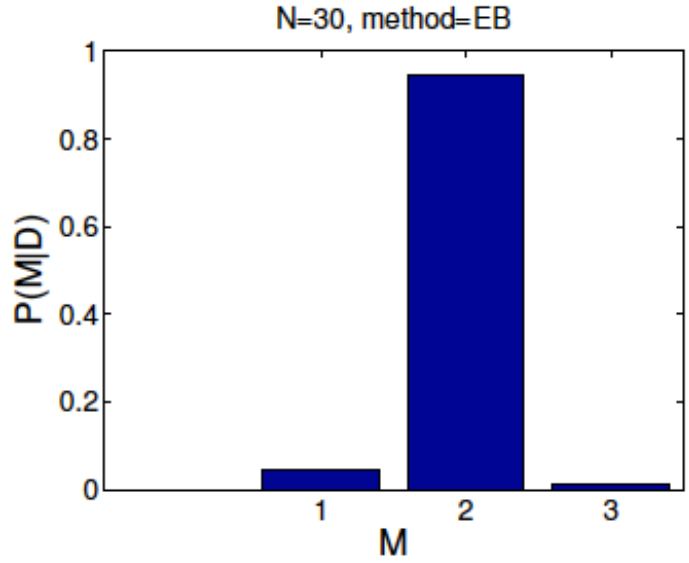
(a)



(b)



(c)



(d)

Classification as Regression

- How can we build a discriminative classification model?
- Classification as regression:
  - Training examples of class 1: $y_i = 1.0$
  - Training examples of class 0: $y_i = 0.0$
  - Least squares linear regression for learning:

$$p(y_i \mid x_i, w) = \mathcal{N}(y_i \mid \phi(x_i)^T w, \beta^{-1})$$

  - Classify test examples by thresholding:

$$\hat{y}_{\text{test}} = \mathbb{I}(\phi(x_{\text{test}})^T w > 0.5)$$

- Are there any problems with this idea?
  - For a training example of class 1, the Gaussian likelihood says that predictions of 0.0 and 2.0 are equally good
  - Does not estimate class probabilities $p(y_i = 1 \mid x_i)$
  - HW: more subtle, but major, problems with >2 classes

Probit Regression

y_i ⟶ binary class label for training example, $y_i \in \{0, 1\}$

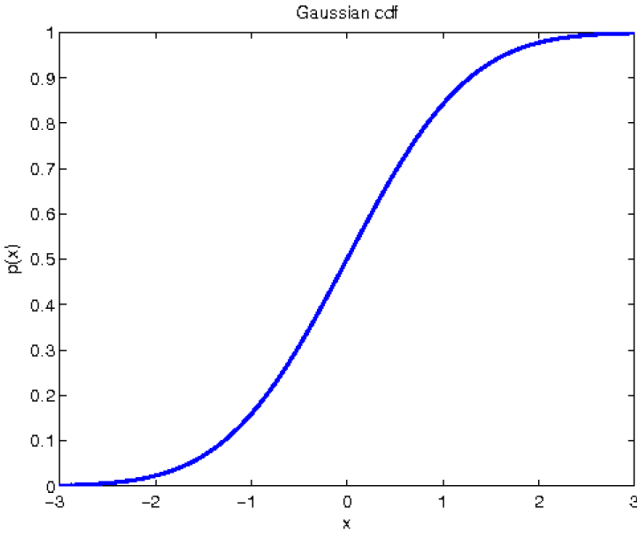
x_i ⟶ input features to be used for classification

w ⟶ weight vector of parameters to be learned

Probit regression is a model for *discriminative binary classification*:

$$z_i \sim \mathcal{N}(w^T \phi(x_i), 1) \qquad y_i = \mathbb{I}(z_i > 0)$$

$$p(y_i \mid x_i, w) = \text{Bernoulli}(F(w^T \phi(x_i)))$$

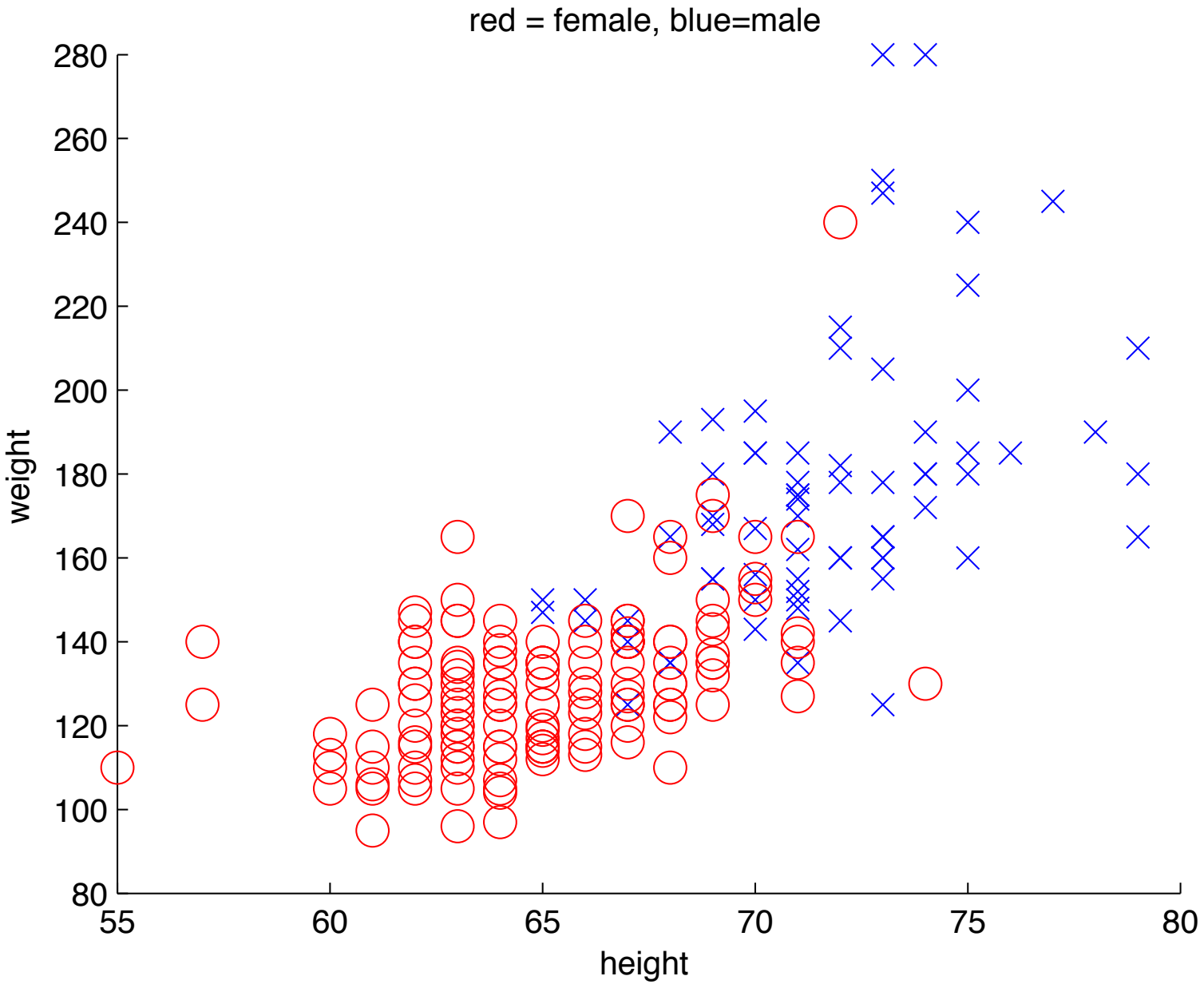


$$F(x) = \int_{-\infty}^{x} \mathcal{N}(z \mid 0, 1)\, dz$$

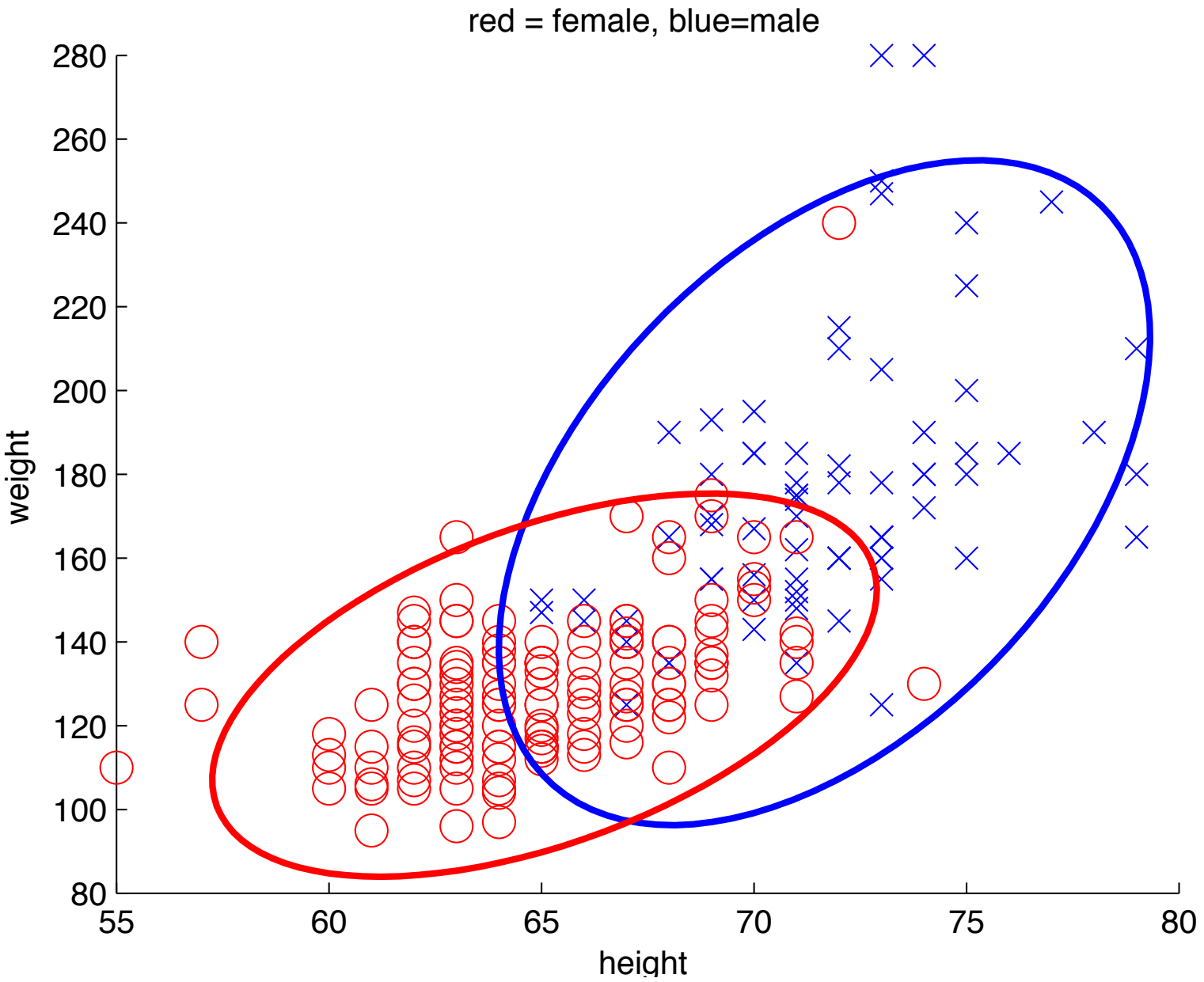
Questions:

- How can we estimate the weights from training data? Are ML, MAP, & Bayesian prediction tractable?
- Why choose to threshold Gaussian noise? Are other options better?

Generative Gaussian Classification



Generative Gaussian Classification



Gaussian Discriminant Analysis

y ⟶ class label in *{1,…,C}*, observed in training

$x \in \mathbb{R}^d$ ⟶ observed features to be used for classification

$$p(y, x \mid \pi, \theta) = p(y \mid \pi) p(x \mid y, \theta)$$

discriminant analysis is a generative classifier!

prior distribution

likelihood function

$$p(y \mid \pi) = \mathrm{Cat}(y \mid \pi)$$

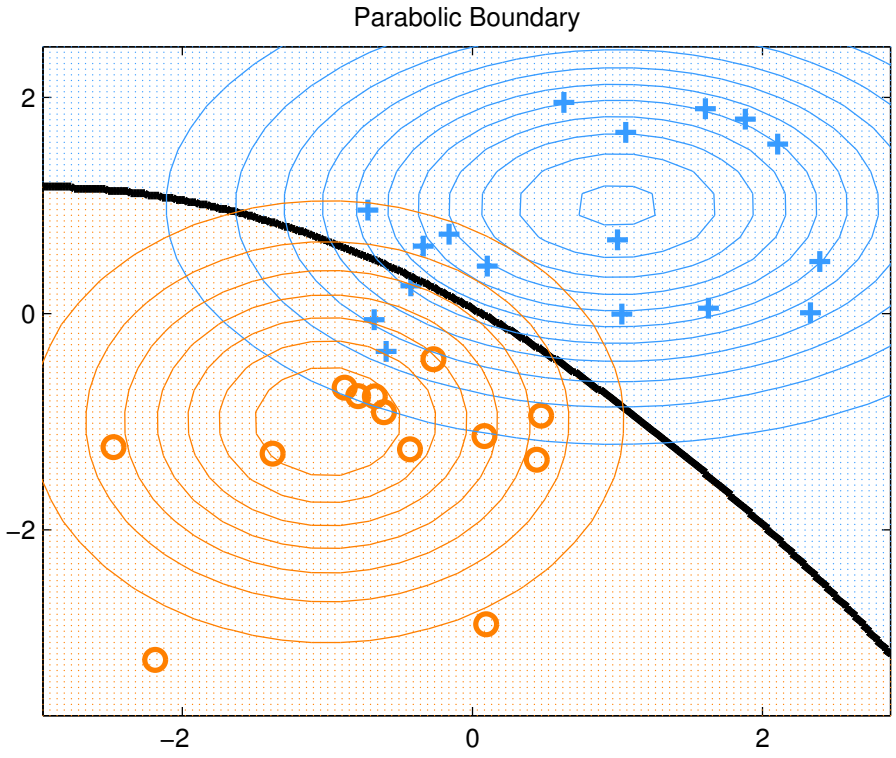
$$p(x \mid y = c, \theta) = \mathcal{N}(x \mid \mu_c, \Sigma_c) \qquad \theta_c = \{\mu_c, \Sigma_c\}$$

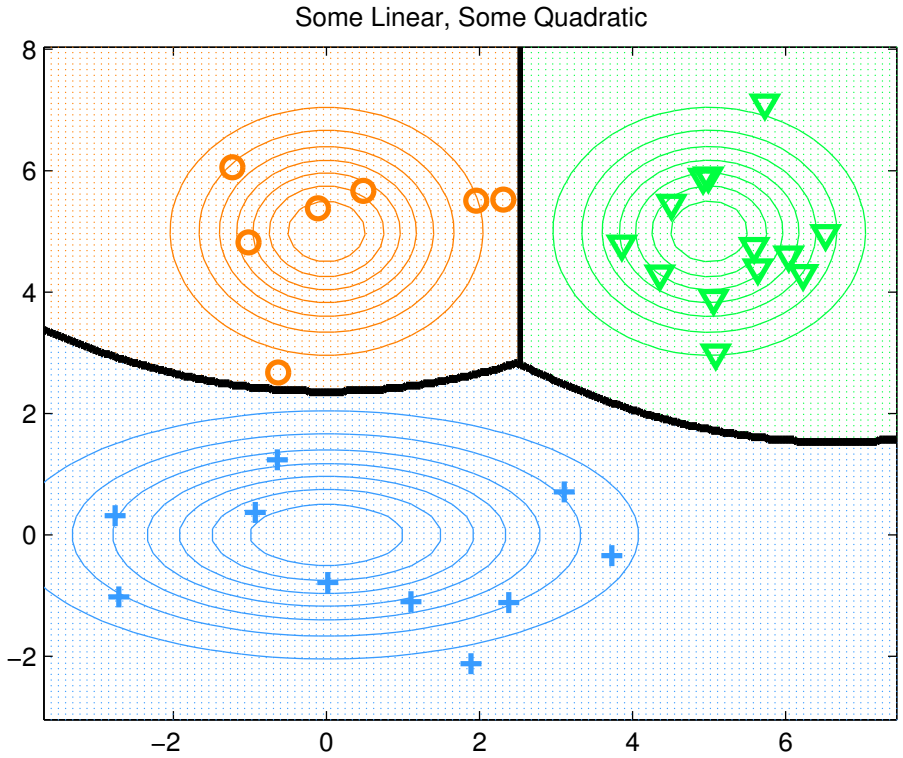
- Derive posterior distribution via Bayes' rule:

$$p(y = c \mid x, \theta, \pi) = \frac{p(y = c \mid \pi) p(x \mid y = c, \theta)}{\sum_{c'=1}^{C} p(y = c' \mid \pi) p(x \mid y = c', \theta)}$$

- What is the connection to the Gaussian naïve Bayes model?

Quadratic Discriminant Analysis

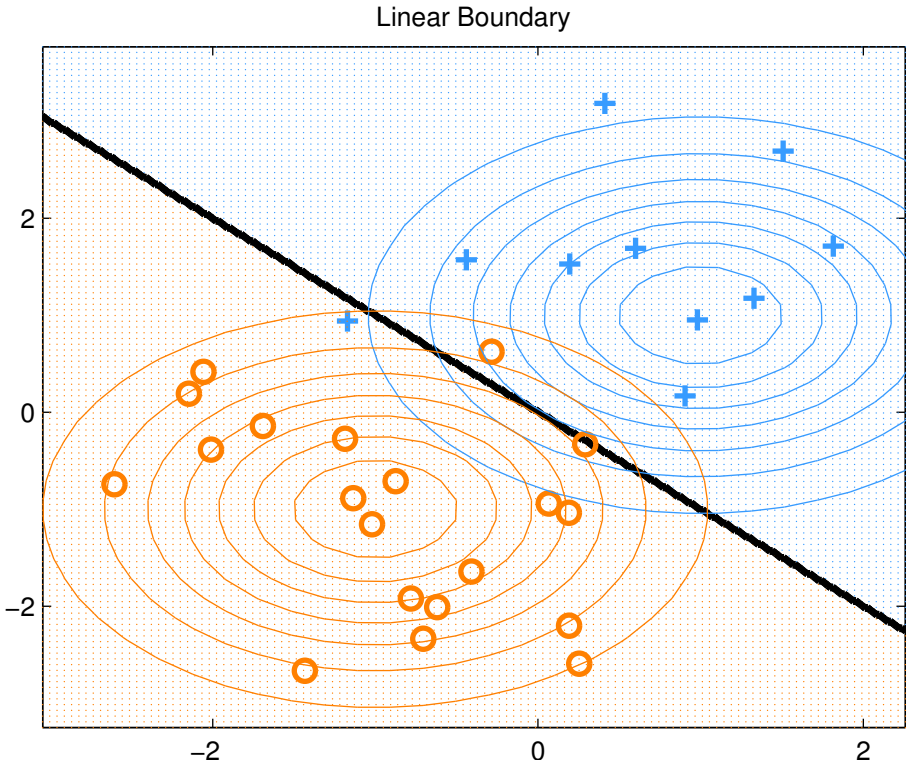


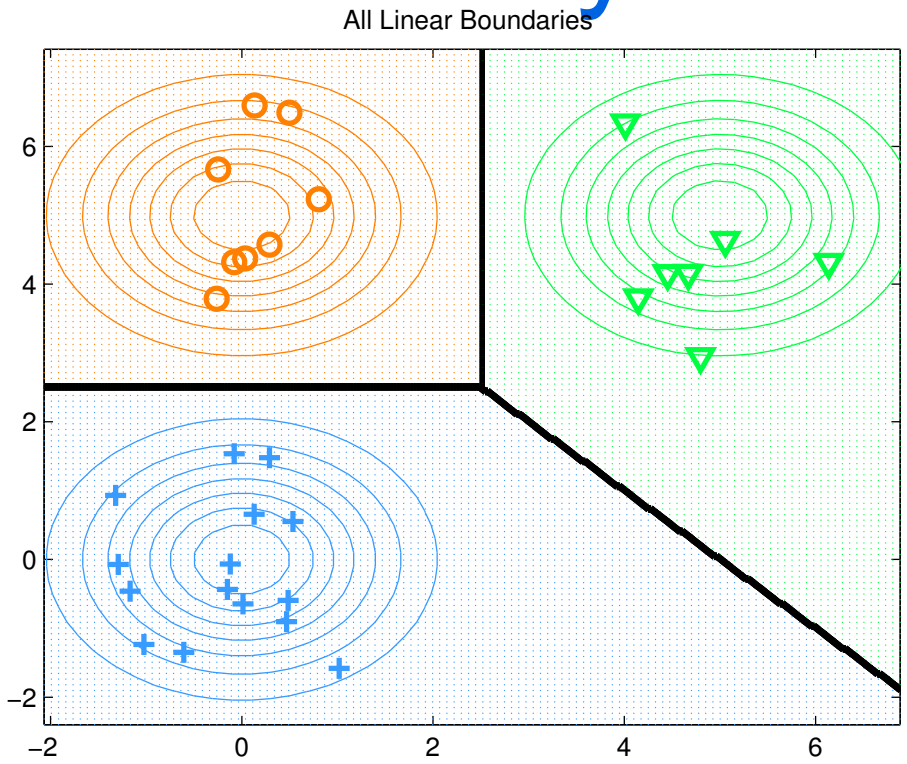


$$p(y = c|\mathbf{x}, \boldsymbol{\theta}) = \frac{\pi_c |2\pi\boldsymbol{\Sigma}_c|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_c)^T \boldsymbol{\Sigma}_c^{-1}(\mathbf{x} - \boldsymbol{\mu}_c)\right]}{\sum_{c'} \pi_{c'} |2\pi\boldsymbol{\Sigma}_{c'}|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_{c'})^T \boldsymbol{\Sigma}_{c'}^{-1}(\mathbf{x} - \boldsymbol{\mu}_{c'})\right]}$$

Optimal decision boundaries are quadratic functions

Linear Discriminant Analysis





$$p(y=c|\mathbf{x},\boldsymbol{\theta}) \propto \pi_c \exp\left[\boldsymbol{\mu}_c^T\boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_c^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_c\right]$$

$$= \exp\left[\boldsymbol{\mu}_c^T\boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_c^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_c + \log\pi_c\right]\exp[-\frac{1}{2}\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x}]$$

Optimal decision boundaries are linear functions if $\Sigma_c = \Sigma$

Linear Discriminant Analysis

Further simplifying:

$$\gamma_c = -\frac{1}{2}\boldsymbol{\mu}_c^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_c + \log \pi_c$$

$$\boldsymbol{\beta}_c = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_c$$

$$p(y=c|\mathbf{x},\boldsymbol{\theta}) = \frac{e^{\boldsymbol{\beta}_c^T \mathbf{x}+\gamma_c}}{\sum_{c'} e^{\boldsymbol{\beta}_{c'}^T \mathbf{x}+\gamma_{c'}}} = \mathcal{S}(\boldsymbol{\eta})_c$$

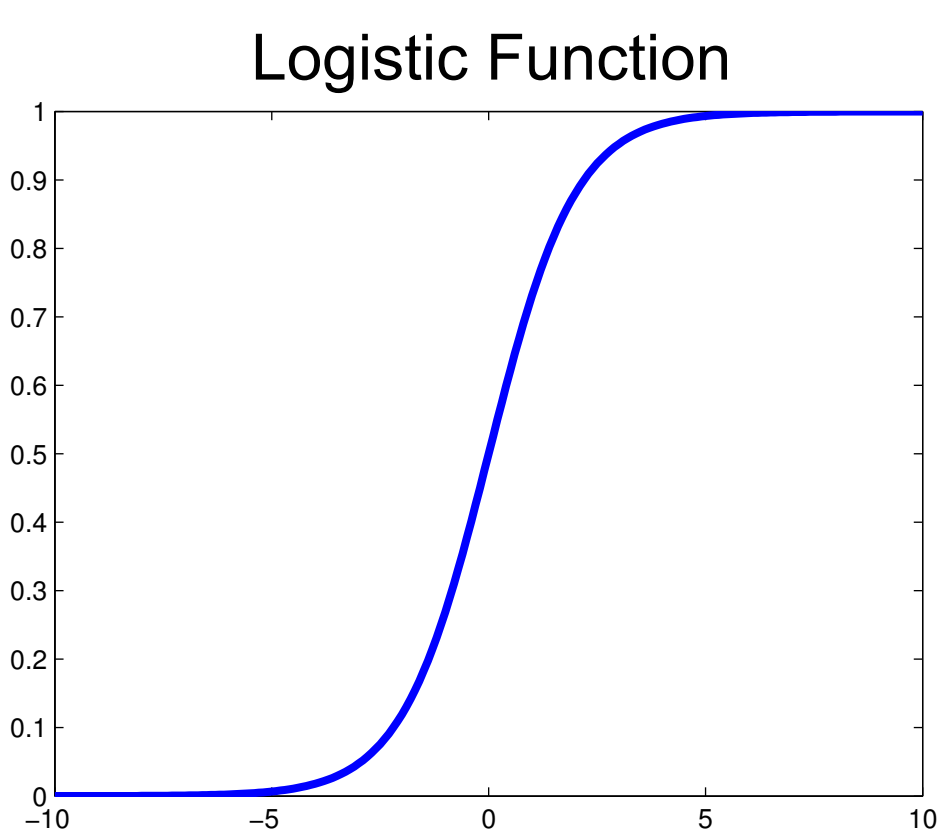
$$\boldsymbol{\eta} = [\boldsymbol{\beta}_1^T \mathbf{x} + \gamma_1, \ldots, \boldsymbol{\beta}_C^T \mathbf{x} + \gamma_C]$$

$$p(y=c|\mathbf{x},\boldsymbol{\theta}) \propto \pi_c \exp\left[\boldsymbol{\mu}_c^T \boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_c^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_c\right]$$

$$= \exp\left[\boldsymbol{\mu}_c^T \boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_c^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_c + \log \pi_c\right] \exp[-\frac{1}{2}\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x}]$$

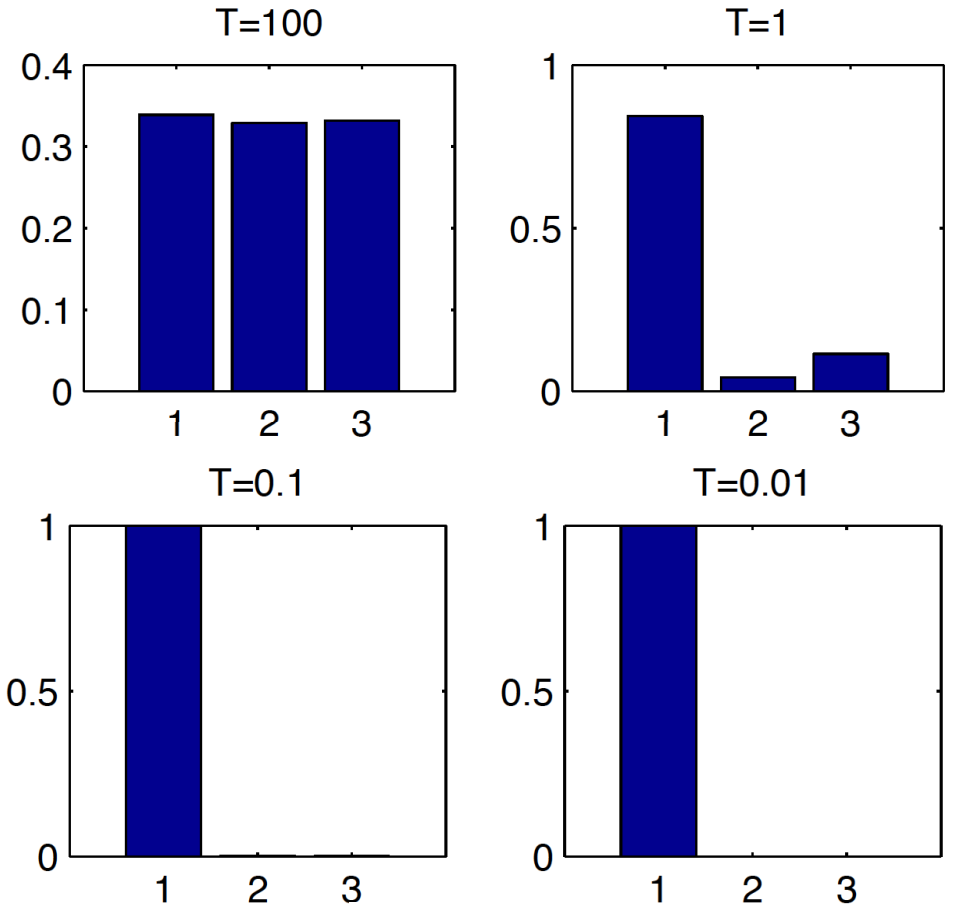
Optimal decision boundaries are linear functions if $\Sigma_c = \Sigma$

Logistic & Softmax Functions



$$\text{sigm}(\eta) := \frac{1}{1+\exp(-\eta)} = \frac{e^{\eta}}{e^{\eta}+1}$$

$$\mathcal{S}(\boldsymbol{\eta})_c = \frac{e^{\eta_c}}{\sum_{c'=1}^{C} e^{\eta_{c'}}}$$

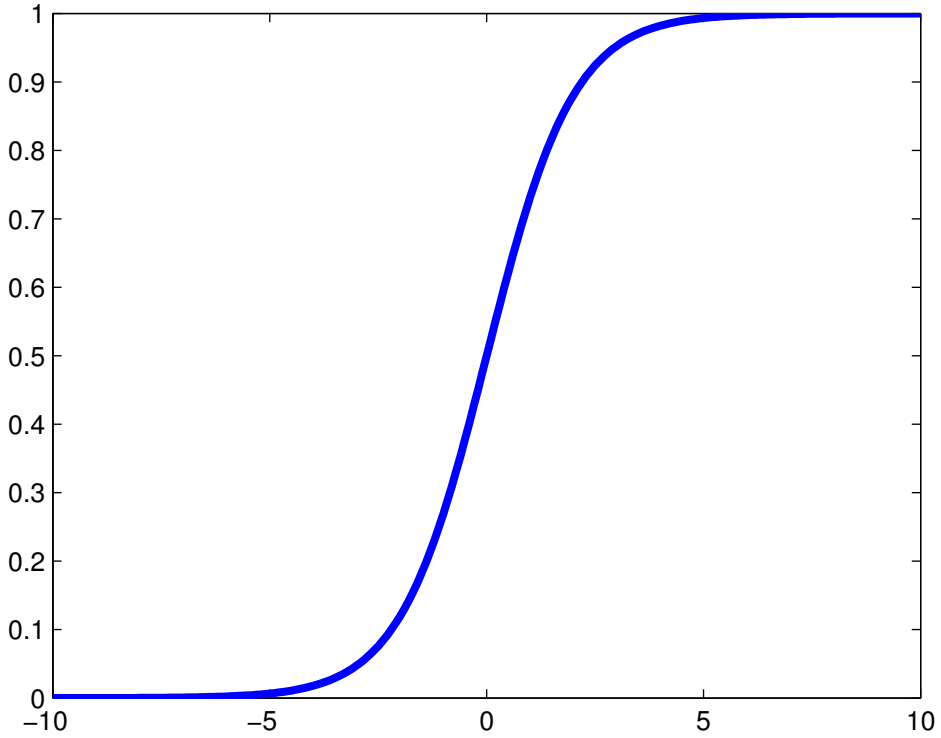


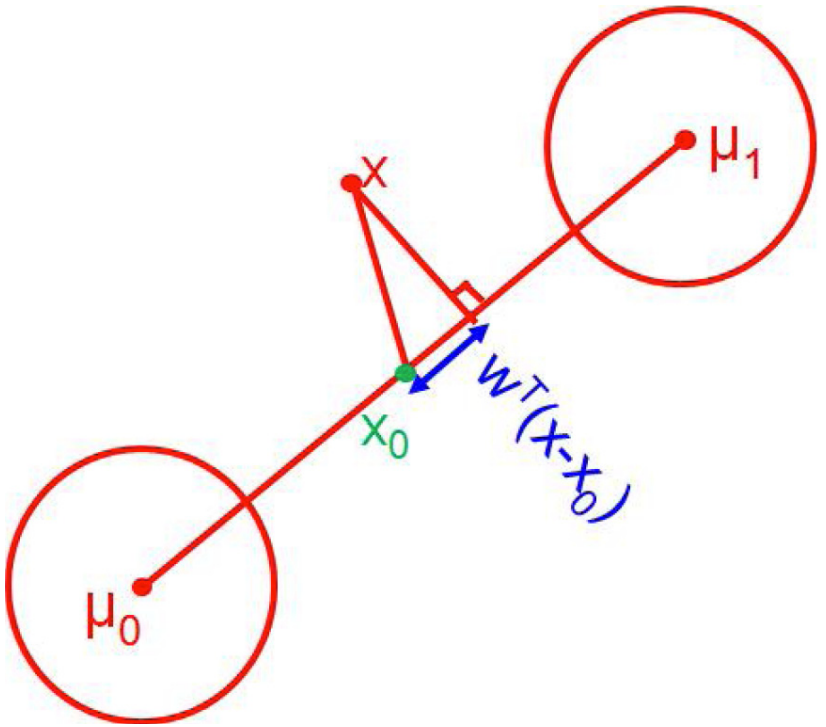
$$\mathcal{S}(\boldsymbol{\eta}/T)$$

$$\boldsymbol{\eta} = (3, 0, 1)$$

Binary Discriminant Analysis

Logistic Function





$$\text{sigm}(\eta) := \frac{1}{1+\exp(-\eta)} = \frac{e^{\eta}}{e^{\eta}+1}$$

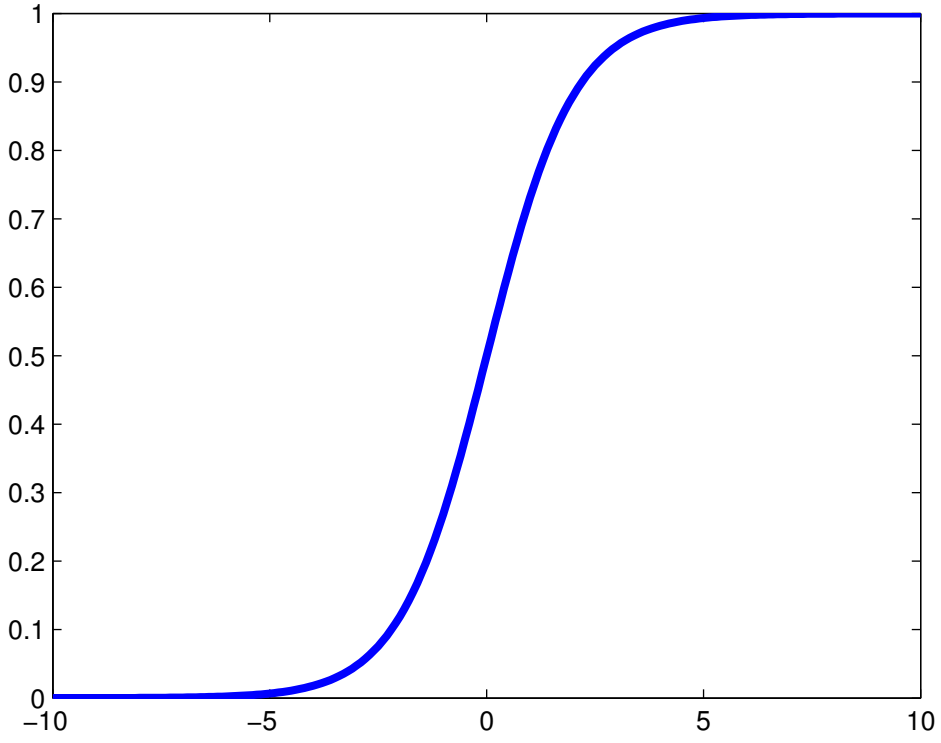
$$p(y=1|\mathbf{x},\boldsymbol{\theta}) = \sigma(\mathbf{w}^T(\mathbf{x}-\mathbf{x}_0))$$

$$\mathbf{w} = \boldsymbol{\beta}_1 - \boldsymbol{\beta}_0 = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)$$

$$\mathbf{x}_0 = \frac{1}{2}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0) - (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)\frac{\log(\pi_1/\pi_0)}{(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)}$$

Logistic vs Probit Functions

Logistic Function



Scaled Probit Function