PATTERN RECOGNITION AND MACHINE LEARNING

CHAPTER 3: LINEAR MODELS FOR REGRESSION

Learning Objectives

1、How to achieve linear regression using basis functions?

2、What are the relationships between maximum likelihood and least squares, between maximum a posterior and regularization, and among expected loss, bias, variance, and noise?

3、What are the common regularization methods for regression?

4、How to achieve Bayesian linear regression?

5、What is the kernel for regression?

6、How to choose the model complexity?

7、What are the evidence approximation and maximization?

Bayesian Machine Learning

Process of Machine Learning:

$$p(\theta|\textit{training data}, \textit{model}) \propto p(\textit{training data} \mid \textit{model}, \theta)\, p_0(\theta|\textit{model})$$

posterior likelihood prior

Process of Prediction:

$$p(\textit{testing data} \mid \textit{training data}, \textit{model}) = \int p\,(\textit{testing data} \mid \textit{model}, \theta)\, p(\theta \mid \textit{training data}, \textit{model})\, d\theta$$

Process of Model Evaluation:

For super-parameter tuning

$$p(\textit{ training data}, \mid \textit{model}) = \int p\,(\textit{training data} \mid \textit{model}, \theta)\, p_0(\theta \mid \textit{model})\, d\theta$$

Bayesian Learning for LGS

Given $$y = Ax + v$$

$$p(x) = \mathcal{N}(x|\mu, \Sigma) \quad p(v) = \mathcal{N}(v|0, Q)$$

$$x = m + u$$

$$p(x|y) = \mathcal{N}(x|m, L) \quad p(u) = \mathcal{N}(u|0, L)$$

we have

$$\begin{cases} L^{-1} = A^T Q^{-1} A + \Sigma^{-1} \\ m = L\{A^T Q^{-1} y + \Sigma^{-1} \mu\} \end{cases}$$

Bayesian Prediction for LGS

Given
$$y = Ax + v$$
$$p(x) = \mathcal{N}(x|\mu, \Sigma) \quad p(v) = \mathcal{N}(v|0, Q)$$
$$x = m + u$$
$$p(x|y) = \mathcal{N}(x|m, L) \quad p(u) = \mathcal{N}(u|0, L)$$

We have
$$p(y|x) = \mathcal{N}(y|Ax, Q)$$
$$p(y') = \int p(y'|x)p(x|y)dx = \mathcal{N}(y'|Am, ALA^T + Q)$$

Bayesian Model Evaluation for LGS

Given $$y = Ax + v$$

$$p(x) = \mathcal{N}(x|\mu, \Sigma) \quad p(v) = \mathcal{N}(v|0, Q)$$

we have

$$p(y|x) = \mathcal{N}(y|Ax, Q)$$

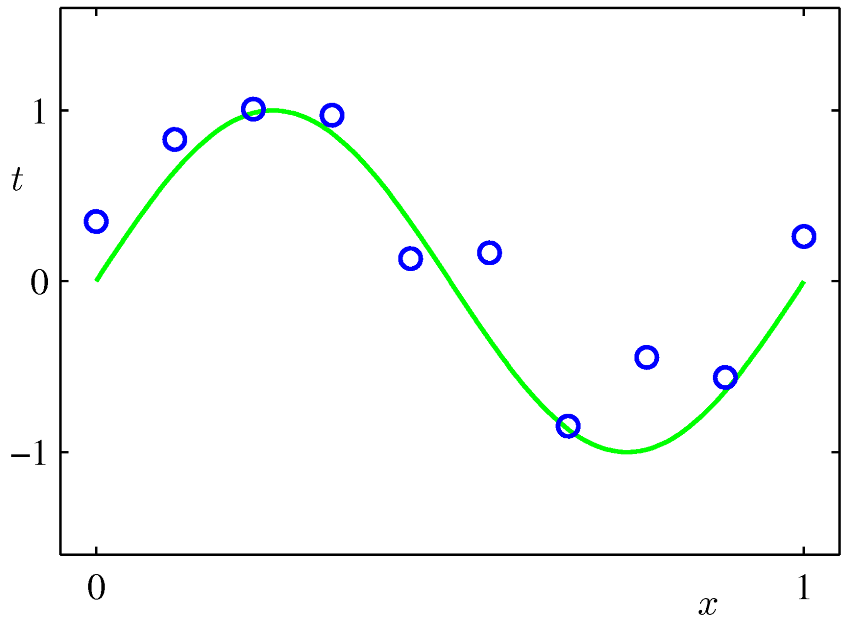
$$p(y) = \int p(y|x)p(x)dx = \mathcal{N}(y|A\mu, A\Sigma A^T + Q)$$

Outlines

- Linear Basis Function Models
- Maximum Likelihood and Least Squares
- Bias Variance Decomposition
- Bayesian Linear Regression
- Predictive Distribution
- Bayesian Model Comparison
- Evidence Approximation and Maximization

Linear Basis Function Models (1)

Example: Polynomial Curve Fitting



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

Linear Basis Function Models (2)

- Generally

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x})$$

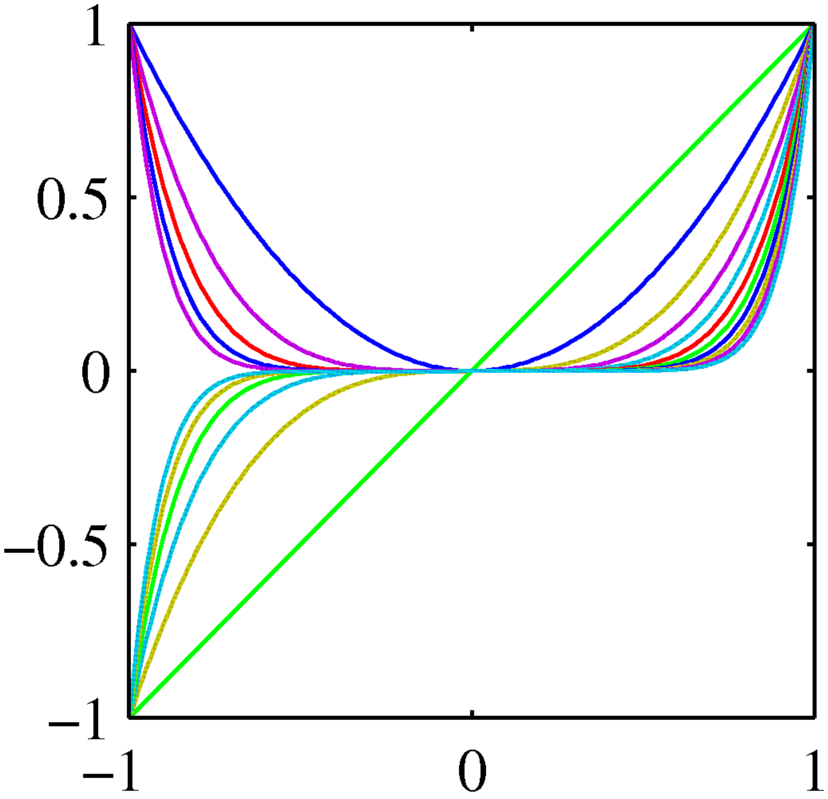
  where $\phi_j(\mathrm{x})$ are known as *basis functions*.
- Typically, $\phi_0(\mathrm{x}) = 1$, so that w_0 acts as a bias.
- In the simplest case, we use linear basis functions : $\phi_d(\mathrm{x}) = x_d$.

Linear Basis Function Models (3)

Polynomial basis functions:

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

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

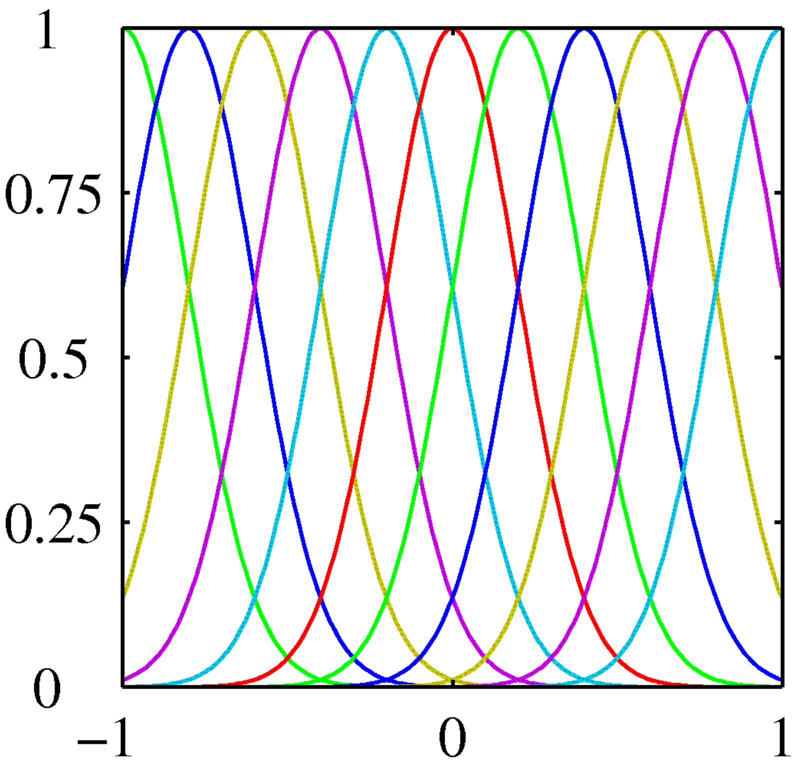


Linear Basis Function Models (4)

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. μ_j and s control location and scale (width).



Linear Basis Function Models (5)

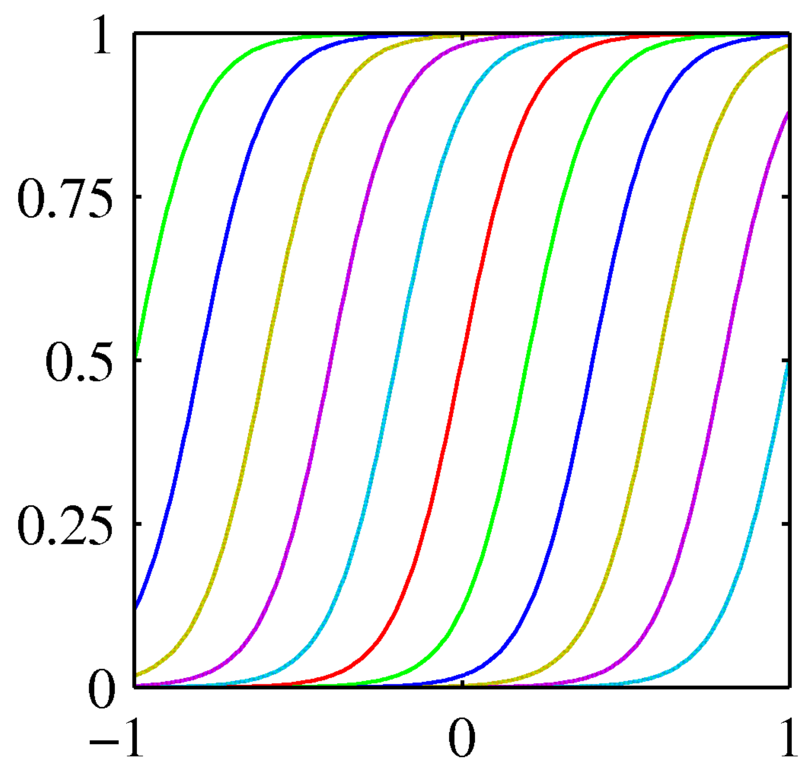
Sigmoidal basis functions:

$$\phi_j(x) = \sigma\left(\frac{x - \mu_j}{s}\right)$$

where

$$\sigma(a) = \frac{1}{1 + \exp(-a)}.$$

Also these are local; a small change in x only affect nearby basis functions. μ_j and s control location and scale (slope).



Outlines

- Linear Basis Function Models
- Maximum Likelihood and Least Squares
- Bias Variance Decomposition
- Bayesian Linear Regression
- Predictive Distribution
- Bayesian Model Comparison
- Evidence Approximation and Maximization

Maximum Likelihood and Least Squares (1)

- Assume observations from a deterministic function with added Gaussian noise:

$$t = y(\mathbf{x}, \mathbf{w}) + \epsilon \qquad \text{where} \qquad p(\epsilon|\beta) = \mathcal{N}(\epsilon|0, \beta^{-1})$$

which is the same as saying,

$$p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1}).$$

- Given observed inputs, $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, and targets, $\mathbf{t} = [t_1, \ldots, t_N]^{\mathrm{T}}$, we obtain the likelihood function

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

Maximum Likelihood and Least Squares (2)

Taking the logarithm, we get

$$\begin{aligned}\ln p(\mathbf{t}|\mathbf{w},\beta) &= \sum_{n=1}^{N} \ln \mathcal{N}(t_n|\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n),\beta^{-1}) \\ &= \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi) - \beta E_D(\mathbf{w})\end{aligned}$$

where

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

is the sum-of-squares error.

Maximum Likelihood and Least Squares (3)

Computing the gradient and setting it to zero yields

$$\nabla_{\mathbf{w}} \ln p(\mathbf{t}|\mathbf{w}, \beta) = \beta \sum_{n=1}^{N} \left\{ t_n - \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_n) \right\} \boldsymbol{\phi}(\mathbf{x}_n)^{\mathrm{T}} = \mathbf{0}.$$

Solving for w, we get

$$\mathbf{w}_{\mathrm{ML}} = \underbrace{\left(\boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\Phi} \right)^{-1} \boldsymbol{\Phi}^{\mathrm{T}}}_{\text{The Moore-Penrose pseudo-inverse, } \boldsymbol{\Phi}^{\dagger}} \mathbf{t}$$

The Moore-Penrose pseudo-inverse, $\boldsymbol{\Phi}^{\dagger}$.

Roger Penrose
2020 Nobel Prize
Laurate in Physics

where

$$\boldsymbol{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}.$$

Geometry of Least Squares

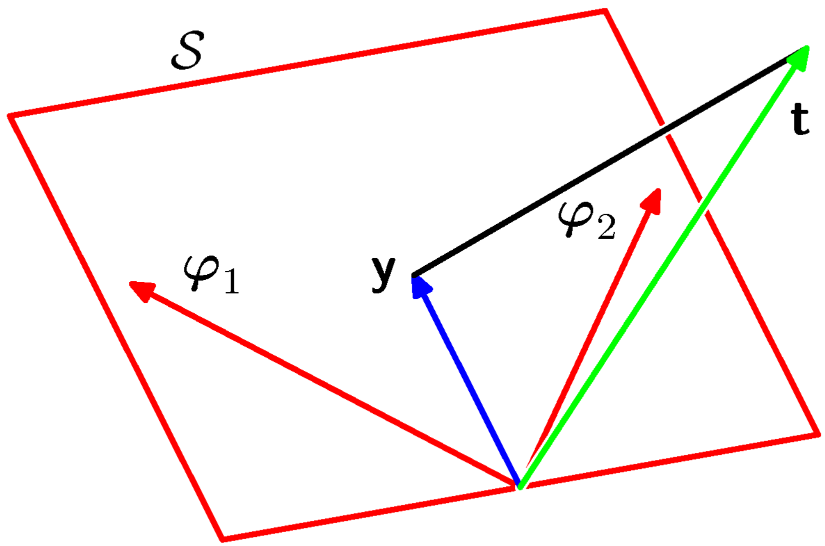
Consider

$$\mathbf{y} = \mathbf{\Phi}\mathbf{w}_{\mathrm{ML}} = [\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_M]\,\mathbf{w}_{\mathrm{ML}}.$$

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

N-dimensional

M-dimensional



S is spanned by $\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_M$.

$\mathbf{w}_{\mathrm{ML}}$ minimizes the distance between **t** and its orthogonal projection on S, i.e. **y**.

Sequential Learning

- Data items considered one at a time (a.k.a. online learning); use stochastic (sequential) gradient descent:

$$
\begin{aligned}
\mathbf{w}^{(\tau+1)} &= \mathbf{w}^{(\tau)} - \eta \nabla E_n \\
&= \mathbf{w}^{(\tau)} + \eta (t_n - \mathbf{w}^{(\tau)\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_n)) \boldsymbol{\phi}(\mathbf{x}_n).
\end{aligned}
$$

- This is known as the *least-mean-squares (LMS) algorithm*. Issue: how to choose η?

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

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

which is minimized by

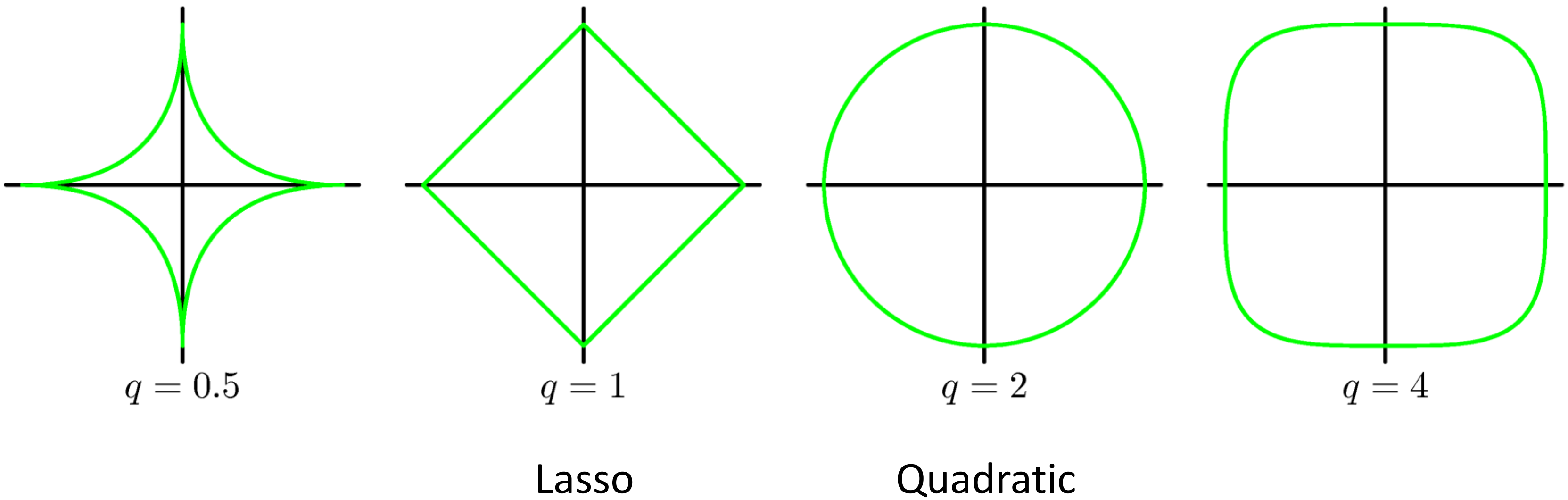
$$\mathbf{w} = \left(\lambda \mathbf{I} + \mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{t}.$$

λ is called the regularization coefficient.

Regularized Least Squares (2)

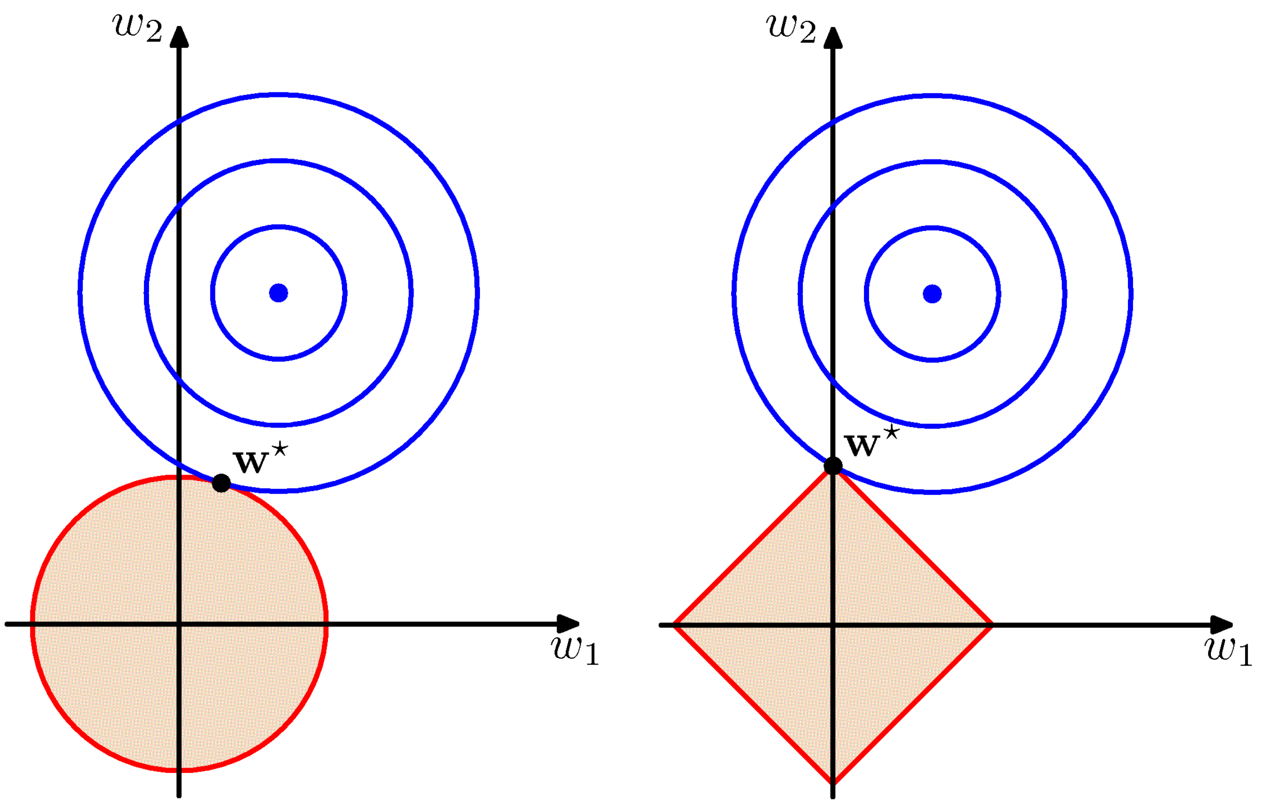
With a more general regularizer, we have

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



Regularized Least Squares (3)

Lasso tends to generate sparser solutions than a quadratic regularizer.



Multiple Outputs (1)

Analogously to the single output case we have:

$$\begin{aligned} p(\mathbf{t}|\mathbf{x}, \mathbf{W}, \beta) &= \mathcal{N}(\mathbf{t}|\mathbf{y}(\mathbf{W}, \mathbf{x}), \beta^{-1}\mathbf{I}) \\ &= \mathcal{N}(\mathbf{t}|\mathbf{W}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}), \beta^{-1}\mathbf{I}). \end{aligned}$$

Given observed inputs, $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, and targets, $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_N]^{\mathrm{T}}$, we obtain the log likelihood function

$$\begin{aligned} \ln p(\mathbf{T}|\mathbf{X}, \mathbf{W}, \beta) &= \sum_{n=1}^{N} \ln \mathcal{N}(\mathbf{t}_n|\mathbf{W}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}\mathbf{I}) \\ &= \frac{NK}{2} \ln\left(\frac{\beta}{2\pi}\right) - \frac{\beta}{2} \sum_{n=1}^{N} \left\| \mathbf{t}_n - \mathbf{W}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) \right\|^2. \end{aligned}$$

Multiple Outputs (2)

- Maximizing with respect to **W**, we obtain

$$\mathbf{W}_{\mathrm{ML}} = \left(\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{T}.$$

- If we consider a single target variable, $\mathbf{t}_{\mathrm{k}}$, we see that

$$\mathbf{w}_k = \left(\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{t}_k = \mathbf{\Phi}^{\dagger}\mathbf{t}_k$$

where $\mathbf{t}_k = [t_{1k}, \ldots, t_{Nk}]^{\mathrm{T}}$, which is identical with the single output case.

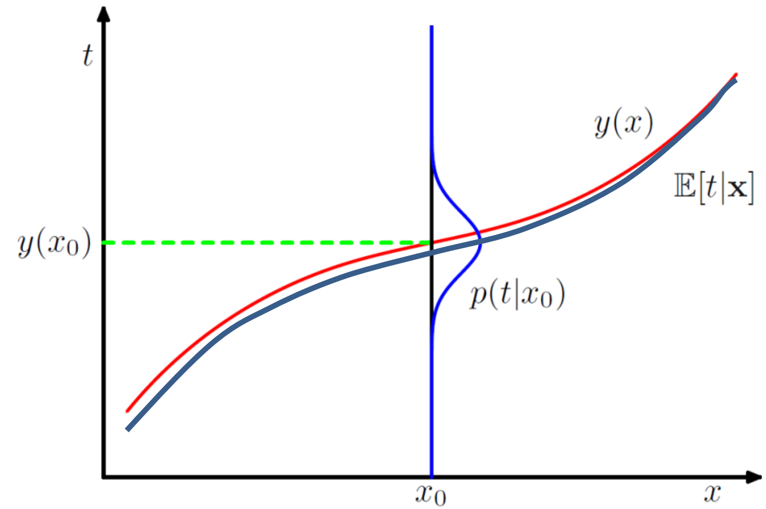
Outlines

- Linear Basis Function Models
- Maximum Likelihood and Least Squares
- Bias Variance Decomposition
- Bayesian Linear Regression
- Predictive Distribution
- Bayesian Model Comparison
- Evidence Approximation and Maximization

The Expected Squared Loss Function

predictor data

$$\mathbb{E}[L] = \iint \{y(\mathbf{x}) - t\}^2 p(\mathbf{x}, t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t$$



ground truth: optimal predictor

$$\{y(\mathbf{x}) - t\}^2 = \{y(\mathbf{x}) - \mathbb{E}[t|\mathbf{x}] + \mathbb{E}[t|\mathbf{x}] - t\}^2$$

$$= \{y(\mathbf{x}) - \mathbb{E}[t|\mathbf{x}]\}^2 + \underbrace{2\{y(\mathbf{x}) - \mathbb{E}[t|\mathbf{x}]\}\{\mathbb{E}[t|\mathbf{x}] - t\}}_{0} + \{\mathbb{E}[t|\mathbf{x}] - t\}^2$$

$$\mathbb{E}[L] = \int \{y(\mathbf{x}) - \mathbb{E}[t|\mathbf{x}]\}^2 p(\mathbf{x})\,\mathrm{d}\mathbf{x} + \int \mathrm{var}\,[t|\mathbf{x}]\,p(\mathbf{x})\,\mathrm{d}\mathbf{x}$$

predictor noise

https://stats.stackexchange.com/questions/228561/loss-functions-for-regression-proof

The Bias-Variance Decomposition (1)

- Recall the *expected squared loss*,

$$\mathbb{E}[L] = \int \{y(\mathbf{x}) - h(\mathbf{x})\}^2 p(\mathbf{x})\,\mathrm{d}\mathbf{x} + \iint \{h(\mathbf{x}) - t\}^2 p(\mathbf{x}, t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t$$

  where

$$h(\mathbf{x}) = \mathbb{E}[t|\mathbf{x}] = \int t p(t|\mathbf{x})\,\mathrm{d}t.$$

- The second term of $\mathbb{E}[L]$ corresponds to the noise inherent in the random variable t.

- What about the first term?

The Bias-Variance Decomposition (2)

- Suppose we were given multiple data sets, each of size N. Any particular data set, $\mathcal{D}$, will give a particular function $y(\mathbf{x};\mathcal{D})$. We then have

$$
\begin{aligned}
&\{y(\mathbf{x};\mathcal{D}) - h(\mathbf{x})\}^2 \\
&\quad = \quad \{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] + \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2 \\
&\quad = \quad \{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}^2 + \{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2 \\
&\qquad\qquad +2\{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}\{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}.
\end{aligned}
$$

The Bias-Variance Decomposition (3)

- Taking the expectation over $\mathbb{D}$ yields

$$
\begin{aligned}
&\mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x};\mathcal{D}) - h(\mathbf{x})\}^2\right] \\
&\quad = \underbrace{\{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2}_{(\text{bias})^2} + \underbrace{\mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}^2\right]}_{\text{variance}}.
\end{aligned}
$$

The Bias-Variance Decomposition (4)

- Thus we can write

$$\text{expected loss} = (\text{bias})^2 + \text{variance} + \text{noise}$$

where

Model:

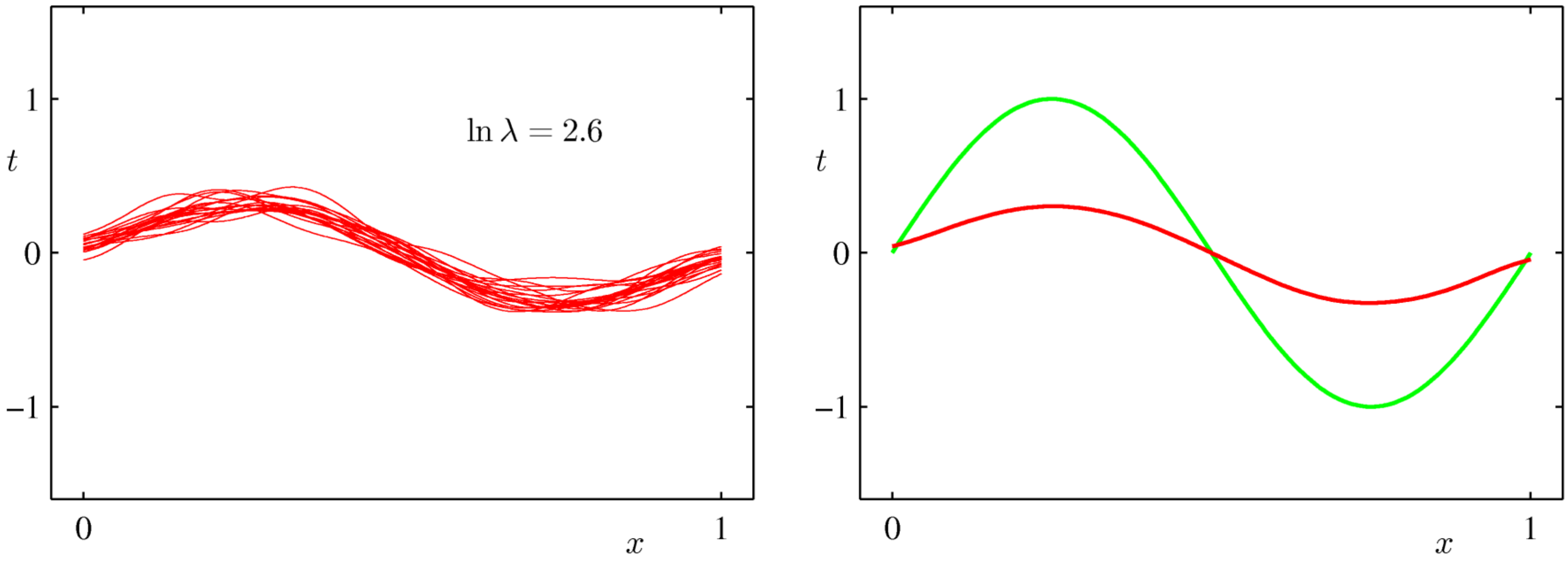
Model:

Data:

The picture can't be displayed.

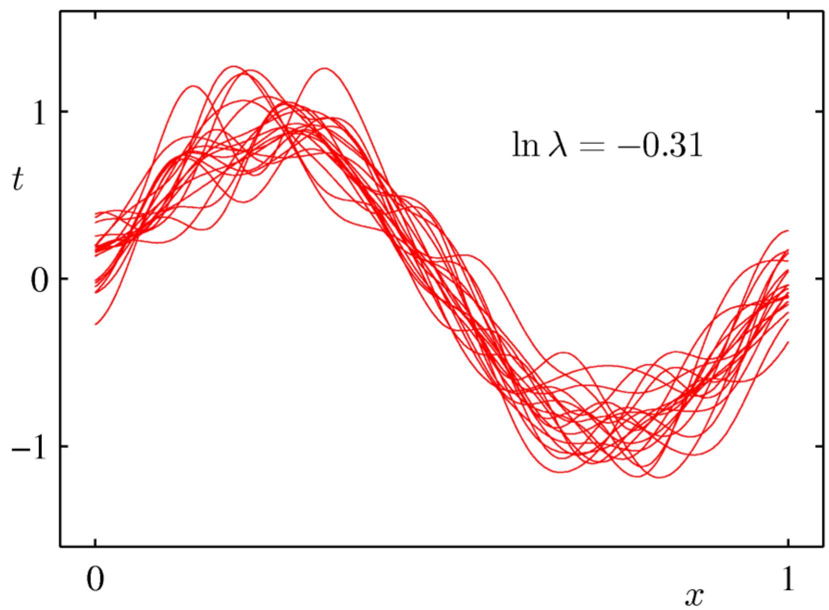
The Bias-Variance Decomposition (5)

- Example: 25 data sets from the sinusoidal, varying the degree of regularization, λ.



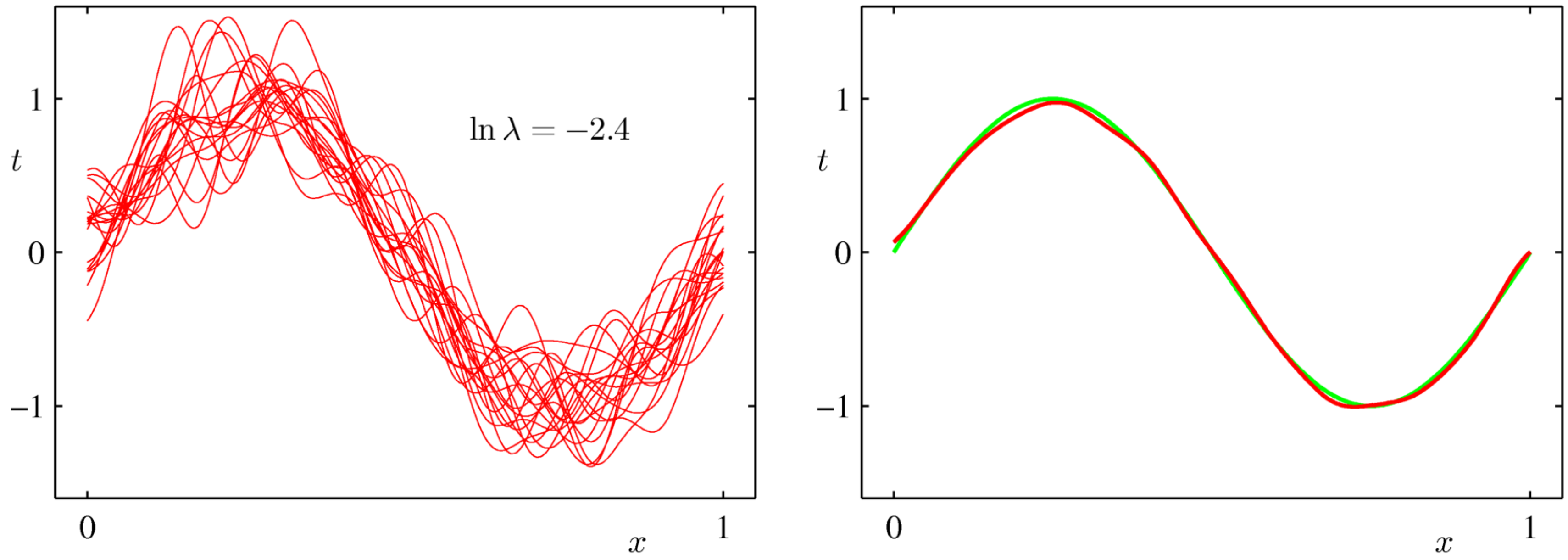
The Bias-Variance Decomposition (6)

- Example: 25 data sets from the sinusoidal, varying the degree of regularization, λ.



The Bias-Variance Decomposition (7)

- Example: 25 data sets from the sinusoidal, varying the degree of regularization, λ.



The Bias-Variance Trade-off

From these plots, we note that an over-regularized model (large λ) will have a high bias, while an under-regularized model (small λ) will have a high variance.