Advanced Machine Learning
Lecture 3

Linear Regression II

02.11.2015

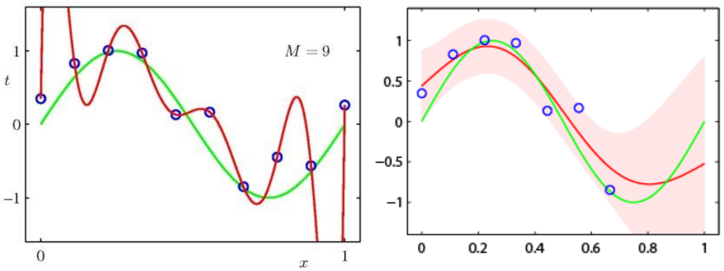
Bastian Leibe
RWTH Aachen
http://www.vision.rwth-aachen.de/

leibe@vision.rwth-aachen.de

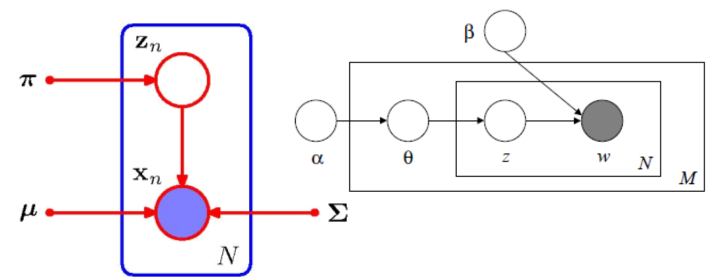
This Lecture: *Advanced Machine Learning*

- Regression Approaches
  - Linear Regression
  - Regularization (Ridge, Lasso)
  - Support Vector Regression
  - Gaussian Processes

$$f : \mathcal{X} \rightarrow \mathbb{R}$$



- Learning with Latent Variables
  - EM and Generalizations
  - Dirichlet Processes



- Structured Output Learning
  - Large-margin Learning

$$f : \mathcal{X} \rightarrow \mathcal{Y}$$

Topics of This Lecture

- Recap: Probabilistic View on Regression
- Properties of Linear Regression
  - Loss functions for regression
  - Basis functions
  - Multiple Outputs
  - Sequential Estimation
- Regularization revisited
  - Regularized Least-squares
  - The Lasso
  - Discussion
- Bias-Variance Decomposition

Recap: Probabilistic Regression

- **First assumption:**
  - Our target function values t are generated by adding noise to the ideal function estimate:

$$t = y(\mathbf{x}, \mathbf{w}) + \epsilon$$

Target function value

Noise

Regression function

Input value

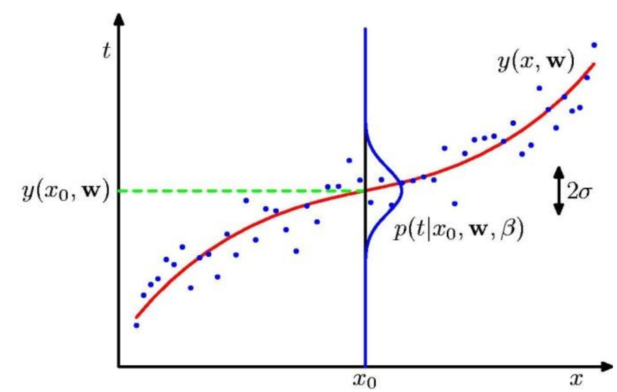
Weights or parameters

- **Second assumption:**
  - The noise is Gaussian distributed.

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

Mean

Variance (β precision)



Recap: Probabilistic Regression

- Given
  - Training data points: $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$
  - Associated function values: $\mathbf{t} = [t_1, \ldots, t_n]^T$

- Conditional likelihood (assuming i.i.d. data)

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

$\Rightarrow$ Maximize w.r.t. $\mathbf{w}$, β

Generalized linear regression function

Recap: Maximum Likelihood Regression

$$\nabla_{\mathbf{w}} \log p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) \;=\; -\beta \sum_{n=1}^{N} (t_n - \mathbf{w}^T \phi(\mathbf{x}_n))\phi(\mathbf{x}_n)$$

- **Setting the gradient to zero:**

$$0 \;=\; -\beta \sum_{n=1}^{N} (t_n - \mathbf{w}^T \phi(\mathbf{x}_n))\phi(\mathbf{x}_n)$$

$$\Leftrightarrow \quad \sum_{n=1}^{N} t_n \phi(\mathbf{x}_n) = \left[\sum_{n=1}^{N} \phi(\mathbf{x}_n)\phi(\mathbf{x}_n)^T \right] \mathbf{w}$$

$$\Leftrightarrow \quad \mathbf{\Phi t} = \mathbf{\Phi \Phi}^T \mathbf{w} \qquad\qquad \mathbf{\Phi} = [\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_n)]$$

$$\Leftrightarrow \quad \mathbf{w}_{\mathrm{ML}} = (\mathbf{\Phi \Phi}^T)^{-1} \mathbf{\Phi t}$$

Same as in least-squares regression!

$\Rightarrow$ ***Least-squares regression is equivalent to Maximum Likelihood under the assumption of Gaussian noise.***

Recap: Role of the Precision Parameter

- **Also use ML to determine the precision parameter β:**

$$\log p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = -\frac{\beta}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^T\phi(\mathbf{x}_n)\}^2 + \frac{N}{2}\log\beta - \frac{N}{2}\log(2\pi)$$

- **Gradient w.r.t. β:**

$$\nabla_\beta \log p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = -\frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^T\phi(\mathbf{x}_n)\}^2 + \frac{N}{2}\frac{1}{\beta}$$

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

$\Rightarrow$ *The inverse of the noise precision is given by the residual variance of the target values around the regression function.*

Recap: Predictive Distribution

- Having determined the parameters $\mathbf{w}$ and β, we can now make predictions for new values of $\mathbf{x}$.

$$p(t|\mathbf{X}, \mathbf{w}_{\mathrm{ML}}, \beta_{\mathrm{ML}}) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}_{\mathrm{ML}}), \beta_{\mathrm{ML}}^{-1})$$

- This means
  - Rather than giving a point estimate, we can now also give an estimate of the estimation uncertainty.

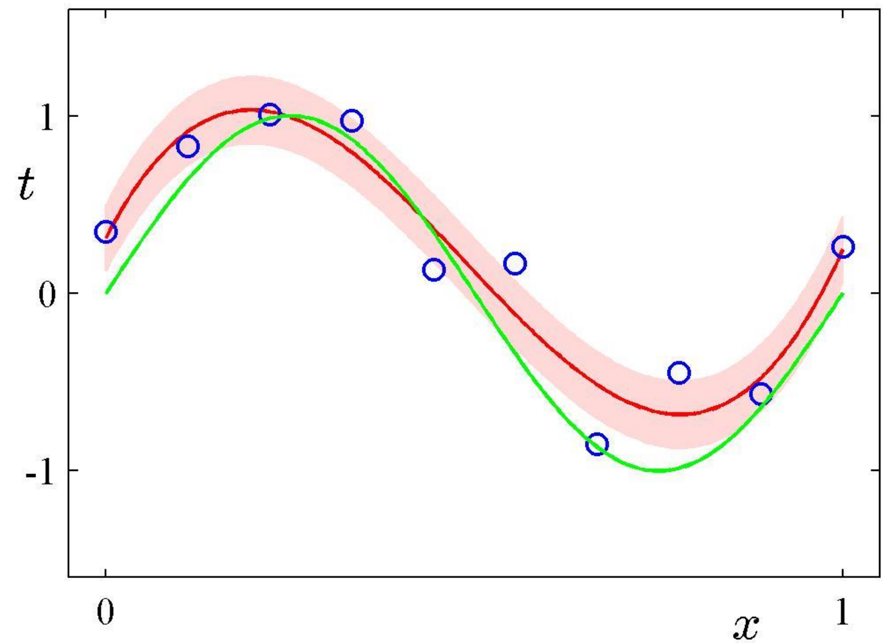


Image source: C.M. Bishop, 2006

Recap: Maximum-A-Posteriori Estimation

- Introduce a prior distribution over the coefficients $\mathbf{w}$.
  - For simplicity, assume a zero-mean Gaussian distribution

$$p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I}) = \left(\frac{\alpha}{2\pi}\right)^{(M+1)/2} \exp\left\{-\frac{\alpha}{2}\mathbf{w}^T\mathbf{w}\right\}$$

  - New hyperparameter α controls the distribution of model parameters.

- Express the posterior distribution over $\mathbf{w}$.
  - Using Bayes' theorem:

$$p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \beta, \alpha) \propto p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta)p(\mathbf{w}|\alpha)$$

  - We can now determine $\mathbf{w}$ by maximizing the posterior.
  - This technique is called maximum-a-posteriori (MAP).

Recap: MAP Solution

- **Minimize the negative logarithm**

$$-\log p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \beta, \alpha) \propto -\log p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) - \log p(\mathbf{w}|\alpha)$$

$$-\log p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \frac{\beta}{2} \sum_{n=1}^{N} \{y(\mathbf{x}_n, \mathbf{w}) - t_n\}^2 + \text{const}$$

$$-\log p(\mathbf{w}|\alpha) = \frac{\alpha}{2} \mathbf{w}^T \mathbf{w} + \text{const}$$

- **The MAP solution is therefore**

$$\arg\min_{\mathbf{w}} \frac{\beta}{2} \sum_{n=1}^{N} \{y(\mathbf{x}_n, \mathbf{w}) - t_n\}^2 + \frac{\alpha}{2} \mathbf{w}^T \mathbf{w}$$

$\Rightarrow$ ***Maximizing the posterior distribution is equivalent to minimizing the regularized sum-of-squares error*** ***(with*** $\lambda = \frac{\alpha}{\beta}$***).***

MAP Solution (2)

$$\nabla_{\mathbf{w}} \log p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \beta, \alpha) = -\beta \sum_{n=1}^{N} (t_n - \mathbf{w}^T \phi(\mathbf{x}_n))\phi(\mathbf{x}_n) + \alpha \mathbf{w}$$

- **Setting the gradient to zero:**

$$0 = -\beta \sum_{n=1}^{N} (t_n - \mathbf{w}^T \phi(\mathbf{x}_n))\phi(\mathbf{x}_n) + \alpha \mathbf{w}$$

$$\Leftrightarrow \sum_{n=1}^{N} t_n \phi(\mathbf{x}_n) = \left[\sum_{n=1}^{N} \phi(\mathbf{x}_n)\phi(\mathbf{x}_n)^T \right] \mathbf{w} + \frac{\alpha}{\beta} \mathbf{w}$$

$$\Leftrightarrow \boldsymbol{\Phi}\mathbf{t} = \left(\boldsymbol{\Phi}\boldsymbol{\Phi}^T + \frac{\alpha}{\beta}\mathbf{I} \right) \mathbf{w} \qquad \boldsymbol{\Phi} = [\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_n)]$$

$$\Leftrightarrow \mathbf{w}_{\mathrm{MAP}} = \left(\boldsymbol{\Phi}\boldsymbol{\Phi}^T + \frac{\alpha}{\beta}\mathbf{I} \right)^{-1} \boldsymbol{\Phi}\mathbf{t}$$

Effect of regularization:
Keeps the inverse well-conditioned

Recap: Bayesian Curve Fitting

- Given
  - Training data points: $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$
  - Associated function values: $\mathbf{t} = [t_1, \ldots, t_n]^T$
  - Our goal is to predict the value of t for a new point $\mathbf{x}$.

- Evaluate the predictive distribution

$$p(t|x, \mathbf{X}, \mathbf{t}) = \int \underbrace{p(t|x, \mathbf{w})} \underbrace{p(\mathbf{w}|\mathbf{X}, \mathbf{t})} d\mathbf{w}$$

What we just computed for MAP

  - Noise distribution - again assume a Gaussian here

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

  - Assume that parameters α and β are fixed and known for now.

Bayesian Curve Fitting

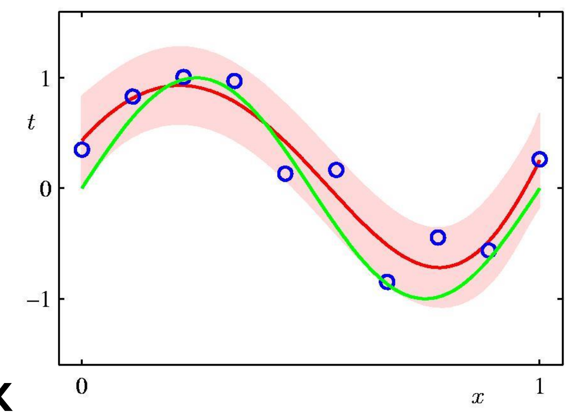
- Under those assumptions, the posterior distribution is a Gaussian and can be evaluated analytically:

$$p(t|x, \mathbf{X}, \mathbf{t}) = \mathcal{N}(t|m(x), s^2(x))$$

  - where the mean and variance are given by

$$m(x) = \beta\phi(x)^T\mathbf{S}\sum_{n=1}^{N}\phi(\mathbf{x}_n)t_n$$

$$s(x)^2 = \beta^{-1} + \phi(x)^T\mathbf{S}\phi(x)$$



  - and S is the regularized covariance matrix

$$\mathbf{S}^{-1} = \alpha\mathbf{I} + \beta\sum_{n=1}^{N}\phi(\mathbf{x}_n)\phi(\mathbf{x}_n)^T$$

Image source: C.M. Bishop, 2006

Analyzing the result

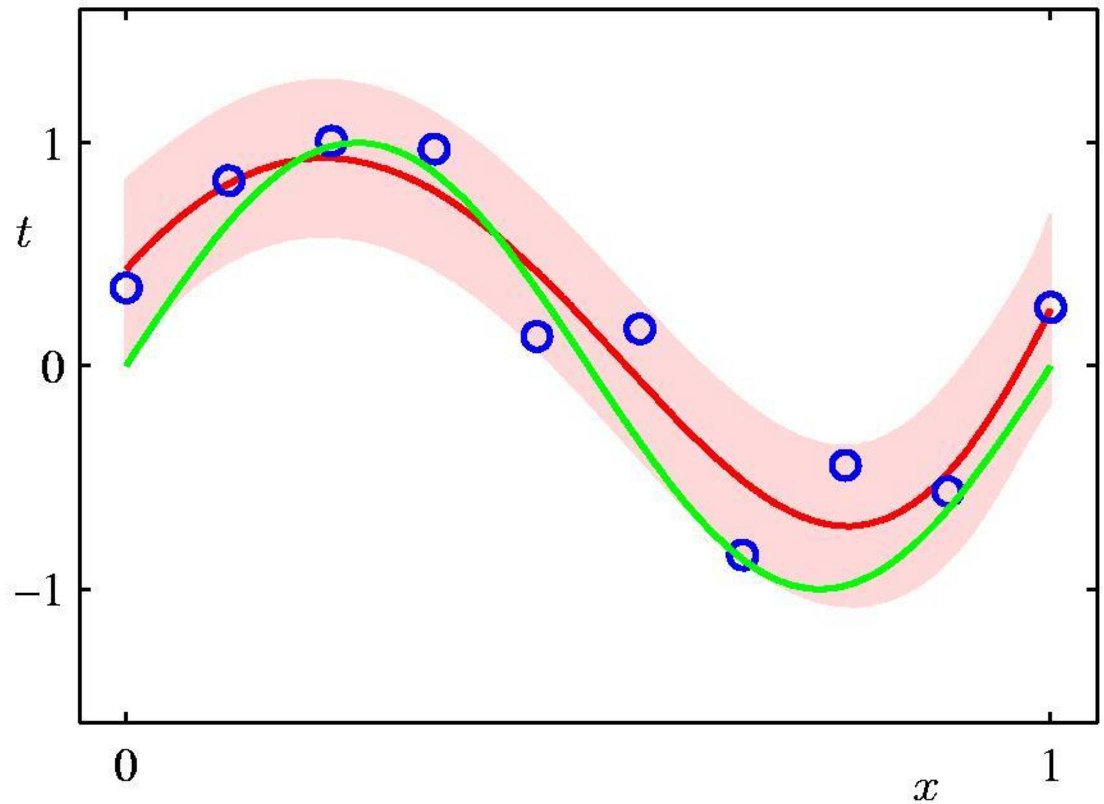
- **Analyzing the variance of the predictive distribution**

$$s(x)^2 = \underbrace{\beta^{-1}}_{} + \underbrace{\phi(x)^T \mathbf{S} \phi(x)}_{}$$

Uncertainty in the predicted value due to noise on the target variables (expressed already in ML)

Uncertainty in the parameters $\mathbf{w}$ (consequence of Bayesian treatment)

Bayesian Predictive Distribution



- **Important difference to previous example**
  - **Uncertainty may vary with test point x!**

Image source: C.M. Bishop, 2006

Discussion

- We now have a better understanding of regression
  - Least-squares regression: Assumption of Gaussian noise

  $\Rightarrow$ We can now also plug in different noise models and explore how they affect the error function.

  - L2 regularization as a Gaussian prior on parameters $\mathbf{w}$.

  $\Rightarrow$ We can now also use different regularizers and explore what they mean.

  $\Rightarrow$ This lecture…

  - General formulation with basis functions $\phi(\mathbf{x})$.

  $\Rightarrow$ We can now also use different basis functions.

Discussion

- **General regression formulation**
  - In principle, we can perform regression in arbitrary spaces and with many different types of basis functions
  - However, there is a caveat… Can you see what it is?

- **Example: Polynomial curve fitting, $M = 3$**

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{i=1}^{D} w_i x_i + \sum_{i=1}^{D}\sum_{j=1}^{D} w_{ij} x_i x_j + \sum_{i=1}^{D}\sum_{j=1}^{D}\sum_{k=1}^{D} w_{ijk} x_i x_j x_k$$

$\Rightarrow$ Number of coefficients grows with D^M!

$\Rightarrow$ The approach becomes quickly unpractical for high dimensions.

- This is known as the curse of dimensionality.
- We will encounter some ways to deal with this later...

Topics of This Lecture

- Recap: Probabilistic View on Regression
- Properties of Linear Regression
  - Loss functions for regression
  - Basis functions
  - Multiple Outputs
  - Sequential Estimation
- Regularization revisited
  - Regularized Least-squares
  - The Lasso
  - Discussion
- Bias-Variance Decomposition

Loss Functions for Regression

- Given $p(y, \mathbf{x}, \mathbf{w}, \beta)$, how do we actually estimate a function value y_t for a new point $\mathbf{x}_t$?

- We need a loss function, just as in the classification case

$$\begin{array}{rccc} L: & \mathbb{R} \times \mathbb{R} & \rightarrow & \mathbb{R}^{+} \\ & (t_n, y(\mathbf{x}_n)) & \rightarrow & L(t_n, y(\mathbf{x}_n)) \end{array}$$

- Optimal prediction: Minimize the expected loss

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

Loss Functions for Regression

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

- **Simplest case**
  - **Squared loss:** $L(t, y(\mathbf{x})) = \{y(\mathbf{x}) - t\}^2$
  - **Expected loss**

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

$$\frac{\partial \mathbb{E}[L]}{\partial y(\mathbf{x})} = 2\int \{y(\mathbf{x}) - t\}\, p(\mathbf{x}, t)\mathrm{d}t \overset{!}{=} 0$$

$$\Leftrightarrow \int t p(\mathbf{x}, t)\mathrm{d}t = y(\mathbf{x}) \int p(\mathbf{x}, t)\mathrm{d}t$$

Loss Functions for Regression

$$\int t p(\mathbf{x}, t) \mathrm{d}t = y(\mathbf{x}) \int p(\mathbf{x}, t) \mathrm{d}t$$

$$\Leftrightarrow y(\mathbf{x}) = \int t \frac{p(\mathbf{x}, t)}{p(\mathbf{x})} \mathrm{d}t = \int t p(t|\mathbf{x}) \mathrm{d}t$$

$$\Leftrightarrow y(\mathbf{x}) = \mathbb{E}[t|\mathbf{x}]$$

- **Important result**
  - **Under Squared loss, the optimal regression function is the mean $\mathbb{E}\left[t\,|\,\mathbf{x}\right]$ of the posterior $p(t\,|\,\mathbf{x})$.**
  - **Also called mean prediction.**
  - **For our generalized linear regression function and square loss, we obtain as result**

$$y(\mathbf{x}) = \int t \mathcal{N}(t|\mathbf{w}^T \phi(\mathbf{x}), \beta^{-1}) \mathrm{d}t = \mathbf{w}^T \phi(\mathbf{x})$$

Visualization of Mean Prediction

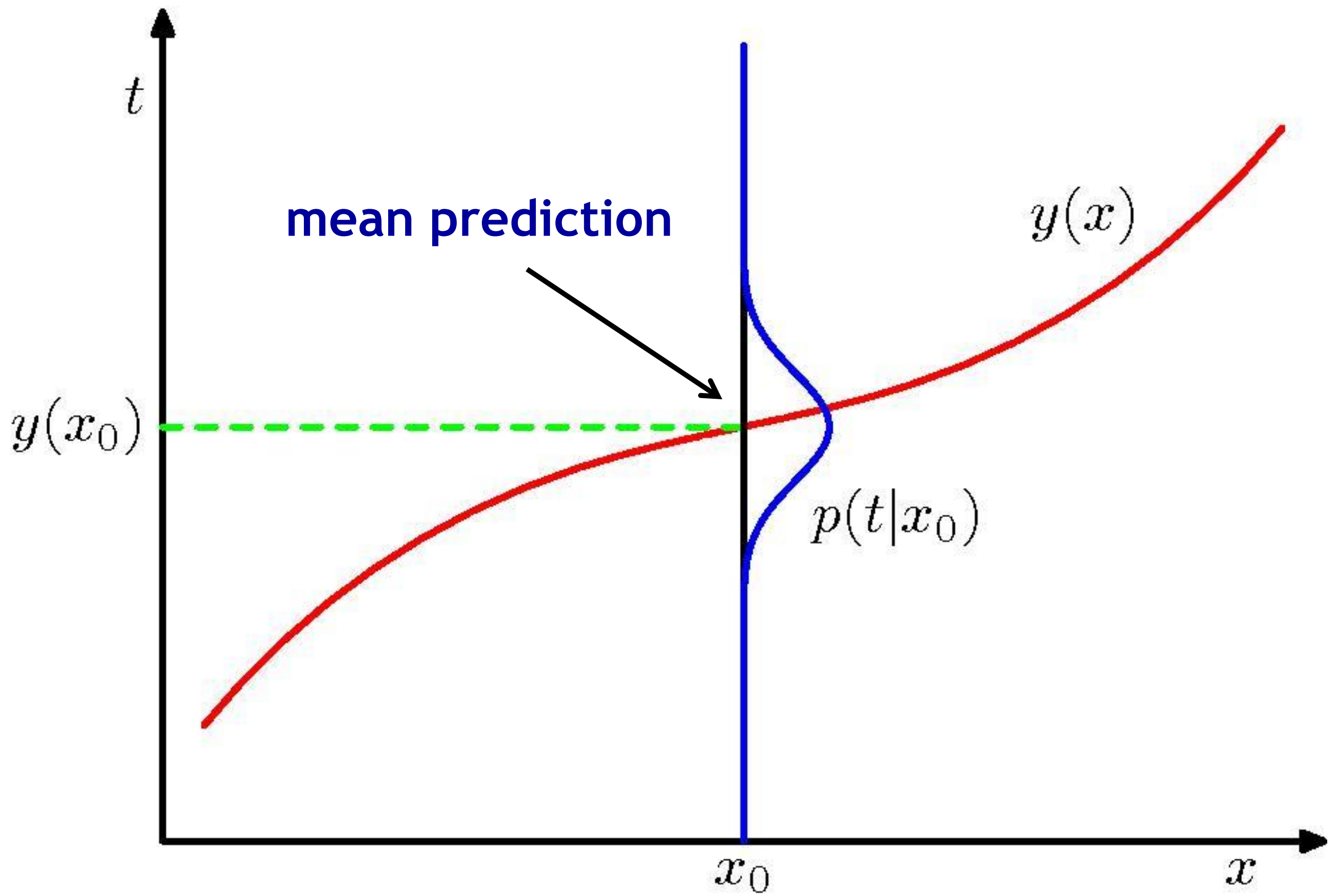


Image source: C.M. Bishop, 2006

Loss Functions for Regression

- **Different derivation: Expand the square term as follows**

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

- **Substituting into the loss function**
  - **The cross-term vanishes, and we end up with**

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

Optimal least-squares predictor given by the conditional mean

Intrinsic variability of target data ⇒ Irreducible minimum value of the loss function

Other Loss Functions

- The squared loss is not the only possible choice
  - Poor choice when conditional distribution $p(t\,|\,\mathbf{x})$ is multimodal.

- Simple generalization: Minkowski loss

$$L(t, y(\mathbf{x})) = |y(\mathbf{x}) - t|^q$$

  - Expectation

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

- Minimum of $\mathbb{E}[L_q]$ is given by
  - Conditional mean for $q = 2$,
  - Conditional median for $q = 1$,
  - Conditional mode for $q = 0$.

RWTH AACHEN UNIVERSITY

Minkowski Loss Functions

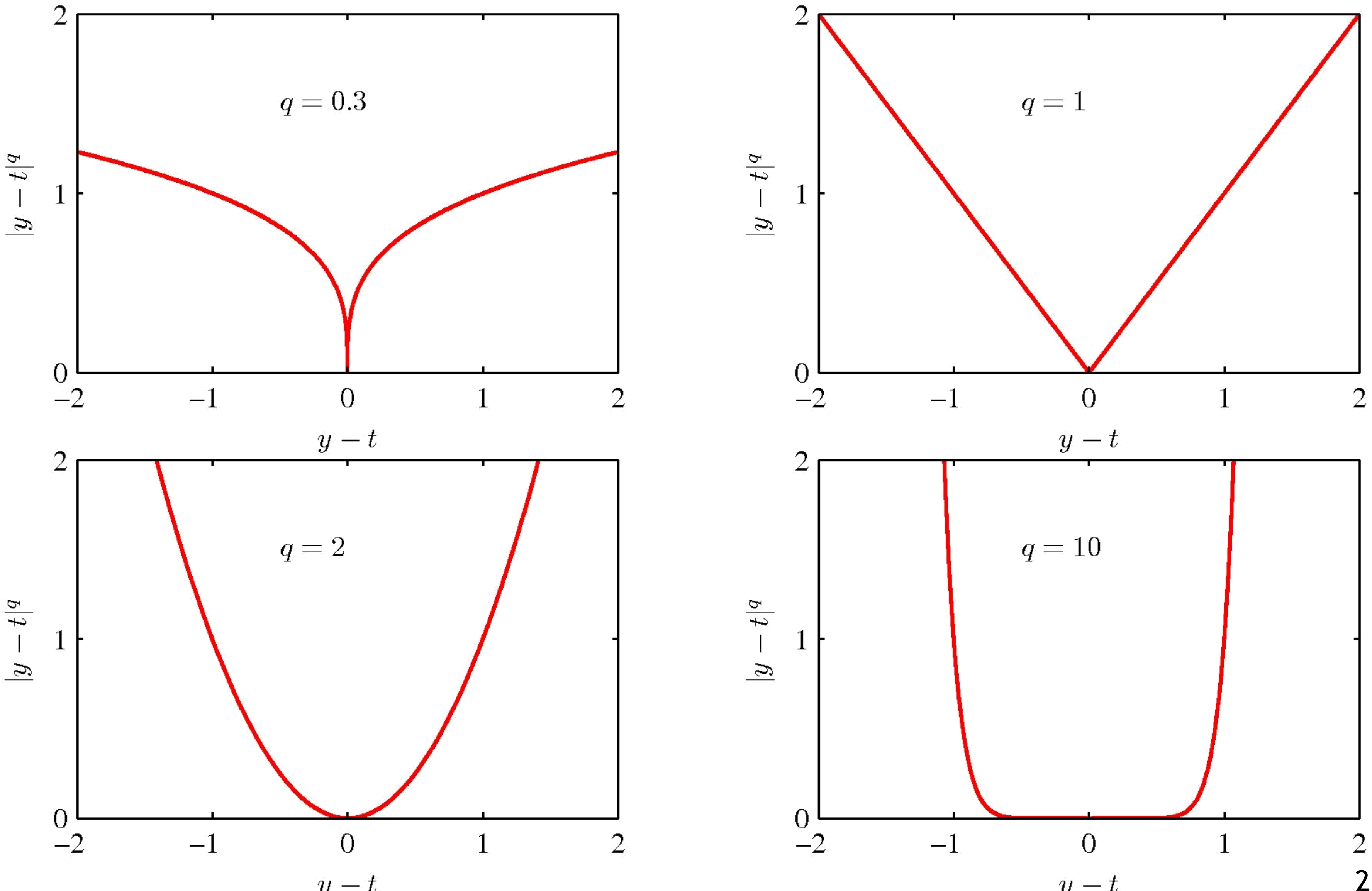


Image source: C.M. Bishop, 2006

Topics of This Lecture

- Recap: Probabilistic View on Regression
- Properties of Linear Regression
  - Loss functions for regression
  - Basis functions
  - Multiple Outputs
  - Sequential Estimation
- Regularization revisited
  - Regularized Least-squares
  - The Lasso
  - Discussion
- Bias-Variance Decomposition

Linear Basis Function Models

- Generally, we consider models of the following form

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

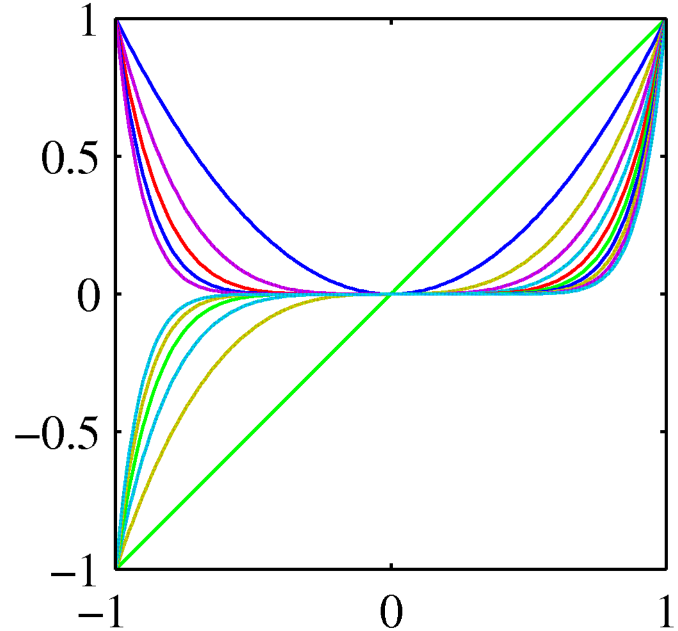
  - where $\phi_j(\mathbf{x})$ are known as *basis functions*.
  - Typically, $\phi_0(\mathbf{x}) = 1$, so that w_0 acts as a bias.
  - In the simplest case, we use linear basis functions: $\phi_d(\mathbf{x}) = x_d$.

- ***Let's take a look at some other possible basis functions...***

Linear Basis Function Models (2)

- **Polynomial basis functions**

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



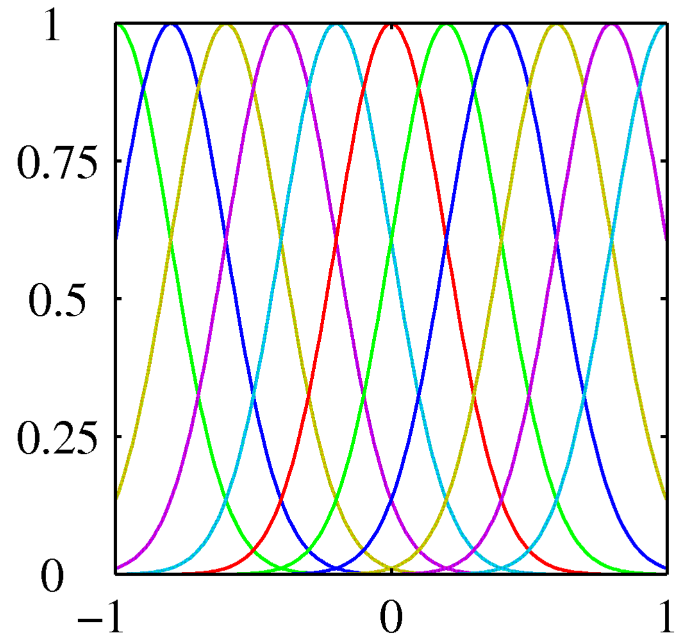
- **Properties**
  - Global

$\Rightarrow$ A small change in x affects all basis functions.

Linear Basis Function Models (3)

- **Gaussian basis functions**

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



- **Properties**
  - **Local**

  ⇒ **A small change in x affects only nearby basis functions.**
  - **μ_j and s control location and scale (width).**

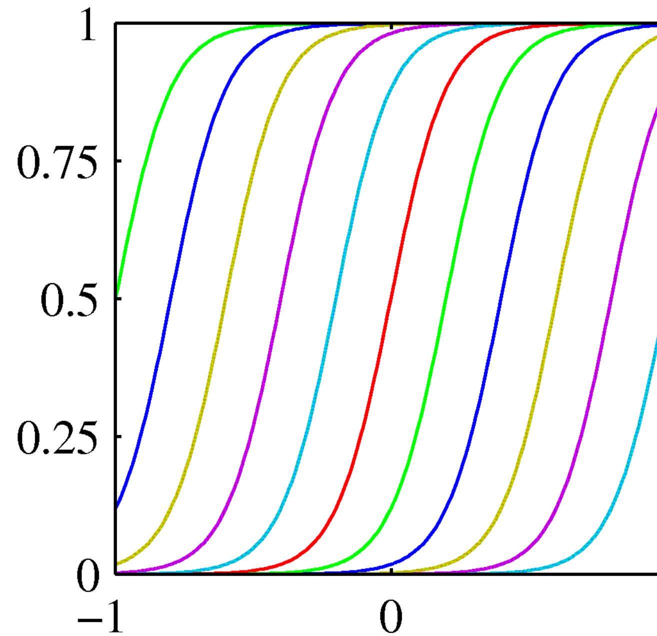
Linear Basis Function Models (4)

- **Sigmoid basis functions**

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

  - where

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



- **Properties**
  - Local

$\Rightarrow$ A small change in x affects only nearby basis functions.

  - μ_j and s control location and scale (slope).

Topics of This Lecture

- Recap: Probabilistic View on Regression
- Properties of Linear Regression
  - Loss functions for regression
  - Basis functions
  - Multiple Outputs
  - Sequential Estimation
- Regularization revisited
  - Regularized Least-squares
  - The Lasso
  - Discussion
- Bias-Variance Decomposition

Multiple Outputs

- **Multiple Output Formulation**
  - So far only considered the case of a single target variable t.
  - We may wish to predict $K > 1$ target variables in a vector $\mathbf{t}$.
  - We can write this in matrix form

$$\mathbf{y}(\mathbf{x}, \mathbf{W}) = \mathbf{W}^T \phi(\mathbf{x})$$

  - where

$$\mathbf{y} = [y_1, \ldots, y_K]^T$$

$$\phi(\mathbf{x}) = [1, \phi_1(\mathbf{x}), \cdots, \phi_{M-1}(\mathbf{x}),]^T$$

$$\mathbf{W} = \begin{bmatrix} w_{0,1} & \cdots & w_{0,K} \\ \vdots & \ddots & \vdots \\ w_{M-1,1} & \cdots & w_{M-1,K} \end{bmatrix}^T$$

Multiple Outputs (2)

- 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 (3)

- **Maximizing with respect to $\mathbf{W}$, 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, t_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.

Topics of This Lecture

- Recap: Probabilistic View on Regression
- Properties of Linear Regression
  - Loss functions for regression
  - Basis functions
  - Multiple Outputs
  - Sequential Estimation
- Regularization revisited
  - Regularized Least-squares
  - The Lasso
  - Discussion
- Bias-Variance Decomposition

Sequential Learning

- Up to now, we have mainly considered batch methods
  - All data was used at the same time
  - Instead, we can also consider data items one at a time (a.k.a. online learning)
- 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 the learning rate η?

Topics of This Lecture

- Recap: Probabilistic View on Regression
- Properties of Linear Regression
  - Loss functions for regression
  - Basis functions
  - Multiple Outputs
  - Sequential Estimation
- **Regularization revisited**
  - Regularized Least-squares
  - The Lasso
  - Discussion
- Bias-Variance Decomposition

Regularization Revisited

- 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

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

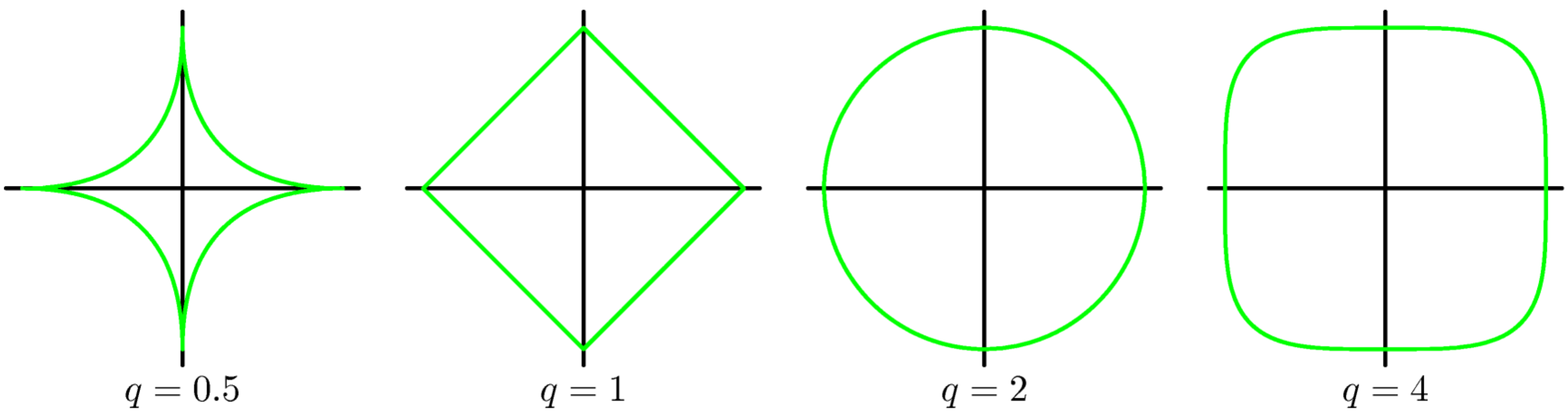
λ is called the **regularization coefficient.**

Regularized Least-Squares

- Let's look at more general regularizers

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

- "L_q norms"



"Lasso"

"Ridge Regression"

Recall: Lagrange Multipliers

Regularized Least-Squares

- We want to minimize

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

- This is equivalent to minimizing

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

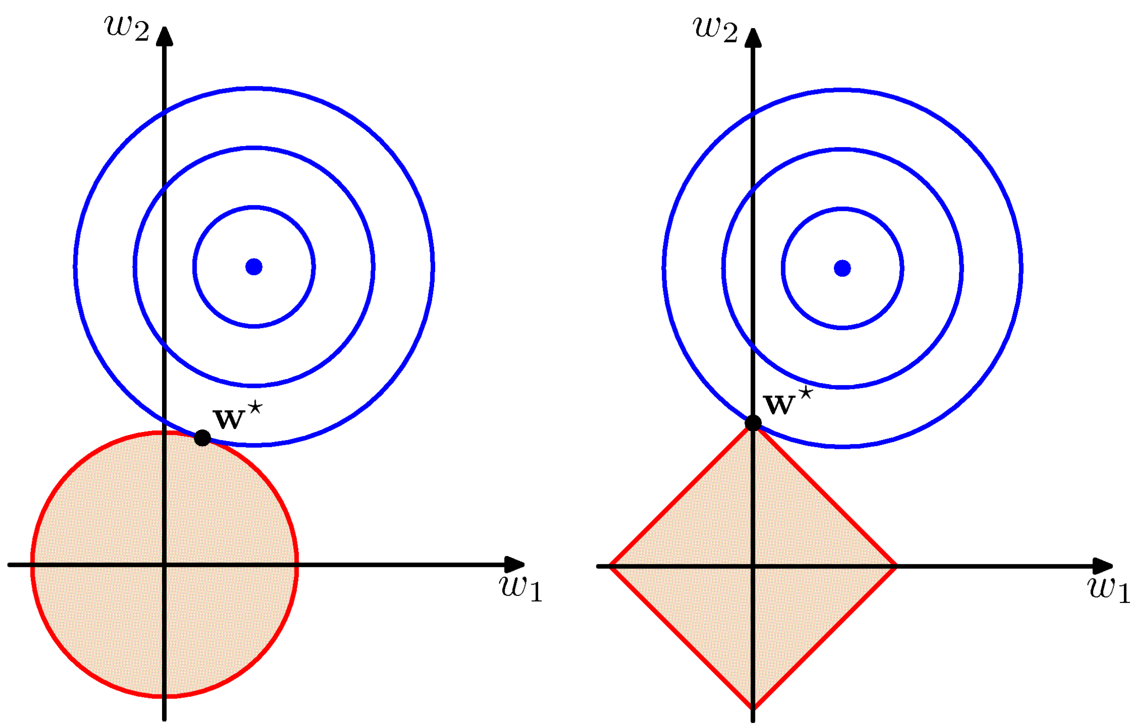
  - subject to the constraint

$$\sum_{j=1}^{M}|w_j|^q \leq \eta$$

  - (for some suitably chosen η)

Regularized Least-Squares

- **Effect: Sparsity for $q \leq 1$.**
  - **Minimization tends to set many coefficients to zero**



- ***Why is this good?***
- ***Why don't we always do it, then? Any problems?***

Image source: C.M. Bishop, 2006

The Lasso

- Consider the following regressor

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

  - This formulation is known as the **Lasso**.

- Properties
  - L_1 regularization $\Rightarrow$ The solution will be sparse (only few coefficients will be non-zero)
  - The L_1 penalty makes the problem non-linear.

  $\Rightarrow$ There is no closed-form solution.

  $\Rightarrow$ Need to solve a quadratic programming problem.
  - However, efficient algorithms are available with the same computational cost as for ridge regression.

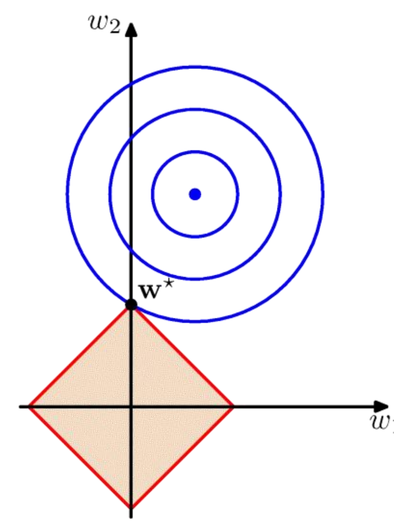


Image source: C.M. Bishop, 2006

Lasso as Bayes Estimation

- **Interpretation as Bayes Estimation**

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

  - **We can think of $|w_j|^q$ as the log-prior density for w_j.**

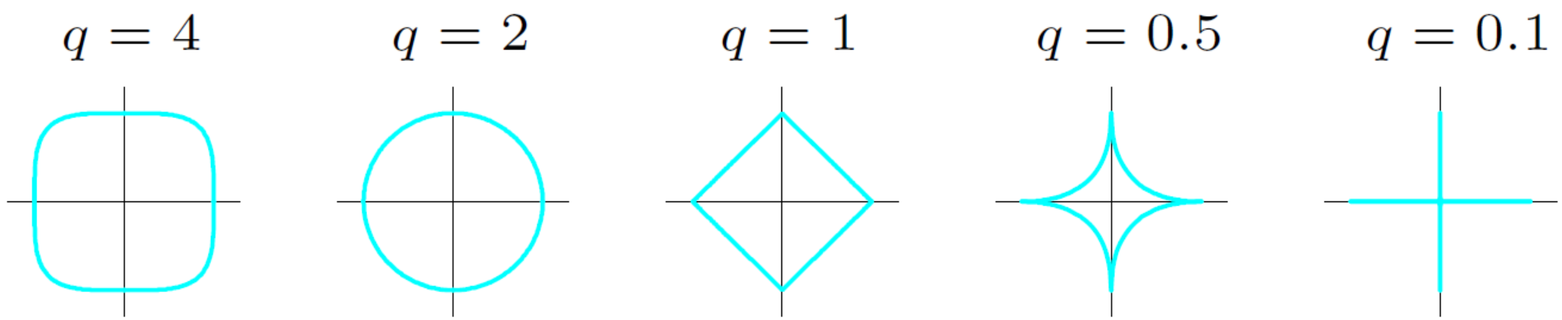
- **Prior for Lasso $(q = 1)$: Laplacian distribution**

$$p(\mathbf{w}) = \frac{1}{2\tau}\exp\{-|\mathbf{w}|/\tau\} \qquad \textbf{with} \qquad \tau = \frac{1}{\lambda}$$

Image source: Friedman, Hastie, Tibshirani, 2009

Analysis

- Equicontours of the prior distribution



- Analysis
  - For $q \leq 1$, the prior is not uniform in direction, but concentrates more mass on the coordinate directions.
  - The case $q = 1$ (lasso) is the smallest q such that the constraint region is convex.

  $\Rightarrow$ Non-convexity makes the optimization problem more difficult.
  - Limit for $q = 0$: regularization term becomes $\sum_{j=1..M} 1 = M$.

  $\Rightarrow$ This is known as Best Subset Selection.

Image source: Friedman, Hastie, Tibshirani, 2009

Discussion

- Bayesian analysis
  - Lasso, Ridge regression and Best Subset Selection are Bayes estimates with different priors.
  - However, derived as maximizers of the posterior.
  - Should ideally use the posterior mean as the Bayes estimate!

  $\Rightarrow$ Ridge regression solution is also the posterior mean, but Lasso and Best Subset Selection are not.

- We might also try using other values of q besides $0,1,2\ldots$
  - However, experience shows that this is not worth the effort.
  - Values of $q \in (1,2)$ are a compromise between lasso and ridge
  - However, $|w_j|^q$ with $q > 1$ is differentiable at 0.

  $\Rightarrow$ Loses the ability of lasso for setting coefficients exactly to zero.

Topics of This Lecture

- Recap: Probabilistic View on Regression
- Properties of Linear Regression
  - Loss functions for regression
  - Basis functions
  - Multiple Outputs
  - Sequential Estimation
- Regularization revisited
  - Regularized Least-squares
  - The Lasso
  - Discussion
- **Bias-Variance Decomposition**

Bias-Variance Decomposition

- 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?

Bias-Variance Decomposition

- 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

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

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

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

Bias-Variance Decomposition

- **Thus we can write**

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

  - **where**

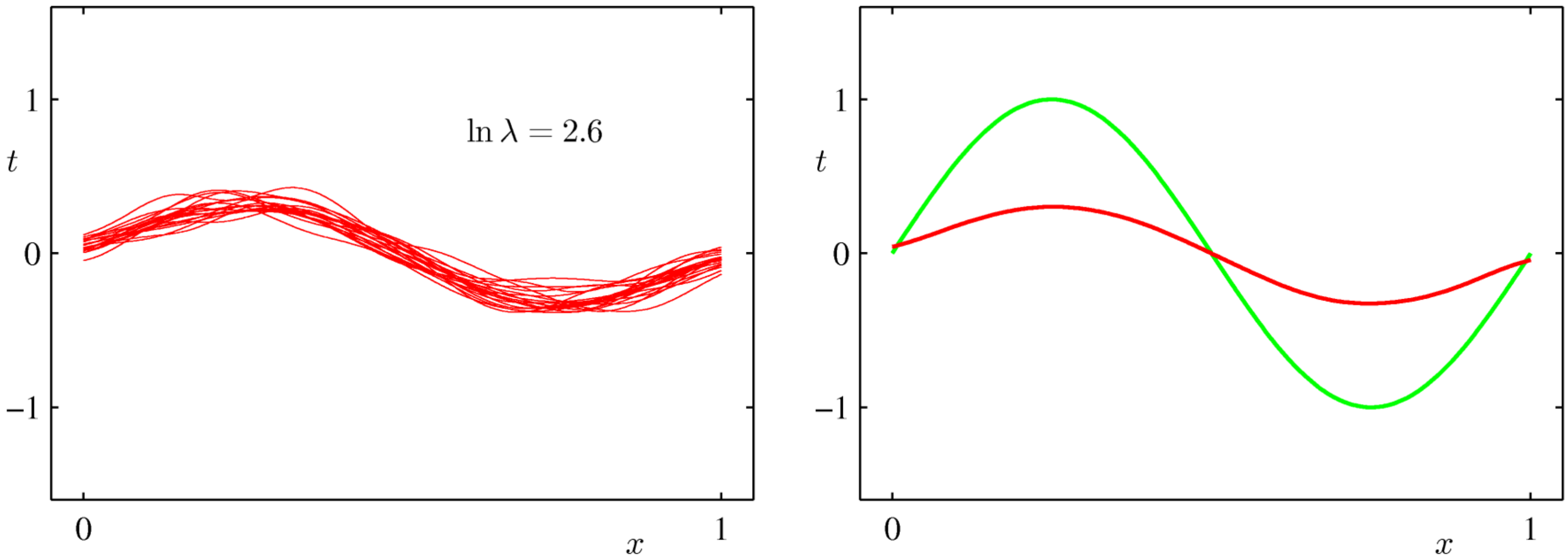
$$(\text{bias})^2 = \int \{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2 p(\mathbf{x})\,\mathrm{d}\mathbf{x}$$

$$\text{variance} = \int \mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}^2\right] p(\mathbf{x})\,\mathrm{d}\mathbf{x}$$

$$\text{noise} = \iint \{h(\mathbf{x}) - t\}^2 p(\mathbf{x}, t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t$$

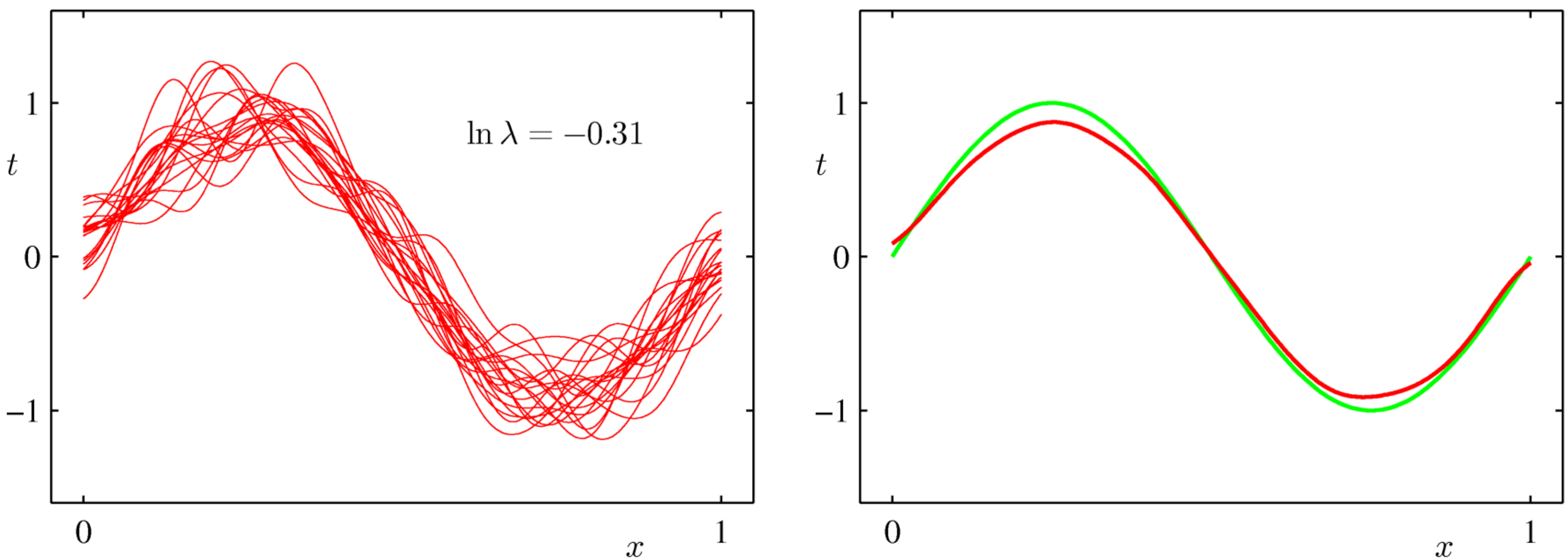
Bias-Variance Decomposition

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



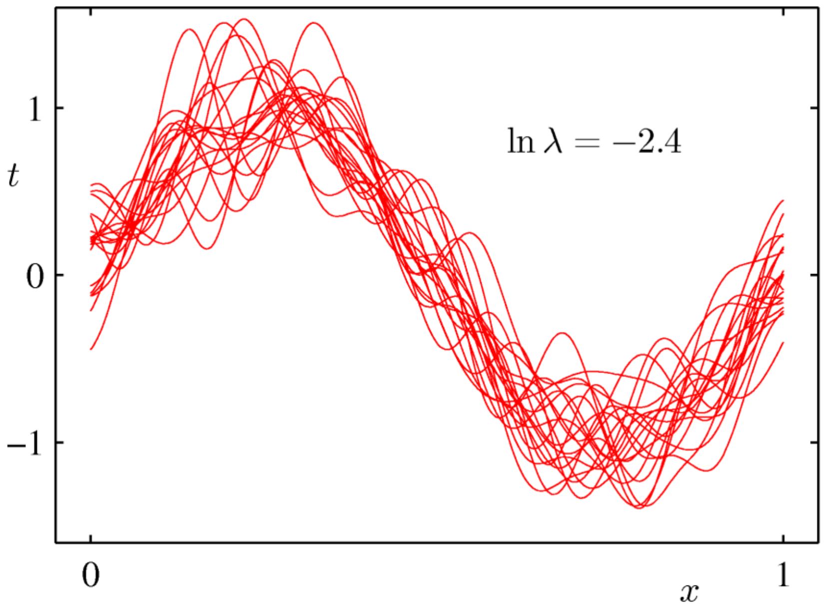
Bias-Variance Decomposition

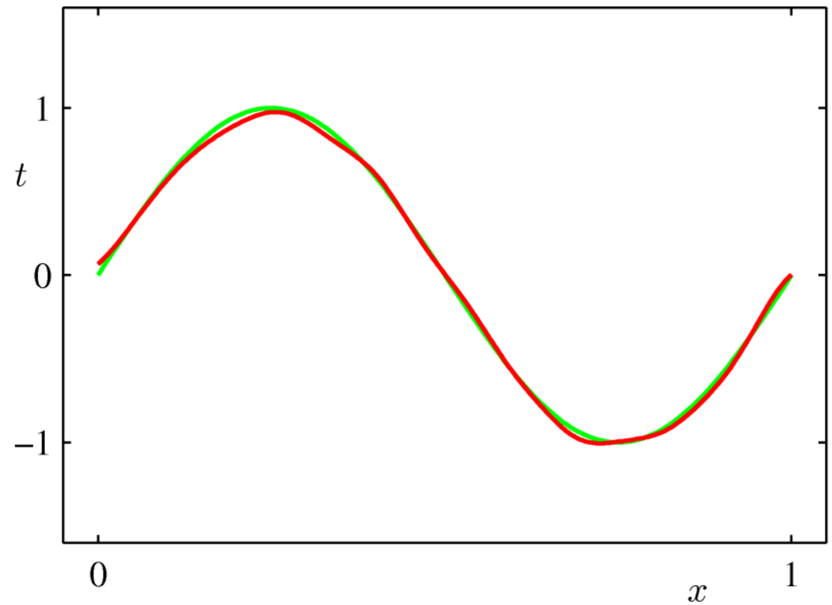
- **Example**
  - **25 data sets from the sinusoidal, varying the degree of regularization, λ.**



Bias-Variance Decomposition

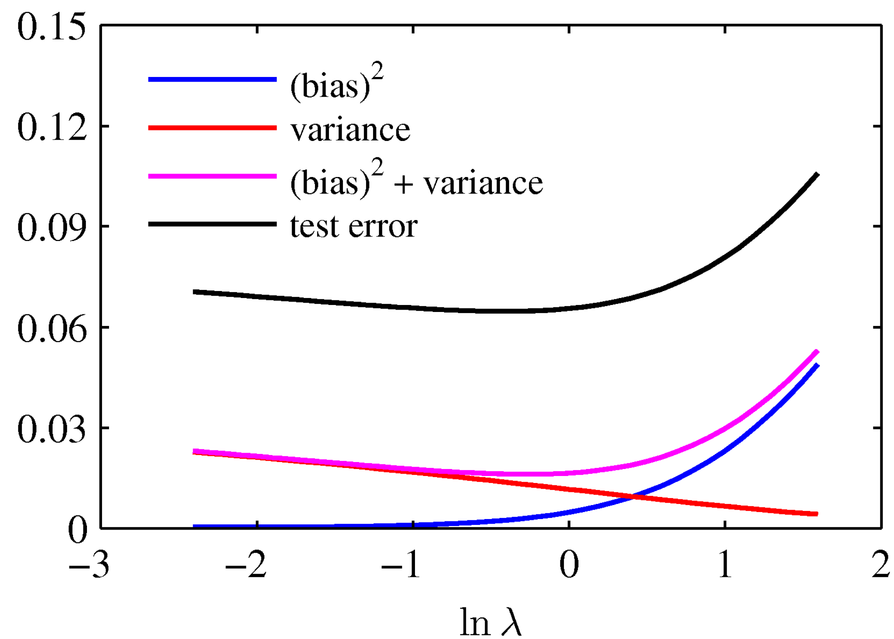
- Example
  - 25 data sets from the sinusoidal, varying the degree of regularization, λ.





The Bias-Variance Trade-Off

- **Result from these plots**
  - **An over-regularized model (large λ) will have a high bias.**
  - **An under-regularized model (small λ) will have a high variance.**



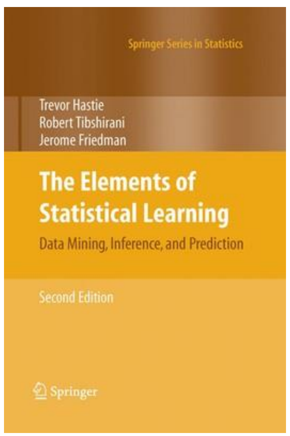
- **We can compute an estimate for the generalization capability this way (magenta curve)!**
  - ***Can you see where the problem is with this?***
  - **$\Rightarrow$ Computation is based on average w.r.t. ensembles of data sets.**
  - **$\Rightarrow$ Unfortunately of little practical value…**

Image source: C.M. Bishop, 2006

RWTH AACHEN UNIVERSITY

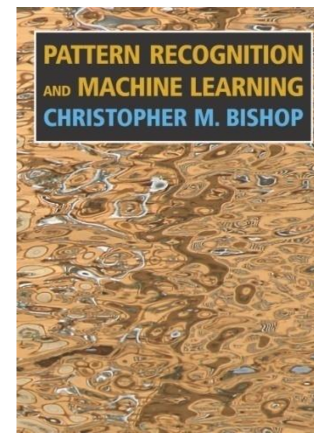
References and Further Reading

- More information on linear regression, including a discussion on regularization can be found in Chapters 1.5.5 and 3.1-3.2 of the Bishop book.



Christopher M. Bishop
Pattern Recognition and Machine Learning
Springer, 2006

T. Hastie, R. Tibshirani, J. Friedman
Elements of Statistical Learning
2nd edition, Springer, 2009



- Additional information on the Lasso, including efficient algorithms to solve it, can be found in Chapter 3.4 of the Hastie book.