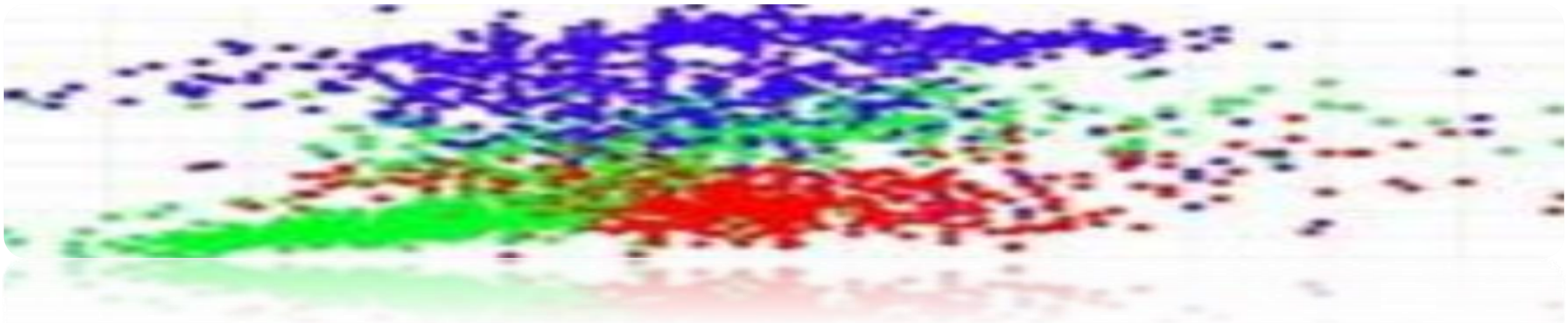


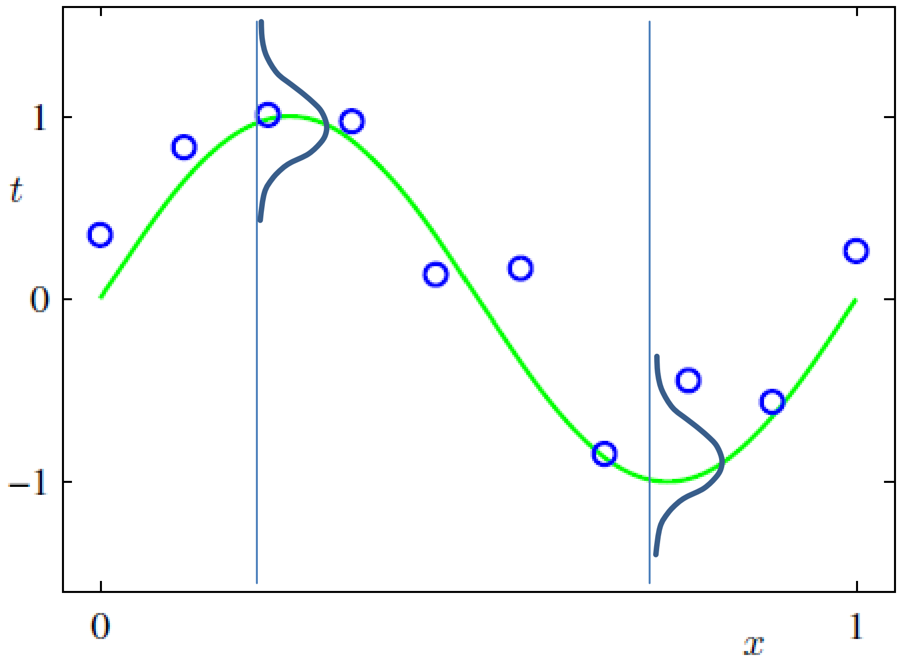
REGRESSION

Outline

- Linear regression
- Regularization functions
- Polynomial curve fitting
- Stochastic gradient descent for regression
- MLE for regression
- Step-wise forward regression

Regression methods

- Statistical techniques for finding the best-fitting curve for a set of perturbed values from unknown function



Points generated from $\sin(2\pi x)$, perturbed with Gaussian noise

Example from C. Bishop: PRML book

Error functions for regression

- Let $(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_n, t_n)$ be pairs of instances and their true values for an unknown function $f: \mathcal{X} \rightarrow \mathbb{R}$, $\mathcal{X} \subseteq \mathbb{R}^k$
- Let $y_1, \ldots, y_n \in \mathbb{R}$ be the values returned by a regression model for instances $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$
- **Sum-of-squares error** (also called quadratic error or least-squares error)

$$e_{sq}(y_1, \ldots, y_n, t_1, \ldots, t_n) = \frac{1}{2} \sum_{i=1}^{n} (y_i - t_i)^2$$

$$E\left(e_{sq}\right) = var + bias^2 + noise$$

- **Mean squared error**

$$mse(y_1, \ldots, y_n, t_1, \ldots, t_n) = 2e_{sq}(y_1, \ldots, y_n, t_1, \ldots, t_n)/n$$

- **Root-mean-square error**

$$e_{rms}(y_1, \ldots, y_n, t_1, \ldots, t_n) = \sqrt{mse(y_1, \ldots, y_n, t_1, \ldots, t_n)}$$

Curve fitting

- General idea:
  - Use approximation function of the form

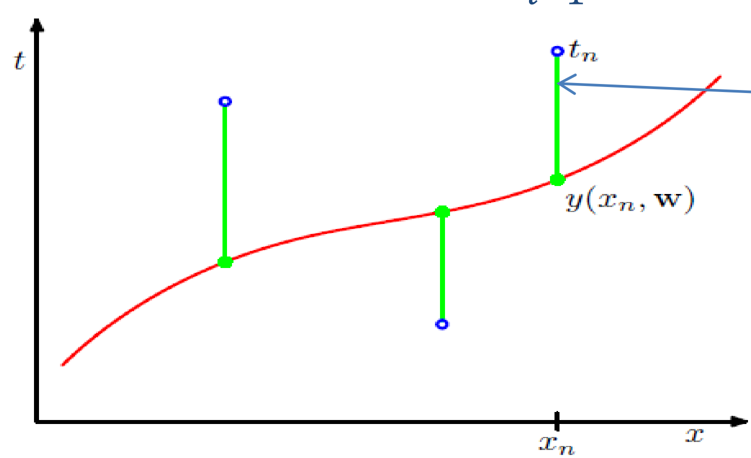
    $y(\mathbf{x}_i, \mathbf{w}) = w_0 + w_1\phi_1(\mathbf{x}_i) + \cdots + w_M\phi_M(\mathbf{x}_i),\ \mathbf{x}_i \in \mathbb{R}^k, \phi_j\colon \mathbb{R}^k \to \mathbb{R}$

    e.g., for $\mathbf{x}_i \in \mathbb{R}$, $y(\mathbf{x}_i, \mathbf{w}) = \sum_{j=o}^{M} w_j \mathbf{x}_i^j$ with $\phi_j(\mathbf{x}_i) = \mathbf{x}_i^j$

    $\phi_j(\mathbf{x}_i)$ are called basis functions

  - Minimize misfit between $y(x_i, \mathbf{w})$ and $t_i, 1 \leq i \leq n$, e.g., the sum-of-squares error

$$\frac{1}{2}\sum_{i=1}^{n}(y(x_i, \mathbf{w}) - t_i)^2$$



displacement/residual

sum-of-squares error

$\equiv \frac{1}{2}$ sum-of-squares of displacements

Example from C. Bishop: PRML book

Univariate linear regression

- General form of univariate linear regression

$$t = w_0 + w_1 x + noise, \qquad x, w_j \in \mathbb{R}$$

- Example
  - Suppose we aim at investigating the relationship between people's height (h_i) and weight (g_i) based on measurements

$$(h_i, g_i), 1 \le i \le n$$

  - Find

$$g_i = w_0 + w_1 h_i, \forall i$$

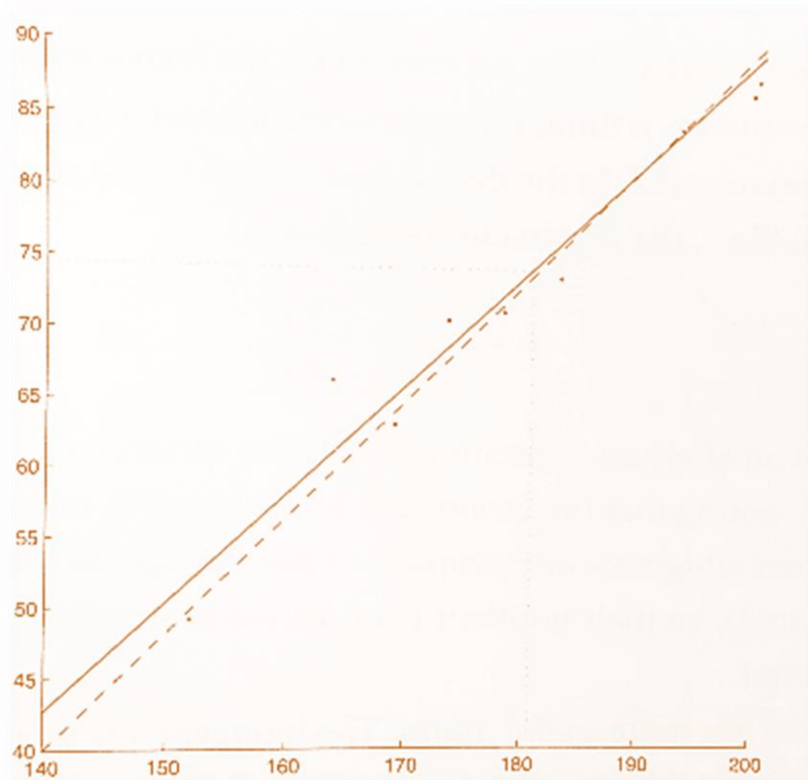
subject to

$$\min_{w_0, w_1} \frac{1}{2} \sum_{i=1}^{n} \left(g_i - (w_0 + w_1 h_i)\right)^2$$

Least-squares method

Example

Example from "Machine Learning" by P. Flach



- 9 simulated measurements by adding Gaussian noise to the dashed linear function
- Solid line represents linear regression applied to the 9 points with mean 0 and variance 5

Optimal parameters for univariate linear regression

- Set derivatives for the intercept (w_0) and the slope (w_1) to zero and solve for each of the variables, respectively:

$$\frac{\partial}{\partial w_0}\frac{1}{2}\sum_{i=1}^{n}\left(g_i - (w_0 + w_1 h_i)\right)^2 = -\sum_{i=1}^{n}\left(g_i - (w_0 + w_1 h_i)\right) = 0$$

$$\Rightarrow \ \hat{w}_0 = \bar{g} - \hat{w}_1 \bar{h}$$

$$\frac{\partial}{\partial w_1}\frac{1}{2}\sum_{i=1}^{n}\left(g_i - (w_0 + w_1 h_i)\right)^2 = -\sum_{i=1}^{n}\left(g_i - (w_0 + w_1 h_i)\right)h_i = 0$$

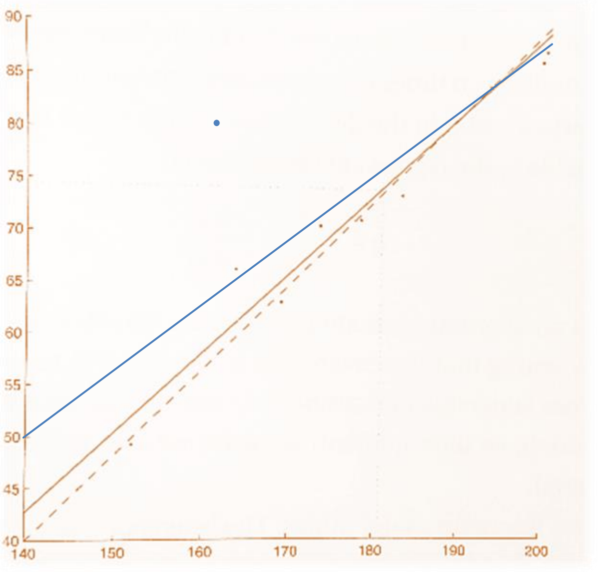
$$\Rightarrow \hat{w}_1 = \frac{\sum_{i=1}^{n}\left(h_i - \bar{h}\right)\left(g_i - \bar{w}\right)}{\sum_{i=1}^{n}\left(h_i - \bar{h}\right)^2} = \frac{n \cdot Cov(h,g)}{n \cdot Var(h)}$$

$$\Rightarrow g = \hat{w}_0 + \hat{w}_1 h = \bar{g} + \hat{w}_1\left(h - \bar{h}\right)$$

Abstract view on univariate linear regression

- For a target variable t that is linearly dependent on a feature x, i.e.,

$$t = w_0 + w_1 x + noise$$



the general solution depends only on

$$\hat{w}_1 = \frac{Cov(x,t)}{Var(x)}$$

- This means that solution is highly sensitive to noise and outliers

- Steps
  1. Normalize the feature by dividing its values by the feature's variance
  2. Calculate the covariance between target variable and normalized feature

Probabilistic view on least-squares

- $t_i = w_0 + w_1 x_i + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2)$ i.i.d. normally distributed errors
- Assumption: $t_i \sim N(w_0 + w_1 x_i, \sigma^2)$

$$P(t_i | w_0, w_1, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left(t_i - (w_0 + w_1 x_i)\right)^2}{2\sigma^2}\right)$$

- For n i.i.d. data points $t_1, \dots, t_n$:

$$P(t_1, \dots, t_n | w_0, w_1, \sigma^2) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left(t_i - (w_0 + w_1 x_i)\right)^2}{2\sigma^2}\right)$$

$$= \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n \exp\left(-\frac{\sum_{i=1}^{n}\left(t_i - (w_0 + w_1 x_i)\right)^2}{2\sigma^2}\right)$$

$$\propto -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\sigma^2) - \frac{\sum_{i=1}^{n}\left(t_i - (w_0 + w_1 x_i)\right)^2}{2\sigma^2}$$

Maximum Likelihood Estimation of w_0, w_1, σ^2

$$\frac{\partial \ln\left(P(t_1, \dots, t_n | w_0, w_1, \sigma^2)\right)}{\partial w_0} = \sum_{i=1}^{n} \left(t_i - (w_0 + w_1 x_i)\right) = 0$$

$$\Rightarrow \; \hat{w}_0 = \bar{t} - \hat{w}_1 \bar{x}$$

$$\frac{\partial \ln\left(P(t_1, \dots, t_n | w_0, w_1, \sigma^2)\right)}{\partial w_1} = \sum_{i=1}^{n} \left(t_i - (w_0 + w_1 x_i)\right) x_i = 0$$

$$\Rightarrow \hat{w}_1 = \frac{Cov(x,t)}{Var(x)}$$

$$\frac{\partial \ln\left(P(t_1, \dots, t_n | w_0, w_1, \sigma^2)\right)}{\partial \sigma^2} = -\frac{n}{2}\frac{1}{\sigma^2} + \frac{\sum_{i=1}^{n} \left(t_i - (w_0 + w_1 x_i)\right)^2}{2(\sigma^2)^2} = 0$$

$$\Rightarrow \sigma^2 = \frac{\sum_{i=1}^{n} \left(t_i - (w_0 + w_1 x_i)\right)^2}{n}$$

Multivariate linear regression

$$t_i = w_0 + w_1 x_i + \epsilon_i, \qquad 1 \le i \le n$$

$$\Leftrightarrow$$

$$\begin{pmatrix} t_1 \\ \cdot \\ \cdot \\ \cdot \\ t_n \end{pmatrix} = \begin{pmatrix} 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{pmatrix} w_0 + \begin{pmatrix} x_1 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{pmatrix} w_1 + \begin{pmatrix} \epsilon_1 \\ \cdot \\ \cdot \\ \cdot \\ \epsilon_n \end{pmatrix}$$

$$\Leftrightarrow$$

$$\begin{pmatrix} t_1 \\ \cdot \\ \cdot \\ \cdot \\ t_n \end{pmatrix} = \begin{pmatrix} 1 & x_1 \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ 1 & x_n \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \cdot \\ \cdot \\ \cdot \\ \epsilon_n \end{pmatrix}$$

- General form of multivariate linear regression

$$\mathbf{t} = \mathbf{X}\mathbf{w} + \boldsymbol{\epsilon}$$

$\mathbf{t} \in \mathbb{R}^{n \times 1}$, vector of target variables
$\mathbf{X} \in \mathbb{R}^{n \times m}$, matrix of n feature vectors (each containing m features)
$\mathbf{w} \in \mathbb{R}^{m \times 1}$, weight vector (i.e., a weight for each feature)
$\boldsymbol{\epsilon} \in \mathbb{R}^{n \times 1}$, noise vector

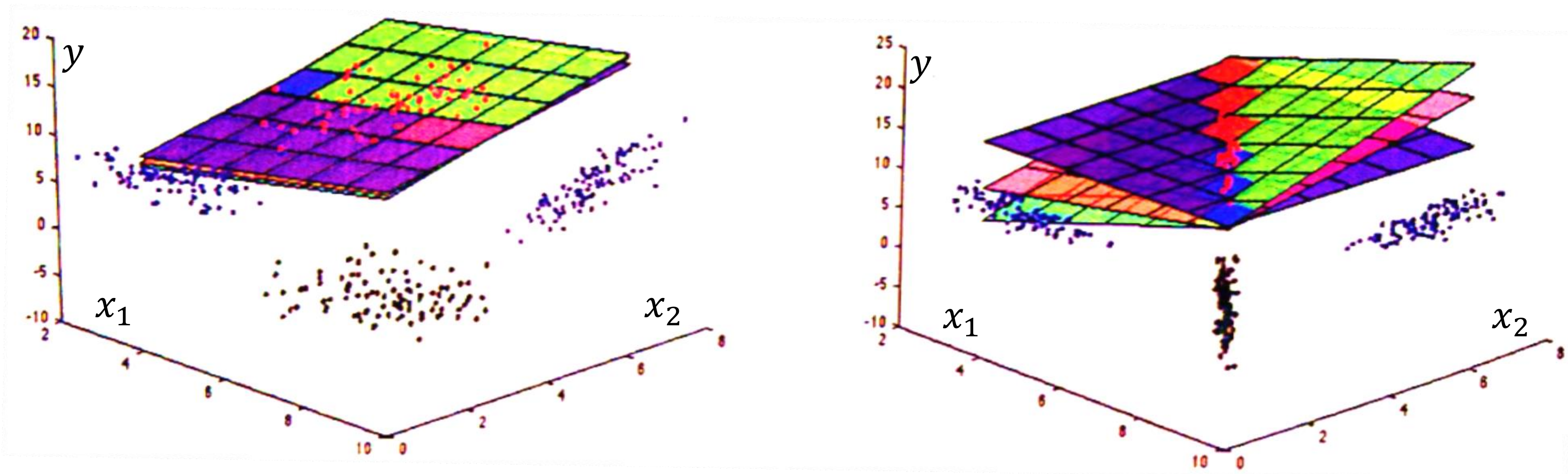
General solution for w in the multivariate case

- For univariate linear regression we found $\hat{w}_1 = \frac{Cov(x,t)}{Var(x)}$
- It turns out that the general solution for the weight vector in the multivariate case

$$\hat{\mathbf{w}} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{t}$$

- Assume feature vectors (i.e., rows) in $\mathbf{X}$ are 0-centered, i.e., from each row $(x_{i1}, \dots, x_{im})$ we have subtracted $(\overline{x_{.1}}, \dots, \overline{x_{.m}})$, where $\overline{x_{.j}} := \frac{1}{n}\sum_{i=1}^{n} x_{ij}$
- Then $\frac{1}{n}\mathbf{X}^{\mathrm{T}}\mathbf{X}$ is the $m \times m$ covariance matrix, i.e., containing the pair-wise covariances between all features (what does it contain in the diagonal?) → $\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}$ decorrelates, centers, and normalizes features
- And $\frac{1}{n}\left(\mathbf{X}^{\mathrm{T}}\mathbf{t}\right)$ is an m-vector holding the covariance between each feature and the output values $\mathbf{t}$

Effect of correlation between features



Example from “Machine Learning” by P. Flach

- Red dots represent noisy samples of y
- Red plane represents true function $y = x_1 + x_2$
- Green plane function learned by multivariate linear regression
- Blue plane function learned by decomposing the problem into two univariate regression problems
- On the right features are highly correlated, the sample gives much less information about the true function

Regularized multivariate linear regression

- Least squares method

$$\mathbf{w}^* = \mathrm{argmin}_{\mathbf{w}} \underbrace{(\mathbf{t} - \mathbf{X}\mathbf{w})^{\mathrm{T}}(\mathbf{t} - \mathbf{X}\mathbf{w})}_{\text{Least-squares error}} + \underbrace{\lambda \|\mathbf{w}\|^2}_{\text{Regularization term}}$$

- Solution is

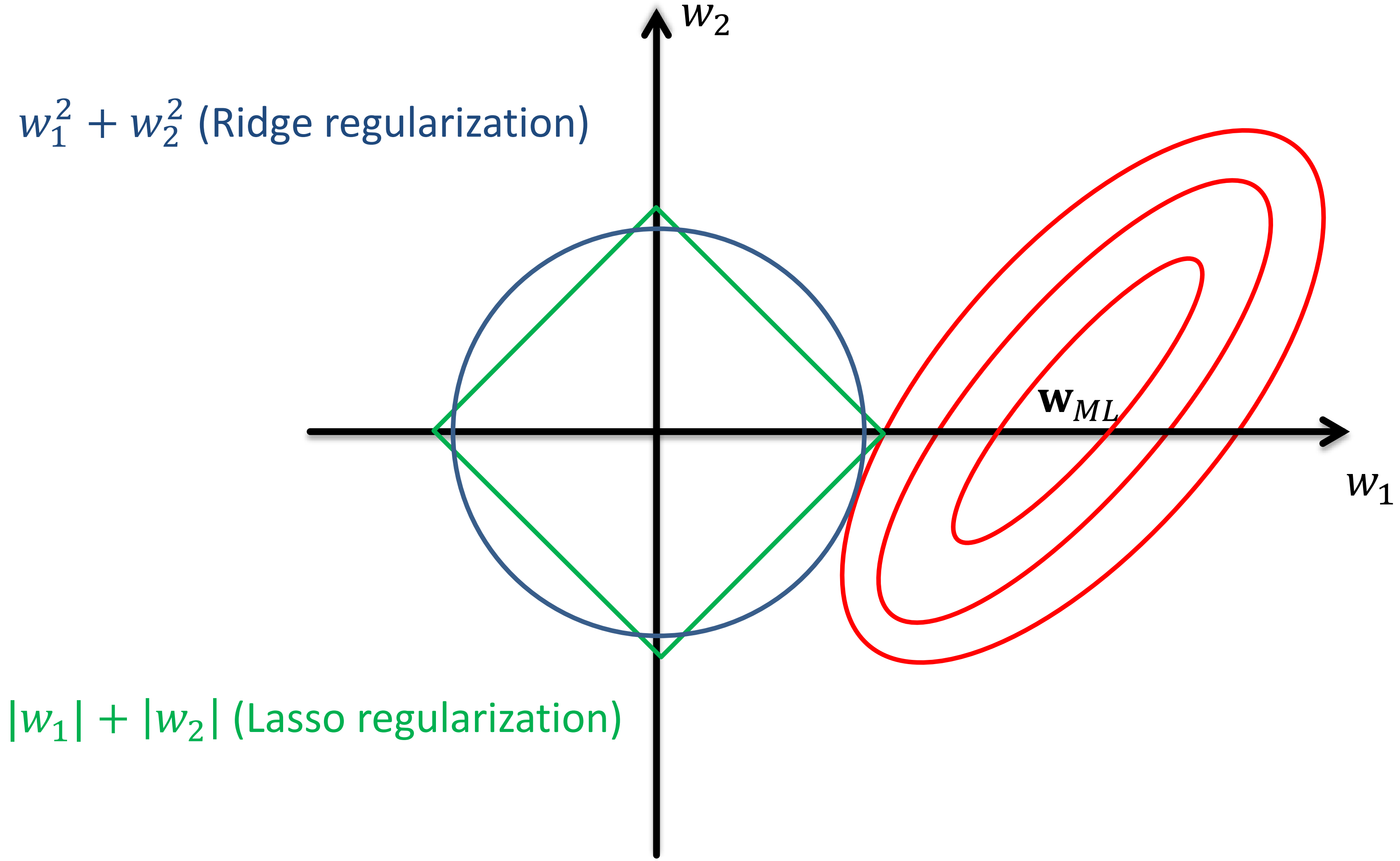
$$\hat{\mathbf{w}} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{t}$$

$\mathbf{I}$ is the identity matrix with 1s in the diagonal and 0s everywhere else

- For the regularization one can use
  - Ridge regularization $\|\mathbf{w}\|^2 = \sum_i w_i^2$ (i.e., L2 norm) → Ridge regression
  - Lasso regularization $|\mathbf{w}| = \sum_i |w_i|$ (i.e., L1 norm), which favors sparser solutions → Lasso regression
  - λ determines the amount of regularization

- Lasso regression is much more sensitive to the choice of λ

Ridge vs. Lasso regularization



Linear regression for classification

- We learned that the general solution for $\mathbf{w}$ is

$$\hat{\mathbf{w}} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{t}$$

- For linear classification the goal is to learn $\mathbf{w}^*$ for a decision boundary

$$\mathbf{w}^* \cdot \mathbf{x} = b$$

- Can we set $\mathbf{w}^* = \hat{\mathbf{w}}$?

- Yes → **Least-squares classifier**
  - $\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}$ decorrelates, centers, and normalizes features (good to have)
  - Suppose $\mathbf{t} = \begin{pmatrix} (+/-)1 \\ \dots \\ (+/-)\ 1 \end{pmatrix}$; what is the result of $\mathbf{X}^{\mathrm{T}}\mathbf{t}$?
  - Caution: Complexity of computing $\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}$ is $O(n^2 m + m^3)$

Univariate polynomial curve fitting

- Use approximation function of the form

$$y(x_i, \mathbf{w}) = w_0 + w_1\phi_1(x_i) + \cdots + w_M\phi_M(x_i),\ \text{where } \phi_j(x_i) = x_i^j$$

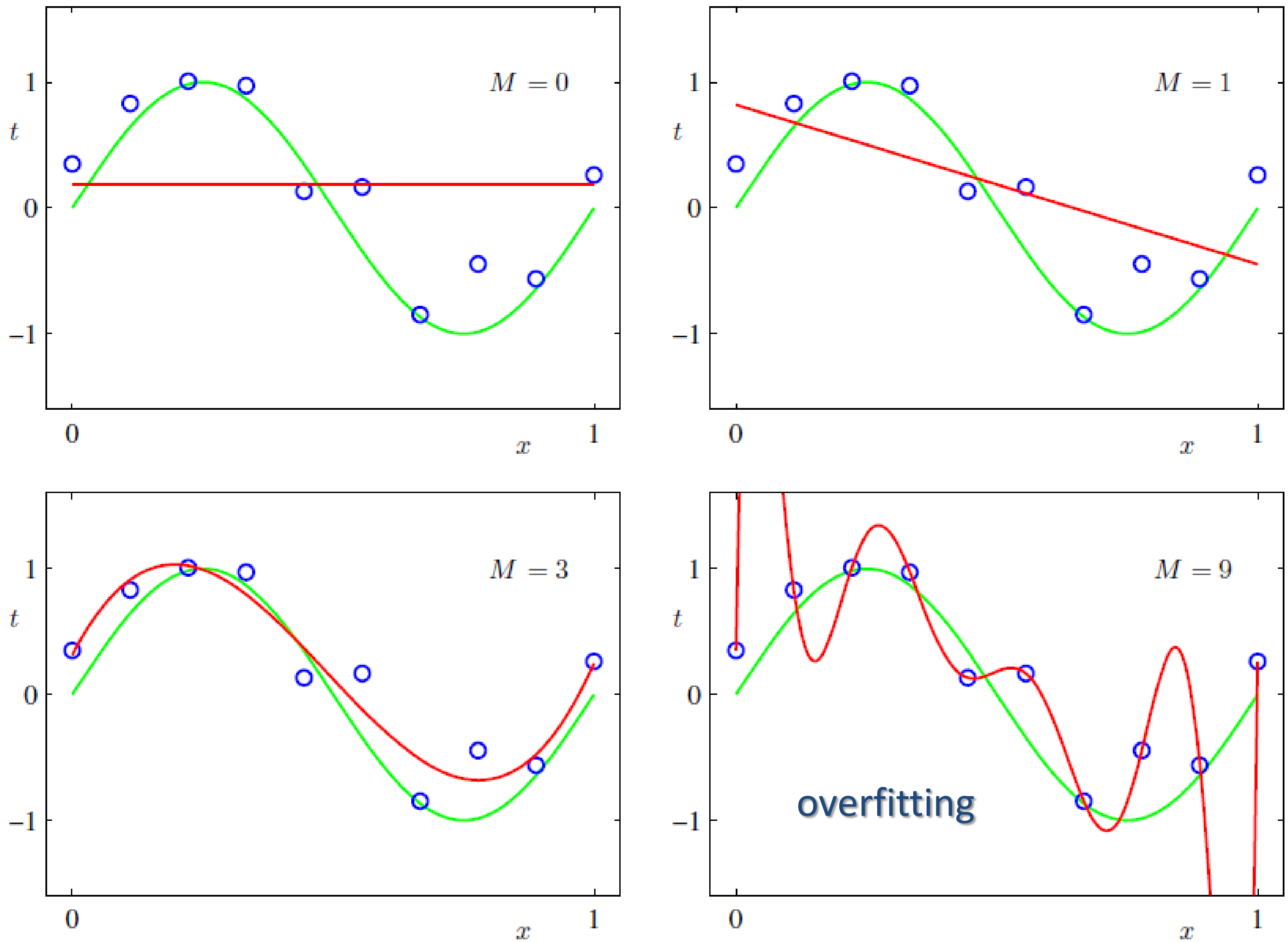
bias term

- Least-squares regression: Minimize misfit between $y(x_i, \mathbf{w})$ and $t_i, 1 \leq i \leq n$, e.g., the sum-of-squares error

$$\frac{1}{2}\sum_{i=1}^{n}(y(x_i, \mathbf{w}) - t_i)^2$$

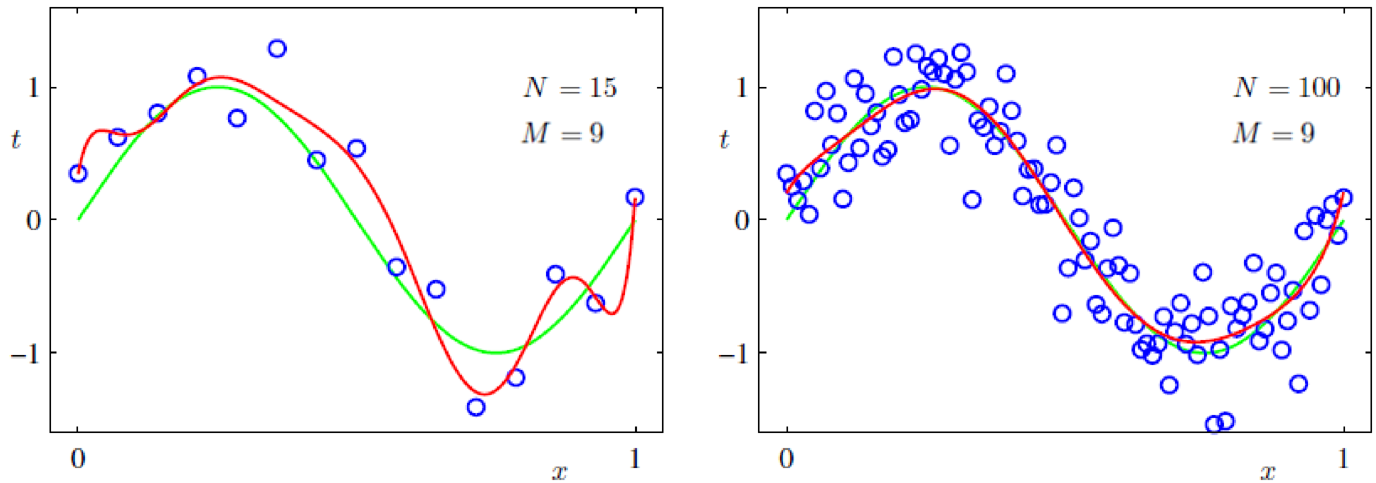
- Still linear in the weights w_i

Example of overfitting



Example from C. Bishop: PRML book

Impact of data and regularization

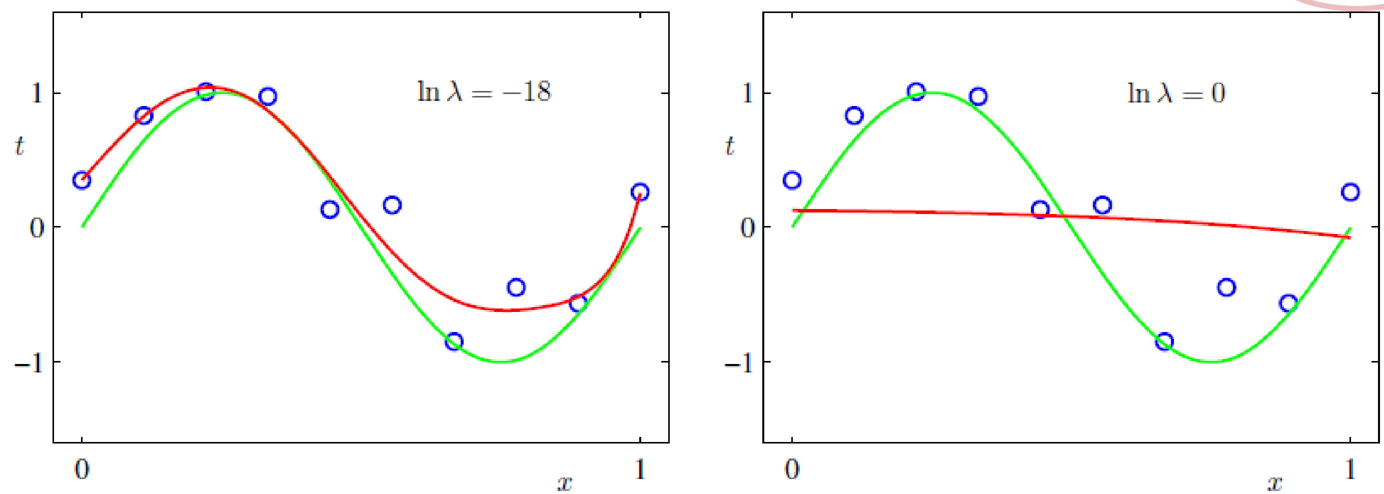


Increasing the number of data points mitigates overfitting

Regularization coefficient

Another possibility: Use regularization $\tilde{e}(\mathbf{w}) = \frac{1}{2}\sum_i (y(x_i, \mathbf{w}) - t_i)^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$

Regularization term



Examples from C. Bishop: PRML book

Polynomial curve fitting: Stochastic Gradient Descent

- **Definition:** *The gradient of a differentiable function* $f(w_1, \dots, w_M)$ *is defined as*

$$\nabla_{\mathbf{w}} f = \frac{\partial f}{\partial w_1}\mathbf{e}_1 + \cdots + \frac{\partial f}{\partial w_M}\mathbf{e}_M$$

where the $\boldsymbol{e}_i$ *are orthogonal unit vectors*

- **Theorem**: For a function f that is differentiable in the neighborhood of a point $\mathbf{w},\ \mathbf{w}' := \mathbf{w} - \eta\nabla_{\mathbf{w}} f(\mathbf{w})$ yields $f(\mathbf{w}') < f(\mathbf{w})$ for small enough $\eta > 0$

- **Least-mean-squares algorithm**

```
For each data point x_i
```

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta\nabla_{\mathbf{w}}\left(\frac{1}{2}\sum_{i=1}^{n}\left(t_i - \mathbf{w}^{(\tau)T}\boldsymbol{\phi}(x_i)\right)^2\right)$$

```
//gradient descent
//η:learning rate
```

$$\approx \mathbf{w}^{(\tau)} - \eta\left(t_i - \mathbf{w}^{(\tau)T}\boldsymbol{\phi}(x_i)\right)\boldsymbol{\phi}(x_i)$$

```
//stochastic gradient descent
//with the least-mean squares rule
```

Polynomial curve fitting: Maximum Likelihood Estimation

- Assume each observation t_i comes from function, with added Gaussian noise

$$t_i = y(x_i, \mathbf{w}) + \varepsilon, \qquad P(\varepsilon|\sigma^2) = N(\varepsilon|0, \sigma^2)$$

$$\Leftrightarrow$$

$$P(t_i|x_i, \mathbf{w}, \sigma^2) = N(t_i|y(x_i, \mathbf{w}), \sigma^2)$$

- We can write the likelihood function based on the observations

$$y(x_i, \mathbf{w}) = w_0 + w_1\phi_1(x_i) + \cdots + w_M\phi_M(x_i) = \mathbf{w}^T\boldsymbol{\phi}(x_i)$$

$$P(\mathbf{t}|\mathbf{x}, \mathbf{w}, \sigma^2) = \prod_i N(t_i|y(x_i, \mathbf{w}), \sigma^2) = \prod_i N(t_i|\mathbf{w}^T\boldsymbol{\phi}(x_i), \sigma^2)$$

Polynomial curve fitting: Maximum Likelihood Estimation (2)

- We can write the likelihood function based on i.i.d. observations

$$P(\mathbf{t}|\mathbf{x},\mathbf{w},\sigma^2) = \prod_i N(t_i|y(x_i,\mathbf{w}),\sigma^2) = \prod_i N(t_i|\mathbf{w}^T\boldsymbol{\phi}(\mathrm{x}_i),\sigma^2)$$

- Taking the logarithm

$$\ln P(\mathbf{t}|\mathbf{x},\mathbf{w},\sigma^2) = \sum_{i=1}^{n} \ln(N(t_i|\mathbf{w}^T\boldsymbol{\phi}(x_i),\sigma^2))$$

$$= -\frac{n}{2}\ln\sigma^2 - \frac{n}{2}\ln(2\pi) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(t_i - \mathbf{w}^T\boldsymbol{\phi}(x_i))^2$$

Polynomial curve fitting: Maximum Likelihood Estimation (3)

- Taking the gradient and setting it to zero

$$\nabla_{\mathbf{w}} \ln P(\mathbf{t}|\mathbf{x},\mathbf{w},\sigma^2) = \frac{1}{\sigma^2}\sum_{i=1}^{N}\big(t_i - \mathbf{w}^T\boldsymbol{\phi}(x_i)\big)\boldsymbol{\phi}(x_i)^T = 0$$

- Solving for $\mathbf{w}$

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

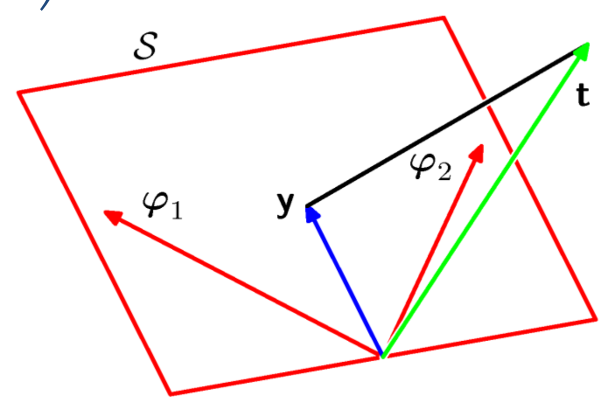
where

$$\mathbf{\Phi} = \begin{pmatrix} \phi_0(x_1) & \cdots & \phi_M(x_1) \\ \vdots & \ddots & \vdots \\ \phi_0(x_N) & \cdots & \phi_M(x_N) \end{pmatrix}$$

- Geometrical interpretation

$\mathbf{y} = \mathbf{\Phi}\mathbf{w}_{ML} = [\varphi_0, \ldots, \varphi_M]\mathbf{w}_{ML} \in S \subseteq \mathcal{T} \ni \mathbf{t}$

($\mathbf{w}_{ML}$ minimizes the distance between $\mathbf{t}$ and its projection on S)



Example from C. Bishop: PRML book

Reviewing the multivariate case

- Generalization to the multivariate case

$$y(\mathbf{x}, \mathbf{w}) = \sum_{i=o}^{M} w_i \, \phi_i(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}), \qquad \mathbf{x} \in \mathbb{R}^k$$

- The discussed algorithms of stochastic gradient descent and MLE generalize to this case as well

- The choice of ϕ_i is crucial for the tradeoff between regression quality and complexity

Choices for basis functions

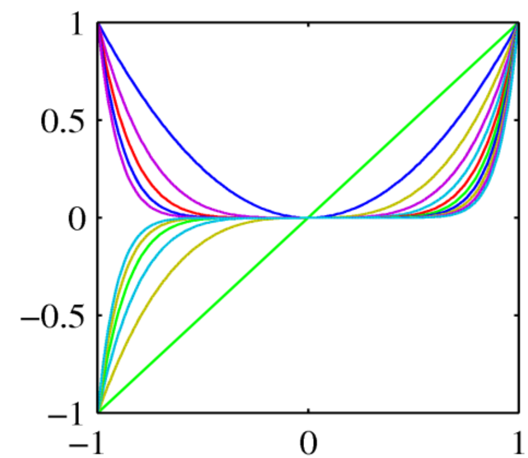
- Simplest case: Return the i'th component of $\mathbf{x}$

$$\phi_i(\mathbf{x}) = x_{(i)}$$

- Polynomial basis function for $x \in \mathbb{R}$

$$\phi_i(x) = x^i$$

(small changes in x affect all basis functions)



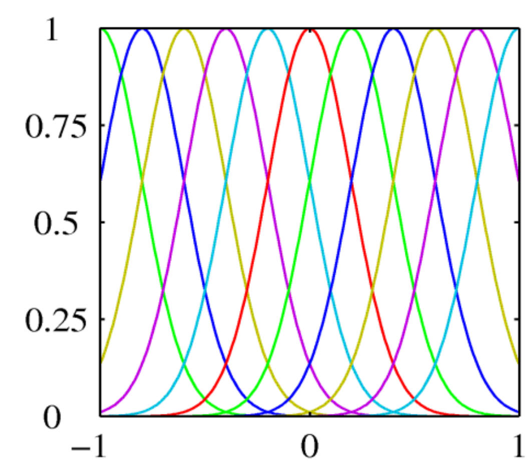
- Gaussian basis function (for $x \in \mathbb{R}$)

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

controls location

controls scale

(small changes in x affect nearby basis functions)



Examples from C. Bishop: PRML book

Forward step-wise regression

- Goal: Find fitting function $\hat{y}(x_i) = y_1(x_i, \mathbf{w}_1) + \cdots + y_n(x_i, \mathbf{w}_n)$

- Step 1: Fit first simple function $y_1(x_i, \mathbf{w}_1)$

$$\mathbf{w}_1 = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^{n} \big(t_i - y_1(x_i, \mathbf{w})\big)^2$$

- Step 2: Fit second simple model $y_2(x_i, \mathbf{w}_2)$ to the residuals of the first:

$$\mathbf{w}_2 = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^{n} \big(t_i - y_1(x_i, \mathbf{w}_1) - y_2(x_i, \mathbf{w})\big)^2$$

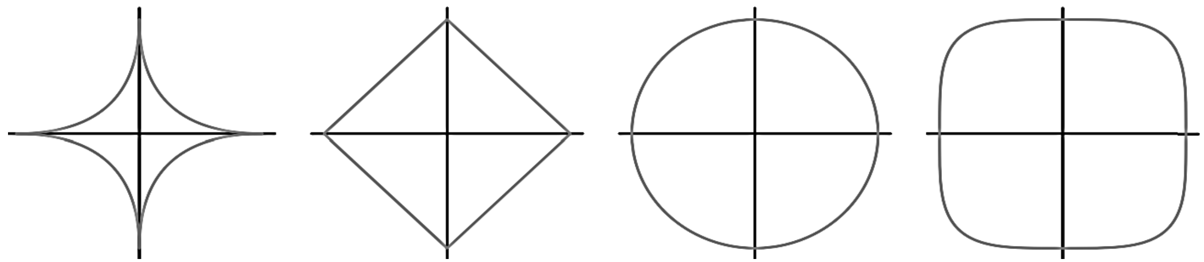
.
.
.

- Step n: Fit a simple model $y_n(x_i, \mathbf{w}_n)$ to the residuals of the previous step

- Stop when no significant improvement in training error is made

Further considerations on regression

- Other choices of regularization functions
  - L_p-regularization is given by $\sum_i |w_i|^p$



p=0.5 p=1 p=2 p=4

  - For $p > 1$, no sparse solutions are achieved

- Tree models can be applied to regression
  - Impurity reduction translates to variance reduction (see also exercises)

Summary

- Main solution for linear regression
  - Univariate

$$\hat{w}_1 = \frac{Cov(x,t)}{Var(x)}$$

  - Multivariate

$$\hat{\mathbf{w}} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{t}$$

- Regularization mitigates overfitting
  - Lasso (L1): With high probability sparse
  - Ridge (L2): Not sparse

- Solution strategies
  - Stochastic gradient descent
  - MLE
  - Forward step-wise regression