Machine Learning and Statistics in Genetics and Genomics

IV: Regularized and Bayesian regression

Christoph Lippert

Microsoft Research
eScience group

Los Angeles , USA

Current topics in computational biology
UCLA
Winter quarter 2014

Outline

Outline

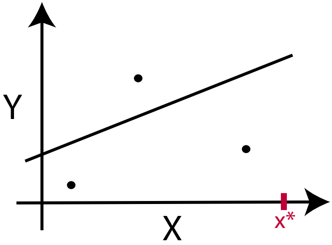
Regression

Linear regression:

- Making predictions
- Comparison of alternative models

Bayesian and regularized regression:

- Uncertainty in model parameters
- Generalized basis functions



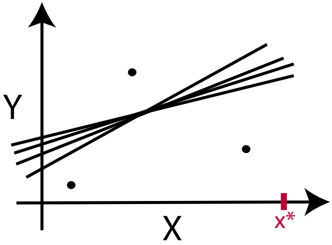
Regression

Linear regression:

- Making predictions
- Comparison of alternative models

Bayesian and regularized regression:

- Uncertainty in model parameters
- Generalized basis functions



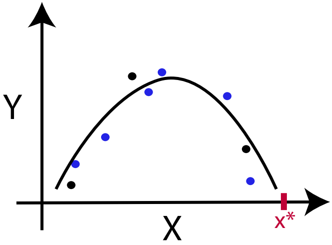
Regression

Linear regression:

- Making predictions
- Comparison of alternative models

Bayesian and regularized regression:

- Uncertainty in model parameters
- Generalized basis functions



Further reading, useful material

- Christopher M. Bishop: Pattern Recognition and Machine learning
- Sam Roweis: Gaussian identities

Outline

Regression

Noise model and likelihood

- Given a dataset $\mathcal{D} = \{\boldsymbol{x}_n, y_n\}_{n=1}^N$, where $\boldsymbol{x}_n = \{x_{n,1}, \ldots, x_{n,D}\}$ is D dimensional (for example D SNPs), fit parameters $\boldsymbol{\theta}$ of a regressor f with added Gaussian noise:

$$y_n = f(\boldsymbol{x}_n; \boldsymbol{\theta}) + \epsilon_n \quad \text{where} \quad p(\epsilon \,|\, \sigma^2) = \mathcal{N}\left(\epsilon \mid 0, \sigma^2\right).$$

- Equivalent likelihood formulation:

$$p(\boldsymbol{y} \,|\, \boldsymbol{X}) = \prod_{n=1}^{N} \mathcal{N}\left(y_n \mid f(\boldsymbol{x}_n; \boldsymbol{\theta}), \sigma^2\right)$$

Regression

Choosing a regressor

- Choose f to be linear:

$$p(\boldsymbol{y} \mid \boldsymbol{X}) = \prod_{n=1}^{N} \mathcal{N}\left(y_n \mid \boldsymbol{x}_n \cdot \boldsymbol{\beta} + c, \sigma^2 \right)$$

- Consider bias free case, $c = 0$, otherwise include an additional column of ones in each $\boldsymbol{x}_n$.

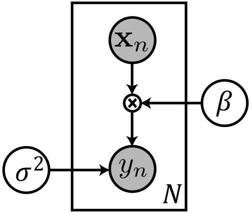
Regression

Choosing a regressor

- Choose f to be linear:

$$p(\boldsymbol{y} \mid \boldsymbol{X}) = \prod_{n=1}^{N} \mathcal{N}\left(y_n \mid \boldsymbol{x}_n \cdot \boldsymbol{\beta} + c, \sigma^2 \right)$$

- Consider bias free case, $c = 0$, otherwise include an additional column of ones in each $\boldsymbol{x}_n$.



Equivalent graphical model

Linear Regression

Maximum likelihood

- Taking the logarithm, we obtain

$$\ln p(\boldsymbol{y} \mid \boldsymbol{\beta}, \boldsymbol{X}, \sigma^2) = \sum_{n=1}^{N} \ln \mathcal{N}\left(y_n \mid \boldsymbol{x}_n \cdot \boldsymbol{\beta}, \sigma^2 \right)$$

$$= -\frac{N}{2} \ln 2\pi\sigma^2 - \frac{1}{2\sigma^2} \underbrace{\sum_{n=1}^{N} (y_n - \boldsymbol{x}_n \cdot \boldsymbol{\beta})^2}_{\text{Sum of squares}}$$

- The likelihood is maximized when the squared error is minimized.
- Least squares and maximum likelihood are equivalent.

Linear Regression

Maximum likelihood

- Taking the logarithm, we obtain

$$\begin{aligned}\ln p(\boldsymbol{y} \mid \boldsymbol{\beta}, \boldsymbol{X}, \sigma^2) &= \sum_{n=1}^{N} \ln \mathcal{N}\left(y_n \mid \boldsymbol{x}_n \cdot \boldsymbol{\beta}, \sigma^2 \right) \\ &= -\frac{N}{2} \ln 2\pi\sigma^2 - \frac{1}{2\sigma^2} \underbrace{\sum_{n=1}^{N} (y_n - \boldsymbol{x}_n \cdot \boldsymbol{\beta})^2}_{\text{Sum of squares}}\end{aligned}$$

- The likelihood is maximized when the squared error is minimized.
- Least squares and maximum likelihood are equivalent.

Linear Regression

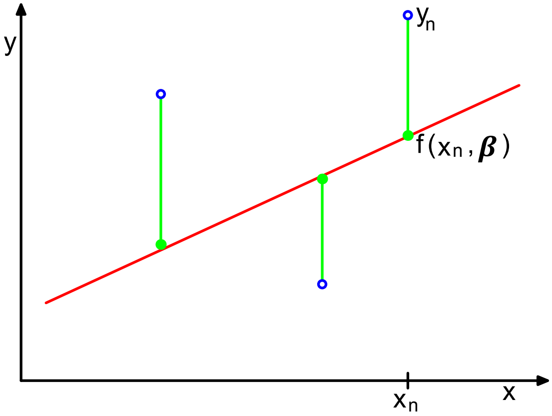
Maximum likelihood

- Taking the logarithm, we obtain

$$\begin{aligned}\ln p(\boldsymbol{y} \mid \boldsymbol{\beta}, \boldsymbol{X}, \sigma^2) &= \sum_{n=1}^{N} \ln \mathcal{N}\left(y_n \mid \boldsymbol{x}_n \cdot \boldsymbol{\beta}, \sigma^2 \right) \\ &= -\frac{N}{2} \ln 2\pi\sigma^2 - \frac{1}{2\sigma^2} \underbrace{\sum_{n=1}^{N} (y_n - \boldsymbol{x}_n \cdot \boldsymbol{\beta})^2}_{\text{Sum of squares}}\end{aligned}$$

- The likelihood is maximized when the squared error is minimized.
- Least squares and maximum likelihood are equivalent.

Linear Regression and Least Squares



(C.M. Bishop, Pattern Recognition and Machine Learning)

$$E(\boldsymbol{\beta}) = \frac{1}{2}\sum_{n=1}^{N}(y_n - \boldsymbol{x}_n \cdot \boldsymbol{\beta})^2$$

Linear Regression and Least Squares

- Derivative w.r.t a single weight entry β_i

$$\frac{d}{\mathrm{d}\beta_i} \ln p(\boldsymbol{y} \,|\, \boldsymbol{\beta}, \sigma^2) = \frac{d}{\mathrm{d}\beta_i} \left[-\frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \boldsymbol{x}_n \cdot \boldsymbol{\beta})^2 \right]$$
$$= \frac{1}{\sigma^2} \sum_{n=1}^{N} (y_n - \boldsymbol{x}_n \cdot \boldsymbol{\beta}) x_i$$

- Set gradient w.r.t to β to zero

$$\nabla_{\boldsymbol{\beta}} \ln p(\boldsymbol{y} \,|\, \boldsymbol{\beta}, \sigma^2) = \frac{1}{\sigma^2} \sum_{n=1}^{N} \boldsymbol{x}_n^\top (y_n - \boldsymbol{x}_n \cdot \boldsymbol{\beta}) = \mathbf{0}$$
$$\implies \beta_{\mathrm{ML}} = \underbrace{(\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top}_{\text{Pseudo inverse}} \boldsymbol{y}$$

- Here, the matrix $\boldsymbol{X}$ is defined as $\boldsymbol{X} = \begin{bmatrix} x_{1,1} & \dots & x1, D \\ \dots & \dots & \dots \\ x_{N,1} & \dots & x_{N,D} \end{bmatrix}$

Linear Regression and Least Squares

- Derivative w.r.t a single weight entry β_i

$$\frac{d}{\mathrm{d}\beta_i} \ln p(\boldsymbol{y} \,|\, \boldsymbol{\beta}, \sigma^2) = \frac{d}{\mathrm{d}\beta_i} \left[-\frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \boldsymbol{x}_n \cdot \boldsymbol{\beta})^2 \right]$$
$$= \frac{1}{\sigma^2} \sum_{n=1}^{N} (y_n - \boldsymbol{x}_n \cdot \boldsymbol{\beta}) x_i$$

- Set gradient w.r.t to $\boldsymbol{\beta}$ to zero

$$\nabla_{\boldsymbol{\beta}} \ln p(\boldsymbol{y} \,|\, \boldsymbol{\beta}, \sigma^2) = \frac{1}{\sigma^2} \sum_{n=1}^{N} \boldsymbol{x}_n^\top (y_n - \boldsymbol{x}_n \cdot \boldsymbol{\beta}) = \mathbf{0}$$
$$\implies \boldsymbol{\beta}_{\mathrm{ML}} = \underbrace{(\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top}_{\text{Pseudo inverse}} \boldsymbol{y}$$

- Here, the matrix $\boldsymbol{X}$ is defined as $\boldsymbol{X} = \begin{bmatrix} x_{1,1} & \dots & x1, D \\ \dots & \dots & \dots \\ x_{N,1} & \dots & x_{N,D} \end{bmatrix}$

Linear Regression and Least Squares

- Derivative w.r.t a single weight entry β_i

$$\frac{d}{\mathrm{d}\beta_i} \ln p(\boldsymbol{y} \,|\, \boldsymbol{\beta}, \sigma^2) = \frac{d}{\mathrm{d}\beta_i} \left[-\frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \boldsymbol{x}_n \cdot \boldsymbol{\beta})^2 \right]$$
$$= \frac{1}{\sigma^2} \sum_{n=1}^{N} (y_n - \boldsymbol{x}_n \cdot \boldsymbol{\beta}) x_i$$

- Set gradient w.r.t to $\boldsymbol{\beta}$ to zero

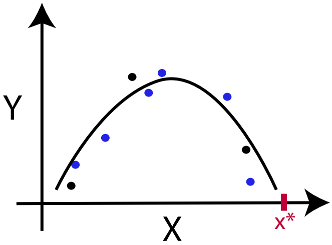
$$\nabla_{\boldsymbol{\beta}} \ln p(\boldsymbol{y} \,|\, \boldsymbol{\beta}, \sigma^2) = \frac{1}{\sigma^2} \sum_{n=1}^{N} \boldsymbol{x}_n^\top (y_n - \boldsymbol{x}_n \cdot \boldsymbol{\beta}) = \mathbf{0}$$
$$\implies \boldsymbol{\beta}_{\mathrm{ML}} = \underbrace{(\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top}_{\text{Pseudo inverse}} \boldsymbol{y}$$

- Here, the matrix $\boldsymbol{X}$ is defined as $\boldsymbol{X} = \begin{bmatrix} x_{1,1} & \dots & x1, D \\ \dots & \dots & \dots \\ x_{N,1} & \dots & x_{N,D} \end{bmatrix}$

Polynomial Curve Fitting

Motivation

- Non-linear relationships.
- Multiple SNPs playing a role for a particular phenotype.



Polynomial Curve Fitting

Univariate input x

- Use the polynomials up to degree K to construct new features from x

$$f(x,\boldsymbol{\beta}) = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_K x^K$$
$$= \sum_{k=1}^{K} \beta_k \phi_k(x) = \boldsymbol{\phi}(x)\boldsymbol{\beta}$$

where we defined $\boldsymbol{\phi}(x) = (1, x, x^2, \ldots, x^K)$.

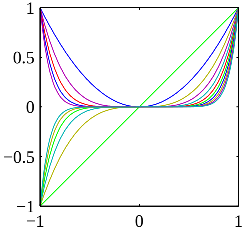
ϕ can be any feature mapping:

- $\phi_j(x) = e^x$, $\phi_j(x) = \log(x)$, ...
- Radial basis functions (also: 'Gaussian' basis functions)

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

- Sigmoidal basis functions

$$\phi_j(x) = \sigma\left(\frac{x-\mu_j}{s}\right), \quad \text{where } \sigma(a) = \frac{1}{1+\exp(-a)}$$



(C.M. Bishop, Pattern Recognition and Machine Learning)

Polynomial Curve Fitting

Univariate input x

- Use the polynomials up to degree K to construct new features from x

$$
\begin{aligned}
f(x, \boldsymbol{\beta}) &= \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_K x^K \\
&= \sum_{k=1}^{K} \beta_k \phi_k(x) = \boldsymbol{\phi}(x)\boldsymbol{\beta}
\end{aligned}
$$

where we defined $\boldsymbol{\phi}(x) = (1, x, x^2, \ldots, x^K)$.

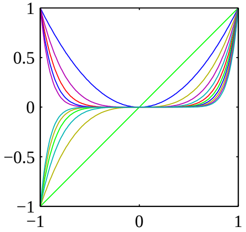
ϕ can be any feature mapping:

- $\phi_j(x) = e^x$, $\phi_j(x) = \log(x)$, $\ldots$
- Radial basis functions (also: 'Gaussian' basis functions)

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

- Sigmoidal basis functions

$$\phi_j(x) = \sigma\left(\frac{x-\mu_j}{s}\right), \quad \text{where } \sigma(a) = \frac{1}{1+\exp(-a)}$$



(C.M. Bishop, Pattern Recognition and Machine Learning)

Polynomial Curve Fitting

Univariate input x

- Use the polynomials up to degree K to construct new features from x

$$f(x, \boldsymbol{\beta}) = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_K x^K$$
$$= \sum_{k=1}^{K} \beta_k \phi_k(x) = \boldsymbol{\phi}(x)\boldsymbol{\beta}$$

where we defined $\boldsymbol{\phi}(x) = (1, x, x^2, \ldots, x^K)$.

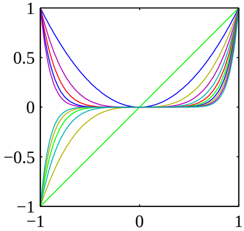
ϕ can be any feature mapping:

- $\phi_j(x) = e^x$, $\phi_j(x) = \log(x)$, ...
- Radial basis functions (also: 'Gaussian' basis functions)

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

- Sigmoidal basis functions

$$\phi_j(x) = \sigma\left(\frac{x-\mu_j}{s}\right), \quad \text{where } \sigma(a) = \frac{1}{1+\exp(-a)}$$



(C.M. Bishop, Pattern Recognition and Machine Learning)

Polynomial Curve Fitting

Univariate input x

- Use the polynomials up to degree K to construct new features from x

$$f(x, \boldsymbol{\beta}) = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_K x^K$$
$$= \sum_{k=1}^{K} \beta_k \phi_k(x) = \boldsymbol{\phi}(x)\boldsymbol{\beta}$$

where we defined $\boldsymbol{\phi}(x) = (1, x, x^2, \ldots, x^K)$.

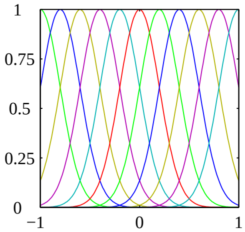
ϕ can be any feature mapping:

- $\phi_j(x) = e^x$, $\phi_j(x) = \log(x)$, ...
- Radial basis functions (also: 'Gaussian' basis functions)

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

- Sigmoidal basis functions

$$\phi_j(x) = \sigma\left(\frac{x-\mu_j}{s}\right), \quad \text{where } \sigma(a) = \frac{1}{1+\exp(-a)}$$



(C.M. Bishop, Pattern Recognition and Machine Learning)

Polynomial Curve Fitting

Univariate input x

- Use the polynomials up to degree K to construct new features from x

$$\begin{aligned} f(x, \boldsymbol{\beta}) &= \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_K x^K \\ &= \sum_{k=1}^{K} \beta_k \phi_k(x) = \boldsymbol{\phi}(x)\boldsymbol{\beta} \end{aligned}$$

where we defined $\boldsymbol{\phi}(x) = (1, x, x^2, \ldots, x^K)$.

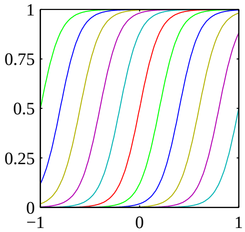
ϕ can be any feature mapping:

- $\phi_j(x) = e^x$, $\phi_j(x) = \log(x)$, ...
- Radial basis functions (also: 'Gaussian' basis functions)

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

- Sigmoidal basis functions

$$\phi_j(x) = \sigma\left(\frac{x-\mu_j}{s}\right), \quad \text{where } \sigma(a) = \frac{1}{1+\exp(-a)}$$

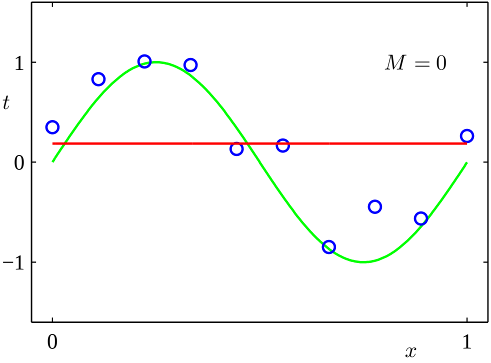


(C.M. Bishop, Pattern Recognition and Machine Learning)

Polynomial Curve Fitting

Overfitting

- The order of the polynomial M is crucial to avoid under- and overfitting.
- Observation: Variance in regression coefficients $\beta = [w_0^\star, \dots, w_9\star]$ grows dramatically with M

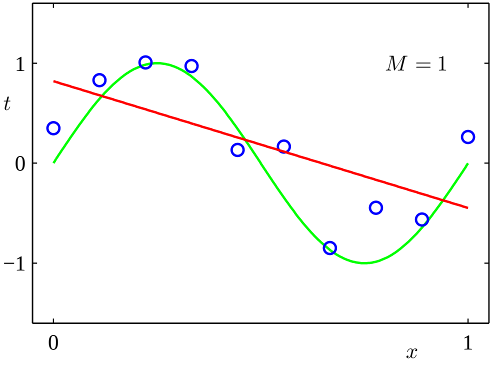


(C.M. Bishop, Pattern Recognition and Machine Learning)

Polynomial Curve Fitting

Overfitting

- The order of the polynomial M is crucial to avoid under- and overfitting.
- Observation: Variance in regression coefficients $\beta = [w_0^\star, \ldots, w_9\star]$ grows dramatically with M

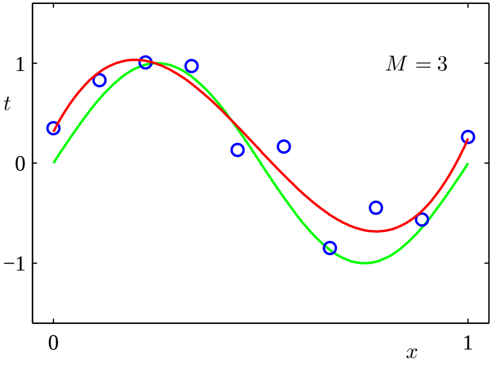


(C.M. Bishop, Pattern Recognition and Machine Learning)

Polynomial Curve Fitting

Overfitting

- The order of the polynomial M is crucial to avoid under- and overfitting.
- Observation: Variance in regression coefficients $\beta = [w_0^\star, \dots, w_9\star]$ grows dramatically with M

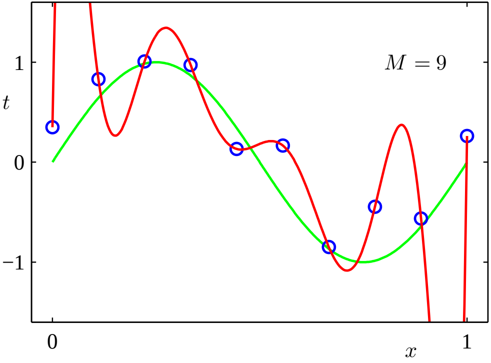


(C.M. Bishop, Pattern Recognition and Machine Learning)

Polynomial Curve Fitting

Overfitting

- The order of the polynomial M is crucial to avoid under- and overfitting.
- Observation: Variance in regression coefficients $\beta = [w_0^\star, \ldots, w_9\star]$ grows dramatically with M



(C.M. Bishop, Pattern Recognition and Machine Learning)

Polynomial Curve Fitting

Overfitting

- The order of the polynomial M is crucial to avoid under- and overfitting.
- Observation: Variance in regression coefficients $\boldsymbol{\beta} = [w_0^\star, \ldots, w_9\star]$ grows dramatically with M

	$M = 0$	$M = 1$	$M = 6$	$M = 9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

(C.M. Bishop, Pattern Recognition and Machine Learning)

Polynomial Curve Fitting

Generalization performance

$$E(\boldsymbol{\beta}) = \frac{1}{2} \sum_{n=1}^{N} (y_n - \boldsymbol{x}_n \cdot \boldsymbol{\beta})^2$$

- Root-mean squared error

$$\begin{aligned} E_{\mathrm{RMS}} &= \sqrt{2E(\boldsymbol{\beta})/N} \\ &= \sqrt{\left(\sum_{n=1}^{N} (y_n - \boldsymbol{x}_n \cdot \boldsymbol{\beta})^2\right)/N} \end{aligned}$$

- Can be measured on training data
- Or when predicting new data (test data)
- **Underfitting**: large E_{RMS} on train and test data
- **Overfitting**: small E_{RMS} on train and large E_{RMS} on test data.

Polynomial Curve Fitting

Generalization performance

$$E(\boldsymbol{\beta}) = \frac{1}{2}\sum_{n=1}^{N}(y_n - \boldsymbol{x}_n \cdot \boldsymbol{\beta})^2$$

- Root-mean squared error

$$\begin{aligned} E_{\mathrm{RMS}} &= \sqrt{2E(\boldsymbol{\beta})/N} \\ &= \sqrt{\left(\sum_{n=1}^{N}(y_n - \boldsymbol{x}_n \cdot \boldsymbol{\beta})^2\right)/N} \end{aligned}$$

- Can be measured on training data
- Or when predicting new data (test data)
- **Underfitting**: large E_{RMS} on train and test data
- **Overfitting**: small E_{RMS} on train and large E_{RMS} on test data.

Polynomial Curve Fitting

Generalization performance

$$E(\boldsymbol{\beta}) = \frac{1}{2}\sum_{n=1}^{N}(y_n - \boldsymbol{x}_n \cdot \boldsymbol{\beta})^2$$

- Root-mean squared error

$$\begin{aligned} E_{\mathrm{RMS}} &= \sqrt{2E(\boldsymbol{\beta})/N} \\ &= \sqrt{\left(\sum_{n=1}^{N}(y_n - \boldsymbol{x}_n \cdot \boldsymbol{\beta})^2\right)/N} \end{aligned}$$

- Can be measured on training data
- Or when predicting new data (test data)
- **Underfitting**: large E_{RMS} on train and test data
- **Overfitting**: small E_{RMS} on train and large E_{RMS} on test data.

Polynomial Curve Fitting

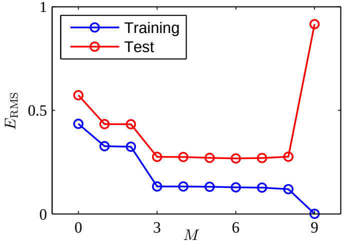
Generalization performance

$$E(\boldsymbol{\beta}) = \frac{1}{2}\sum_{n=1}^{N}(y_n - \boldsymbol{x}_n \cdot \boldsymbol{\beta})^2$$

- Root-mean squared error

$$E_{\mathrm{RMS}} = \sqrt{2E(\boldsymbol{\beta})/N} = \sqrt{\left(\sum_{n=1}^{N}(y_n - \boldsymbol{x}_n \cdot \boldsymbol{\beta})^2\right)/N}$$

- Can be measured on training data
- Or when predicting new data (test data)
- **Underfitting**: large E_{RMS} on train and test data
- **Overfitting**: small E_{RMS} on train and large E_{RMS} on test data.



(C.M. Bishop, Pattern Recognition and Machine Learning)

Polynomial Curve Fitting

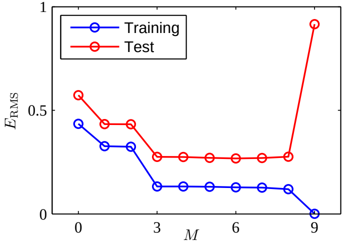
Generalization performance

$$E(\boldsymbol{\beta}) = \frac{1}{2}\sum_{n=1}^{N}(y_n - \boldsymbol{x}_n \cdot \boldsymbol{\beta})^2$$

- Root-mean squared error

$$E_{\mathrm{RMS}} = \sqrt{2E(\boldsymbol{\beta})/N} = \sqrt{\left(\sum_{n=1}^{N}(y_n - \boldsymbol{x}_n \cdot \boldsymbol{\beta})^2\right)/N}$$

- Can be measured on training data
- Or when predicting new data (test data)
- **Underfitting**: large E_{RMS} on train and test data
- **Overfitting**: small E_{RMS} on train and large E_{RMS} on test data.



(C.M. Bishop, Pattern Recognition and Machine Learning)

Polynomial Curve Fitting

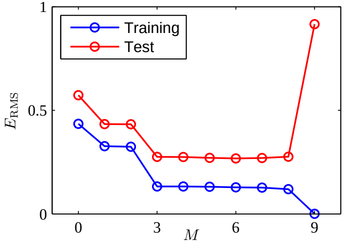
Generalization performance

$$E(\boldsymbol{\beta}) = \frac{1}{2}\sum_{n=1}^{N}(y_n - \boldsymbol{x}_n \cdot \boldsymbol{\beta})^2$$

- Root-mean squared error

$$E_{\mathrm{RMS}} = \sqrt{2E(\boldsymbol{\beta})/N}$$
$$= \sqrt{\left(\sum_{n=1}^{N}(y_n - \boldsymbol{x}_n \cdot \boldsymbol{\beta})^2\right)/N}$$

- Can be measured on training data
- Or when predicting new data (test data)
- **Underfitting**: large E_{RMS} on train and test data
- **Overfitting**: small E_{RMS} on train and large E_{RMS} on test data.

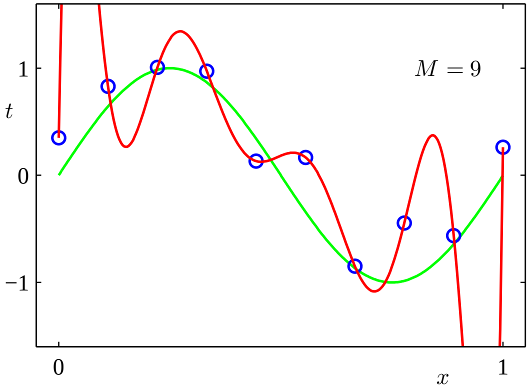


(C.M. Bishop, Pattern Recognition and Machine Learning)

Polynomial Curve Fitting

Overfitting

- The number N of training data is crucial to accurately estimate many parameters without overfitting.

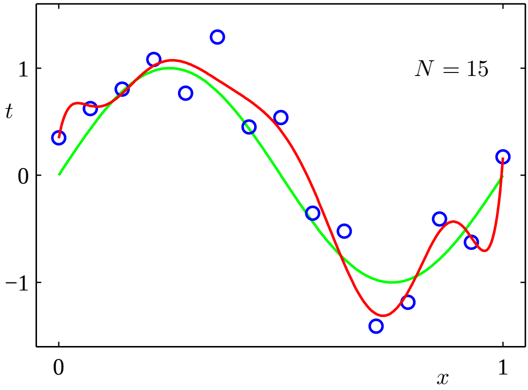


(C.M. Bishop, Pattern Recognition and Machine Learning)

Polynomial Curve Fitting

Overfitting

- The number N of training data is crucial to accurately estimate many parameters without overfitting.

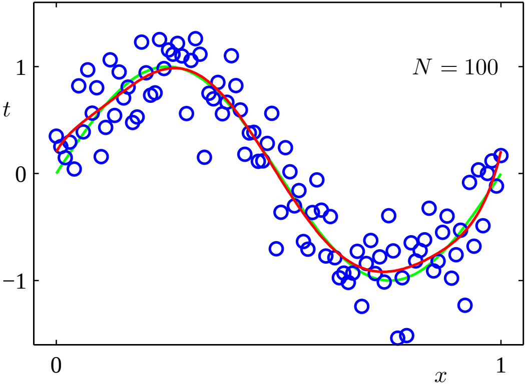


(C.M. Bishop, Pattern Recognition and Machine Learning)

Polynomial Curve Fitting

Overfitting

- The number N of training data is crucial to accurately estimate many parameters without overfitting.



(C.M. Bishop, Pattern Recognition and Machine Learning)

Multivariate regression

Polynomial curve fitting

$$
\begin{aligned}
f(x, \boldsymbol{\beta}) &= \beta_0 + \beta_1 x + \cdots + \beta_K x^K \\
&= \sum_{k=1}^{K} \beta_k \phi_k(x) \\
&= \boldsymbol{\phi}(x) \cdot \boldsymbol{\beta},
\end{aligned}
$$

High dimensional regression

$$
\begin{aligned}
f(x, \boldsymbol{\beta}) &= \sum_{d=1}^{D} \beta_d x_d \\
&= \boldsymbol{x} \cdot \boldsymbol{\beta}
\end{aligned}
$$

- Note: When fitting a single binary variable x_i, a linear model is most general!

Multivariate regression

Polynomial curve fitting

$$
\begin{aligned}
f(x, \boldsymbol{\beta}) &= \beta_0 + \beta_1 x + \cdots + \beta_K x^K \\
&= \sum_{k=1}^{K} \beta_k \phi_k(x) \\
&= \boldsymbol{\phi}(x) \cdot \boldsymbol{\beta},
\end{aligned}
$$

High dimensional regression

$$
\begin{aligned}
f(x, \boldsymbol{\beta}) &= \sum_{d=1}^{D} \beta_d x_d \\
&= \boldsymbol{x} \cdot \boldsymbol{\beta}
\end{aligned}
$$

- Note: When fitting a single binary variable $\boldsymbol{x}_i$, a linear model is most general!

Regularized Least Squares

Ridge regression

- Solutions to avoid overfitting:
  1. Intelligently choose number of parameters
  2. Get more data
  3. Regularize the regression weights β
- Quadratically regularized objective function

$$E(\beta) = \underbrace{\frac{1}{2}\sum_{n=1}^{N}(y_n - \phi(x_n)\cdot\beta)^2}_{\text{Squared error}} + \underbrace{\frac{\lambda}{2}\beta^\top\beta}_{\text{Regularizer}}$$

Regularized Least Squares

Ridge regression

- Solutions to avoid overfitting:
    1. Intelligently choose number of parameters
    2. Get more data
    3. Regularize the regression weights β
- Quadratically regularized objective function

$$E(\beta) = \underbrace{\frac{1}{2}\sum_{n=1}^{N}(y_n - \phi(x_n)\cdot\beta)^2}_{\text{Squared error}} + \underbrace{\frac{\lambda}{2}\beta^\top\beta}_{\text{Regularizer}}$$

Regularized Least Squares

Ridge regression

- Solutions to avoid overfitting:
  1. Intelligently choose number of parameters
  2. Get more data
  3. Regularize the regression weights $\boldsymbol{\beta}$
- Quadratically regularized objective function

$$E(\boldsymbol{\beta}) = \underbrace{\frac{1}{2}\sum_{n=1}^{N}(y_n - \phi(\boldsymbol{x}_n)\cdot\boldsymbol{\beta})^2}_{\text{Squared error}} + \underbrace{\frac{\lambda}{2}\boldsymbol{\beta}^\top\boldsymbol{\beta}}_{\text{Regularizer}}$$

Regularized Least Squares

Ridge regression

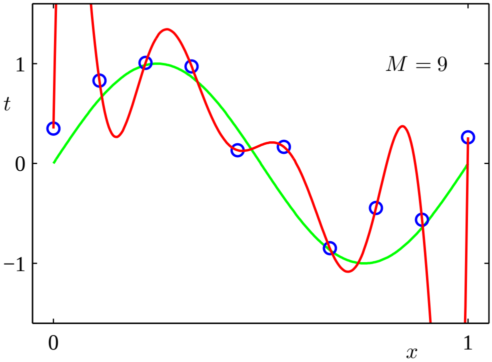
- Solutions to avoid overfitting:
  1. Intelligently choose number of parameters
  2. Get more data
  3. Regularize the regression weights $\boldsymbol{\beta}$
- Quadratically regularized objective function

$$E(\boldsymbol{\beta}) = \underbrace{\frac{1}{2}\sum_{n=1}^{N}(y_n - \boldsymbol{\phi}(\boldsymbol{x}_n)\cdot\boldsymbol{\beta})^2}_{\text{Squared error}} + \underbrace{\frac{\lambda}{2}\boldsymbol{\beta}^\top\boldsymbol{\beta}}_{\text{Regularizer}}$$

Polynomial curve fitting

L_2 regularization

- $M = 9$, different λ values

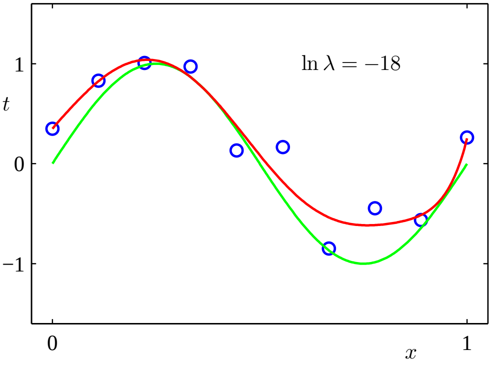


(C.M. Bishop, Pattern Recognition and Machine Learning)

Polynomial curve fitting

L_2 regularization

- $M = 9$, different λ values

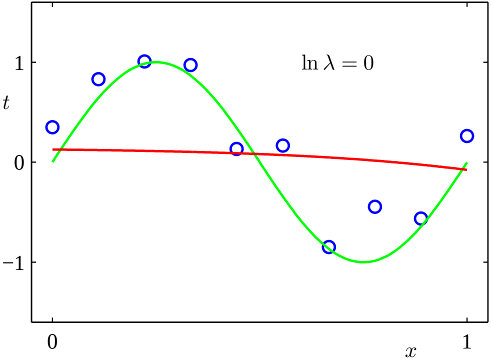


(C.M. Bishop, Pattern Recognition and Machine Learning)

Polynomial curve fitting

L_2 regularization

- $M = 9$, different λ values



(C.M. Bishop, Pattern Recognition and Machine Learning)

Polynomial curve fitting

L_2 regularization

- $M = 9$, different λ values

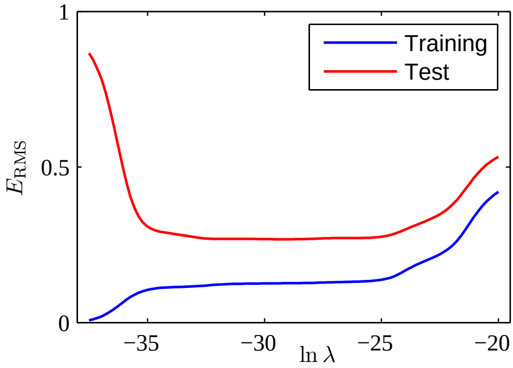
	$\ln \lambda = -\infty$	$\ln \lambda = -18$	$\ln \lambda = 0$
$w_0^\star$	0.35	0.35	0.13
$w_1^\star$	232.37	4.74	-0.05
$w_2^\star$	-5321.83	-0.77	-0.06
$w_3^\star$	48568.31	-31.97	-0.05
$w_4^\star$	-231639.30	-3.89	-0.03
$w_5^\star$	640042.26	55.28	-0.02
$w_6^\star$	-1061800.52	41.32	-0.01
$w_7^\star$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

(C.M. Bishop, Pattern Recognition and Machine Learning)

Polynomial curve fitting

L_2 regularization

- $M = 9$, different λ values



(C.M. Bishop, Pattern Recognition and Machine Learning)

Bias-variance tradeoff

Variance of f^{est}

$$\mathbb{E}_{\mathcal{D}}\left[f^{\text{est}} - \mathbb{E}\left[f^{\text{est}}\right]^2\right]$$

Bias of f^{est}

$$\mathbb{E}_{\mathcal{D}}\left[f^{\text{true}} - f^{\text{est}}\right]$$

Experiment:

- 100 random data sets $(N = 25)$
- learn 25 RBF basis functions
- vary λ

Empirical Observations:

- Bias decreases with smaller λ
- Variance increases with smaller λ

(C.M. Bishop, Pattern Recognition and Machine Learning)

Bias-variance tradeoff

Variance of f^{est}

$$\mathbb{E}_{\mathcal{D}}\left[f^{\text{est}} - \mathbb{E}\left[f^{\text{est}}\right]^2\right]$$

Bias of f^{est}

$$\mathbb{E}_{\mathcal{D}}\left[f^{\text{true}} - f^{\text{est}}\right]$$

Experiment:

- 100 random data sets $(N = 25)$
- learn 25 RBF basis functions
- vary λ

Empirical Observations:

- Bias decreases with smaller λ
- Variance increases with smaller λ

(C.M. Bishop, Pattern Recognition and Machine Learning)

Bias-variance tradeoff

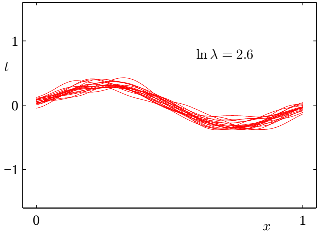
Variance of f^{est}

$$\mathbb{E}_{\mathcal{D}}\left[\left(f^{\text{est}} - \mathbb{E}\left[f^{\text{est}}\right]\right)^2\right]$$

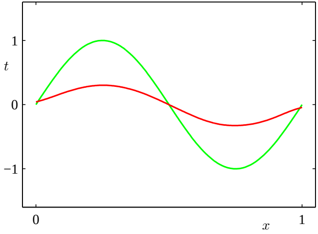
Bias of f^{est}

$$\mathbb{E}_{\mathcal{D}}\left[f^{\text{true}} - f^{\text{est}}\right]$$

Estimated functions



Mean over all estimates



Experiment:

- 100 random data sets ($N = 25$)
- learn 25 RBF basis functions
- vary λ

Empirical Observations:

- Bias decreases with smaller λ
- Variance increases with smaller λ

(C.M. Bishop, Pattern Recognition and Machine Learning)

Bias-variance tradeoff

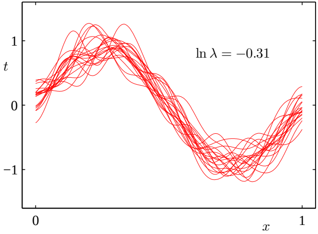
Variance of f^{est}

$$\mathbb{E}_{\mathcal{D}}\left[f^{\text{est}} - \mathbb{E}\left[f^{\text{est}}\right]^2\right]$$

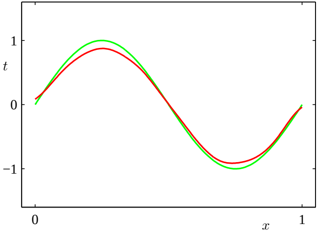
Bias of f^{est}

$$\mathbb{E}_{\mathcal{D}}\left[f^{\text{true}} - f^{\text{est}}\right]$$

Estimated functions



Mean over all estimates



Experiment:

- 100 random data sets $(N = 25)$
- learn 25 RBF basis functions
- vary λ

Empirical Observations:

- Bias decreases with smaller λ

(C.M. Bishop, Pattern Recognition and Machine Learning)

Bias-variance tradeoff

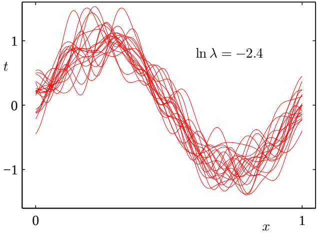
Variance of f^{est}

$$\mathbb{E}_{\mathcal{D}}\left[f^{\text{est}} - \mathbb{E}\left[f^{\text{est}}\right]^2\right]$$

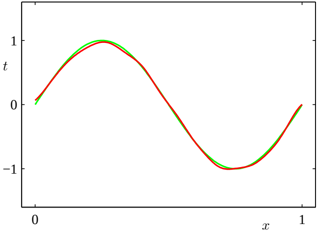
Bias of f^{est}

$$\mathbb{E}_{\mathcal{D}}\left[f^{\text{true}} - f^{\text{est}}\right]$$

Estimated functions



Mean over all estimates



Experiment:

- 100 random data sets $(N = 25)$
- learn 25 RBF basis functions
- vary λ

Empirical Observations:

- Bias decreases with smaller λ
- Variance increases with smaller λ

(C.M. Bishop, Pattern Recognition and Machine Learning)

Bias-variance tradeoff

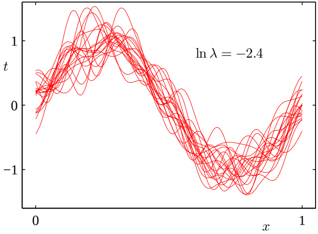
Variance of f^{est}

$$\mathbb{E}_{\mathcal{D}}\left[f^{\text{est}} - \mathbb{E}\left[f^{\text{est}}\right]^2\right]$$

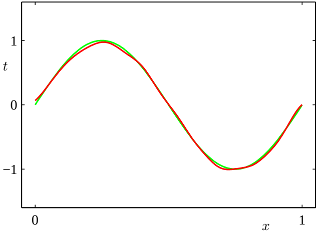
Bias of f^{est}

$$\mathbb{E}_{\mathcal{D}}\left[f^{\text{true}} - f^{\text{est}}\right]$$

Estimated functions



Mean over all estimates



Experiment:

- 100 random data sets ($N = 25$)
- learn 25 RBF basis functions
- vary λ

Empirical Observations:

- **Bias** decreases with smaller λ
- **Variance** increases with smaller λ

(C.M. Bishop, Pattern Recognition and Machine Learning)

Bias-variance tradeoff

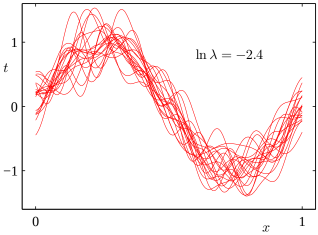
Variance of f^{est}

$$\mathbb{E}_{\mathcal{D}}\left[f^{\text{est}} - \mathbb{E}\left[f^{\text{est}}\right]^2\right]$$

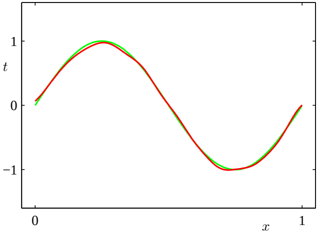
Bias of f^{est}

$$\mathbb{E}_{\mathcal{D}}\left[f^{\text{true}} - f^{\text{est}}\right]$$

Estimated functions



Mean over all estimates



Experiment:

- 100 random data sets $(N = 25)$
- learn 25 RBF basis functions
- vary λ

Empirical Observations:

- **Bias** decreases with smaller λ
- **Variance** increases with smaller λ

(C.M. Bishop, Pattern Recognition and Machine Learning)

Bias-variance tradeoff

Effect on mean squared error

$$y_n = f^{\text{true}}(x_n) + \epsilon_n$$

$$\begin{aligned}\text{mean squared error}(f^{\text{est}}) &= \mathbb{E}_{\mathcal{D}}\left[(y - f^{\text{est}}(x))^2\right] \\ &= (\text{bias})^2 + \text{variance} + \text{noise}\end{aligned}$$

Experiment as before:

- 100 random data sets $(N = 25)$
- learn 25 RBF basis functions
- vary λ
- Compute sample estimates of bias and variance
- Additionally 1000 test data points to estimate mean-squared error

Bias-variance tradeoff

Effect on mean squared error

$$y_n = f^{\text{true}}(x_n) + \epsilon_n$$

$$\begin{aligned}\text{mean squared error}(f^{\text{est}}) &= \mathbb{E}_{\mathcal{D}}\left[(y - f^{\text{est}}(x))^2\right] \\ &= (\text{bias})^2 + \text{variance} + \text{noise}\end{aligned}$$

Experiment as before:

- 100 random data sets ($N = 25$)
- learn 25 RBF basis functions
- vary λ
- Compute sample estimates of bias and variance
- Additionally 1000 test data points to estimate mean-squared error

Bias-variance tradeoff

Effect on mean squared error

$$y_n = f^{\text{true}}(x_n) + \epsilon_n$$

$$\begin{aligned}\text{mean squared error}(f^{\text{est}}) &= \mathbb{E}_{\mathcal{D}}\left[(y - f^{\text{est}}(x))^2\right] \\ &= (\text{bias})^2 + \text{variance} + \text{noise}\end{aligned}$$

Experiment as before:

- 100 random data sets $(N = 25)$
- learn 25 RBF basis functions
- vary λ
- Compute sample estimates of bias and variance
- Additionally 1000 test data points to estimate mean-squared error

Bias-variance tradeoff

Effect on mean squared error

$$y_n = f^{\text{true}}(x_n) + \epsilon_n$$

$$\begin{aligned}\text{mean squared error}(f^{\text{est}}) &= \mathbb{E}_{\mathcal{D}}\left[(y - f^{\text{est}}(x))^2\right] \\ &= (\text{bias})^2 + \text{variance} + \text{noise}\end{aligned}$$

Experiment as before:

- 100 random data sets $(N = 25)$
- learn 25 RBF basis functions
- vary λ
- Compute sample estimates of bias and variance
- Additionally 1000 test data points to estimate mean-squared error

Bias-variance tradeoff

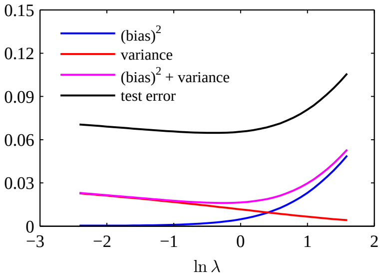
Effect on mean squared error

$$y_n = f^{\text{true}}(x_n) + \epsilon_n$$

$$\begin{aligned}\text{mean squared error}(f^{\text{est}}) &= \mathbb{E}_{\mathcal{D}}\left[(y - f^{\text{est}}(x))^2\right] \\ &= (\text{bias})^2 + \text{variance} + \text{noise}\end{aligned}$$

Experiment as before:

- 100 random data sets ($N = 25$)
- learn 25 RBF basis functions
- vary λ
- Compute sample estimates of bias and variance
- Additionally 1000 test data points to estimate mean-squared error



(C.M. Bishop, Pattern Recognition and Machine Learning)

Regularized Least Squares

More general regularizers

- More general regularization:

$$E(\boldsymbol{\beta}) = \underbrace{\frac{1}{2}\sum_{n=1}^{N}(y_n - \boldsymbol{\phi}(\boldsymbol{x}_n)\cdot\boldsymbol{\beta})^2}_{\text{Squared error}} + \underbrace{\frac{\lambda}{2}\sum_{d=1}^{D}|\beta_d|^{q}}_{\text{Regularizer}}$$

- $q \leq 1$: non-differentiable
- $q < 1$: non-convex (could have local optima)

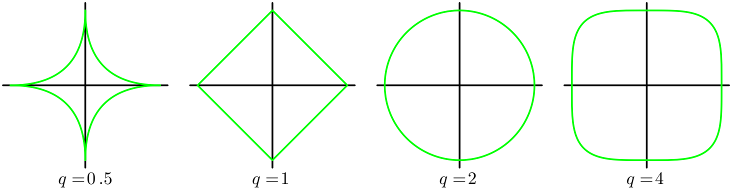
Regularized Least Squares

More general regularizers

- More general regularization:

$$E(\boldsymbol{\beta}) = \underbrace{\frac{1}{2}\sum_{n=1}^{N}(y_n - \boldsymbol{\phi}(\boldsymbol{x}_n)\cdot\boldsymbol{\beta})^2}_{\text{Squared error}} + \underbrace{\frac{\lambda}{2}\sum_{d=1}^{D}|\beta_d|^{q}}_{\text{Regularizer}}$$

- $q \leq 1$: non-differentiable
- $q < 1$: non-convex (could have local optima)



(C.M. Bishop, Pattern Recognition and Machine Learning)

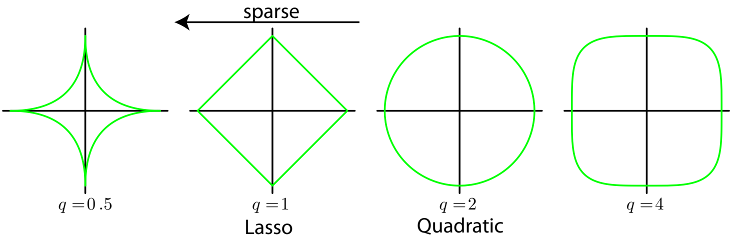
Regularized Least Squares

More general regularizers

- More general regularization:

$$E(\boldsymbol{\beta}) = \underbrace{\frac{1}{2}\sum_{n=1}^{N}(y_n - \boldsymbol{\phi}(\boldsymbol{x}_n)\cdot\boldsymbol{\beta})^2}_{\text{Squared error}} + \underbrace{\frac{\lambda}{2}\sum_{d=1}^{D}|\beta_d|^{q}}_{\text{Regularizer}}$$

- $q \leq 1$: non-differentiable
- $q < 1$: non-convex (could have local optima)



(C.M. Bishop, Pattern Recognition and Machine Learning)

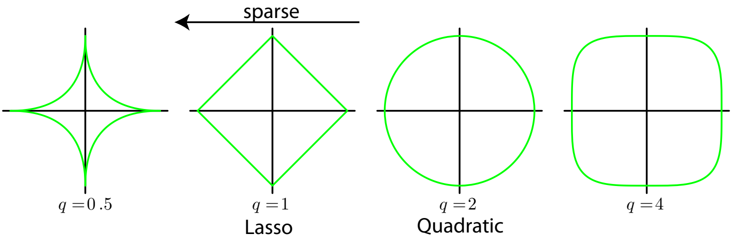
Regularized Least Squares

More general regularizers

- More general regularization:

$$E(\boldsymbol{\beta}) = \underbrace{\frac{1}{2}\sum_{n=1}^{N}(y_n - \boldsymbol{\phi}(\boldsymbol{x}_n)\cdot\boldsymbol{\beta})^2}_{\text{Squared error}} + \underbrace{\frac{\lambda}{2}\sum_{d=1}^{D}|\beta_d|^{q}}_{\text{Regularizer}}$$

- $q \leq 1$: non-differentiable
- $q < 1$: non-convex (could have local optima)



(C.M. Bishop, Pattern Recognition and Machine Learning)

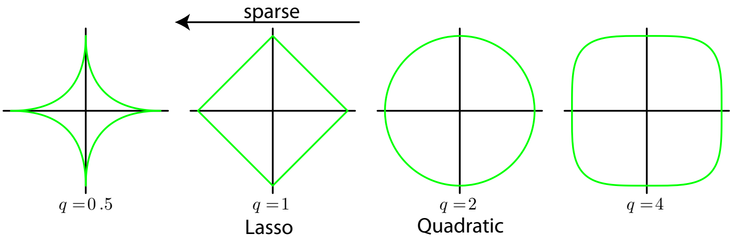
Regularized Least Squares

More general regularizers

- More general regularization:

$$E(\boldsymbol{\beta}) = \underbrace{\frac{1}{2}\sum_{n=1}^{N}(y_n - \boldsymbol{\phi}(\boldsymbol{x}_n)\cdot\boldsymbol{\beta})^2}_{\text{Squared error}} + \underbrace{\frac{\lambda}{2}\sum_{d=1}^{D}|\beta_d|^{q}}_{\text{Regularizer}}$$

- $q \leq 1$: non-differentiable
- $q < 1$: non-convex (could have local optima)

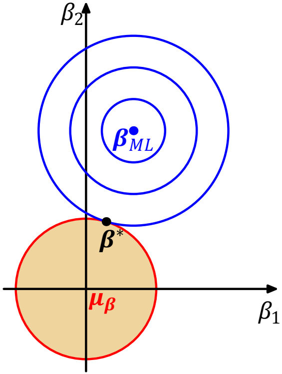


(C.M. Bishop, Pattern Recognition and Machine Learning)

Smaller q yields sparser solution $\boldsymbol{\beta}^{\star}$

- $q = 2$: Ridge regression (L_2)
- $q = 1$: Lasso (L_1)

- Squared error
- Regularizer

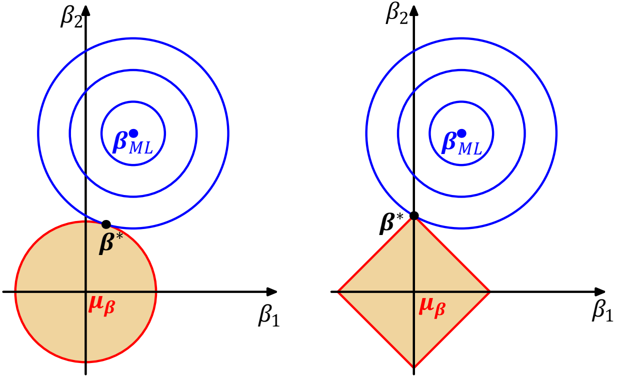


(C.M. Bishop, Pattern Recognition and Machine Learning)

Smaller q yields sparser solution $\boldsymbol{\beta}^\star$

- $q = 2$: Ridge regression (L_2)
- $q = 1$: Lasso (L_1)

- Squared error
- Regularizer



(C.M. Bishop, Pattern Recognition and Machine Learning)

Loss functions and related methods

- Even more general: general loss function

$$E(\boldsymbol{\beta}) = \underbrace{\frac{1}{2}\sum_{n=1}^{N}\mathcal{L}(y_n - \boldsymbol{\phi}(\boldsymbol{x}_n)\cdot\boldsymbol{\beta})}_{\text{Loss}} + \underbrace{\frac{\lambda}{2}\sum_{d=1}^{D}|\beta_d|^q}_{\text{Regularizer}}$$

- Many state-of-the-art machine learning methods can be expressed within this framework.
  - Linear Regression: squared loss, squared regularizer.
  - Support Vector Machine: hinge loss, squared regularizer.
  - Lasso: squared loss, L1 regularizer.
- Inference: minimize the cost function $E(\beta)$, yielding a point estimate for β.
- Q: How to determine q and the a suitable loss function?

Loss functions and related methods

- Even more general: general loss function

$$E(\boldsymbol{\beta}) = \underbrace{\frac{1}{2}\sum_{n=1}^{N}\mathcal{L}(y_n - \boldsymbol{\phi}(\boldsymbol{x}_n)\cdot\boldsymbol{\beta})}_{\text{Loss}} + \underbrace{\frac{\lambda}{2}\sum_{d=1}^{D}|\beta_d|^q}_{\text{Regularizer}}$$

- Many state-of-the-art machine learning methods can be expressed within this framework.
  - Linear Regression: squared loss, squared regularizer.
  - Support Vector Machine: hinge loss, squared regularizer.
  - Lasso: squared loss, L1 regularizer.
- Inference: minimize the cost function $E(\beta)$, yielding a point estimate for β.
- Q: How to determine q and the a suitable loss function?

Loss functions and related methods

- Even more general: general loss function

$$E(\boldsymbol{\beta}) = \underbrace{\frac{1}{2}\sum_{n=1}^{N}\mathcal{L}(y_n - \boldsymbol{\phi}(\boldsymbol{x}_n)\cdot\boldsymbol{\beta})}_{\text{Loss}} + \underbrace{\frac{\lambda}{2}\sum_{d=1}^{D}|\beta_d|^{q}}_{\text{Regularizer}}$$

- Many state-of-the-art machine learning methods can be expressed within this framework.
  - Linear Regression: squared loss, squared regularizer.
  - Support Vector Machine: hinge loss, squared regularizer.
  - Lasso: squared loss, L1 regularizer.
- Inference: minimize the cost function $E(\boldsymbol{\beta})$, yielding a point estimate for $\boldsymbol{\beta}$.
- Q: How to determine q and the a suitable loss function?

Loss functions and related methods

- Even more general: general loss function

$$E(\boldsymbol{\beta}) = \underbrace{\frac{1}{2}\sum_{n=1}^{N}\mathcal{L}(y_n - \boldsymbol{\phi}(\boldsymbol{x}_n)\cdot\boldsymbol{\beta})}_{\text{Loss}} + \underbrace{\frac{\lambda}{2}\sum_{d=1}^{D}|\beta_d|^{q}}_{\text{Regularizer}}$$

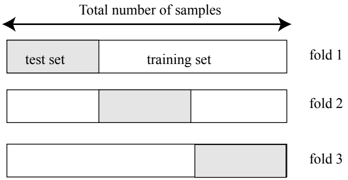
- Many state-of-the-art machine learning methods can be expressed within this framework.
  - Linear Regression: squared loss, squared regularizer.
  - Support Vector Machine: hinge loss, squared regularizer.
  - Lasso: squared loss, L1 regularizer.
- Inference: minimize the cost function $E(\boldsymbol{\beta})$, yielding a point estimate for $\boldsymbol{\beta}$.
- Q: How to determine q and the a suitable loss function?

Loss functions and related methods

Cross validation: minimization of expected loss

Compare candidate models $\mathcal{H}$ on generalization performance
(different λ, different regularizers, different basis functions, etc.)

- Randomly split data into K sets of equal size
- For each fold k:
  1. Train on all data except the k^{th} set
  2. Test evaluation on k^{th} set
- Assess average loss on test sets
  $\frac{1}{K}\sum_{k=1}^{K} E_k^{\text{test}}(\mathcal{H})$
- Pick model $\mathcal{H}$ with lowest average loss
- Re-train optimal $\mathcal{H}$ on all data

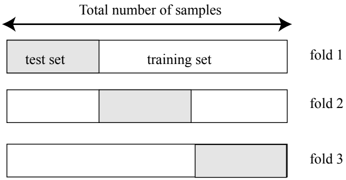


Loss functions and related methods

Cross validation: minimization of expected loss

Compare candidate models $\mathcal{H}$ on generalization performance
(different λ, different regularizers, different basis functions, etc.)

- Randomly split data into K sets of equal size
- For each fold k:
  1. Train on all data except the k^{th} set
  2. Test performance on k^{th} set
- Assess average loss on test sets
  $\frac{1}{K}\sum_{k=1}^{K} E_k^{\text{test}}(\mathcal{H})$
- Pick model $\mathcal{H}$ with lowest average loss
- Re-train optimal $\mathcal{H}$ on all data



Loss functions and related methods

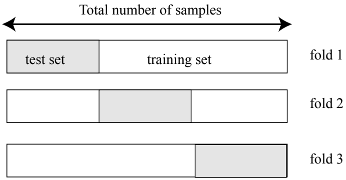
Cross validation: minimization of expected loss

Compare candidate models $\mathcal{H}$ on generalization performance (different λ, different regularizers, different basis functions, etc.)

- Randomly split data into K sets of equal size
- For each fold k:
  1. Train on all data except the k^{th} set
  2. Test evaluation on k^{th} set
- Assess average loss on test sets

$$\frac{1}{K}\sum_{k=1}^{K} E_k^{\text{test}}(\mathcal{H})$$

- Pick model $\mathcal{H}$ with lowest average loss
- Re-train optimal $\mathcal{H}$ on all data

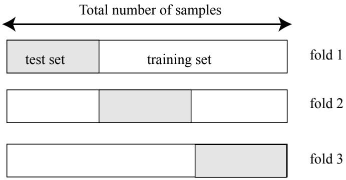


Loss functions and related methods

Cross validation: minimization of expected loss

Compare candidate models $\mathcal{H}$ on generalization performance
(different λ, different regularizers, different basis functions, etc.)

- Randomly split data into K sets of equal size
- For each fold k:
  1. Train on all data except the k^{th} set
  2. Test evaluation on k^{th} set
- Assess average loss on test sets
  $$\frac{1}{K}\sum_{k=1}^{K} E_k^{\text{test}}(\mathcal{H})$$
- Pick model $\mathcal{H}$ with lowest average loss
- Re-train optimal $\mathcal{H}$ on all data

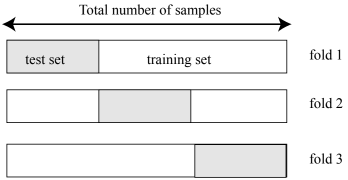


Loss functions and related methods

Cross validation: minimization of expected loss

Compare candidate models $\mathcal{H}$ on generalization performance
(different λ, different regularizers, different basis functions, etc.)

- Randomly split data into K sets of equal size
- For each fold k:
  1. Train on all data except the k^{th} set
  2. Test evaluation on k^{th} set



- Assess average loss on test sets
  $$\frac{1}{K}\sum_{k=1}^{K} E_k^{\text{test}}(\mathcal{H})$$
- Pick model $\mathcal{H}$ with lowest average loss
- Re-train optimal $\mathcal{H}$ on all data

Loss functions and related methods

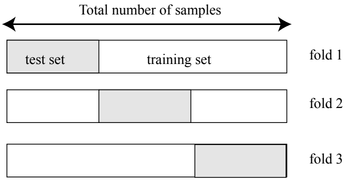
Cross validation: minimization of expected loss

Compare candidate models $\mathcal{H}$ on generalization performance
(different λ, different regularizers, different basis functions, etc.)

- Randomly split data into K sets of equal size
- For each fold k:
  1. Train on all data except the k^{th} set
  2. Test evaluation on k^{th} set
- Assess average loss on test sets

$$\frac{1}{K}\sum_{k=1}^{K} E_k^{\text{test}}(\mathcal{H})$$

- Pick model $\mathcal{H}$ with lowest average loss
- Re-train optimal $\mathcal{H}$ on all data

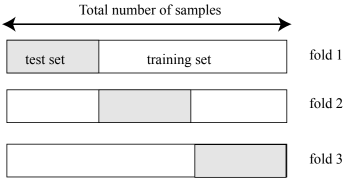


Loss functions and related methods

Cross validation: minimization of expected loss

Compare candidate models $\mathcal{H}$ on generalization performance
(different λ, different regularizers, different basis functions, etc.)

- Randomly split data into K sets of equal size
- For each fold k:
  1. Train on all data except the k^{th} set
  2. Test evaluation on k^{th} set
- Assess average loss on test sets
  $$\frac{1}{K}\sum_{k=1}^{K} E_k^{\text{test}}(\mathcal{H})$$
- Pick model $\mathcal{H}$ with lowest average loss
- Re-train optimal $\mathcal{H}$ on all data

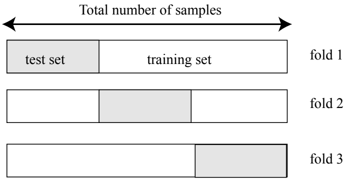


Loss functions and related methods

Cross validation: minimization of expected loss

Compare candidate models $\mathcal{H}$ on generalization performance (different λ, different regularizers, different basis functions, etc.)

- Randomly split data into K sets of equal size
- For each fold k:
  1. Train on all data except the k^{th} set
  2. Test evaluation on k^{th} set
- Assess average loss on test sets $\frac{1}{K}\sum_{k=1}^{K} E_k^{\text{test}}(\mathcal{H})$
- Pick model $\mathcal{H}$ with lowest average loss
- Re-train optimal $\mathcal{H}$ on all data



Probabilistic interpretation

- So far: minimization of error functions.
- Back to probabilities?

$$E(\boldsymbol{\beta}) = \underbrace{\frac{1}{2}\sum_{n=1}^{N}(y_n - \boldsymbol{\phi}(\boldsymbol{x}_n)\cdot\boldsymbol{\beta})^2}_{\text{Squared error}} + \underbrace{\frac{\lambda}{2}\boldsymbol{\beta}^\top\boldsymbol{\beta}}_{\text{Regularizer}}$$

- Regularized regression equivalent to MAP estimation
- Most alternative choices of regularizers and loss functions can be mapped to an equivalent probabilistic representation in a similar way.

Probabilistic interpretation

- So far: minimization of error functions.
- Back to probabilities?

$$E(\boldsymbol{\beta}) = \underbrace{\frac{1}{2}\sum_{n=1}^{N}(y_n - \boldsymbol{\phi}(\boldsymbol{x}_n)\cdot\boldsymbol{\beta})^2}_{\text{Squared error}} + \underbrace{\frac{\lambda}{2}\boldsymbol{\beta}^\top\boldsymbol{\beta}}_{\text{Regularizer}}$$

$$= \quad const. - \sum_{n=1}^{N}\ln\mathcal{N}\left(y_n \mid \boldsymbol{\phi}(\boldsymbol{x}_n)\cdot\boldsymbol{\beta}, \sigma^2\right) \quad - \ln\mathcal{N}\left(\boldsymbol{\beta} \mid \mathbf{0}, \frac{1}{\lambda}\boldsymbol{I}\right)$$

- Regularized regression equivalent to MAP estimation
- Most alternative choices of regularizers and loss functions can be mapped to an equivalent probabilistic representation in a similar way.

Probabilistic interpretation

- So far: minimization of error functions.
- Back to probabilities?

$$
\begin{aligned}
E(\boldsymbol{\beta}) = & \underbrace{\frac{1}{2}\sum_{n=1}^{N}(y_n - \boldsymbol{\phi}(\boldsymbol{x}_n)\cdot\boldsymbol{\beta})^2}_{\text{Squared error}} && + \underbrace{\frac{\lambda}{2}\boldsymbol{\beta}^\top\boldsymbol{\beta}}_{\text{Regularizer}} \\
= & \; const. - \sum_{n=1}^{N} \ln \mathcal{N}\left(y_n \mid \boldsymbol{\phi}(\boldsymbol{x}_n)\cdot\boldsymbol{\beta}, \sigma^2\right) && - \ln \mathcal{N}\left(\boldsymbol{\beta} \mid \mathbf{0}, \frac{1}{\lambda}\boldsymbol{I}\right) \\
= & \; const. - \ln \underbrace{p(\boldsymbol{y} \mid \boldsymbol{\beta}, \boldsymbol{\Phi}(\boldsymbol{X}), \sigma^2)}_{\text{Likelihood}} && - \ln \underbrace{p(\boldsymbol{\beta})}_{\text{prior}}
\end{aligned}
$$

- Regularized regression equivalent to MAP estimation
- Most alternative choices of regularizers and loss functions can be mapped to an equivalent probabilistic representation in a similar way.

Probabilistic interpretation

- So far: minimization of error functions.
- Back to probabilities?

$$
\begin{aligned}
E(\boldsymbol{\beta}) &= \underbrace{\frac{1}{2}\sum_{n=1}^{N}\left(y_n - \boldsymbol{\phi}(\boldsymbol{x}_n)\cdot\boldsymbol{\beta}\right)^2}_{\text{Squared error}} && + \underbrace{\frac{\lambda}{2}\boldsymbol{\beta}^\top\boldsymbol{\beta}}_{\text{Regularizer}} \\
&= const. - \sum_{n=1}^{N}\ln\mathcal{N}\left(y_n \mid \boldsymbol{\phi}(\boldsymbol{x}_n)\cdot\boldsymbol{\beta}, \sigma^2\right) && - \ln\mathcal{N}\left(\boldsymbol{\beta} \mid \mathbf{0}, \frac{1}{\lambda}\boldsymbol{I}\right) \\
&= const. - \ln\underbrace{p(\boldsymbol{y} \mid \boldsymbol{\beta}, \boldsymbol{\Phi}(\boldsymbol{X}), \sigma^2)}_{\text{Likelihood}} && - \ln\underbrace{p(\boldsymbol{\beta})}_{\text{prior}}
\end{aligned}
$$

- Regularized regression equivalent to MAP estimation
- Most alternative choices of regularizers and loss functions can be mapped to an equivalent probabilistic representation in a similar way.

Outline

Bayesian linear regression

- Likelihood as before

$$p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\beta}, \sigma^2) = \prod_{n=1}^{N} \mathcal{N}\left(y_n \mid \boldsymbol{\phi}(\boldsymbol{x}_n) \cdot \boldsymbol{\beta}, \sigma^2 \right)$$

- Define a conjugate prior over β

$$p(\boldsymbol{\beta}) = \mathcal{N}\left(\boldsymbol{\beta} \mid \boldsymbol{m}_0, \boldsymbol{S}_0 \right)$$

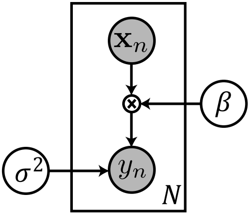
Bayesian linear regression

- Likelihood as before

$$p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\beta}, \sigma^2) = \prod_{n=1}^{N} \mathcal{N}\left(y_n \mid \boldsymbol{\phi}(\boldsymbol{x}_n) \cdot \boldsymbol{\beta}, \sigma^2 \right)$$

- Define a conjugate prior over $\boldsymbol{\beta}$

$$p(\boldsymbol{\beta}) = \mathcal{N}\left(\boldsymbol{\beta} \mid \boldsymbol{m}_0, \boldsymbol{S}_0 \right)$$



Bayesian linear regression

- Posterior probability of $\boldsymbol{\beta}$

$$
\begin{aligned}
p(\boldsymbol{\beta} \,|\, \boldsymbol{y}, \boldsymbol{X}, \sigma^2) &\propto \prod_{n=1}^{N} \mathcal{N}\left(y_n \mid \boldsymbol{\phi}(\boldsymbol{x}_n) \cdot \boldsymbol{\beta}, \sigma^2 \right) \cdot \mathcal{N}\left(\boldsymbol{\beta} \mid \boldsymbol{m}_0, \boldsymbol{S}_0 \right) \\
&= \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{\Phi}(\boldsymbol{X}) \cdot \boldsymbol{\beta}, \sigma^2 \boldsymbol{I} \right) \cdot \mathcal{N}\left(\boldsymbol{\beta} \mid \boldsymbol{m}_0, \boldsymbol{S}_0 \right) \\
&= \mathcal{N}\left(\boldsymbol{\beta} \mid \boldsymbol{\mu}_{\boldsymbol{\beta}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}} \right)
\end{aligned}
$$

- where

$$
\begin{aligned}
\boldsymbol{\mu}_{\boldsymbol{\beta}} &= \boldsymbol{\Sigma}_{\boldsymbol{\beta}} \left(\boldsymbol{S}_0^{-1} \boldsymbol{m}_0 + \frac{1}{\sigma^2} \boldsymbol{\Phi}(\boldsymbol{X})^\top \boldsymbol{y} \right) \\
\boldsymbol{\Sigma}_{\boldsymbol{\beta}} &= \left[\boldsymbol{S}_0^{-1} + \frac{1}{\sigma^2} \boldsymbol{\Phi}(\boldsymbol{X})^\top \boldsymbol{\Phi}(\boldsymbol{X}) \right]^{-1}
\end{aligned}
$$

Bayesian linear regression

Prior choice

- Choice of prior: regularized (ridge) regression

$$p(\boldsymbol{\beta}) = \mathcal{N}\big(\boldsymbol{\beta} \,|\, \boldsymbol{m}_0, \boldsymbol{S}_0\big).$$

- In this case

$$p(\boldsymbol{\beta} \,|\, \boldsymbol{y}, \boldsymbol{X}, \sigma^2) \propto \mathcal{N}\left(\boldsymbol{\beta} \mid \boldsymbol{m}_N, \boldsymbol{S}_N \right)$$

$$\boldsymbol{m}_N = \boldsymbol{S}_N \left(\boldsymbol{S}_0^{-1} \boldsymbol{m}_0 + \frac{1}{\sigma^2} \boldsymbol{\Phi}(\boldsymbol{X})^\top \boldsymbol{y} \right)$$

$$\boldsymbol{S}_N = \left[\boldsymbol{S}_0^{-1} + \frac{1}{\sigma^2} \boldsymbol{\Phi}(\boldsymbol{X})^\top \boldsymbol{\Phi}(\boldsymbol{X}) \right]^{-1}$$

- $\boldsymbol{m}_N$ is equal to the ridge regression (L_2) estimate for β **(Exercise: derive both and compare!)**
- Equivalent to maximum likelihood estimate for $\lambda \to 0$!

Bayesian linear regression

Prior choice

- Choice of prior: regularized (ridge) regression

$$p(\boldsymbol{\beta}) = \mathcal{N}\big(\boldsymbol{\beta} \,|\, \mathbf{0}, \frac{1}{\lambda}\boldsymbol{I}\big).$$

- In this case

$$\begin{aligned} p(\boldsymbol{\beta} \,|\, \boldsymbol{y}, \boldsymbol{X}, \sigma^2) &\propto \mathcal{N}\left(\, \boldsymbol{\beta} \mid \boldsymbol{m}_N, \boldsymbol{S}_N \,\right) \\ \boldsymbol{m}_N &= \boldsymbol{S}_N \left(\qquad\qquad \frac{1}{\sigma^2}\boldsymbol{\Phi}(\boldsymbol{X})^\top \boldsymbol{y} \right) \\ \boldsymbol{S}_N &= \left[\lambda \boldsymbol{I} + \frac{1}{\sigma^2}\boldsymbol{\Phi}(\boldsymbol{X})^\top \boldsymbol{\Phi}(\boldsymbol{X}) \right]^{-1} \end{aligned}$$

- $\boldsymbol{m}_N$ is equal to the ridge regression (L_2) estimate for β **(Exercise: derive both and compare!)**
- Equivalent to maximum likelihood estimate for $\lambda \to 0$!

Bayesian linear regression

Prior choice

- Choice of prior: regularized (ridge) regression

$$p(\boldsymbol{\beta}) = \mathcal{N}\big(\boldsymbol{\beta} \,|\, \mathbf{0}, \frac{1}{\lambda}\boldsymbol{I}\big).$$

- In this case

$$\begin{aligned} p(\boldsymbol{\beta} \,|\, \boldsymbol{y}, \boldsymbol{X}, \sigma^2) &\propto \mathcal{N}\,(\,\boldsymbol{\beta} \mid \boldsymbol{m}_N, \boldsymbol{S}_N\,) \\ \boldsymbol{m}_N &= \boldsymbol{S}_N \left(\qquad\qquad \frac{1}{\sigma^2}\,\boldsymbol{\Phi}(\boldsymbol{X})^\top \boldsymbol{y} \right) \\ \boldsymbol{S}_N &= \left[\lambda \boldsymbol{I} + \frac{1}{\sigma^2}\,\boldsymbol{\Phi}(\boldsymbol{X})^\top \boldsymbol{\Phi}(\boldsymbol{X}) \right]^{-1} \end{aligned}$$

- $\boldsymbol{m}_N$ is equal to the ridge regression (L_2) estimate for $\boldsymbol{\beta}$ **(Exercise: derive both and compare!)**
- Equivalent to maximum likelihood estimate for $\lambda \to 0$!

Bayesian linear regression

Prior choice

- Choice of prior: regularized (ridge) regression

$$p(\boldsymbol{\beta}) = \mathcal{N}\big(\boldsymbol{\beta} \,|\, \mathbf{0}, \frac{1}{\lambda}\boldsymbol{I}\big).$$

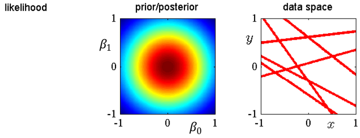
- In this case

$$\begin{aligned} p(\boldsymbol{\beta} \,|\, \boldsymbol{y}, \boldsymbol{X}, \sigma^2) &\propto \mathcal{N}\left(\, \boldsymbol{\beta} \mid \boldsymbol{m}_N, \boldsymbol{S}_N \,\right) \\ \boldsymbol{m}_N &= \boldsymbol{S}_N \left(\qquad\qquad \frac{1}{\sigma^2} \boldsymbol{\Phi}(\boldsymbol{X})^\top \boldsymbol{y} \right) \\ \boldsymbol{S}_N &= \left[\lambda \boldsymbol{I} + \frac{1}{\sigma^2} \boldsymbol{\Phi}(\boldsymbol{X})^\top \boldsymbol{\Phi}(\boldsymbol{X}) \right]^{-1} \end{aligned}$$

- $\boldsymbol{m}_N$ is equal to the ridge regression (L_2) estimate for $\boldsymbol{\beta}$ **(Exercise: derive both and compare!)**
- Equivalent to maximum likelihood estimate for $\lambda \to 0$!

Bayesian linear regression

Example: sequential Bayesian learning



- likelihood

$$\prod_{n=1}^{N} \mathcal{N}\left(y_n \mid \beta_0 + x_n \beta_1 \, , \, \sigma^2 \right)$$

- prior

$$\mathcal{N}\left(\boldsymbol{\beta} \mid \mathbf{0} \, , \, \frac{1}{\lambda} \boldsymbol{I} \right)$$

- This prior is conjugate, so we can do sequential learning
- 1 data point
- 2 data points
- 20 data points

Bayesian linear regression

Example: sequential Bayesian learning

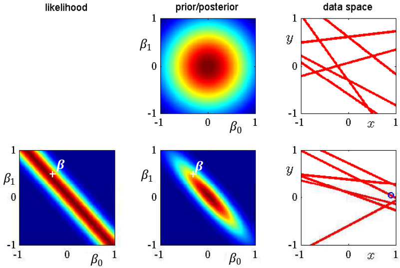
- likelihood

$$\prod_{n=1}^{N} \mathcal{N}\left(y_n \mid \beta_0 + x_n\beta_1 \, , \, \sigma^2\right)$$

- prior

$$\mathcal{N}\left(\boldsymbol{\beta} \mid \mathbf{0} \, , \, \frac{1}{\lambda}\boldsymbol{I}\right)$$

- This prior is conjugate, so we can do sequential learning
- 1 data point
- 2 data points
- 20 data points



Bayesian linear regression

Example: sequential Bayesian learning

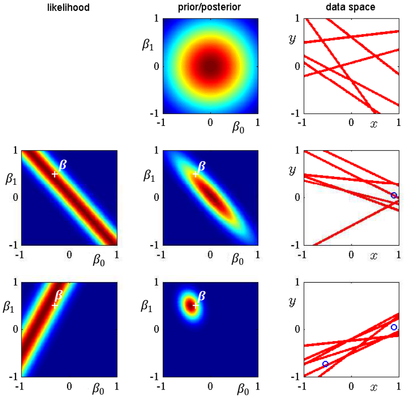
- likelihood

$$\prod_{n=1}^{N} \mathcal{N}\left(y_n \mid \beta_0 + x_n \beta_1 \,,\, \sigma^2\right)$$

- prior

$$\mathcal{N}\left(\boldsymbol{\beta} \mid \mathbf{0} \,,\, \frac{1}{\lambda}\boldsymbol{I}\right)$$

- This prior is conjugate, so we can do sequential learning
- 1 data point
- 2 data points
- 20 data points



Bayesian linear regression

Example: sequential Bayesian learning

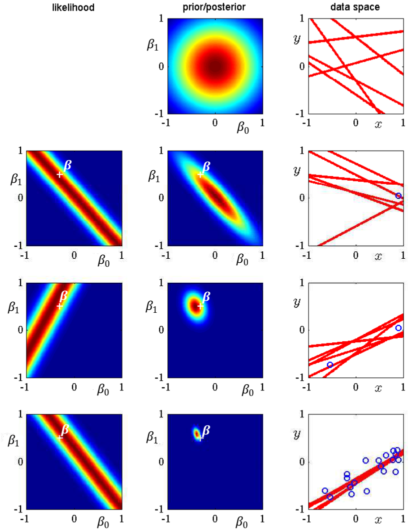
- likelihood

$$\prod_{n=1}^{N} \mathcal{N}\left(y_n \mid \beta_0 + x_n\beta_1 , \sigma^2\right)$$

- prior

$$\mathcal{N}\left(\boldsymbol{\beta} \mid \mathbf{0} , \frac{1}{\lambda}\boldsymbol{I}\right)$$

- This prior is conjugate, so we can do sequential learning
- 1 data point
- 2 data points
- 20 data points



Making predictions

- Prediction for fixed weight estimate $\hat{\boldsymbol{\beta}}$ at input $\boldsymbol{x}^\star$ trivial:

$$p(y^\star \mid \boldsymbol{x}^\star, \hat{\boldsymbol{\beta}}, \sigma^2) = \mathcal{N}\left(y^\star \mid \boldsymbol{\phi}(\boldsymbol{x}^\star)\hat{\boldsymbol{\beta}}, \sigma^2 \right)$$

- Integrate over β to take the posterior uncertainty into account

$$p(y^\star \mid \boldsymbol{x}^\star, \mathcal{D}) \propto \int_{\boldsymbol{\beta}} p(y^\star \mid \boldsymbol{x}^\star, \boldsymbol{\beta}, \sigma^2) p(\boldsymbol{\beta} \mid \boldsymbol{X}, \boldsymbol{y}, \sigma^2) \, \mathrm{d}\boldsymbol{\beta}$$

- Key:
  - prediction is again Gaussian
  - Predictive variance is increased due to the posterior uncertainty in β.

Making predictions

- Prediction for fixed weight estimate $\hat{\boldsymbol{\beta}}$ at input $\boldsymbol{x}^\star$ trivial:

$$p(y^\star \mid \boldsymbol{x}^\star, \hat{\boldsymbol{\beta}}, \sigma^2) = \mathcal{N}\left(y^\star \mid \boldsymbol{\phi}(\boldsymbol{x}^\star)\hat{\boldsymbol{\beta}}, \sigma^2\right)$$

- Integrate over $\boldsymbol{\beta}$ to take the posterior uncertainty into account

$$p(y^\star \mid \boldsymbol{x}^\star, \mathcal{D}) \propto \int_{\boldsymbol{\beta}} p(y^\star \mid \boldsymbol{x}^\star, \boldsymbol{\beta}, \sigma^2) p(\boldsymbol{\beta} \mid \boldsymbol{X}, \boldsymbol{y}, \sigma^2) \, \mathrm{d}\boldsymbol{\beta}$$

- Key:
  - prediction is again Gaussian
  - Predictive variance is increased due to the posterior uncertainty in β.

Making predictions

- Prediction for fixed weight estimate $\hat{\boldsymbol{\beta}}$ at input $\boldsymbol{x}^\star$ trivial:

$$p(y^\star \mid \boldsymbol{x}^\star, \hat{\boldsymbol{\beta}}, \sigma^2) = \mathcal{N}\left(y^\star \mid \boldsymbol{\phi}(\boldsymbol{x}^\star)\hat{\boldsymbol{\beta}}, \sigma^2\right)$$

- Integrate over $\boldsymbol{\beta}$ to take the posterior uncertainty into account

$$p(y^\star \mid \boldsymbol{x}^\star, \mathcal{D}) \propto \int_{\boldsymbol{\beta}} p(y^\star \mid \boldsymbol{x}^\star, \boldsymbol{\beta}, \sigma^2) p(\boldsymbol{\beta} \mid \boldsymbol{X}, \boldsymbol{y}, \sigma^2) \, \mathrm{d}\boldsymbol{\beta}$$

$$\propto \int_{\boldsymbol{\beta}} \mathcal{N}\left(y^\star \mid \boldsymbol{\phi}(\boldsymbol{x}^\star)\boldsymbol{\beta}, \sigma^2\right) \mathcal{N}\left(\boldsymbol{\beta} \mid \boldsymbol{m}_N, \boldsymbol{S}_N\right)$$

- Key:
  - prediction is again Gaussian
  - Predictive variance is increased due to the posterior uncertainty in $\boldsymbol{\beta}$.

Making predictions

- Prediction for fixed weight estimate $\hat{\boldsymbol{\beta}}$ at input $\boldsymbol{x}^\star$ trivial:

$$p(y^\star \mid \boldsymbol{x}^\star, \hat{\boldsymbol{\beta}}, \sigma^2) = \mathcal{N}\left(y^\star \mid \boldsymbol{\phi}(\boldsymbol{x}^\star)\hat{\boldsymbol{\beta}}, \sigma^2 \right)$$

- Integrate over $\boldsymbol{\beta}$ to take the posterior uncertainty into account

$$\begin{aligned} p(y^\star \mid \boldsymbol{x}^\star, \mathcal{D}) &\propto \int_{\boldsymbol{\beta}} p(y^\star \mid \boldsymbol{x}^\star, \boldsymbol{\beta}, \sigma^2) p(\boldsymbol{\beta} \mid \boldsymbol{X}, \boldsymbol{y}, \sigma^2)\, \mathrm{d}\boldsymbol{\beta} \\ &\propto \int_{\boldsymbol{\beta}} \mathcal{N}\left(y^\star \mid \boldsymbol{\phi}(\boldsymbol{x}^\star)\boldsymbol{\beta}, \sigma^2 \right) \mathcal{N}\left(\boldsymbol{\beta} \mid \boldsymbol{m}_N, \boldsymbol{S}_N \right) \\ &\propto \int_{\boldsymbol{\phi}(\boldsymbol{x}^\star)\boldsymbol{\beta}} \mathcal{N}\left(\boldsymbol{\phi}(\boldsymbol{x}^\star)\boldsymbol{\beta} \mid y^\star, \sigma^2 \right) \mathcal{N}\left(\boldsymbol{\phi}(\boldsymbol{x}^\star)\boldsymbol{\beta} \mid \boldsymbol{\phi}(\boldsymbol{x}^\star)\boldsymbol{m}_N, \boldsymbol{\phi}(\boldsymbol{x}^\star)\boldsymbol{S}_N\boldsymbol{\phi}(\boldsymbol{x}^\star)^\top \right) \end{aligned}$$

- Key:
  - prediction is again Gaussian
  - Predictive variance is increased due to the posterior uncertainty in $\boldsymbol{\beta}$.

Making predictions

- Prediction for fixed weight estimate $\hat{\boldsymbol{\beta}}$ at input $\boldsymbol{x}^\star$ trivial:

$$p(y^\star \,|\, \boldsymbol{x}^\star, \hat{\boldsymbol{\beta}}, \sigma^2) = \mathcal{N}\left(y^\star \mid \boldsymbol{\phi}(\boldsymbol{x}^\star)\hat{\boldsymbol{\beta}}, \sigma^2\right)$$

- Integrate over $\boldsymbol{\beta}$ to take the posterior uncertainty into account

$$\begin{aligned}
p(y^\star \,|\, \boldsymbol{x}^\star, \mathcal{D}) &\propto \int_{\boldsymbol{\beta}} p(y^\star \,|\, \boldsymbol{x}^\star, \boldsymbol{\beta}, \sigma^2) p(\boldsymbol{\beta} \,|\, \boldsymbol{X}, \boldsymbol{y}, \sigma^2)\, \mathrm{d}\boldsymbol{\beta} \\
&\propto \int_{\boldsymbol{\beta}} \mathcal{N}\left(y^\star \mid \boldsymbol{\phi}(\boldsymbol{x}^\star)\boldsymbol{\beta}, \sigma^2\right) \mathcal{N}\left(\boldsymbol{\beta} \mid \boldsymbol{m}_N, \boldsymbol{S}_N\right) \\
&\propto \int_{\boldsymbol{\phi}(\boldsymbol{x}^\star)\boldsymbol{\beta}} \mathcal{N}\left(\boldsymbol{\phi}(\boldsymbol{x}^\star)\boldsymbol{\beta} \mid y^\star, \sigma^2\right) \mathcal{N}\left(\boldsymbol{\phi}(\boldsymbol{x}^\star)\boldsymbol{\beta} \mid \boldsymbol{\phi}(\boldsymbol{x}^\star)\boldsymbol{m}_N, \boldsymbol{\phi}(\boldsymbol{x}^\star)\boldsymbol{S}_N\boldsymbol{\phi}(\boldsymbol{x}^\star)^\top\right) \\
&\propto \mathcal{N}\left(y^\star \mid \boldsymbol{\phi}(\boldsymbol{x}^\star) \cdot \boldsymbol{m}_N, \sigma^2 + \boldsymbol{\phi}(\boldsymbol{x}^\star)^\top \boldsymbol{S}_N \boldsymbol{\phi}(\boldsymbol{x}^\star)\right)
\end{aligned}$$

- Key:
  - prediction is again Gaussian
  - Predictive variance is increased due to the posterior uncertainty in β.

Making predictions

- Prediction for fixed weight estimate $\hat{\boldsymbol{\beta}}$ at input $\boldsymbol{x}^\star$ trivial:

$$p(y^\star \mid \boldsymbol{x}^\star, \hat{\boldsymbol{\beta}}, \sigma^2) = \mathcal{N}\left(y^\star \mid \boldsymbol{\phi}(\boldsymbol{x}^\star)\hat{\boldsymbol{\beta}}, \sigma^2\right)$$

- Integrate over $\boldsymbol{\beta}$ to take the posterior uncertainty into account

$$p(y^\star \mid \boldsymbol{x}^\star, \mathcal{D}) \propto \int_{\boldsymbol{\beta}} p(y^\star \mid \boldsymbol{x}^\star, \boldsymbol{\beta}, \sigma^2) p(\boldsymbol{\beta} \mid \boldsymbol{X}, \boldsymbol{y}, \sigma^2)\, \mathrm{d}\boldsymbol{\beta}$$

$$\propto \int_{\boldsymbol{\beta}} \mathcal{N}\left(y^\star \mid \boldsymbol{\phi}(\boldsymbol{x}^\star)\boldsymbol{\beta}, \sigma^2\right) \mathcal{N}\left(\boldsymbol{\beta} \mid \boldsymbol{m}_N, \boldsymbol{S}_N\right)$$

$$\propto \int_{\boldsymbol{\phi}(\boldsymbol{x}^\star)\boldsymbol{\beta}} \mathcal{N}\left(\boldsymbol{\phi}(\boldsymbol{x}^\star)\boldsymbol{\beta} \mid y^\star, \sigma^2\right) \mathcal{N}\left(\boldsymbol{\phi}(\boldsymbol{x}^\star)\boldsymbol{\beta} \mid \boldsymbol{\phi}(\boldsymbol{x}^\star)\boldsymbol{m}_N, \boldsymbol{\phi}(\boldsymbol{x}^\star)\boldsymbol{S}_N\boldsymbol{\phi}(\boldsymbol{x}^\star)^\top\right)$$

$$\propto \mathcal{N}\left(y^\star \mid \boldsymbol{\phi}(\boldsymbol{x}^\star) \cdot \boldsymbol{m}_N, \sigma^2 + \boldsymbol{\phi}(\boldsymbol{x}^\star)^\top \boldsymbol{S}_N \boldsymbol{\phi}(\boldsymbol{x}^\star)\right)$$

- Key:
  - prediction is again Gaussian
  - Predictive variance is increased due to the posterior uncertainty in $\boldsymbol{\beta}$.

Predictive distribution

Marginal variance for $x^\star$

$$\sigma^2 + \boldsymbol{\phi}(\boldsymbol{x}^\star)^\top \boldsymbol{S}_N \boldsymbol{\phi}(\boldsymbol{x}^\star)$$

Predictive covariance

$$\boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{S}_N \boldsymbol{\phi}(\boldsymbol{x}')$$

Experiment:

- 9 Gaussian basis functions

Empirical Observations:

- Variance approaches noise variance for large sample size
- Co-variance between close x values is high

Predictive distribution

Marginal variance for $x^\star$

$$\sigma^2 + \boldsymbol{\phi}(\boldsymbol{x}^\star)^\top \boldsymbol{S}_N \boldsymbol{\phi}(\boldsymbol{x}^\star)$$

Predictive covariance

$$\boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{S}_N \boldsymbol{\phi}(\boldsymbol{x}')$$

Experiment:

- 9 Gaussian basis functions

Empirical Observations:

- Variance approaches noise variance for large sample size
- Co-variance between close x values is high

Predictive distribution

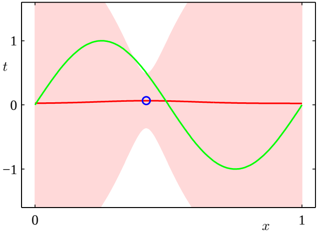
Marginal variance for $x^\star$

$$\sigma^2 + \boldsymbol{\phi}(\boldsymbol{x}^\star)^\top \boldsymbol{S}_N \boldsymbol{\phi}(\boldsymbol{x}^\star)$$

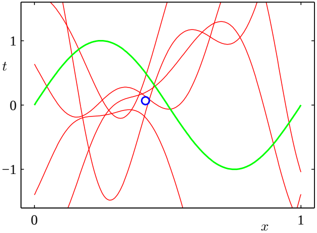
Predictive covariance

$$\boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{S}_N \boldsymbol{\phi}(\boldsymbol{x}')$$

Predictive distribution



Visualize by sampling from the posterior of $\boldsymbol{\beta}$



Empirical Observations:

Experiment:

- 9 Gaussian basis functions

Predictive distribution

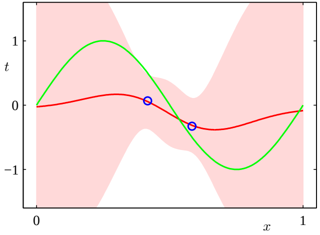
Marginal variance for $x^\star$

$$\sigma^2 + \boldsymbol{\phi}(\boldsymbol{x}^\star)^\top \boldsymbol{S}_N \boldsymbol{\phi}(\boldsymbol{x}^\star)$$

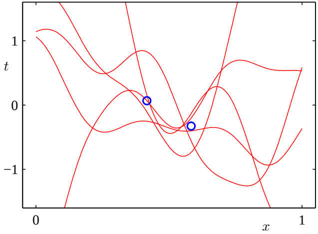
Predictive covariance

$$\boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{S}_N \boldsymbol{\phi}(\boldsymbol{x}')$$

Predictive distribution



Visualize by sampling from the posterior of β



Empirical Observations:

Experiment:

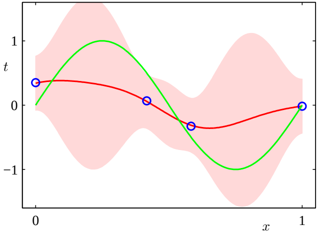
- 9 Gaussian basis functions

Predictive distribution

Marginal variance for $x^\star$

$$\sigma^2 + \boldsymbol{\phi}(\boldsymbol{x}^\star)^\top \boldsymbol{S}_N \boldsymbol{\phi}(\boldsymbol{x}^\star)$$

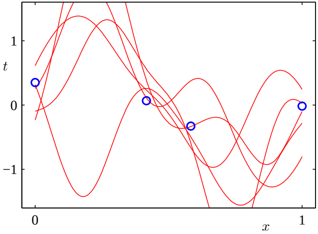
Predictive distribution



Predictive covariance

$$\boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{S}_N \boldsymbol{\phi}(\boldsymbol{x}')$$

Visualize by sampling from the posterior of β



Empirical Observations:

- **Variance** approaches noise variance for large sample size
- Co-variance between close x values is high

Experiment:

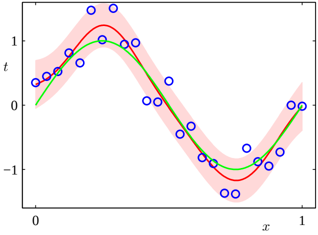
- 9 Gaussian basis functions

Predictive distribution

Marginal variance for $x^\star$

$$\sigma^2 + \boldsymbol{\phi}(\boldsymbol{x}^\star)^\top \boldsymbol{S}_N \boldsymbol{\phi}(\boldsymbol{x}^\star)$$

Predictive distribution



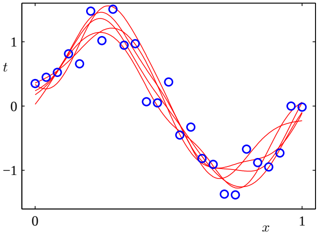
Experiment:

- 9 Gaussian basis functions

Predictive covariance

$$\boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{S}_N \boldsymbol{\phi}(\boldsymbol{x}')$$

Visualize by sampling from the posterior of $\boldsymbol{\beta}$



Empirical Observations:

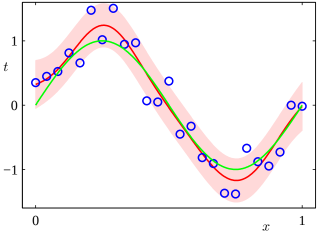
- **Variance** approaches noise variance for large sample size
- **Co-variance** between close x values is high

Predictive distribution

Marginal variance for $x^\star$

$$\sigma^2 + \boldsymbol{\phi}(\boldsymbol{x}^\star)^\top \boldsymbol{S}_N \boldsymbol{\phi}(\boldsymbol{x}^\star)$$

Predictive distribution



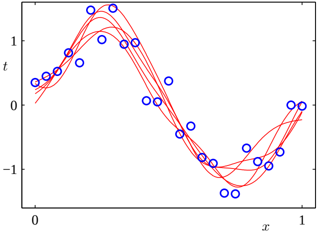
Experiment:

- 9 Gaussian basis functions

Predictive covariance

$$\boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{S}_N \boldsymbol{\phi}(\boldsymbol{x}')$$

Visualize by sampling from the posterior of $\boldsymbol{\beta}$



Empirical Observations:

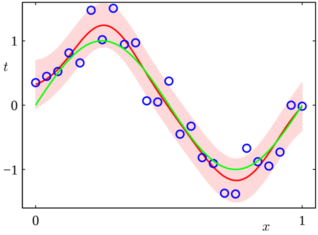
- **Variance** approaches noise variance for large sample size
- **Co-variance** between close x values is high

Predictive distribution

Marginal variance for $x^\star$

$$\sigma^2 + \boldsymbol{\phi}(\boldsymbol{x}^\star)^\top \boldsymbol{S}_N \boldsymbol{\phi}(\boldsymbol{x}^\star)$$

Predictive distribution



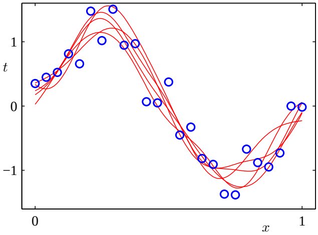
Experiment:

- 9 Gaussian basis functions

Predictive covariance

$$\boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{S}_N \boldsymbol{\phi}(\boldsymbol{x}')$$

Visualize by sampling from the posterior of $\boldsymbol{\beta}$



Empirical Observations:

- **Variance** approaches noise variance for large sample size
- **Co-variance** between close $\boldsymbol{x}$ values is high

Outline

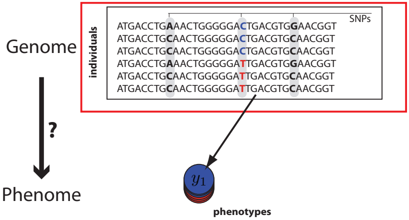
Model comparison

Motivation

- What degree of polynomials describes the data best?
- Is the linear model at all appropriate?
- Association testing.

Model comparison
Motivation

- What degree of polynomials describes the data best?
- Is the linear model at all appropriate?
- Association testing.



Bayesian model comparison

- How do we choose among alternative models?
- Assume we want to choose among models $\mathcal{H}_0, \ldots, \mathcal{H}_M$ for a dataset $\mathcal{D}$.
- Posterior probability for a particular model i

$$p(\mathcal{H}_i \mid \mathcal{D}) \propto \underbrace{p(\mathcal{D} \mid \mathcal{H}_i)}_{\text{Evidence}} \underbrace{p(\mathcal{H}_i)}_{\text{Prior}}$$

Bayesian model comparison

- How do we choose among alternative models?
- Assume we want to choose among models $\mathcal{H}_0, \ldots, \mathcal{H}_M$ for a dataset $\mathcal{D}$.
- Posterior probability for a particular model i

$$p(\mathcal{H}_i \mid \mathcal{D}) \propto \underbrace{p(\mathcal{D} \mid \mathcal{H}_i)}_{\text{Evidence}} \underbrace{p(\mathcal{H}_i)}_{\text{Prior}}$$

Bayesian model comparison

How to calculate the evidence

- The evidence is not the model likelihood!

$$p(\mathcal{D} \,|\, \mathcal{H}_i) = \int_{\boldsymbol{\Theta}} p(\mathcal{D} \,|\, \boldsymbol{\Theta}) p(\boldsymbol{\Theta}) \,\mathrm{d}\boldsymbol{\Theta} \text{ for model parameters } \boldsymbol{\Theta}.$$

- Remember:

$$p(\boldsymbol{\Theta} \,|\, \mathcal{H}_i, \mathcal{D}) = \frac{p(\mathcal{D} \,|\, \mathcal{H}_i, \boldsymbol{\Theta}) p(\boldsymbol{\Theta})}{p(\mathcal{D} \,|\, \mathcal{H}_i)}$$

Bayesian model comparison

How to calculate the evidence

- The evidence is not the model likelihood!

$$p(\mathcal{D} \,|\, \mathcal{H}_i) = \int_{\boldsymbol{\Theta}} p(\mathcal{D} \,|\, \boldsymbol{\Theta}) p(\boldsymbol{\Theta}) \, \mathrm{d}\boldsymbol{\Theta} \quad \text{for model parameters } \boldsymbol{\Theta}.$$

- Remember:

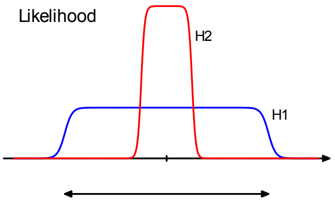
$$p(\boldsymbol{\Theta} \,|\, \mathcal{H}_i, \mathcal{D}) = \frac{p(\mathcal{D} \,|\, \mathcal{H}_i, \boldsymbol{\Theta}) p(\boldsymbol{\Theta})}{p(\mathcal{D} \,|\, \mathcal{H}_i)}$$

$$\text{posterior} = \frac{\text{likelihood} \cdot \text{prior}}{\text{Evidence}}$$

Bayesian model comparison

Bayesian Occam's razor

- The evidence integral penalizes overly complex models.
- A model with few parameters and lower maximum likelihood ($\mathcal{H}_1$) may win over a model with a peaked likelihood that requires many more parameters ($\mathcal{H}_2$).
- When averaging the likelihood over all possible parameters, more complex models have low fit for most of the setting, resulting in a lower evidence
- Complex models have low average over many possible data sets
- Simple models have large evidence on a small range of data sets, extremely low evidence otherwise

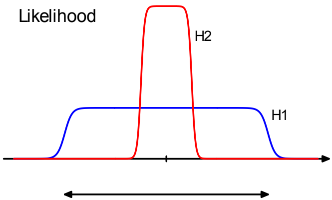


(C.M. Bishop, Pattern Recognition and Machine Learning)

Bayesian model comparison

Bayesian Occam's razor

- The evidence integral penalizes overly complex models.
- A model with few parameters and lower maximum likelihood ($\mathcal{H}_1$) may win over a model with a peaked likelihood that requires many more parameters ($\mathcal{H}_2$).
- When averaging the likelihood over all possible parameters, more complex models have low fit for most of the setting, resulting in a lower evidence
- Complex models have low average over many possible data sets
- Simple models have large evidence on a small range of data sets, extremely low evidence otherwise

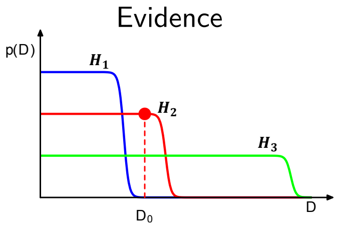


(C.M. Bishop, Pattern Recognition and Machine Learning)

Bayesian model comparison

Bayesian Occam's razor

- The evidence integral penalizes overly complex models.
- A model with few parameters and lower maximum likelihood ($\mathcal{H}_1$) may win over a model with a peaked likelihood that requires many more parameters ($\mathcal{H}_2$).
- When averaging the likelihood over all possible parameters, more complex models have low fit for most of the setting, resulting in a lower evidence
- Complex models have low average over many possible data sets
- Simple models have large evidence on a small range of data sets, extremely low evidence otherwise

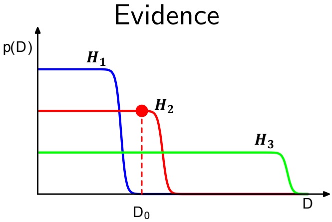


(C.M. Bishop, Pattern Recognition and Machine Learning)

Bayesian model comparison

Bayesian Occam's razor

- The evidence integral penalizes overly complex models.
- A model with few parameters and lower maximum likelihood ($\mathcal{H}_1$) may win over a model with a peaked likelihood that requires many more parameters ($\mathcal{H}_2$).
- When averaging the likelihood over all possible parameters, more complex models have low fit for most of the setting, resulting in a lower evidence
- Complex models have low average over many possible data sets
- Simple models have large evidence on a small range of data sets, extremely low evidence otherwise

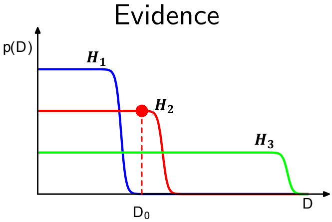


(C.M. Bishop, Pattern Recognition and Machine Learning)

Bayesian model comparison

Bayesian Occam's razor

- The evidence integral penalizes overly complex models.
- A model with few parameters and lower maximum likelihood ($\mathcal{H}_1$) may win over a model with a peaked likelihood that requires many more parameters ($\mathcal{H}_2$).
- When averaging the likelihood over all possible parameters, more complex models have low fit for most of the setting, resulting in a lower evidence
- Complex models have low average over many possible data sets
- Simple models have large evidence on a small range of data sets, extremely low evidence otherwise



(C.M. Bishop, Pattern Recognition and Machine Learning)

Application to GWAS

Relevance of a single SNP

- Consider an association study.
  - $\mathcal{H}_0$: no association

$$p(\boldsymbol{y} \mid \mathcal{H}_0, \boldsymbol{X}, \boldsymbol{\Theta}_0) = \mathcal{N}\left(\boldsymbol{y} \mid \mathbf{0}, \sigma^2 \boldsymbol{I}\right)$$

$$p(\mathcal{D} \mid \mathcal{H}_0) = \int_{\sigma^2} \mathcal{N}\left(\boldsymbol{y} \mid \mathbf{0}, \sigma^2 \boldsymbol{I}\right) p(\sigma^2)$$

  - $\mathcal{H}_1$: linear association

$$p(\boldsymbol{y} \mid \mathcal{H}_1, \boldsymbol{x}_i, \boldsymbol{\Theta}_1) = \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{x}_i \cdot \beta, \sigma^2 \boldsymbol{I}\right)$$

$$p(\mathcal{D} \mid \mathcal{H}_1) = \int_{\sigma^2, \beta} \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{x}_i \cdot \beta, \sigma^2 \boldsymbol{I}\right) p(\sigma^2) p(\beta)$$

- Depending on the choice of priors, $p(\sigma^2)$ and $p(\beta)$, the required integrals are often tractable in closed form. (Conjugate priors!)

Application to GWAS

Relevance of a single SNP

- Consider an association study.
  - $\mathcal{H}_0$: no association

$$p(\boldsymbol{y}\,|\,\mathcal{H}_0, \boldsymbol{X}, \boldsymbol{\Theta}_0) = \mathcal{N}\left(\, \boldsymbol{y} \mid \boldsymbol{0}, \sigma^2 \boldsymbol{I} \,\right)$$
$$p(\mathcal{D}\,|\,\mathcal{H}_0) = \int_{\sigma^2} \mathcal{N}\left(\, \boldsymbol{y} \mid \boldsymbol{0}, \sigma^2 \boldsymbol{I} \,\right) p(\sigma^2)$$

  - $\mathcal{H}_1$: linear association

$$p(\boldsymbol{y}\,|\,\mathcal{H}_1, \boldsymbol{x}_i, \boldsymbol{\Theta}_1) = \mathcal{N}\left(\, \boldsymbol{y} \mid \boldsymbol{x}_i \cdot \beta, \sigma^2 \boldsymbol{I} \,\right)$$
$$p(\mathcal{D}\,|\,\mathcal{H}_1) = \int_{\sigma^2, \boldsymbol{\beta}} \mathcal{N}\left(\, \boldsymbol{y} \mid \boldsymbol{x}_i \cdot \beta, \sigma^2 \boldsymbol{I} \,\right) p(\sigma^2) p(\beta)$$

- Depending on the choice of priors, $p(\sigma^2)$ and $p(\beta)$, the required integrals are often tractable in closed form. (Conjugate priors!)

Application to GWAS

Relevance of a single SNP

- Consider an association study.
  - $\mathcal{H}_0$: no association

$$p(\boldsymbol{y} \,|\, \mathcal{H}_0, \boldsymbol{X}, \boldsymbol{\Theta}_0) = \mathcal{N}\left(\, \boldsymbol{y} \mid \mathbf{0}, \sigma^2 \boldsymbol{I} \,\right)$$

$$p(\mathcal{D} \,|\, \mathcal{H}_0) = \int_{\sigma^2} \mathcal{N}\left(\, \boldsymbol{y} \mid \mathbf{0}, \sigma^2 \boldsymbol{I} \,\right) p(\sigma^2)$$

  - $\mathcal{H}_1$: linear association

$$p(\boldsymbol{y} \,|\, \mathcal{H}_1, \boldsymbol{x}_i, \boldsymbol{\Theta}_1) = \mathcal{N}\left(\, \boldsymbol{y} \mid \boldsymbol{x}_i \cdot \beta, \sigma^2 \boldsymbol{I} \,\right)$$

$$p(\mathcal{D} \,|\, \mathcal{H}_1) = \int_{\sigma^2, \boldsymbol{\beta}} \mathcal{N}\left(\, \boldsymbol{y} \mid \boldsymbol{x}_i \cdot \beta, \sigma^2 \boldsymbol{I} \,\right) p(\sigma^2) p(\beta)$$

- Depending on the choice of priors, $p(\sigma^2)$ and $p(\beta)$, the required integrals are often tractable in closed form. (Conjugate priors!)

Application to GWAS

Scoring models

- Similar to likelihood ratios, the ratio of the evidences, the Bayes factor can be used to score alternative models:

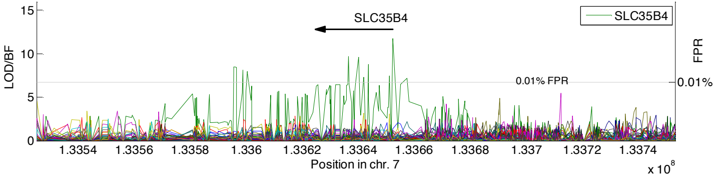
$$BF = \ln \frac{p(\mathcal{D} \,|\, \mathcal{H}_1)}{p(\mathcal{D} \,|\, \mathcal{H}_0)}.$$

Application to GWAS

Scoring models

- Similar to likelihood ratios, the ratio of the evidences, the Bayes factor can be used to score alternative models:

$$BF = \ln \frac{p(\mathcal{D} \,|\, \mathcal{H}_1)}{p(\mathcal{D} \,|\, \mathcal{H}_0)}.$$



Application to GWAS

Posterior probability of an association

- Bayes factors are useful, however we would like a probabilistic answer how certain an association really is.
- Posterior probability of $\mathcal{H}_1$

$$p(\mathcal{H}_1 \,|\, \mathcal{D}) = \frac{p(\mathcal{D} \,|\, \mathcal{H}_1) p(\mathcal{H}_1)}{p(\mathcal{D})}$$
$$= \frac{p(\mathcal{D} \,|\, \mathcal{H}_1) p(\mathcal{H}_1)}{p(\mathcal{D} \,|\, \mathcal{H}_1) p(\mathcal{H}_1) + p(\mathcal{D} \,|\, \mathcal{H}_0) p(\mathcal{H}_0)}$$

- $p(\mathcal{H}_1 \,|\, \mathcal{D}) + p(\mathcal{H}_0 \,|\, \mathcal{D}) = 1$, prior probability of observing a real association.

Application to GWAS

Posterior probability of an association

- Bayes factors are useful, however we would like a probabilistic answer how certain an association really is.
- Posterior probability of $\mathcal{H}_1$

$$p(\mathcal{H}_1 \,|\, \mathcal{D}) = \frac{p(\mathcal{D} \,|\, \mathcal{H}_1)p(\mathcal{H}_1)}{p(\mathcal{D})}$$
$$= \frac{p(\mathcal{D} \,|\, \mathcal{H}_1)p(\mathcal{H}_1)}{p(\mathcal{D} \,|\, \mathcal{H}_1)p(\mathcal{H}_1) + p(\mathcal{D} \,|\, \mathcal{H}_0)p(\mathcal{H}_0)}$$

- $p(\mathcal{H}_1 \,|\, \mathcal{D}) + p(\mathcal{H}_0 \,|\, \mathcal{D}) = 1$, prior probability of observing a real association.

Application to GWAS

Posterior probability of an association

- Bayes factors are useful, however we would like a probabilistic answer how certain an association really is.
- Posterior probability of $\mathcal{H}_1$

$$p(\mathcal{H}_1 \,|\, \mathcal{D}) = \frac{p(\mathcal{D} \,|\, \mathcal{H}_1) p(\mathcal{H}_1)}{p(\mathcal{D})}$$
$$= \frac{p(\mathcal{D} \,|\, \mathcal{H}_1) p(\mathcal{H}_1)}{p(\mathcal{D} \,|\, \mathcal{H}_1) p(\mathcal{H}_1) + p(\mathcal{D} \,|\, \mathcal{H}_0) p(\mathcal{H}_0)}$$

- $p(\mathcal{H}_1 \,|\, \mathcal{D}) + p(\mathcal{H}_0 \,|\, \mathcal{D}) = 1$, prior probability of observing a real association.

Bayes factor versus likelihood ratio

Bayes factor

- Models of different complexity can be objectively compared.
- Statistical significance as posterior probability of a model.
- Typically hard to compute.

Likelihood ratio

- Likelihood ratio scales with the number of parameters.
- Likelihood ratios have known null distribution, yielding p-values.
- Often easy to compute.

Bayes factor versus likelihood ratio

Bayes factor

- Models of different complexity can be objectively compared.
- Statistical significance as posterior probability of a model.
- Typically hard to compute.

Likelihood ratio

- Likelihood ratio scales with the number of parameters.
- Likelihood ratios have known null distribution, yielding p-values.
- Often easy to compute.

Marginal likelihood of variance component models

- Consider a linear model, accounting for a set of measured SNPs $\boldsymbol{X}$
$$p(\boldsymbol{y}\,|\,\boldsymbol{X},\boldsymbol{\beta},\sigma^2) = \mathcal{N}\left(\boldsymbol{y} \mid \sum_{s=1}^{S} \boldsymbol{x}_s \beta_s, \sigma^2 \boldsymbol{I}\right)$$
- Choose identical Gaussian prior for all weights
$$p(\boldsymbol{\beta}) = \prod_{s=1}^{S} \mathcal{N}\left(\beta_s \mid 0, \sigma_{\mathrm{g}}^2\right)$$
- Marginal likelihood
$$p(\boldsymbol{y}\,|\,\boldsymbol{X},) = \int_{\boldsymbol{\beta}} \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{X}\boldsymbol{\beta}, \sigma^2\boldsymbol{I}\right) \mathcal{N}\left(\boldsymbol{\beta} \mid \boldsymbol{0}, \sigma_{\mathrm{g}}^2 \boldsymbol{I}\right)$$
- Number of hyperparameters independent of number of SNPs

Marginal likelihood of variance component models

- Consider a linear model, accounting for a set of measured SNPs $\boldsymbol{X}$
$$p(\boldsymbol{y} \,|\, \boldsymbol{X}, \boldsymbol{\beta}, \sigma^2) = \mathcal{N}\left(\boldsymbol{y} \mid \sum_{s=1}^{S} \boldsymbol{x}_s \beta_s, \sigma^2 \boldsymbol{I}\right)$$
- Choose identical Gaussian prior for all weights
$$p(\boldsymbol{\beta}) = \prod_{s=1}^{S} \mathcal{N}\left(\beta_s \mid 0, \sigma_{\mathrm{g}}^2\right)$$
- Marginal likelihood
$$p(\boldsymbol{y} \,|\, \boldsymbol{X},) = \int_{\boldsymbol{\beta}} \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{X}\boldsymbol{\beta}, \sigma^2 \boldsymbol{I}\right) \mathcal{N}\left(\boldsymbol{\beta} \mid \boldsymbol{0}, \sigma_{\mathrm{g}}^2 \boldsymbol{I}\right)$$
- Number of hyperparameters independent of number of SNPs

Marginal likelihood of variance component models

- Consider a linear model, accounting for a set of measured SNPs $\boldsymbol{X}$
$$p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\beta}, \sigma^2) = \mathcal{N}\left(\boldsymbol{y} \mid \sum_{s=1}^{S} \boldsymbol{x}_s \beta_s, \sigma^2 \boldsymbol{I} \right)$$
- Choose identical Gaussian prior for all weights
$$p(\boldsymbol{\beta}) = \prod_{s=1}^{S} \mathcal{N}\left(\beta_s \mid 0, \sigma_{\mathrm{g}}^2 \right)$$
- Marginal likelihood
$$p(\boldsymbol{y} \mid \boldsymbol{X}, \sigma^2, \sigma_{\mathrm{g}}^2) = \int_{\boldsymbol{\beta}} \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{X}\boldsymbol{\beta}, \sigma^2 \boldsymbol{I} \right) \mathcal{N}\left(\boldsymbol{\beta} \mid \boldsymbol{0}, \sigma_{\mathrm{g}}^2 \boldsymbol{I} \right)$$
$$= \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{0}, \sigma_{\mathrm{g}}^2 \boldsymbol{X}\boldsymbol{X}^\top + \sigma^2 \boldsymbol{I} \right)$$
- Number of hyperparameters independent of number of SNPs

Marginal likelihood of variance component models

- Consider a linear model, accounting for a set of measured SNPs $\boldsymbol{X}$

$$p(\boldsymbol{y} \,|\, \boldsymbol{X}, \boldsymbol{\beta}, \sigma^2) = \mathcal{N}\left(\boldsymbol{y} \mid \sum_{s=1}^{S} \boldsymbol{x}_s \beta_s, \sigma^2 \boldsymbol{I} \right)$$

- Choose identical Gaussian prior for all weights

$$p(\boldsymbol{\beta}) = \prod_{s=1}^{S} \mathcal{N}\left(\beta_s \mid 0, \sigma_{\mathrm{g}}^2 \right)$$

- Marginal likelihood

$$\begin{aligned} p(\boldsymbol{y} \,|\, \boldsymbol{X}, \sigma^2, \sigma_{\mathrm{g}}^2) &= \int_{\boldsymbol{\beta}} \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{X}\boldsymbol{\beta}, \sigma^2 \boldsymbol{I} \right) \mathcal{N}\left(\boldsymbol{\beta} \mid \boldsymbol{0}, \sigma_{\mathrm{g}}^2 \boldsymbol{I} \right) \\ &= \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{0}, \sigma_{\mathrm{g}}^2 \boldsymbol{X}\boldsymbol{X}^{\top} + \sigma^2 \boldsymbol{I} \right) \end{aligned}$$

- Number of hyperparameters independent of number of SNPs

Marginal likelihood of variance component models

Basis functions

- The analogous derivation can be repeated for a feature mapping ϕ
$$p(\boldsymbol{y} \,|\, \boldsymbol{X}, \boldsymbol{\beta}, \sigma^2) = \mathcal{N}\left(\boldsymbol{y} \mid \sum_{s=1}^{S} \phi(\boldsymbol{x}_s)\beta_s, \sigma^2 \boldsymbol{I} \right) =$$
$$\mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{\Phi}(\boldsymbol{X})\boldsymbol{\beta}, \sigma^2 \boldsymbol{I} \right)$$
- Marginal likelihood
$$p(\boldsymbol{y} \,|\, \boldsymbol{X}, \sigma^2, \sigma_{\mathrm{g}}^2) = \int_{\boldsymbol{\beta}} \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{\Phi}(\boldsymbol{X})\boldsymbol{\beta}, \sigma^2 \boldsymbol{I} \right) \mathcal{N}\left(\boldsymbol{\beta} \mid \boldsymbol{0}, \sigma_{\mathrm{g}}^2 \boldsymbol{I} \right)$$
$$= \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{0}, \sigma_{\mathrm{g}}^2 \underbrace{\boldsymbol{\Phi}(\boldsymbol{X})\boldsymbol{\Phi}(\boldsymbol{X})^{\top}}_{\boldsymbol{K}} + \sigma^2 \boldsymbol{I} \right)$$
- $\boldsymbol{K}$: (N x N) kernel or covariance induced by feature mapping ϕ.

Marginal likelihood of variance component models

Basis functions

- The analogous derivation can be repeated for a feature mapping ϕ
$p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\beta}, \sigma^2) = \mathcal{N}\left(\boldsymbol{y} \mid \sum_{s=1}^{S} \phi(\boldsymbol{x}_s)\beta_s, \sigma^2 \boldsymbol{I}\right) =$
$\mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{\Phi}(\boldsymbol{X})\boldsymbol{\beta}, \sigma^2 \boldsymbol{I}\right)$
- Marginal likelihood

$$p(\boldsymbol{y} \mid \boldsymbol{X}, \sigma^2, \sigma_{\mathrm{g}}^2) = \int_{\boldsymbol{\beta}} \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{\Phi}(\boldsymbol{X})\boldsymbol{\beta}, \sigma^2 \boldsymbol{I}\right) \mathcal{N}\left(\boldsymbol{\beta} \mid \boldsymbol{0}, \sigma_{\mathrm{g}}^2 \boldsymbol{I}\right)$$

$$= \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{0}, \sigma_{\mathrm{g}}^2 \underbrace{\boldsymbol{\Phi}(\boldsymbol{X})\boldsymbol{\Phi}(\boldsymbol{X})^\top}_{\boldsymbol{K}} + \sigma^2 \boldsymbol{I}\right)$$

- $\boldsymbol{K}$: (N x N) kernel or covariance induced by feature mapping ϕ.

Marginal likelihood of variance component models

Basis functions

- The analogous derivation can be repeated for a feature mapping ϕ
$p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\beta}, \sigma^2) = \mathcal{N}\left(\boldsymbol{y} \mid \sum_{s=1}^{S} \phi(\boldsymbol{x}_s)\beta_s, \sigma^2 \boldsymbol{I} \right) =$
$\mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{\Phi}(\boldsymbol{X})\boldsymbol{\beta}, \sigma^2 \boldsymbol{I} \right)$
- Marginal likelihood

$$p(\boldsymbol{y} \mid \boldsymbol{X}, \sigma^2, \sigma_{\mathrm{g}}^2) = \int_{\boldsymbol{\beta}} \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{\Phi}(\boldsymbol{X})\boldsymbol{\beta}, \sigma^2 \boldsymbol{I} \right) \mathcal{N}\left(\boldsymbol{\beta} \mid \boldsymbol{0}, \sigma_{\mathrm{g}}^2 \boldsymbol{I} \right)$$

$$= \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{0}, \sigma_{\mathrm{g}}^2 \underbrace{\boldsymbol{\Phi}(\boldsymbol{X})\boldsymbol{\Phi}(\boldsymbol{X})^{\top}}_{\boldsymbol{K}} + \sigma^2 \boldsymbol{I} \right)$$

- $\boldsymbol{K}$: (N x N) kernel or covariance induced by feature mapping ϕ.

The missing heritability paradox

- Complex traits are regulated by a large number of small effects
  - Human height: the best single SNP explains little variance.
  - But: height of the parents are highly predictive for the height of the child!

Marginal likelihood of variance component models

Application to GWAS

Linear additive models for complex traits

- Multiple linear regression model over causal SNPs

$$p(\boldsymbol{y} \,|\, \boldsymbol{X}, \boldsymbol{\beta}, \sigma^2) = \mathcal{N}\big(\boldsymbol{y} \,|\, \sum_{s \in \text{causal}} \boldsymbol{x}_s \beta_s \,,\, \sigma^2 \boldsymbol{I}\big)$$

- Which SNPs are causal ?
  Approximation: consider all S available common SNPs [Yang et al. 2011]

$$p(\boldsymbol{y} \,|\, \boldsymbol{X}, \boldsymbol{\beta}, \sigma^2) = \mathcal{N}\big(\boldsymbol{y} \,|\, \sum_{s=1}^{S} \boldsymbol{x}_s \beta_s \,,\, \sigma^2 \boldsymbol{I}\big)$$

- Causal SNPs either in the model or "tagged" by linkage disequilibrium to nearby common SNPs
- Uncertainty over causal SNPs: Prior on all SNP effects $p(\beta_s) = \mathcal{N}\left(\beta_s \mid 0, \sigma_g^2/S\right)$
- Marginalize out weights

$$p(\boldsymbol{y} \,|\, \boldsymbol{X}, \sigma_g^2, \sigma^2) = \mathcal{N}\big(\boldsymbol{y} \,|\, \boldsymbol{0} \,,\, \sigma_g^2 \sum_{s=1}^{S} \frac{1}{S} \boldsymbol{x}_s \boldsymbol{x}_s^\top + \sigma^2 \boldsymbol{I}\big)$$

- Perform maximum marginal likelihood estimation on σ_g^2 and σ^2.

Marginal likelihood of variance component models

Application to GWAS

Linear additive models for complex traits

- Multiple linear regression model over causal SNPs

$$p(\boldsymbol{y}\,|\,\boldsymbol{X},\boldsymbol{\beta},\sigma^2) = \mathcal{N}\Big(\boldsymbol{y}\,|\,\sum_{s\in\text{causal}} \boldsymbol{x}_s\beta_s\;,\;\sigma^2\boldsymbol{I}\Big)$$

- Which SNPs are causal ?
  Approximation: consider all S available common SNPs [Yang et al. 2011]

$$p(\boldsymbol{y}\,|\,\boldsymbol{X},\boldsymbol{\beta},\sigma^2) = \mathcal{N}\Big(\boldsymbol{y}\,|\,\sum_{s=1}^{S} \boldsymbol{x}_s\beta_s\;,\;\sigma^2\boldsymbol{I}\Big)$$

- Causal SNPs either in the model or "tagged" by linkage disequilibrium to nearby common SNPs
- Uncertainty over causal SNPs: Prior on all SNP effects $p(\beta_s) = \mathcal{N}\left(\beta_s \mid 0, \sigma_g^2/S\right)$
- Marginalize out weights

$$p(\boldsymbol{y}\,|\,\boldsymbol{X},\sigma_g^2,\sigma^2) = \mathcal{N}\Big(\boldsymbol{y}\,|\,\boldsymbol{0}\;,\;\sigma_g^2\sum_{s=1}^{S}\frac{1}{S}\boldsymbol{x}_s\boldsymbol{x}_s^\top + \sigma^2\boldsymbol{I}\Big)$$

- Perform maximum marginal likelihood estimation on σ_g^2 and σ^2.

Marginal likelihood of variance component models

Application to GWAS

Linear additive models for complex traits

- Multiple linear regression model over causal SNPs

$$p(\boldsymbol{y} \,|\, \boldsymbol{X}, \boldsymbol{\beta}, \sigma^2) = \mathcal{N}\big(\boldsymbol{y} \,|\, \sum_{s \in \text{causal}} \boldsymbol{x}_s \beta_s \,,\, \sigma^2 \boldsymbol{I}\big)$$

- Which SNPs are causal ?
  Approximation: consider all S available common SNPs [Yang et al. 2011]

$$p(\boldsymbol{y} \,|\, \boldsymbol{X}, \boldsymbol{\beta}, \sigma^2) = \mathcal{N}\big(\boldsymbol{y} \,|\, \sum_{s=1}^{S} \boldsymbol{x}_s \beta_s \,,\, \sigma^2 \boldsymbol{I}\big)$$

- Causal SNPs either in the model or "tagged" by linkage disequilibrium to nearby common SNPs
- Uncertainty over causal SNPs: Prior on all SNP effects $p(\beta_s) = \mathcal{N}\left(\beta_s \mid 0, \sigma_g^2 / S\right)$
- Marginalize out weights

$$p(\boldsymbol{y} \,|\, \boldsymbol{X}, \sigma_g^2, \sigma^2) = \mathcal{N}\big(\boldsymbol{y} \,|\, \boldsymbol{0} \,,\, \sigma_g^2 \sum_{s=1}^{S} \frac{1}{S} \boldsymbol{x}_s \boldsymbol{x}_s^\top + \sigma^2 \boldsymbol{I}\big)$$

- Perform maximum marginal likelihood estimation on σ_g^2 and σ^2.

Marginal likelihood of variance component models

Application to GWAS

Linear additive models for complex traits

- Multiple linear regression model over causal SNPs

$$p(\boldsymbol{y} \,|\, \boldsymbol{X}, \boldsymbol{\beta}, \sigma^2) = \mathcal{N}\big(\boldsymbol{y} \,|\, \sum_{s \in \text{causal}} \boldsymbol{x}_s \beta_s \,,\, \sigma^2 \boldsymbol{I}\big)$$

- Which SNPs are causal ?
  Approximation: consider all S available common SNPs [Yang et al. 2011]

$$p(\boldsymbol{y} \,|\, \boldsymbol{X}, \boldsymbol{\beta}, \sigma^2) = \mathcal{N}\big(\boldsymbol{y} \,|\, \sum_{s=1}^{S} \boldsymbol{x}_s \beta_s \,,\, \sigma^2 \boldsymbol{I}\big)$$

- Causal SNPs either in the model or "tagged" by linkage disequilibrium to nearby common SNPs
- Uncertainty over causal SNPs: Prior on all SNP effects $p(\beta_s) = \mathcal{N}\left(\beta_s \mid 0, \sigma_{\mathrm{g}}^2 / S\right)$
- Marginalize out weights

$$p(\boldsymbol{y} \,|\, \boldsymbol{X}, \sigma_{\mathrm{g}}^2, \sigma^2) = \mathcal{N}\big(\boldsymbol{y} \,|\, \boldsymbol{0} \,,\, \sigma_{\mathrm{g}}^2 \sum_{s=1}^{S} \frac{1}{S} \boldsymbol{x}_s \boldsymbol{x}_s^\top + \sigma^2 \boldsymbol{I}\big)$$

- Perform maximum marginal likelihood estimation on σ_{g}^2 and σ^2.

Marginal likelihood of variance component models

Application to GWAS

Linear additive models for complex traits

- Multiple linear regression model over causal SNPs

$$p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\beta}, \sigma^2) = \mathcal{N}\big(\boldsymbol{y} \mid \sum_{s \in \text{causal}} \boldsymbol{x}_s \beta_s \,,\, \sigma^2 \boldsymbol{I}\big)$$

- Which SNPs are causal ?
  Approximation: consider all S available common SNPs [Yang et al. 2011]

$$p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\beta}, \sigma^2) = \mathcal{N}\big(\boldsymbol{y} \mid \sum_{s=1}^{S} \boldsymbol{x}_s \beta_s \,,\, \sigma^2 \boldsymbol{I}\big)$$

- Causal SNPs either in the model or "tagged" by linkage disequilibrium to nearby common SNPs
- Uncertainty over causal SNPs: Prior on all SNP effects $p(\beta_s) = \mathcal{N}\left(\beta_s \mid 0, \sigma_{\mathrm{g}}^2 / S\right)$
- Marginalize out weights

$$p(\boldsymbol{y} \mid \boldsymbol{X}, \sigma_{\mathrm{g}}^2, \sigma^2) = \mathcal{N}\big(\boldsymbol{y} \mid \boldsymbol{0} \,,\, \sigma_{\mathrm{g}}^2 \sum_{s=1}^{S} \frac{1}{S} \boldsymbol{x}_s \boldsymbol{x}_s^\top + \sigma^2 \boldsymbol{I}\big)$$

- Perform maximum marginal likelihood estimation on σ_{g}^2 and σ^2.

Marginal likelihood of variance component models

Application to GWAS

- Approximate variance model

$$p(\boldsymbol{y} \mid \boldsymbol{X}, \sigma_{\mathrm{g}}^2, \sigma^2) = \mathcal{N}\big(\boldsymbol{y} \mid \boldsymbol{0}, \sigma_{\mathrm{g}}^2 \frac{1}{S} \boldsymbol{X}\boldsymbol{X}^\top + \sigma^2 \boldsymbol{I}\big)$$

- Genetic variance σ_{g}^2 across chromosomes
- (Narrow-sense) heritability

$$h^2 = \frac{\sigma_{\mathrm{g}}^2}{\sigma_{\mathrm{g}}^2 + \sigma^2}$$

- Narrow-sense refers to linear additive part of the heritability

Marginal likelihood of variance component models

Application to GWAS

- Approximate variance model

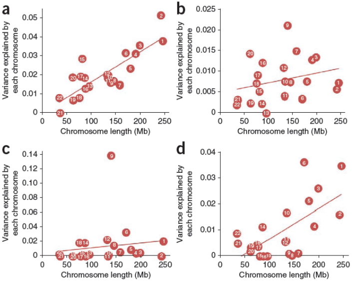
$$p(\boldsymbol{y} \mid \boldsymbol{X}, \sigma_{\mathrm{g}}^2, \sigma^2) = \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{0}, \sigma_{\mathrm{g}}^2 \frac{1}{S} \boldsymbol{X} \boldsymbol{X}^\top + \sigma^2 \boldsymbol{I}\right)$$

- Genetic variance σ_{g}^2 across chromosomes

- (Narrow-sense) heritability

$$h^2 = \frac{\sigma_{\mathrm{g}}^2}{\sigma_{\mathrm{g}}^2 + \sigma^2}$$

- Narrow-sense refers to linear additive part of the heritability



[Yang et al. 2011]

Marginal likelihood of variance component models

Application to GWAS

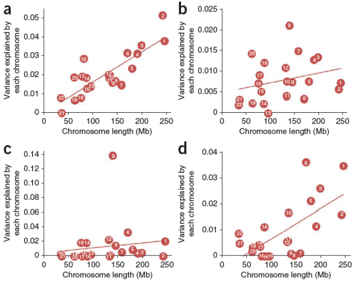
- Approximate variance model

$$p(\boldsymbol{y} \,|\, \boldsymbol{X}, \sigma_{\mathrm{g}}^2, \sigma^2) = \mathcal{N}\left(\boldsymbol{y} \,|\, \boldsymbol{0}, \sigma_{\mathrm{g}}^2 \frac{1}{S} \boldsymbol{X}\boldsymbol{X}^\top + \sigma^2 \boldsymbol{I}\right)$$

- Genetic variance σ_{g}^2 across chromosomes
- (Narrow-sense) heritability

$$h^2 = \frac{\sigma_{\mathrm{g}}^2}{\sigma_{\mathrm{g}}^2 + \sigma^2}$$

- Narrow-sense refers to linear additive part of the heritability



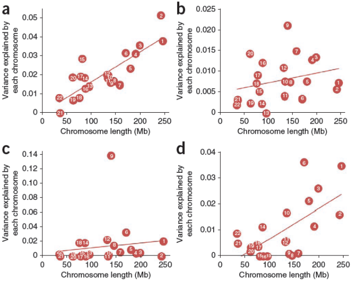
[Yang et al. 2011]

Marginal likelihood of variance component models

Application to GWAS

- Approximate variance model

$$p(\boldsymbol{y} \,|\, \boldsymbol{X}, \sigma_{\mathrm{g}}^2, \sigma^2) = \mathcal{N}\left(\boldsymbol{y} \,|\, \boldsymbol{0}, \sigma_{\mathrm{g}}^2 \frac{1}{S} \boldsymbol{X}\boldsymbol{X}^\top + \sigma^2 \boldsymbol{I}\right)$$



- Genetic variance σ_{g}^2 across chromosomes
- (Narrow-sense) heritability

$$h^2 = \frac{\sigma_{\mathrm{g}}^2}{\sigma_{\mathrm{g}}^2 + \sigma^2}$$

- Narrow-sense refers to linear additive part of the heritability

[Yang et al. 2011]

Outline

Summary

- Linear models for curve fitting and multiple linear regression.
- Maximum likelihood and least squares regression are identical.
- Construction of features using a mapping ϕ.
- Regularized least squares and other models that correspond to different choices of loss functions.
- Bayesian linear regression.
- Model comparison and Occam's razor.
- Variance component models in GWAS.

Outlook

- Estimation technique for σ_{g}^2 and σ^2.
- Use marginal linear model for confounder correction in GWAS testing of single SNPs
  - Linear mixed models for GWAS testing
- Use marginal linear model for testing for significant associations of sets of variants.
  - Idea : Test for $\mathcal{H}_0 : \sigma_{\mathrm{g}}^2 = 0$ vs. $\mathcal{H}_1 : \sigma_{\mathrm{g}}^2 > 0$
  - Random effects testing

Tasks

- Derive ridge regularized $\boldsymbol{\beta}_{MAP}$ in linear regression
- Derive posterior distribution (mean and covariance) of $\boldsymbol{\beta}$ in a linear regression under a Normal prior
- Compare them!
- Derive marginal likelihood for linear regression under a Normal prior on $\boldsymbol{\beta}$
  - hint: The following expression is a Gaussian convolution:

$$\int \mathcal{N}\left(\boldsymbol{a} \mid \boldsymbol{b}, \boldsymbol{\Sigma_a}\right) \cdot \mathcal{N}\left(\boldsymbol{b} \mid \boldsymbol{\mu_b}, \boldsymbol{\Sigma_b}\right) \mathrm{d}\boldsymbol{b}$$

$$= \int \mathcal{N}\left(\boldsymbol{a} - \boldsymbol{b} \mid \boldsymbol{0}, \boldsymbol{\Sigma_a}\right) \cdot \mathcal{N}\left(\boldsymbol{b} \mid \boldsymbol{\mu_b}, \boldsymbol{\Sigma_b}\right) \mathrm{d}\boldsymbol{b}$$