Linear models

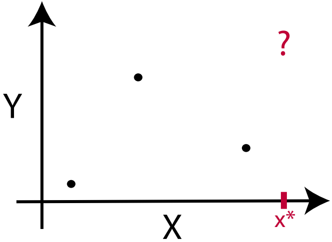
Oliver Stegle and Karsten Borgwardt

Machine Learning and
Computational Biology Research Group,
Max Planck Institute for Biological Cybernetics and
Max Planck Institute for Developmental Biology, Tübingen

Curve fitting

Tasks we are interested in:

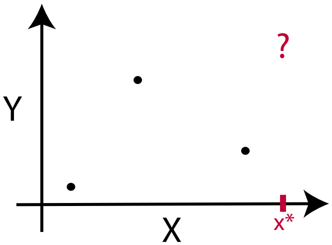
- Making predictions
- Comparison of alternative models



Curve fitting

Tasks we are interested in:

- Making predictions
- Comparison of alternative models



Further reading, useful material

- Christopher M. Bishop: Pattern Recognition and Machine learning.
  - Good background, covers most of the course material and much more!
  - This lecture is largely inspired by chapter 3 of the book.

Outline

Outline

Regression
Noise model and likelihood

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

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

- Equivalent likelihood formulation:

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

Regression
Choosing a regressor

- Choose f to be linear:

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

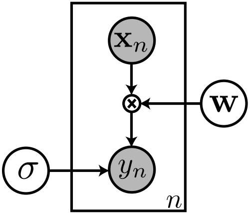
- Consider bias free case, $c = 0$, otherwise inlcude an additional column of ones in each $\mathbf{x}_n$.

Regression
Choosing a regressor

- Choose f to be linear:

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

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



Equivalent graphical model

Linear Regression

Maximum likelihood

- Taking the logarithm, we obtain

$$\ln p(\mathbf{y} \,|\, \mathbf{w}, \mathbf{X}, \sigma^2) = \sum_{n=1}^{N} \ln \mathcal{N}\left(y_n \,\middle|\, \mathbf{w}^{\mathrm{T}} \mathbf{x}_n, \sigma^2\right)$$

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

- 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

$$\ln p(\mathbf{y} \mid \mathbf{w}, \mathbf{X}, \sigma^2) = \sum_{n=1}^{N} \ln \mathcal{N}\left(y_n \mid \mathbf{w}^{\mathrm{T}} \mathbf{x}_n, \sigma^2\right)$$

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

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

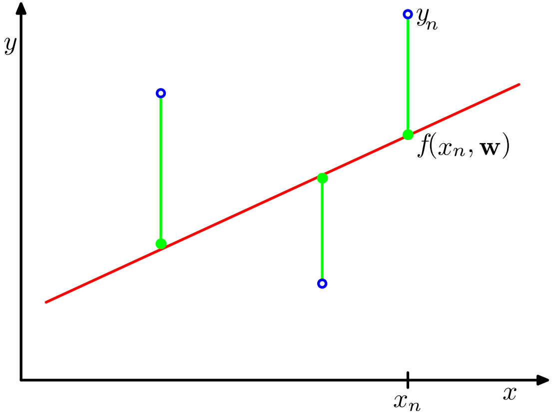
Linear Regression
Maximum likelihood

- Taking the logarithm, we obtain

$$\ln p(\mathbf{y} \mid \mathbf{w}, \mathbf{X}, \sigma^2) = \sum_{n=1}^{N} \ln \mathcal{N}\left(y_n \,\middle|\, \mathbf{w}^{\mathrm{T}} \mathbf{x}_n, \sigma^2\right)$$
$$= -\frac{N}{2} \ln 2\pi\sigma^2 - \frac{1}{2\sigma^2} \underbrace{\sum_{n=1}^{N} (y_n - \mathbf{w}^{\mathrm{T}} \cdot \mathbf{x}_n)^2}_{\text{Sum of squares}}$$

- 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)

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

Linear Regression and Least Squares

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

$$\frac{d}{\mathrm{d}w_i} \ln p(\mathbf{y} \mid \mathbf{w}, \sigma^2) = \frac{d}{\mathrm{d}w_i} \left[-\frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \mathbf{w} \cdot \mathbf{x}_n)^2 \right]$$

$$= \frac{1}{\sigma^2} \sum_{n=1}^{N} (y_n - \mathbf{w} \cdot \mathbf{x}_n) x_i$$

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

$$\nabla_{\mathbf{w}} \ln p(\mathbf{y} \mid \mathbf{w}, \sigma^2) = \frac{1}{\sigma^2} \sum_{n=1}^{N} (y_n - \mathbf{w} \cdot \mathbf{x}_n) \mathbf{x}_n^{\mathrm{T}} = 0$$

$$\Longrightarrow \mathbf{w}_{\mathsf{ML}} = \underbrace{(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}}_{\text{Pseudo inverse}} \mathbf{y}$$

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

Polynomial Curve Fitting

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

$$\begin{aligned} f(x, \mathbf{w}) &= w_0 + w_1 x + w_2 x^2 + \cdots + w_K x^K \\ &= \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}), \end{aligned}$$

  where we defined $\boldsymbol{\phi}(\mathbf{x}) = (1, x, x^2, \ldots, x^K)$.
- Similarly, ϕ can be any feature mapping.
- Possible to show: the feature map ϕ can be expressed in terms of kernels (kernel trick).

Polynomial Curve Fitting

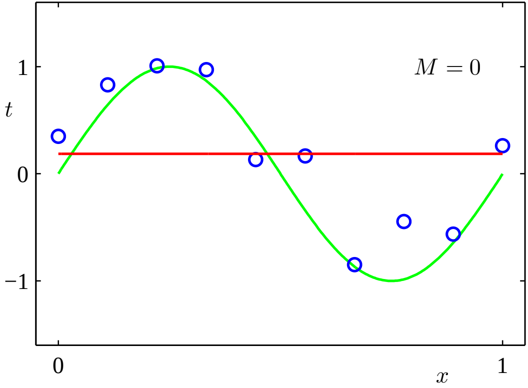
- Use the polynomials up to degree K to construct new features from x

$$
\begin{aligned}
f(x, \mathbf{w}) &= w_0 + w_1 x + w_2 x^2 + \cdots + w_K x^K \\
&= \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}),
\end{aligned}
$$

  where we defined $\boldsymbol{\phi}(\mathbf{x}) = (1, x, x^2, \ldots, x^K)$.
- Similarly, ϕ can be any feature mapping.
- Possible to show: the feature map ϕ can be expressed in terms of kernels (kernel trick).

Polynomial Curve Fitting
Overfitting

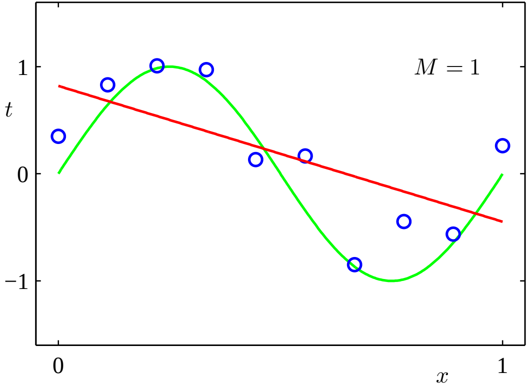
- The degree of the polynomial is crucial to avoid under- and overfitting.



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

Polynomial Curve Fitting
Overfitting

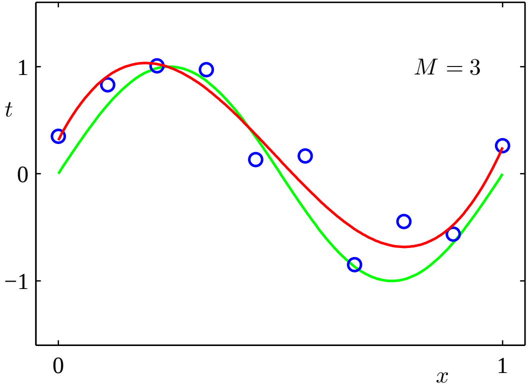
- The degree of the polynomial is crucial to avoid under- and overfitting.



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

Polynomial Curve Fitting
Overfitting

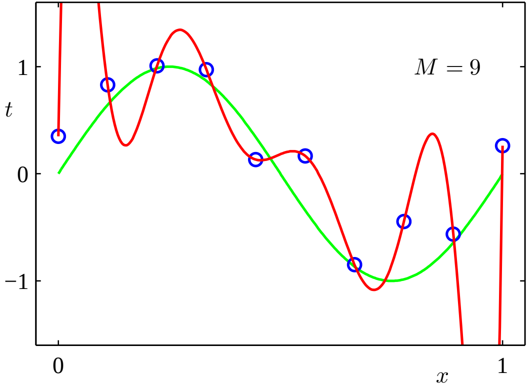
- The degree of the polynomial is crucial to avoid under- and overfitting.



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

Polynomial Curve Fitting
Overfitting

- The degree of the polynomial is crucial to avoid under- and overfitting.



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

Regularized Least Squares

- Solutions to avoid overfitting:
  - Intelligently choose K
  - Regularize the regression weights $\mathbf{w}$
- Construct a smoothed error function

$$E(\mathbf{w}) = \underbrace{\frac{1}{2}\sum_{n=1}^{N}\left(y_n - \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n)\right)^2}_{\text{Squared error}} + \underbrace{\frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}}_{\text{Regularizer}}$$

Regularized Least Squares

- Solutions to avoid overfitting:
  - Intelligently choose K
  - Regularize the regression weights $\mathbf{w}$
- Construct a smoothed error function

$$E(\mathbf{w}) = \underbrace{\frac{1}{2}\sum_{n=1}^{N}\left(y_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\right)^2}_{\text{Squared error}} + \underbrace{\frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}}_{\text{Regularizer}}$$

Regularized Least Squares

More general regularizers

- A more general regularization approach:

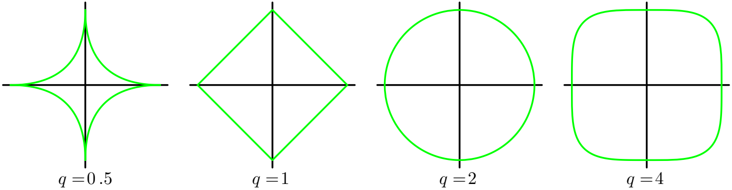
$$E(\mathbf{w}) = \underbrace{\frac{1}{2}\sum_{n=1}^{N}\left(y_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\right)^2}_{\text{Squared error}} + \underbrace{\frac{\lambda}{2}\sum_{d=1}^{D}|w_d|^q}_{\text{Regularizer}}$$

Regularized Least Squares

More general regularizers

- A more general regularization approach:

$$E(\mathbf{w}) = \underbrace{\frac{1}{2}\sum_{n=1}^{N}\left(y_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\right)^2}_{\text{Squared error}} + \underbrace{\frac{\lambda}{2}\sum_{d=1}^{D}|w_d|^q}_{\text{Regularizer}}$$



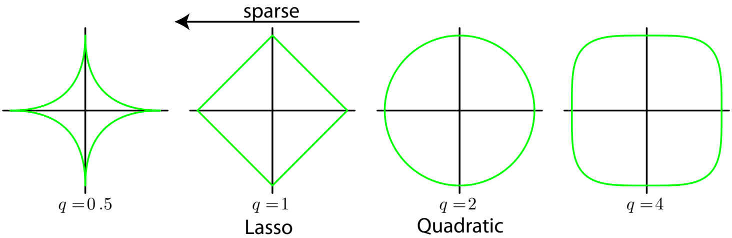
(C.M. Bishop, Pattern Recognition and Machine Learning)

Regularized Least Squares

More general regularizers

- A more general regularization approach:

$$E(\mathbf{w}) = \underbrace{\frac{1}{2}\sum_{n=1}^{N}\left(y_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\right)^2}_{\text{Squared error}} + \underbrace{\frac{\lambda}{2}\sum_{d=1}^{D}|w_d|^q}_{\text{Regularizer}}$$



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

Loss functions and other methods

- Even more general: vary the loss function

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

- 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 $E(\mathbf{w})$, yielding a point estimate for $\mathbf{w}$.

Loss functions and other methods

- Even more general: vary the loss function

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

- 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 $E(\mathbf{w})$, yielding a point estimate for $\mathbf{w}$.

Loss functions and other methods

- Even more general: vary the loss function

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

- 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 $E(\mathbf{w})$, yielding a point estimate for $\mathbf{w}$.

Regularized Least Squares
Probabilistic equivalent

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

$$E(\mathbf{w}) = \underbrace{\frac{1}{2}\sum_{n=1}^{N}\left(y_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\right)^2}_{\text{Squared error}} + \underbrace{\frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}}_{\text{Regularizer}}$$

- Similarly: most other choices of regularizers and loss functions can be mapped to an equivalent probabilistic representation.

Regularized Least Squares
Probabilistic equivalent

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

$$E(\mathbf{w}) = \underbrace{\frac{1}{2}\sum_{n=1}^{N}\left(y_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\right)^2}_{\text{Squared error}} \qquad + \underbrace{\frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}}_{\text{Regularizer}}$$

$$= -\ln p(\mathbf{y} \mid \mathbf{w}, \boldsymbol{\Phi}(\mathbf{X}), \sigma^2) \qquad - \ln p(\mathbf{w})$$

- Similarly: most other choices of regularizers and loss functions can be mapped to an equivalent probabilistic representation.

Regularized Least Squares
Probabilistic equivalent

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

$$
\begin{aligned}
E(\mathbf{w}) &= \underbrace{\frac{1}{2}\sum_{n=1}^{N}\left(y_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\right)^2}_{\text{Squared error}} && + \underbrace{\frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}}_{\text{Regularizer}} \\
&= -\ln p(\mathbf{y} \,|\, \mathbf{w}, \boldsymbol{\Phi}(\mathbf{X}), \sigma^2) && - \ln p(\mathbf{w}) \\
&= -\sum_{n=1}^{N} \ln \mathcal{N}\left(y_n \,\middle|\, \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n), \sigma^2\right) && - \ln \mathcal{N}\left(\mathbf{w} \,\middle|\, \mathbf{0}, \frac{1}{\lambda}\mathbf{I}\right)
\end{aligned}
$$

- Similarly: most other choices of regularizers and loss functions can be mapped to an equivalent probabilistic representation.

Regularized Least Squares
Probabilistic equivalent

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

$$
\begin{aligned}
E(\mathbf{w}) &= \underbrace{\frac{1}{2}\sum_{n=1}^{N}\left(y_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\right)^2}_{\text{Squared error}} && + \underbrace{\frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}}_{\text{Regularizer}} \\
&= -\ln p(\mathbf{y} \mid \mathbf{w}, \boldsymbol{\Phi}(\mathbf{X}), \sigma^2) && -\ln p(\mathbf{w}) \\
&= -\sum_{n=1}^{N}\ln \mathcal{N}\left(y_n \mid \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n), \sigma^2\right) && -\ln \mathcal{N}\left(\mathbf{w} \,\middle|\, \mathbf{0}, \frac{1}{\lambda}\mathbf{I}\right)
\end{aligned}
$$

- Similarly: most other choices of regularizers and loss functions can be mapped to an equivalent probabilistic representation.

Outline

Bayesian linear regression

- Likelihood as before

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

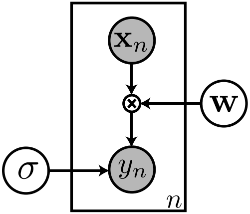
- Define a conjugate prior over $\mathbf{w}$

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

Bayesian linear regression

- Likelihood as before

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



- Define a conjugate prior over $\mathbf{w}$

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

Bayesian linear regression

- Posterior probability of $\mathbf{w}$

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

- where

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

Bayesian linear regression

Prior choice

- A common choice is a prior that corresponds to regularized regression

$$p(\mathbf{w}) = \mathcal{N}\left(\mathbf{w} \,\middle|\, \mathbf{0}, \frac{1}{\lambda}\mathbf{I}\right).$$

- In this case

$$\boldsymbol{\mu}_{\mathbf{w}} = \boldsymbol{\Sigma}_{\mathbf{w}}\left(\mathbf{S}_0^{-1}\mathbf{m}_0 + \frac{1}{\sigma^2}\boldsymbol{\Phi}(\mathbf{X})^{\mathrm{T}}\mathbf{y}\right)$$

$$\boldsymbol{\Sigma}_{\mathbf{w}} = \left[\mathbf{S}_0^{-1} + \frac{1}{\sigma^2}\boldsymbol{\Phi}(\mathbf{X})^{\mathrm{T}}\boldsymbol{\Phi}(\mathbf{X})\right]^{-1}$$

Bayesian linear regression

Prior choice

- A common choice is a prior that corresponds to regularized regression

$$p(\mathbf{w}) = \mathcal{N}\left(\mathbf{w} \,\middle|\, \mathbf{0}, \frac{1}{\lambda}\mathbf{I}\right).$$

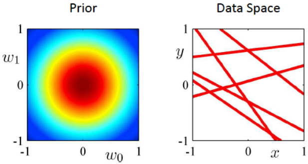
- In this case

$$\boldsymbol{\mu}_{\mathbf{w}} = \boldsymbol{\Sigma}_{\mathbf{w}}\left(\qquad\qquad \frac{1}{\sigma^2}\boldsymbol{\Phi}(\mathbf{X})^{\mathrm{T}}\mathbf{y}\right)$$

$$\boldsymbol{\Sigma}_{\mathbf{w}} = \left[\lambda\mathbf{I} + \frac{1}{\sigma^2}\boldsymbol{\Phi}(\mathbf{X})^{\mathrm{T}}\boldsymbol{\Phi}(\mathbf{X})\right]^{-1}$$

Bayesian linear regression
Example

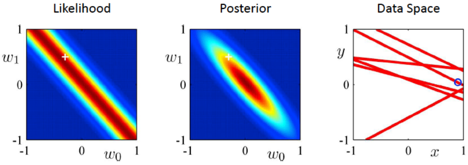
0 Data points



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

Bayesian linear regression
Example

1 Data point

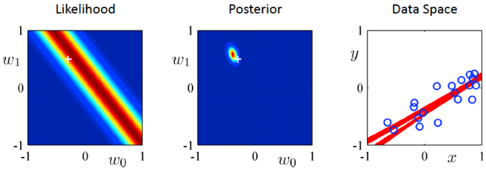


(C.M. Bishop, Pattern Recognition and Machine Learning)

Bayesian linear regression

Example

20 Data points



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

Making predictions

- Prediction for fixed weight $\hat{\mathbf{w}}$ at input $\mathbf{x}^\star$ trivial:

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

- Integrate over $\mathbf{w}$ to take the posterior uncertainty into account

$$\begin{aligned} p(y^\star \mid \mathbf{x}^\star, \mathcal{D}) &= \int_{\mathbf{w}} p(y^\star \mid \mathbf{x}^\star, \mathbf{w}, \sigma^2) p(\mathbf{w} \mid \mathbf{X}, \mathbf{y}, \sigma^2) \\ &= \int_{\mathbf{w}} \mathcal{N}\left(y^\star \,\middle|\, \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}^\star), \sigma^2\right) \mathcal{N}\left(\mathbf{w} \mid \boldsymbol{\mu}_{\mathbf{w}}, \boldsymbol{\Sigma}_{\mathbf{w}}\right) \\ &= \mathcal{N}\left(y^\star \,\middle|\, \boldsymbol{\mu}_{\mathbf{w}}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}^\star), \sigma^2 + \boldsymbol{\phi}(\mathbf{x}^\star)^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathbf{w}}\boldsymbol{\phi}(\mathbf{x}^\star)\right) \end{aligned}$$

- Key:
  - prediction is again Gaussian
  - Predictive variance is increase due to the posterior uncertainty in $\mathbf{w}$.

Making predictions

- Prediction for fixed weight $\hat{\mathbf{w}}$ at input $\mathbf{x}^\star$ trivial:

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

- Integrate over $\mathbf{w}$ to take the posterior uncertainty into account

$$\begin{aligned} p(y^\star \mid \mathbf{x}^\star, \mathcal{D}) &= \int_{\mathbf{w}} p(y^\star \mid \mathbf{x}^\star, \mathbf{w}, \sigma^2) p(\mathbf{w} \mid \mathbf{X}, \mathbf{y}, \sigma^2) \\ &= \int_{\mathbf{w}} \mathcal{N}\left(y^\star \mid \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}^\star), \sigma^2\right) \mathcal{N}\left(\mathbf{w} \mid \boldsymbol{\mu}_{\mathbf{w}}, \boldsymbol{\Sigma}_{\mathbf{w}}\right) \\ &= \mathcal{N}\left(y^\star \mid \boldsymbol{\mu}_{\mathbf{w}}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}^\star), \sigma^2 + \boldsymbol{\phi}(\mathbf{x}^\star)^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathbf{w}}\boldsymbol{\phi}(\mathbf{x}^\star)\right) \end{aligned}$$

- Key:
  - prediction is again Gaussian
  - Predictive variance is increase due to the posterior uncertainty in $\mathbf{w}$.

Making predictions

- Prediction for fixed weight $\hat{\mathbf{w}}$ at input $\mathbf{x}^\star$ trivial:

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

- Integrate over $\mathbf{w}$ to take the posterior uncertainty into account

$$\begin{aligned} p(y^\star \mid \mathbf{x}^\star, \mathcal{D}) &= \int_{\mathbf{w}} p(y^\star \mid \mathbf{x}^\star, \mathbf{w}, \sigma^2) p(\mathbf{w} \mid \mathbf{X}, \mathbf{y}, \sigma^2) \\ &= \int_{\mathbf{w}} \mathcal{N}\left(y^\star \mid \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}^\star), \sigma^2\right) \mathcal{N}\left(\mathbf{w} \mid \boldsymbol{\mu}_{\mathbf{w}}, \boldsymbol{\Sigma}_{\mathbf{w}}\right) \\ &= \mathcal{N}\left(y^\star \mid \boldsymbol{\mu}_{\mathbf{w}}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}^\star), \sigma^2 + \boldsymbol{\phi}(\mathbf{x}^\star)^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathbf{w}}\boldsymbol{\phi}(\mathbf{x}^\star)\right) \end{aligned}$$

- Key:
  - prediction is again Gaussian
  - Predictive variance is increase due to the posterior uncertainty in $\mathbf{w}$.

Outline

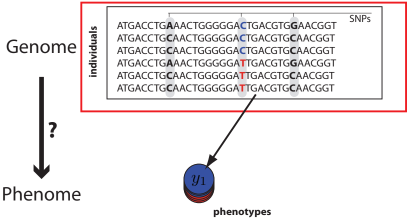
Model comparison

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

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

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!

$$p(\mathcal{D} \mid \mathcal{H}_i) = \int_{\boldsymbol{\theta}} p(\mathcal{D} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}) \text{ for model parameters } \boldsymbol{\theta}.$$

- Remember:

$$p(\boldsymbol{\theta} \mid \mathcal{H}_i, \mathcal{D}) = \frac{p(\mathcal{D} \mid \mathcal{H}_i, \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathcal{D} \mid \mathcal{H}_i)}$$

Bayesian model comparison

How to calculate the evidence

- The evidence is not the model likelihood!

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

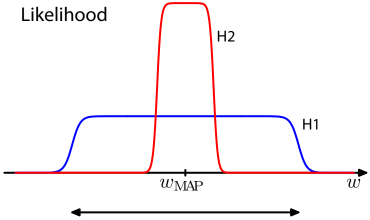
- Remember:

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

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

Bayesian model comparison
Ocam's razor

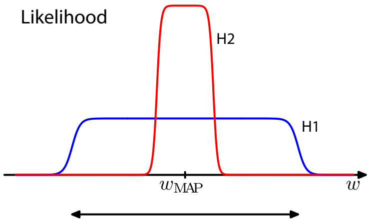
- The evidence integral penalizes overly complex models.
- A model with few parameters and lower maximum likelihood ($\mathcal{H}_1$) may win over a model with a peaked likelihood that requires many more parameters ($\mathcal{H}_2$).



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

Bayesian model comparison
Ocam'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 ($\mathcal{H}_2$).



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

Application to GWA

- Consider an association study.
  - $\mathcal{H}_0$: $p(\mathbf{y} \mid \mathcal{H}_0, \mathbf{X}, \boldsymbol{\theta}) = \mathcal{N}\left(\mathbf{y} \,\middle|\, \mathbf{0}, \sigma^2\mathbf{I}\right)$ (no association)
    $\boldsymbol{\theta} = \{\sigma^2\}$
  - $\mathcal{H}_1$: $p(\mathbf{y} \mid \mathcal{H}_1, \mathbf{X}, \boldsymbol{\theta}) = \mathcal{N}\left(\mathbf{y} \,\middle|\, \mathbf{w}^{\mathrm{T}} \cdot \mathbf{X}, \sigma^2\mathbf{I}\right)$ (linear association)
    $\boldsymbol{\theta} = \{\sigma^2, \mathbf{w}\}$
- Choosing conjugate priors for σ^2 and $\mathbf{w}$, the required integrals are tractable in closed form.

Application to GWA

- Consider an association study.
  - $\mathcal{H}_0$: $p(\mathbf{y} \mid \mathcal{H}_0, \mathbf{X}, \boldsymbol{\theta}) = \mathcal{N}\left(\mathbf{y} \,\middle|\, \mathbf{0}, \sigma^2\mathbf{I}\right)$ (no association)
    $\boldsymbol{\theta} = \{\sigma^2\}$
  - $\mathcal{H}_1$: $p(\mathbf{y} \mid \mathcal{H}_1, \mathbf{X}, \boldsymbol{\theta}) = \mathcal{N}\left(\mathbf{y} \,\middle|\, \mathbf{w}^{\mathrm{T}} \cdot \mathbf{X}, \sigma^2\mathbf{I}\right)$ (linear association)
    $\boldsymbol{\theta} = \{\sigma^2, \mathbf{w}\}$
- Choosing conjugate priors for σ^2 and $\mathbf{w}$, the required integrals are tractable in closed form.

Application to GWA

- Consider an association study.
  - $\mathcal{H}_0$: $p(\mathbf{y} \mid \mathcal{H}_0, \mathbf{X}, \boldsymbol{\theta}) = \mathcal{N}\left(\mathbf{y} \,\middle|\, \mathbf{0}, \sigma^2\mathbf{I}\right)$ (no association)
    $\boldsymbol{\theta} = \{\sigma^2\}$
  - $\mathcal{H}_1$: $p(\mathbf{y} \mid \mathcal{H}_1, \mathbf{X}, \boldsymbol{\theta}) = \mathcal{N}\left(\mathbf{y} \,\middle|\, \mathbf{w}^{\mathrm{T}} \cdot \mathbf{X}, \sigma^2\mathbf{I}\right)$ (linear association)
    $\boldsymbol{\theta} = \{\sigma^2, \mathbf{w}\}$
- Choosing conjugate priors for σ^2 and $\mathbf{w}$, the required integrals are tractable in closed form.

Application to GWA

Scoring models

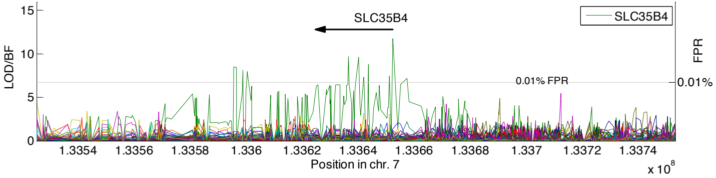
- The ratio of the evidences, the Bayes factor is a common scoring metric to compare two models:

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

Application to GWA
Scoring models

- The ratio of the evidences, the Bayes factor is a common scoring metric to compare two models:

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



Application to GWA

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 $\mathcal{H}_1$

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

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

Application to GWA

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 GWA

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.

Outline

Summary

- Curve fitting and linear regression.
- Maximum likelihood and least squares regression are identical.
- Construction of features using a mapping ϕ.
- Regularized least squares.
- Bayesian linear regression.
- Model comparison and ocam's razor.