ASSIGNMENT 7: MAXIMUM-A-POSTERIORI (MAP) ESTIMATION AND RIDGE REGRESSION

Institute for Machine Learning

JOHANNES KEPLER UNIVERSITY LINZ

Institute for Machine Learning

Contact

Heads: Johannes Kofler,
Markus Holzleitner

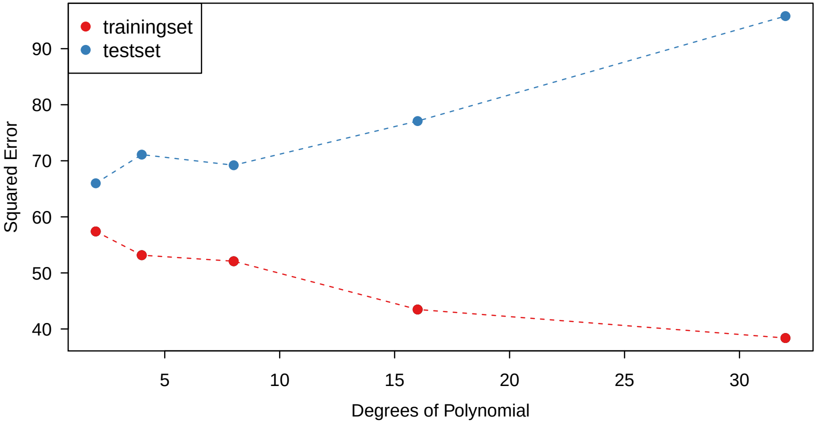
Institute for Machine Learning
Johannes Kepler University
Altenberger Str. 69
A-4040 Linz

E-Mail: `theoretical@ml.jku.at`
Only mails to this list are answered!
Institute Homepage

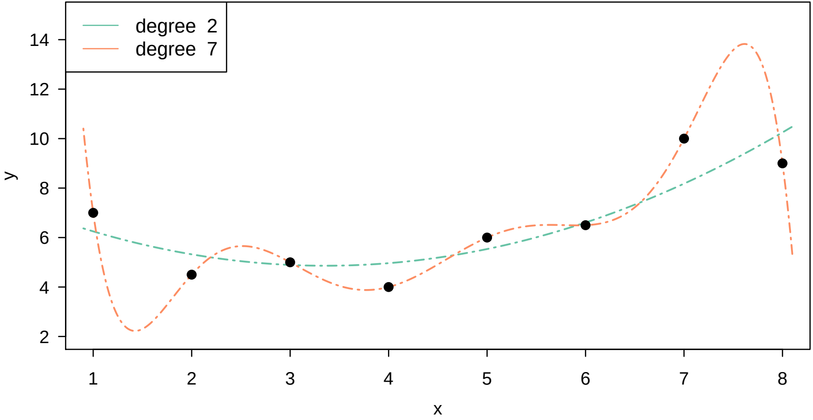
Copyright statement:

Maximum A Posteriori Probability (MAP) Estimation

Polynomial Regression: Training Error and Generalization Error



Overfitting



Decreasing influence of higher degrees of polynomial

- $g(x) = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + \cdots + w_7 x^7$
- Objective:

$$\min_{\mathbf{w}} L = \min_{\mathbf{w}} \sum_{i=1}^{n} \left(g(x_i; \mathbf{w}) - y_i\right)^2$$

- Don't let higher-order polynomials get too much influence:

$$\min_{\mathbf{w}} \sum \left(g(x_i; \mathbf{w}) - y_i\right)^2 + 1\,000\, w_5^2 + 1\,000\, w_6^2 + 1\,000\, w_7^2$$

- $\Rightarrow w_5, w_6, w_7 \approx 0$

Overfitting with many dimensions

- Too many dimensions/too little data to fit dimensions properly
- We can't tell a priori which dimensions to penalize
- Penalize all dimensions:

$$\min_{\mathbf{w}} \Bigg(\underbrace{\sum_{i=1}^{n} (g(\mathbf{x}_i; \mathbf{w}) - y_i)^2}_{\text{loss function}} + \underbrace{\lambda \sum_{j=1}^{m} w_j^2}_{\text{“weight decay” term}} \Bigg)$$

(by convention, w_0 is usually not penalized)

- λ is a hyperparameter (set by the user)
- this is a form of *regularization*

Regularization

- We know for the risk: $R = R_{\mathrm{empirical}} + \Omega_{\mathrm{complexity}}$
- We want to keep complexity low
- Regularization = Penalizing complexity of solutions
- *many* regularization methods exist

L_2 Weight Decay for Regression

- $\min_{\mathbf{w}} [\sum_{i=1}^{n} \left(g(x^i;\mathbf{w}) - y^i\right)^2 + \lambda \sum_{j=1}^{m} w_j^2]$
- $\lambda = 0 \Rightarrow$ standard Regression
- $\lambda = \infty \Rightarrow \mathbf{w} = 0$ (except w_0, straight line through mean)
- Equivalent: keep norm of $\mathbf{w}$ smaller than some given constant:

$$\min_{\mathbf{w}} \sum_{i=1}^{n} \left(g(x_i;\mathbf{w}) - y_i\right)^2$$

$$\text{s.t. } \sum_{j=1}^{m} w_j^2 \leq T$$

- This form of regularization has many names:
  - L_2 regularization (because it penalizes the L_2 norm of $\mathbf{w}$)
  - Gaussian weight prior/decay
  - Ridge regression (only used for Regression, but not in e.g. Neural Nets)
  - Tikhonov regularization

L_1 Weight Decay

- $\min_{\mathbf{w}} [\sum_{i=1}^{n} (g(x_i; \mathbf{w}) - y_i)^2 + \lambda \sum_{j=1}^{m} |w_j|]$
- Penalizes L_1 norm of $\mathbf{w}$
- Has many names:
  - L_1 regularization
  - Laplace weight prior/decay
  - LASSO (least absolute shrinkage and selection operator)

    (only used for Regression, but not in e.g. Neural Nets)
  - Sparsity penality term
- L_2 Regularization $\rightarrow$ small parameters
  L_1 Regularization $\rightarrow$ some parameters being exactly 0
- “sparse” = “contains many zeros”
- LASSO can be used for feature selection: most features “die out” ($w_i = 0$), only good(?) features survive

Bayes Theorem and MAP

- We are given training data $\{\mathbf{z}\} = \{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$
- We have a parametrized model of the data: $p(\{\mathbf{z}\}|\mathbf{w})$.

Remember that the likelihood $\mathcal{L}$ of data $\{\mathbf{z}\}$ to be produced by the model is:

$$\mathcal{L}(\{\mathbf{z}\}|\mathbf{w}) = p(\{\mathbf{z}\}|\mathbf{w}) = \prod_{i=1}^{n} p(\mathbf{z}_i|\mathbf{w})$$

Bayes Theorem and MAP

$$\mathcal{L}(\{\mathbf{z}\}|\mathbf{w}) = p(\{\mathbf{z}\}|\mathbf{w}) = \prod_{i=1}^{n} p(\mathbf{z}_i|\mathbf{w})$$

- Now assume, we have a prior belief about the distribution of $\mathbf{w}$.
- Applying Maximum Likelihood would ignore that belief
- Is it possible to incorporate this belief?

Bayes Theorem and MAP

Bayes Theorem gives us the relation:

$$p(\mathbf{w}|\{\mathbf{z}\}) = \frac{p(\{\mathbf{z}\}|\mathbf{w})\; p(\mathbf{w})}{p_w(\{\mathbf{z}\})}$$

where

- $p(\mathbf{w}|\{\mathbf{z}\})$ is called the "posterior distribution" or in short "posterior"
- $p(\mathbf{w})$ is called the "prior distribution" or short "prior"
- $p_w(\{\mathbf{z}\}) = \int_W p(\{\mathbf{z}\}|\mathbf{w})p(\mathbf{w})\, \mathrm{d}\mathbf{w}$ is called the "evidence"; does not depend on $\mathbf{w}$

Bayes Theorem and MAP

- In our case, the posterior is an "enriched" distribution over the weights
- Instead of maximizing the likelihood, we now can maximize the posterior distribution
- This is called the "MAP" (Maximum A Posteriori)
- MAP incorporates our prior belief about the distribution of $\mathbf{w}$

$$\mathbf{w}_{\mathrm{MAP}} = \arg\max_{\mathbf{w}} p(\mathbf{w}|\{\mathbf{z}\})$$

Bayes Theorem and MAP

- Like in Maximum Likelihood estimation we can apply the log-trick
- Note: the evidence does not depend on $\mathbf{w}$, thus it has no influence on optimization

$$
\begin{aligned}
\mathbf{w}_{\mathrm{MAP}} &= \arg\max_{\mathbf{w}} \; p(\mathbf{w}|\{\mathbf{z}\}) \\
&= \arg\max_{\mathbf{w}} \; \frac{p(\{\mathbf{z}\}|\mathbf{w})\; p(\mathbf{w})}{p_w(\{\mathbf{z}\})} \\
&= \arg\max_{\mathbf{w}} \; p(\{\mathbf{z}\}|\mathbf{w})\; p(\mathbf{w}) \\
&= \arg\min_{\mathbf{w}} \left(-log\; p(\{\mathbf{z}\}|\mathbf{w}) - log\; p(\mathbf{w})\right)
\end{aligned}
$$

Error Bars and Confidence Intervals: the big picture

- Even though we did our best to find good parameters $\mathbf{w}_{\mathrm{MAP}}$, we don't know how reliable our prediction is
- The Bayesian framework enables us to estimate this reliability to some extent
- This is because we have a distribution over weights!
- Two influences:
  - high inherent error in the data (high variance)
  - uncertainty how precisely $\mathbf{w}_{\mathrm{MAP}}$ could be chosen

Error Bars and Confidence Intervals

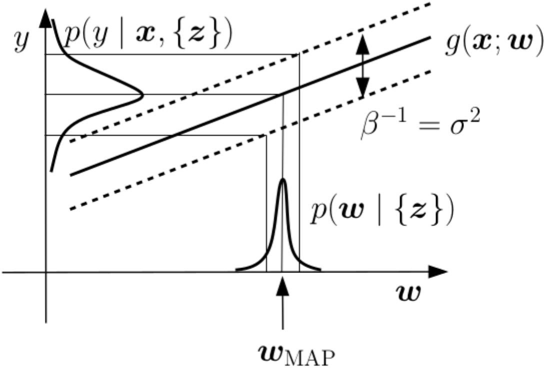


Figure 8.2: Error bars obtained by Bayes technique. Depicted is the double error line corresponding to 2σ. On the y-axis the error bars are given as quantiles of the distribution $p(y \mid \boldsymbol{x}; \{\boldsymbol{z}\})$. The large error bars result from the high inherent error $\beta^{-1} = \sigma^2$ of the data. The parameter $\boldsymbol{w}_{\mathrm{MAP}}$ has been chosen very precisely (e.g. if many training data points were available).

Error Bars and Confidence Intervals

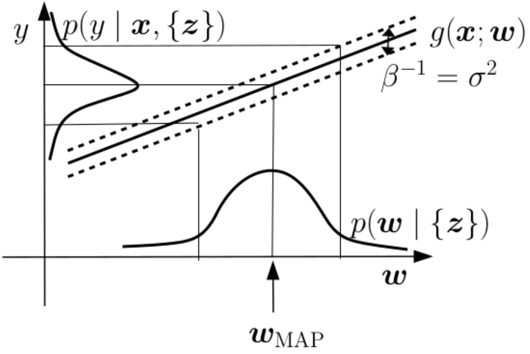


Figure 8.3: Error bars obtained by Bayes technique. As in Fig. 8.2 the double error line corresponds to 2σ but with much smaller noise variance in the data. On the y-axis the error bars are given as quantiles of the distribution $p(y \mid \boldsymbol{x}; \{\boldsymbol{z}\})$. The large error bars result from the broad posterior, that means the parameter $\boldsymbol{w}_{\mathrm{MAP}}$ has not been chosen very precisely (few data points or prior and data were not compatible).

Error Bars and Confidence Intervals

More formally:
For a given input vector $\mathbf{x}$ and with

- a given noise model (e.g. gaussian noise) $p(y|\mathbf{x}, \mathbf{w})$
- and the posterior distribution $p(\mathbf{w}|\{\mathbf{z}\})$

we can write the distribution of outputs the following way:

$$p(y|\mathbf{x}, \{\mathbf{z}\}) = \int_W p(y|\mathbf{x}, \mathbf{w})\, p(\mathbf{w}|\{\mathbf{z}\})\, \mathrm{d}\mathbf{w}$$

Thus, what we need is a approximation of the posterior $p(\mathbf{w}|\{\mathbf{z}\})$ and we need to know the noise model $p(y|\mathbf{x}, \mathbf{w})$

Approximation of the posterior

- all we have is an estimation of $\mathbf{w}_{\mathrm{MAP}}$
- but we want the distribution $p(\mathbf{w}|\{\mathbf{z}\})$
- Use a Taylor expansion around $\mathbf{w}_{\mathrm{MAP}}$ and approximate $p(\mathbf{w}|\{\mathbf{z}\})$ with a gaussian

The whole approximation: lecture notes, section 8.3

Estimating the output distribution

If $p(y|\mathbf{x},\mathbf{w})$ is a Gaussian noise model (see Section 4.1 in the lecture notes), then:

$$p(y|\mathbf{x},\mathbf{w}) = \frac{1}{Z_R(\beta)} \exp\Big(-\frac{\beta}{2}(y - g(\mathbf{x},\mathbf{w}))^2\Big)$$

where

$$Z_R(\beta) = \left(\frac{2\pi}{\beta}\right)^{n/2}$$

We approximate $g(\mathbf{x};\mathbf{w})$ around $\mathbf{w}_{\mathrm{MAP}}$ since we only know $\mathbf{w}_{\mathrm{MAP}}$

$$g(\mathbf{x};\mathbf{w}) = g(\mathbf{x};\mathbf{w}_{\mathrm{MAP}}) + \nabla g(\mathbf{x};\mathbf{w})^T(\mathbf{w} - \mathbf{w}_{\mathrm{MAP}})$$

plug this and the posterior approximation in the integral $\int_W p(y|\mathbf{x},\mathbf{w})p(\mathbf{w}|\{\mathbf{z}\})d\mathbf{w}$:

$$p(y|\mathbf{x},\mathbf{w}) = \frac{1}{\sqrt{2\pi}\sigma_y} \exp\Big(-\frac{1}{2\sigma_y^2}(y - g(\mathbf{x};\mathbf{w}_{\mathrm{MAP}}))^2\Big)$$

Literature Tip

Christopher M. Bishop:

"Neural Networks for Pattern Recognition", Chapter 10

In case the lecture notes are confusing ;-)