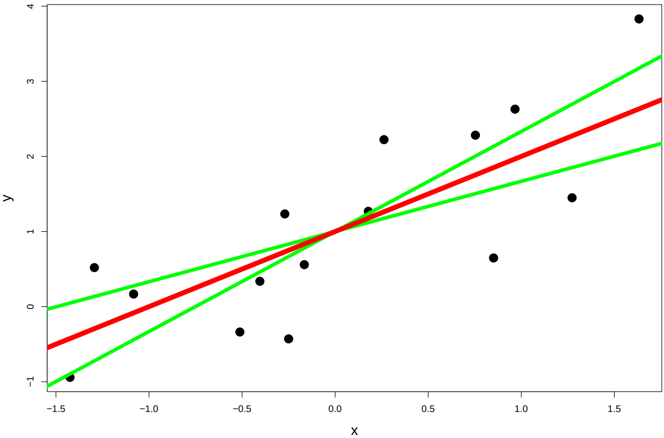
Machine Learning

Volker Roth

Department of Mathematics & Computer Science
University of Basel

Chapter 4: Regression



Least-squares fit (red) and two lines with slopes according to upper (lower) 95% confidence limit (green).

Regression basics

- In regression we assume that a response variable $y \in \mathbb{R}$ is a noisy function of the input variable $\boldsymbol{x} \in \mathbb{R}^d$.

  $$y = f(\boldsymbol{x}) + \eta.$$

- We often assume that f is linear, $f(\boldsymbol{x}) = \boldsymbol{w}^t \boldsymbol{x}$, and that η has a zero-mean Gaussian distribution with constant variance, $\eta \sim N(0, \sigma^2)$.
- This is can equivalently be written as

  $$p(y|\boldsymbol{x}) = N(\mu(\boldsymbol{x}), \sigma^2), \text{ with } \mu(\boldsymbol{x}) = \boldsymbol{w}^t \boldsymbol{x}.$$

- In one dimension: $\mu(\boldsymbol{x}) = w_0 + w_1 x$ and $\boldsymbol{x} = (1, x)$.
  w_0 is the **intercept** or bias term and w_1 is the **slope.**
- If $w_1 > 0$, we expect the output to increase as the input increases.

Least Squares and Maximum Likelihood

- Fit n data points $(\boldsymbol{x}_i, y_i)$ to a model that has $d+1$ parameters $w_j,\ j = 0, \dots, d$.
- Notation: $\boldsymbol{x} \leftarrow (1, \boldsymbol{x}) \rightsquigarrow w_0$ is the intercept.
- Frequentist view: $\boldsymbol{w}$ is an unknown parameter vector, not a RV.
- We assume that the n observations are **iid**.
- **Linear model:** $y_i = \boldsymbol{w}^t \boldsymbol{x}_i + \eta_i, \quad \eta_i \sim N(0, \sigma^2)$.
  Observed y_i generated from a normal distribution centered at $\boldsymbol{w}^t \boldsymbol{x}_i$.
- Model predicts linear relationship between **conditional expectation** of **observations** y_i and **inputs** $\boldsymbol{x}_i$:

  $$E[y_i|\boldsymbol{x}_i] = w_0 + w_1 x_{i1} + \cdots + w_d x_{id} = \boldsymbol{w}^t \boldsymbol{x}_i = f(\boldsymbol{x}_i; \boldsymbol{w}).$$

  Note: the expectation operator is linear and $E[\eta_i] = 0$.

  Regression function = conditional expectation.

LS and Maximum Likelihood

- **Likelihood function:** conditional probability of all observed y_i given their explanation, treated as a function of the model parameters $\boldsymbol{w}$:

$$L(\boldsymbol{w}) \propto \prod_i \exp\left[-\frac{1}{2\sigma^2}(y_i - \boldsymbol{w}^t\boldsymbol{x}_i)^2\right]$$

- **Maximizing L = finding model that best explains observations:**

$$\begin{aligned} \hat{\boldsymbol{w}} &= \arg\max_{\boldsymbol{w}} L(\boldsymbol{w}) = \arg\min_{\boldsymbol{w}}[-L(\boldsymbol{w})] = \arg\min_{\boldsymbol{w}}[-\log(L(\boldsymbol{w}))] \\ &= \arg\min_{\boldsymbol{w}} \sum_i (y_i - \boldsymbol{w}^t\boldsymbol{x}_i)^2 \end{aligned}$$

Least-squares fit = ML estimate under Gaussian error model.

- $\hat{\boldsymbol{w}}_{MLE}$ minimizes the **residual sum of squares**

$$RSS(\boldsymbol{w}) = \sum_{i=1}^n r_i^2 = \sum_{i=1}^n [y_i - f(\boldsymbol{x}_i; \boldsymbol{w})]^2 = \|\boldsymbol{y} - X\boldsymbol{w}\|^2.$$

LS and Maximum Likelihood

- Finding the optimal weights:

$$\frac{\partial RSS(\boldsymbol{w})}{\partial \boldsymbol{w}} = \frac{\partial}{\partial \boldsymbol{w}} \left[\boldsymbol{y}^t \boldsymbol{y} - 2\boldsymbol{y}^t X \boldsymbol{w} + \boldsymbol{w}^t X^t X \boldsymbol{w}\right] \overset{!}{=} \mathbf{0}$$

- Using the following results from matrix calculus,

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{x}} \boldsymbol{y}^t \boldsymbol{x} &= \boldsymbol{y} \\ \frac{\partial}{\partial \boldsymbol{x}} \boldsymbol{x}^t M \boldsymbol{x} &= 2M\boldsymbol{x}, \text{ if } M \text{ is symmetric,} \end{aligned}$$

  we finally arrrive at

$$\hat{\boldsymbol{w}} = (X^t X)^{-1} X^t \boldsymbol{y}.$$

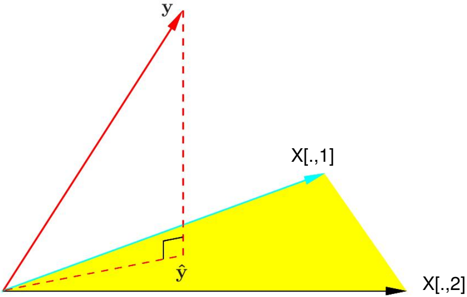
Least squares regression: Geometry

The **residual is $\boldsymbol{r} = \boldsymbol{y} - X\boldsymbol{w}$**. **Gradient at $\boldsymbol{w} = \hat{\boldsymbol{w}}$ vanishes.**

$$\hat{\boldsymbol{w}} = (X^t X)^{-1} X^t \boldsymbol{y} \;\Rightarrow\; X^t(\boldsymbol{y} - X\hat{\boldsymbol{w}}) = X^t \boldsymbol{r} = \boldsymbol{0}.$$

If follows that $\sum_{i=1}^{n} X_{ij} r_i = 0, \; \forall j = 0, 1, \ldots, d$.

$\rightsquigarrow$ **Residual is orthogonal to every input dimension $X_{\bullet j}$.**



Adapted from Fig. 3.2 in (Hastie, Tibshirani, Friedman: The Elements of Statistical Learning Theory. Springer)

Frequentist confidence limits

- **Recall:** $y_i = f(\boldsymbol{x}_i; \boldsymbol{w}) + \eta_i$, with independent Gaussian noise.
- In matrix-vector form: $\boldsymbol{y} = X\boldsymbol{w} + \boldsymbol{\eta}$, with $\boldsymbol{\eta} \sim N(\boldsymbol{0}, \sigma^2 I_n)$.

$$\begin{aligned} \hat{\boldsymbol{w}} &= (X^tX)^{-1}X^t\boldsymbol{y} \\ &= (X^tX)^{-1}X^tX\boldsymbol{w} + (X^tX)^{-1}X^t\boldsymbol{\eta} \\ &= \boldsymbol{w} + (X^tX)^{-1}X^t\boldsymbol{\eta} \end{aligned}$$

$$\Rightarrow \quad \hat{\boldsymbol{w}} - \boldsymbol{w} = (X^tX)^{-1}X^t\boldsymbol{\eta} =: A\boldsymbol{\eta}$$

- **Linear** functions of normals are normal:

$$\boldsymbol{\eta} \sim N(\boldsymbol{0}, \sigma^2 I_n) \;\Rightarrow\; A\boldsymbol{\eta} \sim N(\boldsymbol{0}, \sigma^2 AA^t).$$

  Here: $A = (X^tX)^{-1}X^t \;\Rightarrow\; AA^t = (X^tX)^{-1}$

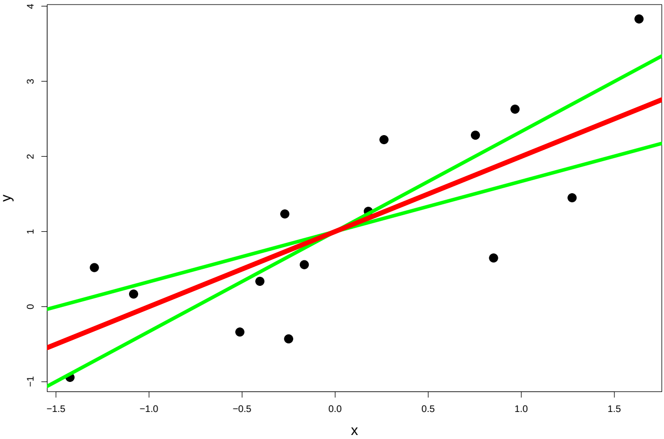
- Conditioned on X and σ^2:

$$\hat{\boldsymbol{w}} - \boldsymbol{w} | X, \sigma^2 \sim N\left(\boldsymbol{0}, \sigma^2(X^tX)^{-1}\right).$$

Frequentist confidence limits

- Distribution completely specified $\leadsto$ **confidence limits:** For k-th component: $\hat{w}_k - w_k \sim N(0, \sigma^2 S^{kk})$, where S^{kk} denotes the k-th diagonal element of $(X^t X)^{-1}$.
- Thus, z_k is standard normal
$$z_k := (w_k - \hat{w}_k)/\sqrt{\sigma^2 S^{kk}} \sim N(0,1)$$
- CDF:
$$P(z_k < k_c) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{k_c} e^{-t^2/2}\, dt =: \Phi(k_c) = 1 - c$$
- **Upper limit** for w_k:
$$\begin{aligned} P(z_k < k_c) &= P(\sqrt{\sigma^2 S^{kk}} z_k < \sqrt{\sigma^2 S^{kk}} k_c) \\ &= P(w_k - (w_k - \hat{w}_k) > w_k - \sqrt{\sigma^2 S^{kk}} k_c) \\ &= P(\hat{w}_k > w_k - \sqrt{\sigma^2 S^{kk}} k_c) \\ &= P(w_k < \hat{w}_k + \sqrt{\sigma^2 S^{kk}} k_c) = 1 - c. \end{aligned}$$
- Same argument for $z'_k = -z_k$ $\leadsto$ **lower limit.**

Frequentist confidence limits



Least-squares fit (red) and two lines with slopes according to upper (lower) 95% confidence limit (green).

Standard parametric rate

- Assume we have estimated the parameters based on n samples:

$$\begin{aligned}(\hat{\boldsymbol{w}}_n - \boldsymbol{w}) &\sim N(\mathbf{0}, \sigma^2 \left(X^t X\right)^{-1}) \\ &= N(\mathbf{0}, \sigma^2 \left(X^t X/n\right)^{-1} \cdot 1/n) \\ \sqrt{n}(\hat{\boldsymbol{w}}_n - \boldsymbol{w}) &\sim N(\mathbf{0}, \sigma^2 \underbrace{\left(X^t X/n\right)}_{\overset{n\to\infty}{\to}\Sigma}{}^{-1})\end{aligned}$$

- Since for $n \to \infty$, $X^t X/n \to \Sigma = const.$, this means that **$\hat{\boldsymbol{w}}_n$ converges to $\boldsymbol{w}$ at a rate of $1/\sqrt{n}$.**
- This is a very general result that holds in an asymptotic sense even without assuming normality, due to the **central limit theorem.**
- Due to its universality, it is called the **standard parametric rate.**
- Equivalent statement:
  $1/\sqrt{n}$ represents the **magnitude of the estimation error.**

Basis functions

- Can be generalized to model non-linear relationships by replacing $\boldsymbol{x}$ with some non-linear function of the inputs, $\phi(\boldsymbol{x})$:

$$p(y|\boldsymbol{x}) = N(\boldsymbol{w}^t \phi(\boldsymbol{x}), \sigma^2).$$

- Predictions can be based on a linear combination of a set of basis functions $\phi(\boldsymbol{x}) = \{g_0(\boldsymbol{x}), g_1(\boldsymbol{x}), \ldots, g_m(\boldsymbol{x})\}$, with $g_i(\boldsymbol{x}) : \mathbb{R}^d \mapsto \mathbb{R}$. Can model the intercept by setting $g_0(\boldsymbol{x}) = 1$:

$$f(\boldsymbol{x}; \boldsymbol{w}) = w_0 + w_1 g_1(\boldsymbol{x}) + \cdots + w_m g_m(\boldsymbol{x}).$$

$\rightsquigarrow$ **additive models**

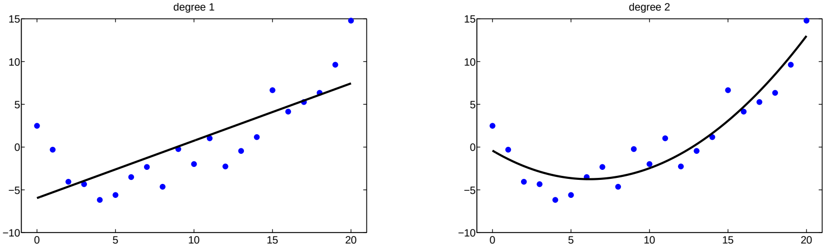


Fig 1.7 in K. Murphy: Machine Learning. MIT Press 2012

Additive models

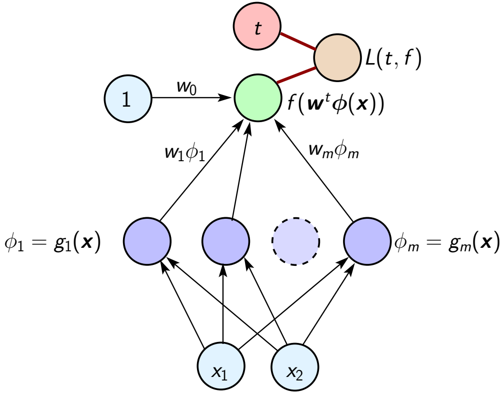
- Examples:
  If $x \in \mathbb{R}^d$ and $m = d+1$, $g_0(\boldsymbol{x}) = 1$ and $g_i(\boldsymbol{x}) = x_i$, $i = 1, \dots, d$, then
  $$f(\boldsymbol{x}; \boldsymbol{w}) = w_0 + w_1 x_1 + \cdots + w_d x_d.$$
  If $x \in \mathbb{R}$, $g_0(\boldsymbol{x}) = 1$ and $g_i(x) = x^i$, $i = 1, \dots, m$, then
  $$f(x; \boldsymbol{w}) = w_0 + w_1 x^1 + \cdots + w_m x^m.$$
- Basis functions can **capture various properties of the inputs**.
  Example: **Document analysis**
  $$\begin{aligned} \boldsymbol{x} &= \text{text document (collection of words)} \\ g_i(\boldsymbol{x}) &= \begin{cases} 1, & \text{if word i appears in the document} \\ 0, & \text{otherwise} \end{cases} \\ f(\boldsymbol{x}; \boldsymbol{w}) &= w_0 + \sum_{i \in \text{words}} w_i g_i(\boldsymbol{x}). \end{aligned}$$

Additive models cont'd

- We can also make predictions by gauging the **similarity of examples to prototypes.**
- For example, our additive regression function could be
  $$f(\boldsymbol{x};\boldsymbol{w}) = w_0 + w_1 g_1(\boldsymbol{x}) + \cdots + w_m g_m(\boldsymbol{x}),$$
  where the basis functions are **radial basis functions**
  $$g_k(\boldsymbol{x}) = \exp(-\frac{1}{2\sigma^2}\|\boldsymbol{x} - \boldsymbol{x}_k\|^2)$$
  measuring the similarity to the prototypes $\boldsymbol{x}_k$.
- The variance σ^2 controls how quickly the basis function vanishes as a function of the distance to the prototype.
- **Training examples themselves could serve as prototypes.**

Additive models cont'd

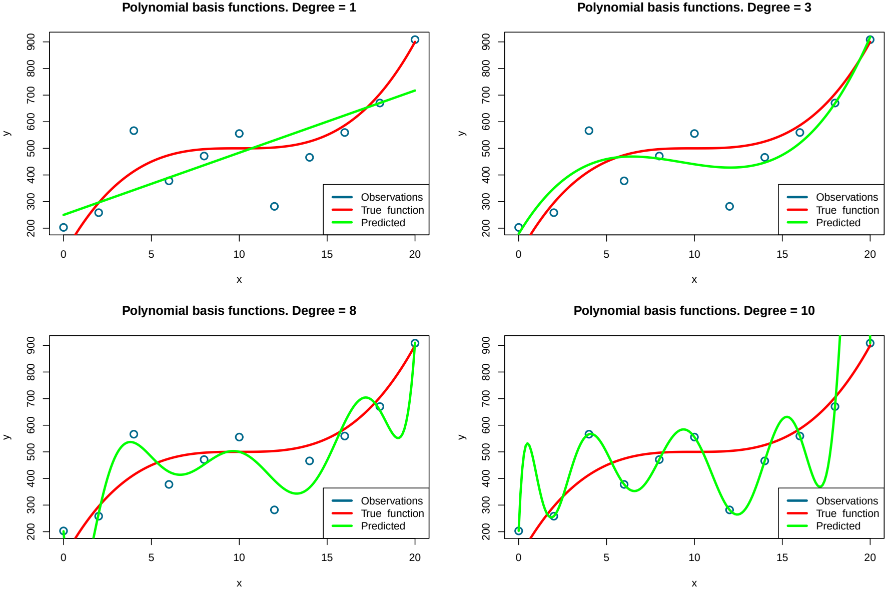
Can view additive models graphically in terms of **units** and **weights**.



t
$L(t, f)$
1
w_0
$f(\boldsymbol{w}^t\phi(\boldsymbol{x}))$
$w_1\phi_1$
$w_m\phi_m$
$\phi_1 = g_1(\boldsymbol{x})$
$\phi_m = g_m(\boldsymbol{x})$
x_1
x_2

In **Multi Layer Perceptrons** the basis functions have adjustable parameters.

Example: Polynomial regression



Complexity and overfitting

With limited training examples our polynomial regression model may achieve zero training error but nevertheless has a large expected error.

$$\text{training} \qquad \frac{1}{n}\sum_{i=1}^{n}(y_i - f(\boldsymbol{x}_i; \hat{\boldsymbol{w}})^2 \approx 0$$

$$\text{expectation} \qquad E_{(\boldsymbol{x},y)\sim p}\,(y - f(\boldsymbol{x}; \hat{\boldsymbol{w}})^2 \gg 0$$

We suffer from **over-fitting**
$\rightsquigarrow$ should reconsider our model $\rightsquigarrow$ **model selection.**

We will discuss model selection from a **Bayesian perspective** first.
A frequentist approach will follow later in the chapter on **statistical learning theory.**

Bayesian interpretation: priors

- Suppose our generative model takes an input $\boldsymbol{x} \in \mathbb{R}^d$ and maps it to a real valued output y according to
$$p(y|\boldsymbol{x}, \boldsymbol{w}, \sigma^2) = N(y|\boldsymbol{w}^t\boldsymbol{x}, \sigma^2)$$
- We will keep σ^2 fixed and only try to estimate $\boldsymbol{w}$.
- Given data $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_n, y_n)\}$, the **likelihood function** is
$$L(\boldsymbol{w}; \mathcal{D}) = \prod_{i=1}^n N(y_i|\boldsymbol{w}^t\boldsymbol{x}_i, \sigma^2) = \prod_{i=1}^n \frac{1}{Z} \exp\left(-\frac{1}{2\sigma^2}(y_i - \boldsymbol{w}^t\boldsymbol{x}_i)^2\right).$$
- Predictions in classical regression based on maximizing parameters $\hat{\boldsymbol{w}}$.
- **In Bayesian analysis we keep all regression functions**, just weighted by their **posterior probability:**
$$p(y|\boldsymbol{x}, \mathcal{D}, \sigma^2) = \int p(y|\boldsymbol{x}, \boldsymbol{w}, \sigma^2)\, p(\boldsymbol{w}|\mathcal{D}, \sigma^2)\, d\boldsymbol{w}$$

Bayesian regression: Prior and posterior

- We specify our **prior belief** about the parameter values as $p(\boldsymbol{w})$. For instance, we could prefer small parameter values:

  $$p(\boldsymbol{w}) = N\left(\boldsymbol{w}|0, \tau^2 I\right)$$

  The smaller τ^2 is, the smaller values of $\boldsymbol{w}$ we prefer **prior to seeing the data**.
- **Posterior** proportional to prior $p(\boldsymbol{w})$ times likelihood:

  $$p(\boldsymbol{w}|\mathcal{D}, \cdot) \propto L(\boldsymbol{w}; \mathcal{D}) p(\boldsymbol{w})$$
- Here: posterior is **Gaussian** $p(\boldsymbol{w}|\mathcal{D}, \sigma^2) = N(\boldsymbol{w}|\boldsymbol{w}_n, V_n)$ with conditional mean $\boldsymbol{w}_n$ and conditional covariance V_n (i.e. conditioned on dataset of size n) given by

  $$\boldsymbol{w}_n = (X^t X + \lambda I)^{-1} X^t \boldsymbol{y}, \quad V_n = \sigma^2 (X^t X + \lambda I)^{-1},$$

  with $\lambda = \frac{\sigma^2}{\tau^2}$.

Bayesian regression: Posterior computation

Given variables $\boldsymbol{w} \in \mathbb{R}^d$ and $\boldsymbol{y} \in \mathbb{R}^n$, assume **linear Gaussian system:**

$$
\begin{aligned}
p(\boldsymbol{w}) &= N(\boldsymbol{w}|\boldsymbol{\mu}_w, \Sigma_w) \quad (\rightsquigarrow \text{prior}) \\
p(\boldsymbol{y}|\boldsymbol{w}) &= N(\boldsymbol{y}|A\boldsymbol{w} + \boldsymbol{b}, \Sigma_y) \quad (\rightsquigarrow \text{likelihood})
\end{aligned}
$$

- **The posterior is also Gaussian** with conditional mean $\boldsymbol{\mu}_{w|y}$ and conditional covariance $\Sigma_{w|y}$:

$$
\begin{aligned}
p(\boldsymbol{w}|\boldsymbol{y}) &= N(\boldsymbol{w}|\boldsymbol{\mu}_{w|y}, \Sigma_{w|y}) \\
\Sigma_{w|y}^{-1} &= \Sigma_w^{-1} + A^t \Sigma_y^{-1} A \\
\boldsymbol{\mu}_{w|y} &= \Sigma_{w|y} \left(A^t \Sigma_y^{-1} (\boldsymbol{y} - \boldsymbol{b}) + \Sigma_w^{-1} \boldsymbol{\mu}_w \right).
\end{aligned}
$$

Gaussian likelihood and Gaussian prior form a conjugate pair.

- The normalization constant (denominator in Bayes formula) is

$$
p(\boldsymbol{y}) = N(\boldsymbol{y}|A\boldsymbol{\mu}_w + \boldsymbol{b}, \Sigma_y + A\Sigma_w A^t).
$$

Bayesian regression: Posterior predictive

- Prediction of y for new $\boldsymbol{x}$: use posterior as weights for predictions based on individual $\boldsymbol{w}$'s $\rightsquigarrow$ **Posterior predictive:**

$$\begin{aligned} p(y|\boldsymbol{x},\mathcal{D},\sigma^2) &= \int p(y|\boldsymbol{x},\boldsymbol{w},\sigma^2)p(\boldsymbol{w}|\mathcal{D},\sigma^2)\,d\boldsymbol{w} \\ &= \int N(y|\boldsymbol{x}^t\boldsymbol{w},\sigma^2)N(\boldsymbol{w}|\boldsymbol{w}_n,V_n)\,d\boldsymbol{w} \\ &= N(y|\boldsymbol{w}_n^t\boldsymbol{x},\sigma_n^2(\boldsymbol{x})), \text{ with} \\ \sigma_n^2(\boldsymbol{x}) &= \sigma^2 + \boldsymbol{x}^t V_n \boldsymbol{x}. \end{aligned}$$

- The variance in this prediction, $\sigma_n^2(\boldsymbol{x})$, depends on two terms:
  - the variance of the observation noise, σ^2
  - the variance in the parameters, V_n

    $\rightsquigarrow$ depends on how close $\boldsymbol{x}$ is to training data $\mathcal{D}$

    $\rightsquigarrow$ error bars get larger as we move away from training points.

Bayesian regression: Posterior predictive

- By contrast, the **plugin approximation** uses only the ML-parameter estimate with the degenerate distribution $p(\mathbf{w}|\mathcal{D},\sigma^2) = \delta_{\hat{\mathbf{w}}}(\mathbf{w})$:
$p(\mathbf{y}|\mathbf{x},\mathcal{D},\sigma^2) \approx \int p(y|\mathbf{x},\mathbf{w},\sigma^2)\delta_{\hat{\mathbf{w}}}(\mathbf{w})\,d\mathbf{w} = p(y|\mathbf{x},\hat{\mathbf{w}},\sigma^2) = N(y|\mathbf{x}^t\hat{\mathbf{w}},\sigma^2)$.

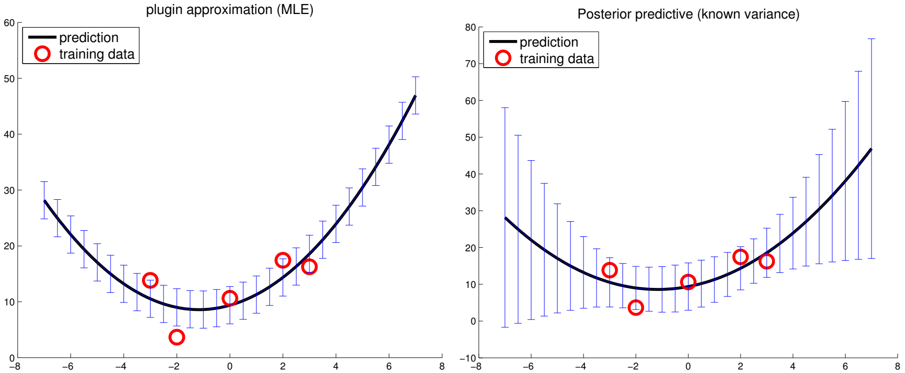


Fig. 7.12 in K. Murphy: Machine Learning. MIT Press 2012. Example with quadratic basis functions: posterior predictive distribution (mean and $\pm 1\sigma$).

Sampling from posterior predictive

Left: plugin approximation: $f(y) = \phi(\boldsymbol{x})^t \hat{\boldsymbol{w}}$,
where $\phi(\boldsymbol{x})$ is the expanded input vector $(1, x, x^2)^t$.
Right: sampled functions $\phi(\boldsymbol{x})^t \boldsymbol{w}^{(s)}$, where $w^{(s)}$ are samples from the posterior

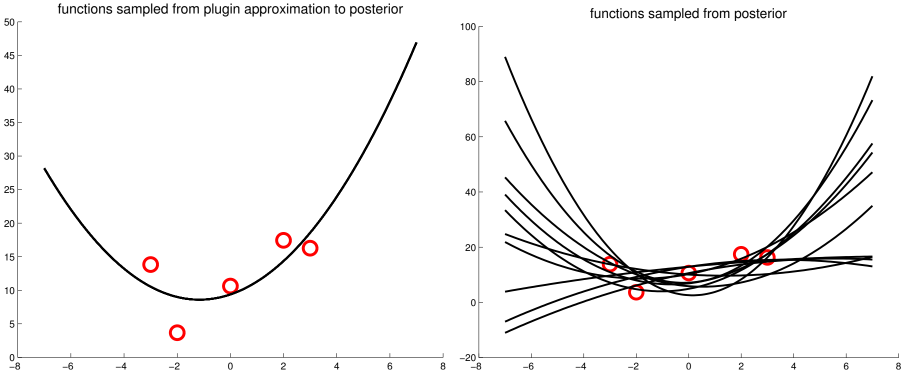


Fig. 7.12 in K. Murphy: Machine Learning. MIT Press 2012

MAP approximation and ridge regression

- Posterior proportional to prior $p(\boldsymbol{w}) = N\left(\boldsymbol{w}|0, \tau^2 I\right)$ times likelihood.
- The MAP estimate is

$$\begin{aligned}
\boldsymbol{w}_{\text{MAP}} &= \arg\max\{\log[L(\boldsymbol{w}; \mathcal{D})] + \log[p(\boldsymbol{w})]\} \\
&= \arg\min\{-\log[L(\boldsymbol{w}; \mathcal{D})] - \log[p(\boldsymbol{w})]\} \\
&= \arg\min\{\frac{1}{2\sigma^2}\sum_i (y_i - \boldsymbol{w}^t\boldsymbol{x}_i)^2 + \frac{1}{2\tau^2}\boldsymbol{w}^t\boldsymbol{w}\} \\
&= \arg\min\{\sum_i (y_i - \boldsymbol{w}^t\boldsymbol{x}_i)^2 + \frac{\sigma^2}{\tau^2}\boldsymbol{w}^t\boldsymbol{w}\} \\
&= \arg\min\{\sum_i (y_i - \boldsymbol{w}^t\boldsymbol{x}_i)^2 + \lambda\boldsymbol{w}^t\boldsymbol{w}\}
\end{aligned}$$

- In classical statistics, this is called **ridge regression:**

$$\boldsymbol{w}_{\text{MAP}} = \boldsymbol{w}_{\text{ridge}} = (X^t X + \lambda I)^{-1} X^t \boldsymbol{y}.$$

- In regularization theory, this is an example of **Tikhonov Regularization.**

Bayesian regression (again)

- Suppose our model within the **model family $\mathcal{F}$** takes an input $\boldsymbol{x} \in \mathbb{R}^d$ and maps it to a real valued output y according to
$$p(y|\boldsymbol{x}, \boldsymbol{w}, \sigma^2) = N(y; \boldsymbol{w}^t \boldsymbol{x}, \sigma^2)$$
- We will keep σ^2 fixed and only try to estimate $\boldsymbol{w}$.
- Given data $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_n, y_n)\}$, define likelihood
$$L(\boldsymbol{w}; \mathcal{D}) = \prod_{i=1}^{n} N(y_i; \boldsymbol{w}^t \boldsymbol{x}_i, \sigma^2) = \prod_{i=1}^{n} \frac{1}{Z} \exp\left(-\frac{1}{2\sigma^2}(y_i - \boldsymbol{w}^t \boldsymbol{x}_i)^2\right).$$
- Predictions in classical regression based on maximizing parameters $\hat{\boldsymbol{w}}$.
- **In Bayesian analysis we keep all regression functions**, just weighted by their **posterior probability:**
$$p(y|\boldsymbol{x}, \mathcal{D}, \sigma^2) = \int p(y|\boldsymbol{x}, \boldsymbol{w}, \sigma^2) p(\boldsymbol{w}|\mathcal{D}, \sigma^2)\, d\boldsymbol{w}$$

Bayesian regression (again)

- We specify our **prior belief** about the parameter values as $p(\boldsymbol{w}|\mathcal{F})$. For instance, we could prefer small parameter values:

  $$p(\boldsymbol{w}|\mathcal{F}) = N(\boldsymbol{w}; 0, \tau^2 I)$$

  Small τ^2 ⇝ small values $\boldsymbol{w}$ preferred **prior to seeing the data.**

- Posterior proportional to prior times likelihood:

  $$p(\boldsymbol{w}|\mathcal{D}, \cdot) = \frac{p(\boldsymbol{y}|\boldsymbol{w}, X)p(\boldsymbol{w}|\mathcal{F})}{p(\boldsymbol{y}|\mathcal{F}, X)} \propto L(\boldsymbol{w}; \mathcal{D})p(\boldsymbol{w}|\mathcal{F})$$

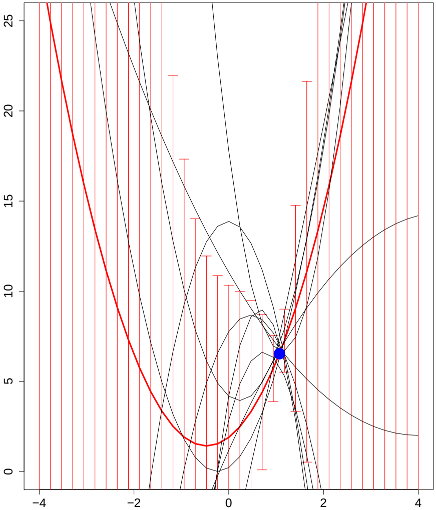
- Normalization constant, a.k.a. **marginal likelihood:**

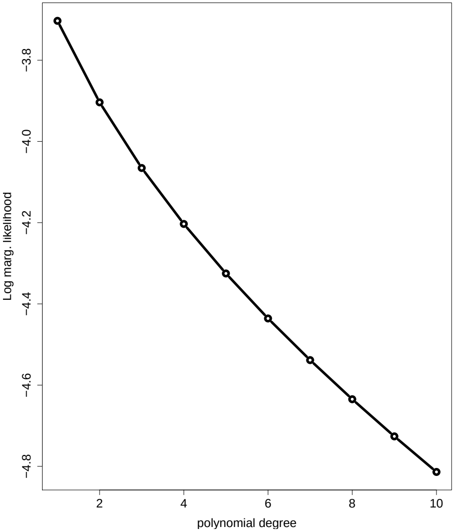
  $$p(\boldsymbol{y}|\mathcal{F}, X) = \int \underbrace{L(\boldsymbol{w}; \mathcal{D})}_{p(\boldsymbol{y}|\boldsymbol{w}, X)} p(\boldsymbol{w}|\mathcal{F}) d\boldsymbol{w} = \int p(\boldsymbol{y}, \boldsymbol{w}|\mathcal{F}, X) d\boldsymbol{w},$$

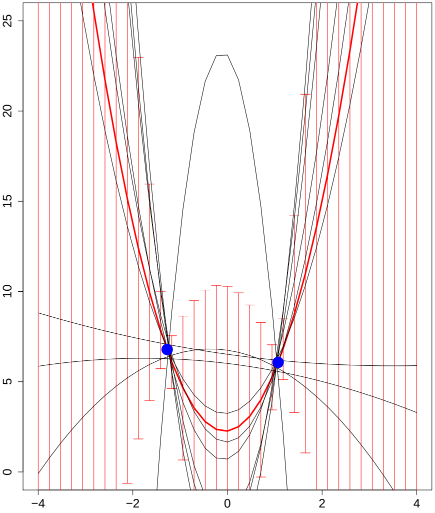
  depends on model family $\mathcal{F}$, but not on parameter values of a specific model in the family.

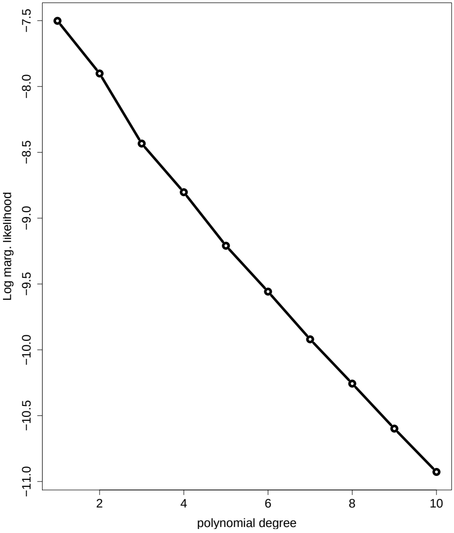
Example: Bayesian regression

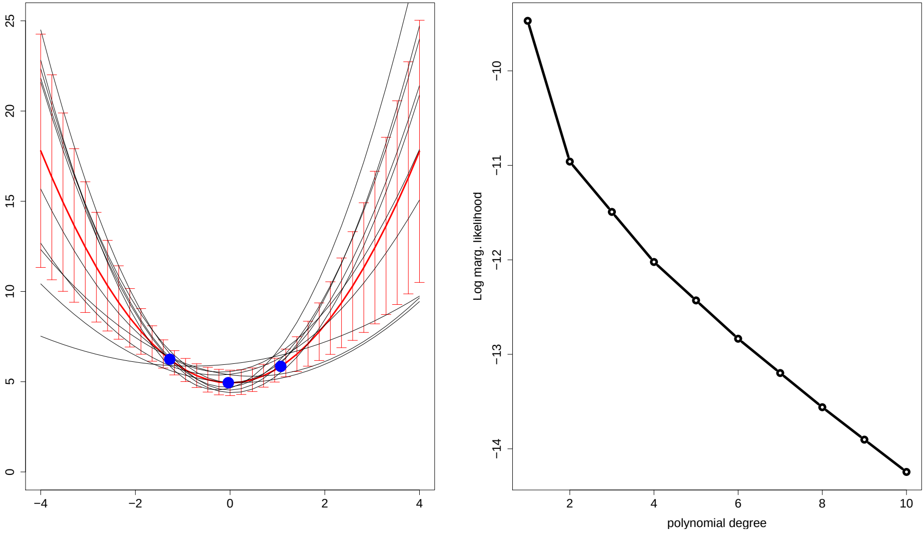
- **Goal: choose among regression model families**, specified by different feature mappings (basis functions) $\boldsymbol{x} \to \phi(\boldsymbol{x})$.
- Example: linear $\phi_1(\boldsymbol{x}) \in \mathbb{R}^{d_1}$ and quadratic $\phi_2(\boldsymbol{x}) \in \mathbb{R}^{d_2}$.
- For both families, we specify a Gaussian regression model:
  $$\mathcal{F}_i \quad : \quad p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{w}_i, \sigma^2) = N(\boldsymbol{y}|\boldsymbol{w}_i^t \phi_i(\boldsymbol{x}), \sigma^2), \quad i \in \{1, 2\}.$$
- Considering the posterior predictive, there are two possibilities:
  - $\mathcal{F}$ **too flexible:** posterior requires many training examples before it focuses on useful parameter values;
  - $\mathcal{F}$ **too simple:** posterior concentrates quickly but the predictions remain poor. **But how can we formalize this intuition?**
- **Posterior of model family:** $p(\mathcal{F}|\boldsymbol{y}, X) \propto p(\boldsymbol{y}|\mathcal{F}, X) P(\mathcal{F})$.
- Pragmatic choice: Uniform prior over families $\rightsquigarrow$ select the family whose **marginal likelihood** (a.k.a. Bayesian score) **is larger.**
- After seeing $\mathcal{D}$, select family $\mathcal{F}_1$ if $p(\boldsymbol{y}|\mathcal{F}_1, X) > p(\boldsymbol{y}|\mathcal{F}_2, X)$.

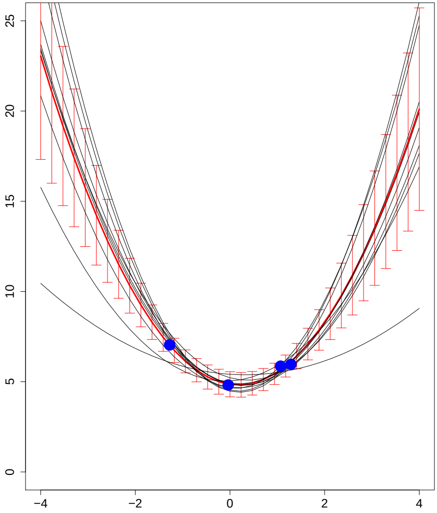

25
20
15
10
5
0
−4
−2
0
2
4

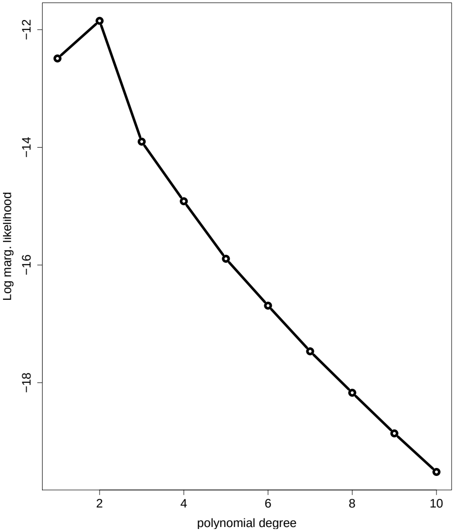

Log marg. likelihood
−3.8
−4.0
−4.2
−4.4
−4.6
−4.8
2
4
6
8
10
polynomial degree

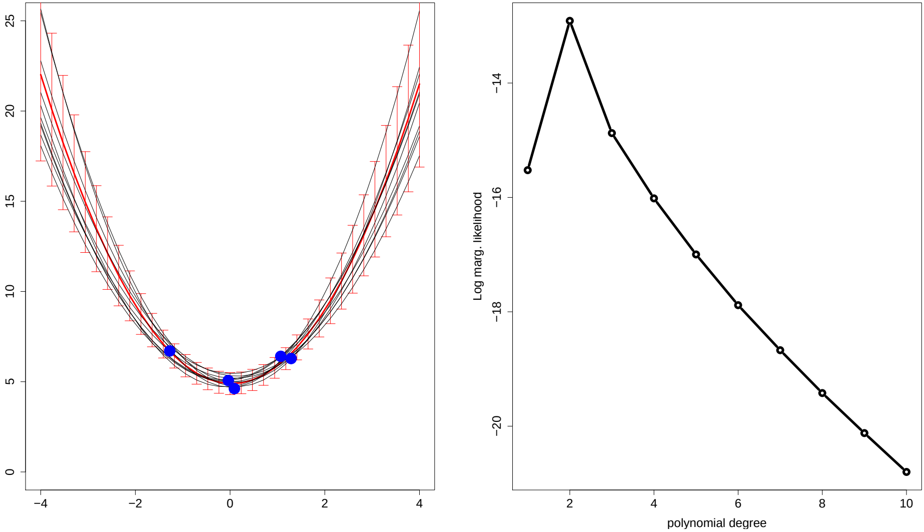

25
20
15
10
5
0
−4
−2
0
2
4

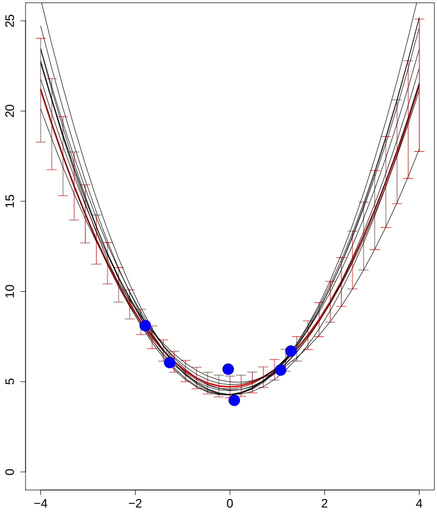

Log marg. likelihood
−7.5
−8.0
−8.5
−9.0
−9.5
−10.0
−10.5
−11.0
2
4
6
8
10
polynomial degree


0
5
10
15
20
25
−4
−2
0
2
4
Log marg. likelihood
−10
−11
−12
−13
−14
2
4
6
8
10
polynomial degree

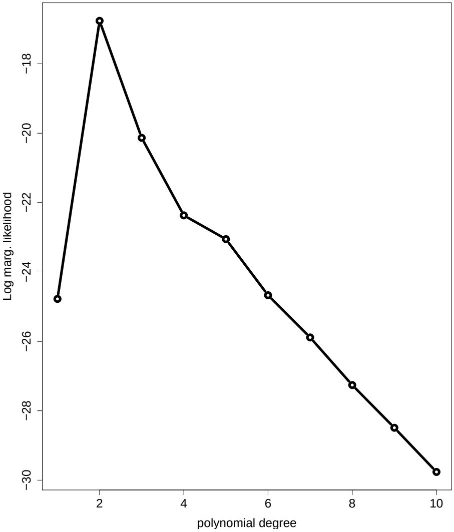

25
20
15
10
5
0
−4
−2
0
2
4


−12
−14
−16
−18
Log marg. likelihood
2
4
6
8
10
polynomial degree

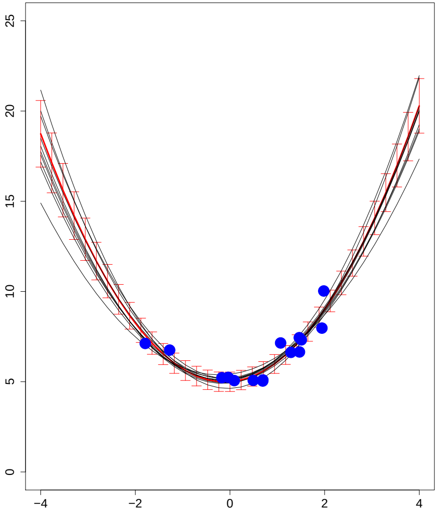

25
20
15
10
5
0
−4
−2
0
2
4
Log marg. likelihood
−14
−16
−18
−20
2
4
6
8
10
polynomial degree

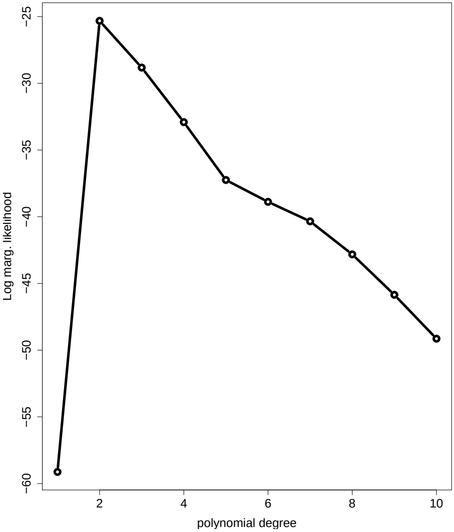


−4
−2
0
2
4
0
5
10
15
20
25



Log marg. likelihood
−18
−20
−22
−24
−26
−28
−30
2
4
6
8
10
polynomial degree


25
20
15
10
5
0
−4
−2
0
2
4


−25
−30
−35
−40
−45
−50
−55
−60
Log marg. likelihood
2
4
6
8
10
polynomial degree

Approximating the marginal likelihood

- Problem: In most cases we cannot compute the marginal likelihood in closed form $\rightsquigarrow$ approximations are needed.
- A specific approximation will lead to the **Bayesian Information Criterion (BIC)**.
- Key insight: when computing

  $$p(\boldsymbol{y}|\mathcal{F}, X) = \int L(\boldsymbol{w}; \mathcal{D}) p(\boldsymbol{w}|\mathcal{F}) d\boldsymbol{w},$$

  the integrand is a product of two densities $\rightsquigarrow$ integrand itself is an unnormalized density.
- Laplace's approximation uses a clever trick to approximate such integrals...

Approximation details: Laplace's Method

- Assume unnormalized density $p^*(\theta)$ has peak at $\hat{\theta}$. Goal: calculate normalizing constant

$$Z_p = \int p^*(\theta) d\theta$$

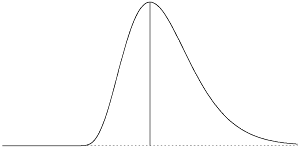
- Taylor-expand logarithm around $\hat{\theta}$:

$$\ln p^*(\theta) \approx \ln p^*(\hat{\theta}) - \frac{c}{2}(\theta - \hat{\theta})^2 + \cdots,$$

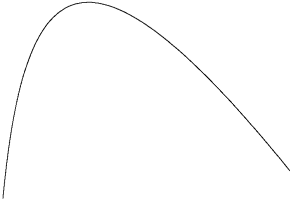
where

$$c := -\frac{\partial^2}{\partial\theta^2} \ln p^*(\theta)\big|_{\theta=\hat{\theta}}.$$

(note that first order term vanishes)



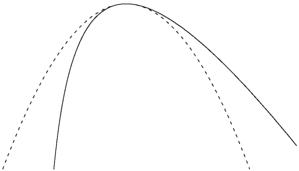
$p^*(\theta)$



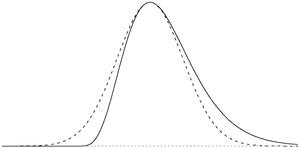
$\ln p^*(\theta)$

Laplace's Method (cont'd)

- Approximate $p^*(\theta)$ by unnormalized Gaussian
$$Q^*(\theta) := p^*(\hat{\theta}) \exp\left[-c/2 \cdot (\theta - \hat{\theta})^2\right]$$
- A normalized Gaussian would be:
$$Q(\theta \mid \mu = \hat{\theta}, \sigma^2) = \frac{1}{Z_Q} \exp\left[-\frac{(\theta - \hat{\theta})^2}{2\sigma^2}\right],$$
with $Z_Q = \sqrt{2\pi\sigma^2}$
$= \int \exp\left[-1/(2\sigma^2) \cdot (\theta - \hat{\theta})^2\right] d\theta$
- Approximate $Z_p = \int p^*(\theta)\, d\theta$ by
$$\begin{aligned} Z_p &\approx \int Q^*(\theta)\, d\theta \\ &= p^*(\hat{\theta}) \int \exp\left[-c/2 \cdot (\theta - \hat{\theta})^2\right]\, d\theta \\ &= p^*(\hat{\theta}) \sqrt{2\pi/c} \end{aligned}$$
$\rightsquigarrow$ c is the inverse variance



$\ln p^*(\theta)$ & $\ln Q^*(\theta)$



$p^*(\theta)$ & $Q^*(\theta)$

Laplace's Method (cont'd)

- Multivariate generalization in d dimensions: second derivative ⇝ **Hessian matrix**

$$H_{ij} = \left. \frac{\partial^2 \ln p^*(\boldsymbol{\theta})}{\partial \theta_i \partial \theta_j} \right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}$$

$$Z_p \approx p^*(\hat{\boldsymbol{\theta}}) \int \exp\left[-\frac{1}{2}(\boldsymbol{\theta}-\hat{\boldsymbol{\theta}})^t H (\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}) \right] d\boldsymbol{\theta}$$

$$= p^*(\hat{\boldsymbol{\theta}}) \sqrt{\frac{(2\pi)^d}{|H|}} = p^*(\hat{\boldsymbol{\theta}}) \left| \frac{H}{2\pi} \right|^{-\frac{1}{2}},$$

  where the last equation follows from the properties of the determinant: $|aM| = a^d |M|$ for $M \in \mathbb{R}^{d \times d}$, $a \in \mathbb{R}$.

- Interpretation:

  $p(\boldsymbol{\theta})$ **is approximated by a Gaussian centered at the mode** $\hat{\boldsymbol{\theta}}$**:**

$$p(\boldsymbol{\theta}) \approx \mathcal{N}(\boldsymbol{\theta} | \boldsymbol{\mu} = \hat{\boldsymbol{\theta}}, \Sigma = H^{-1}).$$

Bayesian Information Criterion (BIC)

$$\begin{aligned}
p(\mathcal{D}|\mathcal{F}) &= \int p(\mathcal{D}|\boldsymbol{w}) \cdot p(\boldsymbol{w}|\mathcal{F}) d\boldsymbol{w} \\
&\approx p(\mathcal{D}|\boldsymbol{w}^*) \cdot p(\boldsymbol{w}^*|\mathcal{F}) |H/(2\pi)|^{-\frac{1}{2}} \overset{\text{flat prior}}{\approx} p(\mathcal{D}|\hat{\boldsymbol{w}}) |H/(2\pi)|^{-\frac{1}{2}} \\
\log p(\mathcal{D}|\mathcal{F}) &\approx \log p(\mathcal{D}|\hat{\boldsymbol{w}}) - \frac{1}{2} \log |H| + C, \quad \text{with} \quad \hat{\boldsymbol{w}} = \boldsymbol{w}_{MLE} \text{ in } \mathcal{F}.
\end{aligned}$$

- Focus on last term:

  $$H = \sum_{i=1}^{n} H_i, \quad \text{with} \quad H_i = \nabla_{\boldsymbol{w}} \nabla_{\boldsymbol{w}} \log p(\mathcal{D}_i|\boldsymbol{w}).$$

  Let's approximate each H_i with a **fixed** matrix H'

  $$\log |H| = \log |nH'| = \log(n^d |H'|) = d \log n + \log(|H'|).$$

- For **model family selection**, **last term is irrelevant constant**, because it is **independent of $\mathcal{F}$ and n:**

  $$\log p(\mathcal{D}|\mathcal{F}) + C' \approx \log p(\mathcal{D}|\hat{\boldsymbol{w}}) - \tfrac{d}{2} \log n =: \text{ BIC}(\mathcal{F}, n|\mathcal{D}).$$

Intuitive interpretation of BIC

- The **Shannon information content** of a specific outcome a of a random experiment is

$$h(a) = -\log_2 P(a) = \log \frac{1}{P(a)}.$$

It measures the "surprise" (in bits):
Outcomes that are less probable have larger values of surprise.

- **Information theory:** Can find a code so that the **number of bits** used to encode each symbol $a \in \mathcal{A}$ is essentially $-\log_2 P(a)$**.**
- Here:

$$-\mathrm{BIC}(\mathcal{F}, n|\mathcal{D}) = \overbrace{\sum_{i=1}^{n} \left(\underbrace{-\log_2 p(y_i|\boldsymbol{x}_i, \hat{\boldsymbol{w}})}_{\text{surprise of } y_i} \right)}^{\text{DL of observations given model}} + \frac{d}{2}\log_2(n)$$

- The sum of surprises of all observations is the **description length of the observations given the (most probable) model in $\mathcal{F}$.**

Intuitive interpretation of BIC

Second term: **DL of the model.** Intuitive explanation:

- The model, i.e. $\hat{\boldsymbol{w}} \in \mathbb{R}^d$, was estimated based on n samples.
- Can quantize every component into $\sqrt{n}$ levels. Why?
- Recall the **standard parametric rate:**

$$\sqrt{n}(\hat{\boldsymbol{w}}_n - \boldsymbol{w}) \sim N(\mathbf{0}, \sigma^2 \underbrace{(X^t X/n)}_{\overset{n\to\infty}{\to} \Sigma}{}^{-1})$$

$\rightsquigarrow$ $\hat{\boldsymbol{w}}_n$ converges to true $\boldsymbol{w}$ at a rate of $1/\sqrt{n}$

$\rightsquigarrow$ $1/\sqrt{n}$ **represents the magnitude of the estimation error**

$\rightsquigarrow$ **no need for encoding with greater precision:**

- Assume $w \in \mathbb{R}$, and range of w rescaled to unit interval $[0, 1]$.
- Instead of communicating exact numerical value of $\hat{w}_n$ over the communication channel, we can partition the unit interval into $\sqrt{n}$ bins and **communicate only the number of the bin.**

Intuitive interpretation of BIC

- In $\mathbb{R}^d$: Grid of $(\sqrt{n})^d$ possible values for describing the model.
- We only need $\log_2((\sqrt{n})^d) = \log_2 n^{(d/2)} = (d/2) \log_2 n$ **bits to encode $\hat{\boldsymbol{w}}$ with sufficient precision.**
- In summary:

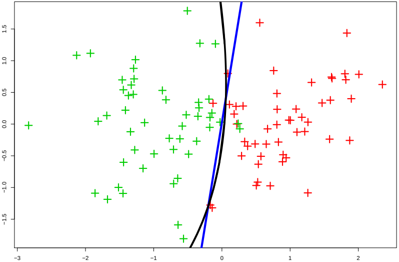
$$\begin{aligned} -\mathrm{BIC} = & \quad -\log_2 p(\mathcal{D}|\hat{\boldsymbol{w}}) & + \frac{d}{2} \log_2 n \\ = & \quad \mathrm{DL}(\mathrm{data}|\mathrm{model}) & + \mathrm{DL}(\mathrm{model}). \end{aligned}$$

- Maximizing BIC = minimizing the joint DL of data and model $\rightsquigarrow$ **Minimum Description Length principle.**

Example: Bayesian logistic regression

Example: polynomial logistic regression, $n = 100$.
$\phi_1(\boldsymbol{x}) = (1, x_1, x_2)^t$, $\phi_2(\boldsymbol{x}) = (1, x_1, x_2, (x_1 + x_2)^2)^t$.

$$-\mathrm{BIC} = \sum_{i=1}^{n} (-\log_2 p(y_i | \boldsymbol{x}_i, \hat{\boldsymbol{w}})) + \frac{d}{2} \log_2(n)$$

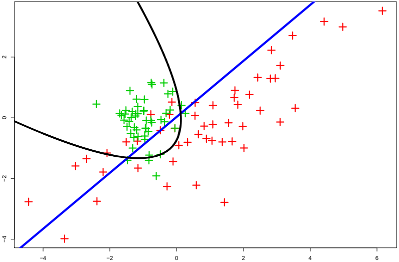


degree	#(param)	DL(data|model)	DL(model)	BIC score
1	3	16.36 bits	9.97 bits	**-26.33**
2	4	15.77 bits	13.29 bits	-29.06

Example: Bayesian logistic regression

Example: polynomial logistic regression, $n = 100$.
$\phi_1(\boldsymbol{x}) = (1, x_1, x_2)^t$, $\phi_2(\boldsymbol{x}) = (1, x_1, x_2, (x_1 + x_2)^2)^t$.

$$-\mathrm{BIC} = \sum_{i=1}^{n} (-\log_2 p(y_i|\boldsymbol{x}_i, \hat{\boldsymbol{w}})) + \frac{d}{2}\log_2(n)$$



degree	#(param)	DL(data|model)	DL(model)	BIC score
1	3	58.56 bits	9.97 bits	-68.53
2	4	38.05 bits	13.29 bits	**-51.34**

Sparse Models

- Sometimes, we have many more dimensions d than training cases n.
- Corresponding design matrix X is "short and fat", rather than "tall and skinny".
- This is called **small n , large d problem**.
- For example, with **gene microarrays**, it is common to measure the expression levels of $d \approx 20,000$ genes, but to only get $n \approx 100$ samples (for instance, from 100 patients).
- Q: what is the **smallest set of features** that can accurately predict the response in order to **prevent overfitting**, to **reduce the cost** of building a diagnostic device, or to help with **scientific insight** into the problem?

Bayesian variable selection

- Let $\gamma_j = 1$ if feature j is **relevant**, and let $\gamma_j = 0$ otherwise.
- Our goal is to compute the posterior over models

$$p(\boldsymbol{\gamma}|\mathcal{D}) \propto p(\mathcal{D}|\boldsymbol{\gamma})p(\boldsymbol{\gamma})$$

- Example: generate $n = 20$ samples from a $d = 10$ dimensional linear model, $y_i \sim N(w^t x_i, \sigma^2)$, in which $K = 5$ elements of w are non-zero.
- Enumerate all $2^{10} = 1024$ models. Note that a model is expressed as a specific **sparsity pattern** via a bit string, such as

$$(0, 1, 1, 0, 0, 1, 0, 1, 1, 0).$$

  Then, compute $p(\boldsymbol{\gamma}|\mathcal{D})$ for each one.
- Interpreting the posterior over a large number of models is difficult $\rightsquigarrow$ seek **summary statistics**.
- Natural choice: **MAP estimate:** $\hat{\boldsymbol{\gamma}} = \arg\max_{\boldsymbol{\gamma}} p(\boldsymbol{\gamma}|\mathcal{D})$.

Bayesian variable selection

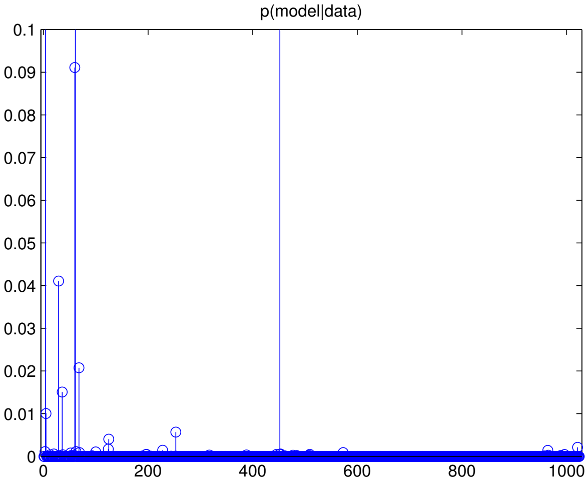


Fig 13.1 in K. Murphy: Machine Learning. MIT Press 2012. Posterior over all 1024 models. Vertical scale has been truncated at 0.1 for clarity.

Bayesian variable selection

- The above example illustrates the **gold standard** for variable selection: the problem was small ($d = 10$)
  $\rightsquigarrow$ we were able to **compute the full posterior exactly**.
- Of course, variable selection is most useful in the cases where the number of dimensions is **large**.
- There are 2^d possible models (bit vectors)
  $\rightsquigarrow$ **impossible** to compute the full posterior in general.
- Even finding summaries is **intractable**
  $\rightsquigarrow$ **algorithmic speedups** necessary.
- But first, focus on the computation of the posterior $p(\gamma|\mathcal{D})$.

The spike and slab model

- The posterior is given by

$$p(\boldsymbol{\gamma}|\mathcal{D}) \propto p(\boldsymbol{\gamma})p(\mathcal{D}|\boldsymbol{\gamma})$$

- It is common to use the following prior:

$$\begin{aligned} p(\boldsymbol{\gamma}) &= \prod_{j=1}^{d} Ber(\gamma_j|\pi_0) = \pi_0^{\|\gamma\|_0}(1-\pi_0)^{d-\|\gamma\|_0}, \\ \log p(\boldsymbol{\gamma}|\pi_0) &= -\lambda\|\boldsymbol{\gamma}\|_0 + const., \end{aligned}$$

  where π_0 is the probability that a feature is relevant, and $\|\boldsymbol{\gamma}\|_0 = \sum_{j=1}^{d} \gamma_j$ **is the ℓ_0 pseudo-norm**, i.e., the **number of non-zero elements**.

- $\lambda = \log \frac{1-\pi_0}{\pi_0}$ controls the **sparsity** of the model.
- Setting $\sigma^2 = 1$, we can write the (marginal) likelihood as follows:

$$p(\mathcal{D}|\gamma) = p(\boldsymbol{y}|X,\gamma) = \int p(\boldsymbol{y}|X,\boldsymbol{w},\gamma)p(\boldsymbol{w}|\gamma)\,d\boldsymbol{w}$$

The spike and slab model

- Focus on prior $p(\boldsymbol{w}|\boldsymbol{\gamma})$. If $\gamma_j = 0$, feature j is **irrelevant**, so we expect $w_j = 0$. If $\gamma_j = 1$, we expect w_j to be non-zero.
- Assume a **Gaussian prior**, $N(0, \sigma_w^2)$, where σ_w^2 reflects our expectation of the coefficients associated with the **relevant variables**:

$$p(w_j|\gamma_j) = \begin{cases} \delta_0(w_j) & \text{, if } \gamma_j = 0 \\ N(w_j|0, \sigma_w^2) & \text{, else} \end{cases}$$

- The first term is a **spike at the origin.**
- As $\sigma_w^2 \to \infty$, the distribution $p(w_j|\gamma_j = 1)$ approaches a **uniform distribution** $\rightsquigarrow$ second term is **slab of constant height.**
- **Spike and slab model** (Mitchell and Beauchamp 1988).
- Full Bayesian treatment is computationally challenging!

Simplifying the model

- Assume $\sigma_w^2 \to \infty$ ($\rightsquigarrow$ uniform prior $p(w_j|\gamma_j)$ over nonzero components) and approximate the likelihood using **BIC**:

$$\begin{aligned}\log p(\mathcal{D}|\gamma) &= \int p(\boldsymbol{y}|X, \boldsymbol{w}, \gamma) p(\boldsymbol{w}|\gamma)\, d\boldsymbol{w} \\ &\approx \log p(\boldsymbol{y}|X, \hat{\boldsymbol{w}}_\gamma) - \frac{1}{2} \underbrace{\|\hat{\boldsymbol{w}}_\gamma\|_0}_{\text{"effective" dimension}} \log n,\end{aligned}$$

  where $\hat{\boldsymbol{w}}_\gamma$ is the ML estimate.

- Another view of this model: select $\hat{\boldsymbol{w}}$ by minimizing the negative log likelihood under a ℓ_0 penalty:

$$\text{minimize } -\log p(\boldsymbol{y}|X, \boldsymbol{w}) + \lambda \|\boldsymbol{w}\|_0.$$

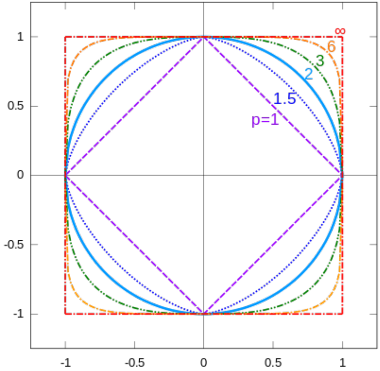
- Practical problem: ℓ_0 pseudo-norm is highly non-convex!

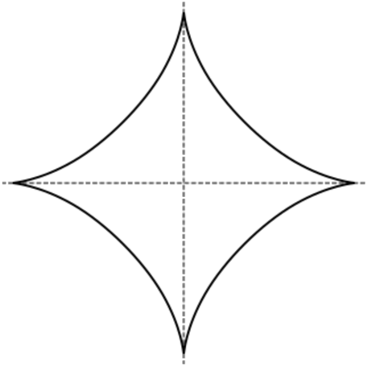
Vector norms

The **vector p-norms (ℓ_p norms)** are defined by

$$\|\boldsymbol{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}, \quad 1 \leq p \leq \infty,$$

$$\|\boldsymbol{x}\|_\infty = \max(|x_1|, \cdots |x_n|).$$





Quartl, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=17428655

Simplifying the model further

- ℓ_0 penalty $\rightsquigarrow$ **combinatorial optimization problem**
- When we have many variables, it is computationally difficult to find the the minimizer of $-\log p(\boldsymbol{y}|X, \boldsymbol{w}) + \lambda\|\boldsymbol{w}\|_0$.
- Idea: replace discrete variables with continuous ones. Use continuous priors that "encourage" $w_j = 0$ by putting a lot of probability density near the origin, such as a zero-mean Laplace distribution.

$$p(\boldsymbol{w}|\lambda) = \prod_{j=1}^{d} Lap(w_j|0, 1/\lambda) \propto \prod_{j=1}^{d} \exp(-\lambda|w_j|)$$

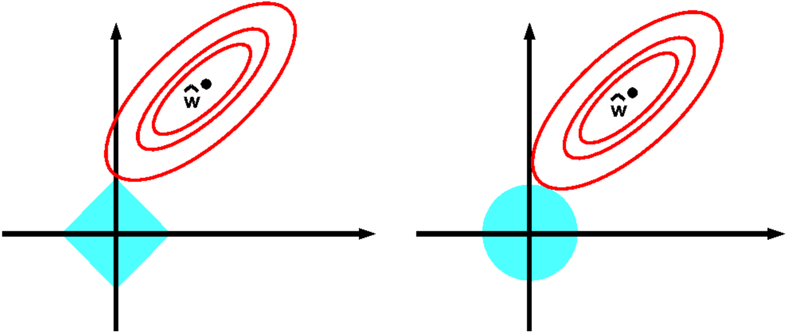
- Let us perform MAP estimation with this prior:

$$f(\boldsymbol{w}) = -\log p(\mathcal{D}|\boldsymbol{w}) - \log p(\boldsymbol{w}|\lambda) = NLL(\boldsymbol{w}) + \lambda\|\boldsymbol{w}\|_1.$$

where $\|\boldsymbol{w}\|_1 = \sum_{j=1}^{d} |w_j|$ is the ℓ_1 norm of $\boldsymbol{w}$ and NNL means **negative log-likelihood**.

The Lasso

- Can be thought of as a **convex approximation** to the ℓ_0 norm.
- For suitably large λ, the estimate $\hat{\boldsymbol{w}}$ will still be **sparse.**
- This model has the colorful name **least absolute shrinkage and selection operator**.



The Lasso

- Unfortunately, the $\|\boldsymbol{w}\|_1$ term is not differentiable at 0
  $\rightsquigarrow$ **convex, but non-smooth optimization problem.**
- The **subderivative or subgradient** of a (convex) function $f : \mathcal{I} \to \mathbb{R}$ at a point x_0 is a scalar c such that
  $$f(x) - f(x_0) \geq c(x - x_0),\ \forall x \in \mathcal{I}$$
  where $\mathcal{I}$ is some interval containing x_0.
  Note that c is a **linear lower bound** to f at x_0.

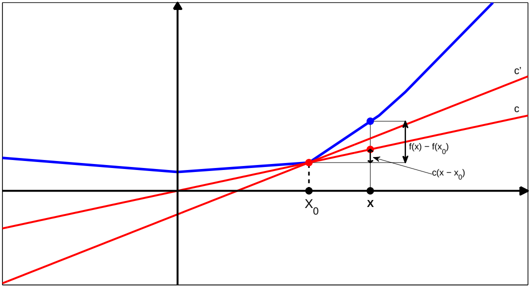


Fig. 13.4 in K. Murphy: Machine Learning. MIT Press 2012

The Lasso

- The set of all subderivatives is called the **subdifferential**
- For the **absolute value function** $f(x) = |x|$:

$$\partial f(x) = \begin{cases} -1 & \text{, if } x < 0 \\ [-1, 1] & \text{, if } x = 0 \\ +1 & \text{, if } x > 0 \end{cases}$$

- For least-squares regression, it is easy to show that

$$\begin{aligned} \frac{\partial}{\partial w_j} RSS(\boldsymbol{w}) &= a_j w_j - c_j \\ a_j &= 2 \sum_{i=1}^{n} x_{ij}^2 \\ c_j &= 2 \sum_{i=1}^{n} x_{ij}(y_i - \boldsymbol{w}_{-j}^t \boldsymbol{x}_{i,-j}). \end{aligned}$$

where $\boldsymbol{w}_{-j}$ is $\boldsymbol{w}$ without component j.

The Lasso

- c_j is (proportional to) the correlation between the j'th feature $\boldsymbol{x}_{\cdot j} = (x_{1j}, x_{2j}, \dots, x_{nj})^t$ and the **residual due to other features:**

$$c_j \propto \boldsymbol{x}_{\cdot j}^t \boldsymbol{r}_{(-j)} \text{ , with } \boldsymbol{r}_{(-j)} = \boldsymbol{y} - \underbrace{X_{-j}}_{X \text{ w/o } j\text{-th col}} \boldsymbol{w}_{-j}.$$

- Recall that the **residual from the least squares estimate** is **orthogonal to every input feature.**
  $\rightsquigarrow$ **magnitude of c_j indicates how relevant feature j is, relative to all other features.**

- Adding the ℓ_1 penalty term:

$$\begin{aligned} \partial_{w_j} f(\boldsymbol{w}) &= (a_j w_j - c_j) + \lambda \partial_{w_j} \|\boldsymbol{w}\|_1 \\ &= \begin{cases} a_j w_j - c_j - \lambda & \text{, if } w_j < 0 \\ [-c_j - \lambda, -c_j + \lambda] & \text{, if } w_j = 0 \\ a_j w_j - c_j + \lambda & \text{, if } w_j > 0 \end{cases} \end{aligned}$$

The Lasso

- Depending on the value of c_j, the solution to $\partial_{w_j} f(\boldsymbol{w}) = 0$ can occur at three different values of w_j:

$$\hat{w}_j = \begin{cases} (c_j + \lambda)/a_j & \text{, if } c_j < -\lambda \\ 0 & \text{, if } c_j \in [-\lambda, \lambda] \\ (c_j - \lambda)/a_j & \text{, if } c_j > \lambda \end{cases}$$

- We can write this as follows:

$$\hat{w}_j = \text{soft}\left(\frac{c_j}{a_j}; \frac{\lambda}{a_j}\right),$$

  where $\text{soft}(a; \delta) = \text{sign}(a)(|a| - \delta)_+$

  and $x_+ = \max(x, 0)$ is the positive part of x.

- This is called **soft thresholding.**

The Lasso

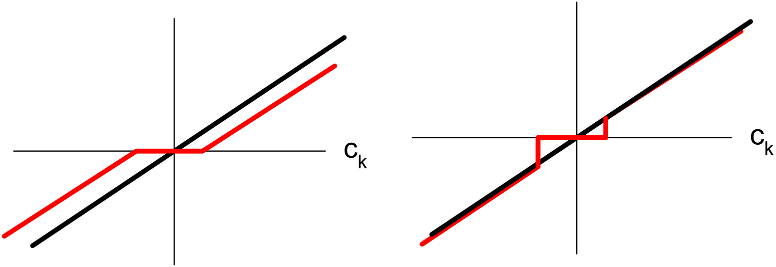


Fig. 13.5 in K. Murphy: Machine Learning. MIT Press 2012.
Black line: **Least squares fit** $w_k = c_k/a_k$.
The red line (the regularized estimate) $\hat{w}_k(c_k)$, shifts the black line down (or up) by λ, except when $-\lambda \leq c_k \leq \lambda$, in which case it sets $w_k = 0$.
By contrast, **hard thresholding** sets values of w_k to 0 if $-\lambda \leq c_k \leq \lambda$, but it **does not shrink the values of w_k outside of this interval.**

Lasso Algorithms: Coordinate-wise Descent

Sometimes it is hard to optimize all variables simultaneously, but it is easy to **optimize them one by one.** Assume that we can efficiently solve for the j-th coefficient w_j with all other coefficients held fixed:
$\hat{w}_j = \arg\min_z f(\boldsymbol{w} + z\boldsymbol{e}_j)$, where $\boldsymbol{e}_j$ is the j-th unit vector.
Then cycle through these component-wise updates.
For the Lasso, this is particularly simple:
for $j = 1, \ldots, d$ **do:**

$$\begin{aligned} a_j &= 2\sum_{i=1}^{n} x_{ij}^2 \\ c_j &= 2\sum_{i=1}^{n} x_{ij}(y_i - \boldsymbol{w}_{-j}^t \boldsymbol{x}_{i,-j}) \\ \hat{w}_j &= \text{soft}\left(\frac{c_j}{a_j}; \frac{\lambda}{a_j}\right). \end{aligned}$$