Pattern recognition and machine learning: mathematical basis

Rokhlin D.B.

IMMCS, SFU, 2021

Links:

- Andreas C. Müller (Columbia University) Applied Machine Learning
- Bernd Bischl (Ludwig-Maximilians-Universität München) et. al. Introduction to Machine Learning
- Kostia Zuev (California Institute of Technology) Fundamentals of Statistical Learning
- Kilian Weinberger (Cornell University) Machine Learning for Intelligent Systems

Books

- Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning 2001, 2009.
- Muller A.C., Guido S. Introduction to machine learning with Python, 2016. link to code on github
- Johansson R. Numerical Python, 2019. link to code on github
- Geron A. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow, 2019. link to code on github

Supervised learning: informal problem formulation

We are interested in algorithms that can learn from data.

- Examples (dataset): $S = (x_i, y_i)_{i=1}^n$:
- $x_i \in \mathcal{X} \subset \mathbb{R}^d$: instances, features, inputs,
- $y_i \in \mathcal{Y} \subset \mathbb{R}$: labels, responses.

Given i.i.d. examples:

$$z_i = (x_i, y_i) \sim \mathcal{P}, \quad (x_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$$

we want to find $\hat{f} : \hat{f}(x_i) \approx y_i$ with good generalization properties: $\hat{f}(x) \approx y$ for unseen examples (x, y).

Two types of the supervised learning:

- Regression: $\mathcal{Y} = \mathbb{R}$; Y is quantitative (continuous)
- Classification: $\mathcal{Y}$ is finite; Y is qualitative (discrete)

There is a natural metrics on $\mathbb{R}$, but there is no natural distance between classes.

Examples:

- S is a set of houses; features: the number of rooms, distance to the center, crime rate, ...; labels: house prices;
- S is a set of e-mails; features: words in the email; labels: spam/non-spam;
- S is a set of images of cats and dogs; features: $k \times m$ matrices of integer pixel intensities from 0 (white) to 256 (black); labels: cat/dog.
- recommender systems, automatic stock trading, automatic translation, cancer (or Covid-19) detection, data driven discoveries in science

Hypothesis class

Given the data S, we want to construct a mapping $\widehat{f} : \mathcal{X} \mapsto \mathcal{Y}$ from some hypothesis class $\mathcal{H}$. This is performed by some algorithm $\mathcal{A}$: $\widehat{f} = \mathcal{A}(S)$. This process is called *training* or *fitting*.
Hypothesis class can be given in parametric form: $\mathcal{H} = \{f_\theta : \theta \in \Theta\}$. In this case we need to select θ: $\widehat{\theta} = \mathcal{A}(S)$.

Examples:

- $\mathcal{H} = \{x \mapsto \langle w, x\rangle + b : w \in \mathbb{R}^d, b \in \mathbb{R}\}$
- $\mathcal{H} = \{x \mapsto w_0 + w_1 x_1 + w_2 x_2 + \sum_{i,j=1}^{2} w_{i,j} x_i x_j : w_0, w_i, w_{ij} \in \mathbb{R}\}$
- $\mathcal{H} = \{x \mapsto \sum_{i=1}^{k} c_i I_{[a_i, b_i]} : k \in \mathbb{N}, a_i < b_i\}$: step functions

Loss function

The quality of a predictor is evaluated by a loss function

$$L : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}_+.$$

- Classification problem $\mathcal{Y} = \{1, \dots, k\}$ ($k > 2$: multiclass classification). Zero-one loss:

$$L(y, y') = I_{\{y \neq y'\}} = \begin{cases} 1, & y' \neq y, \\ 0, & y' = y; \end{cases}$$

- Regression problem: $\mathcal{Y} = \mathbb{R}$. Quadratic loss:

$$L(y, y') = (y - y')^2,$$

absolute loss:

$$L(y, y') = |y - y'|.$$

True risk and empirical risk

(Expected) true risk:

$$\mathcal{R}(\widehat{f}) = \mathsf{E} L(Y, Y') = \mathsf{E} L(Y, \widehat{f}(X)).$$

Empirical risk:

$$\mathcal{R}_S(\widehat{f}) = \frac{1}{n} \sum_{i=1}^{n} L(Y_i, \widehat{f}(X_i)).$$

True risk is unknown we can compute and optimize the empirical risk and estimate the true risk. In most cases fitting requires optimization:

$$\mathcal{R}_S(f) \to \min_{f \in \mathcal{H}}.$$

Model selection

Why the model selection is needed?

- A small (simple) model can be unable to give a good representation of data (only a raw representation): *underfitting*.
- A large (complex) model can tend to give an arbitrary explanation of data (memorize data), it may not ensure good predictions: *overfitting*.

The model (= hypothesis class) complexity should be adequate to the available data.

- (Underfitting) Assume that the data comes from a model

$$y_i = f(x_i) + \text{“noize”}, \quad f(x) = ax^2 + bx + c.$$

  Consider the regression problem with the square loss and linear predictors $h(x) = wx + w_0$. Typically the error will be high for a given “training data“ $S = (z_1, \ldots, z_m)$ for any parameters (w, w_0).

- (Overfitting) $\mathcal{X} = [0, 1]$, $\mathcal{Y} = \{0, 1\}$, $\mathcal{H}$: the set of all functions from $\mathcal{X}$ to $\mathcal{Y}$. Consider the classification problem with 0-1 loss. For any given examples $S = (z_1, \ldots, z_m)$ the function

$$h(x) = \begin{cases} 1, & x = x_i, y = 1 \\ 0, & otherwise \end{cases}$$

  have zero “training loss”. However, it simply memorizes data and for a new example the result is not related to data at all.

Hold-out validation

Previous example shows that the quality of the predictor cannot be correctly estimated on the training set S.
Split S into training and validation (testing) parts:

$$\{1, \ldots, n\} = T \cup V, \quad T \cap V = \emptyset.$$

Fit the model on T and get $\hat{f}$. Compute the average loss on the validation part:

$$\mathcal{R}_V(\widehat{f}) = \frac{1}{|V|} \sum_{i \in V} L(Y_i, \widehat{f}(X_i)).$$

This is an unbiased estimate of the true risk $\mathcal{R}(\widehat{f})$, since $\widehat{f}$ depends only on $(X_i, Y_i)_{i \in T}$, which are independent of $(X_i, Y_i)_{i \in V}$.

Cross-validation

Split S into K subsets $S_1, \ldots, S_K$ at random, and denote by $f^{(-k)}$ the model f fitted to $S \backslash S_k$. The cross-validation estimate of the test error of $\widehat{f}$:

$$\mathrm{CV}(\widehat{f}) = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{|S_k|} \sum_{i \in S_k} L(y_i, \widehat{f}^{(-k)}(x_i)).$$

For $K = n$ we get the leave-one-out cross-validation (LOOCV). In this case $f^{(-i)} \approx \widehat{f}$,

$$\mathrm{CV}(\widehat{f}) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, \widehat{f}^{(-i)}(x_i)).$$

For large K all models $\widehat{f}^{(-i)}$ are close to each other and to $\widehat{f}$. For small K they are different. Optimal K depend on the data set. $K = 5$ or $K = 10$ are recommended.

Cross-validation for model selection

For a family of models $f_\theta = f(x;\theta)$ indexed by a hyperparameter (tuning parameter) θ find $\widehat{\theta}$ with the best CV estimate of the true loss:

$$\widehat{\theta} \in \arg\min_{\theta} \mathrm{CV}(\widehat{f_\theta}).$$

The result will depend on random partition except the case of LOOCV. For small K $\widehat{\theta}$ can significantly depend on a random partition: see the examples from James, Witten, Hastie, Tibshirani "An Introduction to Statistical Learning with Applications in R" (2014).

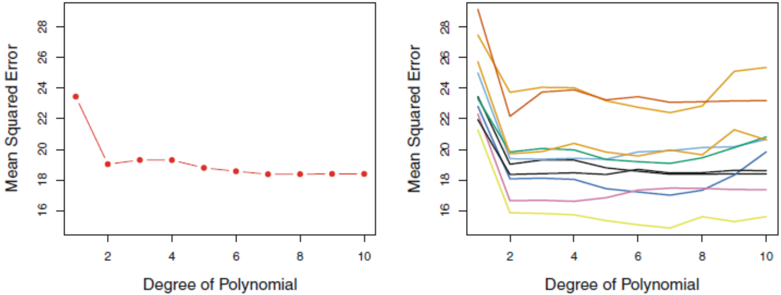


FIGURE 5.2. *The validation set approach was used on the* `Auto` *data set in order to estimate the test error that results from predicting* `mpg` *using polynomial functions of* `horsepower`. Left: *Validation error estimates for a single split into training and validation data sets.* Right: *The validation method was repeated ten times, each time using a different random split of the observations into a training set and a validation set. This illustrates the variability in the estimated test MSE that results from this approach.*

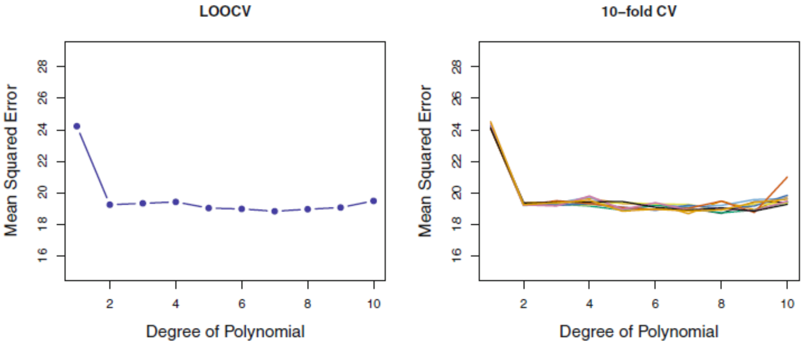


FIGURE 5.4. *Cross-validation was used on the* `Auto` *data set in order to estimate the test error that results from predicting* `mpg` *using polynomial functions of* `horsepower`. Left: *The LOOCV error curve.* Right: 10*-fold CV was run nine separate times, each with a different random split of the data into ten parts. The figure shows the nine slightly different CV error curves.*

Bayes optimal classifier

$$\mathsf{E}L(Y,\widehat{Y}) = \mathsf{E}\mathsf{E}\left(L(Y,f(X)|X\right) = \int_{\mathcal{X}} \mathsf{E}\left(L(Y,f(X)|X=x\right)\mathsf{P}_X(dx)$$

$$\mathsf{E}\left(L(Y,f(X)|X=x\right) = \int_{\mathcal{Y}} L(y,f(x))\mathsf{P}_{Y|X}(dy|x)$$

The problem

$$\mathsf{E}L(Y,\widehat{Y}) \to \min_f$$

reduces to

$$\int_{\mathcal{Y}} L(y,c)\mathsf{P}_{Y|X}(dy|x) \to \min_c .$$

An optimal solution $f^*(x) = c^*(x)$ is called the Bayes (optimal) classifier. The correspondent loss $\mathsf{E}L(Y,f^*(X))$ is called the Bayes risk.

Regression with squared loss

$$\begin{aligned}\int L(y,c)\mathsf{P}_{Y|X}(dy|x) &= \int (y-c)^2\mathsf{P}_{Y|X}(dy|x)\\ &= \int y^2\mathsf{P}_{Y|X}(dy|x) - 2c\int y\mathsf{P}_{Y|X}(dy|x) + c^2 \to \min_c .\end{aligned}$$

f^* is the conditional expectation:

$$f^*(x) = \int y\mathsf{P}_{Y|X}(dy|x) = \mathsf{E}(Y|X=x).$$

Regression with absolute loss

Assume that $P_{Y|X}(dy|x)$ is a continuous distribution with the density $p_{Y|X}(y|x)$. Then

$$\begin{aligned}\int L(y,c)\mathsf{P}_{Y|X}(dy|x) &= \int |y-c|\mathsf{P}_{Y|X}(dy|x)\\ &= \int_{-\infty}^{c}(c-y)p_{Y|X}(y|x)dy + \int_{c}^{\infty}(y-c)p_{Y|X}(y|x)dy.\end{aligned}$$

Take the derivative w.r.t. c:

$$\begin{aligned}0 &= \int_{-\infty}^{c} p_{Y|X}(y|x)dy - \int_{c}^{\infty} p_{Y|X}(y|x)dy\\ &= \mathsf{P}(Y\le c|X=x) - \mathsf{P}(Y>c|X=x) = 2\mathsf{P}(Y\le c|X=x)-1.\end{aligned}$$

$f^*(x)$ is the conditional median of Y:

$$\mathsf{P}(Y\le f^*(x)|X=x) = 1/2.$$

Classification with zero-one loss

Let $\mathcal{Y}$ be finite.

$$\int_{\mathcal{Y}} L(y,c)\mathsf{P}_{Y|X}(dy|x) = \int_{\mathcal{Y}} I_{\{y \neq c\}}\mathsf{P}_{Y|X}(dy|x) = 1 - P_{Y|X}(c|x)$$

$f^*(x)$ is the conditionally most probable label:

$$f^*(x) = \arg\max_{y \in \mathcal{Y}} \mathsf{P}_{Y|X}(y|x) = \arg\max_{y \in \mathcal{Y}} \mathsf{P}(Y = y|X = x).$$

Nearest neighbor model for regression

Estimate $\mathsf{E}(Y|X=x)$ directly from training data:

$$\mathsf{E}(Y|X=x) \approx \frac{1}{|J(x)|} \sum_{i \in J(x)} y_i, \quad J(x) = \{i : x_i = x\}.$$

However, $J(x)$ can be empty. Let us relax the condition $x_i = x$ to

$$x_i \in \mathcal{N}_k(x) = \text{ the set of } k \text{ points } x_i \text{ closest to } x.$$

Approximation of the regression function:

$$f^*(x) = \mathsf{E}(Y|X=x) \approx \widehat{f}(x) = \frac{1}{k} \sum_{i : x_i \in \mathcal{N}_k} y_i.$$

As $n \to \infty$ the points in $\mathcal{N}_k$ become closer to x, as $k \to \infty$ the average becomes closer to conditional expectations. It is known that $\widehat{f}(x) \to f^*(x)$ as $k, n \to \infty$, $k/n \to 0$. However, this convergence can be very slow.

Nearest neighbor model for classification

- Estimate $\mathsf{P}(Y = y|X = x)$ directly from data
- Classify x by selecting a class with the largest estimated probability

To estimate $\mathsf{P}(Y = y|X = x)$ again we relax the condition $X = x$ to $X \in \mathcal{N}_k(x)$:

$$\mathsf{P}(Y = y|X = x) \approx \frac{1}{k} \sum_{i:x_i \in \mathcal{N}_k(x)} I_{\{y_i = y\}}$$

This is the fraction of labels y among all labels of the neighbors.
Approximate Bayes classifier:

$$f^*(x) \approx \widehat{f}(x) = \arg\max_{y \in \mathcal{Y}} \sum_{i:x_i \in \mathcal{N}_k(x)} I_{\{y_i = y\}}.$$

It is equivalent to the majority vote in the neighborhood $\mathcal{Y}_k$.

Curse of dimensionality: first aspect

The approximation $\widehat{f}(x)$ uses only local information: the labels of points in $\mathcal{N}_k(x)$ which contains $\alpha = k/n \ll 1$ fraction of training data. Assume that $X_i \in U(C_d)$, $C_d = [0,1]^d$. What space occupy k nearest neighbors of 0?

To cover a fraction α of the sample by $C_d(l) = [0,l]^d$ on average we need l such that $\mathrm{Vol}(C_p(l)) = \alpha$:

$$l = \alpha^{1/d} \to 1, \quad d \to \infty.$$

For instance,

$$\alpha = 0.01,\ d = 20 \Longrightarrow l \approx 0.79,$$

$$\alpha = 0.01,\ d = 100 \Longrightarrow l \approx 0.95.$$

Thus, "nearest neighbors" can in fact be very far from $x = 0$ in high-dimensional case.

Curse of dimensionality: second aspect

- How many points are needed to cover the hyper-cube C_d in such a way that for any point x distance to a nearest neighbor is not greater than ε?

In one dimension if $x_i = i/n$, $i = 0, \dots, n$, then

$$\min_{0 \le i \le n} |x - x_i| \le \frac{1}{2n}.$$

In d dimensions:

$$\min \left\{ \|x - x'\| : x' \in \{0, 1/n, \dots, 1\}^d \right\} \le \sqrt{\sum_{i=1}^{d} \left(\frac{1}{2n} \right)^2} = \frac{1}{2n}\sqrt{d} \le \varepsilon,$$

Thus, $n \ge \frac{\sqrt{d}}{2\varepsilon}$ and the number of required points $N \ge \left(1 + \frac{\sqrt{d}}{2\varepsilon}\right)^d$ increases exponentially in d. For instance, $\varepsilon = 0.01$, $d = 25$, $N \ge 251^{25}$.

Curse of dimensionality: third aspect

The fraction of volume in the ''ε-edge'' of a d-dimensional ball is

$$1-(1-\varepsilon)^d.$$

If such an apple is ''peeled'', almost nothing will remain.

- If $x_1,\dots,x_n$ are sampled uniformly over a high dimensional ball $B_d=\{x\in\mathbb{R}^d : \|x\|\le 1\}$, then they are close to its boundary.

Let $X_i\sim U(B_d)$ and let D be the distance from the origin to its nearest neighbor:

$$D=\min\{\|X_1\|,\dots,\|X_n\|\}.$$

Let's find the cdf of D:

$$F_D(t)=1-\mathsf{P}(D>t)=1-\prod_{i=1}^{n}\mathsf{P}(\|X_i\|>t)$$

$$\begin{aligned} F_D(t) &= 1 - \prod_{i=1}^{n}(1 - \mathsf{P}(\|X_i\| \le t)) = 1 - (1 - \mathsf{P}(\|X\| \le t))^n \\ &= 1 - \left(1 - \frac{\mathrm{Vol}(B_d(t))}{\mathrm{Vol}(B_d)}\right)^n = 1 - \left(1 - t^d\right)^n \end{aligned}$$

Consider the median of D (“middle value”, “typical value”):

$$F_D(\widetilde{t}) = 1/2 \Longrightarrow \widetilde{t} = \left(1 - \left(\frac{1}{2}\right)^{1/n}\right)^{1/d} \to 1, \quad d \to \infty.$$

For instance,

$$d = 20, n = 1000 \Longrightarrow \widetilde{t} \approx 0.7,$$

$$d = 100, n = 10000 \Longrightarrow \widetilde{t} \approx 0.9.$$

For $x = 0$ the nearest neighbor will perform extrapolation from far away points to the origin. This will result in bad approximation.

Introduction to Machine Learning (I2ML)
https://introduction-to-machine-learning.netlify.app/
created by Bernd Bischl et. al.
https://introduction-to-machine-learning.netlify.app/team-and-li
Chapter 13.02: Curse of Dimensionality - Examples
https://introduction-to-machine-learning.netlify.app/chapter13-0

Notation

- $x_i = (x_{i0}, \dots, x_{id})^T$ is the i-th observed value X, $i = 1, \dots, n$.
- $x^{(j)} = (x_1^{(j)}, \dots, x_n^{(j)})^T = (x_{1j}, \dots, x_{nj})^T$ is the vector of all observation of the j-th component X_j of X,

$$\mathbb{X} = \begin{pmatrix} x_{1,0} & \cdots & x_{1,d} \\ \vdots & \vdots & \vdots \\ x_{n,0} & \cdots & x_{n,d} \end{pmatrix} = \begin{pmatrix} x_1^T \\ \vdots \\ x_n^T \end{pmatrix} = (x^{(0)}, \dots, x^{(d)}).$$

- $y = (y_1, \dots, y_n)^T$ is the vector of observed values of Y.

Linear regression

Instead of estimating the regression function $f(X)$ let's model it. Suppose that f is linear:

$$f(x) = x^T\beta, \quad x^T = (1, x_1, \ldots, x_d), \quad \beta^T = (\beta_0, \ldots, \beta_d).$$

Then

$$\mathrm{MSE}(f) = \mathsf{E}(Y - f(x))^2 = \mathsf{E}(Y - X^T\beta)^2 = \mathrm{MSE}(\beta),$$

$$\begin{aligned}\nabla_\beta \mathrm{MSE}(\beta) &= \mathsf{E}\nabla_\beta (Y - X^T\beta)^2 = 2\mathsf{E}[(Y - X^T\beta)\nabla_\beta(Y - X^T\beta)\\ &= -2\mathsf{E}[(Y - X^T\beta)X],\end{aligned}$$

since the gradient and X are column vectors. The optimality condition $\nabla_\beta \mathrm{MSE}(\beta) = 0$ gives

$$\mathsf{E}(YX) = \mathsf{E}((X^T\beta)X) = \mathsf{E}(X(X^T\beta)) = \mathsf{E}((XX^T)\beta) = \mathsf{E}(XX^T)\beta,$$

$$\beta = (\mathsf{E}(XX^T))^{-1}\mathsf{E}(YX).$$

Estimating β

$$\mathsf{E}(XX^T) \approx \frac{1}{n}\sum_{i=1}^n x_i x_i^T,$$

$$\left(\sum_{i=1}^n x_i x_i^T\right)_{kj} = \sum_{i=1}^n x_{ik}x_{ij} = \sum_{i=1}^n (x^T)_{ki}x_{ij} = (\mathbb{X}^T\mathbb{X})_{kj},$$

$$\mathsf{E}(XX^T) \approx \frac{1}{n}\mathbb{X}^T\mathbb{X}.$$

$$\mathsf{E}(YX) \approx \frac{1}{n}\sum_{i=1}^n y_i x_i = \frac{1}{n}\mathbb{X}^T y.$$

$$\begin{aligned}\beta = (\mathsf{E}(XX^T))^{-1}\mathsf{E}(YX) &\approx \left(\frac{1}{n}\mathbb{X}^T\mathbb{X}\right)^{-1}\frac{1}{n}\mathbb{X}^T y \\ &= \widehat{\beta} := \left(\mathbb{X}^T\mathbb{X}\right)^{-1}\mathbb{X}^T y.\end{aligned}$$

Prediction: $\widehat{Y} = \widehat{f}(X) := X^T\widehat{\beta}$.

Accuracy of the linear regression

Assume that the true model is the following:

$$Y = X^T\beta + \varepsilon, \quad \varepsilon \sim N(0,\sigma^2), \quad X \perp \varepsilon.$$

How do σ^2 and d affect the accuracy of the linear regression? Consider the MSE conditioned on X (fix $X = x$):

$$\mathrm{MSE}(X) = \mathsf{E}((Y - \widehat{Y})^2|X) = \mathsf{E}((X^T\beta + \varepsilon - X^T\widehat{\beta})^2|X).$$

The expectation is taken with respect to the randomness in $\varepsilon, \mathbb{X}, y$. If $y_i = x_i^T\beta + e_i$, $e_i \sim N(0,\sigma^2)$, then

$$\widehat{\beta} = (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T y = (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T(\mathbb{X}\beta + e) = \beta + (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T e,$$

$$
\begin{aligned}
\mathrm{MSE}(X) &= \mathsf{E}((X^T\beta + \varepsilon - X^T\beta - X^T(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T e)^2|X) \\
&= \mathsf{E}((\varepsilon - X^T(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T e)^2|X) \\
&= \mathsf{E}(\varepsilon^2 - 2\varepsilon X^T(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T e + (X^T(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T e)^2|X) \\
&= \sigma^2 + \mathsf{E}((X^T(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T e)^2|X).
\end{aligned}
$$

Since $\mathsf{E}(\varepsilon^2|X) = \sigma^2$,

$$
\mathsf{E}(\varepsilon X^T(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T e|X) = X^T\mathsf{E}(\varepsilon|X)\mathsf{E}(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T e|X) = 0.
$$

Formally transpose the scalar

$$
X^T(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T e = (X^T(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T e)^T = e^T\mathbb{X}(\mathbb{X}^T\mathbb{X})^{-1}X
$$

Then

$$\begin{aligned}
&\mathsf{E}((X^T(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^Te)^2|X) = \mathsf{E}(X^T(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^Te\cdot e^T\mathbb{X}(\mathbb{X}^T\mathbb{X})^{-1}X|X)\\
&= X^T\mathsf{E}((\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^Tee^T\mathbb{X}(\mathbb{X}^T\mathbb{X})^{-1}|X)X\\
&= X^T\mathsf{E}((\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^Tee^T\mathbb{X}(\mathbb{X}^T\mathbb{X})^{-1})X\\
&= X^T\mathsf{E}\mathsf{E}((\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^Tee^T\mathbb{X}(\mathbb{X}^T\mathbb{X})^{-1}|\mathbb{X})X\\
&= X^T\mathsf{E}((\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathsf{E}(ee^T|\mathbb{X})\mathbb{X}(\mathbb{X}^T\mathbb{X})^{-1})X,
\end{aligned}$$

$$\mathsf{E}(ee^T|\mathbb{X}) = \mathsf{E}(ee^T) = \sigma^2 I_n,$$

$$\begin{aligned}
\mathsf{E}((X^T(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^Te)^2|X) &= \sigma^2X^T\mathsf{E}((\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbb{X}(\mathbb{X}^T\mathbb{X})^{-1})X\\
&= \sigma^2X^T\mathsf{E}((\mathbb{X}^T\mathbb{X})^{-1})X
\end{aligned}$$

$$\mathrm{MSE}(X) = \sigma^2 + \sigma^2X^T\mathsf{E}((\mathbb{X}^T\mathbb{X})^{-1})X.$$

$$\mathsf{E}(XX^T) \approx \frac{1}{n}\mathbb{X}^T\mathbb{X} \Longrightarrow (\mathbb{X}^T\mathbb{X})^{-1} \approx \frac{1}{n}(\mathsf{E}(XX^T))^{-1}.$$

$$\mathrm{MSE}(X) \approx \sigma^2 + \frac{\sigma^2}{n}X^T(\mathsf{E}(XX^T))^{-1}X.$$

$$\mathsf{E}[\mathrm{MSE}(X)] \approx \sigma^2 + \frac{\sigma^2}{n}\mathsf{E}(X^T(\mathsf{E}(XX^T))^{-1}X).$$

$$\begin{aligned}
&\mathsf{E}(X^T(\mathsf{E}(XX^T))^{-1}X) = \mathsf{E}(\mathrm{Tr}(X^T(\mathsf{E}(XX^T))^{-1}X)) \\
&= \mathsf{E}(\mathrm{Tr}(XX^T(\mathsf{E}(XX^T))^{-1})) = \mathrm{Tr}(\mathsf{E}(XX^T(\mathsf{E}(XX^T))^{-1})) \\
&= \mathrm{Tr}\left(\mathsf{E}(XX^T)(\mathsf{E}(XX^T))^{-1}\right) = \mathrm{Tr}(I_{d+1}) = d+1.
\end{aligned}$$

Thus,

$$\mathsf{E}[\mathrm{MSE}(X)] \approx \sigma^2 + \frac{\sigma^2}{n}(d+1).$$

We need small noize (small σ^2) and n of order d to get good predictions. No curse of dimensionality for approximately linear data.

The bias-variance trade-off

Assume that the true model is of the form

$$Y = f(X) + \varepsilon, \quad \mathsf{E}\varepsilon = 0,\ \mathrm{Var}(\varepsilon) = \sigma^2,\ \varepsilon \perp X,$$

$$\mathsf{E}(Y|X) = f(X) : \text{the regression function.}$$

Assume that $\widehat{f}(x)$ is obtained from the training data $S = \{(x_i, y_i)\}_{i=1}^n$ by some method. A good predictor should perform well on the training data. This performance is quantified by the training error (empirical loss, empirical risk):

$$\mathscr{L}_S(\widehat{f}) := \frac{1}{n}\sum_{i=1}^n L(y_i, \widehat{f}(x_i)).$$

However we cannot use the training error for choosing between different predictors $\widehat{f}$. In fact, we are interested in the performance of $\widehat{f}$ on new (unseen) inputs: $Y \approx \widehat{f}(X)$.

Example. For k-NN method taking $k = 1$ we get zero training error for any data set.

Example. For $d = 1$ we can always find a polynomial $\widehat{f}(x)$ of sufficiently large order such that $y_i = \widehat{f}(x_i)$. It will produce zero training error, but its predictions can be much worse than the predictions of the linear regression if the dependence is approximately linear.

Example. For binary prediction problem with $\mathcal{Y} = \{0, 1\}$ the classifier

$$\widehat{f}(x) = \begin{cases} y_i, & x = x_i \\ 0, & \text{otherwise} \end{cases}$$

which memorizes the data has zero training error, but can be arbitrary bad on the new data.

We are interested in how well $\widehat{f}$ can generalize from S to new data. The quantity of interest is the test error (generalization error):

$$\mathscr{L}(\widehat{f}) = \mathsf{E} L(Y, \widehat{f}(X)).$$

Let us consider the regression problem:

$$\mathscr{L}(\widehat{f}) = \mathsf{E}(Y - \widehat{f}(X))^2.$$

By the freezing lemma

$$\mathsf{E}((Y - \widehat{f}(X))^2 | X) = \mathsf{E}((\varepsilon + f(X) - \widehat{f}(X))^2 | X) = h(X),$$

$$\begin{aligned} h(x) &= \mathsf{E}((\varepsilon + f(x) - \widehat{f}(x))^2) = \mathsf{E}\varepsilon^2 + \mathsf{E}(f(x) - \widehat{f}(x))^2 \\ &+ 2\mathsf{E}(\varepsilon(f(x) - \widehat{f}(x))) = \sigma^2 + \mathsf{E}(f(x) - \mathsf{E}\widehat{f}(x) + \mathsf{E}\widehat{f}(x) - \widehat{f}(x))^2 \\ &= \sigma^2 + \mathsf{E}(f(x) - \mathsf{E}\widehat{f}(x))^2 + \mathsf{E}(\widehat{f}(x) - \mathsf{E}\widehat{f}(x))^2, \end{aligned}$$

since

$$\mathsf{E}(\varepsilon(f(x) - \widehat{f}(x))) = \mathsf{E}\varepsilon \cdot \mathsf{E}(f(x) - \widehat{f}(x)) = 0,$$

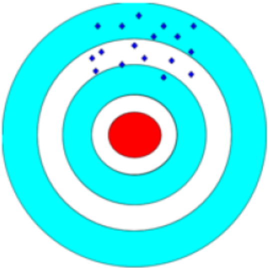
$$\mathsf{E}(f(x) - \mathsf{E}\widehat{f}(x))(\mathsf{E}\widehat{f}(x) - \widehat{f}(x)) = (f(x) - \mathsf{E}\widehat{f}(x))\mathsf{E}(\mathsf{E}\widehat{f}(x) - \widehat{f}(x)) = 0.$$

The bias-variance trade-off

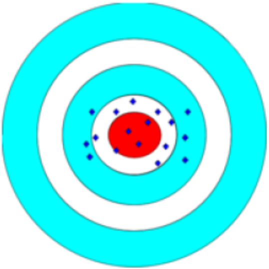
$$\mathsf{E}((Y-\widehat{f}(X))^2|X=x)=\sigma^2+(f(x)-\mathsf{E}\widehat{f}(x))^2+\mathrm{Var}(\widehat{f}(x)).$$

- σ^2: irreducible noise.
- $\mathsf{B}^2(\widehat{f}(x))=(f(x)-\mathsf{E}\widehat{f}(x))^2$: the squared bias of $\widehat{f}$. Quantifies the average error made by approximating the true value $f(x)$ by $\widehat{f}(x)$.
- $\mathrm{Var}(\widehat{f}(x))$: variance of $\widehat{f}$. Quantifies the variability of $\widehat{f}(x)$ with respect to S.

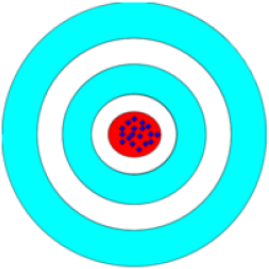
Flexibility of $\widehat{f}$ (complexity of the class from which it is taken) $\uparrow \Longrightarrow$ the squared bias $\mathsf{B}^2(\widehat{f}(x))\downarrow$ but the variance $\mathrm{Var}(\widehat{f}(x))\uparrow$.
Flexibility of $\widehat{f}\uparrow \Longrightarrow$ training error $\mathscr{L}_S(\widehat{f})\downarrow$.
X_i, X should be independent, but need not be identically distributed. In particular, X_i can be deterministic.



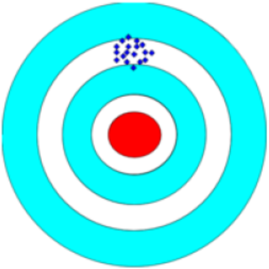
Board of player 1



Board of player 2

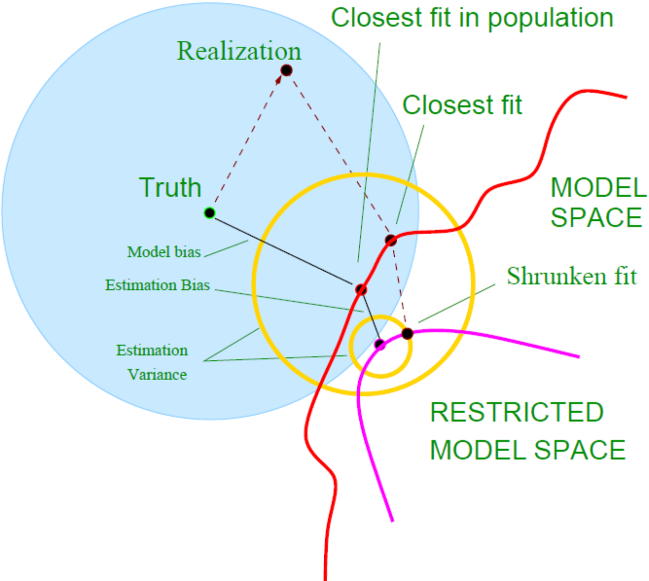


Board of player 3



Board of player 4

Hastie et. al, The Elements of Statistical Learning, 2009 (p. 225)



Overfitting and underfitting

Two bad cases:

- Overfitting: $\mathcal{R}_S(\widehat{f})$ is small but $\mathcal{R}(\widehat{f})$ is large. Very flexible predictor, which adapts too closely to S, and does not generalize well, since it has large variance.
- Underfitting: model is very rigid, it is unable to approximate f. Both $\mathcal{R}_S(\widehat{f})$ and $\mathcal{R}(\widehat{f})$ are large, although the variance is small.

We want to choose the predictor flexibility (complexity of the considered class of predictors) to minimize the test error by getting the right balance between bias and variance.

Example: k-NN regression method

$$\widehat{f}(x) = \frac{1}{k} \sum_{i : x_i \in \mathcal{N}_k} y_i .$$

- $k = N$: $\widehat{f}(x) = \frac{1}{k} \sum_{i=1}^{n} y_i = \widehat{y}$. The regression function f is approximated by a constant, which depends on the data set S.
- $k = 1$: $\widehat{f}(x) = y_{i(x)}$, $i(x) = \arg\min_{i=1,\ldots,n} \operatorname{dist}(x, x_i)$. Consider

  $$V(x_j) = \{x : x_j \text{ is the closest to } x \text{ among all } x_1, \ldots, x_n\}.$$

  The sets $V(x_j)$ are called Voronoi cells. The partition $\mathbb{R}^{d+1} = \bigcup_{i=1}^{n} V(x_i)$ is called a Voronoi diagram (tesselation, decomposition). $\widehat{f}$ is locally constant:

  $$f(x) = y_i, \quad x \in V(x_i).$$

  It is the most flexible k-NN method, it contains n parameters y_i and has zero training loss.

In general, k controls the flexibility of the k-NN regression method (complexity of the k-NN model): $k \uparrow \iff$ flexibility $\downarrow$.
Consider the true model of the form

$$Y_i = f(x_i) + \varepsilon_i, \quad \mathsf{E}\varepsilon_i = 0, \quad \mathrm{Var}(\varepsilon_i) = \sigma^2,$$

where ε_i are i.i.d. and x_i are non-random. Note that Y_i are independent but not identically distributed.
Let's compute the bias and variance of

$$\widehat{f}_k(x) = \frac{1}{k} \sum_{x_i \in \mathcal{N}_k(x)} (f(x_i) + \varepsilon_i).$$

- $\mathsf{B}^2(\widehat{f}_k(x)) = (f(x) - \mathsf{E}\widehat{f}_k(x))^2 = \left(f(x) - \frac{1}{k}\sum_{x_i \in \mathcal{N}_k(x)} f(x_i)\right)^2.$
- $\mathrm{Var}(\widehat{f}_k(x)) = \frac{1}{k^2}\sum_{i=1}^{k} \mathrm{Var}(\varepsilon_i) = \frac{\sigma^2}{k}.$

$$\text{Flexibility} \uparrow \implies k \downarrow \implies \mathsf{B}^2(\widehat{f}_k(x)) \downarrow;\ \mathrm{Var}(\widehat{f}_k(x)) \uparrow.$$

Linear regression: residual sum of squares

Previously:

$$\mathrm{MSE}(\beta) = \mathsf{E}(Y - X^T\beta)^2 \to \min_{\beta}$$

$$\beta^* = (\mathsf{E}(XX^T))^{-1}\mathsf{E}(YX) \approx \widehat{\beta} := \left(\mathbb{X}^T\mathbb{X}\right)^{-1}\mathbb{X}^T y.$$

But it is possible to estimate β directly from data by minimizing the residual sum of squares:

$$\mathcal{R}(\beta) = \mathcal{R}(\beta) = \sum_{i=1}^{n}(y_i - x_i^T\beta)^2 = \|y - \mathbb{X}\beta\|^2 \to \min_{\beta}$$

which is proportional to the training error (empirical risk).

$$\widehat{\beta} \in \arg\min_{\beta\in\mathbb{R}^{d+1}} \mathcal{R}(\beta),$$

$\widehat{\beta}$ is a solution of the system $\mathbb{X}\beta = y$ in the least squares sense.

In the subspace

$$\operatorname{im} X = \{\mathbb{X}\beta : \beta \in \mathbb{R}^{d+1}\} \subset \mathbb{R}^n$$

we want to find an element closest to y. Such element exists, it is unique and it is called a projection of y to $\operatorname{im} X$: notation Πy. Thus, there exists $\widehat{\beta}$: least squares solution (not necessarily unique) such that $\Pi y = \mathbb{X}\widehat{\beta}$. Lets' find $\widehat{\beta}$:

$$\begin{aligned}\mathcal{R}(\beta) &= \|y - \mathbb{X}\beta\|^2 = (y - \mathbb{X}\beta)^T(y - \mathbb{X}\beta) = (y^T - \beta^T\mathbb{X}^T)(y - \mathbb{X}\beta)\\ &= y^T y - 2\beta^T\mathbb{X}^T y + \beta^T\mathbb{X}^T\mathbb{X}\beta,\end{aligned}$$

since $y^T\mathbb{X}\beta = (y^T\mathbb{X}\beta)^T = \beta^T\mathbb{X}^T y$. We have

$$\frac{\partial}{\partial\beta_k}(\beta^T\mathbb{X}^T y) = \frac{\partial}{\partial\beta_k}\sum_{j=0}^{d}\beta_j(\mathbb{X}^T y)_j = (\mathbb{X}^T y)_k.$$

Put $A = \mathbb{X}^T\mathbb{X}$. Then

$$\frac{\partial}{\partial \beta_k}(\beta^T A \beta) = \frac{\partial}{\partial \beta_k} \sum_{i,j=0}^{d} a_{ij}\beta_i\beta_j = \sum_{i,j=0}^{d} a_{ij}\frac{\partial \beta_i}{\partial \beta_k}\beta_j + \sum_{i,j=0}^{d} a_{ij}\beta_i\frac{\partial \beta_j}{\partial \beta_k}$$
$$= \sum_{j=0}^{d} a_{kj}\beta_j + \sum_{i=0}^{d} a_{ik}\beta_i = \sum_{j=0}^{d} a_{kj}\beta_j + \sum_{i=0}^{d} a_{ki}\beta_i = 2(A\beta)_k$$

Thus,

$$\nabla_\beta(\beta^T\mathbb{X}^T y) = \mathbb{X}^T y, \quad \nabla_\beta(\beta^T\mathbb{X}^T\mathbb{X}\beta) = 2\mathbb{X}^T\mathbb{X}\beta,$$
$$\nabla_\beta\mathcal{R}(\beta) = -2\mathbb{X}^T y + 2\mathbb{X}^T\mathbb{X}\beta,$$

and $\widehat{\beta}$ satisfies the system of normal equations

$$\mathbb{X}^T\mathbb{X}\beta = \mathbb{X}^T y.$$

From the existence of a projection it follows that the solution always exists. A solution is unique if and only if one of the following equivalent conditions hold true

- $\ker(\mathbb{X}^T\mathbb{X}) = \{0\}$
- $\ker(\mathbb{X}) = \{0\}$
- the columns $x^{(0)}, \dots, x^{(d)}$ are independent
- $\mathrm{rank}(\mathbb{X}) = d + 1$ (assuming that $n \geq d + 1$)

In any of these cases $\mathbb{X}^T\mathbb{X}$ is positive definite and

$$\widehat{\beta} = (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T y.$$

This is indeed the global minimum, since $\mathrm{RSS}(\beta)$ is convex. It is called an ordinary least squares (OLS) estimate of β.
An OLS is not unique if

- $\mathbb{X}$ is flat (too many features, high-dimensional data): $n < d + 1$,
- $\mathbb{X}$ is flat but still there are dependent columns: redundant features.

Let $\mathbb{X}^T\mathbb{X}$ be invertible. Then

$$\widehat{\beta} = (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T y.$$

The predicted values:

$$\widehat{y}_i = x_i^T\widehat{\beta}.$$

In the matrix form:

$$\widehat{y} = \mathbb{X}\widehat{\beta} = \mathbb{X}(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T y.$$

The "hat matrix"

$$\mathbb{H} = \mathbb{X}(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T$$

puts the hat on the top of y. Since $\widehat{y}$ is the orthogonal projection of y on $\mathbb{X}$, it follows that $\mathbb{H}$ is the corresponding projection operator.

Let (x_i, y_i) be the data, coming from the true model

$$y_i = x_i^T\beta + \varepsilon_i$$

where ε_i are uncorrelated, and satisfy the conditions $\mathsf{E}\varepsilon_i = 0$, $\mathrm{Var}(\varepsilon_i) = \sigma^2$, x_i are non-random, y_i are random.
The OLS estimate $\widehat{\beta} = (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^Ty$ of β is unbiased (if $\mathbb{X}$ is non-random):

$$\begin{aligned}\mathsf{E}\widehat{\beta} &= \mathsf{E}(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^Ty = (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathsf{E}y = (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathsf{E}(\mathbb{X}\beta + \varepsilon)\\ &= (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbb{X}\beta = \beta, \qquad \varepsilon = (\varepsilon_1, \dots, \varepsilon_n)^T.\end{aligned}$$

Covariance matrix of $\widehat{\beta}$:

$$\mathrm{Cov}(\widehat{\beta}) = \mathsf{E}[(\widehat{\beta} - \mathsf{E}\widehat{\beta})(\widehat{\beta} - \mathsf{E}\widehat{\beta})^T]$$

$$\begin{aligned}\widehat{\beta} - \mathsf{E}\widehat{\beta} &= (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^Ty - \beta = (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^Ty - (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbb{X}\beta\\ &= (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T(y - \mathbb{X}\beta) = (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\varepsilon.\end{aligned}$$

$$\begin{aligned}\mathrm{Cov}(\widehat{\beta}) &= \mathsf{E}[(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\varepsilon\cdot\varepsilon^T\mathbb{X}(\mathbb{X}^T\mathbb{X})^{-1}]\\ &= (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathsf{E}[\varepsilon\varepsilon^T]\mathbb{X}(\mathbb{X}^T\mathbb{X})^{-1} = \sigma^2(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbb{X}(\mathbb{X}^T\mathbb{X})^{-1}\\ &= \sigma^2(\mathbb{X}^T\mathbb{X})^{-1},\end{aligned}$$

since $\mathsf{E}[\varepsilon\varepsilon^T] = \sigma^2 I_n$. But σ^2 is unknown. Its unbiased estimate is

$$\widehat{\sigma}^2 = \frac{1}{n-d-1}\sum_{i=1}^n (y_i - \widehat{y}_i)^2, \quad \widehat{y}_i = x_i^T\widehat{\beta}.$$

To prove this consider

$$\begin{aligned}\sum_{i=1}^n (y_i-\widehat{y}_i)^2 = (y-\widehat{y})^T(y-\widehat{y}) &= (\mathbb{X}\beta+\varepsilon-\mathbb{X}\widehat{\beta})^T(\mathbb{X}\beta+\varepsilon-\mathbb{X}\widehat{\beta})\\ &= (\varepsilon-\mathbb{X}(\widehat{\beta}-\beta))^T(\varepsilon-\mathbb{X}(\widehat{\beta}-\beta))\\ &= \varepsilon^T\varepsilon - 2\varepsilon^T\mathbb{X}(\widehat{\beta}-\beta) + (\widehat{\beta}-\beta)^T\mathbb{X}^T\mathbb{X}(\widehat{\beta}-\beta),\end{aligned}$$

since $(\widehat{\beta}-\beta)^T\mathbb{X}^T\varepsilon = ((\widehat{\beta}-\beta)^T\mathbb{X}^T\varepsilon)^T = \varepsilon^T\mathbb{X}(\widehat{\beta}-\beta)$.

$$\mathsf{E}[\varepsilon^T\varepsilon] = \mathsf{E}\sum_{i=1}^{n}\varepsilon_i^2 = n\sigma^2,$$

$$\begin{aligned}
\mathsf{E}\left[\varepsilon^T\mathbb{X}(\widehat{\beta}-\beta)\right] &= \mathsf{E}\left[\varepsilon^T\mathbb{X}((\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^Ty-\beta)\right] \\
&= \mathsf{E}\left[\varepsilon^T\mathbb{X}((\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T(\mathbb{X}\beta+\varepsilon)-\beta)\right] \\
&= \mathsf{E}\left[\varepsilon^T\mathbb{X}(\beta+(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\varepsilon-\beta)\right] \\
&= \mathsf{E}\left[\varepsilon^T\mathbb{X}(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\varepsilon\right] = \mathsf{E}\left[\mathrm{Tr}(\varepsilon^T\mathbb{X}(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\varepsilon)\right] \\
&= \mathsf{E}\left[\mathrm{Tr}(\mathbb{X}^T\varepsilon\varepsilon^T\mathbb{X}(\mathbb{X}^T\mathbb{X})^{-1})\right] = \mathrm{Tr}\,\mathsf{E}\left[\mathbb{X}^T\varepsilon\varepsilon^T\mathbb{X}(\mathbb{X}^T\mathbb{X})^{-1}\right] \\
&= \mathrm{Tr}\left(\mathbb{X}^T\mathsf{E}[\varepsilon\varepsilon^T]\mathbb{X}(\mathbb{X}^T\mathbb{X})^{-1}\right) = \sigma^2\mathrm{Tr}\left(\mathbb{X}^T\mathbb{X}(\mathbb{X}^T\mathbb{X})^{-1}\right) \\
&= \sigma^2\mathrm{Tr}(I_{d+1}) = \sigma^2(d+1).
\end{aligned}$$

$$
\begin{aligned}
\mathsf{E}[(\widehat{\beta}-\beta)^T\mathbb{X}^T\mathbb{X}(\widehat{\beta}-\beta)] &= \mathsf{E}\left[\mathrm{Tr}\left((\widehat{\beta}-\beta)^T\mathbb{X}^T\mathbb{X}(\widehat{\beta}-\beta)\right)\right] \\
&= \mathsf{E}\left[\mathrm{Tr}\left(\mathbb{X}^T\mathbb{X}(\widehat{\beta}-\beta)(\widehat{\beta}-\beta)^T\right)\right] \\
&= \mathrm{Tr}\left(\mathbb{X}^T\mathbb{X}\mathsf{E}\left[\widehat{\beta}-\beta)(\widehat{\beta}-\beta)^T\right]\right) \\
&= \mathrm{Tr}\left(\mathbb{X}^T\mathbb{X}\cdot\mathrm{Cov}(\widehat{\beta})\right) = \sigma^2\mathrm{Tr}\left(\mathbb{X}^T\mathbb{X}(\mathbb{X}^T\mathbb{X})^{-1}\right) \\
&= \sigma^2\mathrm{Tr}(I_{d+1}) = \sigma^2(d+1).
\end{aligned}
$$

Thus,

$$
\mathsf{E}\sum_{i=1}^{n}(y_i-\widehat{y}_i)^2 = \sigma^2 n - 2\sigma^2(d+1) + \sigma^2(d+1) = (n-d-1)\sigma^2.
$$

Bias-variance decomposition for linear regression

$$\mathsf{E}((Y-\widehat{f}(X))^2|X=x)=\sigma^2+(f(x)-\mathsf{E}\widehat{f}(x))^2+\mathrm{Var}(\widehat{f}(x)).$$

$$f(x)=\langle\beta,x\rangle,\quad \widehat{f}(x)=\langle\widehat{\beta},x\rangle.$$

Since $\widehat{\beta}$ is an unbiased estimate of β, the bias is zero and

$$\mathsf{E}((Y-\widehat{f}(X))^2|X=x)=\sigma^2+\mathrm{Var}(\langle\widehat{\beta},x\rangle).$$

Furthermore,

$$\mathrm{Var}(\langle\widehat{\beta},x\rangle)=\sum_{i,j=0}^{d}\mathrm{Cov}(\widehat{\beta}_i,\widehat{\beta}_j)x_ix_j=\langle\mathrm{Cov}(\widehat{\beta})x,x\rangle=\sigma^2\langle(\mathbb{X}^T\mathbb{X})^{-1}x,x\rangle.$$

As was mentioned above,

$$\mathsf{E}(XX^T)\approx\frac{1}{n}\mathbb{X}^T\mathbb{X}\Longrightarrow(\mathbb{X}^T\mathbb{X})^{-1}\approx\frac{1}{n}(\mathsf{E}(XX^T))^{-1}.$$

Hence,

$$\mathsf{E}((Y - \widehat{f}(X))^2|X) \approx \sigma^2 + \frac{\sigma^2}{n}\langle(\mathsf{E}(XX^T))^{-1}X, X\rangle.$$

It was shown that

$$\mathsf{E}(X^T(\mathsf{E}(XX^T))^{-1}X) = d + 1.$$

Thus,

$$\mathsf{E}(Y - \widehat{f}(X))^2 \approx \sigma^2 + \frac{\sigma^2}{n}(d + 1).$$

This result was obtained above by somewhat another calculations. Now we see that it is related to bias-variance decomposition.

LOOCV for OLS

Let $\widehat{f}$ be a linear regression model, fitted to the whole data set: $\widehat{f}(x) = x^T \widehat{\beta}$. The LOOCV estimate of its test error is

$$CV(\widehat{f}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - x_i^T \widehat{\beta}_{(-i)})^2.$$

Here $\widehat{\beta}_{(-i)}$ is the OLS estimate of β based on $S_i = S \backslash \{(x_i, y_i)\}$:

$$\widehat{\beta}_{(-i)} = (\mathbb{X}_{(-i)}^T \mathbb{X}_{(-i)})^{-1} \mathbb{X}_{(-i)}^T y_{(-i)},$$

$$\mathbb{X}_{(-i)} = \begin{pmatrix} x_1^T \\ \vdots \\ x_{i-1}^T \\ x_{i+1}^T \\ \vdots \\ x_n^T \end{pmatrix} \in \mathbb{R}^{(n-1)\times(d+1)}, \quad y_{(-i)} = \begin{pmatrix} y_1^T \\ \vdots \\ y_{i-1}^T \\ y_{i+1}^T \\ \vdots \\ y_n^T \end{pmatrix} \in \mathbb{R}^{(n-1)\times 1}.$$

Sherman-Morrison formula (Wikipedia)

Theorem

Suppose $A \in \mathbb{R}^{n \times n}$ is an invertible matrix and $u, v \in \mathbb{R}^n$ are column vectors. Then $A + uv^T$ is invertible if and only if $1 + v^T A^{-1} u \neq 0$. In this case,

$$\left(A + uv^T\right)^{-1} = A^{-1} - \frac{A^{-1} u v^T A^{-1}}{1 + v^T A^{-1} u}.$$

Proof. "$\Longleftarrow$" Assume that $1 + v^T A^{-1} u \neq 0$ and put

$$Y = A^{-1} - \frac{A^{-1} u v^T A^{-1}}{1 + v^T A^{-1} u}.$$

We have

$$\begin{aligned}
(A+uv^T)Y &= (A+uv^T)\left(A^{-1} - \frac{A^{-1}uv^TA^{-1}}{1+v^TA^{-1}u}\right) \\
&= AA^{-1} + uv^TA^{-1} - \frac{AA^{-1}uv^TA^{-1} + uv^TA^{-1}uv^TA^{-1}}{1+v^TA^{-1}u} \\
&= I + uv^TA^{-1} - \frac{uv^TA^{-1} + uv^TA^{-1}uv^TA^{-1}}{1+v^TA^{-1}u} \\
&= I + uv^TA^{-1} - \frac{u\left(1+v^TA^{-1}u\right)v^TA^{-1}}{1+v^TA^{-1}u} \\
&= I + uv^TA^{-1} - uv^TA^{-1} = I
\end{aligned}$$

$\implies \operatorname{im}(A+uv^T) = \mathbb{R}^n \implies A+uv^T$ is invertible and its inverse is Y.
"$\implies$" If $1+v^TA^{-1}u = 0$, then $A+uv^T$ is not invertible:

$$\left(A+uv^T\right)A^{-1}u = u + uv^TA^{-1}u = u(1+v^TA^{-1}u) = 0. \quad \Box$$

Theorem

$$CV(\widehat{f}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i^T\widehat{\beta}_{(-i)})^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - x_i^T\widehat{\beta}}{1-H_{ii}}\right)^2,$$

where $\widehat{\beta}$ is the OLS estimate obtained from the full data S, H is the hat matrix:

$$\widehat{y} = Hy, \quad H = \mathbb{X}(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T.$$

Proof. Let us show that $y_i - x_i^T\widehat{\beta}_{(-i)} = \frac{y_i - x_i^T\widehat{\beta}}{1-H_{ii}}$. To this end let us express

$$\widehat{\beta}_{(-i)} = (\mathbb{X}_{(-i)}^T\mathbb{X}_{(-i)})^{-1}\mathbb{X}_{(-i)}^T y_{(-i)}$$

in terms of $\mathbb{X}$ and y:

$$\left(\mathbb{X}_{(-i)}^T y_{(-i)}\right)_k = \sum_{s=1}^{n}\mathbb{X}_{ks}^T y_s - \mathbb{X}_{ki}^T y_i = (\mathbb{X}^T y)_k - \mathbb{X}_{ik} y_i = (\mathbb{X}^T y)_k - (x_i y_i)_k,$$

$$\mathbb{X}_{(-i)}^T y_{(-i)} = \mathbb{X}^T y - x_i y_i.$$

$$\left(\mathbb{X}_{(-i)}^T \mathbb{X}_{(-i)}\right)_{jk} = \sum_{s=1}^{n-1} \left(\mathbb{X}_{(-i)}^T\right)_{js} \left(\mathbb{X}_{(-i)}\right)_{sk} = \sum_{s=1}^{n-1} \left(\mathbb{X}_{(-i)}\right)_{sj} \left(\mathbb{X}_{(-i)}\right)_{sk}$$

$$= \sum_{s=1}^{n} x_{sj} x_{sk} - x_{ij} x_{ik} = (\mathbb{X}^T \mathbb{X})_{jk} - (x_i x_i^T)_{jk},$$

$$\mathbb{X}_{(-i)}^T \mathbb{X}_{(-i)} = \mathbb{X}^T \mathbb{X} - x_i x_i^T.$$

Put $A = \mathbb{X}^T \mathbb{X}$, $u = x_i$, $v = -x_i$ in the Sherman-Morrison formula:

$$\left(\mathbb{X}_{(-i)}^T \mathbb{X}_{(-i)}\right)^{-1} = \left(\mathbb{X}^T \mathbb{X} - x_i x_i^T\right)^{-1}$$

$$= \left(\mathbb{X}^T \mathbb{X}\right)^{-1} + \frac{\left(\mathbb{X}^T \mathbb{X}\right)^{-1} x_i x_i^T \left(\mathbb{X}^T \mathbb{X}\right)^{-1}}{1 - x_i^T \left(\mathbb{X}^T \mathbb{X}\right)^{-1} x_i}$$

$$= \left(\mathbb{X}^T \mathbb{X}\right)^{-1} + \frac{\left(\mathbb{X}^T \mathbb{X}\right)^{-1} x_i x_i^T \left(\mathbb{X}^T \mathbb{X}\right)^{-1}}{1 - H_{ii}},$$

since $x_i^T \left(\mathbb{X}^T \mathbb{X}\right)^{-1} x_i = (\mathbb{X}(\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X}^T)_{ii} = H_{ii}$.

To apply this formula we need that $H_{ii} \neq 1$. The equalities $H = H^2$, $H = H^T$ imply that

$$H_{ii} = \sum_{k=1}^{n} H_{is} H_{si} = H_{ii}^2 + \sum_{s \neq i} H_{is}^2 .$$

Thus, $0 \leq H_{ii} \leq 1$. If $H_{ii} = 1$, then $H_{is} = 0$, $s \neq i$. In this exotic case $\widehat{y}_i = (Hy)_i = y_i$.

$$\begin{aligned}
\widehat{\beta}_{(-i)} &= (\mathbb{X}_{(-i)}^T \mathbb{X}_{(-i)})^{-1} \mathbb{X}_{(-i)}^T y_{(-i)} \\
&= \left(\left(\mathbb{X}^T \mathbb{X}\right)^{-1} + \frac{\left(\mathbb{X}^T \mathbb{X}\right)^{-1} x_i x_i^T \left(\mathbb{X}^T \mathbb{X}\right)^{-1}}{1 - H_{ii}} \right) (\mathbb{X}^T y - x_i y_i) \\
&= \left(\mathbb{X}^T \mathbb{X}\right)^{-1} \mathbb{X}^T y - \left(\mathbb{X}^T \mathbb{X}\right)^{-1} x_i y_i \\
&+ \frac{\left(\mathbb{X}^T \mathbb{X}\right)^{-1} x_i}{1 - H_{ii}} \left(x_i^T \left(\mathbb{X}^T \mathbb{X}\right)^{-1} \mathbb{X}^T y - x_i^T \left(\mathbb{X}^T \mathbb{X}\right)^{-1} x_i y_i \right)
\end{aligned}$$

$$\begin{aligned}
\widehat{\beta}_{(-i)} &= \widehat{\beta} - \frac{\left(\mathbb{X}^T\mathbb{X}\right)^{-1} x_i}{1-H_{ii}}(1-H_{ii})y_i + \frac{\left(\mathbb{X}^T\mathbb{X}\right)^{-1} x_i}{1-H_{ii}} \left(x_i^T\widehat{\beta} - H_{ii}y_i\right) \\
&= \widehat{\beta} - \frac{\left(\mathbb{X}^T\mathbb{X}\right)^{-1} x_i}{1-H_{ii}} \left(y_i - x_i^T\widehat{\beta}\right)
\end{aligned}$$

$$\begin{aligned}
y_i - x_i^T\widehat{\beta}_{(-i)} &= y_i - x_i^T \left(\widehat{\beta} - \frac{\left(\mathbb{X}^T\mathbb{X}\right)^{-1} x_i}{1-H_{ii}} \left(y_i - x_i^T\widehat{\beta}\right)\right) \\
&= y_i - x_i^T\widehat{\beta} + \frac{x_i^T \left(\mathbb{X}^T\mathbb{X}\right)^{-1} x_i}{1-H_{ii}} \left(y_i - x_i^T\widehat{\beta}\right) \\
&= (y_i - x_i^T\widehat{\beta})\left(1 + \frac{H_{ii}}{1-H_{ii}}\right) = \frac{y_i - x_i^T\widehat{\beta}}{1-H_{ii}}. \quad \square
\end{aligned}$$

Ridge regression

Ridge estimates $\widehat{\beta}^R$ are obtained from the optimization problem

$$\mathcal{R}(\beta) = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{d} \beta_j x_{ij} \right)^2 \to \min, \quad \sum_{j=1}^{d} \beta_j^2 \leq s.$$

The intercept β_0 is not penalized. By shrinking β_j we want to reduce the influence of the associated feature X_j. But β_0 is not associated to any feature: it is the response, when all features are zero.
The Ridge regression optimization problem can be equivalently rewritten in the Lagrangian form:

$$\mathcal{R}(\beta; \lambda) = \mathcal{R}(\beta) + \lambda \sum_{j=1}^{d} \beta_j^2 \to \min_{\beta}.$$

That is, for any $s > 0$ there exists $\lambda > 0$ such that the optimal solutions are the same. λ controls the model flexibility: $\lambda \uparrow \Longrightarrow \text{flexibility} \downarrow$.

- $\lambda = 0$: Ridge regression = OLS regression,
- $\lambda \to \infty$: $\widehat{\beta}^R \to (\widehat{\beta}_0^R, 0, \ldots, 0)$; Ridge regression → Null model.

Since all β_j are equally penalized, it is desirable to standardize the features

$$\widetilde{x}_{ij} = \frac{x_{ij} - \overline{x}_j}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \overline{x}_j)^2}}, \quad \overline{x}_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}.$$

For the transformed features,

$$\overline{\widetilde{x}}_j = \frac{1}{n}\sum_{i=1}^{n}\widetilde{x}_{ij} = 0, \quad \frac{1}{n}\sum_{i=1}^{n}(\widetilde{x}_{ij} - \overline{\widetilde{x}}_j)^2 = \frac{1}{n}\sum_{i=1}^{n}\widetilde{x}_{ij}^2 = 1.$$

The new (test) inputs $X = (X_1, \ldots, X_d)$ must be transformed in the same way (using only the training data) before making predictions:

$$\widetilde{X}_{ij} = \frac{X_{ij} - \overline{x}_j}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \overline{x}_j)^2}}, \qquad \widehat{Y} = \widetilde{X}^T \widehat{\beta}^R.$$

Assume that the features are standardized. Let us find $\widehat{\beta}_0^R$:

$$\begin{aligned}\frac{\partial \mathcal{R}(\beta;\lambda)}{\partial \beta_0} = 0 &\iff \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{d}\beta_j x_{ij}\right) = 0\\ &\iff \sum_{i=1}^{n} y_i - n\beta_0 - \sum_{j=1}^{d}\beta_j \sum_{i=1}^{n} x_{ij} = 0\\ &\iff \sum_{i=1}^{n} y_i - n\beta_0 = 0.\end{aligned}$$

Thus, $\widehat{\beta}_0^R = \overline{y} := \frac{1}{n}\sum_{i=1}^{n} y_i$. For centered labels $\widetilde{y}_i = y_i - \overline{y}$ we would get $\widehat{\beta}_0^R = 0$. In this case we can consider the Ridge regression model without the intercept:

$$\mathcal{R}(\beta;\lambda) = \sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{d}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{d}\beta_j^2 \to \min_{\beta\in\mathbb{R}^d}.$$

In the matrix form:

$$\mathcal{R}(\beta;\lambda) = \|y - \mathbb{X}\beta\|_2^2 + \lambda\|\beta\|_2^2,$$

$\mathbb{X} \in \mathbb{R}^{n\times d}$ with standardized columns, $y \in \mathbb{R}^{n\times 1}$: centered. Prediction:

$$\widehat{Y} = \overline{y} + \widetilde{X}^T \widehat{\beta}^R.$$

Optimal solution $\widehat{\beta}_R$ for ridge regression:

$$\nabla_\beta \mathcal{R}(\beta;\lambda) = -2\mathbb{X}^T y + 2\mathbb{X}^T\mathbb{X}\beta + 2\lambda\beta = 0,$$

$$\widehat{\beta}^R = (\mathbb{X}^T\mathbb{X} + \lambda I_d)^{-1}\mathbb{X}^T y.$$

$\mathbb{X}^T\mathbb{X} + \lambda I_d$ is always positive definite for $\lambda > 0$. λI_d is called "ridge". Formula works for singular $\mathbb{X}^T\mathbb{X}$, in particular, for $d > n$.

Ridge regression with cross-validation

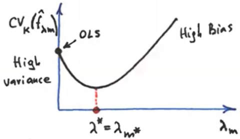
1. Choose a grid of λ values: $0 = \lambda_1 < \lambda_2 < \ldots < \lambda_M$ ⟵ defines M competing models
2. Partition the full data D into K folds at random, $D = F_1 \sqcup \ldots \sqcup F_K$
3. For each fold $k = 1, \ldots, K$ ⟵ defines the split $D = T_k \sqcup V_k,\ T_k = D \setminus F_k,\ V_k = F_k$
   (a) Preprocess the training $T_k = \{\mathbb{X}_{(-V_k)}, y_{(-V_k)}\}$ and validation $V_k = \{\mathbb{X}_{(V_k)}, y_{(V_k)}\}$ sets:
   - Standardize $\mathbb{X}_{(-V_k)}$, center $y_{(-V_k)}$
   - Transform $\mathbb{X}_{(V_k)}$ and $y_{(V_k)}$ in the same way

   $$\hat\beta^R_{\lambda_m} = (\mathbb{X}^T_{(-V_k)}\mathbb{X}_{(-V_k)} + \lambda_m I_p)^{-1}\mathbb{X}^T_{(-V_k)} y_{(-V_k)}$$

   For each model $m = 1, \ldots, M$ ⟵ defines λ_m

   $$\hat f_{\lambda_m}(x) = x^T \hat\beta^R_{\lambda_m}$$

   (b) Fit the ridge regression model with $\lambda = \lambda_m$ to the training data T_k: $\quad f_{\lambda_m} \to T_k \to \hat f_{\lambda_m}$
   (c) Compute the test error of $\hat f_{\lambda_m}$ on the validation set V_k: $\quad \mathrm{Err}(k,m) = \frac{1}{|V_k|}\sum_{i\in V_k}(y_i - x_i^T\hat\beta^R_{\lambda_m})^2$
4. Compute the cross-validation estimate of the test error of model f_{λ_m}:
   $$\mathrm{Err}(f_{\lambda_m}) = CV_K(\hat f_{\lambda_m}) = \frac{1}{K}\sum_{k=1}^K \mathrm{Err}(k,m)$$
5. Find the optimal value of λ: $\lambda^* = \lambda_{m^*},\quad m^* = \arg\min_m CV_K(\hat f_{\lambda_m})$
6. Refit the selected model to the full data D:
   - Standardize $\mathbb{X}$, center y
   - Compute $\hat\beta^R_{\lambda^*} = (\mathbb{X}^T\mathbb{X} + \lambda^* I_p)^{-1}\mathbb{X}^T y$



$\Rightarrow$ model prediction

$$\hat Y = \hat f_{\lambda^*}(X) = \bar y + \overset{\text{transformed}}{X^T}\hat\beta^R_{\lambda^*}$$

Geometric interpretation

Let $\widehat{\beta}$ be an OLS solution. Then $\mathcal{R}'(\widehat{\beta}) = 0$ and

$$\mathcal{R}(\beta) = \mathcal{R}(\widehat{\beta}) + \frac{1}{2}\langle \mathcal{R}''(\widehat{\beta})(\beta - \widehat{\beta}, \beta - \widehat{\beta}\rangle.$$

Let us find the solution $\widehat{\beta}_R$ of the regularized problem:

$$\mathcal{R}(\beta;\lambda) = \mathcal{R}(\beta) + \frac{\lambda}{2}\|\beta\|_2^2 = \mathcal{R}(\widehat{\beta}) + \frac{1}{2}\langle \mathcal{R}''(\widehat{\beta})(\beta - \widehat{\beta}, \beta - \widehat{\beta}\rangle + \frac{\lambda}{2}\|\beta\|_2^2,$$

$$\nabla_\beta \mathcal{R}(\beta;\lambda) = \mathcal{R}''(\widehat{\beta})(\beta - \widehat{\beta}) + \lambda\beta = 0,$$

$$\widehat{\beta}_R = (\lambda I + \mathcal{R}''(\widehat{\beta}))^{-1}\mathcal{R}''(\widehat{\beta})\widehat{\beta}.$$

Spectral decomposition:

$$\widehat{\beta}_R = Q\Sigma Q^T,$$

$$Q = (v_1, \ldots, v_n), \quad \Sigma = \operatorname{diag}(\sigma_1^2, \ldots, \sigma_n^2),$$

v_i: orthonormal eigenvectors of $\mathcal{R}''(\widehat{\beta})$, σ_i^2: eigenvalues of $\mathcal{R}''(\widehat{\beta})$.

$$\widehat{\beta}_R = (\lambda I + Q\Sigma Q^T)^{-1} Q\Sigma Q^T \widehat{\beta},$$

$$(\lambda I + Q\Sigma Q^T)^{-1} = (Q(\lambda I + \Sigma)Q^T)^{-1} = Q(\lambda I + \Sigma)^{-1} Q^T,$$

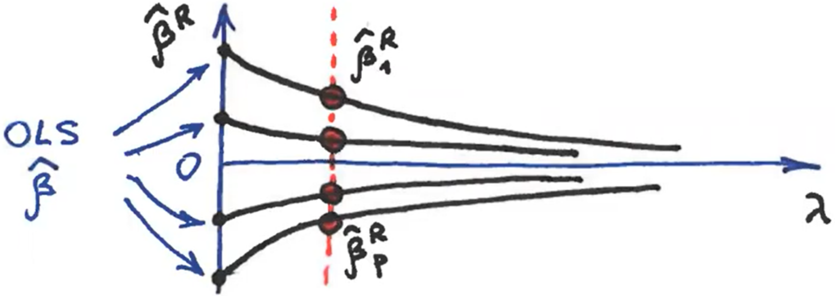
$$\widehat{\beta}_R = Q(\lambda I + \Sigma)^{-1} \Sigma Q^T \widehat{\beta}.$$

$$(\lambda I + \Sigma)^{-1}\Sigma Q^T \widehat{\beta} = \operatorname{diag}\left(\frac{\sigma_1^2}{\sigma_1^2 + \lambda}, \ldots, \frac{\sigma_n^2}{\sigma_n^2 + \lambda}\right) \begin{pmatrix} \langle v_1, \widehat{\beta} \rangle \\ \vdots \\ \langle v_n, \widehat{\beta} \rangle \end{pmatrix}$$

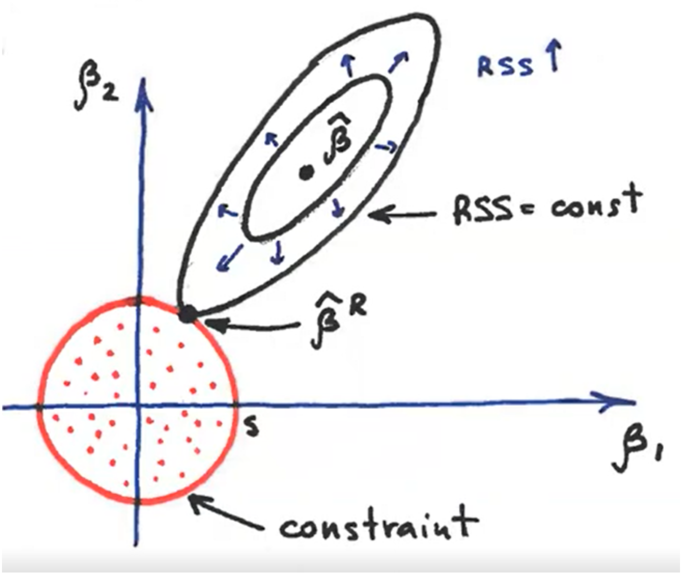
$$= \begin{pmatrix} \frac{\sigma_1^2}{\sigma_1^2+\lambda} \langle v_1, \widehat{\beta} \rangle \\ \vdots \\ \frac{\sigma_n^2}{\sigma_n^2+\lambda} \langle v_n, \widehat{\beta} \rangle \end{pmatrix}$$

$$\widehat{\beta}_R = Q \begin{pmatrix} \frac{\sigma_1^2}{\sigma_1^2+\lambda} \langle v_1, \widehat{\beta} \rangle \\ \vdots \\ \frac{\sigma_n^2}{\sigma_n^2+\lambda} \langle v_n, \widehat{\beta} \rangle \end{pmatrix} = \sum_{i=1}^{n} \frac{\sigma_i^2}{\sigma_i^2 + \lambda} \langle v_i, \widehat{\beta} \rangle v_i .$$

$\widehat{\beta}_R$ is obtained by shrinking $\widehat{\beta}$ along the directions v_i.


$\hat{\beta}^R$
$\hat{\beta}_1^R$
OLS
$\hat{\beta}$
0
λ
$\hat{\beta}_p^R$

Usually all coefficients $\widehat{\beta}_j^R$ are non-zero.



Lasso

Lasso: Least Absolute Shrinkage and Selection Operator. Lasso estimates $\widehat{\beta}^L$:

$$\mathcal{R}(\beta) = \|y - \mathbb{X}\beta\|_2^2 \to \min, \quad \|\beta\|_1 = \sum_{j=1}^{d} |\beta_j| \leq s.$$

Lagrangian form:

$$\mathcal{R}(\beta) + \lambda \sum_{j=1}^{d} |\beta_j| \to \min_{\beta}.$$

λ controls the model flexibility: $\lambda \uparrow \Longrightarrow \text{flexibility} \downarrow$.

- $\lambda = 0$: Lasso = OLS regression,
- $\lambda \to \infty$: $\widehat{\beta}^L \to (\widehat{\beta}_0^L, 0, \ldots, 0)$; Lasso → Null model.

As for Ridge regression, it is desirable to standardize the features

$$\widetilde{x}_{ij} = \frac{x_{ij} - \overline{x}_j}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \overline{x}_j)^2}}, \quad \overline{x}_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}.$$

If the inputs are standardized, then $\widehat{\beta}_0^L = \overline{y}$: the proof is the same as for Ridge regression. For centered labels $\overline{y} = 0$ we can consider the Lasso model without the intercept:

$$\sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{d} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{d} |\beta_j| \to \min_{\beta \in \mathbb{R}^d} .$$

In the matrix form:

$$\|y - \mathbb{X}\beta\|_2^2 + \lambda \|\beta\|_1 \to \min_{\beta \in \mathbb{R}^d},$$

$\mathbb{X} \in \mathbb{R}^{n \times d}$ with standardized columns, $y \in \mathbb{R}^{n \times 1}$: centered. Prediction:

$$\widehat{Y} = \overline{y} + \widetilde{X}^T \widehat{\beta}^L.$$

Lasso with cross-validation

1. Choose a grid of λ values: $0 = \lambda_1 < \lambda_2 < \ldots < \lambda_M$ ⟵ defines M competing models
2. Partition the full data D into K folds at random, $D = F_1 \sqcup \ldots \sqcup F_K$
3. For each fold $k = 1, \ldots, K$ ⟵ defines the split $D = T_k \sqcup V_k,\ T_k = D \setminus F_k,\ V_k = F_k$

   (a) Preprocess the training $T_k = \{\mathbb{X}_{(-V_k)}, y_{(-V_k)}\}$ and validation $V_k = \{\mathbb{X}_{(V_k)}, y_{(V_k)}\}$ sets:

   - Standardize $\mathbb{X}_{(-V_k)}$, center $y_{(-V_k)}$
   - Transform $\mathbb{X}_{(V_k)}$ and $y_{(V_k)}$ in the same way

   For each model $m = 1, \ldots, M$ ⟵ defines λ_m

   computed by solving the optimization problem numerically

   (b) Fit the LASSO model with $\lambda = \lambda_m$ to the training data T_k: $f_{\lambda_m} \to T_k \to \hat{f}_{\lambda_m}$ $\quad \hat{f}_{\lambda_m}(x) = x^T \hat{\beta}^L_{\lambda_m}$

   (c) Compute the test error of $\hat{f}_{\lambda_m}$ on the validation set V_k: $\quad \mathrm{Err}(k,m) = \frac{1}{|V_k|} \sum_{i \in V_k} (y_i - x_i^T \hat{\beta}^L_{\lambda_m})^2$

4. Compute the cross-validation estimate of the test error of model f_{λ_m}:

$$\mathrm{Err}(f_{\lambda_m}) = CV_K(\hat{f}_{\lambda_m}) = \frac{1}{K} \sum_{k=1}^{K} \mathrm{Err}(k,m)$$

5. Find the optimal value of λ: $\lambda^* = \lambda_{m^*},\ \ m^* = \arg\min_m CV_K(\hat{f}_{\lambda_m})$

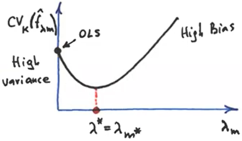
6. Refit the selected model to the full data D:

   - Standardize $\mathbb{X}$, center y
   - Compute $\hat{\beta}^L_{\lambda^*}$

⇒

model prediction

$$\hat{Y} = \hat{f}_{\lambda^*}(X) = \bar{y} + X^T \hat{\beta}^L_{\lambda^*}$$



Geometric interpretation

Let $\widehat{\beta}$ be an OLS solution. Then $\mathcal{R}'(\widehat{\beta}) = 0$ and

$$\mathcal{R}(\beta) = \mathcal{R}(\widehat{\beta}) + \frac{1}{2}\langle \mathcal{R}''(\widehat{\beta})(\beta - \widehat{\beta}, \beta - \widehat{\beta}\rangle,$$

$$\mathcal{R}(\beta;\lambda) = \mathcal{R}(\beta) + \frac{\lambda}{2}\|\beta\|_2^2 = \mathcal{R}(\widehat{\beta}) + \frac{1}{2}\langle \mathcal{R}''(\widehat{\beta})(\beta - \widehat{\beta}, \beta - \widehat{\beta}\rangle + \lambda\|\beta\|_1.$$

Assume that $\mathcal{R}''(\widehat{\beta}) = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_n^2)$. Then the optimization problem $\mathcal{R}(\beta;\lambda) \to \min_\beta$ decomposes:

$$\varphi_j(\beta_j) = \frac{1}{2}\sigma_j^2(\beta_i - \widehat{\beta}_j)^2 + \lambda|\beta_j| \to \min_{\beta_j}.$$

The exists a unique minimum point $\widehat{\beta}_j^L$ (why?). If $\widehat{\beta}_j^L > 0$, then

$$\varphi_j'(\beta_j) = \sigma_j^2(\beta_j - \widehat{\beta}_j) + \lambda = 0 \Longrightarrow \widehat{\beta}_j^L = \widehat{\beta}_j - \frac{\lambda}{\sigma_j^2} \iff \widehat{\beta}_j > \frac{\lambda}{\sigma_j^2}$$

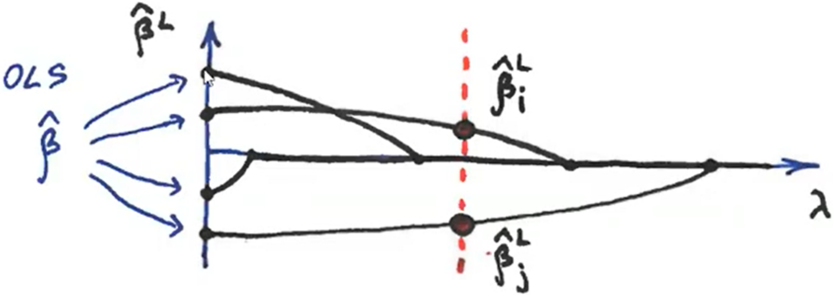
If $\widehat{\beta}_j^L < 0$, then

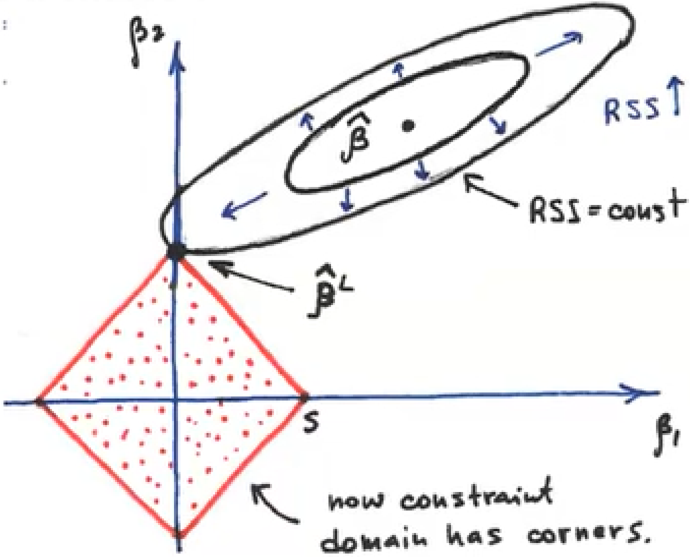
$$\varphi_j'(\beta_j) = \sigma_j^2(\beta_j - \widehat{\beta}_j) - \lambda = 0 \Longrightarrow \widehat{\beta}_j^L = \widehat{\beta}_j + \frac{\lambda}{\sigma_j^2} \iff \widehat{\beta}_j < -\frac{\lambda}{\sigma_j^2}.$$

For $|\widehat{\beta}_j| \le \lambda/\sigma_j^2$ the only possibility is $\widehat{\beta}_j^L = 0$. Thus,

$$\widehat{\beta}_j^L = \begin{cases} \widehat{\beta}_j + \lambda/\sigma_j^2, & \widehat{\beta}_j < -\lambda/\sigma_j^2, \\ 0, & |\widehat{\beta}_j| \le \lambda/2, \\ \widehat{\beta}_j - \lambda/\sigma_j^2, & \widehat{\beta}_j > \lambda/\sigma_j^2. \end{cases}$$

In contrast to Ridge regression, Lasso not only shrinks β_j, but also forces some $\beta_j = 0$ when λ is large enough.





β_2
$\hat{\beta}$
RSS ↑
RSS = const
$\hat{\beta}^L$
s
β_1
now constraint domain has corners.

Probabilistic approach to regression

$(X_i, Y_i) \in \mathbb{R}^d \times \mathbb{R}$. Model:

$$Y_i = f_w(x_i) + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2), \quad X_i \perp \varepsilon_i.$$

For linear regression, $f_w(x_i) = \langle \beta, x_i \rangle$. Clearly, $Y_i \sim N(f_w(X_i), \sigma^2)$ conditionally on X_i:

$$p_{Y_i|X_i}(y_i|x_i; w) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(f_w(x_i) - y_i)^2}{2\sigma^2} \right).$$

MLE gives the OLS (ordinary least squares) problem:

$$\begin{aligned} w^* &\in \arg\max_w p(s; w) = \arg\max_w \prod_{i=1}^n p(x_i, y_i; w) \\ &= \arg\max_w \prod_{i=1}^n p(y_i | x_i; w) p(x_i) = \arg\max_w \prod_{i=1}^n p(y_i | x_i; w) \\ &= \arg\max_w \sum_{i=1}^n \ln p(y_i | x_i; w) = \arg\min_w \sum_{i=1}^n (f_w(x_i) - y_i)^2. \end{aligned}$$

Bayesian approach

Assume that w is random with a prior distribution density p_W. Assume also that $p_{X|W}(x|w) = p_X(x)$ and (x_i, y_i) are conditionally independent, given w.
Le us find w^*, giving maximum a posteriori probability:

$$
\begin{aligned}
w^* &\in \arg\max_w p(w|s) = \arg\max_w \frac{p(s|w)p(w)}{p(s)} \\
&= \arg\max_w (\ln p(s|w) + \ln p(w)) \\
&= \arg\max_w \left(\sum_{i=1}^{n} \ln p(x_i, y_i|w) + \ln p(w) \right) \\
p(x_i, y_i|w) &= p(y_i|x_i, w)p(x_i|w) = p(y_i|x_i, w)p(x_i) \\
w^* &= \arg\max_w \left(\sum_{i=1}^{n} \ln p(y_i|x_i, w) + \ln p(w) \right) \\
&= \arg\max_w \left(-\frac{1}{2\sigma^2} \sum_{i=1}^{n} (f_w(x_i) - y_i)^2 + \ln p(w) \right).
\end{aligned}
$$

For the Gaussian prior:

$$p(w) = \frac{1}{(2\pi\tau^2)^{d/2}} \prod_{i=1}^{d} e^{-w_i^2/(2\tau^2)}$$

we get the regularized least squares problem (*ridge* regression):

$$\sum_{i=1}^{n} (f_w(x_i) - y_i)^2 + \frac{\sigma^2}{\tau^2} \sum_{i=1}^{d} w_i^2 \to \min_w .$$

For the Laplace prior:

$$p(w) = \frac{1}{(2b)^d} \prod_{i=1}^{d} e^{-|w_i|/b}$$

we get the *lasso* regularization:

$$\sum_{i=1}^{n} (f_w(x_i) - y_i)^2 + \frac{1}{b} \sum_{i=1}^{d} |w_i| \to \min_w .$$

Logistic regression

This is a classification model. Consider first the binary classification task: $\mathcal{Y} = \{0, 1\}$ or $\mathcal{Y} = \{0, 1\}$. We model the conditional probability of 1:

$$\mathsf{P}(Y = 1|X = x) = p_{Y|X}(y|x) \approx \sigma(\langle \beta, x \rangle), \quad \sigma = \frac{e^x}{1 + e^x},$$

σ is called the sigmoid function $(x_0 = 1, \beta = (\beta_0, \beta'))$.
MLE:

$$\begin{aligned}\widehat{\beta} &= \arg\max_\beta \prod_{i=1}^n p_{X,Y}(x_i, y_i; \beta) = \arg\max_\beta \prod_{i=1}^n p_{Y|X}(y_i|x_i; \beta) p_X(x_i) \\ &= \arg\max_\beta \sum_{i=1}^n \ln p_{Y|X}(y_i|x_i; \beta).\end{aligned}$$

$\mathcal{Y} = \{0, 1\}$:

$$-\sum_{i=1}^{n} \ln p_{Y|X}(y_i|x_i;\beta) = -\sum_{i:y_i=1} \ln p_{Y|X}(1|x_i;\beta) - \sum_{i:y_i=0} \ln p_{Y|X}(0|x_i;\beta)$$

$$= -\sum_{i=1}^{n} [y_i \ln \sigma(\langle\beta, x_i\rangle) + (1 - y_i) \ln(1 - \sigma(\langle\beta, x_i\rangle))] \to \min_{\beta}.$$

$\mathcal{Y} = \{-1, 1\}$:

$$p_{Y|X}(1|x_i;\beta) = \sigma(\langle\beta, x_i\rangle) = \sigma(y_i\langle\beta, x_i\rangle), \quad y_i = 1,$$
$$p_{Y|X}(-1|x_i;\beta) = 1 - \sigma(\langle\beta, x_i\rangle) = \sigma(-\langle\beta, x_i\rangle) = \sigma(y_i\langle\beta, x_i\rangle), \quad y_i = -1,$$

$$-\sum_{i=1}^{n} \ln p_{Y|X}(y_i|x_i;\beta) = -\sum_{i=1}^{n} \ln \sigma(y_i\langle\beta, x_i\rangle)$$
$$= \sum_{i=1}^{n} \ln(1 + e^{-y_i\langle\beta, x_i\rangle}) \to \min_{\beta}.$$

Multiclass case

Let $\sigma : \mathbb{R}^K \mapsto [0,1]^K$ be the softmax function:

$$\sigma_k(z) = \frac{e^{z_k}}{\sum_{j=1}^K e^{z_j}}$$

Multiclass logistic regression: $\mathcal{Y} = \{1, \ldots, K\}$,

$$p_{Y|X}(k|x;\beta) = \sigma_k(\langle \beta_k, x\rangle),$$

$$\text{MLE:} \quad \arg\max_\beta \prod_{i=1}^n p_{X,Y}(x_i, y_i; \beta) = \arg\max_\beta \prod_{i=1}^n p_{Y|X}(y_i|x_i;\beta) p_X(x_i)$$

$$= \arg\max_\beta \prod_{i=1}^n p_{Y|X}(y_i|x_i;\beta) = \arg\max_\beta \prod_{i=1}^n \prod_{k=1}^K p_{Y|X}^{\nu_{ik}}(k|x_i;\beta)$$

$$= \arg\max_\beta \sum_{i=1}^n \sum_{k=1}^K \nu_{ik} \ln \sigma_k(\langle \beta_k, x_i\rangle), \quad \nu_{ik} = I_{\{y_i = k\}}.$$

Kullback-Leibler divergence (relative entropy)

$$D(\mathsf{P}||\mathsf{Q}) = \int \ln \frac{d\mathsf{P}}{d\mathsf{Q}}\, d\mathsf{P} = \int \frac{d\mathsf{P}}{d\mathsf{Q}} \ln \frac{d\mathsf{P}}{d\mathsf{Q}}\, d\mathsf{Q}, \quad Q \ll P.$$

Discrete distributions:

$$D(\mathsf{P}||\mathsf{Q}) = \sum_x p(x) \ln \frac{p(x)}{q(x)}.$$

Continuous distributions:

$$D(\mathsf{P}||\mathsf{Q}) = \int p(x) \ln \frac{p(x)}{q(x)}\, dx.$$

KL divergence is a useful ”distance” from P to Q. It satisfies the Gibbs inequality:

$$D(\mathsf{P}||\mathsf{Q}) \geq 0,$$

but is neither symmetric no satisfies the triangle inequality.

P: true model, Q: approximation. Important example: P is the empirical distribution:

$$\mathsf{P}_S = \frac{1}{n}\sum_{i=1}^{n}\delta_{x_i},$$

Q comes from a parametric family: $q(x;\theta)$. Best parameter (P is discrete):

$$\widehat{\theta} \in \arg\min_{\theta} D(\mathsf{P}_S||\mathsf{Q}_\theta) = \arg\min_{\theta}\sum_{x} p_S(x)\ln\frac{p(x)}{q(x;\theta)}$$

$$= \arg\max_{\theta}\sum_{x} p_S(x)\ln q(x;\theta) = \arg\max_{\theta}\sum_{i=1}^{n}\ln q(x_i;\theta) : \quad \text{MLE.}$$

Let X be a random variable with values in a finite set $\mathcal{X}$. Distance from P_X to the uniform discrete distribution:

$$\begin{aligned} D(\mathsf{P}_X||U) &= \sum_x p(x) \ln \frac{p(x)}{1/|\mathcal{X}|} = \sum_x p(x) \ln p(x) + \sum_x p(x) \ln |\mathcal{X}| \\ &= \ln |\mathcal{X}| - H(X). \end{aligned}$$

The quantity

$$H(X) = H(\mathsf{P}_X) = H(p) := -\sum_x p(x) \ln p(x)$$

is called the entropy of X. We have

$$0 \leq H(X) \leq \ln |\mathcal{X}|,$$

$$H(X) = \begin{cases} 0, & X \text{ is deterministic}, \\ \ln |\mathcal{X}|, & X \text{ is uniform}. \end{cases}$$

Entropy of a distribution is a measure of its uncertainty. Larger the entropy, closer the distribution to the uniform one.

$$D(P||Q) = \sum_x p(x)\ln p(x) - \sum_x p(x)\ln q(x) = H(p,q) - H(p).$$

The quantity

$$H(p,q) = -\sum_x p(x)\ln q(x)$$

is called the cross-entropy of p and q. It is the negative expected log-likelihood of q when data come from p. By the Gibbs inequality,

$$H(p,q) \geq H(p).$$

The larger the cross-entropy of p and q, the larger the KL distance from p to q.

The KL distance from p to q equals to the difference of expected log-likelihood of p when data come from p (negative entropy) and expected log-likelihood of q when data come from p (negative cross entropy).

MLE is equivalent to the minimization of the cross entropy of the empirical distribution of data p_S and the model q_θ:

$$\begin{aligned}\widehat{\theta} &\in \arg\max_\theta \sum_{i=1}^n \ln q(x_i;\theta) = \arg\min_\theta \left(-\frac{1}{n} \sum_{i=1}^n \ln q(x_i;\theta) \right) \\ &= \arg\min_\theta H(p_S, q_\theta).\end{aligned}$$

MLE for supervised learning

$p_{X,Y}$ is a model:

$$
\begin{aligned}
\widehat{\theta} &= \arg\max_{\theta} \prod_{i=1}^{n} p_{X,Y}(x_i, y_i; \theta) = \arg\max_{\theta} \prod_{i=1}^{n} p_{Y|X}(y_i|x_i;\theta) p_X(x_i) \\
&= \arg\max_{\theta} \sum_{i=1}^{n} \ln p_{Y|X}(y_i|x_i;\theta) = \arg\min_{\theta} \sum_{i=1}^{n} H(\delta_{y_i}(\cdot)|p_{Y|X}(\cdot|x_i;\theta)) \\
&= \arg\min_{\theta} \sum_{i=1}^{n} D(\delta_{y_i}(\cdot)||p_{Y|X}(\cdot|x_i;\theta)) \\
&= \arg\min_{\theta} \sum_{i=1}^{n} D(p_S(\cdot|x_i)||p_{Y|X}(\cdot|x_i;\theta)) \\
&= \arg\min_{\theta} \mathsf{E}_{x\in p_S} D(p_S(\cdot|x)||p_{Y|X}(\cdot|x;\theta)) \\
&= \arg\min_{\theta} \mathsf{E}_{x\in p_S} H(p_S(\cdot|x)|p_{Y|X}(\cdot|x;\theta))
\end{aligned}
$$

Bayesian approach to regularization of the logistic regression

Maximum a posteriori probability (MAP):

$$\widehat{\beta} = \arg\max_{\theta} \left(\sum_{i=1}^{n} \ln p(y_i | x_i, \beta) + \ln p(\beta) \right).$$

For binary classification with $\mathcal{Y} = \{-1, 1\}$ and the Gaussian prior

$$p(w) = \frac{1}{(2\pi\tau^2)^{d/2}} \prod_{i=1}^{d} e^{-\beta_i^2/(2\tau^2)}$$

we get

$$\sum_{i=1}^{n} \sigma(\langle \beta, x_i \rangle) - \frac{1}{2\tau^2} \|\beta\|_2^2 \to \max_{\beta},$$

$$\sum_{i=1}^{n} \ln(1 + e^{-y_i \langle \beta, x_i \rangle}) + \frac{1}{2\tau^2} \|\beta\|_2^2 \to \min_{\beta}.$$