Linear Models for Regression

CSci 5525: Advanced Machine Learning

Instructor: Nicholas Johnson

September 7, 2023

Announcements

- HW0 posted last lecture (due next Tue, Sept. 12)
- HW1 will be posted next Tue

Problem

Suppose you work at a restaurant and want to predict how much a customer will tip. You are given the following data consisting of the total bill amount and the tip added for each customer.

Total Bill	Tip
16.99	1.01
10.34	1.66
21.01	3.50
23.68	3.31
24.59	3.61
25.29	4.71
8.77	2.00

How should we predict the tip?

Problem

Suppose you work at a restaurant and want to predict how much a customer will tip. You are given the following data consisting of the total bill amount and the tip added for each customer.

Total Bill	Tip
16.99	1.01
10.34	1.66
21.01	3.50
23.68	3.31
24.59	3.61
25.29	4.71
8.77	2.00

How should we predict the tip?

One heuristic is to predict the average tip: \$2.83.

Problem

Suppose you work at a restaurant and want to predict how much a customer will tip. You are given the following data consisting of the total bill amount and the tip added for each customer.

Total Bill	Tip
16.99	1.01
10.34	1.66
21.01	3.50
23.68	3.31
24.59	3.61
25.29	4.71
8.77	2.00

How should we predict the tip?

One heuristic is to predict the average tip: \$2.83.

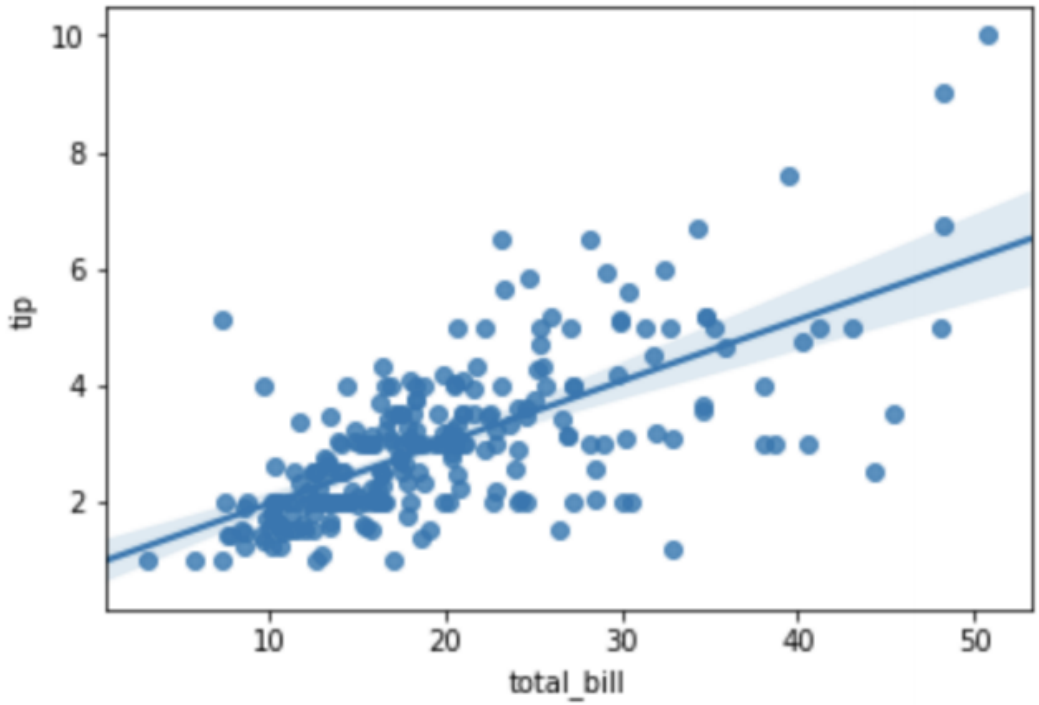
Can we do better?

Regression

- Dataset: $\mathcal{D} = \{(\text{total bill}_i, \text{tip}_i)\}_{i=1}^n = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$
- Assume data are independent and identically distributed (iid)
  - What does this mean?
- Features $\mathbf{x}_i$ and targets y_i
- Supervised learning problem
  - $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^d$, $y \in \mathcal{Y} \subset \mathbb{R}$ (regression)
- Goal: find prediction function $f : \mathcal{X} \to \mathcal{Y}$

Linear Functions

- Choose hypothesis (prediction function) class $\mathcal{C}$ to be linear functions
- $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$ (we often write $\mathbf{x} = [\mathbf{x}; 1]$)
- Many functions to choose from:



Which function should we choose?

Two Approaches

Empirical Risk Minimization (ERM)

- Pick class of predictors $\mathcal{C}$ (linear in this lecture)
- Pick loss function $\ell(\cdot)$
- Minimize empirical risk over model/parameters

Maximum Likelihood Estimation (MLE)

- Choose statistical model (conditional distribution)
- Maximize likelihood function

Loss Functions

- Learning is often based on *minimizing expected loss*
- 0/1 Loss: $L(f, \mathbf{x}, y) = \mathbb{1}_{[f(\mathbf{x}) \neq y]}$, expected loss

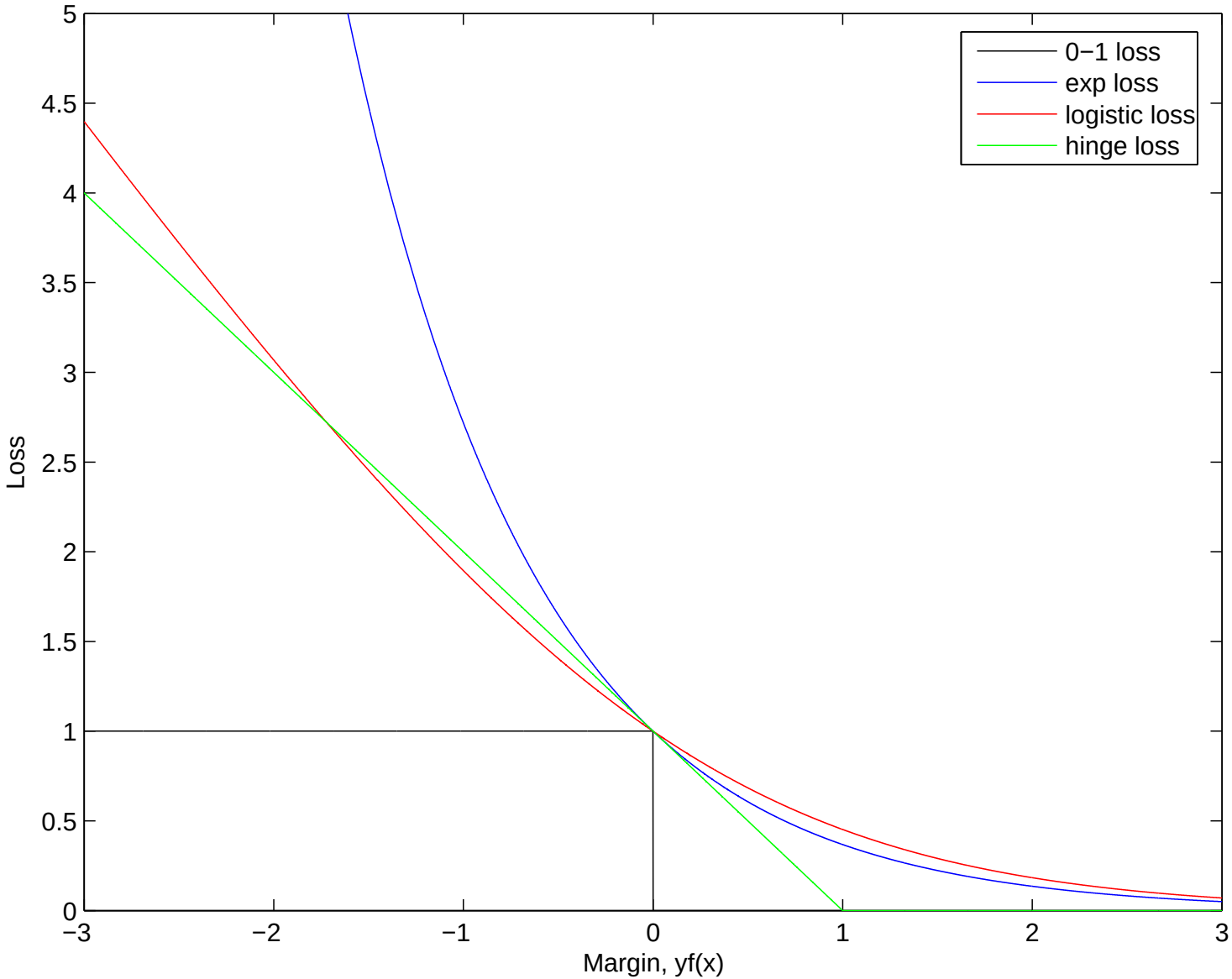
$$\mathbb{E}[L(f, \mathbf{x}, y)] = \mathbb{E}[\mathbb{1}_{[f(\mathbf{x}) \neq y]}] = P(f(\mathbf{x}) \neq y)$$

- Squared Loss: $L(f, \mathbf{x}, y) = (y - f(\mathbf{x}))^2$
- Hinge Loss:

$$L(f, \mathbf{x}, y) = \max(0, 1 - yf(\mathbf{x})) = \begin{cases} 1 - yf(\mathbf{x}) & \text{if } yf(\mathbf{x}) < 1, \\ 0 & \text{otherwise.} \end{cases}$$

- Exponential Loss: $L(f, \mathbf{x}, y) = \exp(-yf(\mathbf{x}))$
- Logistic Loss: $L(f, \mathbf{x}, y) = \log(1 + \exp(-yf(\mathbf{x})))$

Loss Functions



Approach 1: ERM

- Ideally want to learn function which minimizes risk

$$\mathcal{R}(f) = \mathbb{E}_{(X,Y)\sim D}[\ell(Y, f(X))]$$

  where D is unknown distribution

- Instead minimize empirical risk ($(\mathbf{x}_i, y_i) \in \mathcal{D}$)

$$\hat{\mathcal{R}}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(\mathbf{x}_i))$$

Approach 1: ERM

- Loss function: squared loss for prediction $\hat{y} = f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$

$$\ell(y, \hat{y}) = (y - \hat{y})^2$$

- Minimize least squares empirical risk:

$$\hat{\mathcal{R}}(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(\mathbf{x}_i)) = \frac{1}{n} \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i))^2$$

- For linear functions, find $\mathbf{w} \in \mathbb{R}^d$ such that

$$\text{argmin}_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} (y_i - \mathbf{w}^\top \mathbf{x}_i)^2$$

Least Squares Solution

- Design matrix:

$$\mathbf{X} = \begin{bmatrix} \leftarrow & \mathbf{x}_1^\top & \rightarrow \\ & \vdots & \\ \leftarrow & \mathbf{x}_n^\top & \rightarrow \end{bmatrix}$$

- Response vector:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

- Empirical risk can be written as

$$\hat{\mathcal{R}}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbf{w}^\top \mathbf{x}_i)^2 = \frac{1}{n}\|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2$$

Least Squares Solution

- Rescaling does not change solution:

$$\mathbf{w}^* = \text{argmin}_{\mathbf{w} \in \mathbb{R}^d} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2$$

- Necessary condition for $\mathbf{w}$ to be minimizer of $\hat{\mathcal{R}}$ is that it needs to be a stationary point: $\nabla \hat{\mathcal{R}}(\mathbf{w}) = 0$
- This gives the condition: $(\mathbf{X}^\top \mathbf{X})\mathbf{w} = \mathbf{X}^\top \mathbf{y}$
- If $\mathbf{X}$ is full-rank then we can invert so: $\mathbf{w}^* = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$
- Otherwise, use pseudoinverse (can compute it via SVD)

Approach 2: MLE

- Statistical model:

$$y_i = \mathbf{w}^\top \mathbf{x}_i + \epsilon_i \qquad \text{where } \epsilon_i \sim N(0, \sigma^2)\ \forall i$$

- The distribution of y_i given $\mathbf{x}_i$ is:

$$y_i|\mathbf{x}_i \sim N(\mathbf{w}^\top \mathbf{x}_i, \sigma^2)$$

$$\Rightarrow P(y_i|\mathbf{x}_i, \mathbf{w}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(\mathbf{w}^\top \mathbf{x}_i - y_i)^2}{2\sigma^2}\right\}$$

Conditional Distribution

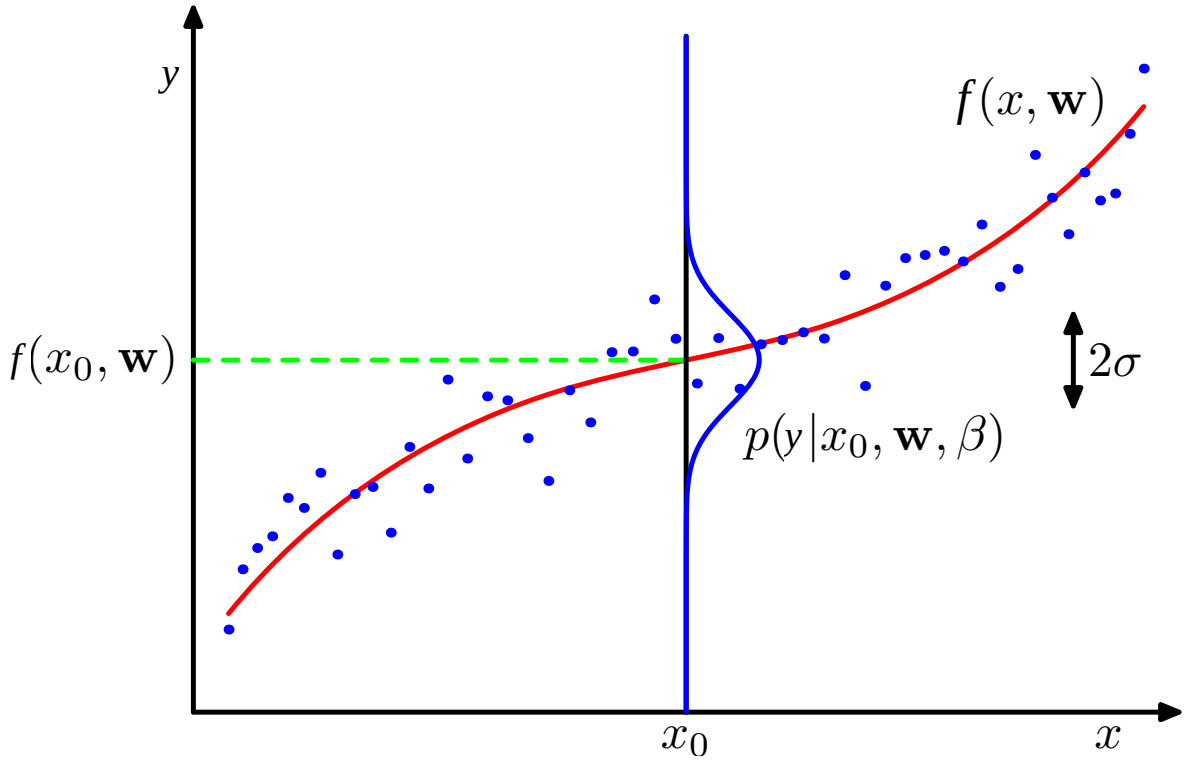


Figure: Fig 1.16, p.29 from PRML

- Red curve is unknown polynomial function
- Blue circles are data points from dataset
- Mean of Gaussian conditional distribution is given by polynomial function and precision is $\beta = \sigma^2$

A Statistical View

- Maximum likelihood estimation (MLE) aims to maximize:

$$P(\text{observed data}|\text{model parameters}) = P(y_1, \mathbf{x}_1, \ldots, y_n, \mathbf{x}_n|\mathbf{w})$$

MLE

$$
\begin{aligned}
\mathbf{w} &= \text{argmax}_{\mathbf{w}} P(y_1, \mathbf{x}_1, \ldots, y_n, \mathbf{x}_n | \mathbf{w}) \\
&= \text{argmax}_{\mathbf{w}} \prod_{i=1}^{n} P(y_i, \mathbf{x}_i | \mathbf{w}) \\
&= \text{argmax}_{\mathbf{w}} \prod_{i=1}^{n} P(y_i | \mathbf{x}_i, \mathbf{w}) P(\mathbf{x}_i | \mathbf{w}) \\
&= \text{argmax}_{\mathbf{w}} \prod_{i=1}^{n} P(y_i | \mathbf{x}_i, \mathbf{w}) P(\mathbf{x}_i) \\
&= \text{argmax}_{\mathbf{w}} \prod_{i=1}^{n} P(y_i | \mathbf{x}_i, \mathbf{w}) \\
&= \text{argmax}_{\mathbf{w}} \sum_{i=1}^{n} \log P(y_i | \mathbf{x}_i, \mathbf{w})
\end{aligned}
$$

MLE

$$
\begin{aligned}
\mathbf{w} &= \text{argmax}_{\mathbf{w}} P(y_1, \mathbf{x}_1, \ldots, y_n, \mathbf{x}_n | \mathbf{w}) \\
&= \text{argmax}_{\mathbf{w}} \prod_{i=1}^{n} P(y_i, \mathbf{x}_i | \mathbf{w}) && \text{(Independence)} \\
&= \text{argmax}_{\mathbf{w}} \prod_{i=1}^{n} P(y_i | \mathbf{x}_i, \mathbf{w}) P(\mathbf{x}_i | \mathbf{w}) && \text{(Chain rule)} \\
&= \text{argmax}_{\mathbf{w}} \prod_{i=1}^{n} P(y_i | \mathbf{x}_i, \mathbf{w}) P(\mathbf{x}_i) && (\mathbf{x}_i \text{ independent of } \mathbf{w}) \\
&= \text{argmax}_{\mathbf{w}} \prod_{i=1}^{n} P(y_i | \mathbf{x}_i, \mathbf{w}) && (P(\mathbf{x}_i) \text{ does not depend on } \mathbf{w}) \\
&= \text{argmax}_{\mathbf{w}} \sum_{i=1}^{n} \log P(y_i | \mathbf{x}_i, \mathbf{w}) && \text{(log is a monotonic function)}
\end{aligned}
$$

MLE (cont.)

$$= \text{argmax}_{\mathbf{w}} \sum_{i=1}^{n} \log P(y_i|\mathbf{x}_i, \mathbf{w}) \qquad \text{(log is a monotonic function)}$$

$$= \text{argmax}_{\mathbf{w}} \sum_{i=1}^{n} \log \frac{1}{\sqrt{2\pi\sigma^2}} + \log \exp\left\{-\frac{(\mathbf{w}^\top \mathbf{x}_i - y_i)^2}{2\sigma^2}\right\}$$

(Plugging in Gaussian distribution)

$$= \text{argmax}_{\mathbf{w}} -\frac{1}{2\sigma^2} \sum_{i=1}^{n} (\mathbf{w}^\top \mathbf{x}_i - y_i)^2$$

(First term is a constant and $\log(\exp(z)) = z$)

$$= \text{argmin}_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} (\mathbf{w}^\top \mathbf{x}_i - y_i)^2$$

(Equivalent to minimizing least squares risk)

Structural Risk Minimization (SRM)

- SRM similar to ERM but takes into account model complexity
- SRM balances fitting to training data and model complexity
- Typical problem:

$$\text{argmin}_\theta \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \hat{y}_i) + \lambda R(\theta)$$

- First term is loss on training data
- Second term measures model complexity (i.e., regularizer)

SRM

- SRM recipe:
  - Pick class of predictors $\mathcal{C}$ (linear in this lecture)
  - Pick loss function $\ell(\cdot)$
  - Pick regularizer $R(\cdot)$
  - Minimize structural risk over model/parameters

Ridge Regression

- Predictor class $\mathcal{C}$ = linear functions
- Loss: squared error $\ell(y, \mathbf{w}^\top \mathbf{x}) = (y - \mathbf{w}^\top \mathbf{x})^2$
- Regularizer: $R(\mathbf{w}) = \|\mathbf{w}\|^2 = \mathbf{w}^\top \mathbf{w} = \sum_i (w(i))^2$
- Structural risk minimization:

$$\mathbf{w}^* = \text{argmin}_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} (y_i - \mathbf{w}^\top \mathbf{x}_i)^2 + \lambda \|\mathbf{w}\|^2$$
$$= \text{argmin}_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2 + \lambda \|\mathbf{w}\|^2$$

Ridge Regression

- Solve similarly as least squares (minimizer is stationary point)

$$\mathbf{w}^* = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{y}$$

  where $\mathbf{I}$ is $d \times d$ identity matrix
- Solution always unique even if $\mathbf{X}$ is not full rank (e.g., $n < p$)
- Regularizer $\lambda \|\mathbf{w}\|^2$ encourages "shorter" solutions
- λ manages tradeoff between fitting to data and magnitude of $\mathbf{w}$ (model complexity)

SRM

- Why do we care about model complexity (or making $\mathbf{w}$ small)?
- Bounding model complexity can prevent overfitting - model has small training error but large test error
- Setting λ can be tricky
  - Too small and we can overfit
  - Too large and we can underfit
  - Typically λ chosen via cross validation

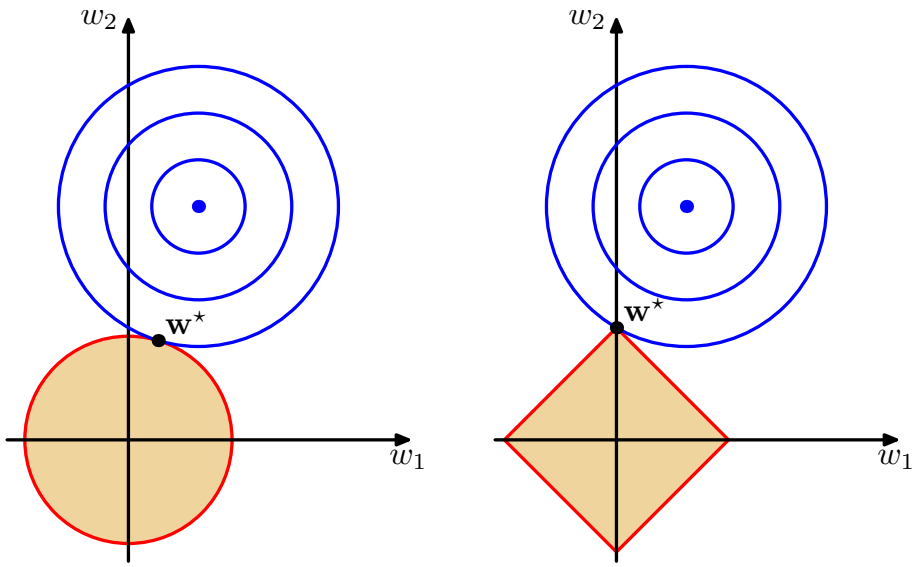
Lasso

- Predictor class $\mathcal{C}$ = linear functions
- Loss: squared error $\ell(y, \mathbf{w}^\top \mathbf{x}) = (y - \mathbf{w}^\top \mathbf{x})^2$
- Regularizer: $R(\mathbf{w}) = \|\mathbf{w}\|_1 = \sum_i |w(i)|$
- Structural risk minimization:

$$\mathbf{w}^* = \text{argmin}_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2 + \lambda \|\mathbf{w}\|_1$$

- Lasso encourages sparse solutions, often used when $n \ll p$
- No closed-form solution

Regularized Least Squares



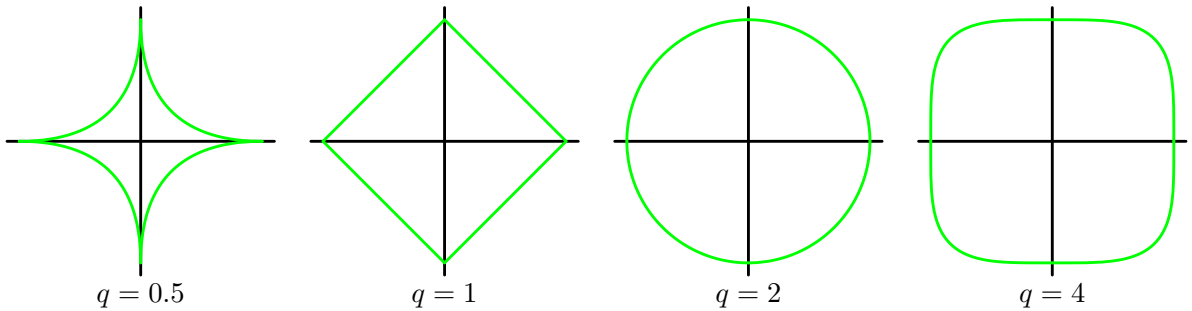
- Regression with L_2 regularization: Ridge Regression

$$\text{argmin}_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2 + \lambda\|\mathbf{w}\|^2$$

- Regression with L_1 regularization: Lasso

$$\text{argmin}_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2 + \lambda\|\mathbf{w}\|_1$$

Regularized Least Squares



General classes of regularizers:

$$\mathbf{w}^* = \text{argmin}_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2 + \lambda \sum_i |w(i)|^q$$

Maximum A Posteriori (MAP) Estimation

- MAP estimation solves $P(\text{model parameter}|\text{observed data})$
- Statistical model:

$$y_i = \mathbf{w}^\top \mathbf{x}_i + \epsilon_i \qquad \text{where } \epsilon_i \sim N(0, \sigma^2)\ \forall i$$

- The distribution of y_i given $\mathbf{x}_i$ is:

$$y_i|\mathbf{x}_i \sim N(\mathbf{w}^\top \mathbf{x}_i, \sigma^2)$$

$$\Rightarrow P(y_i|\mathbf{x}_i, \mathbf{w}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\mathbf{w}^\top \mathbf{x}_i - y_i)^2}{2\sigma^2}\right)$$

- A priori distribution of $\mathbf{w}$ is $P(\mathbf{w}) = \frac{1}{\sqrt{2\pi\tau^2}} \exp\left(\frac{-\mathbf{w}^\top \mathbf{w}}{2\tau^2}\right)$

Maximum A Posteriori (MAP) Estimation

$$
\begin{aligned}
\mathbf{w}^* &= \text{argmax}_{\mathbf{w}} \, P(\mathbf{w}|y_1, \mathbf{x}_1, \ldots, y_n, \mathbf{x}_n) \\
&= \text{argmax}_{\mathbf{w}} \, \frac{P(y_1, \mathbf{x}_1, \ldots, y_n, \mathbf{x}_n|\mathbf{w})P(\mathbf{w})}{P(y_1, \mathbf{x}_1, \ldots, y_n, \mathbf{x}_n)} \\
&= \text{argmax}_{\mathbf{w}} \, P(y_1, \mathbf{x}_1, \ldots, y_n, \mathbf{x}_n|\mathbf{w})P(\mathbf{w}) \\
&= \text{argmax}_{\mathbf{w}} \left[\prod_{i=1}^{n} P(y_i, \mathbf{x}_i|\mathbf{w}) \right] P(\mathbf{w}) \\
&= \text{argmax}_{\mathbf{w}} \left[\prod_{i=1}^{n} P(y_i|\mathbf{x}_i, \mathbf{w})P(\mathbf{x}_i|\mathbf{w}) \right] P(\mathbf{w}) \\
&= \text{argmax}_{\mathbf{w}} \left[\prod_{i=1}^{n} P(y_i|\mathbf{x}_i, \mathbf{w})P(\mathbf{x}_i) \right] P(\mathbf{w})
\end{aligned}
$$

Maximum A Posteriori (MAP) Estimation

$$= \text{argmax}_{\mathbf{w}} \left[\prod_{i=1}^{n} P(y_i|\mathbf{x}_i, \mathbf{w}) P(\mathbf{x}_i) \right] P(\mathbf{w})$$

$$= \text{argmax}_{\mathbf{w}} \left[\prod_{i=1}^{n} P(y_i|\mathbf{x}_i, \mathbf{w}) \right] P(\mathbf{w})$$

$$= \text{argmax}_{\mathbf{w}} \sum_{i=1}^{n} \log P(y_i|\mathbf{x}_i, \mathbf{w}) + \log P(\mathbf{w})$$

$$= \text{argmin}_{\mathbf{w}} \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \mathbf{w}^\top \mathbf{x}_i)^2 + \frac{1}{2\tau^2} \mathbf{w}^\top \mathbf{w}$$

$$= \text{argmin}_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} (y_i - \mathbf{w}^\top \mathbf{x}_i)^2 + \lambda \|\mathbf{w}\|^2$$

(Equivalent to ridge regression)

Tuning Hyperparameters

- The regularization parameter λ balances fitting to the data and model complexity
- λ is a hyperparameter
- How do we choose the best λ?
- Use cross validation:
  - Training set: learn predictor $\hat{f}$ by fitting this dataset
  - Validation set: tune hyperparameters; use loss on this set to find the best hyperparameter
  - Test set: assess **risk** of model: $\mathcal{R}(f) = \mathbb{E}_{(X,Y)\sim P}[\ell(Y, f(X))]$
- We want to predict well on future data; goal is to find $\hat{f}$ that minimizes risk (instead of empirical risk on training set)