THE UNIVERSITY OF CHICAGO

STAT 37710 / CMSC 35400 / CAAM 37710 Machine Learning

Linear regression and its regularization

Cong Ma

Review: statistical framework for regression

- Fundamental assumption: Our data set is generated ***independently and identically distributed*** (*iid*) from some unknown distribution *P*

$$(\mathbf{x}_i, y_i) \sim P(\mathbf{X}, Y)$$

- Our goal is to minimize the ***expected error (true risk)*** under *P*

$$R(h) = \int P((x), y)\ell(y; h(\mathbf{x}))d\mathbf{x}dy = \mathbb{E}_{\mathbf{x},y}\left[\ell(y; h(\mathbf{x})\right]$$

- In particular, we take the loss function to be the squared loss

Empirical risk minimization (model-free)

- Choose a model class $\mathcal{H}$

$$\hat{h} = \hat{h}_D = \arg\min_{h \in \mathcal{H}} \sum_{(\mathbf{x}, y) \in D} (y - h(\mathbf{x}))^2$$

- When $\mathcal{H}$ is linear, we have

$$\arg\min_{\mathbf{w}} \sum_{i=1}^{n} (y_i - \mathbf{w}^T \mathbf{x}_i)^2$$

A second (model-based) view on linear regression

- Based on well-specified parametric statistical model

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

- Based on a classical principle: maximum likelihood estimation

An interlude: MLE for parameter estimation

- Parameter estimation:
  - We observe $z_i \overset{\text{iid}}{\sim} p_\theta,\ \theta \in \Theta$, and the goal is to determine the θ that produced $\{z_i\}_{i=1}^n$.
- Likelihood function $p(z \mid \theta)$
- Maximizing the likelihood function is equivalent to maximizing the log-likelihood function

$$\hat{\theta} = \arg\max_{\theta \in \Theta} p(z \mid \theta)$$

Example: estimating mean of Bernoulli

- We toss a coin n times
- Observe following outcomes

$$\mathscr{D} = \{y_i\}_{i=1}^{n}, y_i \in \{-1,1\}$$ ($y_i = 1$ means head in i's trial, -1 means tail)

- How do you estimate the bias of the coin? $\frac{1}{n}\sum_{i=1}^{n} 1\{y_i = 1\}$

Statistical model for coin tossing

- Each outcome is independently sampled from $\text{Bern}(\theta^\star)$
- What's the probability of observing the data?

$$P(\mathcal{D} \mid \theta) = \theta^{n_1}(1-\theta)^{n-n_1}$$

- MLE principle: Find θ that **maximizes the likelihood** of the data

$$\hat{\theta}_{mle} = \arg\max_{\theta \in [0,1]} P(\mathcal{D} \mid \theta)$$

Computing MLE for coin tossing

Example: Estimate mean of Gaussian

- Data $\mathscr{D} = \{x_i\}_{i=1}^n, x_i \in \mathbb{R}^d$
- Assume they follow $\mathcal{N}(\mu^\star, \boldsymbol{I})$

- How to estimate using MLE?

$$P(\mathscr{D} \mid \mu) = \prod_{i=1}^{n} \frac{1}{\sqrt{(2\pi)^d}} \exp\left(-\frac{1}{2}(x_i - \mu)^\top (x_i - \mu)\right)$$

$$\arg\max_{\mu} \sum_{i=1}^{n} -(x_i - \mu)^\top (x_i - \mu)$$

A probabilistic model for regression

- Consider linear regression. Let's make the statistical assumption that the noise is Gaussian:

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

- Then we can compute the (conditional) likelihood of the data given any candidate model ***w*** as:

Maximum (conditional) likelihood estimation

$$\theta^* = \arg\max_\theta \hat{P}(y_1, \ldots, y_n \mid \mathbf{x}_1, \ldots, \mathbf{x}_n, \theta)$$

- The negative log likelihood is given by

$$L(\mathbf{w}) = -\log P(y_1, \ldots, y_n \mid \mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{w}) = \frac{n}{2}\log(2\pi\sigma^2) + \sum_{i=1}^{n} \frac{(y_i - \mathbf{w}^\top \mathbf{x}_i)^2}{2\sigma^2}$$

- Thus, under the "conditional linear Gaussian" assumption, maximizing the likelihood is equivalent to least squares estimation:

$$\arg\max_{\mathbf{w}} P(y_1, \ldots, y_n \mid \mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{w}) = \arg\min_{\mathbf{w}} \sum_{i=1}^{n} (y_i - \mathbf{w}^\top \mathbf{x}_i)^2$$

Maximum likelihood estimation (MLE)

- MLEs are a very important type of estimator:
  - The MLE is often simple and easy to compute
  - MLEs are invariant under reparameterization (HW)
  - MLEs often have asymptotic optimal properties, e.g.,
    - Consistency
    - Asymptotic efficiency (smallest variance among all “well-behaved” estimators for large n)
    - Asymptotic normality --- allows uncertainty quantification
- All these properties are asymptotic (hold as $n \to \infty$)
  - For finite *n*, we must avoid overfitting! (see *later lecture*)

Computational aspect of linear regression

How to solve the following problem?

$$\arg\min_{\mathbf{w}} \sum_{i=1}^{n} (y_i - \mathbf{w}^T \mathbf{x}_i)^2$$

A detour on optimization

- A (differentiable) loss function

$$f : \mathbb{R}^d \mapsto \mathbb{R}$$

- Our goal is to solve the following optimization problem

$$\underset{x}{\text{minimize}} \quad f(x)$$

Convex functions

- We say a function is convex if the following is true

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y)$$

$$\text{for all } x, y, \text{ and } \lambda \in [0,1]$$

- Examples:

$$f(x) = x^2 \qquad f(x) = \exp(x)$$

First-order optimality

- Claim: If $f : \mathbb{R}^d \mapsto \mathbb{R}$ is a convex and differentiable function, then x is the minimizer of f if and only if

$$\nabla f(x) = 0$$

Matrix notation and normal equations

Define $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ to be the design matrix

$\boldsymbol{y} \in \mathbb{R}^n$ the collection of outcomes

$$f(w) = \|\boldsymbol{X}w - \boldsymbol{y}\|_2^2$$

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

Iterative methods for solving linear regression

- Gradient descent

$$x_{t+1} = x_t - \eta \nabla f(x_t)$$

- Stochastic gradient descent

Ridge regression

- Colinearity
- Stability of estimates

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

- This is equivalent to solving the following opt problem

$$\min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \mathbf{w}^\top \mathbf{x}_i\right)^2 + \lambda \left\|\mathbf{w}\right\|_2^2$$

Benefits of ridge regression

- Strict generalization of linear regression
- When choosing regularization properly, MSE is smaller than linear regression
- The regularization has fundamental connections to the smoothness of the function
- This is a form of inductive bias

Sparse regression (feature selection)

- Goal: learning a sparse linear classifier (i.e., with weight vector **w** containing at most *k* non-zero entries)
- Want to solve:

$$\hat{w} = \arg\min_{\mathbf{w}} \sum_{i=1}^{n} (y_i - \mathbf{w}^\top \mathbf{x}_i)^2 + \lambda \|\mathbf{w}\|_0$$

  - This is a difficult combinatorial optimization problem

- **Key idea**: Replace $\|\mathbf{w}\|_0$ by a more **tractable** term

The “sparsity trick“: convex relaxation

$$\|\mathbf{w}\|_0 \quad \Rightarrow \quad \|\mathbf{w}\|_1$$

Sparse regression: The Lasso

- Ridge regression $\min_{\mathbf{w}} \lambda ||\mathbf{w}||_2^2 + \sum_{i=1}^{n} (y_i - \mathbf{w}^T \mathbf{x}_i)^2$
  - Uses $\|\mathbf{w}\|_2^2$ to control the weights

- Slight modification:
  - replace $\|\mathbf{w}\|_2^2$ by $\|\mathbf{w}\|_1$
  - L1-regularized regression (the LASSO) $\min_{\mathbf{w}} \lambda \|\mathbf{w}\|_1 + \sum_{i=1}^{n} \left(y_i - \mathbf{w}^\top \mathbf{x}_i\right)^2$

- This alternative penalty encourages coefficients to be exactly 0
  - automatic feature selection!

Summary

- What we have learned today:
  - A new perspective on linear regression: MLE of well-specified linear model
  - Computational methods for linear regression
  - Regularized / penalized linear regression