University of Amsterdam
Amsterdam School of Economics

Machine Learning for Econometrics, 2022

Lecture 1: Statistical Learning Framework

Yi He

November 1, 2022

Plan for Today

1. Course Introduction
2. Motivating Example: Support Vector Machines
3. Beyond SVM: Statistical Learning Framework
4. Unifying Classification and Regression

Course Introduction

ML for Econometrics 2022–2023

Our team:

Yi He

Mario Rothfelder

Sander Barendse

- Tuesday lectures and Friday tutorials on campus
- Submit pre-class questions by every Thursday noon.
- Computer labs by my colleagues Mario and Sander

What is Machine Learning . . . for Economists?

Athey S. (2018), "The Impact of Machine Learning on Economics":

> . . ., *machine learning is a field that develops algorithms designed to be applied to datasets, with the main areas of focus being prediction [regression], classification, and clustering or grouping tasks.*

- Statistical Learning Framework: Week 1
- Deep Learning I: Week 2
- Deep Learning II + Guest Lecture (*Fri 18 Nov*): Week 3
- Boosting: Week 4
- Random Forest: Week 5
- Causal inference: Week 6

ML for Econometrics 2022–2023: Assessment

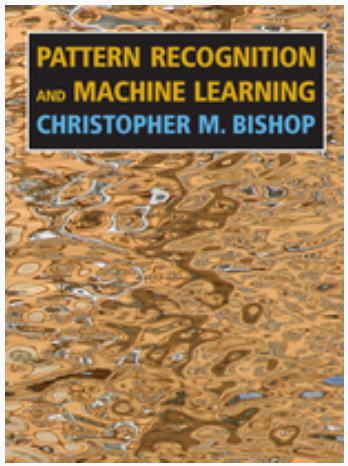
- Computer assignments: 20%+20%=40%
- Final exam, 2 hours: 60%
- Form assignment groups voluntarily until Mon 8 Nov via **Canvas**
- Max 3 students in each group
- On Tue 9 Nov, the remaining students will be randomly assigned.

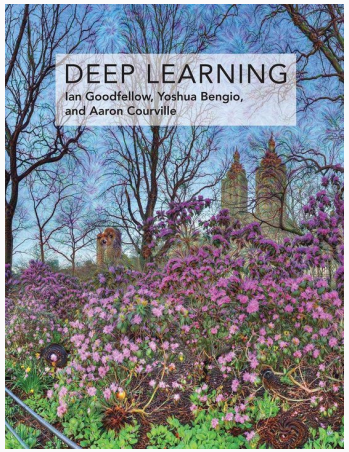
Learning Materials

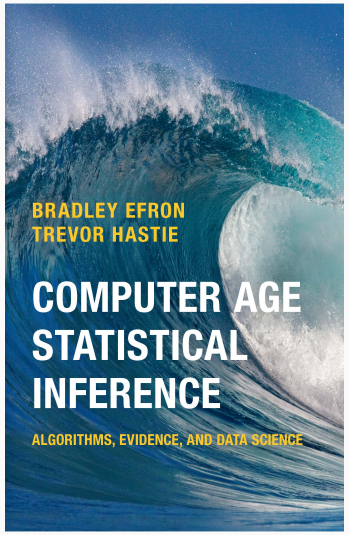
Our main learning materials are in the online reader:

`https://machinelearningtheory.org/`

More reference materials:







- Links for free online reading of the reference books on Canvas

Motivating Example: Support Vector Machines

Optimal Separating Hyperplane

$$S = \{Y_i, X_i : i = 1, \ldots, n\},$$
$$X_i = (X_{i1}, \ldots, X_{id})^T \in \mathbb{R}^d, Y_i \in \{-1, +1\}$$

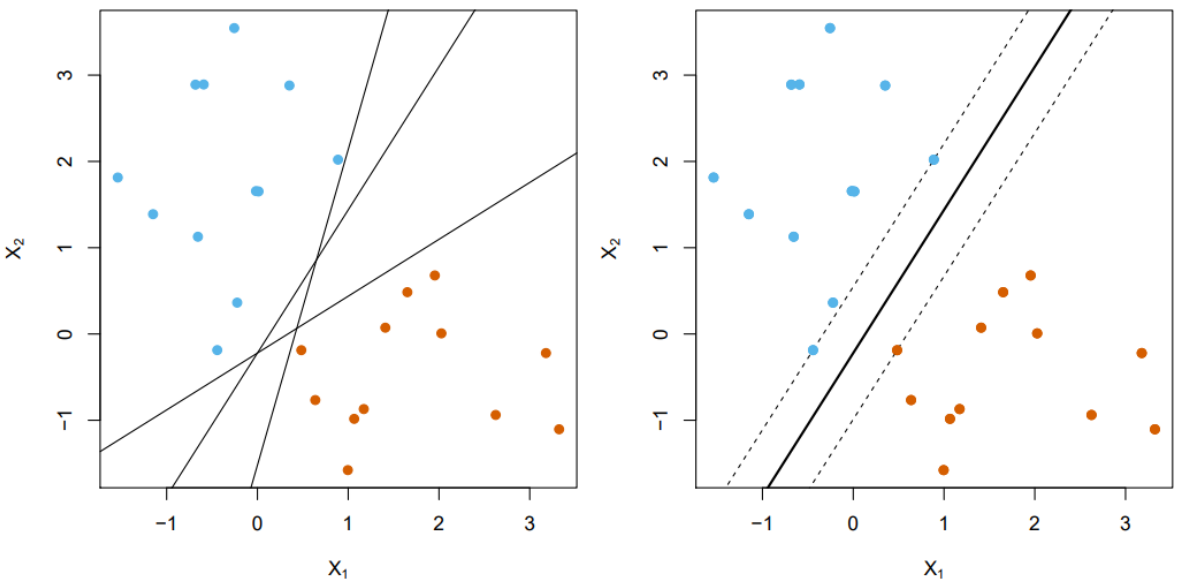


Figure 19.1 of Efron and Hastie (2016): linear discriminant function $f(x) = w^T x + w_0$, decision rule $C(x) = \text{sign}(f(x))$

Signed Distances

Signed distances from points X_i to the decision boundary

$$\{x : w^T x + w_0 = 0, \|w\| = 1\}$$

are given by:

$$w^T X_i + w_0 = \begin{cases} > 0 & \Rightarrow C(X_i) = 1 \\ = 0 & \text{on the boundary}, \\ < 0 & \Rightarrow C(X_i) = -1 \end{cases} \quad i = 1, \ldots, n$$

Signed distances taking into account the classification errors:

$$Y_i \cdot (w^T X_i + w_0) = \begin{cases} > 0 & \Rightarrow C(X_i) = Y_i \\ = 0 & \text{on the boundary}, \\ < 0 & \Rightarrow C(X_i) \neq Y_i \end{cases} \quad i = 1, \ldots, n$$

Hard-Margin Support Vector Machine

Assuming separability, maximizing the margins gives hard-margin SVM [Tute Q1]:

$$\underset{w,w_0,M}{\text{maximize}}\ M$$

$$\text{subject to}\ \underbrace{Y_i \cdot (w^T X_i + w_0) \geq M}_{\text{Beyond Margin}}\ \forall\ i$$

$$\|w\| = 1$$

$$\Leftrightarrow\ \underset{\beta,\beta_0}{\text{minimize}}\ \frac{1}{2}\beta^T\beta$$

$$\text{subject to}\ Y_i(\beta^T X_i + \beta_0) \geq 1\ \forall\ i$$

. . . in the sense that they give the same decision boundary

$$\{x : \widehat{w}^T x + \widehat{w}_0 = 0\} = \{x : \widehat{\beta}^T x + \widehat{\beta}_0 = 0\}$$

Soft-Margin Support Vector Machine

The soft-margin SVM, with some hyperparameter $C > 0$:

$$\begin{aligned} &\underset{\beta,\beta_0,\{\xi_i\}}{\text{minimize}} \ \frac{1}{2}\beta^T\beta + C\sum_{i=1}^{n}\xi_i \\ &\text{subject to } Y_i(\beta^T X_i + \beta_0) \geq \underbrace{1-\xi_i}_{\text{Violating Margins}} \quad \forall i \\ &\xi_i \geq 0 \quad \forall i \end{aligned}$$

Sending $C \to \infty$ gives the hard-margin version.

For each i, the constraints are

$$\begin{cases} \xi_i \geq 1 - Y_i(\beta^T X_i + \beta_0) \\ \xi_i \geq 0 \end{cases} \Leftrightarrow \xi_i \geq \max\{1 - Y_i(\beta^T X_i + \beta_0), 0\}$$

Alternative Formulation of Soft-Margin SVM

Minimizing ξ_i for all i gives that

$$\xi_i = \max\{1 - Y_i(\beta^T X_i + \beta_0), 0\}$$

Plugging in the objective function yields a unconstrained problem:

$$\underset{\beta,\beta_0,\cancel{\{\xi_i\}}}{\text{minimize}} \ \frac{1}{2}\beta^T\beta + C\sum_{i=1}^{n} \underbrace{\max\{1 - Y_i(\beta^T X_i + \beta_0), 0\}}_{\xi_i}$$

Dividing the objective function by the constant nC does NOT change the solution of β, β_0

$$\underset{\beta,\beta_0}{\text{minimize}} \ \lambda\beta^T\beta + \frac{1}{n}\sum_{i=1}^{n} \max\{1 - Y_i(\beta^T X_i + \beta_0), 0\}$$

with $\lambda = \frac{1}{2nC} > 0$.

Hinge Loss

The objective function

$$\lambda\beta^T\beta + \frac{1}{n}\sum_{i=1}^{n}\max\{1 - Y_i(\underbrace{\beta^T X_i + \beta_0}_{f(X_i;\beta,\beta_0)}), 0\}$$

$$=\lambda\beta^T\beta + \frac{1}{n}\sum_{i=1}^{n}\ell_{\mathsf{H}}(f(X_i;\beta,\beta_0), Y_i)$$

where ℓ_{H} is called the *hinge loss* function given by

$$\ell_{\mathsf{H}}(f(x), y) = \max\{1 - y \cdot f(x), 0\}$$

and $f(x;\beta,\beta_0)$ is a candidate linear prediction rule in form of

$$f(x;\beta,\beta_0) = \beta^T x + \beta_0$$

Regularized Estimation

The Lagrange multiplier theorem states that:

$$\underset{\beta,\beta_0}{\text{minimize}}\ \lambda\beta^T\beta + \frac{1}{n}\sum_{i=1}^{n}\ell_{\text{H}}(f(X_i;\beta,\beta_0),Y_i) \tag{1}$$

is equivalent to

$$\underset{\beta,\beta_0}{\text{minimize}}\ \underbrace{\frac{1}{n}\sum_{i=1}^{n}\ell_{\text{H}}(f(X_i;\beta,\beta_0),Y_i)}_{\text{Empirical Risk}}$$

$$\boxed{\text{subject to } \beta^T\beta \leq b} \quad \Rightarrow \quad \text{Regularization}$$

where $b = \widehat{\beta}'\widehat{\beta}$ where $\widehat{\beta}$ is the solution of (1).

Empirical Risk

The empirical hinge risk (= average hinge loss)

$$L_S(f) = \frac{1}{n}\sum_{i=1}^{n} \ell_{\mathsf{H}}(f(X_i; \beta, \beta_0), Y_i)$$

is an estimator of the population hinge risk (= expected hinge loss)

$$L_{\mathcal{D}}(f) = \mathbb{E} L_S(f) = \mathbb{E}\left[\ell_{\mathsf{H}}(f(X; \beta, \beta_0), Y)\right]$$

if Y_i, X_i are i.i.d. observations from the distribution $\mathcal{D}$ of Y, X.

- The regularization improves statistical performance: we will come back to this point later

SVM: What Does It Estimate?

Now consider a general, not necessarily linear, candidate prediction rule $f : \mathbb{R}^d \to \mathbb{R}$ with the hinge risk

$$L_{\mathcal{D}}(f) = \mathbb{E}\left[\ell_{\mathsf{H}}(f(X), Y)\right] = \mathbb{E}\left[\max\{0, 1 - Y \cdot f(X)\}\right].$$

The *ideal* prediction rule f^* minimizing this risk function is

$$\begin{aligned} f^*(x) &= \operatorname{sign}\left(\mathbb{E}[Y|X = x]\right) \\ &= \begin{cases} 1 & \mathbb{P}(Y = 1|X = x) > \mathbb{P}(Y = -1|X = x) \\ -1 & \mathbb{P}(Y = 1|X = x) < \mathbb{P}(Y = -1|X = x) \end{cases} \\ &=: C^{\text{Bayes}}(x) \end{aligned}$$

C^{Bayes} is called the Bayes classifier

In other words, the soft-margin SVM is directly estimating the Bayes classifier $C^{\text{Bayes}}(x)$.

Why Bayes Classifier?

The Bayes classifier $C^{\text{Bayes}}(x)$ minimizes the probability of making classification mistakes:

$$\underset{g}{\text{minimize}}\ \mathbb{P}(g(X) \neq Y) = \underset{g}{\text{minimize}}\ \mathbb{E}[\ell_{0\text{-}1}(g(X), Y)]$$

over the class of classifier $g : \mathbb{R}^d \to \{-1, +1\}$, where

$$\ell_{0\text{-}1}(\widehat{y}, y) = \mathbb{1}[\widehat{y} \neq y]$$

is called the zero-one loss function.

Remark: Linear Rule Is Not Enough

Recall that the soft-margin SVM is directly estimating the Bayes classifier $C^{\text{Bayes}}(x) \in \{-1, 1\}$, which is nonlinear. However, the standard soft-margin SVM uses only linear prediction rules

$$f(x) = \beta^T x + \beta_0.$$

That's why we need to use the kernel tricks (not to be discussed) to generate a richer space

$$\mathcal{F} = \{f : f(x) = \sum_{i=1}^{n} \alpha_i K(x, x_i)\},$$

where $K \in \mathcal{K}$ is a kernel function so we can approximate $C^{\text{Bayes}}(x)$ with $\left\| C^{\text{Bayes}}(x) - f^*(x) \right\|_{\mathcal{K}} < \epsilon$ for a (arbitrarily) small $\epsilon > 0$ for some $f^* \in \mathcal{F}$.

Regularization: Variance Reduction

Recall that we only optimize the empirical risk over a smaller subset in SVM given by

$$\mathcal{H} = \{f(x) = \beta^T x + \beta_0 \in \mathcal{F} : \|\beta\|^2 \leq b\}$$

and our estimator is given by

$$\widehat{f} = \underset{f \in \mathcal{H}}{\text{argmin}}\ L_S(f).$$

The motivation is to reduce variance of $\widehat{f}$ due to the randomness of Y_i, X_i, which increases in general with the worst estimation error of the risk function:

$$\sup_{f \in \mathcal{H}} |L_S(f) - L_{\mathcal{D}}(f)| \ \uparrow \ \text{as} \ \mathcal{H} \ \uparrow .$$

Regularization: Bias Variance Tradeoff

However, shrinking the space $\mathcal{H}$ restricts our choices of candidate prediction rules. The best population risk we can reach is only

$$\inf_{f \in \mathcal{H}} \; L_{\mathcal{D}}(f)$$

so the bias

$$\inf_{f \in \mathcal{H}} \; L_{\mathcal{D}}(f) - \inf_{f \in \mathcal{F}} \; L_{\mathcal{D}}(f) \; \uparrow \text{ as } \mathcal{H} \; \downarrow .$$

In practice, we need to trade-off bias and variance by tuning the hyperparameter λ (and hence b) properly.

Beyond SVM: Statistical Learning Framework

Statistical Learning Problems

Target variable $Y \in \mathcal{Y}$, d features $X \in \mathcal{X} \subset \mathbb{R}^d$

Before observing Y:

We make a prediction $f(X)$ with some prediction rule $f : \mathcal{X} \to \mathcal{Y}$

After observing Y: We quantify the predictive loss by

$$\ell(f(X), Y)$$

where $\ell : \mathcal{Y} \times \mathcal{Y} \to [0, \infty)$ is some loss function specified by the user. A larger loss implies a worse prediction performance, and the loss remains non-negative such that

$$\ell(f(X), Y) \geq \ell(Y, Y) = 0.$$

Our goal is to find the ideal prediction rule that minimizes the population risk function given by

$$L_{\mathcal{D}}(f) = \mathbb{E}[\ell(f(X), Y)], \quad f \in \mathcal{F}.$$

Ideal Prediction Rule

- The regression function

$$\mu(x) = \mathbb{E}[Y \mid X = x]$$

is ideal for the (half) squared loss function

$$\ell(\widehat{y}, y) = \frac{1}{2} \|\widehat{y} - y\|^2$$

- The Bayes classifier

$$C^{\text{Bayes}}(x) = \underset{\mathcal{C}_k, 1 \leq k \leq K}{\operatorname{argmax}} \ \mathbb{P}(Y = \mathcal{C}_k \mid X = x)$$

is ideal for the zero-one loss $\ell_{0\text{-}1}$ among classifiers and hinge loss ℓ_{H} among all real-valued functions as discussed before.

Empirical Risk Minimization

Given a random sample

$$S = \{Y_i, X_i : i = 1, \ldots, n\},$$

the empirical risk function is equal to the average loss given by

$$L_S(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(X_i), Y_i), \quad f \in \mathcal{F}.$$

The empirical risk minimization (ERM) paradigm solves the optimization problem:

$$\text{minimize } L_S(f)$$

$$\boxed{\text{subject to } f \in \mathcal{H}} \quad \Rightarrow \quad \text{Regularization}$$

How to choose $\mathcal{H} \subset \mathcal{F}$?

Penalized Method

- Parameterize $f = f(x; \theta) \in \mathcal{H}$ with parameters $\theta \in \Theta$
- Measure model complexity with some criterion function $C(\theta)$
- Choose $\mathcal{H}$ with a limited complexity such that

$$\mathcal{H} = \{f(\cdot; \theta) \in \mathcal{F} : C(\theta) \leq b,\ \theta \in \Theta\}.$$

This is equivalent to minimize the penalized empirical risk function

$$\text{minimize}\ \ L_S(f) + \lambda \cdot C(\theta)$$

where $\lambda = \lambda(b)$ is a tuning parameter.

Unifying Classification and Regression

One-Hot Encoding

- Classification algorithms generate discrete outputs but regression algorithms generate continuous outputs.
- There is a strong connection between these two tasks:

$$\underbrace{C^{\text{Bayes}}(x)}_{\text{classification}} = \underset{\mathcal{C}_k, 1 \le k \le K}{\text{argmax}} \; \mathbb{P}(Y = \mathcal{C}_k \mid X = x)$$

$$= \underset{\mathcal{C}_k, 1 \le k \le K}{\text{argmax}} \; \underbrace{\mathbb{E}\left[\mathbb{1}[Y = \mathcal{C}_k] \mid X = x\right]}_{\text{regression}}$$

- $(Y^{(1)}, \ldots, Y^{(K)})$ is the one-hot encoding of Y with $Y^{(k)} = \mathbb{1}[Y = \mathcal{C}_k]$.
- We can estimate the multivariate regression function

$$\mu(x) = (\mu_1(x), \ldots, \mu_K(x)), \quad \mu_k(x) = \mathbb{E}[Y^{(k)} \mid X = x]$$

using regression algorithms, and then construct the Bayes classifier.

Softmax Function

- However, the regression function for one-hot encoded target must satisfy the axioms of probability:

$$\mu(x) \in \mathcal{P}_k = \left\{ y \in (0,1)^K : \sum_{k=1}^{K} y_k = 1 \right\}, \quad \forall x \in \mathcal{X}.$$

- Re-parameterize the candidate regression function $f(x) = (f_1(x), \ldots, f_K(x)) \in \mathcal{P}_k$ jointly by

$$f(x) = \begin{pmatrix} f_1(x) \\ \vdots \\ f_K(x) \end{pmatrix} = \boldsymbol{\sigma} \begin{pmatrix} a_1(x) \\ \vdots \\ a_K(x) \end{pmatrix} = \boldsymbol{\sigma}(a(x))$$

$$\boldsymbol{\sigma}(a) = \begin{pmatrix} \boldsymbol{\sigma}(a)_1 \\ \vdots \\ \boldsymbol{\sigma}(a)_K \end{pmatrix}, \boldsymbol{\sigma}(a)_k = \frac{\exp(a_k)}{\sum_{j=1}^{K} \exp(a_j)}.$$

- Note that $a(x) \in \mathbb{R}^K$ but set $a_K(x) \equiv 0$ for identification.

Cross Entropy

- Note that $(Y^{(1)}, \ldots, Y^{(K)}) \in \mathcal{P}_K$ is a distribution
- Candidate regression function $(f_1(x), \ldots, f_K(x)) \in \mathcal{P}_K$ is another distribution
- The cross-entropy loss function $\ell_{\mathsf{CE}} : \mathcal{P}_K \times \mathcal{P}_K \to [0, \infty)$ given by

$$\ell_{\mathsf{CE}}(f(x), y) = \sum_{k=1}^{K} -y_k \cdot \log f_k(x)$$

  measures the 'distance' between these two distributions
- Relax the definition to allow for $y_k \in \{0, 1\}$ in algorithms
- We can show that the true regression function $\mu(x)$ for the one-hot encoded target is the ideal prediction rule.
- It is more specialized than the Euclidean distance induced by the squared loss

$$\ell_2(f(x), y) = \frac{1}{2} \|f(x) - y\|^2 .$$

Unified View of Regression and Classification

- After one-hot encoding, a classification task translates into a regression task.
- One may use soft-max transformation to enforce the axioms of probability, and the cross-entropy loss instead of the squared loss to estimate the multivariate regression function.
- Therefore, other ML methods in this course are primarily for regression tasks but apply to classification tasks.
- The main differences are their choices of candidate prediction rules, loss function, and regularization procedures.