Mathematical techniques in data science

Vu Dinh

Lecture 5: Classification – Logistic regression

February 25th, 2019

Schedule

Week	Chapter
1	Chapter 2: Intro to statistical learning
3	Chapter 4: Classification
4	Chapter 9: Support vector machine and kernels
5, 6	Chapter 3: Linear regression
7	Chapter 8: Tree-based methods + Random forrest
8	
9	Neural network
10	Bootstrap and CV + Bayesian methods + UQ
11	Clustering: K-means $\rightarrow$ Spectral Clustering
12	PCA $\rightarrow$ Manifold learning
13	Reinforcement learning/Online learning/Active learning
14	Project presentation

Generalization bound for bounded loss

Assumption

- the loss function L is bounded, that is

$$0 \leq L(y, y') \leq c \quad \forall y, y' \in \mathcal{Y}$$

- the hypothesis space is a finite set, that is

$$\mathcal{H} = \{h_1, h_2, \ldots, h_m\}.$$

PAC estimate for ERM

Theorem

For any $\delta > 0$ *and* $\epsilon > 0$, *if*

$$n \geq \frac{8c^2}{\epsilon^2} \log\left(\frac{2|\mathcal{H}|}{\delta}\right)$$

then $\hat{h}_n$ *is* ϵ*-accurate with probability at least* $1 - \delta$.

PAC estimate for ERM

$$n = \frac{8c^2}{\epsilon^2} \log\left(\frac{2|\mathcal{H}|}{\delta}\right)$$

- Fix a level of confidence δ, the accuracy ϵ of the ERM is

$$\mathcal{O}\left(\frac{1}{\sqrt{n}}\sqrt{\log\left(\frac{1}{\delta}\right) + \log(|\mathcal{H}|)}\right)$$

- If we want $\epsilon \to 0$ as $n \to \infty$:

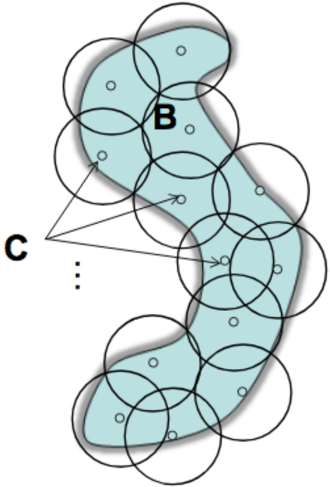
$$\log(|\mathcal{H}|) \ll n$$

- The convergence rate will not be better than $\mathcal{O}(n^{-1/2})$

Covering numbers

Remark: If $\mathcal{H}$ is a bounded $k-$dimensional manifold/algebraic surface, then we now that

$$\mathcal{N}(\epsilon, \mathcal{H}, d) = \mathcal{O}\left(\epsilon^{-k}\right)$$



Generalization bound using covering number.

- Assumption: $\mathcal{H}$ is a metric space with distance d defined on it.
- For $\epsilon > 0$, we denote by $\mathcal{N}(\epsilon, \mathcal{H}, d)$ the *covering number* of $(\mathcal{H}, d)$; that is, $\mathcal{N}(\epsilon, \mathcal{H}, d)$ is the minimal number of balls of radius ϵ needed to cover $\mathcal{H}$.
- Assumption: loss function L satisfies:

$$|L(h(x), y) - L(h'(x), y)| \leq C d(h, h') \quad \forall, x \in \mathcal{X}; y \in \mathcal{Y}; h, h' \in \mathcal{H}$$

Generalization bound using covering number

Theorem

For all $\epsilon > 0$, $\delta > 0$, *if*

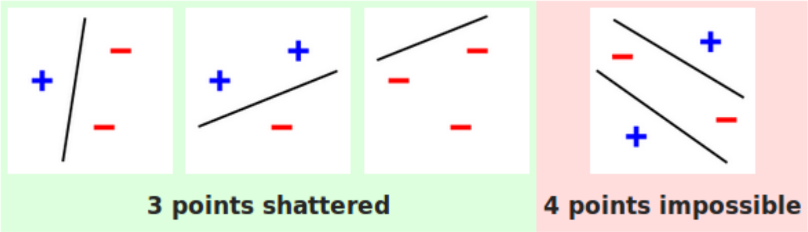
$$n \geq \frac{c^2}{2\epsilon^2} \log\left(\frac{2\mathcal{N}(\epsilon, \mathcal{H}, d)}{\delta}\right)$$

then

$$|R_n(h) - R(h)| \leq (2C + 1)\epsilon \quad \forall h \in \mathcal{H}.$$

with probability at least $1 - \delta$.

Vapnik–Chervonenkis dimension



The set of straight lines (as a binary classification model on points) in a two-dimensional plane has VC dimension 3.

Rademacher complexity

- measures richness of a class of real-valued functions *with respect to a probability distribution*
- Given a sample $S = (x_1, x_2, \ldots, x_n)$ and a class $\mathcal{H}$ of real-valued functions defined on the input space $\mathcal{X}$, the empirical Rademacher complexity of $\mathcal{H}$ given S is defined as:

$$Rad(\mathcal{H}) = \mathbb{E}_\sigma \left[\sup_{f \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^{m} \sigma_i f(x_i) \right]$$

where $\sigma_1, \sigma_2, \ldots, \sigma_m$ are independent random variables drawn from the Rademacher distribution

$$P[\sigma_i = 1] = P[\sigma_i = -1] = 1/2$$

Remarks

If we want $\epsilon \to 0$ as $n \to \infty$:

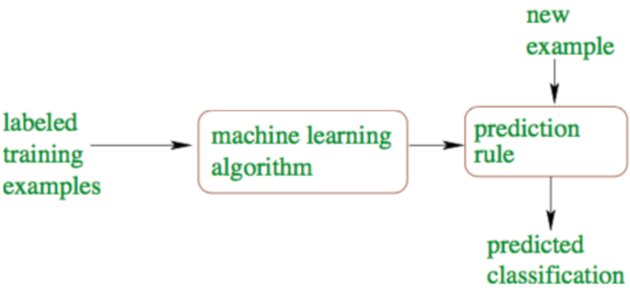
$$dimension(\mathcal{H}) \ll n$$

How do we get that?

- Model selection
- Feature selection
- Regularization:
  - Work for the case $dimension(\mathcal{H}) \gg n$
  - Stabilize an estimator $\to$ force it to live in a neighborhood of a lower-dimensional surface
  - Requires a stability bound instead of a uniform generalization bound

Classification: Logistic regression

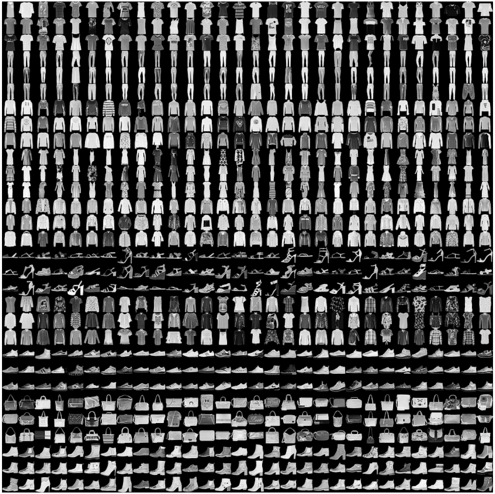
Supervised learning



Learning a function $h : \mathcal{X} \to \mathcal{Y}$ that maps an input to an output based on example input-output pairs

Classify hand-written characters

Classify images of clothing



Classification

- Logistic regression
- Linear Discriminant Analysis
- Support Vector Machines
- Nearest neighbours

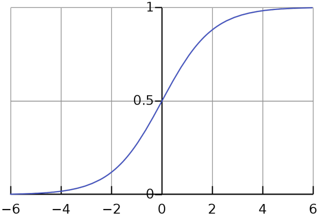
Logistic regression

- Suppose we work with binary outputs $\mathcal{Y} = \{0, 1\}$, and $\mathcal{X}$ is a subset of $\mathbb{R}^d$
- Note: Data are withdrawn from a joint distribution $P_{X,Y} \to$ even if we fix X, the label Y might be different from times to times
- Goal: Given input X, we want to model the probability that $Y = 1$

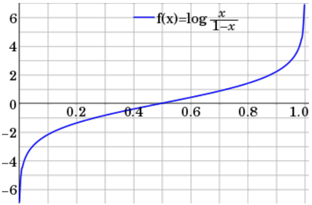
$$P[Y = 1|X = x]$$

- This is a function of x, with values in $[0, 1]$

Transformation between $(-\infty, \infty)$ and $[0, 1]$



$$f(x) = \frac{e^x}{1 + e^x}$$



$$logit(p) = \log \frac{p}{1-p}$$

Logistic regression: Assumptions

Assumption

Given $X = x$, Y is a Bernoulli random variable with parameter $p(x) = P[Y = 1|X = x]$ and

$$logit(p(x)) = \log \frac{p(x)}{1 - p(x)} = \log \frac{P[Y = 1|X = x]}{P[Y = 0|X = x]} = x^T \beta$$

for some vector $\beta \in \mathbb{R}^{d+1}$.

Note: Here we denote

$$x^T \beta = \beta_0 + \beta_1 x_1 + \ldots + \beta_d x_d$$

A GLM consists of

- A probability distribution for $Y|X = x$
- A linear predictor $\eta = x^T\beta$
- An activation function g such that $g(E[Y|X = x]) = \eta$

Logistic regression: Assumptions

Assumption

Given $X = x$, Y is a Bernoulli random variable with parameter $p(x) = P[Y = 1|X = x]$ and

$$logit(p(x)) = \log \frac{p(x)}{1 - p(x)} = \log \frac{P[Y = 1|X = x]}{P[Y = 0|X = x]} = x^T \beta$$

for some vector $\beta \in \mathbb{R}^{d+1}$.

Implicit agreement: Real data are generated from this model with a "true" parameter β^*. Our task is to find this β^*.

Parameter estimation: maximum likelihood

- Remember that for Bernoulli r.v. with parameter p

$$P[Y = y] = p^y(1-p)^{1-y}, \quad y \in \{0, 1\}$$

- Given samples $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, we have

$$L(\beta) = \prod_{i=1}^{n} p(x_i, \beta)^{y_i}(1 - p(x_i, \beta))^{1-y_i}$$

- Maximum likelihood (ML): maximize this likelihood function

Parameter estimation: maximum likelihood

The log-likelihood can be computed as

$$\begin{aligned}
\ell(\beta) &= \log L(\beta) \\
&= \sum_{i=1}^{n} [y_i \log p(x_i, \beta) + (1 - y_i) \log(1 - p(x_i, \beta))] \\
&= \sum_{i=1}^{n} \left[y_i x_i^T \beta - y_i \log(1 + e^{x_i^T \beta}) - (1 - y_i) \log(1 + e^{x^T \beta}) \right] \\
&= \sum_{i=1}^{n} \left[y_i x_i^T \beta - \log(1 + e^{x_i^T \beta}) \right].
\end{aligned}$$

In the language of statistical learning

- The hypothesis space: the set of all possible values for β (including the true parameter β^*)
- The loss function

$$loss_\beta(x, y) = -yx^T\beta + \log(1 + e^{x^T\beta})$$

- It can be proved that the risk function

$$R(\beta) = E[loss_\beta(x, y)]$$

has a unique minimizer at β^*

Logistic regression: estimating the parameter

- We want to maximize

$$\ell(\beta) = \sum_{i=1}^{n} \left[y_i x_i^T \beta - \log(1 + e^{x_i^T \beta}) \right].$$

- Derivative with respect to the parameter

$$\frac{\partial \ell}{\partial \beta_j}(\beta) = \sum_{i=1}^{n} \left[y_i x_{ij} - x_{ij} \frac{e^{x_i^T \beta}}{1 + e^{x_i^T \beta}} \right].$$

- The optimization needs to be performed by a numerical optimization method
- Penalties can be added to regularize the problem to avoid overfitting

$$\min_{\beta} -\ell(\beta) + \alpha\|\beta\|_1$$

or

$$\min_{\beta} -\ell(\beta) + \alpha\|\beta\|_2$$

Logistic regression with more than 2 classes

- Suppose now the response can take any of $\{1, \ldots, K\}$ values
- We use the categorical distribution instead of the Bernoulli distribution

$$P[Y = k | X = x] = p_k(x), \quad \sum_{k=1}^{K} p_k(x) = 1.$$

- Model

$$p_k(x) = \frac{e^{x^T \beta^{(k)}}}{\sum_{k=1}^{K} e^{x^T \beta^{(k)}}}$$

One versus one

- Train a classifier for each possible pair of classes
- Classify a new points according to a majority vote: count the number of times the new point is assign to a given class, and pick the class with the largest number

One versus all

- Fit the model to separate each class against the remaining classes $\rightarrow$ obtain

$$p_k(x) = \frac{e^{x^T\beta^{(k)}}}{1 + e^{x^T\beta^{(k)}}}$$

- Choose the label k that maximize $p_k(x)$