DDA5001 Machine Learning

Linear Classification II: Logistic Regression

Xiao Li

School of Data Science
The Chinese University of Hong Kong, Shenzhen

Recap: VC Dimension Generalization Result

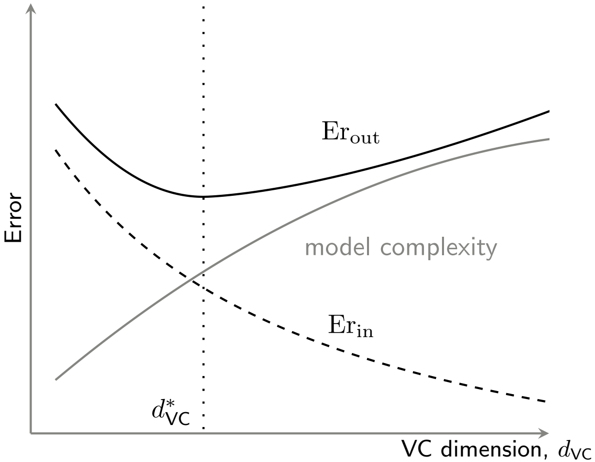
VC generalization bound

For any $\delta > 0$, with probability at least $1 - \delta$, we have the following generalization bound:

$$\boxed{\forall f \in \mathcal{H} \qquad \mathrm{Er}_{\mathrm{out}}(f) \leq \mathrm{Er}_{\mathrm{in}}(f) + \mathcal{O}\left(\sqrt{\frac{d_{\mathsf{VC}}}{n}}\right)}$$

- This result is very general to cover all cases, and hence it is a loose result.
- It still provides meaningful information about learning. For instance, more training data is always better and larger d_{VC} has a worse generalization ability.

Recap: Learning Curve from VC Analysis



- The optimal model is the one that minimizes the combinations of $\mathrm{Er}_{\mathrm{in}}$ and generalization error.

Logistic Regression

Conditional Probability for Classification

- We are going to classify 'Approve' and 'Reject'.
- Labeling: 'Approve' $y = +1$, 'Reject' $y = -1$.
- Now you have a test data $\boldsymbol{x}$ without labeling

- Suppose now you know

$$\Pr[y = +1|\boldsymbol{x}] = 0.8, \quad \Pr[y = -1|\boldsymbol{x}] = 0.2$$

Which class you will assign $\boldsymbol{x}$ to?

Optimal Classifier Induced by Conditional Probability

Bayes-optimal classifier

The classifier

$$y \leftarrow \operatorname*{argmax}_{y \in \mathcal{Y}} \Pr[y|\boldsymbol{x}]$$

is optimal over all possible classifiers.

- $\Pr[y|\boldsymbol{x}]$ is called a-posteriori probability of y.
- Implication: Compute $\Pr[y|\boldsymbol{x}]$ for optimal classification.
- How can we know $\Pr[y|\boldsymbol{x}]$?
- Suppose we have training data

$$\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$$

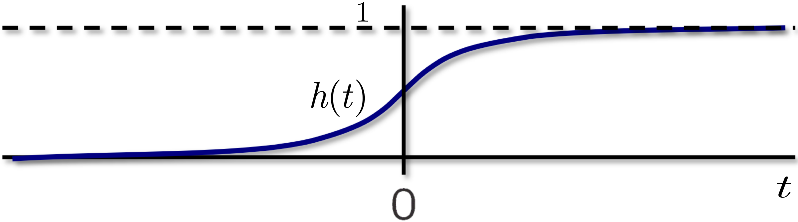
- We can learn an estimator $\Pr_{\boldsymbol{\theta}}[y|\boldsymbol{x}]$ for $\Pr[y|\boldsymbol{x}]$ based on the training data.

Logistic Function / Sigmoid

The function

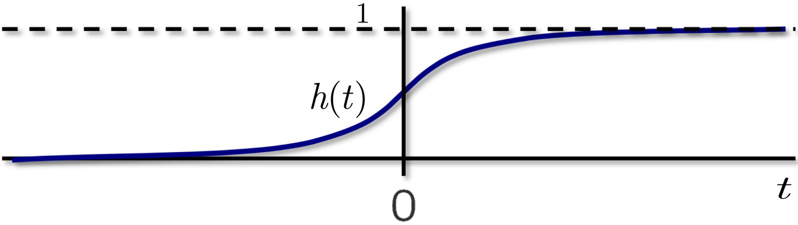
$$h(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$$

is called the logistic function or sigmoid.



Sigmoid: 'S'-like function.

Logistic Function: Probability Interpretation



- $h(t) \in [0, 1]$ — can be interpreted as probability.
- $\Pr[y|\boldsymbol{x}]$ is also a kind of probability.

Link? Using $h(t)$ to approximate $\Pr[y|\boldsymbol{x}]$.

Logistic Regression Model for Binary Classification

Logistic regression (LR) has the following $\Pr_{\boldsymbol{\theta}}[y|\boldsymbol{x}]$ for modeling $\Pr[y|\boldsymbol{x}]$:

$$\Pr_{\boldsymbol{\theta}}[y=+1|\boldsymbol{x}] = h(\boldsymbol{\theta}^\top \boldsymbol{x}) = \frac{1}{1+e^{-\boldsymbol{\theta}^\top \boldsymbol{x}}}$$

$$\Pr_{\boldsymbol{\theta}}[y=-1|\boldsymbol{x}] = 1 - \Pr_{\boldsymbol{\theta}}[y=+1|\boldsymbol{x}] = \frac{1}{1+e^{\boldsymbol{\theta}^\top \boldsymbol{x}}}$$

Thus,

$$\boxed{\Pr_{\boldsymbol{\theta}}[y|\boldsymbol{x}] = \frac{1}{1+\exp\left(-y \cdot \boldsymbol{\theta}^\top \boldsymbol{x}\right)}}$$

- The learning process is to learn a $\widehat{\boldsymbol{\theta}}$ such that $\Pr_{\widehat{\boldsymbol{\theta}}}[y|\boldsymbol{x}]$ approximates the underlying $\Pr[y|\boldsymbol{x}]$ well (at least on training data).
- Logistic regression is actually a classification technique.
- Intrinsically, it is tailored for binary classification, $y \in \{+1, -1\}$.

Logistic Regression is a Linear Classifier

Suppose we have learned $\boldsymbol{\theta}$

$$\Pr_{\boldsymbol{\theta}}[y = +1|\boldsymbol{x}] = \frac{1}{1 + e^{-\boldsymbol{\theta}^\top \boldsymbol{x}}} \; > \; \frac{1}{2} \qquad \text{(classify } \boldsymbol{x} \text{ as class } +1)$$

This is equivalent to

$$e^{-\boldsymbol{\theta}^\top \boldsymbol{x}} < 1$$

This is further equivalent to

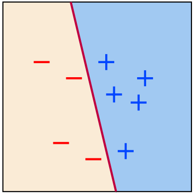
$$\boldsymbol{\theta}^\top \boldsymbol{x} > 0$$

Thus

$$y = \begin{cases} +1, & \boldsymbol{\theta}^\top \boldsymbol{x} > 0 \\ -1, & \boldsymbol{\theta}^\top \boldsymbol{x} < 0 \end{cases}$$

Logistic Regression is a Linear Classifier

$$y = \begin{cases} +1, & \boldsymbol{\theta}^\top \boldsymbol{x} > 0 \\ -1, & \boldsymbol{\theta}^\top \boldsymbol{x} < 0 \end{cases}$$



- This reduces to our linear classification model $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{\theta}^\top \boldsymbol{x}$.
- LR and the perceptron are two different methodologies for learning $f_{\boldsymbol{\theta}}(\boldsymbol{x})$.
- In LR, How to choose $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ from $\mathcal{H}$?

Logistic Regression

- Recall training data pairs:

$$\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$$

- Represent a-posteriori probability for $(\boldsymbol{x}_i, y_i)$

$$\boxed{\Pr_{\boldsymbol{\theta}}[y_i|\boldsymbol{x}_i] = \frac{1}{1 + \exp\left(-y_i \cdot \boldsymbol{\theta}^\top \boldsymbol{x}_i\right)}}$$

- Observation: The likelihood of $(\boldsymbol{x}_i, y_i)$ given parameter $\boldsymbol{\theta}$.

How to learn parameter $\boldsymbol{\theta}$?

Maximum likelihood estimation principle.

Logistic Regression: The Learning Problem

- The likelihood of all data $\{(\boldsymbol{x}_i, y_i)\}$ (i.i.d.):

$$\prod_{i=1}^{n} \Pr_{\boldsymbol{\theta}} [y_i | \boldsymbol{x}_i]$$

- The log-likelihood of all data $\{(\boldsymbol{x}_i, y_i)\}$:

$$\sum_{i=1}^{n} \log \left(\Pr_{\boldsymbol{\theta}} [y_i | \boldsymbol{x}_i]\right) = -\sum_{i=1}^{n} \log \Big(1 + \exp \Big(-y_i \cdot \boldsymbol{\theta}^\top \boldsymbol{x}_i\Big)\Big)$$

- Maximum likelihood estimation leads to the LR problem:

$$\boxed{\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \mathbb{R}^d}{\text{argmin}} \; \frac{1}{n} \sum_{i=1}^{n} \underbrace{\log \Big(1 + \exp \Big(-y_i \cdot \boldsymbol{\theta}^\top \boldsymbol{x}_i\Big)\Big)}_{\ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), y_i)}}$$

What we are going to minimize? Training error measured by logistic loss, sometimes also called cross-entropy loss. Also related to minimizing in-sample error $\mathrm{Er}_{\mathrm{in}}$ with 0-1 error measure.

Revisiting Generalization: How to Make $\mathrm{Er}_{\mathrm{out}}$ Small

- Generalization theory says:

$$\forall f_{\boldsymbol{\theta}} \in \mathcal{H} \qquad \mathrm{Er}_{\mathrm{out}}(f_{\boldsymbol{\theta}}) \leq \mathrm{Er}_{\mathrm{in}}(f_{\boldsymbol{\theta}}) + \mathcal{O}\left(\sqrt{\frac{d_{\mathsf{VC}}}{n}}\right).$$

- The goal: Make $\mathrm{Er}_{\mathrm{out}}$ small.
- The generalization error is fixed when $\mathcal{H}$ and training data are fixed.
- Make the $\mathrm{Er}_{\mathrm{in}}(f_{\boldsymbol{\theta}})$ small by choosing a specific $f_{\boldsymbol{\theta}} \in \mathcal{H}$.

How? Design algorithm for training to pick a $\widehat{\boldsymbol{\theta}}$ such that:

$$\boxed{\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \mathrm{Er}_{\mathrm{in}}(f_{\boldsymbol{\theta}}) \leftarrow \widehat{\boldsymbol{\theta}} = \operatorname*{argmin}_{\boldsymbol{\theta} \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), y_i\right).}$$

Learned model: $f_{\widehat{\boldsymbol{\theta}}} \in \mathcal{H}$, provides small $\mathrm{Er}_{\mathrm{out}}(f_{\widehat{\boldsymbol{\theta}}})$.

$\rightsquigarrow$ Gives the motivation for formulating the logistic regression problem.

Logistic Regression vs. Perceptron

- Perceptron: Find any linear classifier that correctly classification $+1$'s and -1's, i.e., $\mathrm{sign}(\boldsymbol{\theta}^\top \boldsymbol{x})$ is correct.
- Logistic Regression: Tend to simultaneously classify $+1$'s and -1's into its right-most and left-most sides, respectively.
- In addition, logistic regression does not assume linearly separable data.

LR is intuitively better compared to perceptron.

Logistic Regression vs. Least Squares

- Logistic regression: logistic loss

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \mathbb{R}^d}{\operatorname{argmin}} \ \frac{1}{n} \sum_{i=1}^{n} \log \left(1 + \exp \left(-y_i \cdot \boldsymbol{\theta}^\top \boldsymbol{x}_i \right) \right).$$

- Least squares: squared ℓ_2-norm loss

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \mathbb{R}^d}{\operatorname{argmin}} \ \|\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y}\|_2^2.$$

Optimization

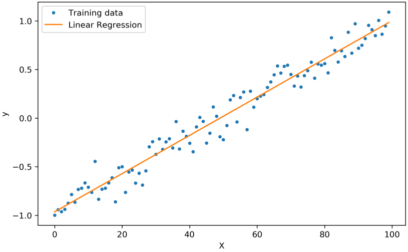
- Least squares: Closed-form solution.
- Logistic regression: No closed-form.

Regression vs. classification

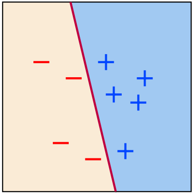
- Least squares: Tailored for Regression.
- Logistic regression: Tailored for classification.

Recall: Regression v.s. Classification

Regression



Classification



- Regression is to fit a continuous quantity, $y \in \mathbb{R}$ is continuous.
- Classification is to fit a discrete labels, $y \in \{-1, +1\}$ is categorical.

Extension: Multi-class Logistic Regression

Softmax: Extension of Logistic Function

- The logistic regression developed so far is for binary classification.

How about when number of classes $K > 2$?

- The key idea is to assign each class $k = 1, \ldots, K$ a parameter / weight vector $\boldsymbol{\theta}_k$.
- Let $\boldsymbol{\Theta} = \left[\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K\right] \in \mathbb{R}^{(d+1)\times K}$ and $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ be the training data.
- The model for estimating the a-posteriori of y_i is given by

$$\boxed{\Pr_{\boldsymbol{\Theta}}\left[y_i = k|\boldsymbol{x}_i\right] = \frac{\exp(\boldsymbol{\theta}_k^\top \boldsymbol{x}_i)}{\sum_{j=1}^K \exp(\boldsymbol{\theta}_j^\top \boldsymbol{x}_i)}}$$

also known as softmax. It is clear that $\Pr\left[y_i = k|\boldsymbol{\Theta}, \boldsymbol{x}_i\right]$ sum to 1 over k.

Multi-class Logistic Regression

Using the reasoning of MLE, we can formulate the learning problem as

$$\boxed{\widehat{\boldsymbol{\Theta}} = \underset{\boldsymbol{\Theta}\in\mathbb{R}^{d\times K}}{\operatorname{argmin}} \; -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} 1_{\{y_i=k\}} \log\left(\frac{\exp(\boldsymbol{\theta}_k^\top \boldsymbol{x}_i)}{\sum_{j=1}^{K}\exp(\boldsymbol{\theta}_j^\top \boldsymbol{x}_i)}\right),}$$

where $1_{\{y_i=k\}}$ is the indicator function defined as

$$1_{\{y_i=k\}} = \begin{cases} 1, & y_i \text{ is } k\text{-th class} \\ 0, & \text{otherwise} \end{cases}$$

- Why MLE leads to such a formulation? (HW2).

Summary of Logistic Regression

- The most important concept in LR is to use logistic function / softmax to approximate $\Pr[y|\boldsymbol{x}]$, i.e.,

$$\Pr[y|\boldsymbol{x}] \leftarrow \Pr_{\boldsymbol{\theta}}[y|\boldsymbol{x}] = \frac{1}{1 + \exp\left(-y \cdot \boldsymbol{\theta}^\top \boldsymbol{x}\right)}.$$

- LR is to use the data $\{(\boldsymbol{x}_i, y_i)\}$ directly to learn such a model $\Pr_{\boldsymbol{\theta}}[y|\boldsymbol{x}]$.
- Later, we will study that deep neural networks and language models are also learning this model $\Pr_{\boldsymbol{\theta}}[y|\boldsymbol{x}]$ but not directly using the data $\{(\boldsymbol{x}_i, y_i)\}$. $\rightsquigarrow$ Later lectures on deep learning.

How to Learn $\widehat{\boldsymbol{\theta}}$?

The objective function is (using binary logistic regression as an example):

$$\mathcal{L}(\boldsymbol{\theta}) := \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + \exp\left(-y_i \cdot \boldsymbol{\theta}^\top \boldsymbol{x}_i\right)\right)$$

The learning problem is formulated as

$$\boxed{\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}\in\mathbb{R}^d}{\mathrm{argmin}}\ \mathcal{L}(\boldsymbol{\theta})}$$

- Bad news ✗: No closed-form solution.
- Good news ✓: The objective function $\mathcal{L}(\boldsymbol{\theta})$ is convex in $\boldsymbol{\theta}$.

$\rightsquigarrow$ Next lectures: Convex optimization and gradient-based optimization algorithms.