Optimization Problems: Linear Classification

Chih-Jen Lin
National Taiwan University

Last updated: April 23, 2021

Minimizing Training Errors

- Basically a classification method starts with minimizing the training errors

$$\min_{\text{model}} \quad (\text{training errors})$$

- That is, all or most training data with labels should be correctly classified by our model
- A model can be a decision tree, a neural network, or other types

Minimizing Training Errors (Cont'd)

- For simplicity, let's consider the model to be a vector $\boldsymbol{w}$
- That is, the decision function is

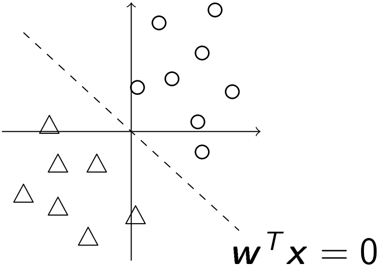
$$\text{sgn}(\boldsymbol{w}^T \boldsymbol{x})$$

- For any data, $\boldsymbol{x}$, the predicted label is

$$\begin{cases} 1 & \text{if } \boldsymbol{w}^T \boldsymbol{x} \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

Minimizing Training Errors (Cont'd)

- The two-dimensional situation



- This seems to be quite restricted, but practically $\boldsymbol{x}$ is in a much higher dimensional space

Minimizing Training Errors (Cont'd)

- To characterize the training error, we need a loss function $\xi(\boldsymbol{w}; y, \boldsymbol{x})$ for each instance $(y, \boldsymbol{x})$, where

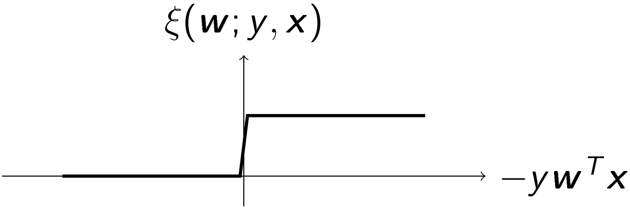
  $y = \pm 1$ is the label and $\boldsymbol{x}$ is the feature vector

- Ideally we should use 0–1 training loss:

$$\xi(\boldsymbol{w}; y, \boldsymbol{x}) = \begin{cases} 1 & \text{if } y\boldsymbol{w}^T\boldsymbol{x} < 0, \\ 0 & \text{otherwise} \end{cases}$$

Minimizing Training Errors (Cont'd)

- However, this function is discontinuous. The optimization problem becomes difficult



- We need continuous approximations

Common Loss Functions

- Hinge loss (l1 loss)

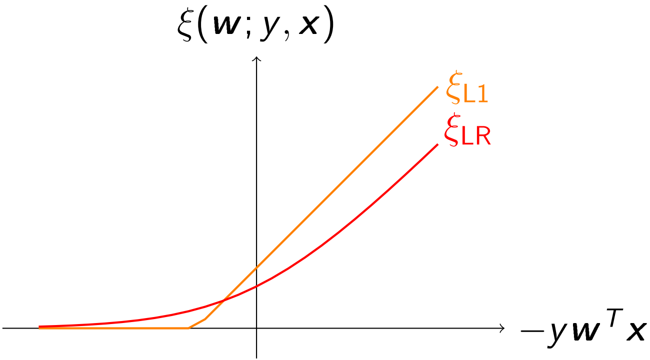
$$\xi_{\mathrm{L1}}(\boldsymbol{w}; y, \boldsymbol{x}) \equiv \max(0, 1 - y\boldsymbol{w}^T\boldsymbol{x}) \qquad (1)$$

- Logistic loss

$$\xi_{\mathrm{LR}}(\boldsymbol{w}; y, \boldsymbol{x}) \equiv \log(1 + e^{-y\boldsymbol{w}^T\boldsymbol{x}}) \qquad (2)$$

- Support vector machines (SVM): Eq. (1). Logistic regression (LR): (2)
- SVM and LR are two very fundamental classification methods

Common Loss Functions (Cont'd)



- Logistic regression is very related to SVM
- Their performance is usually similar

Common Loss Functions (Cont'd)

- However, minimizing training losses may not give a good model for future prediction
- Overfitting occurs

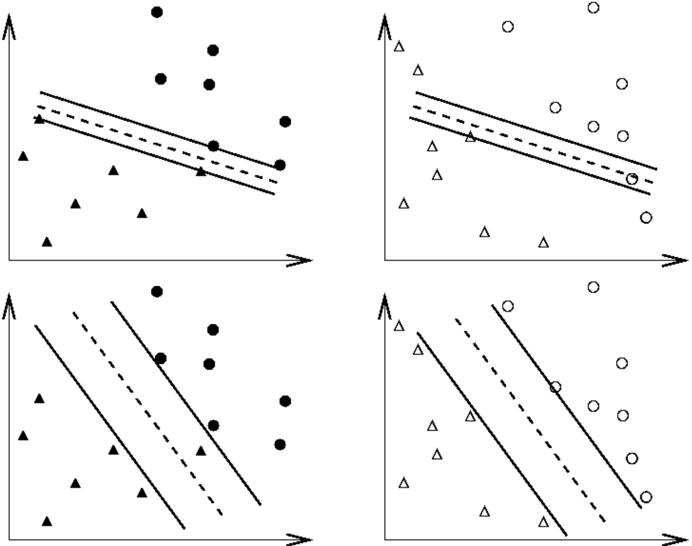
Overfitting

- See the illustration in the next slide
- For classification, you can easily achieve 100% training accuracy
- This is useless
- When training a data set, we should

  Avoid underfitting: small training error

  Avoid overfitting: small testing error

● and ▲: training; ○ and △: testing



Regularization

- To minimize the training error we manipulate the $\boldsymbol{w}$ vector so that it fits the data
- To avoid overfitting we need a way to make $\boldsymbol{w}$'s values less extreme.
- One idea is to make $\boldsymbol{w}$ values closer to zero
- We can add, for example,

$$\frac{\boldsymbol{w}^T\boldsymbol{w}}{2} \quad \text{or} \quad \|\boldsymbol{w}\|_1$$

to the function that is minimized

General Form of Linear Classification

- Training data $\{y_i, \boldsymbol{x}_i\}, \boldsymbol{x}_i \in R^n, i = 1, \ldots, l, \; y_i = \pm 1$
- l: # of data, n: # of features

$$\min_{\boldsymbol{w}} f(\boldsymbol{w}), \quad f(\boldsymbol{w}) \equiv \frac{\boldsymbol{w}^T \boldsymbol{w}}{2} + C \sum_{i=1}^{l} \xi(\boldsymbol{w}; y_i, \boldsymbol{x}_i)$$

- $\boldsymbol{w}^T \boldsymbol{w}/2$: regularization term
- $\xi(\boldsymbol{w}; y, \boldsymbol{x})$: loss function
- C: regularization parameter (chosen by users)