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
  \[ \min_{\text{model}} \ (\text{training errors}) \]

- 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
For simplicity, let’s consider the model to be a vector \( \mathbf{w} \)

That is, the decision function is

\[
\text{sgn}(\mathbf{w}^T \mathbf{x})
\]

For any data, \( \mathbf{x} \), the predicted label is

\[
\begin{cases} 
1 & \text{if } \mathbf{w}^T \mathbf{x} \geq 0 \\
-1 & \text{otherwise}
\end{cases}
\]
Minimizing Training Errors (Cont’d)

- The two-dimensional situation

\[ w^T x = 0 \]

- This seems to be quite restricted, but practically \( x \) is in a much higher dimensional space
Minimizing Training Errors (Cont’d)

To characterize the training error, we need a loss function \( \xi(w; y, x) \) for each instance \((y, x)\), where \( y = \pm 1 \) is the label and \( x \) is the feature vector.

Ideally we should use 0–1 training loss:

\[
\xi(w; y, x) = \begin{cases} 
1 & \text{if } yw^T x < 0, \\
0 & \text{otherwise}
\end{cases}
\]
However, this function is discontinuous. The optimization problem becomes difficult

$$\xi(w; y, x)$$

We need continuous approximations
Common Loss Functions

- Hinge loss (l1 loss)
  \[ \xi_{L1}(\mathbf{w}; y, x) \equiv \max(0, 1 - y\mathbf{w}^T\mathbf{x}) \] (1)

- Logistic loss
  \[ \xi_{LR}(\mathbf{w}; y, x) \equiv \log(1 + e^{-y\mathbf{w}^T\mathbf{x}}) \] (2)

- Support vector machines (SVM): Eq. (1). Logistic regression (LR): (2)
- SVM and LR are two very fundamental classification methods
Logistic regression is very related to SVM
Their performance is usually similar
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 $w$ vector so that it fits the data.
- To avoid overfitting we need a way to make $w$’s values less extreme.
- One idea is to make $w$ values closer to zero.
- We can add, for example,

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

...to the function that is minimized.
General Form of Linear Classification

- Training data \( \{y_i, x_i\}, x_i \in R^n, i = 1, \ldots, l, y_i = \pm 1 \)
- \( l \): # of data, \( n \): # of features

\[
\min_w f(w), \quad f(w) \equiv \frac{w^T w}{2} + C \sum_{i=1}^{l} \xi(w; y_i, x_i)
\]

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