Chapter 4 Fitting Model to Data

10/27/19 10:51 PM

Parameter learning / parametric modeling

1. Choose a model with parameters unspecified
2. Use data mining techniques to decide attributes (e.g. in chapter 3)
3. Fit model (parameters) to best represent the data

Linear Discriminant function

- F(x) as a linear function of all numeric values
- Determine the class of target by observing f(x) positive or negative
- Graph representation: a line separating dots on a plain, or hyperplane for higher dimension
- Parameters: linear weights. Informally weights can represent importance of features

In applications of scoring and ranking:

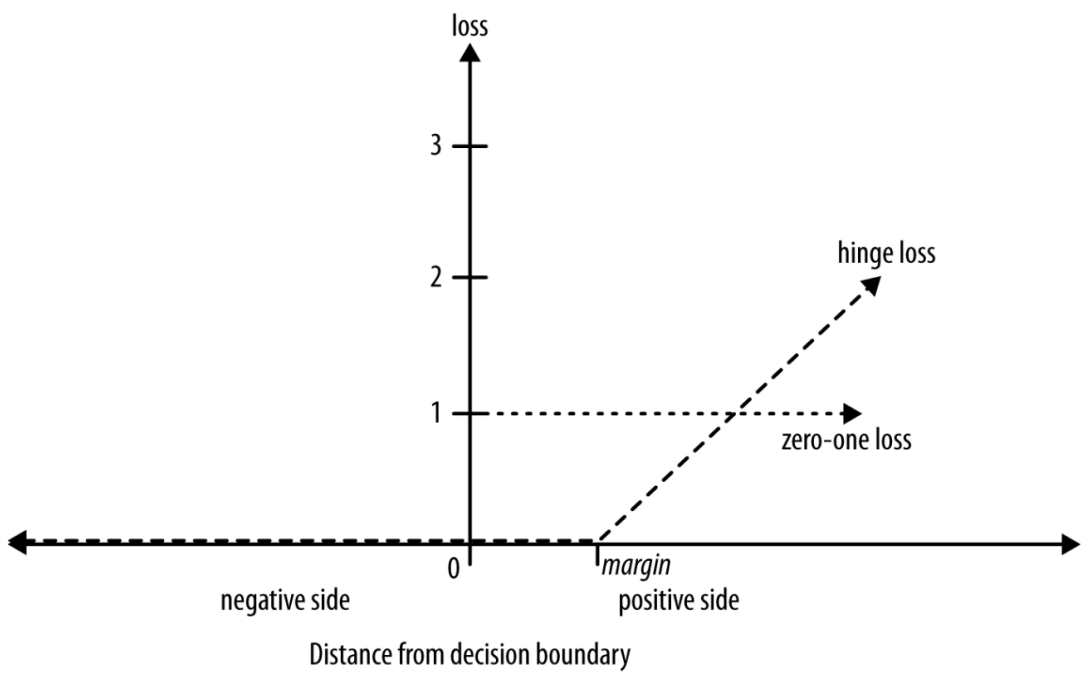
- Scoring: probability estimation for becoming (more than one) classes
- Ranking: Highest possible instances (doesn't necessarily need actual probability)

- Linear discriminant function can produce a ranking for free, f(x) itself generates intuitive ranking outcomes.

Loss Function

General idea: penalty assigned to each instance based on error in model's predictive value

- X axis: distance from the separation boundary (negative side if correctly assigned)
- Y axis: penalty for the instance
- Example in graph: two loss functions, hinge loss for SVM and zero-one loss



- SVM has hinge loss function because penalty is only assigned if the instance is on wrong side and beyond margin (no penalty on wrong side but within margin)

Linear Regression

- Use mean square error (or sum of squared errors) as loss function
- Favors squared errors because it assigns large penalty on extreme data
- It makes linear regression very sensitive to outliers. If not fitting a business problem, can change to absolute error (no square)

Logistic Regression

- Very similar to linear discriminant, but the output is log-odds
  - Good for probability estimation!
- Threshold set as 0.5 by nature of function

Non-linear Functions

Non-linear SVM:

- A kernel function that maps the original features to another feature space
- Linear classification on the mapped feature space, similar as before

Neural Network:

- Stack of models, learn from output from next layer
- Targets are only the last layer
- Can be seen as a huge complex parametric models, parameters assigned for each layer, so no targets needed for each layer's training