Notes 4-Linear Models for Classification

Junda Du

2019-9-13

We will discuss examples of all three approaches in this chapter which has been mention in charpter 1.

- Discriminant function: construct a discriminant function that directly assigns each vector $\boldsymbol{x}$ to a specific class.
- Inference and decision: models the $p(C_k|\boldsymbol{x})$ in an inference stage, and then uses this distribution to make optimal decisions.(discussed in 1.5.4). There are two approaches to determine the conditional probablity $p(C_k|\boldsymbol{x})$,
  - Parameteric model: represent ithem as parameteric models and then optimizing the parameters using a training set.
  - Generative approach: model the class-conditional desities $p(\boldsymbol{x}|C_k)$ and prior probablities $p(C_k)$, and then compute the required posterior probablities using Bayes' theorem.

Points:

- Representation of target vector: 1-of-K coding scheme where $\boldsymbol{t}$ is a vector such that if the class is C_j then $t_j = 1$ and $t_k = 0, k \neq j$.
- Relationship with linear regression models: for regression problem, the simplest model take the form $y(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0$, y is a real number; for classification problem, we need to use nonlinear function to transform the linear function of $\boldsymbol{w}$ beacuse we wish to predict discrete class labels or more genetally posterior probablities that lie in the range (0,1).
- The form of linear regression model: $y(\boldsymbol{x}) = f(\boldsymbol{w}^T\boldsymbol{x} + w_0)$, where $f(\cdot)$ is a nonlinear function, known as activation function, whereas its inverse is called a link functoin in the statistics literature. It's called the *generalized linear model* because the decision surfaces correspond to $y(\boldsymbol{x}) = const$, so that $\boldsymbol{w}^T\boldsymbol{x}+w_0 = const$ and hence the decision surfaces are linear functions of $\boldsymbol{x}$ but not linear in the parameters.?
- The algorithms discussed in this chapter will be equally applicable if we fist make a fixed nonlinear transformation $\boldsymbol{\phi}(\boldsymbol{x})$ which is used in Section 4.3.

1 Discriminant Functions

A discriminant is a function that takes an input vector $\boldsymbol{x}$ and assigns it to one of K classes, denoted C_k. In this chapter, we shall restrict attention to linear discriminants, namely those for which the decision surfaces are hyperplanes.

1.1 Two classes

The simpliest representation of a linear discriminant function is $y(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0$, The corresponding decision boundary is $y(\boldsymbol{x}) = 0$, $\boldsymbol{x}$ is assigned to class C_1 if $y(\boldsymbol{x}) \geq 0$ and to class C_2 otherwise.
Geometry meaning:

- $\boldsymbol{w}$ determines the orientation of the decision surface because $y(\boldsymbol{x}_A) = y(\boldsymbol{x}_B) = 0$, then $\boldsymbol{w}^T(\boldsymbol{x}_A - \boldsymbol{x}_B) = 0$
- $\dfrac{y(\boldsymbol{x})}{\|\boldsymbol{w}\|}$ is the distance between the point P $\boldsymbol{x}$ and the boundary hyperplane. Because $\dfrac{\boldsymbol{w}^T\boldsymbol{x}}{\|\boldsymbol{w}\|}$ is the projection? from OP to $\boldsymbol{w}$, and $\dfrac{w_0}{\|\boldsymbol{w}\|}$ is the projection? from OP' to $\boldsymbol{w}$, P' is the projected point on hyperplane.

1.2 Multiple classes

Now consider the extension of linear discriminants to $K > 2$ classes.

- *one-versus-the-rest* classifier: for each boundary, it discriminate $C_i(y(\boldsymbol{x}) > 0)$ and not $C_i(y(\boldsymbol{x}) < 0)$?.Not applicable $\times$.
- *one-versus-one* classifier: it discriminate $C_i(y(\boldsymbol{x}) > 0)$ and $C_j(y(\boldsymbol{x}) < 0)$?.Not applicable $\times$.
- A single K-class discriminant comprising K linear functions: $y_k(\boldsymbol{x}) = \boldsymbol{w}_k^T\boldsymbol{x} + w_{k0}$, assign a point $\boldsymbol{x}$ to class C_k if $y_k(\boldsymbol{x}) > y_j(\boldsymbol{x})$ for all $j \neq k$.Applicable $\checkmark$!

Properties about the K linear functions:

- The decision boundary between class C_k and C_j is therefore given by $y_k(\boldsymbol{x}) = y_j(\boldsymbol{x})$, i.e. $(\boldsymbol{w}_k - \boldsymbol{w}_j)^T\boldsymbol{x} + (w_{k0} - w_{j0}) = 0$. It have the same form and propertied with two-classes case.
- Singly connected and convex: consider two points $\boldsymbol{x}_A, \boldsymbol{x}_B$ both of which lie inside decision region R_k, any point $\hat{\boldsymbol{x}}$ that lies on the line connecting $\boldsymbol{x}_A, \boldsymbol{x}_B$ also lies inside R_k.Show it: $\hat{\boldsymbol{x}}$ can be expressed in the form $\hat{\boldsymbol{x}} = \lambda\boldsymbol{x}_A + (1-\lambda)\boldsymbol{x}_B$ where $0 \leq \lambda \leq 1$

$$\hat{\boldsymbol{x}} = \lambda\boldsymbol{x}_A + (1-\lambda)\boldsymbol{x}_B$$
$$\xLeftrightarrow{Linearity\,of\,the\,discriminant\,functions} y_k(\hat{\boldsymbol{x}}) = \lambda y_k(\boldsymbol{x}_A) + (1-\lambda)y_k(\boldsymbol{x}_B)$$
$$\xLeftrightarrow{y_k(\boldsymbol{x}_A)>y_j(\boldsymbol{x}_A),y_k(\boldsymbol{x}_B)>y_j(\boldsymbol{x}_B)} y_k(\hat{\boldsymbol{x}}) > y_j(\hat{\boldsymbol{x}}), for\ all\ j \neq k$$

1.3 Least squares for classification

$y_k(\boldsymbol{x}) = \boldsymbol{w}_k^T\boldsymbol{x} + w_{k0}$ where $k = 1, \ldots, K$. We can group these together using vector notation so that $\boldsymbol{y}(\boldsymbol{x}) = \widetilde{\boldsymbol{W}}^T\widetilde{\boldsymbol{x}}$, $\widetilde{\boldsymbol{W}} = (\widetilde{\boldsymbol{w}}_1, \ldots, \widetilde{\boldsymbol{w}}_K)$, $\widetilde{\boldsymbol{w}}_k = (w_{k0}, \boldsymbol{w}_k^T)^T$, $\widetilde{\boldsymbol{x}} = (1, \boldsymbol{x}^T)^T$
The sum-of-squares error function can then be written as

$$E_D(\widetilde{\boldsymbol{W}}) = \frac{1}{2}Tr\left\{(\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{W}} - \boldsymbol{T})^T(\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{W}} - \boldsymbol{T})\right\}$$

Setting the derivative with respect to $\widetilde{\boldsymbol{W}}$ to zero, we obtain the solution for $\widetilde{\boldsymbol{W}}$ in the form

$$\widetilde{\boldsymbol{W}} = (\widetilde{\boldsymbol{X}}^T\widetilde{\boldsymbol{X}})^{-1}\widetilde{\boldsymbol{X}}^T\boldsymbol{T} = \widetilde{\boldsymbol{X}}^\dagger\boldsymbol{T}$$

We then obtain the discriminant function int the form

$$\boldsymbol{y}(\boldsymbol{x}) = \widetilde{\boldsymbol{W}}^T\widetilde{\boldsymbol{x}} = \boldsymbol{T}^T\left(\widetilde{\boldsymbol{X}}^\dagger\right)^T\widetilde{\boldsymbol{x}}$$

Properties:

- If every target vector in the training set satisfied some linear constraint $\boldsymbol{a}^T\boldsymbol{t}_n + b = 0$, then the model prediction will satisify the same constraint so that $\boldsymbol{a}^T\boldsymbol{y}(\boldsymbol{x}) + b = 0$
- If we use a 1-of-K coding scheme for K classes, $\boldsymbol{y}(\boldsymbol{x})$ will sum to 1, but it can't be interpreted as probablities because they are not constrained to lie within the intervel (0,1).
- Lack robustness to outliers, that is outliers have a great effect on the decision boundaries. It has more severe problems than robustness. Why? Because least squares for classification corresponds to maximum likelihood under the assumption of a Gaussian conditional distribution, whereas binary target vectors have a distribution that is far from Gaussian. We can solve this by using a more appropriate probabilistic models.

1.4 Fisher's linear discriminant

Consider $y(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x}$, assign $\boldsymbol{x}$ to class C_1, otherwise class C_2. By adjusting the components of the weight vector $\boldsymbol{w}$, we can select a projection that maximizes the class separation. We introduce two methods to measure the seperation of the classes.

1. Maximize the class mean separation
   Maximize $m_2 - m_1 = \boldsymbol{w}^T(\boldsymbol{m}_2 - \boldsymbol{m}_1), \boldsymbol{m}_k = \frac{1}{N_k}\sum_{n\in C_k}\boldsymbol{x}_n$, where $\sum_i w_i^2 = 1$
   Using a Lagrange multiplier to perform the constrained maximization, we then find $\boldsymbol{w} \propto (\boldsymbol{m}_2 - \boldsymbol{m}_1)$.
   Two classes may have considerable overlap when projected onto the line

joining their means. This difficulty arises from the within-class distribution is not junyunde?, that's strongly nondiagonal covariances of the class distributions.
Fisher solves it by maximizing a function that will give a large separation between the projected class mean while also giving a small variance within each projected class, thereby minimizing the class overlap.

2. Fisher discriminant criteria
The within-class variance of the transformed data from class C_k is given by $s_k^2 = \sum\limits_{n \in C_k} (y_n - m_k)^2, y_n = \boldsymbol{w}^T \boldsymbol{x}_n, m_k = \boldsymbol{w}_T \boldsymbol{m}$.

The Fisher criterion is defined to be the ratio of the between-class variance(class mean separation) to the within-class variance and is given by

$$\begin{aligned} J(\boldsymbol{w}) =& \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} \\ =& \frac{\boldsymbol{w}^T \boldsymbol{S}_B \boldsymbol{w}}{\boldsymbol{w}^T \boldsymbol{S}_W \boldsymbol{w}}, \boldsymbol{S}_B = (\boldsymbol{m}_2 - \boldsymbol{m}_1)(\boldsymbol{m}_2 - \boldsymbol{m}_1)^T, \\ & \boldsymbol{S}_W = \sum_{n \in C_1} (\boldsymbol{x}_n - \boldsymbol{m}_1)(\boldsymbol{x}_n - \boldsymbol{m}_1)^T + \sum_{n \in C_2} (\boldsymbol{x}_n - \boldsymbol{m}_2)(\boldsymbol{x}_n - \boldsymbol{m}_2)^T \end{aligned}$$

Differentiate $J(\boldsymbol{w})$ with respect to $\boldsymbol{w}$, we find that $J(\boldsymbol{w})$ is maximized when

$$\begin{aligned} & (\boldsymbol{w}^T \boldsymbol{S}_B \boldsymbol{w}) \boldsymbol{S}_W \boldsymbol{w} = (\boldsymbol{w}^T \boldsymbol{S}_W \boldsymbol{w}) \boldsymbol{S}_B \boldsymbol{w} \\ \xLeftrightarrow{Ignore\ magnitude} & \boldsymbol{S}_W \boldsymbol{w} \propto \boldsymbol{S}_B \boldsymbol{w} \\ \xLeftrightarrow{?\boldsymbol{S}_B = (\boldsymbol{m}_2 - \boldsymbol{m}_1)(\boldsymbol{m}_2 - \boldsymbol{m}_1)^T} & \boldsymbol{S}_W \boldsymbol{w} \propto (\boldsymbol{m}_2 - \boldsymbol{m}_1) \\ \Longleftrightarrow & \boldsymbol{w} \propto \boldsymbol{S}_W^{-1} (\boldsymbol{m}_2 - \boldsymbol{m}_1) \end{aligned}$$

1.5 Realtion to least squares

1.6 The perceptron algorithm

2 Probablistic Generative Models

2.1 Continous inputs

2.2 Maximum likelihood solution

2.3 Discrete features

2.4 Exponential family

3 Probabilistic Discriminative Models

3.1 Fixed basis functions

3.2 Logistic regression

3.3 Iterative reweighted least squares

3.4 Multiclass logistic regression

3.5 Probit regression

3.6 Canonical link functions

4 The Laplace Apprpximation

4.1 Model comparison and BIC

5 Bayesian Logistic Regression

5.1 Laplace approximation

5.2 Predictive distribution