Lecture 2. Linear Models for Classification

Bao Wang
Department of Mathematics
Scientific Computing and Imaging Institute
The University of Utah
Math 5750/6880, Fall 2023

Discriminant functions

- Discriminant: $\boldsymbol{x} \to y(\boldsymbol{x}) := \mathcal{C}_k \in \{1, 2, \cdots, K\}$.

- Two classes linear discriminant function:

$$y(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + w_0,$$

where $\boldsymbol{w}$ is the *weight vector* and w_0 is the *bias*.

- How to classify the input $\boldsymbol{x}$?

Discriminant functions

- Discriminant: $\boldsymbol{x} \to y(\boldsymbol{x}) := \mathcal{C}_k \in \{1, 2, \cdots, K\}$.

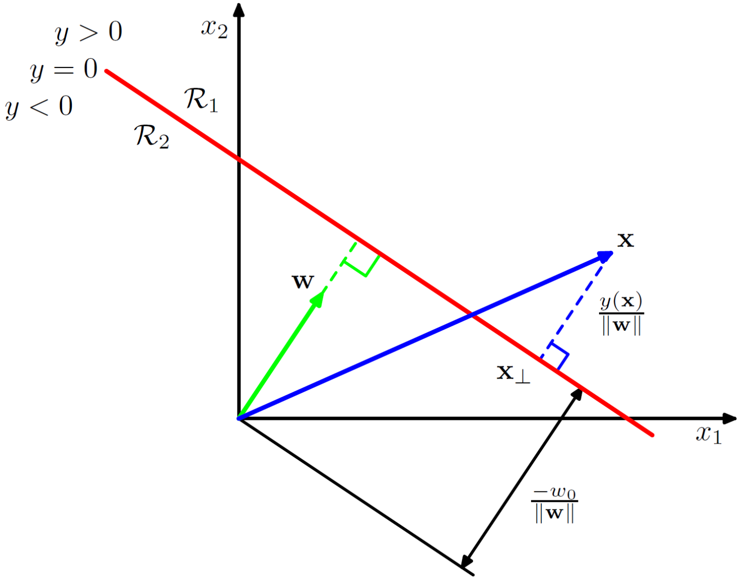
- Two classes linear discriminant function:

$$y(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + w_0,$$

where $\boldsymbol{w}$ is the *weight vector* and w_0 is the *bias*.

- $\boldsymbol{x} \to \mathcal{C}_1$ if $y(\boldsymbol{x}) \geq 0$ and $\boldsymbol{x} \to \mathcal{C}_2$ otherwise.

Discriminant functions in 2D



Decision boundary

- Decision boundary: $y(\boldsymbol{x}) = 0$, which corresponds to a $(D-1)$-dimensional hyperplane within the D-dimensional input space.

- $\boldsymbol{w}$ is orthogonal to every vector lying within the decision surface: $\forall \boldsymbol{x}_A$ and $\boldsymbol{x}_B$ lie on the decision surface, we have $y(\boldsymbol{x}_A) = y(\boldsymbol{x}_B) = 0 \Rightarrow \boldsymbol{w}^\top(\boldsymbol{x}_A - \boldsymbol{x}_B) = 0$.

- The normal distance from the origin to the decision surface is: $-\frac{w_0}{\|\boldsymbol{w}\|}$.

We need to find α such that $\alpha\boldsymbol{w}$ is on the decision surface, i.e. $\boldsymbol{w}^\top(\alpha\boldsymbol{w}) + w_0 = 0$, thus $\alpha = -w_0/\|\boldsymbol{w}\|^2$, i.e., the normal distance is $-w_0/\|\boldsymbol{w}\|$.

Discriminant functions in 2D

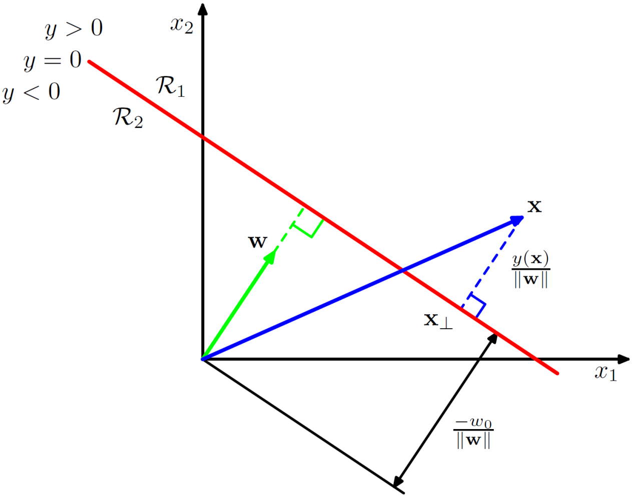


Figure: The decision surface, shown in red, is perpendicular to $\boldsymbol{w}$, and its displacement from the origin is controlled by the bias parameter w_0. Also, the signed orthogonal distance of a general point $\boldsymbol{x}$ from the decision surface is given by $y(\boldsymbol{x})/\|\boldsymbol{w}\|$.

- The value of $y(\boldsymbol{x})$ is a signed measure of the perpendicular distance r of the point $\boldsymbol{x}$ from the decision surface.

- Consider an arbitrary point $\boldsymbol{x}$ and let $\boldsymbol{x}_\perp$ be its orthogonal projection onto the decision surface, so that

$$\boldsymbol{x} = \boldsymbol{x}_\perp + r\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}, \quad \text{orthogonal decomposition.} \tag{1}$$

- Multiplying both sides of this result by $\boldsymbol{w}^\top$ and adding w_0, and making use of $y(\boldsymbol{x}) = \boldsymbol{w}^\top\boldsymbol{x} + w_0$ and $y(\boldsymbol{x}_\perp) = \boldsymbol{w}^\top\boldsymbol{x}_\perp + w_0 = 0$, we have

$$r = \frac{y(\boldsymbol{x})}{\|\boldsymbol{w}\|}, \quad \text{distant formula.} \tag{2}$$

- It is sometimes convenient to use a more compact notation in which we introduce a dummy 'input' value $x_0 = 1$ and then define $\tilde{\boldsymbol{w}} = (w_0, \boldsymbol{w})$ and $\tilde{\boldsymbol{x}} = (x_0, \boldsymbol{x})$ so that

$$y(\boldsymbol{x}) = \tilde{\boldsymbol{w}}^\top \tilde{\boldsymbol{x}}. \tag{3}$$

- In this case, the decision surfaces are D-dimensional hyperplanes passing through the origin of the $(D+1)$-dimensional expanded input space.

How to generalize the discriminant function to multiple classes?

Multiple classes: Infeasible approaches

- One-versus-the-rest: combines $K - 1$ binary classifiers, each of which separate points in a particular class $\mathcal{C}_k$ from points not in that class.

- One-versus-one: uses $K(K-1)/2$ binary discriminant functions, one for every possible pair of classes.

Multiple classes: Infeasible approaches

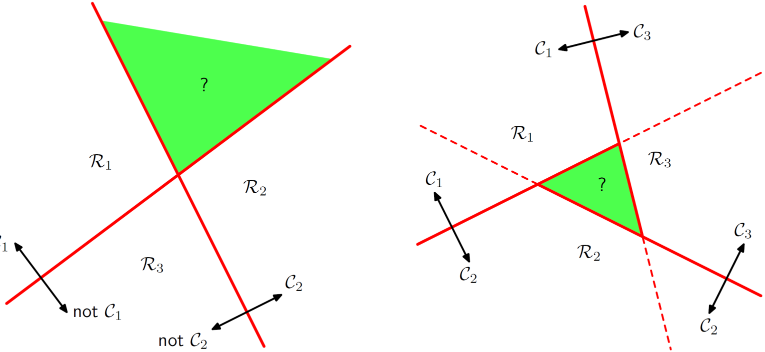


Figure: Left: the use of two discriminants designed to distinguish points in class $\mathcal{C}_k$ from points not in class $\mathcal{C}_k$. Right: three discriminant functions each of which is used to separate a pair of classes $\mathcal{C}_k$ and $\mathcal{C}_j$. Ambiguous regions is shown in green.

- Consider a single K-class discriminant comprising K linear functions of the form

$$y_k(\boldsymbol{x}) = \boldsymbol{w}_k^\top \boldsymbol{x} + w_{k0}. \tag{4}$$

Then $\boldsymbol{x} \to \mathcal{C}_k$ if $y_k(\boldsymbol{x}) > y_j(\boldsymbol{x})$ for all $j \neq k$.

- The decision boundary between class $\mathcal{C}_k$ and class $\mathcal{C}_j$ is therefore given by $y_k(\boldsymbol{x}) = y_j(\boldsymbol{x})$ and hence corresponds to a $(D-1)$-dimensional hyperplane defined by

$$(\boldsymbol{w}_k - \boldsymbol{w}_j)^\top \boldsymbol{x} + (w_{k0} - w_{j0}) = 0. \tag{5}$$

This has the same form as the decision boundary for the two-class case.

- The decision regions are always singly connected and convex.

- $\boldsymbol{x}_A$ and $\boldsymbol{x}_B$ both of which lie inside decision region $\mathcal{R}_k$. Any point $\hat{\boldsymbol{x}}$ that lies on the line connecting $\boldsymbol{x}_A$ and $\boldsymbol{x}_B$ can be expressed in the form

$$\hat{\boldsymbol{x}} = \lambda \boldsymbol{x}_A + (1-\lambda)\boldsymbol{x}_B, \text{ where } 0 \leq \lambda \leq 1. \tag{6}$$

From the linearity of the discriminant functions, it follows that

$$y_k(\hat{\boldsymbol{x}}) = \lambda y_k(\boldsymbol{x}_A) + (1-\lambda) y_k(\boldsymbol{x}_B). \tag{7}$$

Because both $\boldsymbol{x}_A$ and $\boldsymbol{x}_B$ lie inside $\mathcal{R}_k$, it follows that $y_k(\boldsymbol{x}_A) > y_j(\boldsymbol{x}_A)$, and $y_k(\boldsymbol{x}_B) > y_j(\boldsymbol{x}_B)$, for all $j \neq k$, and hence $y_k(\hat{\boldsymbol{x}}) > y_j(\hat{\boldsymbol{x}})$, and so $\hat{\boldsymbol{x}}$ also lies inside $\mathcal{R}_k$. Thus $\mathcal{R}_k$ is singly connected and convex.

How to learn $\boldsymbol{w}_k$? A least square approach

- $$y_k(\boldsymbol{x}) = \boldsymbol{w}_k^\top \boldsymbol{x} + w_{k0}, \quad k = 1, \cdots, K \qquad \Leftrightarrow \qquad \boldsymbol{y}(\boldsymbol{x}) = \tilde{\boldsymbol{W}}^\top \tilde{\boldsymbol{x}},$$

  where $\tilde{\boldsymbol{W}}$ is a matrix whose k-th column comprises the $D+1$-dimensional vector $\tilde{\boldsymbol{w}}_k = (w_{k0}, \boldsymbol{w}_k^\top)^\top$ and $\tilde{\boldsymbol{x}}$ is the corresponding augmented input vector $(1, \boldsymbol{x}^\top)^\top$ with a dummy input $x_0 = 1$. A new input $\boldsymbol{x}$ is then assigned to the class for which the output $y_k = \tilde{\boldsymbol{w}}_k^\top \tilde{\boldsymbol{x}}$ is largest.

How to learn $\boldsymbol{w}_k$? A least square approach

- Consider a training data set $\{\boldsymbol{x}_n, \boldsymbol{t}_n\}$ where $n = 1, \cdots, N$, and define a matrix $\boldsymbol{T}$ whose n-th row is the vector $\boldsymbol{t}_n^\top$, together with a matrix $\tilde{\boldsymbol{X}}$ whose n-th row is $\tilde{\boldsymbol{x}}_n^\top$. The sum-of-squares error function can then be written as

$$E_D(\tilde{\boldsymbol{W}}) = \frac{1}{2}\mathrm{Tr}\left\{(\tilde{\boldsymbol{X}}\tilde{\boldsymbol{W}} - \boldsymbol{T})^\top(\tilde{\boldsymbol{X}}\tilde{\boldsymbol{W}} - \boldsymbol{T})\right\}. \tag{8}$$

- Setting the derivative with respect to $\tilde{\boldsymbol{W}}$ to zero, and rearranging, we then obtain the solution for $\tilde{\boldsymbol{W}}$ in the form

$$\tilde{\boldsymbol{W}} = (\tilde{\boldsymbol{X}}^\top\tilde{\boldsymbol{X}})^{-1}\tilde{\boldsymbol{X}}^\top\boldsymbol{T} = \tilde{\boldsymbol{X}}^\dagger\boldsymbol{T}, \tag{9}$$

where $\tilde{\boldsymbol{X}}^\dagger$ is the pseudo-inverse of the matrix $\tilde{\boldsymbol{X}}$. We then obtain the discriminant function in the form

$$\boldsymbol{y}(\boldsymbol{x}) = \tilde{\boldsymbol{W}}^\top\tilde{\boldsymbol{x}} = \boldsymbol{T}^\top\left(\tilde{\boldsymbol{X}}^\dagger\right)^\top\tilde{\boldsymbol{x}}. \tag{10}$$

Probabilistic Generative Models

- Probabilistic view of classification: we *model the class-conditional densities* $p(\boldsymbol{x}|\mathcal{C}_k)$, *as well as the class priors* $p(\mathcal{C}_k)$, and then use these to compute posterior probabilities $p(\mathcal{C}_k|\boldsymbol{x})$ through Bayes' theorem.

- Consider first of all the case of two classes. The posterior probability for class $\mathcal{C}_1$ can be written as

$$p(\mathcal{C}_1|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\boldsymbol{x}|\mathcal{C}_1)p(\mathcal{C}_1) + p(\boldsymbol{x}|\mathcal{C}_2)p(\mathcal{C}_2)} = \frac{1}{1+\exp(-a)} = \sigma(a) \tag{11}$$

where we have defined

$$a = \ln \frac{p(\boldsymbol{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\boldsymbol{x}|\mathcal{C}_2)p(\mathcal{C}_2)} \tag{12}$$

and $\sigma(a)$ is the *logistic sigmoid* function defined by

$$\sigma(a) = \frac{1}{1+\exp(-a)}. \tag{13}$$

- The inverse of the logistic sigmoid is given by

$$a = \ln\left(\frac{\sigma}{1-\sigma}\right) \tag{14}$$

and is known as the *logit* function.

- For the case of $K > 2$ classes, we have

$$p(\mathcal{C}_k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{\sum_j p(\boldsymbol{x}|\mathcal{C}_j)p(\mathcal{C}_j)} = \frac{\exp(a_k)}{\sum_j \exp(a_j)} \tag{15}$$

which is known as the *normalized exponential* and can be regarded as a multiclass generalization of the logistic sigmoid. Here the quantities a_k are defined by

$$a_k = \ln p(\boldsymbol{x}|\mathcal{C}_k)p(\mathcal{C}_k). \tag{16}$$

- The normalized exponential is also known as the *softmax function*, as it represents a smoothed version of the 'max' function because, if $a_k \gg a_j$ for all $j \neq k$, then $p(\mathcal{C}_k|\boldsymbol{x}) \approx 1$, and $p(\mathcal{C}_j|\boldsymbol{x}) \approx 0$.

- Assume that the class-conditional densities are Gaussian with the same covariance matrix, i.e.

$$p(\boldsymbol{x}|\mathcal{C}_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} \exp\Big\{ -\frac{1}{2}(\boldsymbol{x} - \mu_k)^\top \Sigma^{-1} (\boldsymbol{x} - \mu_k) \Big\}, \; k = 1, 2. \tag{17}$$

Let us consider the posterior probabilities for two classes, from (11) and (12), we have

$$p(\mathcal{C}_1|\boldsymbol{x}) = \sigma(\boldsymbol{w}^\top \boldsymbol{x} + w_0) \tag{18}$$

where we have defined

$$\boldsymbol{w} = \Sigma^{-1}(\mu_1 - \mu_2); \quad w_0 = -\frac{1}{2}\mu_1^\top \Sigma^{-1} \mu_1 + \frac{1}{2}\mu_2^\top \Sigma^{-1} \mu_2 + \ln \frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}. \tag{19}$$

- How to estimate $\pi, \mu_1, \mu_2, \Sigma$?

- Observation: $\{\boldsymbol{x}_n, t_n\}_{n=1}^N$. Here $t_n = 1$ denotes class $\mathcal{C}_1$ and $t_n = 0$ denotes class $\mathcal{C}_2$.

- Let the prior class probability $p(\mathcal{C}_1) = \pi$ and $p(\mathcal{C}_2) = 1 - \pi$. By Bayes' theorem we have

$$p(\boldsymbol{x}_n, \mathcal{C}_1) = p(\mathcal{C}_1)p(\boldsymbol{x}_n|\mathcal{C}_1) = \pi\mathcal{N}(\boldsymbol{x}_n|\mu_1, \Sigma);$$
$$p(\boldsymbol{x}_n, \mathcal{C}_2) = p(\mathcal{C}_2)p(\boldsymbol{x}_n|\mathcal{C}_2) = (1 - \pi)\mathcal{N}(\boldsymbol{x}_n|\mu_2, \Sigma).$$

- Thus the likelihood function is given by

$$p(\boldsymbol{t}|\pi, \mu_1, \mu_2, \Sigma) = \prod_{n=1}^N \left[\pi\mathcal{N}(\boldsymbol{x}_n|\mu_1, \Sigma)\right]^{t_n} \left[(1 - \pi)\mathcal{N}(\boldsymbol{x}_n|\mu_2, \Sigma)\right]^{1-t_n}, \tag{20}$$

where $\boldsymbol{t} = (t_1, \cdots, t_N)^\top$.

- Instead of maximize the likelihood, we consider the log-likelihood!
- π:

$$\max_{\pi} \sum_{n=1}^{N} \{t_n \ln \pi + (1 - t_n) \ln(1 - \pi)\},$$

therefore,

$$\pi = \frac{1}{N} \sum_{n=1}^{N} t_n = \frac{N_1}{N} = \frac{N_1}{N_1 + N_2}, \quad N_i = \#C_i.$$

- μ_1:

$$\sum_{n=1}^{N} t_n \ln \mathcal{N}(\boldsymbol{x}_n | \mu_1, \Sigma) = -\frac{1}{2} \sum_{n=1}^{N} t_n (\boldsymbol{x}_n - \mu_1)^\top \Sigma^{-1} (\boldsymbol{x}_n - \mu_1) + \text{const},$$

therefore,

$$\mu_1 = \frac{1}{N_1} \sum_{n=1}^{N} t_n \boldsymbol{x}_n.$$

- Similarly, $\mu_2 = \frac{1}{N_2} \sum_{n=1}^{N} (1 - t_n) \boldsymbol{x}_n$. How to find Σ?

Probabilistic Discriminative Models

- So far, we have modeled

$$p(\mathcal{C}_1|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\boldsymbol{x}|\mathcal{C}_1)p(\mathcal{C}_1) + p(\boldsymbol{x}|\mathcal{C}_2)p(\mathcal{C}_2)} = \frac{1}{1+\exp(-a)} = \sigma(a),$$

for a wide choice of class-conditional distributions $p(\boldsymbol{x}|\mathcal{C}_k)$. For specific choices of the class-conditional densities $p(\boldsymbol{x}|\mathcal{C}_k)$, we have used maximum likelihood to determine the parameters of the densities as well as the class priors $p(C_k)$ and then used Bayes' theorem to find the posterior class probabilities.

- We can also generalize $\boldsymbol{x}$ to $\phi(\boldsymbol{x})$ with ϕ being a basis function, resulting in generalized linear models. Note that classes that are linearly separable in the feature space $\phi(\boldsymbol{x})$ need not be linearly separable in the original observation space $\boldsymbol{x}$.

- Generative modeling. Indirectly find the parameters of a generalized linear model, by *fitting class-conditional densities and class priors separately* and then applying Bayes' theorem. We could take such a model and generate synthetic data by drawing values of $\boldsymbol{x}$ from the marginal distribution $p(\boldsymbol{x})$.

- We need to find $p(\boldsymbol{x}|\mathcal{C}_k)$ and $p(\mathcal{C}_k)$. We can then perform sample $p(\boldsymbol{x}|\mathcal{C}_k)$.

- Discriminative modeling. Directly maximize the likelihood function defined through the conditional distribution $p(\mathcal{C}_k|\boldsymbol{x})$. It may also lead to improved predictive performance, particularly when the class-conditional density assumptions give a poor approximation to the true distributions.

- We only care about $p(\mathcal{C}_k|\boldsymbol{x})$.

• Let us consider two-class classification problem, the posterior probability of class $\mathcal{C}_1$ can be written as a logistic sigmoid acting on a linear function of the feature vector ϕ so that

$$p(\mathcal{C}_1|\phi) = y(\phi) = \sigma(\boldsymbol{w}^\top \phi) \tag{21}$$

with $p(\mathcal{C}_2|\phi) = 1 - p(\mathcal{C}_1|\phi)$. Here $\sigma(\cdot)$ is the *logistic sigmoid* function. This model is known as *logistic regression*, which is a classification model.

- Maximum likelihood for parameters estimation. First note that for the sigmoid function, we have

$$\frac{d\sigma}{da} = \sigma(1-\sigma). \tag{22}$$

- For a data set $\{\phi_n, t_n\}$, where $t_n \in \{0,1\}$ and $\phi_n = \phi(\boldsymbol{x}_n)$, with $n = 1, \cdots, N$, the likelihood function is

$$p(\boldsymbol{t}|\boldsymbol{w}) = \prod_{n=1}^{N} y_n^{t_n}(1-y_n)^{1-t_n}, \tag{23}$$

where $\boldsymbol{t} = (t_1, \cdots, t_N)^\top$ and $y_n = p(\mathcal{C}_1|\phi_n)$.

- Taking the negative logarithm of the likelihood, resulting in the *cross-entropy* error:

$$E(\boldsymbol{w}) = -\ln p(\boldsymbol{t}|\boldsymbol{w}) = -\sum_{n=1}^{N}\{t_n \ln y_n + (1 - t_n)\ln(1 - y_n)\}, \tag{24}$$

where $y_n = \sigma(a_n)$ and $a_n = \boldsymbol{w}^\top \phi_n$.

- Taking the gradient of the error function with respect to $\boldsymbol{w}$, we obtain

$$\nabla E(\boldsymbol{w}) = \sum_{n=1}^{N}(y_n - t_n)\phi_n, \tag{25}$$

where we have used the fact that $\frac{d\sigma}{da} = \sigma(1 - \sigma)$.

• In our discussion of generative models for multiclass classification, we have seen that for a large class of distributions, the posterior probabilities are given by a softmax transformation of linear functions of the feature variables, so that

$$p(\mathcal{C}_k|\phi) = y_k(\phi) = \frac{\exp(a_k)}{\sum_j \exp(a_j)}, \tag{26}$$

where the 'activations' a_k are given by

$$a_k = \boldsymbol{w}_k^\top \phi. \tag{27}$$