Lecture Outline

Reading: Chapter 4 - Linear Models for Classification

This lecture covers

- Inference and decision (1.5.4)
- Probabilistic generative models (4.2)
- Probabilistic generative models under Gaussian assumption (4.2.1)
- Maximum likelihood solution for $\mathbf{x}$ (4.2.2)

1.5.4 Inference and Decision

We have broken the classification problem down into two separate stages: 1) the *inference stage* in which we use training data to learn a model for $p(\mathcal{C}_k|\mathbf{x})$ and 2) the *decision* stage in which we use these posterior probabilities to make optimal class assignments.

1) We first solve the inference problem by determining the class-conditional densities, $p(\mathbf{x}|\mathcal{C}_k)$ for each class $\mathcal{C}_k$ individually and separately, infering the prior class probabilities, $p(\mathcal{C}_k)$. Then use Bayes' theorem in the form

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\mathbf{x})} \tag{1}$$

to find the posterior class probabilities, $p(\mathcal{C}_k|\mathbf{x})$. The normalization factor can be found via

$$p(\mathbf{x}) = \sum_k p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k) \tag{2}$$

2) In the decision stage, for a given $\mathbf{x}$, we use $p(\mathcal{C}_k|\mathbf{x})$ to make the optimal class assignment.

Figure 1: inference and decision

Approaches that explicitly or implicitly model the distribution of inputs as well as output are known as *generative* models, because by sampling from them it is possible to generate synthetic data points in the input space.

4.2 Probabilistic Generative Models

We turn next to a probabilistic view of classification and show how models with linear decision surfaces arise from simple assumptions about the distribution of the data. Here we shall adopt a generative approach in

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

Consider the case of a binary classifier, i.e. two classes. The posterior probability of $\mathcal{C}_1$ can be written as

$$\begin{aligned} p(\mathcal{C}_1|\mathbf{x}) &= \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1) + p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)} \\ &= \frac{1}{1+\exp(-a)} \\ &\equiv \sigma(a) \end{aligned} \tag{3}$$

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

$$a = \ln \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)}. \tag{4}$$

As we will see shortly, the use of the logistic sigmoid is significant because if we assume the class conditional densities are normal, $a(\mathbf{x})$ takes on a simple, linear functional form, i.e. the posterior probability is governed by a (sigmoidal) linear model.

Equation (4) is easily found:

$$1 + \exp(-a) = \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1) + p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)}{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}, \tag{5}$$

$$\begin{aligned} \exp(-a) &= \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1) + p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)}{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)} - 1 \\ &= \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1) + p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)}{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)} - \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)} \\ &= \frac{p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)}{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}, \end{aligned} \tag{6}$$

and we easily see (4).

For multi-class classifiers, i.e. $K > 2$ classes, we have

$$\begin{aligned} p(\mathcal{C}_k|\mathbf{x}) &= \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{\sum_j p(\mathbf{x}|\mathcal{C}_j)p(\mathcal{C}_j)} \\ &= \frac{\exp(a_k)}{\sum_j \exp(a_j)} \end{aligned} \tag{7}$$

where

$$a_k = \ln\left[p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)\right]. \tag{8}$$

The posterior as expressed is known as the *normalized exponential* and can be regarded as a multiclass generalization of the logistic sigmoid. 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 $p(\mathcal{C}_k|\mathbf{x}) \simeq 1$ and $p(\mathcal{C}_j|\mathbf{x}) \simeq 0$.

4.2.2 Continuous Inputs

Let us assume that the class-conditional densities, $p(\mathbf{x}|\mathcal{C}_1)$ and $p(\mathbf{x}|\mathcal{C}_2)$ are **Gaussian** and then explore the resulting form of the posterior probabilities (3).

To start, we shall assume that both classes have the same covariance matrix, $\boldsymbol{\Sigma}$, i.e. **shared covariance**. Thus the class conditional density for class $\mathcal{C}_k$ is given by

$$\begin{aligned} p(\mathbf{x}|\mathcal{C}_k) &= \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}) \\ &= \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right\}. \end{aligned} \tag{9}$$

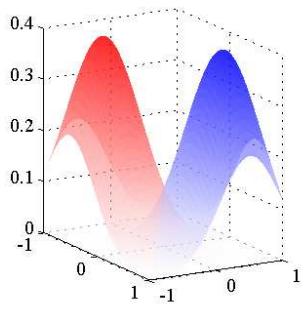
Substituting (9) into (4) we have

$$\begin{aligned} a &= \ln \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)} \\ &= \ln p(\mathbf{x}|\mathcal{C}_1) + \ln p(\mathcal{C}_1) - \ln p(\mathbf{x}|\mathcal{C}_2) - \ln p(\mathcal{C}_2) \\ &= \ln\left[\frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}}\right] - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_1) + \ln p(\mathcal{C}_1) + \\ &\quad - \ln\left[\frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}}\right] + \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_2)^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_2) - \ln p(\mathcal{C}_2) \\ &= \boldsymbol{\mu}_1^T\boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_1^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2^T\boldsymbol{\Sigma}^{-1}\mathbf{x} + \frac{1}{2}\boldsymbol{\mu}_2^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2 + \ln\frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)} \\ &= \underbrace{\left(\boldsymbol{\mu}_1^T - \boldsymbol{\mu}_2^T\right)\boldsymbol{\Sigma}^{-1}}_{\mathbf{w}^T}\mathbf{x} \underbrace{- \frac{1}{2}\boldsymbol{\mu}_1^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2 + \ln\frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}}_{w_0} \\ &= \mathbf{w}^T\mathbf{x} + w_0 \end{aligned} \tag{10}$$

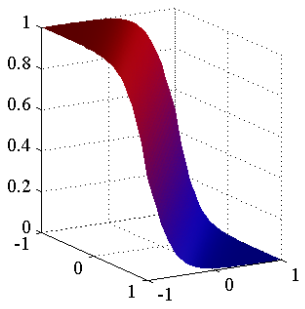
From (3), we see that the posterior probability is governed by a (sigmoidal) linear model (as desired):

$$p(\mathcal{C}_1|\mathbf{x}) = \sigma(a) = \sigma(\mathbf{w}^T\mathbf{x} + w_0). \tag{11}$$

We see that the quadratic terms in $\mathbf{x}$ from the exponents of the Gaussian densities have cancelled (due to the shared covariance assumption) leading to a linear function of $\mathbf{x}$ in the argument of the logistic sigmoid. The result is illustrated for the case of a two-dimensional input space in Figure (a). The resulting decision boundaries correspond to surfaces along which the posterior probabilities $p(\mathcal{C}_k|\mathbf{x})$ are constant and so will be given by linear functions of $\mathbf{x}$ and therefore the decision boundaries are linear in input space. The prior probabilities $p(\mathcal{C}_k)$ enter only through the bias parameter w_0 so that changes in the priors have the effect of making parallel shifts of the decision boundary.



(a) $p(\mathbf{x}|\mathcal{C}_1)$ (red) and $p(\mathbf{x}|\mathcal{C}_2)$ (blue)



(b) $p(\mathcal{C}_1|\mathbf{x})$

Note that this solution for $\mathbf{w}$ and w_0 doesn't tell us how to actually get $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$, $\boldsymbol{\Sigma}$, $p(\mathcal{C}_1)$, $p(\mathcal{C}_2)$ from the training data. We'll do this in the next section.

For the general case of K classes, using the softmax form of (8) we have,

$$a_k = \mathbf{w}_k^T \mathbf{x} + w_{k0} \tag{12}$$

where

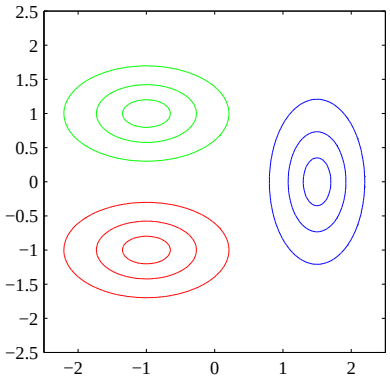
$$\mathbf{w}_k = \mathbf{\Sigma}^{-1} \mu_k \tag{13}$$

and

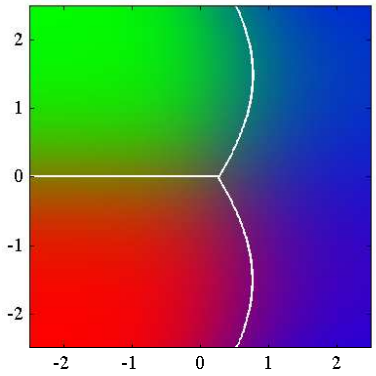
$$w_{k0} = -\frac{1}{2} \boldsymbol{\mu}_k^T \mathbf{\Sigma}^{-1} \boldsymbol{\mu}_k + \ln p(\mathcal{C}_k). \tag{14}$$

We see again that the a_k are linear functions of $\mathbf{x}$ as a consequence of the cancellation of the quadratic terms due to the shared covariance matrices.

If we relax the assumption of a shared covariance matrix and allow each class-conditional density $p(\mathbf{x}|\mathcal{C}_k)$ to have its own covariance matrix $\mathbf{\Sigma}_k$, then the earlier cancellations no longer occur, and we will obtain quadratic functions of $\mathbf{x}$, giving rise to a *quadratic discriminant*.



(c) red and green class have shared covariance, blue covariance is different



(d) linear decision boundary between red and green; quadratic between other pairs

4.2.2 Maximum Likelihood Solution

We still need to solve for the training data's distributional parameters, i.e. $\{\boldsymbol{\mu}_1,\ \boldsymbol{\mu}_2,\ \mathbf{\Sigma},\ p(\mathcal{C}_1),\ p(\mathcal{C}_2)\}$ so that we can obtain $\mathbf{w}$ and w_0 for our binary classifier. Our approach will, of course, be to maximize the likelihood with respect to each of these parameters.

Suppose we have a data set $\{\mathbf{x}_n, t_n\}$ where $n = 1, \ldots, N$. Here $t_n = 1$ denotes class $\mathcal{C}_1$ and $t_n = 0$ denotes class $\mathcal{C}_2$. We also denote the prior class probabilities as $p(\mathcal{C}_1) = \pi$ and $p(\mathcal{C}_2) = 1 - \pi$. Note that from (10), finding $\mathbf{w}$ and w_0 means finding $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$, $\mathbf{\Sigma}$, and π from the training data.

The likelihood function is given by

$$p(\mathbf{t}, \mathbf{X}|\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \mathbf{\Sigma}) = \prod_{n=1}^{N} \left[\pi \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_1, \mathbf{\Sigma})\right]^{t_n} \left[(1-\pi)\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_2, \mathbf{\Sigma})\right]^{1-t_n} \tag{15}$$

where $\mathbf{t} = (t_1, \ldots, t_N)^T$. Setting the derivative with respect to the proper variables and solving leads to the values necessary to compute $\mathbf{w}$ and w_0. These are all intuitive:

$$\pi = \frac{N_1}{N_1 + N_2} \tag{16}$$

where N_1 is the number of datapoints in $\mathcal{C}_1$, N_2 is the number of datapoints in $\mathcal{C}_2$, and $N = N_1 + N_2$;

$$\boldsymbol{\mu}_1 = \frac{1}{N_1} \sum_{\mathbf{x}_j \in \mathcal{C}_1} \mathbf{x}_j = \frac{1}{N_1} \sum_{n=1}^{N} t_n \mathbf{x}_n; \tag{17}$$

$$\boldsymbol{\mu}_2 = \frac{1}{N_2} \sum_{\mathbf{x}_j \in \mathcal{C}_2} \mathbf{x}_j = \frac{1}{N_2} \sum_{n=1}^{N} (1 - t_n) \mathbf{x}_n; \tag{18}$$

and

$$\boldsymbol{\Sigma} = \frac{N_1}{N} \mathbf{S}_1 + \frac{N_2}{N} \mathbf{S}_2, \tag{19}$$

i.e. weighted average of the sample covariance matrices for each class where

$$\mathbf{S}_1 = \frac{1}{N_1} \sum_{\mathbf{x}_j \in \mathcal{C}_1} (\mathbf{x}_n - \boldsymbol{\mu}_1) (\mathbf{x}_n - \boldsymbol{\mu}_1)^T \tag{20}$$

and

$$\mathbf{S}_2 = \frac{1}{N_2} \sum_{\mathbf{x}_j \in \mathcal{C}_2} (\mathbf{x}_n - \boldsymbol{\mu}_2) (\mathbf{x}_n - \boldsymbol{\mu}_2)^T. \tag{21}$$

Recap: Given a training data set $\{\mathbf{x}_n\}$ with corresponding labels $\{t_n\}$ ($1 \to \mathcal{C}_1$ or $0 \to \mathcal{C}_2$) and assuming the class-conditional densities $p(\mathbf{x}|\mathcal{C}_k)$ are Gaussian and the covariance matrix is same (shared), we use the above to compute π, $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$, $\boldsymbol{\Sigma}$. Then use (10) to compute $\mathbf{w}$, w_0. For a given test point $\mathbf{x}'$, the posterior is given by $p(\mathcal{C}_1|\mathbf{x}') = \sigma(\mathbf{w}^T\mathbf{x}' + w_0)$, $p(\mathcal{C}_2|\mathbf{x}') = 1 - p(\mathcal{C}_1|\mathbf{x}')$ and the classification decision is based on the maximum a posterior (MAP) rule.

Final Note: (p. 225) Linear models have useful analytical and computation properties but practical applicability is limitied by the curse of dimensionality. In order to apply such models to large scale problems, it is neccessary to adapt basic functions to the data. For this we will investigate in Chapter 5, Neural Networks.