

# IKM 1st Practice

## 1 Introduction and Overview

### 1.1 Bayesian Approach to Classification

Some classifiers rely on probability theory. Essentially they want to predict a probability of class ( $y$ ) given features ( $x$ ) –  $p(y|x)$ . But as you can imagine, there is hardly any application where we could sample every possible  $x$  so many times to have a reasonable statistic about  $y$ . Moreover, if  $x$  is continuous, the problem becomes intractable. Bayes rule is a straightforward application of one of the axioms of conditional probability. It states that the unreachable a posteriori probability ( $p(y|x)$ ) can be expressed as a fraction:

$$p(y, x) = p(x, y) \quad (1)$$

$$p(y|x)p(x) = p(x|y)p(y) \quad (2)$$

$$p(y|x) = \frac{\overbrace{p(x|y)}^{\text{likelihood}} \overbrace{p(y)}^{\text{prior}}}{\underbrace{p(x)}_{\text{marginal likelihood}}} \quad (3)$$

The divisor – marginal likelihood can be omitted in the comparison between two (or more) a posteriori probabilities because it acts as a multiplicative constant. For this reason, a bayesian approach to binary classification problems can be simplified into

$$\text{prediction} = \arg \max_i p(x|y_i)p(y_i). \quad (4)$$

One question remains: How to model the likelihood ( $p(x|y)$ ). We could try complex statistical models like a Gaussian mixture model (see first lecture notes), but it is usually computationally expensive and may create additional problems like overtraining. A good rule of thumb is to use simpler models first and then improve based on results. The central limit theorem states that if a random variable (in our case  $\sim p(x|y)$ ) is the outcome of a summation of a number of independent random variables, its pdf approaches a normal distribution. It is quite sensible to model likelihood as a normal distribution - it has only a few parameters and theoretically nice properties. The multivariate normal distribution is defined as

$$\mathcal{N}(x|\mu, \Sigma) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)\right). \quad (5)$$

Where  $\Sigma$  is a (co)variance matrix ( $D \times D$ ),  $\mu$  is a  $D$ -dimensional mean vector. Combining equation 4 and 5, we obtain

$$\text{prediction} = \arg \max_i p(y_i|x) = \arg \max_i p(x|y_i)p(y_i) = \arg \max_i \mathcal{N}(x|\mu_i, \Sigma_i)p(y_i). \quad (6)$$

## 1.2 Quadratic Discriminant Analysis

Let's investigate the properties of decision hyperplanes of such a classifier. In the case of binary classification, the decision hyperplane can be obtained by the comparison of two probabilities:

$$\mathcal{N}(\mu_0, \Sigma_0)p(y_0) \stackrel{?}{<} \mathcal{N}(\mu_1, \Sigma_1)p(y_1) \quad (7)$$

$$\frac{\mathcal{N}(\mu_0, \Sigma_0)p(y_0)}{\mathcal{N}(\mu_1, \Sigma_1)p(y_1)} \stackrel{?}{<} 1 \quad (8)$$

$$\underbrace{x^T (-\Sigma_0^{-1} + \Sigma_1^{-1}) x}_{\text{quadratic term}} + \underbrace{2 (\mu_0^T \Sigma_0^{-1} - \mu_1^T \Sigma_1^{-1}) x}_{\text{linear term}} - \underbrace{\mu_0^T \Sigma_0^{-1} \mu_0 + \mu_1^T \Sigma_1^{-1} \mu_1 - c}_{\text{constant term}} \stackrel{?}{<} 0 \quad (9)$$

Where

$$c = \log(|\Sigma_0|) - \log(|\Sigma_1|) - 2 \log(p(y_0)) + 2 \log(p(y_1)).$$

So the resulting method – quadratic discriminant analysis (QDA) produce quadratic classifier, and the decision boundaries of such a classifier would be hyperquadrics (i.e., ellipsoids, parabolas, hyperbolas, etc.); see [wiki](#) for pictures. In practice, we keep only a discriminative function ( $q_i(x)$ ) defined as

$$q_i^{\text{QDF}}(x) = -x^T \Sigma_i^{-1} x + 2 \mu_i^T \Sigma_i^{-1} x - \mu_i^T \Sigma_i^{-1} \mu_i - \log(|\Sigma_i|) + 2 \log(p(y_i)) \quad (10)$$

The final classification is as simple as

$$\text{prediction} = \arg \max_i q_i(x) \quad (11)$$

But there is a catch: It makes sense to use QDA only if we have enough data. Most of the time, it is sensible to use at least one sample for each model's free variable. In this case, the number of free parameters<sup>1</sup> is

$$C \left( \underbrace{\frac{D(D+1)}{2}}_{\Sigma} + \underbrace{D}_{\mu} \right) + \underbrace{C-1}_{p(y)}$$

$C$  is the number of classes.

Let's put it into perspective: The popular task of recognizing handwritten digits (see figure 1) has  $28 \times 28 = 784$  pixels per sample and ten classes. So we would need at least 315,569 examples (1 example per parameter) while only 60,000 are provided for training. Without any dimensionality reduction technique, the classifier will make an arbitrary decision about boundaries which probably hinders our efforts.

## 1.3 Linear Discriminant Analysis

We can modify QDA to reduce the number of free variables. Let's assume that  $\forall_i (\Sigma_i = \Sigma)$ . Then the discriminative function in eq. 10 become linear because the quadratic term in eq. 21 will be canceled out. The linear discriminative function is defined as

$$q_i^{\text{LDF}}(x) = 2 \mu_i^T \Sigma^{-1} x - \mu_i^T \Sigma^{-1} \mu_i + 2 \log(p(y_i)). \quad (12)$$

This method is called Linear discriminative analysis<sup>2</sup> (LDA). For LDA, the required number of samples is significantly reduced to

$$\underbrace{DC}_{\mu} + \underbrace{C-1}_{p(y)}$$

The covariance matrix  $\Sigma$  is not an effective parameter because it is not directly responsible for the prediction. Our MNIST example then needs 7,849 training examples which is seven times lower than what is being provided by the dataset.

The question is whether risking with a model that can make a random choice about decision surface is better than assuming something about data that is not true and hoping it will not make a difference.

<sup>1</sup>One additional class can be spared by subtraction of a particular discriminative function from all the other ones.

<sup>2</sup>Fisher's linear discriminant analysis is something different.

## 1.4 Gaussian Naive Bayes

Let's assume that the covariance is diagonal – i.e., one feature does not affect another. The classifier is still quadratic; equation 21 can be split by dimension. In other words, the classifier creates likelihoods for each dimension separately, assuming conditional independence given class. Then it uses the product rule – a product of probabilities of independent events is a probability of the joint event (eq. 13). If all dataset features are independent, the QDA produce the same result as Gaussian Naive Bayes. Unfortunately, this is hardly true; for example, let's consider dataset classifying people, then weight and height are expected to be dependent features because taller individuals are generally heavier. On the other hand, violating the assumption does not mean that the classifier won't work entirely, and many techniques make the data at least linearly independent, e.g. Singular Value Decomposition and its generalized variants.

$$\hat{y} = \arg \min_i p(y_i) \prod_{j=1}^D p(x_j|y_i) = \arg \min_i p(y_i) \prod_{j=1}^D \frac{1}{\sqrt{2\pi}\sigma_{i,j}} \exp -\frac{1}{2} \left( \frac{x_j - \mu_{i,j}}{\sigma_{i,j}} \right)^2 \quad (13)$$

## 1.5 General Naive Bayes

The Naive Bayes is a more general classifier to the Gaussian Naive Bayes. The difference is in the statistical modelling – the Naive Bayes allows any distribution to be the likelihood, whereas Gaussian Naive Bayes allows only the normal distribution. Other aspects are the same – the Naive Bayes assumes conditional independence for all features given class. Hence there is no consideration for other features when fitting the distributions of the classifier. The Naive Bayes is not restricted to continuous feature spaces; it is a popular classifier whenever binary or categorical variables hold the most information.

$$\hat{y} = \arg \min_i p(y_i) \prod_{j=1}^D p(x_j|y_i) \quad (14)$$

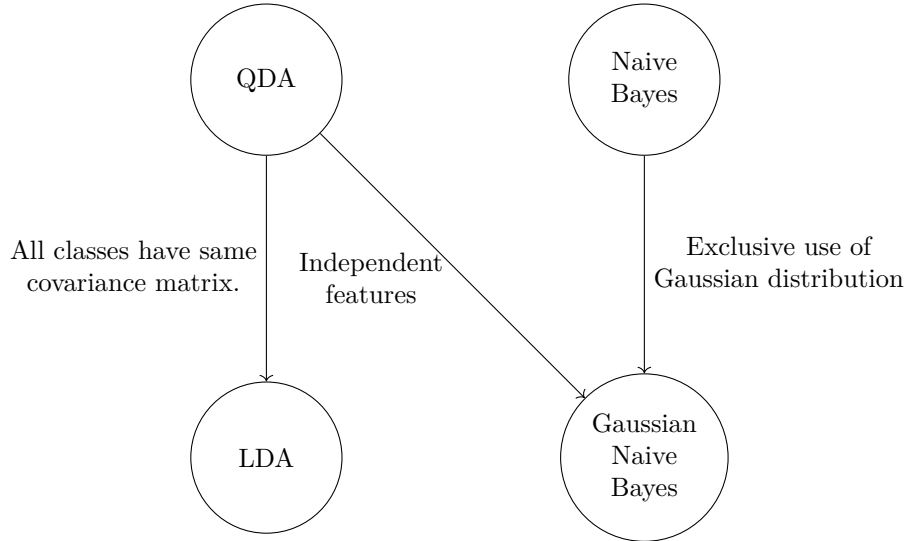


Figure 1: The relationships between all four types of probabilistic classifiers mentioned up until now.

## Appendix A Derivation of Decision Boudaries for Quadratic Discriminant Analysis

$$\frac{\mathcal{N}(\mu_0, \Sigma_0)p(y_0)}{\mathcal{N}(\mu_1, \Sigma_1)p(y_1)} \stackrel{?}{<} 1 \quad (15)$$

$$\frac{\frac{1}{(2\pi)^{D/2}|\Sigma_0|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu_0)^T \Sigma_0^{-1}(x - \mu_0)\right) p(y_0)}{\frac{1}{(2\pi)^{D/2}|\Sigma_1|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu_1)^T \Sigma_1^{-1}(x - \mu_1)\right) p(y_1)} \stackrel{?}{<} 1 \quad (16)$$

$$\begin{aligned} & \left[ \log\left(\frac{1}{(2\pi)^{D/2}|\Sigma_0|^{1/2}}\right) - \frac{1}{2}(x - \mu_0)^T \Sigma_0^{-1}(x - \mu_0) + \log(p(y_0)) \right] \\ & - \left[ \log\left(\frac{1}{(2\pi)^{D/2}|\Sigma_1|^{1/2}}\right) - \frac{1}{2}(x - \mu_1)^T \Sigma_1^{-1}(x - \mu_1) + \log(p(y_1)) \right] \stackrel{?}{<} 0 \end{aligned} \quad (17)$$

$$\begin{aligned} & \left[ -\frac{1}{2} \log(|\Sigma_0|) - \frac{1}{2}(x - \mu_0)^T \Sigma_0^{-1}(x - \mu_0) + \log(p(y_0)) \right] \\ & - \left[ -\frac{1}{2} \log(|\Sigma_1|) - \frac{1}{2}(x - \mu_1)^T \Sigma_1^{-1}(x - \mu_1) + \log(p(y_1)) \right] \stackrel{?}{<} 0 \end{aligned} \quad (18)$$

$$[-(x - \mu_0)^T \Sigma_0^{-1}(x - \mu_0)] - [-(x - \mu_1)^T \Sigma_1^{-1}(x - \mu_1)] \stackrel{?}{<} c \quad (19)$$

$$- [x^T \Sigma_0^{-1} x - 2\mu_0^T \Sigma_0^{-1} x + \mu_0^T \Sigma_0^{-1} \mu_0] + [x^T \Sigma_1^{-1} x - 2\mu_1^T \Sigma_1^{-1} x + \mu_1^T \Sigma_1^{-1} \mu_1] \stackrel{?}{<} c \quad (20)$$

$$\underbrace{x^T (-\Sigma_0^{-1} + \Sigma_1^{-1}) x}_{\text{quadratic term}} + \underbrace{2(\mu_0^T \Sigma_0^{-1} - \mu_1^T \Sigma_1^{-1}) x}_{\text{linear term}} - \underbrace{\mu_0^T \Sigma_0^{-1} \mu_0 + \mu_1^T \Sigma_1^{-1} \mu_1 - c}_{\text{constant term}} \stackrel{?}{<} 0 \quad (21)$$