2 Introduction to classification

To *classify* means that an object or event is ordered into one out of several classes. For example, we could have a Lego robot with a color camera, and the task is to classify the object the robot is looking at into one of several classes. The classes could be e.g. *yellow Lego*, *red Lego*, *blue Lego*, *green carpet*, and *black wall*.

We write the object (or event) as a feature vector $\mathbf{x}$ defined as

$$\mathbf{x}(n) = \begin{pmatrix} x_1(n) \\ x_2(n) \\ \vdots \\ x_D(n) \end{pmatrix} \tag{2.1}$$

where $x_k(n)$ are the features that describe the object. These features are such things that we think are important for the classification task. In the color classification example above, we might use $x_1 = \text{red}, x_2 = \text{green}$, and $x_3 = \text{blue}$. The features can be either continuous or discrete.

The category, i.e. the output from our classifier, is one of several discrete (usually a mutually exclusive and complete set) categories $c_k, k = 1, \ldots, K$.

It is common to include the categories "doubt" ("reject") and "outlier". An outlier is an observation that deviates much from previous seen examples. Doubt is when the classifier is uncertain about which decision to make. We will for the most part ignore the reject option and the outlier option to keep the discussion simple. In applications, however, it is very important to include these two categories. Ripley (1996) and Webb (1999), for instance, discuss this more closely.

The classification is thus a mapping from a feature space $\mathbf{X}^D$ to a category space $\mathbf{C}^K$

$$f : \mathbf{X}^D \to \mathbf{C}^K \tag{2.2}$$

where $\mathbf{X}^D \subset \mathbf{R}^D$ and $\mathbf{C}^K = \{0, 1\}^K$.

A classifier is usually designed for one of the following two purposes:

- To classify new data;
- To analyze data and understand what separates different categories, i.e. what features have a strong discriminant power.

The latter of these is unfortunately often in conflict with the first. A simple classifier is simpler to understand (dissect) than a complex classifier. However, a complex classifier is often better at classifying.

Three examples of classification tasks are listed below:

Cushing's syndrome: Cushing's syndrome is a disease characterized by over-secretrion of cortisol by the adrenal gland (Ripley 1996). There are three types of the disease, which we here denote a, b and c. These three forms depend on the underlying reason for the over-production. Which type a patient suffers from can be determined histo-pathologically, which however is painful. It is possible to classify which type a patient belongs to by measuring the excretion rates (*mg/day*) of tetrahydrocortisone and pregnanetriol. The input signal is in this case 2-dimensional ($D = 2$). The output is 3-dimensional ($K = 3$), denoting the categories a, b and c. (If one also allows the category "unknown", then the output becomes 4-dimensional.)

Color classification for Lego robot: Here the task is to classify what color the robot is looking at (different objects have different colors). By working with normalized *rgb* coefficients, it is possible to make the input signal 2-dimensional. The output space is typically 5-dimensional ($K = 5$) representing the categories *yellow Lego*, *red Lego*, *blue Lego*, *black sideboard*, and *green carpet*.

ALVINN: ALVINN (Autonomous Land Vehicle In a Neural Network) is a system that drives a car (Pomerleau 1993), by responding with steering signals to an input. The input signal is an image, i.e. a matrix, with (e.g.) 625 pixels ($D = 625$) and there are 20 output categories ($K = 20$) which correspond to different steering angles.

2.1 Statistical decision theory

Classification is a decision, one decides to categorize an observation into a category. The final decision of course depends on the consequences of the decision and not just the probability that an observation $\mathbf{x}(n)$ belongs to a given category c_k. Medical applications are excellent examples of this.

Statistical decision theory tells us how we should proceed to make an optimal decision, given that we know the costs associated with our decisions and the probabilities for the different categories.

We use the following notation:

$$\begin{aligned}
p(\mathbf{x}) &= \text{Probability density for } \mathbf{x};\\
p(c_k) &= \text{A priori probability for category } c_k;\\
p(\mathbf{x}|c_k) &= \text{Probability density for all } \mathbf{x} \in c_k;\\
p(c_k|\mathbf{x}) &= \text{A posteriori probability for category } c_k\text{, given } \mathbf{x};\\
p(\mathbf{x}, c_k) = p(c_k, \mathbf{x}) &= \text{Joint probability for observing both } c_k \text{ and } \mathbf{x};\\
\alpha_i &= \text{Decision } i;\\
\lambda(\alpha_i|c_k) = \lambda_{ik} &= \text{Cost incurred when making decision } \alpha_i \text{ if } \mathbf{x} \in c_k.
\end{aligned}$$

We have, of course, that $p(\mathbf{x}, c_k) = p(c_k)p(\mathbf{x}|c_k) = p(\mathbf{x})p(c_k|\mathbf{x}) = p(c_k, \mathbf{x})$, which yields the *Bayes' rule*:

$$p(c_k|\mathbf{x}) = \frac{p(c_k)p(\mathbf{x}|c_k)}{p(\mathbf{x})}. \tag{2.3}$$

From now on, we assume that the categories c_k are mutually exclusive and also form a complete set $\{c_k\}$, i.e. $\mathbf{x}$ belongs to one but only one of the categories c_k. In this case, we have

$$\sum_{k=1}^{K} p(c_k) = 1; \tag{2.4}$$

$$\sum_{k=1}^{K} p(c_k)p(\mathbf{x}|c_k) = p(\mathbf{x}); \tag{2.5}$$

$$\sum_{k=1}^{K} p(c_k|\mathbf{x}) = 1. \tag{2.6}$$

We also have that

$$\int p(\mathbf{x})d^D x = 1; \tag{2.7}$$

$$\int p(c_k|\mathbf{x})p(\mathbf{x})d^D x = p(c_k). \tag{2.8}$$

We define the *expected conditional risk* associated with decision α_i, given the input signal $\mathbf{x}$, as

$$R(\alpha_i|\mathbf{x}) = \sum_{k=1}^{K} \lambda(\alpha_i|c_k)p(c_k|\mathbf{x}). \tag{2.9}$$

The optimal decision strategy, called the *Bayes classifier*, is the strategy that always chooses the decision that minimizes the expected conditional risk (2.9)

$$\alpha_{Bayes}(\mathbf{x}) = \mathrm{argmin}_i \left[R(\alpha_i|\mathbf{x}) \right]. \tag{2.10}$$

If we include a reject option (when in doubt) then this is modified into

$$\alpha_{Bayes}(\mathbf{x}) = \begin{cases} \mathrm{argmin}_i \left[R(\alpha_i|\mathbf{x}) \right] & \text{if } \min_i \left[R(\alpha_i|\mathbf{x}) \right] < \theta_R \\ \text{Reject} & \text{otherwise} \end{cases} \tag{2.11}$$

It is sometimes beneficial to use other decision strategies than the Bayes optimal one, e.g. when prior probabilities are uncertain. In such cases it is better to use a *Neyman-Pearson* decision rule or a *minimax* decision rule, as described by Webb (1999) or Duda, Hart & Stork (2001).

Bayes' decision rule (2.10) minimizes the expected total risk

$$R_B = \int R(\alpha_B(\mathbf{x})|\mathbf{x}) p(\mathbf{x}) d^D x. \tag{2.12}$$

Bayes' decision rule represents the best possible decision if the probabilities $p(\mathbf{x}|c_k)$ and $p(c_k)$ are known, and the costs λ_{ik} associated with each decision are known. This means that the *Bayes error* R_B is the smallest possible error. The Bayes error is unattainable except in theoretical cases, because all probabilities and costs are never known exactly in practice.

It is common to assume so-called zero-one loss, where it is assumed that to each category c_k there is a corresponding correct decision with zero loss and that all other decisions are equally wrong, incurring a unit cost:

$$\lambda(\alpha_i|c_k) = 1 - \delta_{ik} = \begin{cases} 0 & \text{if } i = k \\ 1 & \text{if } i \neq k \end{cases} \tag{2.13}$$

where δ_{ik} is the Kronecker delta function

$$\delta_{ik} = \begin{cases} 1 & \text{if } i = k \\ 0 & \text{if } i \neq k \end{cases} \tag{2.14}$$

The expected conditional risk (2.9) is then simplified to

$$R(\alpha_i|\mathbf{x}) = \sum_{k=1}^{K} (1 - \delta_{ik}) p(c_k|\mathbf{x}) = \sum_{k \neq i} p(c_k|\mathbf{x}) = 1 - p(c_i|\mathbf{x}) \tag{2.15}$$

where we have used (2.6) in the last step.

Thus, in the zero-one loss case, the Bayes optimal decision becomes to select the decision α_k associated with the category c_k that has the largest a posteriori probability $p(c_i|\mathbf{x})$.

Example: Two categories and two decisions. We assume a medical diagnosis task where we must decide whether a person is ill or healthy. We have made a set of tests (blood samples, urine samples,...) on the person and have a feature vector $\mathbf{x}$. Furthermore, we assume that the suspected disease is serious so that the cost incurred if we accidentally classify a ill person as being healthy is five times the cost incurred when we classify a healthy person as being ill (we probably have the possibilities for doing more tests if we are uncertain about the person's status). That is, $\lambda(\text{healthy}|\text{ill}) = 5\lambda(\text{ill}|\text{healthy})$. The cost for classifying an ill person as ill, or a healthy person as being healthy, is zero. The expected conditional risks for classifying the person as being ill or being healthy are then

$$\begin{aligned} R(\text{ill}|\mathbf{x}) &= \lambda(\text{ill}|\text{healthy})p(\text{healthy}|\mathbf{x}); \\ R(\text{healthy}|\mathbf{x}) &= \lambda(\text{healthy}|\text{ill})p(\text{ill}|\mathbf{x}) = 5\lambda(\text{ill}|\text{healthy})p(\text{ill}|\mathbf{x}). \end{aligned}$$

The optimal Bayes classifier then chooses

$$\begin{aligned} \text{if } p(\text{healthy}|\mathbf{x}) < 5p(\text{ill}|\mathbf{x}) &\Rightarrow \text{the person is healthy;} \\ \text{if } p(\text{healthy}|\mathbf{x}) > 5p(\text{ill}|\mathbf{x}) &\Rightarrow \text{the person is ill.} \end{aligned}$$

2.2 Discrimination functions and decision regions

We can view a classifier as a set of discrimination functions $g_i(\mathbf{x})$ such that

$$\text{if } g_i(\mathbf{x}) > g_j(\mathbf{x}) \text{ for all } j \text{ then decide } \alpha_i. \tag{2.16}$$

The discrimination functions $g_i(\mathbf{x})$ define *decision regions* Ω_i and *decision boundaries* $\partial\Omega_i$

$$\begin{aligned} \Omega_i &= \{\mathbf{x}; g_i(\mathbf{x}) > g_j(\mathbf{x}) \text{ for all } j\}; &(2.17) \\ \partial\Omega_i &= \{\mathbf{x}; g_i(\mathbf{x}) = g_j(\mathbf{x}) \text{ for any } j\}. &(2.18) \end{aligned}$$

Examples of such discrimination functions are, for the Bayes classifier, $g_i(\mathbf{x}) = R(\alpha_i|\mathbf{x})$ or, with zero-one loss, $g_i(\mathbf{x}) = -p(c_i|\mathbf{x})$.

Example: Two categories with Gaussian probability densities and zero-one loss. If we have two classes with Gaussian conditional densities

$$p(\mathbf{x}|c_k) = \frac{1}{(2\pi)^{D/2}\sqrt{|\mathbf{\Sigma}_k|}} \exp\left[\frac{-(\mathbf{x}-\mu_k)^T\mathbf{\Sigma}_k^{-1}(\mathbf{x}-\mu_k)}{2}\right] \tag{2.19}$$

where μ_k is the mean for category c_k, $\mathbf{\Sigma}_k$ is the covariance matrix for category c_k and $|\mathbf{\Sigma}_k|$ is the determinant for the covariance matrix. The Bayes optimal decisoion, with zero-one loss, is to choose the category with the maximum a posteriori probability. The decision boundary is then defined by all $\mathbf{x}$ such that the a posteriori probabilities are equal

$$\begin{aligned}
p(c_1|\mathbf{x}) &= p(c_2|\mathbf{x}) \Rightarrow \\
p(c_1)p(\mathbf{x}|c_1) &= p(c_2)p(\mathbf{x}|c_2) \Rightarrow \\
\ln\left[\frac{p(c_1)}{p(c_2)}\right] &= \ln\left[\frac{p(\mathbf{x}|c_2)}{p(\mathbf{x}|c_1)}\right] \Rightarrow \\
\ln\left[\frac{p(c_1)}{p(c_2)}\right] &= \frac{1}{2}\ln\left[\frac{|\mathbf{\Sigma}_1|}{|\mathbf{\Sigma}_2|}\right] \\
&\quad + \frac{(\mathbf{x}-\mu_1)^T\mathbf{\Sigma}_1^{-1}(\mathbf{x}-\mu_1)}{2} \\
&\quad - \frac{(\mathbf{x}-\mu_2)^T\mathbf{\Sigma}_2^{-1}(\mathbf{x}-\mu_2)}{2} \Rightarrow \\
0 &= \mathbf{x}^T\mathbf{A}\mathbf{x} + \beta^T\mathbf{x} + \gamma \qquad (2.20)
\end{aligned}$$

where

$$\begin{aligned}
\mathbf{A} &= \frac{1}{2}\left(\mathbf{\Sigma}_1^{-1} - \mathbf{\Sigma}_2^{-1}\right); && (2.21) \\
\beta &= \mathbf{\Sigma}_2^{-1}\mu_2 - \mathbf{\Sigma}_1^{-1}\mu_1; && (2.22) \\
\gamma &= \ln\left[\frac{|\mathbf{\Sigma}_1|}{|\mathbf{\Sigma}_2|}\right] - \ln\left[\frac{p(c_1)}{p(c_2)}\right] + \frac{1}{2}\left(\mu_1^T\mathbf{\Sigma}_1^{-1}\mu_1 - \mu_2^T\mathbf{\Sigma}_2^{-1}\mu_2\right). && (2.23)
\end{aligned}$$

Equation (2.20) is a quadratic form, i.e. the decision boundary between two Gaussian distributed classes is a quadratic surface (ellipsoid, hyperboloid, or paraboloid). The decision boundary will be linear (i.e. a hyperplane) in the special case of $\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2$.

2.3 Approaches to construct classifiers

It is common to group classifiers into three groups, depending on the philosphy behind their construction:

1. **A posteriori classifiers:** Model the a posteriori probabilities $p(c_k|\mathbf{x})$.
2. **Probability density classifiers:** Model the conditional probabilities $p(\mathbf{x}|c_k)$ and combine them with Bayes' rule (2.3).
3. **Decision boundary classifiers:** Construct discrimination functions g_k and thus also the decision boundaries $\partial\Omega_k$.

The first two are more powerful than the third one, in the sense that it is easy to adjust the classifier afterwards if new information is supplied (e.g. the costs λ_{ik} change, or prior probabilities change).

Which strategy to choose depends on which mathematical model ine wants to use. One model may be very suitable for modelling $p(\mathbf{x}|c_k)$, but not at all suitable for modelling $p(c_k|\mathbf{x})$, and vice versa. This is because the a prosteriori probability $p(c_k|\mathbf{x})$ often has a shape that is entirely different from the conditional probability density $p(\mathbf{x}|c_k)$. It is sometimes much easier to estimate $p(c_k|\mathbf{x})$ than to estimate $p(\mathbf{x}|c_k)$.

Example: Two Gaussian distributed classes. We have two categories c_1 and c_2 that have the same a priori probabilities $p(c_1) = p(c_2) = 0.5$. The two distributions are Gaussian distributed with the same covariance matrix $\mathbf{\Sigma}$

$$
\begin{aligned}
p(\mathbf{x}|c_1) &= \frac{1}{(2\pi)^{D/2}\sqrt{|\mathbf{\Sigma}|}} \exp\left[\frac{-(\mathbf{x}-\mu_1)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu_1)}{2}\right]; \\
p(\mathbf{x}|c_2) &= \frac{1}{(2\pi)^{D/2}\sqrt{|\mathbf{\Sigma}|}} \exp\left[\frac{-(\mathbf{x}-\mu_2)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu_2)}{2}\right].
\end{aligned}
$$

To model these probabilities it is better to use functions that are bell-shaped, like the Gaussian. The a posteriori probability, however, equals

$$
\begin{aligned}
p(c_1|\mathbf{x}) &= \frac{p(c_1)p(\mathbf{x}|c_1)}{p(c_1)p(\mathbf{x}|c_1) + p(c_2)p(\mathbf{x}|c_2)} \\
&= \frac{1}{1 + \exp\left[\beta^T\mathbf{x} + \gamma\right]}
\end{aligned}
$$

where $\boldsymbol{\beta}$ are γ are defined in (2.22) and (2.23). Expression (2.24) is a so-called logistic function, which is s-shaped (or “sigmoid”). Thus, modelling the a posteriori probability is best done with s-shaped functions.

2.4 Parametric and non-parametric models

When modeling, it is common to make a distinction between *parametric* and *non-parametric* models. Parametric models are models where one has made an assumption about the probability density (or a posteriori probability). Non-parametric models are models where no assumption is made, so-called *general approximators* are used instead.

The distinction is somewhat artificial, since all models have parameters. It is more correct to speak of models with many free parameters (non-parametric), and models with few free parameters (parametric).

The advantage with parametric models is that they are simple and quick to construct. One can afford to try many different setups. The drawback of parametric models is that one may have assumed the “wrong” parametic family, in the sense that the Bayes classifier (the optimal classifier) is not a member of the hypothesis family. This leads to a model *bias*, meaning that we will never be able to model the Bayes optimal classifier.

The benefit of non-parametric classifiers is that they are general and that we run little risk of having a model bias. The drawback, however, is that they take a lot of effort to construct, there will be little time for experimenting, and they are likely to “overtrain’ and have a large model *variance*. This means that the resulting classifier will depend very much on the set of observations used to construct it, and if we change one or a few observations then the resulting classifier will also change significantly. A classifier with a large model variance is unlikely to be able to generalize well to new observations.

It is therefore a good idea to work in the intermediate area between non-parametric (many free parameters) and parametric (few free parameters) clasifiers. In this region it is important to do a trade-off between model bias and model variance, because both contribute to the generalization error.

References

Duda, R. O., Hart, P. E. & Stork, D. G. (2001), *Pattern Classification (*2^{nd} *edition)*, John Wiley & Sons, New York.

Pomerleau, D. A. (1993), *Neural Network Perception for Mobile Robot Guidance*, Kluwer Academic Publishers, Boston, MA.

Ripley, B. D. (1996), *Pattern Recognition and Neural Networks*, Cambridge University Press, Cambridge.

Webb, A. (1999), *Statistical Pattern Recognition*, Arnold (a Member of the Hodder Headline Group), London.