2 Bayesian classification

As seen before, Bayesian classification methods explicitly use conditional probabilities. Many other methods such as neural networks do so implicitly. Approaches such as minimization of the sum of squared errors have Bayesian origins. Concepts such as Occam's razor and minimum description length also have Bayesian origins. Bayesian methods however require knowledge of many probabilities and hence require lots of data. Further, they may be computationally intensive, if the optimal answer is sought.

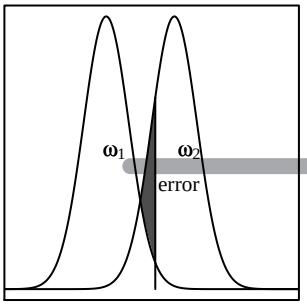

Fig. 2.1: The shaded area ought to be reduced for optimal classification.

2.1 Bayesian classification with known prior distributions

Bayesian Classification: 2 class case

Let ω_1, ω_2 be two classes (gene/non-gene, male/female). Assume that $P(\omega_1)$ and $P(\omega_2)$ are known (from previous data). Out of N previous data points, if $N_1 \in \omega_1$ and $N_2 \in \omega_2$ (where $N_1 + N_2 = N$), then

$$P(\omega_1) \approx \frac{N_1}{N} \qquad P(\omega_2) \approx \frac{N_2}{N} \qquad N = N_1 + N_2$$

Let $P(x|\omega_1)$ and $P(x|\omega_2)$ be class-conditional pdfs. If unknown, they can be estimated. $P(x|\omega_i)$ is the likelihood function of ω_i w.r.t. x.

$$P(\omega_i|x) = \frac{P(x|\omega_i)P(\omega_i)}{P(x)} \tag{2.1}$$

$$P(x) = \sum_i P(x|\omega_i)P(\omega_i) = P(x|\omega_1)P(\omega_1) + P(x|\omega_2)P(\omega_2) \tag{2.2}$$

The Bayesian classification rule for 2 classes may be applied as follows:
Assign x to the class which is most probable

We are performing a maximum *a posteriori* or MAP classification.

- If $P(\omega_1|x) > P(\omega_2|x)$, then x belongs to ω_1
- If $P(\omega_2|x) > P(\omega_1|x)$, then x belongs to ω_2
- If $P(\omega_1|x) = P(\omega_2|x)$, then x is randomly assigned to either class.

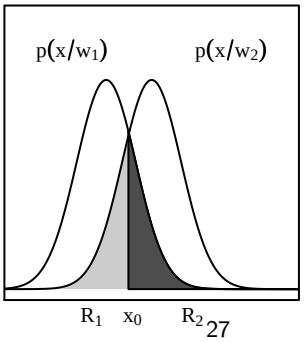

Fig. 2.2: Optimal classification.

Using Bayes' rule, $P(\omega_1|x) \gtrless P(\omega_2|x)$ implies

$$\frac{P(x|\omega_1)P(\omega_1)}{P(x)} \gtrless \frac{P(x|\omega_2)P(\omega_2)}{P(x)} \quad \Rightarrow \quad P(x|\omega_1)P(\omega_1) \gtrless P(x|\omega_2)P(\omega_2) \tag{2.3}$$

If $P(\omega_1) = P(\omega_2) = 1/2$ (both classes are equally likely), then $P(\omega_1|x) \gtrless P(\omega_2|x)$ implies

$$P(x|\omega_1) \gtrless P(x|\omega_2) \tag{2.4}$$

Decision errors

The total probability P_e of committing a decision error is

$$\begin{aligned} P_e &= P(x \in R_2,\ \text{we choose } \omega_1) + P(x \in R_1,\ \text{we choose } \omega_2) \\ &= \int_{-\infty}^{x_0} P(x|\omega_2)P(\omega_2)dx + \int_{x_0}^{\infty} P(x|\omega_1)P(\omega_1)dx \end{aligned} \tag{2.5}$$

For equiprobable classes,

$$P_e = \frac{1}{2}\left[\int_{-\infty}^{x_0} P(x|\omega_2)dx + \int_{x_0}^{\infty} P(x|\omega_1)dx\right]$$

The Bayesian classification rule defined above minimizes P_e!
Proof:

$$\begin{aligned} P_e &= P(x \in R_2|\omega_1)P(\omega_1) + P(x \in R_1|\omega_2)P(\omega_2) \\ &= P(\omega_1)\int_{R_2} P(x|\omega_1)dx + P(\omega_2)\int_{R_1} P(x|\omega_2)dx \\ &= \int_{R_2} P(\omega_1|x)P(x)dx + \int_{R_1} P(\omega_2|x)P(x)dx \end{aligned}$$

But

$$\int_{R_1} P(\omega_1|x)P(x)dx + \int_{R_2} P(\omega_1|x)P(x)dx = P(\omega_1)$$

Therefore

$$P_e = P(\omega_1) - \int_{R_1} \left[P(\omega_1|x) - P(\omega_2|x)\right] p(x)dx$$

Minimizing P_e = maximizing the integral = choosing R_1 (i.e. x_0) such that $P(\omega_1|x) > P(\omega_2|x)$ in R_1. Therefore $P(\omega_2|x) > P(\omega_1|x)$ in R_2.

Generalizing the Bayesian Classification Rule:

Given m equiprobable classes ω_1, ω_2,..., ω_m, x is assigned to class ω_i if

$$P(\omega_i|x) > P(\omega_j|x) \quad \forall\, j \neq i \tag{2.6}$$

Risk minimization

In minimizing P_e, we summed two error terms (R_1 and R_2 areas) which implied giving each error *equal weightage*. If some errors are more serious than others, then some weights (penalty terms) are needed. Given m classes (ω_i, $i = 1, 2, ..., m$), when a point is misclassified, (e.g. $x \in \omega_k$, but is assigned to ω_i, $k \neq i$), let a penalty term λ_{ki} (= loss) be associated with this misclassification. $\mathbf{L} = [\lambda_{ki}]$ is the loss matrix. The risk (or loss) associated with the class ω_k is

λ_{ki} is the penalty applied when an object from class k is assigned to class i.

$$r_k = \sum_{i=1}^{m} \lambda_{ki} \int_{R_i} P(x|\omega_k)dx \tag{2.7}$$

where $\int_{R_i} P(x|\omega_k)dx$ is the overall probability of a feature (point) from class k being wrongly classified in ω_i. The average risk therefore is

r vs. P_e :
Consider the case when $\lambda_{ki} = 1$ when $k \neq i$ and 0 when $k = i$. Then $\lambda_{ki} = 1 - \delta_{ki}$ where δ_{ki} = Kronecker delta = 0 ($k \neq i$) and 1 ($k = i$). Then minimizing r is the same as minimizing P_e.

$$\mathcal{E}[r] = \sum_{k=1}^{m} r_k P(\omega_k) = \sum_{i=1}^{m} \int_{R_i} \left(\sum_{k=1}^{m} \lambda_{ki} P(x|\omega_k)P(\omega_k)\right) dx \tag{2.8}$$

This average risk must be minimized. This amounts to minimizing each integral. Using the notation

$$l_i = \sum_{k=1}^{m} \lambda_{ki} P(x|\omega_k)P(\omega_k) \tag{2.9}$$

$$x \in R_i \text{ if } \quad l_i < l_j \quad \forall\, j \neq i$$

When the classes are equiprobable, this simplifies to

$$\sum_{k=1}^{m} \lambda_{ki} P(x|\omega_k) < \sum_{k=1}^{m} \lambda_{kj} P(x|\omega_k) \quad \forall\, j \neq i \tag{2.10}$$

Risk minimization: Two-class case.

$$\begin{aligned} l_1 &= \lambda_{11} P(x|\omega_1) P(\omega_1) + \lambda_{21} P(x|\omega_2) P(\omega_2) \\ l_2 &= \lambda_{12} P(x|\omega_1) P(\omega_1) + \lambda_{22} P(x|\omega_2) P(\omega_2) \end{aligned} \tag{2.11}$$

$x \in \omega_1$ if $l_1 < l_2$, i.e.

$$(\lambda_{21} - \lambda_{22}) P(x|\omega_2) P(\omega_2) < (\lambda_{12} - \lambda_{11}) P(x|\omega_1) P(\omega_1) \tag{2.12}$$

Usually λ_{ij} is assumed to be greater than λ_{ii}: wrong decisions must be penalized more than correct ones!

Then

$$x \in \omega_1 \text{ if } \quad l_{12} = \frac{P(x|\omega_1)}{P(x|\omega_2)} > \frac{P(\omega_2)}{P(\omega_1)} \frac{\lambda_{21} - \lambda_{22}}{\lambda_{12} - \lambda_{11}} \tag{2.13}$$

where l_{12} is called the likelihood ratio.

Let $P(\omega_1) = P(\omega_2) = 1/2$. Let the loss matrix be

$$\mathbf{L} = \begin{bmatrix} 0 & \lambda_{12} \\ \lambda_{21} & 0 \end{bmatrix} \tag{2.14}$$

Assume that $\lambda_{21} > \lambda_{12}$ (i.e. misclassifying a pattern from 2 is more serious).

$$\begin{aligned} &x \in \omega_1 \text{ if } \frac{P(x|\omega_1)}{P(x|\omega_2)} > \frac{\lambda_{21}}{\lambda_{12}} \quad \text{or} \quad P(x|\omega_1) \frac{\lambda_{12}}{\lambda_{21}} > P(x|\omega_2) \\ &x \in \omega_2 \text{ if } P(x|\omega_2) > P(x|\omega_1) \frac{\lambda_{12}}{\lambda_{21}} \end{aligned} \tag{2.15}$$

$\lambda_{12}/\lambda_{21} < 1$ and therefore the threshold is moved to the left of x_0. The size of R_2 is increased.

An extension to this risk minimization approach involves using a reject option: when the largest $p(\omega_k|x)$ is far < 1, a decision would be hard to make. We reject this point from our analysis therefore, in an attempt to improve our error rate.

Example: P_e vs r minimization:

Let $P(x|\omega_1)$ and $P(x|\omega_2)$ be $\mathcal{N}(0, 1/2)$ and $\mathcal{N}(1, 1/2)$ gaussians respectively

$$\begin{aligned} P(x|\omega_1) &= \frac{1}{\sqrt{\pi}} \exp\left(-x^2\right) \\ P(x|\omega_2) &= \frac{1}{\sqrt{\pi}} \exp\left(-(x-1)^2\right) \end{aligned}$$

x_0 for minimum error probability is when

$$\exp\left(-x_0^2\right) = \exp\left(-(x_0 - 1)^2\right) \quad \Rightarrow \quad x_0 = 1/2$$

Minimum risk case assuming $\mathbf{L} = \begin{bmatrix} 0 & 0.5 \\ 1.0 & 0 \end{bmatrix}$

$$\exp\left(-x_0^2\right) = \frac{1.0}{0.5} \exp\left(-(x_0 - 1)^2\right) \quad \Rightarrow \quad x_0 = \frac{1 - \ln 2}{2}$$

which is $< 1/2$ and hence the threshold has moved left.

Also, if $P(\omega_1) > P(\omega_2)$, the threshold moves right: it is better to make fewer errors with the more probable class.

Discriminant and decision surfaces

$P(\omega_i|x) - P(\omega_j|x) = 0$ defines a decision surface between ω_i and ω_j, assuming minimum error probablity. Instead of directly using $P(\omega_i|x)$, we could use

$$g(x) \equiv f\left(P(\omega_i|x)\right) \tag{2.16}$$

where f is a monotonically increasing function. $g_i(x)$ is a discriminant function.

$$x \in \omega_i \text{ if } g_i(x) > g_j(x) \quad \forall\, j \neq i \tag{2.17}$$

The decision surface is now

$$g_{ij} \equiv g_i(x) - g_j(x) = 0, \quad i,j = 1,...,m, \quad i \neq j \tag{2.18}$$

Forms of discriminants include

$$g(\mathbf{x}) = P(\omega_1|\mathbf{x}) - P(\omega_2|\mathbf{x}) \tag{2.19}$$

$$g(\mathbf{x}) = \ln \frac{P(\mathbf{x}|\omega_1)}{P(\mathbf{x}|\omega_2)} + \ln \frac{P(\omega_1)}{P(\omega_2)} \tag{2.20}$$

Bayesian Classification for Normal Distributions

The multivariate normal density function in d dimensions, for the ω_i class (out of m classes) is

$$P(\mathbf{x}|\omega_i) = \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}_i|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^\mathsf{T}\mathbf{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)\right) \qquad i = 1,...,m \tag{2.21}$$

where $\boldsymbol{\mu}_i = \mathcal{E}[\mathbf{x}]$ = mean value of $\mathbf{x}$ in the ω_i class, $\mathbf{\Sigma}_i$ = covariance matrix = $\mathcal{E}\left[(\mathbf{x}-\boldsymbol{\mu}_i)(\mathbf{x}-\boldsymbol{\mu}_i)^\mathsf{T}\right]$, and $|\mathbf{\Sigma}_i|$ = determinant of $\mathbf{\Sigma}_i$.

The general notation for a multivariate normal is $\mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$.

If $d = 1$,

$$P(\mathbf{x}|\omega_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{1}{2}\frac{(x-\mu_i)^2}{\sigma_i^2}\right) \qquad i = 1,...,m \tag{2.22}$$

The discriminant function is

$$\begin{aligned} g_i(\mathbf{x}) &= \ln[P(\mathbf{x}|\omega_i)P(\omega_i)] = \ln P(\mathbf{x}|\omega_i) + \ln P(\omega_i) \\ &= -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^\mathsf{T}\mathbf{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i) + \ln P(\omega_i) + C \end{aligned} \tag{2.23}$$

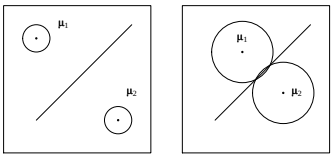

Fig. 2.3: Classification is good when clusters are small and compact (left) and poor when clusters are large (right).

where $C = -(d/2)\ln 2\pi - (1/2)\ln|\mathbf{\Sigma}_i|$. Expanding,

$$g_i(\mathbf{x}) = -\frac{1}{2}\mathbf{x}^\mathsf{T}\mathbf{\Sigma}_i^{-1}\mathbf{x} + \frac{1}{2}\mathbf{x}^\mathsf{T}\mathbf{\Sigma}_i^{-1}\boldsymbol{\mu}_i - \frac{1}{2}\boldsymbol{\mu}_i\mathbf{\Sigma}_i^{-1}\boldsymbol{\mu}_i + \frac{1}{2}\boldsymbol{\mu}_i^\mathsf{T}\mathbf{\Sigma}_i^{-1}\mathbf{x} + \ln P(\omega_i) + C_i \tag{2.24}$$

This is in general of a nonlinear, quadratic form.

- If $d = 2$ and $\mathbf{\Sigma}_i = \begin{bmatrix} \sigma_i^2 & 0 \\ 0 & \sigma_i^2 \end{bmatrix}$

$$g_i(\mathbf{x}) = -\frac{1}{2\sigma_i^2}(x_1^2 + x_2^2) + \frac{1}{\sigma_i^2}(\mu_{i1}x_1 + \mu_{i2}x_2) - \frac{1}{2\sigma_i^2}(\mu_{i1}^2 + \mu_{i2}^2) + \ln P(\omega_i) + C_i \tag{2.25}$$

  The associated decision curve, $g_i(\mathbf{x}) - g_j(\mathbf{x})$, is a quadric: ellipsoid, hyperbolic, parabolic, or line pairs. For $d > 2$, decision surfaces are hyperquadric. The quadratic nature is due to the $\mathbf{x}^\mathsf{T}\mathbf{\Sigma}_i^{-1}\mathbf{x}$ term.

- If $\mathbf{\Sigma}_i$ is the same in all classes $= \mathbf{\Sigma}$, then $\mathbf{x}^\mathsf{T}\mathbf{\Sigma}_i^{-1}\mathbf{x}$ is the same in all g_{ij} terms and does not matter as it gets subtracted out in $g_{ij} = g_i - g_j$. Hence we may redefine $g_i(\mathbf{x})$ as

$$g_i(\mathbf{x}) = \mathbf{w}_i^\mathsf{T}\mathbf{x} + w_{i0} \tag{2.26}$$

  where $\mathbf{w}_i = \mathbf{\Sigma}^{-1}\boldsymbol{\mu}_i$ and $w_{i0} = \ln P(\omega_i) - (1/2)\boldsymbol{\mu}_i^\mathsf{T}\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_i$ = scalar. Therefore $g_i(\mathbf{x})$ = linear function of $\mathbf{x}$ and the decision surfaces are hyperplanes.

- If $\mathbf{\Sigma}$ = diagonal matrix with equal elements

$$\mathcal{E}[(x_i - \mu_i)(x_j - \mu_j)] = \sigma^2\delta_{ij} \qquad \Rightarrow \qquad \mathbf{\Sigma} = \sigma^2\mathbf{I} \tag{2.27}$$

where $\mathbf{I}$ is the d-dimensional identity matrix, and therefore

$$g_i(\mathbf{x}) = \frac{1}{\sigma^2}\boldsymbol{\mu}_i^\mathsf{T}\mathbf{x} + w_{i0} \tag{2.28}$$

$$g_{ij}(\mathbf{x}) \equiv g_i(\mathbf{x}) - g_j(\mathbf{x}) = \mathbf{w}^\mathsf{T}(\mathbf{x} - \mathbf{x}_0) = 0 \tag{2.29}$$

where $\mathbf{w} = \boldsymbol{\mu}_i - \boldsymbol{\mu}_j$

$$\mathbf{x}_0 = \frac{1}{2}(\boldsymbol{\mu}_i + \boldsymbol{\mu}_j) - \sigma^2 \ln\left[\frac{P(\omega_i)}{P(\omega_j)}\right] \frac{\boldsymbol{\mu}_i - \boldsymbol{\mu}_j}{\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|^2} \tag{2.30}$$

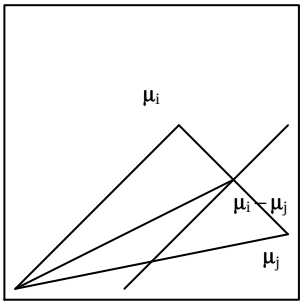

Fig. 2.4: The decision hyperplane.

The definition of g_{ij} implies that $\mathbf{x}_0$ is a point on the hyperplane. Then if any other point $\mathbf{x}$ lies on the decision hyperplane, the vector $\mathbf{x} - \mathbf{x}_0$ also lies on the hyperplane. Then $g_{ij} = 0$ implies that $\mathbf{w}^\mathsf{T}(\mathbf{x}-\mathbf{x}_0) = (\boldsymbol{\mu}_i-\boldsymbol{\mu}_j)^\mathsf{T}(\mathbf{x}-\mathbf{x}_0) = 0$ on the hyperplane. Therefore $\boldsymbol{\mu}_i - \boldsymbol{\mu}_j$ is orthogonal to the decision hyperplane.

The decision surface is a hyperplane passing through $\mathbf{x}_0$ (Fig. 2.4). $\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|^2$ is the Euclidean distance. If $P(\omega_i) = P(\omega_j)$, then

$$\mathbf{x}_0 = \frac{1}{2}(\boldsymbol{\mu}_i + \boldsymbol{\mu}_j) \tag{2.31}$$

(We saw proof of this in the P_e vs. r minimization example above with $X_0 = 0.5$). The decision hyperplane is a straight line orthogonal to $\boldsymbol{\mu}_i - \boldsymbol{\mu}_j$. Also, if $P(\omega_i) < P(\omega_j)$, then the hyperplane is located closer to $\boldsymbol{\mu}_i$.

- If $\boldsymbol{\Sigma}$ is nondiagonal,

$$g_{ij}(\mathbf{x}) = \mathbf{w}^\mathsf{T}(\mathbf{x} - \mathbf{x}_0) = 0 \tag{2.32}$$

where $\mathbf{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)$ and

$$\mathbf{x}_0 = \frac{1}{2}(\boldsymbol{\mu}_i + \boldsymbol{\mu}_j) - \ln\left[\frac{P(\omega_i)}{P(\omega_j)}\right] \frac{\boldsymbol{\mu}_i - \boldsymbol{\mu}_j}{\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|^2_{\boldsymbol{\Sigma}^{-1}}} \tag{2.33}$$

where $\|x\|_{\boldsymbol{\Sigma}^{-1}} \equiv (\mathbf{x}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\mathbf{x})^{1/2}$ is the $\boldsymbol{\Sigma}^{-1}$ norm of $\mathbf{x}$. The decision hyperplane is *not* orthogonal to $\boldsymbol{\mu}_i - \boldsymbol{\mu}_j$ but to $\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)$.

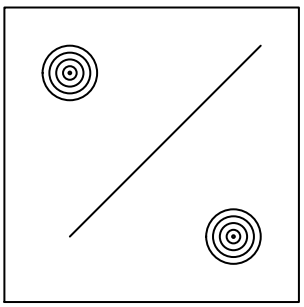

Fig. 2.5: Contours corresponding to $d_e = c$ (constant). $d_m = c$ would give elliptical contours.

Minimum distance classifiers

Assuming equiprobable classes with the same $\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}$, and neglecting constants

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^\mathsf{T}\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_i) \tag{2.34}$$

- When $\boldsymbol{\Sigma} = \sigma^2 I$, maximum $g_i(\mathbf{x})$ implies minimum Euclidean distance d_e (2.5)

$$d_e = \|\mathbf{x} - \boldsymbol{\mu}_i\| \tag{2.35}$$

- When $\boldsymbol{\Sigma} \neq \sigma^2 I$ (nondiagonal $\boldsymbol{\Sigma}$), maximum $g_i(\mathbf{x})$ implies minimum $\boldsymbol{\Sigma}^{-1}$ norm

$$d_m = \left((\mathbf{x} - \boldsymbol{\mu}_i)^\mathsf{T}\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_i)\right)^{1/2} \tag{2.36}$$

This distance is known as the Mahalanobis distance.

For a distance d_{ij} between i and j to be considered a metric, the following must hold:

1. $d_{ii} = 0$.
2. $d_{ij} = d_{ji}$.
3. $d_{ij} \leq d_{ik} + d_{kj}$.

Distances do **not** have to be metrics always.

The covariance matrix is symmetric and can be diagonalized by a unitary transform

$$\boldsymbol{\Sigma} = \Phi\Lambda\Phi^\mathsf{T} \tag{2.37}$$

where $\Phi^\mathsf{T} = \Phi^{-1}$, Λ is a diagonal matrix of eigenvalues of $\boldsymbol{\Sigma}$, and Φ has the unit orthonormal vectors (eigenvectors corresponding to the eigenvalues) as its columns.

Naive Bayes Classifier

The naive Bayes classifier assumes conditional independence between attributes. Thus

$$P(x \in \omega_j | a, b) = P(\omega_j | a) P(\omega_j | b) \tag{2.38}$$

$$\begin{aligned} p(\omega_j|a,b) &\propto p(a,b|\omega_j)p(\omega_j) \\ &\propto p(a|\omega_j)p(b|\omega_j)p(\omega_j) \\ &\propto p(\omega_j|a)p(\omega_j|b) \end{aligned}$$

Given the attribute values a_i, the naive Bayes classifier is

$$\omega_{NB} = \underset{\omega_j}{\text{argmax}} \prod_i P(\omega_j|a_i) = \underset{\omega_j}{\text{argmax}}\ P(\omega_j) \prod_i P(a_i|\omega_j) \tag{2.39}$$

This is a simple classifier to implement; yet in certain situations, this classifier does as well as neural networks and decision trees.

The play tennis example: On 14 days, the decision to play tennis depends on 4 attributes: outlook, temperature, humidity and wind. Today's attributes are {sunny, cool, high, strong}. Do we play?

Day	Outlook	Temp.	Humidity	Wind	Play?
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no

From the given data, $P(\text{yes}) = 9/14 = 0.64$, $P(\text{No}) = 5/14 = 0.36$. $P(\text{strong wind} \mid \text{yes}) = 3/9 = 0.33$ and $P(\text{strong wind} \mid \text{no}) = 3/5 = 0.60$ etc.

$$P(\text{yes})P(\text{sunny}|\text{yes})P(\text{cool}|\text{yes})P(\text{high}|\text{yes})P(\text{strong}|\text{yes}) = 0.0053$$

$$P(\text{no})P(\text{sunny}|\text{no})P(\text{cool}|\text{no})P(\text{high}|\text{no})P(\text{strong}|\text{no}) = 0.0206$$

$\omega_{NB} = \text{No}!$

$$P(\text{No}|\text{sunny, cool, high, strong}) = \frac{0.0206}{0.0206 + 0.0051} = 0.795$$

2.2 Bayesian classification when training data is available

For example, the Binomial or Normal distribution.

As before, we wish to identify a class ω_k which maximizes $P(\mathbf{x}|\omega_i)$. Let $P(\mathbf{x}|\omega_i)$ be defined by a distribution with parameters $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$. Then estimating the value of $P(\mathbf{x}|\omega_i)$ is converted to a problem of estimating the parameters $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$. Maximum likelihood estimation (MLE) and maximum a posteriori estimation (MAP) attempt to identify these model parameters using a Frequentist approach: i.e. these parameters have unique unknown values, and their best estimates must be those which maximize the probability of observing the data the data which is now available. The Bayesian classification approach instead considers the parameters as random variables themselves and looks to update prior estimates of these parameters given the new data.

Let $\mathbf{D}$ represent new data available to us (observations from experiments). Let $\mathbf{M}$ represent a set of models (hypotheses) such as M_1, M_2, ... Prior probabilities of the various models (i.e. $p(M_1)$, $p(M_2)$ etc. are known. $\mathbf{M}$ must be a set of mutually exclusive models and $\sum_i p(M_i) = 1$. $p(\mathbf{D})$ represents the probability that data set $\mathbf{D}$ is observed, regardless of which model is correct. $p(\mathbf{D}|M_1)$ is the probability of observing $\mathbf{D}$ according to model M_1. The prior probability of that model $= p(M_1)$ (*independent of* $\mathbf{D}$!). According to Bayes rule

We want $p(M_1|\mathbf{D})$, the posterior probability.

$$p(M_1|\mathbf{D}) = \frac{p(\mathbf{D}|M_1)p(M_1)}{p(\mathbf{D})} \tag{2.40}$$

- $p(M_1|\mathbf{D})$ is proportional to $p(M_1)$
- $p(M_1|\mathbf{D})$ is proportional to $p(\mathbf{D}|M_1)$
- $p(M_1|\mathbf{D})$ is *inversely* proportional to $p(\mathbf{D})$: the more probable it is that we see the data $\mathbf{D}$ (i.e. $p(\mathbf{D})$ increases), then less likely it is that $\mathbf{D}$ supports M_1 specifically.

There are three common approaches to handling classification when the prior distribution is not known, but when training data is available.

1. Maximum likelihood estimation
2. Maximum a posteriori estimation
3. Bayesian estimation

We have already learnt how to estimate model parameters using the first two approaches. In particular, we looked at binomial proportions and Gaussian parameters. The third approach is well described in Duda, Hart and Stork and has been briefly discussed in the previous chapter.

2.3 Nonparametric approaches to density estimation

MLE, MAP and bayesian estimation, from the previous chapter, were *parametric* approaches to modelling densities, given specific functional forms.

In the previous chapter, we have focussed on the estimation of parameters in density functions which have been explicitly defined (for example as Gaussians or binomials). Available training data was used to determine the values of these parameters. In several situations, the appropriate form of the density is unknown, with the usual candidates turning out to be poor fits. In such situations *nonparametric* forms are preferred, where no explicit definition of the functional form of the density function is made.

Histogram density models

Histogram density models.

Histograms end up partitioning data x into distinct bins of width Δ_i, with n_i observations falling in bin i. Normalized probability densities may be derived from these, by dividing by the total number of observations N and by the bin width Δ_i (bin widths are usually chosen to be constant: $\Delta_i = \Delta$):

$$p_i = \frac{n_i}{N\Delta_i} \tag{2.41}$$

The goodness of fit strongly depends on the number of points, and the bin width. As with parametric density functions, once estimation is done, the raw data is not used further. Also, histograms are easily updated if/when data comes in sequentially. A significant disadvantage of histograms is that they do not cope well with high dimensionality; if each variable in d dimensions were to have m bins, we would have m^d bins. The binning procedure employed in generating histograms involves looking at local neighbourhoods of points. Thw bin width also worked as a 'smoothing' parameter. These two observations are used in the next two nonparametric methods.

Curse of dimensionality.

Kernel density estimators

Kernel density estimation.

Let $p(\mathbf{x})$ be some unknown pdf in d-dimensional Euclidean space. We wish to estimate $p(\mathbf{x})$ and therefore we focus on a small region $\mathcal{R}$ containing $\mathbf{x}$. The probability mass in this region is

$$P = \int_{\mathcal{R}} p(\mathbf{x}) d\mathbf{x} \tag{2.42}$$

If there are N observations, each point has probability P of falling within $\mathcal{R}$. Then the total number of points k that lie inside $\mathcal{R}$ follow a binomial:

$$\text{Bin}(k|N,P) = \frac{N!}{k!(N-k)!} P^k (1-P)^{1-k} \tag{2.43}$$

The mean number of points inside the region would be $\mathcal{E}[k/N] = P$ and $\text{Var}[k/N] = P(1-P)/N$. For large N, there would be sharp peak around the mean, and so

$$k \simeq NP \tag{2.44}$$

We assume $\mathcal{R}$ to be very small, and with volume V: $p(\mathbf{x})$ is constant throughout; then

$$P \simeq p(\mathbf{x})V \tag{2.45}$$

Our density estimate then is

$$p(\mathbf{x}) = \frac{k}{NV} \tag{2.46}$$

There is a contradiction in what has been done: we have assumed the region $\mathcal{R}$ to be small enough for the density to be constant inside it, but we have also assumed that the number of points falling inside it is large enough for the binomial to be sharply peaked. At this point, we either fix k and determine V from the data (k-nearest-neighbour density estimation) or else we fix V and compute k (kernel density estimation). As $N \to \infty$, both estimates converge to the true probability density.

The Parzen window.

In the kernel method, $\mathcal{R}$ is a small hypercube centered on $\mathbf{x}$. To count k in this region, we define a unit cube centered on the origin:

$$K(\mathbf{u}) = \begin{cases} 1, & |u_i| \leq 1/2, \quad i = 1, \ldots, d \\ 0, & \text{otherwise} \end{cases} \tag{2.47}$$

The function $k(\mathbf{u})$ is a *kernel function*, and in this specific case is also known as a *Parzen window*. Then $K((\mathbf{x} - \mathbf{x}_n)/h)$ will be one if $\mathbf{x}_n$ lies inside a cube of side h centered on $\mathbf{x}$, and zero otherwise. The total number of points lying inside this cube would be

$$k = \sum_{n=1}^{N} K\left(\frac{\mathbf{x} - \mathbf{x}_n}{h}\right) \tag{2.48}$$

Using Eq. 2.46 gives

$$p(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{n} \frac{1}{h^d} K\left(\frac{\mathbf{x} - \mathbf{x}_n}{h}\right) \tag{2.49}$$

where $V = h^d$. Since $K(\mathbf{u})$ is symmetric, the equation above may be interpreted, instead of a single cube centered on $\mathbf{x}$, as the sum over N cubes, centred on the N data points $\mathbf{x}_n$. The kernel function above suffers from discontinuities at the boundaries of the cubes. We could use a smoother kernel function, for example a Gaussian, as

Radial Basis Function. We will see more of this when we describe multiperceptron methods.

$$p(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{(2\pi h^2)^{1/2}} \exp\left\{-\frac{||\mathbf{x} - \mathbf{x}_n||^2}{2h^2}\right\} \tag{2.50}$$

where h represents the 'standard deviation'. Effectively we have placed a Gaussian over each

data point, and have then added up the whole data set and divided by N. h works as a smoothing parameter here. In general, we can choose any kernel function $k(\mathbf{u})$ subject to

$$K(\mathbf{u}) \geq 0 \tag{2.51}$$

$$\int K(\mathbf{u}) d\mathbf{u} = 1 \tag{2.52}$$

The Parzen window estimator in Eq. 2.49 is simple to implement. It's weakness is that density evaluation linearly increases in cost with the size of the data set.

Nearest-neighbour method based classification

If instead, we fix k and try to determine V, we can visualize a small sphere centred on $\mathbf{x}$ where we wish to estimate $p(\mathbf{x})$. We then grow the radius till we have k points in the sphere. The choice of k is now critical. If we have N_i points in class ω_i, with $\sum_i N_i = N$, we draw a sphere centred on $\mathbf{x}$ containing exactly k points, irrespective of their class. Let this sphere have volume V and contain points k_i points from class ω_i. Then an estimate of the density associated with each class is (using Eq. 2.46)

$$p(\mathbf{x}|\omega_i) = \frac{k_i}{N_i V} \tag{2.53}$$

The unconditional density is

$$p(\mathbf{x}) = \frac{k}{NV} \tag{2.54}$$

and the priors are given by

$$p(\omega_i) = \frac{N_i}{N} \tag{2.55}$$

Using Bayes' theorem, we have

$$p(\omega_i|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_i)p(\omega_i)}{p(\mathbf{x})} = \frac{k_i}{k} \tag{2.56}$$

To minimize the probability of misclassification, assign $\mathbf{x}$ to the class having the largest posterior probability, i.e. the largest value of k_i/k. The extent of smoothing seen here is a function of the choice of k. For even $k = 1$, it has been shown that as $N \to \infty$, the error rate is less than twice that of the minimum achievable error rate of an optimal classifier.

Assign a point to that class which has most representatives in the neighbourhood. In the case of a tie, allocate randomly to one of the tied groups.

We will see more of these concepts later when we discuss data clustering approaches.