第二章 基礎理論介紹

本章將探討本論文的研究背景與動機目的，並大略的說明各個章節的主要內容及重要特性。

2.1 *k*NN 基本理論

在圖形辨識的領域中，最基本的一環便是希望能判斷出某個不知道類別的資料點究竟歸屬於哪一類。這時候我們便需要某種分類的法則來加以區別。

最近鄰居決定法 (nearest neighbor decision rule) [6]是目前已被廣泛使用在統計圖樣識別的一種非參數分類法則，本身並沒有包含太多的理論基礎，靠的是最直覺的方式來決定如何分類，亦即以最接近自己的鄰居來決定自己的類別。假如現在要對一維度為r的未知類別的輸入圖樣x做分類，則在所有的碼字$\{y^1, y^2, \cdots, y^t\}$中找出與$x$最接近的一個碼字$y^j$，並以$y^j$的類別來決定$x$該歸類在哪個類別，距離的測量方法可以是任何的方法，例如歐幾里得平方距離：

$$D(x,y) = \|x - y\|^2 = \sum_{i=1}^{r} (x_i - y_i)^2 \tag{2.1}$$

當滿足下面數學式時，將 x 歸入 y^j 這一類：

$$D\left(x, y^j\right) < D\left(x, y^i\right) \text{，} i = 1,2,\ldots,t \quad i \neq j \tag{2.2}$$

圖 2-1 是最接近鄰居決定法的示意圖，此例中會將輸入向量 x 歸類為 Class1。

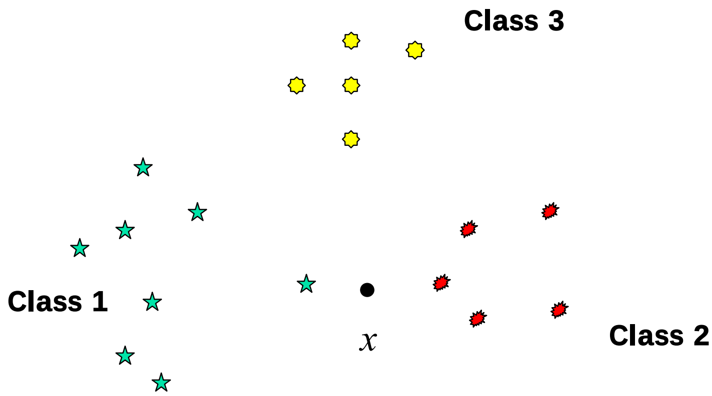

圖 2-1 最近鄰居決定法示意圖

可是，單以一個鄰居來決定類別有欠公平，將最近鄰居決定法延伸演成 k 最近鄰居決定法 (k nearest neighbor decision rule)，這樣的決定法簡稱 *k*NN，顧名思義是由最接近的 k 個鄰居投票決定自己的類別。圖 2-2 是 *k*NN 的示意圖，假設 k 的值設定是 5，因為最接近 x 的 5 個碼字中，Class2 的碼字佔有最多數共 3 個，所以輸入向量 x 將被歸類為 Class2。

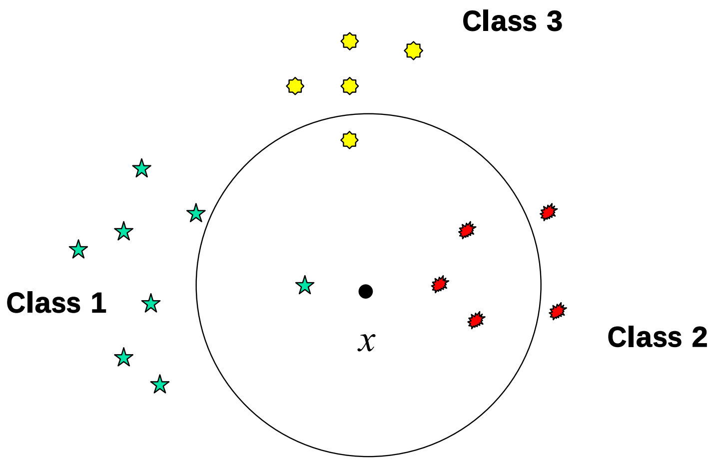

圖 2-2 *k*NN 示意圖

由上面例子可以很直接看出分類時所做的碼字比對是全搜尋的，碼字數目的增加或碼字維度的增加將會大幅提升計算複雜度及設計集的儲存複雜度，因此如何有效降低計算複雜度是一個重要的課題。

2.2 離散小波轉換

在這一小節我們將簡單的介紹小波轉換。在訊號轉換處理的過程中，訊號被投影到一組基底函數上，相對應於基底函數的係數正是我們要處理的部份。有效的處理方式能將訊號藉由轉換過程使轉換後的訊號能量集中在少數的係數上，而小波轉換正符合此特性，使得我們可以捨棄不重要的高頻係數資訊而不會嚴重影響分類的正確性。

令$\mathbf{X}$是維度$2^n \times 2^n$的向量x做n階離散小波轉換 (n-stage DWT)之後的結果，如圖 2-3 所示，$\mathbf{X}$亦為$2^n \times 2^n$的區塊且由$\mathbf{x}_{L0}$和$\mathbf{x}_{Vi}$、$\mathbf{x}_{Hi}$、$\mathbf{x}_{Di}$所組成，其中$i = 0,1,\cdots,n-1$。在離散小波轉換的過程當中，區塊$\mathbf{x}_{L(m\text{-}1)}$(低頻區塊，$m = 1,2,\cdots,n$) 以及$\mathbf{x}_{V(m\text{-}1)}$、$\mathbf{x}_{H(m\text{-}1)}$、$\mathbf{x}_{D(m\text{-}1)}$(高頻區塊，$m = 1,2,\cdots,n$)是由$\mathbf{x}_{Lm}$遞迴獲得，而$\mathbf{x}_{Ln}$等於輸入向量x，$\mathbf{x}_{Lm}$、$\mathbf{x}_{Vm}$、$\mathbf{x}_{Hm}$和$\mathbf{x}_{Dm}$這些區塊的維度是$2^m \times 2^m$。也就是說，這些區塊$\mathbf{x}_{L0}$和$\mathbf{x}_{Vi}$、$\mathbf{x}_{Hi}$、$\mathbf{x}_{Di}$，其中$i = 0,1,\cdots,m-1$，是由$\mathbf{x}_{Lm}$經過$m$階的離散小波轉換後得來。而$\mathbf{x}_{Lm}$可經由一個簡單的鏡像濾波器 (quadrature mirror filter，QMF) 分解成$\mathbf{x}_{L(m\text{-}1)}$、$\mathbf{x}_{V(m\text{-}1)}$、$\mathbf{x}_{H(m\text{-}1)}$和$\mathbf{x}_{D(m\text{-}1)}$四個區塊 [10]。

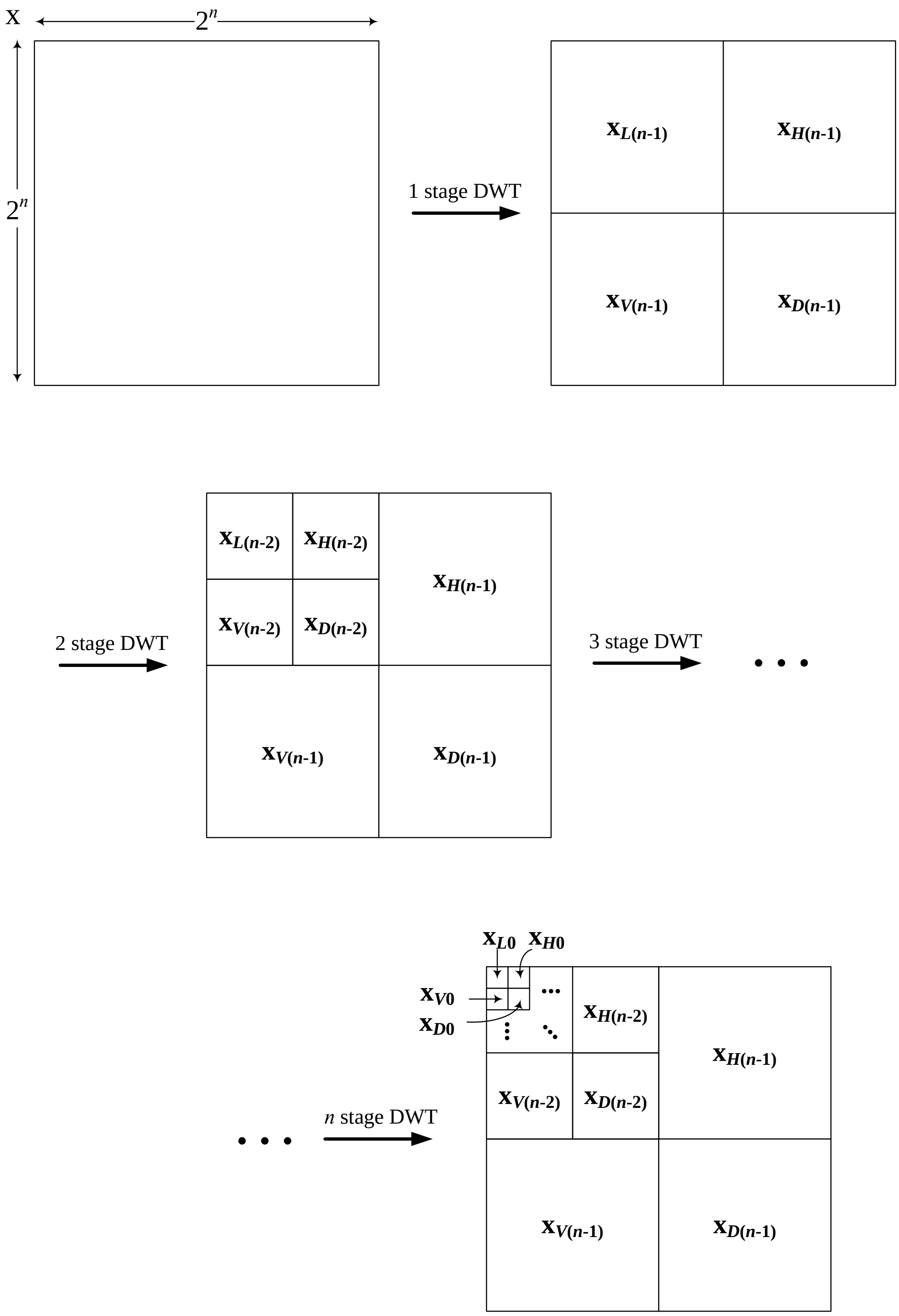

圖 2-3 n 階離散小波轉換示意圖

2.3 部分距離碼字搜尋演算法則

碼字的搜尋是計算輸入向量與碼字向量之間的量化誤差，也就是歐幾里得平方距離，找出 k 個最小量化誤差的碼字，搜尋的過程包含了加法、乘法、邏輯判斷以及排序等運算，稱之為運算複雜度 (computational complexity)。運算複雜度可以說是評估 kNN 分類器的一個重要指標，而搜尋 k 個最接近碼字的程序是 kNN 分類器最費時的一環，於是降低運算複雜度便成為增進 kNN 分類器效能最重要的工作，在這一節裡，我們將介紹以小波轉換為基礎的部分距離搜尋演算法則 (Partial Distance Search，PDS) 來達到降低運算複雜度的目的。

假設 kNN 分類器中的設計集 S 中有 $\{y^1, y^2 \ldots, y^t\}$ 共 t 個碼字，每個碼字 y^j 的維度都是 $r = 2^n \times 2^n$ ，分為 N 個子集合： $S_1, S_2, \ldots, S_N$ ，每個子集合 S_i 都是一個關係類別 (associated class) ω_i ，即 S 分為 $\omega_1, \omega_2, \ldots, \omega_N$ 類別，令 k_i 是 S_i 中擁有最近 k 個向量的向量數量，$1 \le i \le N$ 。kNN 分類器即是將擁有相同維度的輸入向量 (input vector) x 分類至 ω_p ，用數學式表示為 $z = \arg\max_i k_i$ ，而設計集的關係集合示意如圖 2-4。

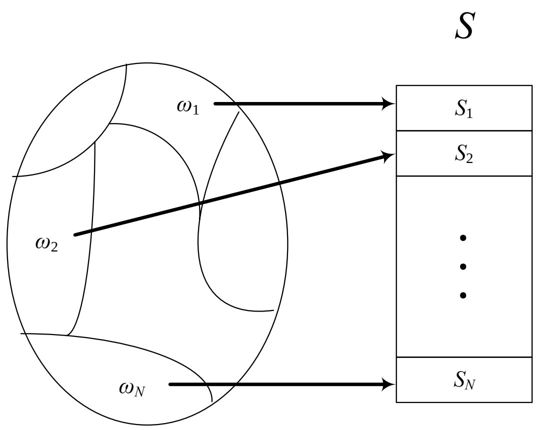

圖 2-4 類別-設計集映射示意圖

令 X 與 Y^j 分別代表 x 與 y^j 做 n 階離散小波轉換後的結果，同時令 $D(u,v)=\sum_i (u_i - v_i)^2$ 表示向量 u 和 v 的平方距離。對於一個正交的離散小波轉換 (orthogonal DWT)，我們可以得到這樣的結果：

$$D(x, y^j) = D(X, Y^j) \tag{2.3}$$

讓 $\mathcal{F}_k = \{f_1, f_2, \ldots, f_k\}$ 代表與輸入向量 x 最接近的 k 個碼字的index，且 $D(X, Y^{f_1}) \le D(X, Y^{f_2}) \le \ldots \le D(X, Y^{f_t})$。

部分距離搜尋演算法則的目的是在減少搜尋與 $\mathcal{F}_k$ 的運算時間。一般來說部分距離搜尋演算法則包含兩個步驟。給定一個下限值 (lower bound)，第一個步驟我們會先計算輸入向量與目標碼字部分維度的平方距離，若部分維度的平方距離超過此下限值，則目標碼字不需與輸

入向量經過所有維度的平方距離運算即可排除成為需要碼字的可能性，否則才需要進行第二個步驟，即算出輸入向量與目標碼字所有維度的平方距離來進行是否需要碼字的判斷。

令 X_i 與 Y_i^j 分別代表 X 與 Y^j 的第 i 個小波係數，這些係數在小波領域是以zig-zag的順序來標記，如圖2-5所示：

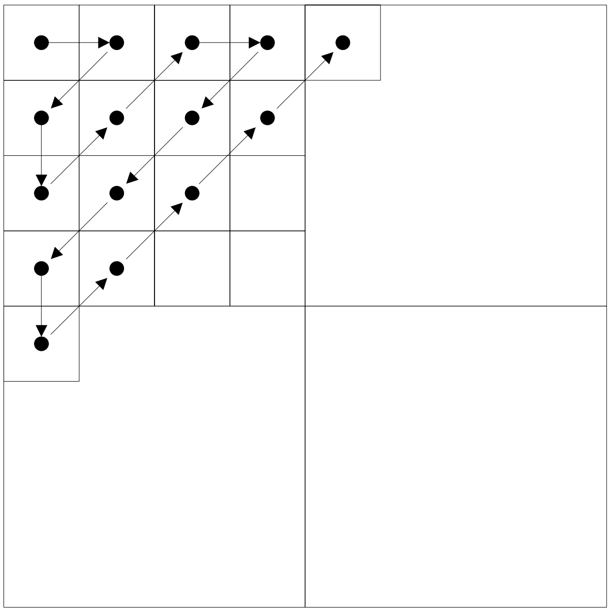

圖 2-5 zig-zag 排序方式

此外，令 $D^q\left(X,Y^j\right)=\sum_{i=1}^{q}\left(X_i-Y_i^j\right)^2$ 代表 X 與 Y^j 之間的部分距離，當 $D\left(X,Y^j\right)>D^q\left(X,Y^j\right)$，代表的是：

$$D\left(x,y^j\right)>D^q\left(X,Y^j\right) \quad (2.4)$$

令D_i代表在PDS過程中輸入向量x與目前最近第i筆距離碼字y^{f_i}的平方距離，$i=1,2\ldots,k$，則$D_1 \le D_2 \le \ldots \le D_k$。在執行 PDS 之前，我們將$z$的值設定為 0，將$D_i$設定為無限大，$i=1,2\ldots,k$，所有的$f_i$的值將在 PDS 執行過程中產生。

PDS找到$\mathcal{F}_k$的流程如下：檢查從Y^1到Y^t為止的每一個向量，對於每一個向量，做$\left|Y_1^j - X_1\right|$的計算，如果

$$\left|Y_1^j - X_1\right| > \sqrt{D_k} \tag{2.5}$$

根據 (2.4) 式，可以得到$\sqrt{D\left(x, y^j\right)} > \sqrt{D_k}$，於是可以得知$y^j$不屬於$k$個最近距離碼字，然後進行下一個碼字的比對運算。如果 (2.5)式不成立，則繼續進行接下來的 PDS 運算。從$q=2$開始，首先計算$D^q\left(X, Y^j\right)$，若是$D^q\left(X, Y^j\right) > D_k$，根據(2.4)式得知$\sqrt{D\left(x, y^j\right)} > \sqrt{D_k}$且剔除$y^j$屬於$k$個最近距離碼字的可能性；否則進行下一個$q$值重覆相同的處理程序直到$q = 2^n \times 2^n$為止。其中，部分距離$D^q\left(X, Y^j\right)$可以展開如下：

$$D^q\left(X, Y^j\right) = D^{q-1}\left(X, Y^j\right) + \left(X_q - Y_q^j\right)^2 \tag{2.6}$$

因此，對於新值q的部分距離可以累加前一個q值的部分距離而僅須計算$\left(X_q - Y_q^j\right)^2$。

重複的PDS處理程序會持續到y^j被剔除為k個最近距離碼字的可能或是q值達到$2^n \times 2^n$為止。當$q = 2^n \times 2^n$時$D^q\left(X, Y^j\right)$會與D_k進行比較，此時$D^q\left(X, Y^j\right) = D\left(X, Y^j\right)$，若是$D\left(X, Y^j\right) < D_k$，則$f_k$將從$\mathcal{F}_k$中移除並將$j$加入$\mathcal{F}_k$，再對$\mathcal{F}_k$中的所有元素重新排序。當碼簿中的所有碼字都完成搜尋處理後，我們便計算每一個類別ω_i的k_i，擁有最高k_i的類別將會被指定為x所屬的類別，這便完成了以PDS為基礎的kNN分類。