

4.2. Clasificación de imágenes por prototipos

Los clasificadores prototipo **LVQ**, **k-NN** y **L-k-NN** tienen en común el principio que mantengan copias de muestras de entrenamiento en la memoria, y la clasificación, la decisión $g(x)$ se basa en las distancias entre los prototipos memorizados y el vector entrante x . Los vectores de entrenamiento se conservan como tales o se utiliza algún tipo de fase de entrenamiento para extraer propiedades de una multitud de vectores de entrenamiento para cada uno de los prototipos memorizados. En cualquier caso, los clasificadores prototipo son representantes típicos de los métodos de clasificación no paramétricos.

1. Clasificadores de los pixeles k vecinos más cercanos

En un clasificador del *k-vecino* más cercano (**k-NN**), cada clase está representada por un conjunto

de vectores prototipo. Los *k-vecinos* más cercanos de un vector de patrón de entrada son los que se encuentran entre todos los prototipos y la etiqueta de clase se decide por la regla de votación mayoritaria. Un posible empate de dos o más clases se puede romper, por ejemplo, disminuyendo k por uno y revotándose.

En el reconocimiento de patrones clásico, el método de clasificación no paramétrico **k-NN** ha sido muy popular desde la primera publicación de Fix y Hodges y una importante prueba de precisión limitante de Cover y Hart. La regla de **k-NN** incluso ahora debería considerarse como una especie de clasificador de referencia, contra el cual otros se deben comparar clasificadores estadísticos y neuronales. Su ventaja es que no se necesita tiempo para entrenar al clasificador, y la desventaja correspondiente es que se necesitan enormes cantidades de memoria y tiempo durante la fase de clasificación. Se puede lograr una mejora importante en el consumo de memoria, manteniendo moderada la precisión de la clasificación, utilizando algún método de edición. Un algoritmo conocido como multiedición elimina los vectores espurios del conjunto de entrenamiento. Otro algoritmo conocido como condensación agrega nuevos vectores al clasificador cuando no puede clasificar el patrón correctamente. En ambos métodos, un conjunto de vectores utilizado originalmente como clasificador **k-NN** se convierte en un conjunto editado más pequeño para usarse como clasificador **1-NN**.

2. Aprendizaje de la cuantización de vectores

El algoritmo del cuantificador vectorial de aprendizaje (LVQ) [59] produce un conjunto de prototipos o vectores de patrones de libro de códigos m que pueden usarse en un clasificador **1-NN**. El entrenamiento consiste en mover un número fijo i de vectores de libro de códigos de forma iterativa lejos de las muestras de entrenamiento x_f . Las variaciones del algoritmo LVQ difieren en la forma en que se actualizan los vectores del libro de códigos. El proceso de aprendizaje LVQ puede ser interpretado como un movimiento iterativo de los límites de decisión entre clases vecinas, o como una forma de generar un conjunto de vectores de libro de códigos cuya densidad refleja la forma de las funciones definidas como

$$s(x) = P_j f_j(x) - \max_{k \neq j} (P_k f_k(x)) \quad (4.7)$$

Donde $j = g_{BAYES}(x)$. Tenga en cuenta que el conjunto cero de s consta del óptimo de Bayes de los límites de decisión.

3. Aprendizaje Clasificador k-NN

Además de editar reglas, se pueden aplicar algoritmos de aprendizaje iterativos a los clasificadores **k-NN**. Las reglas de aprendizaje del aprendizaje **k-NN (L-k-NN)** se parecen a las de LVQ pero al mismo tiempo el clasificador todavía utiliza la precisión de clasificación mejorada proporcionada por la regla de votación por mayoría. El desempeño de la norma. El clasificador **k-NN** depende de la calidad y el tamaño del conjunto de entrenamiento, y el rendimiento del clasificador disminuye si los recursos informáticos disponibles limitan la cantidad de vectores de entrenamiento que se pueden usar. En tal caso, la regla de aprendizaje **k-NN** puede utilizar mejor los datos disponibles al utilizar todo el conjunto de entrenamiento para optimizar la clasificación en función de un conjunto más pequeño de vectores prototipo.

Para el entrenamiento del clasificador **k-NN**, se utilizan tres esquemas de entrenamiento ligeramente diferentes como en el LVQ, las reglas de aprendizaje **k-NN** que utilizan un número fijo de vectores de código m_{ij} con etiquetas de clase predeterminadas j para clasificación. Una vez los vectores de código se han ajustado moviéndolos a dichas posiciones en el espacio de entrada dan una tasa de error mínima, la regla de decisión para un vector de entrada desconocido está basado en la etiqueta mayoritaria entre sus k vectores de código más cercanos. El objetivo de todas las reglas de aprendizaje es realizar la correcta clasificación de las muestras de entrenamiento más probables. Este objetivo se logra moviendo progresivamente de algunos de los vectores en la vecindad de un vector de entrada de entrenamiento hacia el espacio muestra de entrenamiento y algunas alejadas de ella. Para todas las reglas, las modificaciones a las de los vectores de código m_i se construyen según la regla LVQ:

$$m_i(t+1) = m_i(t) \pm \alpha(t)(x(t) - m_i(t)) \quad (4.8)$$

donde $x(t)$ es la muestra de entrenamiento en el paso t . Con signo positivo de $\alpha(t)$, el movimiento del vector de código se dirige hacia la muestra de entrenamiento, y con signo negativo lejos de él. La tasa de aprendizaje $\alpha(t)$ debería disminuir lentamente para hacer convergente el algoritmo; en la práctica puede ser suficiente utilizar un pequeño valor constante.

4.2.1. Clasificación de imágenes por distancia

La motivación para los clasificadores subespaciales se origina en la compresión y reconstrucción óptima de datos multidimensionales. El uso de subespacios lineales como modelos de clases se basa en el supuesto de que los datos dentro de cada clase se encuentran aproximadamente en un subespacio de dimensiones inferiores del espacio de patrones \mathbb{R}^d . Luego, un vector de una clase desconocida se puede clasificar según su distancia más corta a los subespacios de clase.

La media muestral $\hat{\mu}$ de todo el conjunto de entrenamiento se resta primero del patrón de vectores. Para cada clase j , se estima la matriz de correlación $\hat{\mathbb{R}}_j$ y su primera pocos vectores propios u_{1j}, \dots, u_{l_jj} se utilizan como columnas de una matriz base U_j . La regla de clasificación del algoritmo de compresión de información de características de clase (CLAFIC) se puede expresar como

$$g_{CLAFIC}(x) = \operatorname{argmax}_{j=1, \dots, c} \|U_j^T x\|^2 \quad (4.9)$$

El método del subespacio de aprendizaje promediado (ALSM) introducido es una versión de aprendizaje iterativo de CLAFIC, en donde las matrices de correlación de clases de muestra no normalizadas $\hat{S}_j(0) = \sum_{i=1}^{n_j} x_{ij}x_{ij}^T$ están ligeramente modificadas según la exactitud de las clasificaciones,

$$\hat{S}_j(k+1) = \hat{S}_j(k) + \alpha \sum_{i \in A_j} x_i x_i^T - \beta \sum_{i \in B_j} x_i x_i^T \quad (4.10)$$

Aquí x_{1j}, \dots, x_{n_jj} son las muestras de entrenamiento de la clase j , α y β son constantes pequeñas positivas, A_j es el conjunto de índices i para los cuales x_i proviene de la clase j pero se clasifica erróneamente en una clase diferente, y B_j consta de aquellos índices para los cuales x_i se clasifica en j aunque en realidad se origina en una clase diferente. Las matrices base U_j se recalculan después de cada época de entrenamiento como los vectores propios dominantes del \hat{S}_j modificado. Las dimensiones del subespacio l_j necesitan ser fijadas de alguna manera se ha desarrollado un algoritmo de búsqueda iterativo eficaz y una nueva solución de ponderación.

4.2.2. Clasificación de imágenes por distancia entre cadenas

Hay muchos tipos de grupos. El resto se basa en calcular la distancia entre vectores visibles. Además de los métodos que utilizan neuronas artificiales, existen otros métodos, incluidos métodos basados en la probabilidad de unirse a un grupo u otro (goteros basados en modelos ocultos de Markov) y métodos que utilizan lógica difusa basada en reglas.

Clasificador de distancia euclídea determinista

Clasificador determinístico y supervisado. Se basa en el cálculo de un centroide para cada una de las K clases en las que se divide el universo de trabajo. Esto puede verse como una indicación o paradigma de la naturaleza vectorial del grupo. Para muestras desconocidas, la distancia euclidiana de la muestra esperada se calcula para cada K muestras. un ejemplo desconocido:

$$d_E(X, Z_k) = \sqrt{X^T \cdot X - 2 \cdot X^T \cdot Z_k + Z_k^T \cdot Z_k} \quad (4.11)$$

Este grupo divide el área en secciones utilizando planos equidistantes de los nodos. La siguiente imagen muestra 3 categorías y 2 formas. Cada línea discontinua separa el punto más cercano al más joven (estos son los puntos negros).

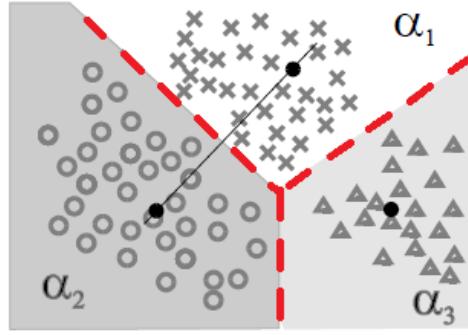


Figura 4.3 Separación lineal entre clases.

Para la clase K , se necesitan $\alpha_1, \alpha_2, \dots, \alpha_K$, con K prototipos Z_1, Z_2, \dots, Z_K . Para realizar un clasificado del patrón, siga las instrucciones como en la figura 4.4, el sistema de distribución. En el caso de tener K clases $\alpha_1, \alpha_2, \dots, \alpha_K$, se requieren K prototipos Z_1, Z_2, \dots, Z_K . Para clasificar el patrón X se calcula la distancia del vector de características X a los vectores de características de cada uno de los K prototipos (Z_1, Z_2, \dots, Z_K), clasificando X como perteneciente a la clase cuyo prototipo esté más próximo.

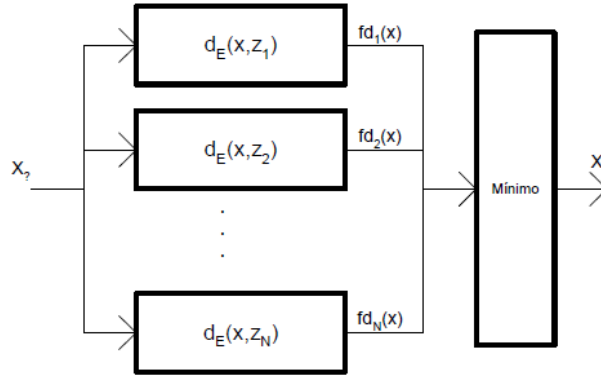


Figura 4.4 Clasificador euclídeo.

La discriminabilidad representada por la ecuación 4.11 se puede probar usando la raíz cuadrada; el término $X^T \cdot X$ también se determina porque es la misma para todos los grupos. Finalmente, invierte el signo y divide por 2, ver ecuación 4.12. La transferencia de signos será una función significativa con un valor alto, que representa pertenecer a la clase de X .

$$fd_k(X) = X^T \cdot Z_k - \frac{1}{2} Z_k^T \cdot Z_k \quad \forall k = 1, 2, 3, \dots, K \quad (4.12)$$

El cálculo de muestras es una heurística que crea muestras de una clase calculando los pesos de un conjunto de P elementos de la clase, lo que requiere muestras de muchos individuos para todos los N grupos. La matriz debe ser lo suficientemente grande como para mostrar la matriz de objetos relacionados.

$$Z_k = \frac{1}{P_k} \sum_{p=1}^{P_k} X_p \quad \forall X_p \text{ patrón de la clase } \alpha_k \quad (4.13)$$

Clasificador de distancia (aprendizaje supervisado)

Si el problema de clasificación se considera un problema de optimización, el vector X se puede fijar en una de las K categorías. Para un conjunto de K categorías $\{\alpha_1, \alpha_2, \dots, \alpha_K\}$, elija la solución al problema del vector desconocido con función discriminante $fd_i(X)$ dé un resultado máximo:

$$x \in \alpha_k \Leftrightarrow fd_k(x) > fd_j(x) \quad \forall j = 1, 2, 3, \dots, K / j \neq k \quad (4.14)$$

Las funciones discriminantes, son lineales y tienen la forma:

$$fd_k(x) = W_k^T \cdot X \quad \forall k = 1, 2, \dots, K \quad (4.15)$$

X es el vector asociado con el valor 1 y se trata como un elemento separado de la fila de la matriz. El objetivo es encontrar valor W_k que hagan de fd_k una función discriminante. Para resolver estos problemas se utilizan ecuaciones de regresión descendente, ecuaciones con valores constantes que cambian los parámetros de estas ecuaciones hasta llegar a una solución (no óptima) del problema. En el reconocimiento de patrones, este proceso secuencial se llama aprendizaje.

La diferencia con estos primeros puntos de referencia (no es necesario aprender) es que cómo se calculan las promociones es importante; sin embargo, en estos puntos de referencia las campañas seleccionadas se calculan mediante un proceso continuo. Los métodos de aprendizaje son particularmente útiles cuando se desconoce la ecuación y no se puede calcular una solución analítica. A continuación, defina el grupo euclidiano.

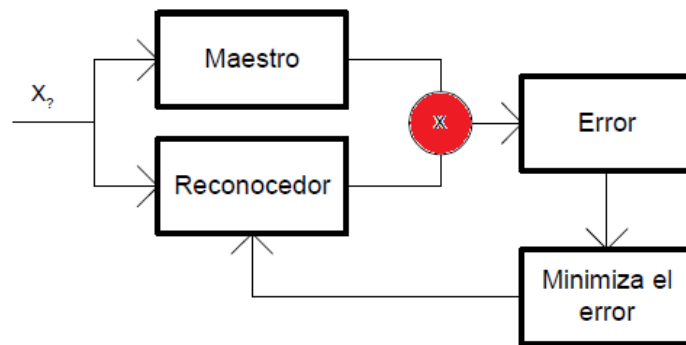


Figura 4.5 Reconocimiento con aprendizaje

El módulo principal calcula el error en función del estado anterior para decidir cómo cambiar los parámetros de clasificación para reducir el error.

Algoritmo de aprendizaje

El algoritmo de aprendizaje comienza con una función de clasificación aleatoria y la mejora gradualmente hasta encontrar la función de clasificación que coloca la mayor cantidad de parámetros en el mundo real. A veces detecta funciones discriminantes incorrectas durante el aprendizaje, pero los cambios en el aprendizaje cambian los parámetros de la función discriminante para reducir los errores en el reconocimiento de objetos. Este proceso se sigue repitiendo.

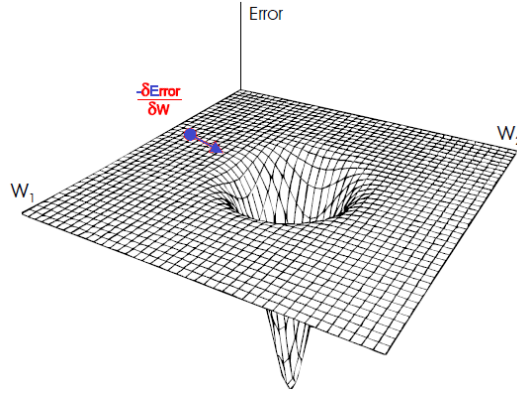


Figura 4.6 Error de la función W.

Deben tenerse en cuenta los errores en los parámetros W del clasificador. Para una W determinada, se produce un error. La derivada es una indicación de la reducción de errores que se produce en función de los datos objetivo (objeto) y los datos obtenidos a través del proceso de discriminación en curso.

$$\text{Error} = \text{diferencia entre lo que se desea obtener y lo que se obtiene} \quad (4.16)$$

El ejemplo que se muestra en la Figura 4.5 solo tiene dos lados, por lo que hay dos lados en la parte superior. La salida de la función de error con respecto a W muestra la pendiente de la pendiente. Por tanto, la forma en que se reduce el error está determinada por la dirección del flujo; El método de corrección W para reducir el error es saltar el factor de tiempo μ en la dirección donde ocurre el error. El costo:

$$W(t+1) = W(t) - \mu \frac{\partial \text{Error}}{\partial W} \quad (4.17)$$

El factor μ principal es la velocidad de aprendizaje, que se determina detalladamente para cada problema (mediante prueba y error), ya que depende del problema considerado y del modelo del problema involucrado. Elegir valores más pequeños de μ hará que el aprendizaje sea más lento, mientras que valores más grandes harán que se omitan las soluciones (no se ajusten a los agujeros más altos). Desafortunadamente, los mayores y menores dependen de cada pregunta, los métodos de enseñanza cambian y los errores cambian. En este momento, se cree que la causa del error es la siguiente:

$$\text{Error} = \frac{1}{4} (f d m - W_i^T X)^2 \quad (4.18)$$

donde

$$f d m = \begin{cases} +1 & \text{si } X \in \alpha_i \\ -1 & \text{si } X \notin \alpha_i \end{cases} \quad (4.19)$$

Nombrando al error e se tiene:

$$e = (f d m - W_i^T X) \quad (4.20)$$

Y también

$$\nabla \text{Error} = \frac{\partial \text{Error}}{\partial W} = \frac{1}{4} 2 (f d m - W_i^T X) (-X) \Rightarrow \nabla \text{Error} = -\frac{1}{2} e X \quad (4.21)$$

Por lo tanto

$$W(t+1) = W(t) + \mu \frac{1}{2} eX \quad (4.22)$$

La ecuación anterior muestra cómo cambiar los pesos de todas las funciones de clasificación para reducir el error. El algoritmo se repite hasta que el error cae por debajo de un nivel aceptable y el vector W cambia.

Algoritmo de aprendizaje (clasificador euclídeo) pseudocódigo.

1)

Tome un conjunto de P muestras de aprendizaje o conjunto de entrenamiento (CE).

$$CE = \{X_0, X_0, X_0, \dots, X_{P-1}\}$$

Inicie aleatoriamente los valores de W_i (pesos). También inicie $t = 0$ y $p = 0$. Fije un nivel de error máximo aceptable E .

2)

Presente la muestra de entrenamiento X_p (que pertenece a α_K) y se calcule las K funciones discriminantes.

$$fd_1(X_p), fd_2(X_p), \dots, fd_K(X_p)$$

3)

Para cada fd calcule el error cometido. Si el error es menor al E fijado PARE, en otro caso actualice W según:

$$W(t+1) = W(t) + 0.5 \cdot \mu \cdot error \cdot X$$

Regrese al paso 2.

Algoritmo distancias encadenadas

Este algoritmo de distancia en cadena es un algoritmo que no requiere información sobre el número de clases disponibles, solo cambia el umbral que coloca el nivel de histéresis en el centro de la clase, cuantas más familias cambia la clase. El algoritmo comienza a partir de los vectores de muestra X_1, X_2, \dots, X_P y selecciona aleatoriamente uno de ellos X_i . Luego los vectores se organizan en el siguiente orden:

$$X_i(0), X_i(1), X_i(2), \dots, X_i(3) \quad (4.23)$$

esta secuencia en la que el siguiente vector de la cadena está más cerca en distancia euclídea del vector anterior. Es decir $X_i(1)$ es el más próximo a $X_i(0)$, $X_i(2)$ es el más próximo a $X_i(1)$, etc. Siendo $X_i(0) = X_i$. Finalmente, elija un valor que haga referencia a la distancia máxima que puede existir entre los objetos del grupo. El primer objeto de clase 0 es Si la distancia entre los dos objetos del proceso es mayor que el límite, entonces la secuencia comienza una nueva clase; de lo contrario, el objeto buscado es el grupo 1 del objeto original.

Algoritmo Chain Map -

- 1) Sacar un elemento X al azar del conjunto de muestras de aprendizaje M y denominarlo V . Meter V en $C[0]$, e iniciar $a = 0$.
- 2) Calcule la distancia de V al resto de elementos de M y tome el elemento X con menor distancia a V .
- 3) Si la distancia de X a V es mayor al umbral incremente $a = a + 1$.
- 4) Meter en la clase $C[a]$, hacer $V = X$ y eliminar X de M .
- 5) Si M está vacío finalizar, en otro caso regrese al paso 2.

La ventaja de este algoritmo es que no requiere una gran cantidad de clases y se completa en un solo paso, por lo que es muy rápido. La dificultad radica en el problema de establecer valores y la solución se encuentra en la inicialización del algoritmo.