4 Teoría de clasificadores

4.1 Introducción

Clasificar un objeto consiste en asignarlo a una de las clases disponibles. Los objetos se pueden definir por una serie de características, como pueden ser el color de sus píxeles, su textura o su tamaño.

Para poder clasificar objetos es necesario definir las fronteras entre las diferentes clases. Normalmente estas fronteras se calculan mediante un proceso de entrenamiento en el que se usan las características de una serie de prototipos de ejemplo de las clases. Hablamos de fronteras por claridad, en general el clasificador infiere unas reglas de decisión durante el entrenamiento.

Clasificar un objeto desconocido consiste en asignarlo a la clase en la cual las características usadas durante el entrenamiento tienen más correspondencia con las características del objeto.

Se suele usar la clasificación frente a otras técnicas cuando los objetos tienen similitudes, pero sujetas a variaciones desconocidas. Si estas variaciones son muy pequeñas, existen otros métodos más sencillos para reconocer el objeto como por ejemplo el emparejamiento por plantilla (*template matching*). Los clasificadores se usan para:

- Segmentación de imágenes (por color, textura, etc.)
- Reconocimiento de objetos
- Control de calidad
- Detección de novedad (*novelty detection*), para detectar cambios o defectos en los objetos.
- Reconocimiento óptico de caracteres (OCR, Optical Character Recognition)

El proceso de clasificación, independientemente del tipo de clasificador seleccionado, consta de una serie de pasos:

1. Se reúnen muestras de objetos de clases conocidas. Se elige un juego de características (vector de características) apropiado y se calculan las características de los objetos de muestra (prototipos).
2. El conjunto de vectores de características se usa para entrenar el clasificador. Se calculan las fronteras entre las clases.
3. Se extraen las mismas características de los objetos desconocidos a clasificar.
4. El clasificador usa las fronteras calculadas durante el entrenamiento para decidir a qué clases pertenecen los vectores de características de los objetos que queremos reconocer.

Los pasos 1 y 2 se realizan fuera de línea, es decir, fuera de la aplicación de reconocimiento, mientras que los pasos 3 y 4 se realizan en línea.

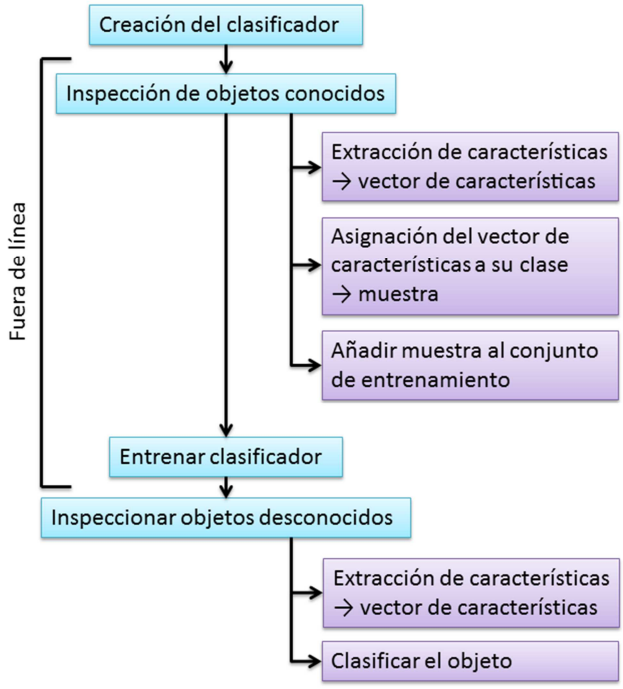

Figura 8: Pasos del proceso de entrenamiento y uso de un clasificador.

El software empleado en este proyecto, Halcon, proporciona varios tipos de clasificadores:

1. Hiper-cubos y clasificador de mínima distancia euclídea (para segmentación de color)
2. Modelos de Mezcla de Gaussianas, para segmentación de imágenes y tareas de clasificación generales (en inglés, Gaussian Mixture Models (GMM)).
3. Redes neuronales (Perceptrón Multicapa, en inglés Multilayer Perceptron (MLP)), para segmentación de imágenes, tareas de clasificación en general y OCR.
4. Máquinas de vectores soporte (Support Vector Machines (SVM)), también para segmentación de imágenes, clasificación en general y OCR.

Para poder decidir a qué clase pertenece un objeto es necesario conocer las diferencias existentes entre las clases, y las similitudes existentes entre los miembros de una misma clase. Este conocimiento se puede obtener analizando las características típicas de los objetos, como por ejemplo:

- Área
- Circularidad
- Rectangularidad
- Anisometría
- Etc.

Las características elegidas se agrupan en un vector de características que a su vez define el denominado espacio de características, donde cada característica se representa en un eje coordenado. El espacio de características puede tener cualquier dimensión.

La idea es observar cómo se disponen los vectores de características en el espacio de características para ver si los conglomerados (*clusters*) que se forman se pueden separar.

Precisamente, la tarea de un clasificador es separar los *clusters* (durante el proceso de entrenamiento), y asignar los vectores de características de entrada a las clases conocidas.

Los clasificadores que usan hiperplanos para separar las clases se denominan clasificadores lineales. Otros clasificadores construyen superficies arbitrarias que permiten separar *clusters* muy cercanos e incluso que se solapen. Estos últimos se denominan clasificadores no lineales.

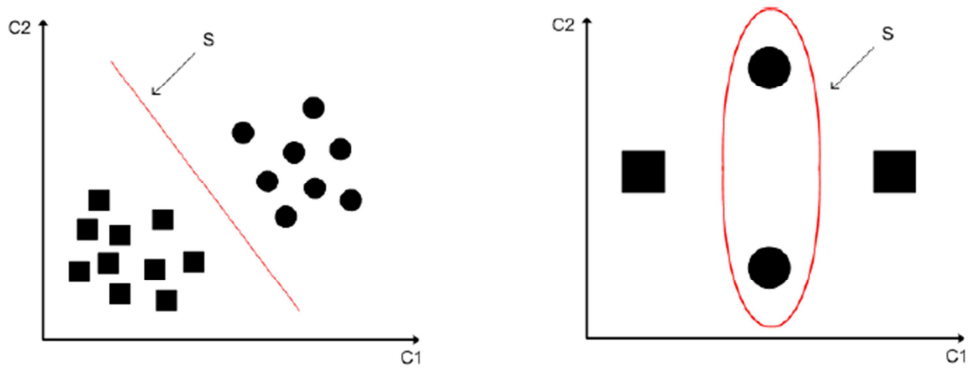

Figura 9: Ejemplos de problemas de clasificación lineal (izquierda) y no lineal (derecha).

4.2 Clasificadores de mínima distancia euclídea y de hipercubos

Los clasificadores de mínima distancia euclídea y de hipercubos son los clasificadores más simples. El primero construye hiperesferas centradas en los *clusters*, calculando los radios de manera que los *clusters* queden separados entre sí. El segundo, en lugar de hiperesferas utiliza hipercubos paralelos a los ejes del espacio de características, por lo que puede interpretarse como una umbralización en un espacio multidimensional. Los clasificadores de distancia euclídea sólo se usan en Halcon para segmentación de imágenes, ya que el método sólo es estable para vectores de características de baja dimensión.

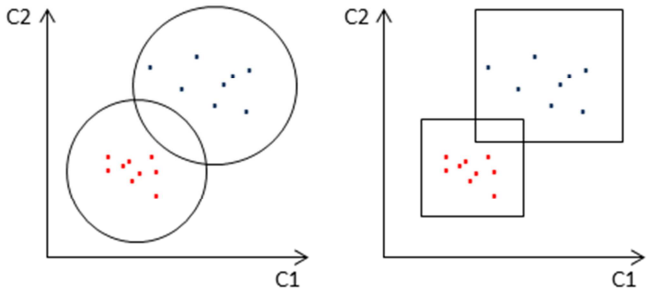

Figura 10: Superficies de separación del clasificador de mínima distancia euclídea (izq.) y de hipercubos (der.).

4.3 Clasificador GMM

El clasificador de modelos de mezcla de gaussianas es más complejo. Se basa en el Teorema de Bayes que, en resumen, nos dice que para minimizar la probabilidad de clasificar erróneamente un vector de características hay que maximizar la probabilidad de que el vector de características pertenezca a una clase. Se trata de maximizar esta probabilidad, denominada 'prioridad a posteriori'. El clasificador dividirá el espacio de características en regiones mutuamente disjuntas mediante hipersuperficies. Concretamente, las hipersuperficies se definen por los puntos en los que dos clases vecinas son igualmente probables. El Teorema de Bayes se expresa de la siguiente manera:

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

donde:

- $P(\omega_i)$ es la probabilidad de que la clase ω_i ocurra.
- $P(x)$ es la probabilidad de que el vector de características x ocurra.
- $P(x|\omega_i)$ es la probabilidad a priori de que el vector de características x ocurra, sabiendo que su clase es ω_i.
- $P(\omega_i|x)$ es la probabilidad a posteriori.

Se trata de maximizar la probabilidad a posteriori para todas las clases.

La probabilidad $P(x)$ es una constante si x existe y normalmente se elige la probabilidad $P(\omega_i)$ igual para todas las clases, es decir, igual a $1/m$ siendo m el número de clases total.

Para estimar la probabilidad $P(x|\omega_i)$ existen varios métodos, pero el más recomendable y usado en la práctica consiste en asumir que las probabilidades tienen una distribución normal, de manera que sólo hay que estimar dos parámetros: la media y la desviación típica (o la matriz de covarianzas).

Pero ocurre que, en ciertos casos, una distribución normal puede no representar suficientemente bien a una clase. Por ejemplo, las letras suelen tener varias grafías diferentes, y al hacer que estas grafías de diferente forma asociadas a la misma letra pertenezcan a una única clase, dentro de ésta habrá grandes variaciones del vector de características. Pero estas clases con grandes variaciones, donde existe una mezcla de densidades de probabilidad, se pueden descomponer a su vez en varias distribuciones normales. De ahí que este clasificador se llame Modelo de Mezcla de Gaussianas (en inglés Gaussian Mixture Models).

La idea es que el clasificador GMM usa las funciones de densidad de probabilidad de las clases y las expresa como combinaciones lineales de distribuciones Gaussianas. Haciendo una analogía con los clasificadores anteriores, podemos imaginar que el clasificador GMM construye elipsoides (n-dimensionales) alrededor de los centros de los *clusters*.

Los clasificadores GMM funcionan bien para vectores de características de pocas dimensiones (hasta 15 características aproximadamente), así que se pueden usar para segmentación de imágenes y clasificación de características generales, pero no para OCR. Sus usos típicos son la segmentación de imágenes y la detección de novedad (*novelty detection*). Precisamente la detección de novedad es una funcionalidad específica del clasificador GMM y permite rechazar los vectores de características que no pertenezcan a ninguna de las clases entrenadas. La detección de novedad también se puede aplicar con el clasificador SVM, pero éste sólo puede manejar problemas de dos clases (sólo se entrena una clase y se rechazan los vectores de características que no pertenezcan a esa clase).

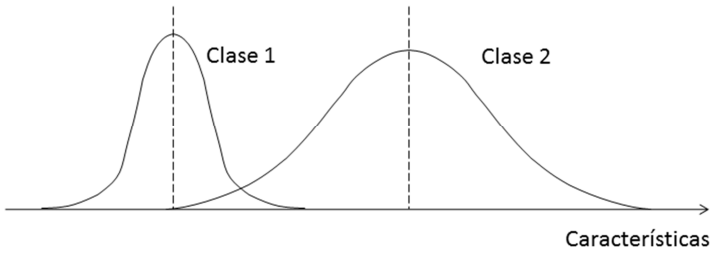

Figura 11: Clases con varianzas muy distintas.

Haciendo una analogía con el clasificador de mínima distancia, se puede considerar que los *clusters* se aproximan por elipses (la distribución del error). La distancia a la distribución gaussiana del error es un criterio de separación mejor que la distancia al centro del *cluster* cuando las varianzas de las distintas clases son muy distintas, como se puede observar en las figuras **Figura** 11 y **Figura** 12.

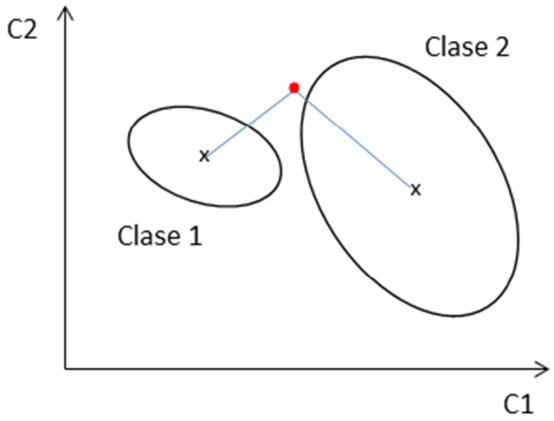

Figura 12: El vector de características (punto rojo) está más cerca de la elipse de error de la clase 2 aunque la distancia al centro del *cluster* de esa clase es mayor que la distancia al centro del *cluster* de las clase 1. El vector se asigna a la clase 2.

Los clasificadores MLP y SVM usan una aproximación geométrica al problema de separación de los *clusters*, en lugar de una aproximación estadística. Estos dos clasificadores construyen explícitamente hipersuperficies entre las distintas clases.

4.4 Clasificador MLP

Una red neuronal con una única capa es un clasificador lineal, es decir, separará las clases mediante hiperplanos. Por tanto, será aplicable cuando las clases verdaderamente sean linealmente separables, cosa que no siempre ocurre.

Para que una red neuronal pueda separar clases de forma no lineal, es necesario añadir más capas que se denominan capas ocultas. Así, la red constará de una capa de entrada, una o varias capas ocultas y una capa de salida. Con sólo una capa oculta ya es posible aproximar cualquier hipersuperficie de separación.

Las redes neuronales se entrenan haciendo uso de la denominada regla delta, que permite realimentar el error de clasificación para modificar los pesos de la red.

Una vez entrenada la red, el funcionamiento consiste en calcular para cada neurona la combinación lineal del vector resultado de la capa anterior (vector de características si es la primera capa):

$$a_j^{(l)} = \sum_{i=1}^{n_l} w_{ij}^{(l)} x_i^{(l-1)} + b_j^{(l)}$$

Donde:

- $x_i^{(0)}$ es el vector de características
- $x_i^{(l)}$ es el vector resultado de la capa *l*
- $w_{ij}^{(l)}$ y $b_j^{(l)}$ son los pesos de la capa *l* y términos independientes respectivamente.

Estos resultados son la entrada de la denominada función de activación que es una función no lineal derivable:

$$x_j^{(l)} = f_{act}(a_j^{(l)})$$

Para las neuronas de la capa oculta la función de activación es la tangente hiperbólica:

$$f_{act}(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Para las neuronas de la capa de salida se usa la función *softmax*:

$$f_{sal}(x) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$$

Donde n es el número de nodos de la capa de salida. Esta función asegura que todos los valores de salida están entre cero y uno, y que su suma vale uno.

El proceso de entrenamiento consiste en el denominado algoritmo de *retropropagación* del error (*Backpropagation*). Se introducen en la red vectores de características cuya clase, y por tanto su salida, es conocida. El vector de características es procesado produciendo un resultado en la capa de salida. Esta salida es comparada con la salida deseada, es decir, la que nosotros hayamos especificado por pertenecer a cierta clase. El error entre la salida real y la deseada se realimenta hacia atrás por las capas, para variar los valores de los pesos con el objetivo de ir minimizando el error de en cada iteración del proceso de entrenamiento.

En Halcon, el proceso de ajuste de los pesos se realiza con un algoritmo numéricamente estable que consigue mejores resultados que el algoritmo de *retropropagación* del error clásico.

El clasificador MLP devuelve el nivel de confianza de la clasificación, pero en contraposición a las probabilidades que devuelve el clasificador GMM, los valores pueden ser inesperados para los *outliers* (valores atípicos):

- El nivel de confianza puede ser alto para un patrón que está lejos del resto de patrones de su clase, pero cerca de la superficie de separación.

- Cuando las clases se superponen, la confianza puede ser baja para patrones que están contenidos en su *cluster* pero cerca de las superficies separadoras.
-

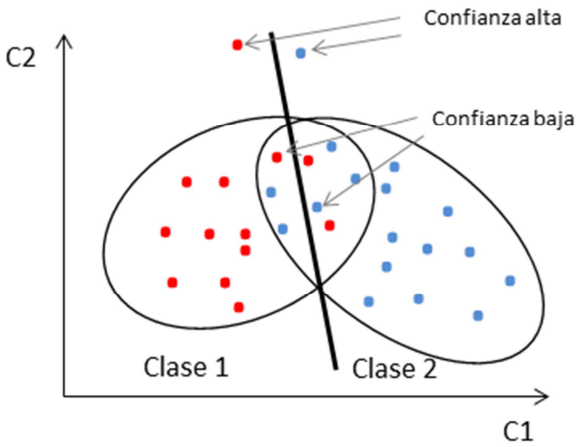

Figura 13: Valores inesperados de confianza para valores atípicos de las clases.

4.5 Clasificador SVM

Otro tipo de clasificadores que puede separar clases no linealmente separables son los SVM. Con este clasificador no se obtiene una hipersuperficie no lineal, sino que el espacio de características se transforma en un espacio con más dimensiones, de manera que las características sean separables linealmente (**Figura 14**).

Por su naturaleza, el clasificador SVM sólo puede separar dos clases. Pero es posible usarlo para resolver un problema multi-clase de dos formas:

- Se construyen pares de clases y se crea un clasificador binario para cada par. La clase que gana la mayoría de las comparaciones se elige como la más probable.
- Cada clase se compara con el resto de los ejemplos de entrenamiento (unión del resto de las clases) y se selecciona la clase que está a mayor distancia del hiperplano.

En un clasificador SVM, el hiperplano que separa las clases se elige de manera que el margen entre las dos clases sea lo mayor posible. Se define el margen como la menor distancia entre el hiperplano y el vector de características de cualquier prototipo de entrenamiento. Por tanto, se prueban varias superficies de separación y se elige la que tenga más margen. Los vectores de características de las dos clases que tengan exactamente la menor distancia al hiperplano se denominan vectores de soporte (**Figura 14**).

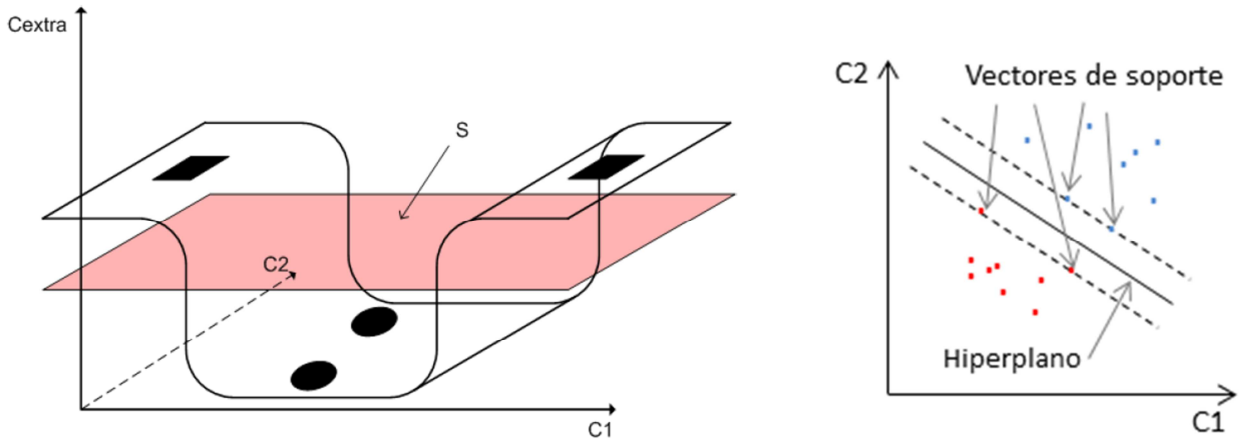

Figura 14: Transformación del espacio de características para separar las clases de la **Figura 9** con hiperplanos (izq.) y representación de los vectores de soporte (der.).

4.6 Consideraciones de diseño de clasificadores

4.6.1 Elección del clasificador más adecuado para la aplicación

El clasificador MLP consigue un buen porcentaje de acierto en el reconocimiento y además es muy rápido. Sin embargo el proceso de entrenamiento puede llegar a ser muy lento, especialmente cuando los conjuntos de entrenamiento son muy grandes. Si la aplicación es restrictiva en el tiempo de procesamiento y el entrenamiento se puede realizar fuera de línea, entonces las redes neuronales son una buena elección. En su versión para segmentación de imágenes, los vectores de características que no coinciden con las clases disponibles se pueden asignar a una clase de rechazo. Pero en comparación con el clasificador GMM, la clase de rechazo puede verse influenciada por la presencia de *outliers*, siendo recomendable entrenar una clase de rechazo explícita.

El clasificador SVM tiene un ratio de acierto ligeramente superior al del clasificador MLP y el proceso de entrenamiento es más rápido, sobre todo para conjuntos de entrenamiento grandes. Sin embargo, el proceso de clasificación es más lento que en un clasificador MLP y necesita mucha más memoria.

El clasificador GMM tiene la ventaja de que, si se desea, los vectores de características que no se parecen suficientemente a ninguna de las clases entrenadas, se pueden asignar a una clase de rechazo con bastante fiabilidad. La desventaja del clasificador GMM es que no consigue ratios de acierto tan elevados como los clasificadores SVM y MLP. Además la dimensión máxima que puede tener el vector de características no es muy elevada (aproximadamente 15 características como mucho, mientras que los otros dos pueden manejar más de 500 características). Así que el clasificador GMM sólo se recomienda cuando los objetos se pueden describir con pocas características.

Los clasificadores de hipercubos y de mínima distancia euclídea se pueden usar cuando el vector de características tiene baja dimensión, por ejemplo para segmentar la imagen por color.

4.6.2 Elección de las características apropiadas

Las mejores características dependen de la aplicación concreta y de los objetos concretos que se pretende clasificar. No existen reglas fijas para seleccionar las características, es un proceso que necesita observación y experiencia. Pero podemos distinguir casos:

- Para clasificación general de objetos puede servir cualquier característica:
  - Características derivadas de la región asociada al objeto (forma)
    - Circularidad
    - Rectangularidad
    - Número de agujeros
    - Etc.
  - Color o textura del objeto (no a nivel de pixel)
  - Etc.
- Para segmentación de imágenes se suele usar como característica el color o la textura de los pixeles de imágenes multidimensionales (un pixel de la imagen multidemensional, una característica).
- Para reconocimiento óptico de caracteres (OCR) se usan características apropiadas para definir caracteres:
  - Anisometría
  - Compacidad
  - Convexidad
  - Momentos de distintos órdenes, centrales e invariantes (de la imagen binaria y de la imagen en escala de grises)
  - Nivel de intensidad de los pixeles en escala de grises (un pixel una característica)
  - Región del carácter como imagen binaria (un pixel una característica)
  - Nivel de intensidad de los pixeles en escala de grises con escalado al valor máximo de la región
  - Proyecciones horizontal y vertical (y también proyecciones invariantes)
  - Relación de aspecto
  - Etc.

4.6.3 Elección de ejemplos de entrenamiento

El clasificador necesita ejemplos representativos de las clases para 'aprender' las similitudes y variaciones de los objetos de las que ya se ha hablado. Por tanto, es importante que entre las muestras haya mucha variedad de ejemplos con las variaciones permitidas, es decir, que se desvíen del objeto ideal. Esto incluye muestras con ruido (ruido previsible o aceptable). Si no se siguen estas indicaciones es probable que el clasificador no tenga suficiente capacidad de generalización.

Si no se dispone de suficientes muestras reales de los objetos de todas las clases, o si las muestras no cubren todas las posibles desviaciones, es posible generar de manera artificial más prototipos mediante la modificación de las muestras disponibles. Por ejemplo, si se trata de segmentar una imagen por su textura, se puede añadir una cierta cantidad de ruido a las muestras disponibles. Si se trata de clasificar objetos, se puede variar ligeramente su tamaño y forma mediante operaciones morfológicas (erosiones y dilatados), se puede modificar la orientación, etc.

4.6.4 Preprocesamiento

Normalmente se procesa el vector de características de entrada al clasificador para ayudar al proceso de entrenamiento (puede disminuir el tiempo de entrenamiento al facilitar la convergencia del algoritmo) y también para mejorar el ratio de acierto.

El preprocesamiento más utilizado (y siempre recomendable) consiste en la normalización del vector de características. La idea es que tras la normalización, los vectores de características del conjunto de entrenamiento tendrán media nula y desviación estándar unidad. El proceso de normalización consiste en calcular la media y la desviación estándar por componentes de los vectores de características del conjunto de entrenamiento y realizar la siguiente operación para cada componente '*i*' de cada vector:

$$x_i^N = (x_i - \mu_{x_i})/\sigma_{x_i}$$

Donde:

$$\mu_{x_i} = \mathbb{E}[x_i]$$

$$\sigma_{x_i}^2 = \mathbb{V}[x_i] = \mathbb{E}\left[(x_i - \mu_{x_i})^2\right]$$

También es posible reducir la dimensión del vector de características realizando un análisis de las componentes principales (*Principal Components Analisys*, PCA). Para ello, en primer lugar, se normalizan los vectores de características y después se les aplica una transformación que diagonaliza la matriz de covarianzas. De esta forma se reduce la dimensión del vector sin perder mucha información.