UNIVERSIDAD DE LAS PALMAS
DE GRAN CANARIA

ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA

PROYECTO FIN DE CARRERA

Identificador biométrico off-line basado en el contorno de la firma y forma de la palma

Autor: Fernando Pitters Figueroa

Tutores: Dr. Carlos Manuel Travieso González
Dr. Jesús B. Alonso Hernández

Fecha: Julio 2017

UNIVERSIDAD DE LAS PALMAS
DE GRAN CANARIA

ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA

PROYECTO FIN DE CARRERA

Identificador biométrico off-line basado en el contorno de la firma y forma de la palma

HOJA DE FIRMAS

Alumno/a

Fdo.: Fernando Pitters Figueroa

Tutor/a

Fdo.: Dr. Carlos M. Travieso González

Tutor/a

Fdo.: Dr. Jesús B. Alonso Hernández

Fecha: Julio 2017

UNIVERSIDAD DE LAS PALMAS
DE GRAN CANARIA

ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA

PROYECTO FIN DE CARRERA

Identificador biométrico off-line basado en el contorno de la firma y forma de la palma

HOJA DE EVALUACIÓN

Calificación: _____

Presidente

Fdo.:

Vocal **Secretario/a**

Fdo.: Fdo.:

Fecha: Julio 2017

Contenidos

Índice de figuras

Índice de tablas

Glosario de términos

BBDD – Bases de datos

MOM – Modelos Ocultos de Markov

MVS – Máquina de Vectores Soporte

API – Application Programming Interface (Interfaz de programación de aplicaciones)

HTTP – Hypertext Transfer Protocol (Protocolo de transferencia de hipertexto)

SDK – Software Development Kit (Kit de desarrollo software)

IDE – Integrated Development Environment (Entorno de desarrollo integrado)

RAM – Random Access Memory (Memoria de acceso aleatorio)

GB – Gigabytes

SSD – Solid State Disk (Disco de estado sólido)

COIT – Colegio oficial de ingenieros de telecomunicaciones

IGIC - Impuesto General Indirecto Canario

Memoria

1 Introducción

Un sistema de identificación biométrico como el propuesto se engloba dentro del conjunto de técnicas conocidas como “Aprendizaje automático” caracterizadas por la capacidad de aprender a partir de unos datos característicos y posteriormente emitir predicciones basadas en esa experiencia de aprendizaje. En este capítulo se definen las técnicas de Aprendizaje Automático así como el marco técnico en el que se desarrolla el presente proyecto.

1.1 Biometría

La biometría es la disciplina aplicada al estudio de medidas obtenidas de los seres vivos. Así por ejemplo, las tecnologías que aplicadas al ser humano permiten captar los impulsos cerebrales mediante electroencefalogramas, las señales eléctricas del corazón a través de electrocardiogramas, obtener y analizar muestras de ADN, leer huellas dactilares de los dedos, todas se engloban dentro de la ciencia biométrica.

La aplicación directa de las medidas biométricas es la salud de las personas, a través de medidas cuantitativas biométricas se puede analizar el estado de salud llegando a obtener un diagnóstico para ciertas patologías. Por otro lado las cualidades intrínsecas discriminatorias de las medidas biométricas permiten identificar a distintos individuos o diferenciarlos dentro de un grupo. Por ello el estudio e implementación de la biometría sobre el ámbito de la seguridad se encuentra en continuo crecimiento.

Una de las aplicaciones principales de la biometría es la identificación y la verificación de un individuo para el acceso a un servicio seguro o un espacio restringido. Se pueden distinguir tres grupos de sistemas de autenticación en función del tipo de característica en el que se basan [1]: algo que el individuo conoce, algo que el individuo posee y por último algo que el individuo es o hace. Las tecnologías basadas en biometría se engloban en este último grupo, es decir, cualidades intrínsecas a un individuo por su naturaleza biológica o por su conocimiento adquirido.

Existen varias opciones de acreditación que actualmente cubren las necesidades de reconocimiento no basados en medidas biométricas.

- Contraseña o pin. Se recurre a la memorización de un número secreto por parte del usuario. Está extendido de forma masiva en los sistemas online actuales, pero los riegos son significativos [4]. Una contraseña segura requiere una longitud elevada, caracteres aleatorios incluyendo alfanuméricos y especiales, única por sistema. A pesar de ser conocidos los problemas de contraseñas débiles, la gran mayoría de usuarios hace uso de ellas, principalmente por desconocimiento.
- Documento de identidad. Documento emitido por entidades públicas que tiene un reconocimiento extendido para autenticar a la persona portadora del documento. El principal hándicap es el requerimiento físico de la persona ante la autoridad validadora. Cabe destacar que el documento de identidad, al hacer uso de una fotografía y en estos últimos años la inclusión de las huellas dactilares ya cuentan con marcadores biométricos.

- Mensajes unipersonal a través de correo electrónico o móvil. Delegan la verificación del usuario a un sistema de terceros con la seguridad disponible en los mismos. A través de estos sistemas se comunica al usuario un código temporal que permite su entrada al sistema. Son muy comunes para los procesos de recuperación de contraseña o la autenticación en 2 pasos.
- Memorias externas con certificado electrónico. Requieren de un protocolo mediante el cuál validar dicho certificado y de una interfaz hardware para leerlo, como pueden ser una conexión USB o un lector de tarjetas.

La necesidad de sistemas más seguros y fiables es la principal razón del auge de medidas biométricas para el proceso de reconocimiento de personas, se denominan sistemas biométricos de reconocimiento.

1.1.1 Sistemas biométricos de reconocimiento

Como se ha mencionado se persiguen dos propósitos a través de reconocimiento: la identificación o la verificación de una persona ante un sistema.

- La identificación intenta encontrar para una muestra de entrada la opción, dentro del grupo obtenido a través de análisis previos, que más se asemeja y que mejor resultado ofrece. Se da a la salida por tanto la correspondencia más probable en caso de superarse los márgenes de similitud establecidos o la no pertenencia de las muestras al conjunto.
- La verificación busca confirmar que una muestra indicada como perteneciente a cierto individuo ciertamente se corresponde con una identificación correcta en base a los datos de análisis previos. De igual forma se obtiene un valor de semejanza o similitud y en caso de superar un nivel establecido se ofrecerá una resolución positiva.

Ciertamente las características biométricas que se pueden obtener del ser humano ofrecen ciertas ventajas para el reconocimiento pero no existe ninguna característica biométrica que sea completamente estable y distintiva de forma unívoca para aplicar en todas las situaciones y grupos [3]. Es importante destacar también que numerosos factores influyen en las muestras biométricas y el resultado del sistema [2].

- Variaciones de medidas en grupos y personas. Las características biométricas y la información capturada de las mismas por los sistemas puede estar afectada por cambios en la edad, el ambiente, enfermedades, estrés, aspectos socioculturales...

- Sensores. Multitud de variables afectan a las medidas biométricas, lo que se traduce a la salida de los mismos como ruido en las muestras: la edad y la calibración de los sensores, variabilidad entre los distintos dispositivos usados, las condiciones del entorno que influyen en el sensor como la luz, el sonido, el uso de la interfaz del sensor.
- Características y procesamiento. La necesidad de transformar los datos ofrecidos por los sensores en muestras adecuadas para el sistema influye notablemente en los resultados de comparación.

En general, las propiedades que un método biométrico debe poseer y a su vez tratar de maximizar son:

- Unicidad: la información debe ser lo suficientemente distintiva para el grupo en estudio.
- Permanencia: el rango temporal en el que el marcador es viable debe ser superior al tiempo en el que el sistema biométrico será usado.
- Universalidad: cobertura de dicha propiedad sobre la población en estudio. No será posible usar alguna característica si hay individuos que no cuentan con ella o no es correcta.
- Disponibilidad: accesibilidad para obtener la medida teniendo en cuenta el coste de sensores y la posibilidad de uso de los mismo o sus requisitos.
- Comparabilidad: facilidad para comparar y obtener resultados. El tiempo puede ser un factor limitante para el sistema o la característica.
- Invasividad: la inferencia sobre el sujeto debe ser mínima.
- Rendimiento: exactitud, velocidad y errores acotados.
- Aceptabilidad: la medida y la resolución del sistema debe contar con el soporte del entorno en el que será implantado.
- Robustez: dificultad para falsear y engañar medidas.

La ciencia de la biometría se encuentra en continua evolución por el rápido cambio en otras disciplinas científicas sobre las que se sustenta: biología, genética, química, procesamiento digital, matemáticas, electrónica. A continuación se recogen una lista de sistemas biométricos actuales y una posible categorización.

1.1.2 Tipos

Los sistemas biométricos pueden clasificarse en dos grandes grupos en función del tipo de medida biométrica analizada: biológicos y conductuales.

- Sistemas biológicos. También denominados fisiológicos, miden características propias de cada individuo dada su naturaleza biológica.

  A su vez los sistemas biológicos se pueden separar en:

  - Bioquímicos.
  - Visuales.
  - Espaciales.
  - Olfativo.
  - Auditivo.

- Sistemas conductuales. También referidos como comportamentales están basados en características adquiridas o aprendidas por el individuo.

En la tabla 1.1 se recogen varias métricas biométricas usadas en la actualidad [1].

Tipos de sistemas biométricos usados en la actualidad.

DNA. Análisis de cadenas de DNA.	Biológico / Bioquímico.
Forma de la oreja.	Biológico / Visual
Iris. Escaneo del iris.	Biológico / Visual
Retina. Patrones en las venas en el fondo del ojo.	Biológico / Visual
Rostros/Caras. Características	Biológico / Visual
Huella dactilar. Análisis de las crestas y valles de la piel.	Biológico / Visual - Espacial
Geometría dedos. Información 2D o 3D la forma de los dedos.	Biológico / Visual - Espacial
Geometría mano. Forma de la mano y dedos.	Biológico / Visual - Espacial
Paso. Análisis de la forma de andar de un indifiviu.	Conductual
Olor corporal.	Biológico/ Olfativo
Firma.	Conductual
Mecanografía. Estilo de introducción de textos en teclados físicos.	Conductual
Venas.	Biológico / Visual
Voz.	Biológica / Auditiva y Condutual

Tabla 1.1: Sistemas biométricos usados en la actualidad junto con el tipo al que pertenecen

1.2 Aprendizaje automático

El Aprendizaje Automático, quizá más conocido como “Machine Learning”, es una disciplina en auge que se ha desarrollado en los últimos 50 años y de forma exponencial en las últimas dos década, en paralelo con el desarrollo de la computación y su crecimiento. Está estrechamente relacionado además con otras materias muy sonoras como inteligencia artificial, minería de datos, “big data” (Nombre en inglés el análisis de datos que no son manejables a través de las técnicas comunes por su alto volumen). Existen referencias en la literatura [6][7][8] que incluyen todas estas técnicas bajo la terminología de Ciencia de Datos (En inglés “Data Science”).

1.2.1 Definición

En las ciencia de procesamiento de información y computación se conoce como Aprendizaje Automático al proceso en el que a partir de un conjunto de datos de entrada se obtiene como salida una predicción sobre los valores de entrada, como pudiera ser la pertenencia a un grupo, valores futuros estimados, valores omitidos. A través de la “experiencia” de los datos de entrada se busca maximizar el grado de acierto de la predicción.

La aplicación de estas técnicas se encuentran sumamente extendidas. Algunos ejemplos en donde se hace uso de dichos procesos de forma intensiva son:

- En Economía para predicciones en mercados bursátiles. Cabe destacar que de acuerdo a [6] aproximadamente el 50% de las transacciones en los mercados de valores son realizadas en la actualidad por procesos automatizados.
- Gestión de contenido en redes sociales o medios de comunicación para dar prioridad en base al análisis del comportamiento de usuarios. La cantidad de información en las fechas actuales puede llegar a ser abrumadora por lo que se buscan procesos para priorizarla de forma personalizada para cada usuario.
- Los gestores de correo electrónico, a través de análisis del texto, ordenan las bandejas de entrada priorizando los que pueden ser de mayor relevancia para el usuario e incluso descarta mensajes que se consideran contenido no deseado.
- Recomendación de contenidos en plataformas online de películas, música, libros o compras.
- Marketing y publicidad, donde se analiza el comportamiento o características de un individuo para ofrecer contenido publicitario más acertado.

- Detección de fraude y análisis de riesgos en sistemas transaccionales.
- Detección de amenazas de seguridad.
- Predicciones meteorológicas.
- Conducción autónoma. Aplican numerosos procesos automatizados para evaluar grandes cantidades de datos obtenidos por sensores y cámaras, y obtener un respuesta de conducción adecuada y acorde a lo que una persona haría al volante.

Las ventajas de utilizar estos sistemas para delegar la toma de decisiones se hacen patentes por el éxito que las acompaña. Se pueden enumerar por ejemplo la automatización de procesos, eficiencia, eficacia y sobretodo la escalabilidad. Las cantidades de datos que estos sistemas pueden procesar no son manejables a nivel humano.

Muchos algoritmos, sistemas o procesos se engloban bajo la disciplina del Aprendizaje Automático pero es conveniente definir los diferentes tipos que existen así como ubicar el lugar que ocupan los sistemas biométricos bajo estudio dentro de esta categorización.

1.2.2 Tipos

La clasificación más extendida en la literatura para los sistemas de Aprendizaje Automático identifica:

- Sistemas supervisados. Se basan en el conocimiento de la etiqueta o clase a la que pertenece cada muestra de los datos de entrada. Para una muestra de entrada dada la salida del sistema es la predicción de la clase a la que pertenece.
- Sistemas no supervisados. No se dispone de una clasificación para el conjunto de datos de entrada por lo que las características comunes deben ser inferidas de los mismos datos para poder tener así grupos de muestras a la salida. Adicionalmente pueden permitir obtener información subyacente o variables “latentes” que podrían comportarse como mejores características de predicción.

Se pueden considerar además los sistemas semi-supervisados en los que se disponen de datos etiquetados y sin etiquetar.

Otra posible clasificación puede ser agrupar las diferentes técnicas en función del tipo de problema que tratan de resolver.

- Clasificación. Buscan determinar la clase discreta a la que pertenece cada muestra del conjunto de datos analizado. Los sistemas de identificación biométricos pertenecen a este grupo.
- Regresión. Predicción de valores reales y continuos de salida para diferentes clases. Casos de usos son predicciones en los mercados de valores, pronósticos de demanda, estimación de precios, predicción del tiempo y eventos deportivos.
- Recomendación. Estimar las mejores opciones dentro de las alternativas posibles en función de ciertos valores o gustos dados como entrada. Entran en esta categoría las recomendaciones de productos y contenidos, asesoramiento en candidatos para un empleo, sistemas de citas online.
- Deducción. Inferir valores faltantes en un conjunto de entrada. Se aplican por ejemplo a datos de usuarios con parámetros ausentes, incorrectos o perdidos en ámbitos muy variados: información médica, de clientes o usuarios, valores censales de la población.
- Clustering (Agrupación). Infieren grupos a partir de los datos sin que deban corresponderse con clases reales.

También es posible discernir en base a la disponibilidad de los datos de entrada al proceso en sistemas estáticos y dinámicos.

- Sistemas estáticos u offline. El conjunto de datos de entrada se encuentra almacenado tras un proceso de digitalización, por ejemplo mediante escaneado o fotografiado. Es importante tener en cuenta ciertos errores añadidos a las muestras por el proceso. Se hace patente también la pérdida de información que podría determinar características clave como puede ser la variable temporal del trazo de una firma.
- Sistemas dinámicos u online. Las muestras de entrada son obtenidas directamente de la fuente, como pudieran ser sensores o datos analíticos en tiempo real. También pueden existir errores intrínsecos a los sistemas de obtención de los datos.

Dadas todas las técnicas y procesos descritos se puede encuadrar el presente sistema biométrico de identificación a través de la forma de la mano y el borde de la firma como un sistema supervisado orientado a la clasificación de individuos y con un conjunto de muestras de entrada, firmas manuscritas y manos escaneadas, almacenadas digitalmente de forma offline.

1.2.3 Aplicación de un Sistema Automático Supervisado

Se suele denominar al proceso de crear un sistema de procesamiento automático y aplicarlo sobre un conjunto de datos como creación de un “Modelo” de aprendizaje automático. Se refiere esto último a un proceso iterativo en el que el conjunto de las partes va evolucionando para llegar a obtener el mejor resultado en cuanto a predicciones.

En la figura 1.1 se plantea el flujo para la creación de un Modelo de Aprendizaje Automático.

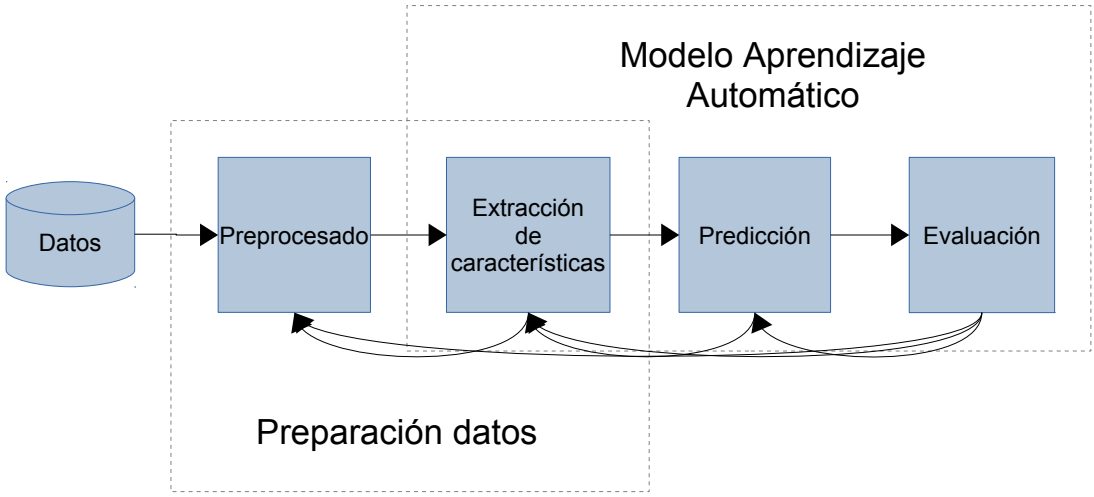

Figura 1.1: Flujo de creación de un Modelo de Aprendizaje Automatizado

Las partes que conforman la creación del modelo se describen a continuación y serían: obtención de los datos de entrada, preparación de los datos mediante preprocesado y posterior extracción de las características más relevantes, creación del algoritmo de predicción y obtención de la salida del sistema.

Como se ha comentado el desarrollo de las distintas etapas condicionan y hacen evolucionar las etapas anteriores a través de sucesivas iteraciones. El proceso para evaluar la eficacia del sistema viene dado por la fase de evaluación que permite la comparación de las predicciones con valores reales y así conocer el desempeño que presenta.

1.2.3.1 Entrada de datos

Los datos se presentan en numerosos formatos: bases de datos SQL (Lenguaje de consulta estructurados, o “Structured query language” en inglés), almacenamiento

NoSQL (consultas no estructuradas), flujo de datos en directo, archivos de logs, etc. En todos los casos es necesario una revisión de los datos disponibles buscando datos incompletos o corruptos que bloqueen por completo al sistema. Es necesario definir la interfaz de entrada al sistema para dichos datos, por ejemplo, las bases de datos deben ser leídas, los archivos cargados y los streams (flujos de datos) descompuestos para crear una estructura de datos adecuada para el sistema.

1.2.3.2 Preprocesado de datos

Se requiere nuevamente una revisión de los datos de entrada, buscando esta vez resolver pequeñas incoherencias que puedan ser subsanadas. Se puede tener por ejemplo palabras mal deletreadas en bases de datos de texto escrito, muestras con ruido por los procesos de obtención de los datos o el entorno de los mismos, datos incompletos para algunas muestras. Se tratará de normalizar el conjunto de datos y solventar los inconvenientes mencionados, en caso de no ser posible puede ser más recomendable eliminar la muestra.

1.2.3.3 Extracción de características

Es el proceso para obtener los patrones que caracterizan a los datos de entrada y permiten al sistema automático aprender de los mismos y realizar las predicciones. Las características pueden ser muy variadas, con forma homogénea constituidas todas por la misma métrica o heterogéneas de diferentes tipos. Se considera así la transformación de los datos iniciales de forma que los vectores de características finales permitan obtener la mejor respuesta del sistema.

Es una de las partes claves del todo el sistema. Llegar a obtener las características más significativas de un modelo de entrada y que el posterior algoritmo predictor sea capaz de interpretarlo correctamente es un proceso costoso y junto con la etapa previa de preprocesado de datos pueden llegar a ocupar la mayor parte del tiempo de desarrollo.

1.2.3.4 Función predictora

Es la fase analítica del sistema, donde a través de un método o función se llevan a cabo las predicciones para cada muestra de entrada. Existen diferentes técnicas utilizadas como función de predicción, pero se pueden englobar en general en modelos generativos basados en técnicas estadísticas y modelos discriminativos basados en reglas lógicas o geométricas.

El proceso requiere que la función de predicción aprenda a través de las muestras de entrada, lo cuál se conoce como proceso de entrenamiento, para una vez listo pueda pasar a emitir posibles salidas para datos de entrada nuevos.

1.2.3.5 Evaluación

Es la última etapa a la que se recurre en el proceso de creación de un modelo de aprendizaje automático y permite valorar el comportamiento el sistema en su conjunto.

Durante el proceso de evaluación se enfrenta el modelo creado a través de la definición de las etapas descritas en los puntos anteriores, a datos de entrada nuevos para el clasificador. Se analiza la salida del mismo para comprobar su correspondencia con la salida esperada, puesto que para el conjunto de datos utilizado la salida predicha es conocida.

1.3 Objetivos y propuestas

El objetivo principal del presente desarrollo se puede definir como la implementación de un sistema identificador biométrico basado en la combinación de datos off-line del contorno de la firma y la forma de la palma de la mano a través de un proceso de aprendizaje automático.

Adicionalmente se analizarán los sistemas biométricos de forma independiente para las base de datos de manos y firmas, lo que permitirá obtener resultados de referencia así como la mejora del sistema global a través del desarrollo de los sistemas biométricos separados.

El proceso de creación de un modelo de aprendizaje automático establece, como se ha mencionado en el apartado 1.2.3, la revisión de distintas fases de forma iterativa que se pueden resumir en: tratamiento de datos, extracción de características, entrenamiento del clasificador y evaluación. Se propone por tanto un desarrollo en paralelo para la creación de sendos modelos de aprendizaje automático para manos y firmas siguiendo las etapas mencionadas. Tras poder analizar los sistemas independientes y establecer las características que reportan los mejores resultados se procederá a combinar las muestras de las distintas bases de datos en un sistema biométrico único y a su posterior evaluación.

Se establece para los modelos de aprendizaje automático el uso de los clasificadores Modelos Ocultos de Markov y Máquinas de Vectores Soporte, así como la combinación de ambos a través de la transformación del Kernel de Fisher.

Por último se propone la creación de una demo para dispositivos móviles Android, que a modo de ejemplo exponga los flujos de identificación para los sistemas biométricos independientes y combinados.

1.4 Estructura de la memoria

El estudio desarrollado para el presente Proyecto Fin de Carrera se ha estructurado en una memoria de 7 capítulos.

En el **Capítulo 1**, el capítulo actual, se han introducido los conceptos de biometría y los sistemas biométricos basados en procesos de aprendizaje supervisado.

En el **Capítulo 2** se expone un análisis detallado de los conjuntos de datos considerados para el presente trabajo, las bases de datos de manos y firmas, así como el tratamiento que se hace necesario sobre los mismos.

Las técnicas de extracción de características se describen en detalle en el **Capítulo 3,** definiendo los procesos realizados para obtener las distintas características que se utilizarán en las etapas posteriores.

En el **Capítulo 4** se profundiza en los modelos clasificadores que se han implementado: Modelos Ocultos de Markov, Máquinas de Vectores Soporte y la transformación mediante el Kernel de Fisher. Se introduce una breve descripción de los mismos y la aplicación que se hace de ellos.

El **Capítulo 5** recoge el proceso de evaluación de los sistemas implementados a través de los distintos experimentos llevados a cabo para el conjunto vectores de características extraídos. Se analizan los sistemas biométricos de forma independiente para firmas y manos y posteriormente el sistema biométrico combinado objetivo de este estudio.

Se incluye en el **Capítulo 6** una descripción detallada de la demo desarrollada para una aplicación móvil Android.

Por último, en el **Capítulo 7** se recogen las conclusiones obtenidas a través de los diferentes experimentos. Además se detallan posibles desarrollos futuros para la mejora del sistema.

Adicionalmente se incluyen tres Anexos:

- Anexo A : Análisis extendido de los Modelos Ocultos de Markov.
- Anexo B : Extensión en las Máquinas de Vectores Soporte.
- Anexo C : Propuesta de publicación del presente desarrollo para un congreso de IEEE (Institute of Electrical and Electronics Engineers).

2 Preprocesado de bases de datos

La primera etapa para la construcción de un sistema biométrico a través de un modelo de aprendizaje automático consiste en un análisis previo de los datos, revisando la disponibilidad de los mismos, el formato en el que se encuentran y la posibilidad de transformalos a una estructura de datos más adecuada para el conjunto final. Se recoge en este capítulo dicho análisis inicial, así como las modificaciones llevadas a cabo.

2.1 Introducción

En un sistema biométrico, las muestras de entrada, ya sean firmas manuscritas digitalizadas, manos escaneadas u otros patrones biométricos en general, pueden presentar diferentes formas. Podrían encontrarse diferentes tipos de datos, texto, vídeo, audio, imágenes, codificados a su vez en multitud de formatos. Además, para una medida biométrica dada, digamos la firma, el proceso de obtención de la misma a través de interfaces como sensores, escáneres o fotografiado puede introducir distorsiones o impurezas adicionales, que es necesario considerar para la extracción de los datos representativos finales.

Se hace por tanto necesario analizar los datos disponibles para el presente estudio y posteriormente realizar las modificaciones y conversiones pertinentes. Hay disponibles sendas bases de datos para firmas manuscritas y palmas de la mano. A continuación se mencionan los pasos descritos de análisis así como las transformaciones requeridas que buscan abstraer la muestra del proceso de obtención de la misma y del formato de datos en que se encuentran.

2.2 Base de datos de firmas

2.2.1 Análisis

La base de datos empleada se denomina “Off-line Handwritten GPDS300 Signature CORPUS”. Ha sido creada por el Grupo de Procesamiento Digital de la Señal del IDeTIC (Instituto para el Desarrollo Tecnológico y la Innovación en Comunicaciones), que permite su uso en investigación con fines no comerciales. Está compuesta por 300 firmantes que aportan cada uno 24 firmas originales escaneadas. Adicionalmente para cada individuo, la base de datos consta de 30 firmas falsas.

En su formato inicial, cada firma se encuentra almacena en formato Mapa de Bits de Windows (BMP, Windows Bitmap Picture), con un tamaño variable y en formato RGB de 8 bits de profundidad. Todas las imágenes son en blanco y negro, encontrándose la firma manuscrita en negro sobre un fondo blanco. El trazo de la firma tiene una definición aceptable, con un grosor variable determinado probablemente por el bolígrafo o lápiz utilizado por el firmante. Un detalle importante es la variabilidad de estilos. Algunas firmas están compuestas por un simple nombre, otras por un trazo abstracto al ojo ajeno, o lo más común, la combinación de ambos. También es destacable la aparición de ciertas marcas de ruido, que deben ser eliminadas durante el preprocesado.

A modo de ejemplo, se muestran a continuación (Figura 2.1) cuatro firmas genuinas junto con cuatro firmas falsas del mismo tipo o clase, elegidas en ambos casos de forma (pseudo) aleatoria.

Figura 2.1: Muestras aleatorias base de datos firmas

2.2.2 Cambio de formato

Tras el análisis previo, se opta por un cambio del formato de almacenamiento de la base de datos. El formato BMP, no es muy eficiente en el uso de espacio en disco, y a ello se suma que no se aprovechan las características propias de las muestras: todas se encuentran en blanco y negro. Por tanto, se recurre al formato TIFF (Tagged Image File Format), que permite el almacenamiento de imágenes binarias sin compresión.

Mediante el citado cambio de formato, se consigue una disminución notable del tamaño de la base de datos sin ninguna pérdida de información. De 1.3 Gigabytes iniciales se pasa a menos de 38 Megabytes.

2.2.3 Preprocesado

El esquema para el preprocesado seguido con la base de datos de firmas es el siguiente (Figura 2.2):

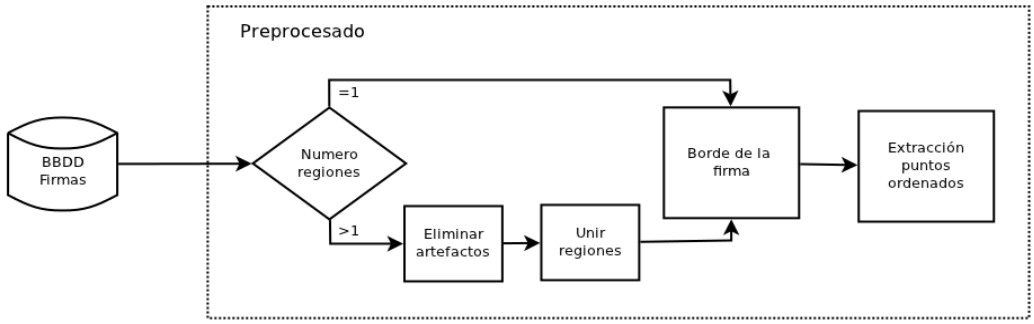

Figura 2.2: Procesado aplicado a la base de datos de firmas

Para cada muestra se determina si la firma consta de un único trazo o de varios objetos separados, pudiendo tratarse de partes de la misma firma o no. En el caso de no contar con un cuerpo unido, es necesario obtenerlo, puesto que se requiere el borde del mismo. Con la firma fusionada se separa el borde y se extraen las coordenadas del mismo de forma ordenada.

2.2.3.1 Eliminación de artefactos

Existen, aunque de forma minoritaria, algunas firmas que no poseen un trazo bien definido que en la imagen puede conllevar a la pérdida de la continuidad del mismo, y la aparición de puntos aislados. Del mismo modo, existen en algunas muestras píxeles o grupos de píxeles completamente separados, que dada su desconocida naturaleza no formarán parte del objeto firma final.

Con el fin de eliminar los mencionados puntos se procede como sigue. Se determinan el número de regiones y tras comprobar que existen más de una, se eliminan todas aquellos objetos que posean un área inferior a un número de píxeles establecido como umbral.

En Figura 2.3 tenemos un ejemplo de una firma que muestra ambas anomalías. La firma ha sido etiquetada por regiones, con un color distinto para cada una. Se encuentran resaltados tanto los píxeles aislados como el trazo indefinido.

Figura 2.3: Firma etiquetada por regiones

La elección del umbral, como suele ocurrir, se convierte en una relación de compromiso. Si es muy pequeño (Entre 5 y 20 píxeles), ruido indeseado puede pasar al proceso de unión y distorsionar el resultado, y por contra, entre mayor es el umbral (>30 píxeles), más información relevante puede ser eliminada. Por ello, se toma un valor de 25 píxeles de área para eliminar regiones pequeñas.

2.2.3.2 Unión de regiones

Tras la eliminación de objetos menores al umbral, las regiones restantes deben ser unidas. Este punto es de gran importancia puesto que las uniones se realizan mediante el trazado de líneas y, en consecuencia, el trazo de la firma original es modificado. Es por ello que las uniones deben que ser mínimamente intrusivas. Se han desarrollado cuatro alternativas.

Una primera aproximación basada en la unión de los centros de masas, o centroides, de las regiones más próximas entre sí. Aunque sencillo y eficiente, el resultado no cumple con los objetivos de la tarea. Las uniones se alejan de la naturalidad del trazo de la firma, pudiendo acarrear errores a la hora de identificar y clasificar.

A continuación, en la figura 2.4, se pueden observar dos ejemplos. En ambos casos las regiones a unir se deciden en función de la distancia de sus centros de masas.

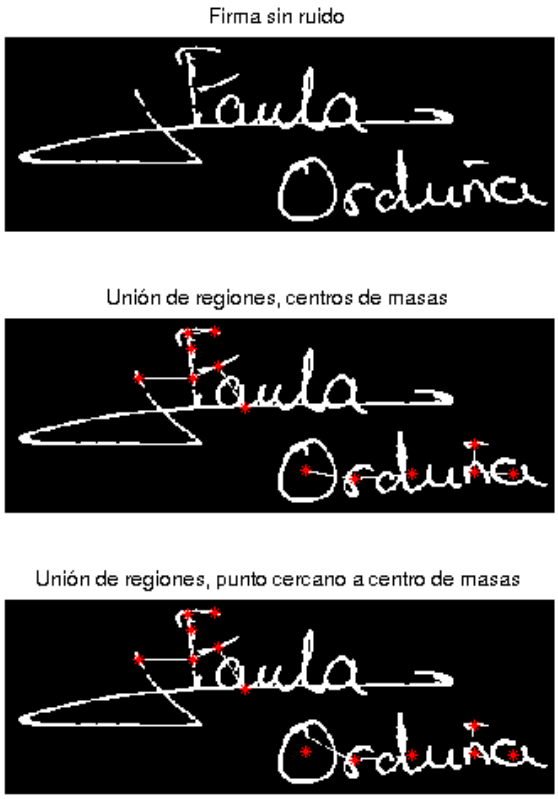

Figura 2.4: Unión de regiones por centros de masas

La primera imagen se corresponde con la imagen del apartado anterior tras la eliminación del ruido. En las imagen de la fila 2 se unen los centroides más próximos. Se hace patente el problema que conlleva, en ocasiones estos no coinciden con ningún punto de la región del trazo, pudiendo resultar en una línea “al aire” que no cumple la función de unir regiones. En la última imagen, para subsanar este último problema, se busca el punto más próximo al centroide que sí pertenezca al objeto, y se realizan las uniones.

Lamentablemente algunas uniones no cumplen con el principio propuesto de mínimos cambios en la firma original. El procedimiento es descartado como posible implementación final.

Un funcionamiento idóneo se basaría en la identificación de cada una de las regiones a unir y posteriormente comprobar entre las restantes cual es la candidata para una unión de longitud mínima. Es un problema de cálculo directo, que se trata de resolver en una segunda aproximación mostrada a continuación (Figura 2.5).

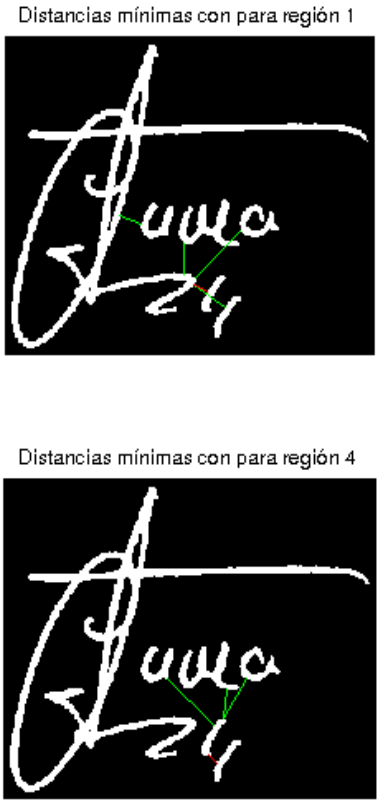

Figura 2.5: Unión de regiones por cálculo directo

Para dos regiones distintas, 1 y 3, se muestran todas las mínimas distancias con el resto de regiones (verde) eligiendo a su vez la menor de todas, para la unión final (rojo). El principal inconveniente está en que es necesario computar la distancia de cada píxel de un grupo con todos los del otro grupo, y de igual forma para el resto. Para regiones de grandes áreas puede elevar mucho el tiempo de computación. Además, el tiempo es proporcional al número de regiones y no a la distancia entre ellas. Se arrojan consumos de entre 2 y 15 segundos para fusionar una firma, un resultado que no es asumible.

Con el fin de agilizar el algoritmo se proponen otras alternativas.

La tercera aproximación, que intenta reducir el tiempo de procesado, propone utilizar un enmascaramiento sobre cada región. Sólo se buscarán regiones próximas dentro de dicha máscara, y el cálculo se realizará sólo entre la región enmascarada y los puntos encontrados. Para la máscara se recurre a la denominada Bounding Box (Borde de contenedor) que encierra cada una de las regiones. En caso de no encontrarse puntos candidatos dentro del área, se ampliarán los bordes de la misma iterativamente. La

máscara en un primer momento se eligió con la misma forma rectangular de la Bounding Box, pero en algunas situaciones podría mostrar uniones incorrectas con puntos situados en las esquinas de la máscara. En su lugar, se enmascara con una elipse contenida en el rectángulo, eliminando el problema con las esquinas.

En la ilustración 2.6 se muestran las sucesivas máscaras empleadas (las primeras 9). Incluso entre las dos primeras se puede apreciar el crecimiento para una misma región puesto que en la primera iteración no se encontró ningún punto de interés.

Figura 2.6: Proceso de enmascaramiento con elipses

El resultado se recoge en la siguiente Figura 2.7. El resultado es visiblemente mejor que la conexión realizada mediante centroides.

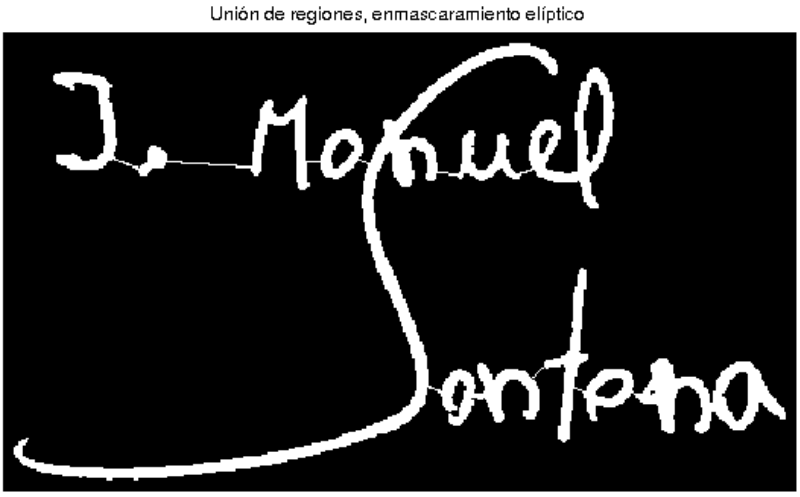

Figura 2.7: Resultado firma de unión de regiones por máscaras elípticas

La cuarta y última aproximación en vez de hacer uso de un enmascaramiento ajeno a la región, propone el uso de la misma región dilatada como máscara. Para dilatar se recurre a una estructura de disco de tamaño creciente en sucesivas iteraciones (Inicialmente se da un valor de 15 píxeles). Al igual que en el caso anterior, si no se encuentran puntos próximos dentro de la máscara, se itera dilatando nuevamente con un disco de radio dos veces mayor.

En la figura 3.1 se recogen las primeras 8 iteraciones del proceso de enmascarado por dilatación de las regiones. Se puede ver, como se hacen crecer las máscaras en caso de no encontrarse puntos próximos y la búsqueda de distancias mínimas se vuelve más exacta dentro de la región dilatada.

En la figura 3.2 se ofrece el resultado del proceso de unión de regiones por dilatación de cada una de las regiones.

Aunque esta última solución, puede arrojar resultados idóneos, requiere más tiempo de ejecución que el algoritmo anterior, llegando a duplicar o triplicar los tiempos. Fundamentalmente se debe a que el proceso de dilatación es más lento que enmascarar con una elipse.

Como conclusión se recoge a continuación una comparativa con los tiempos medios (10 iteraciones) de procesado para las 10 primeras firmas encontradas con varios objetos en las mismas. Se aplican de forma independiente los 4 algoritmos desarrollados.

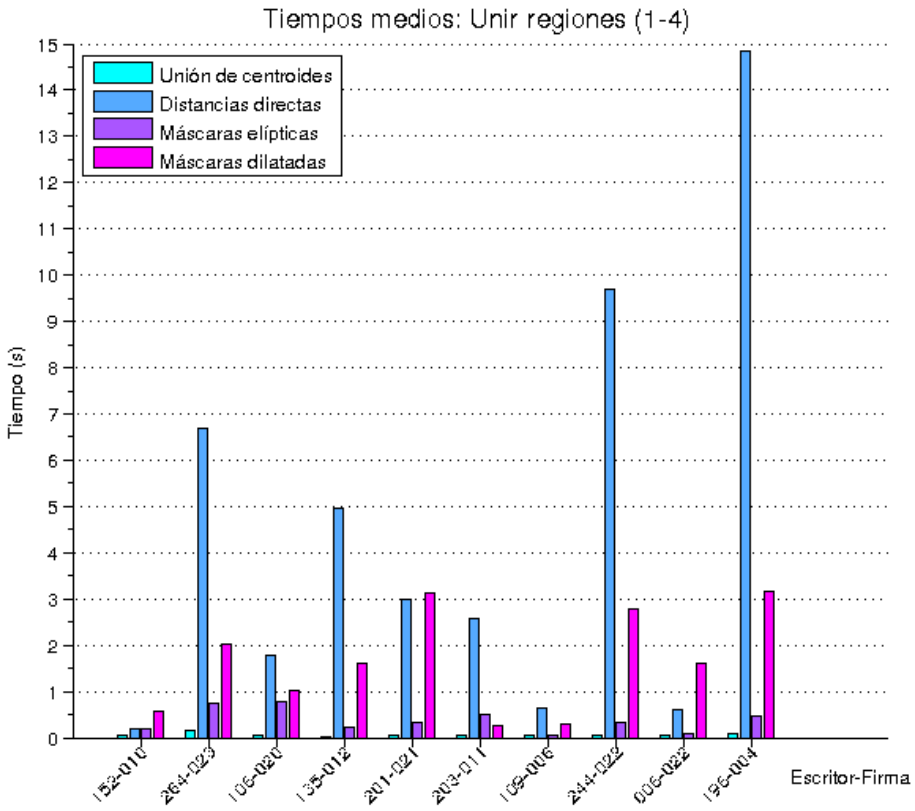

Figura 2.8: Tiempos medios, algoritmos unir regiones

Dados los tiempos se procesamiento elevados que se han obtenido tanto para la unión por cálculo directo como para el enmascaramiento por dilatación, se decide optar por el enmascaramiento mediante elipses como principal solución. La unión de regiones por el proceso de enmascaramiento con elipses es el que proporciona los tiempos más competitivos aparte de ofrecer un resultado válido.

2.3 Base de datos de manos

2.3.1 Análisis

La base de datos empleada también ha sido creada por el Grupo de Procesamiento Digital de la Señal del IdeTIC. Está constituida por las manos escaneadas de 144 individuos. Para cada persona hay un total de 10 imágenes en formato JPEG, de tamaño variable y en escala de grises con 256 niveles.

A continuación, en la ilustración 2.9 aparecen 8 ejemplos elegidos aleatoriamente entre la totalidad de muestras.

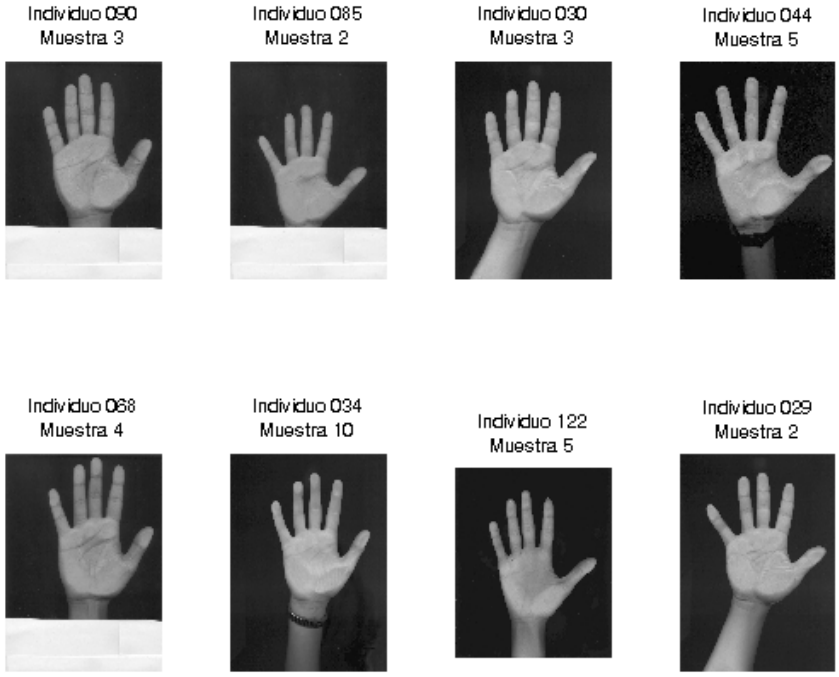

Figura 2.9: Muestras aleatorias de la base de datos

Las manos se encuentran orientadas y centradas en el lienzo. Muchas cuentan con algún papel u objeto similar que recubre parte del antebrazo. También es importante para la extracción de características tener en cuenta que las palmas escaneadas de algunos colaboradores pueden tener anillos, relojes u otros inconvenientes. Es necesario además, un proceso para eliminar la presencia de cierto ruido, debido principalmente a la presencia de polvo durante el escaneo.

2.3.2 Preprocesado

Dadas las características descritas en el apartado anterior, se hace necesario la transformación de las imágenes de la base de datos en primer lugar para obtener una imagen binaria que permita la correcta extracción del borde, también se procede a eliminar el ruido.

2.3.2.1 Binarización

Dado que sólo se busca obtener el borde de la huella de la mano, la información presente en el formato de escala de grises no es relevante, siempre que la citada forma quede patente en una representación binaria. Además, dicha representación se hace necesaria para los sucesivos procesos de extracción.

Por tanto, para el proceso de binarización es necesario elegir un umbral entre 0 y 1, que establece el nivel de nivel de gris de corte a partir del cual los valores serán convertidos a blanco, los valores menores al nivel establecido serán marcados como negro. Un valor inadecuado para dicho umbral puede dar como resultado la pérdida de la información, para el caso concreto de este proyecto, el contorno de la mano. Véase la Figura 2.10 como ejemplo.

Figura 2.10: Binarización en función del umbral

Fijando un valor del 0.2 se obtiene buenos resultados, pero con el fin de generalizar y dinamizar el proceso se recurre a la función graythresh.m. La función misma hace uso del método Otsu (Otsu's Method) que permite obtener un valor umbral que hace mínima la varianza de cada clase (blanco y negro). Aplicado a las imágenes de la base de datos, el valor devuelto por el método graythresh.m no siempre ofrece los mejores resultados, en

ocasiones dejando inteligible la forma real de la mano. Por tanto, se realiza una corrección sobre el mismo, reduciéndolo un 40%, que si da muy buenos resultados.

Los resultados conseguidos mediante esta implementación genérica son visibles en la siguiente figura, ilustración 2.11.

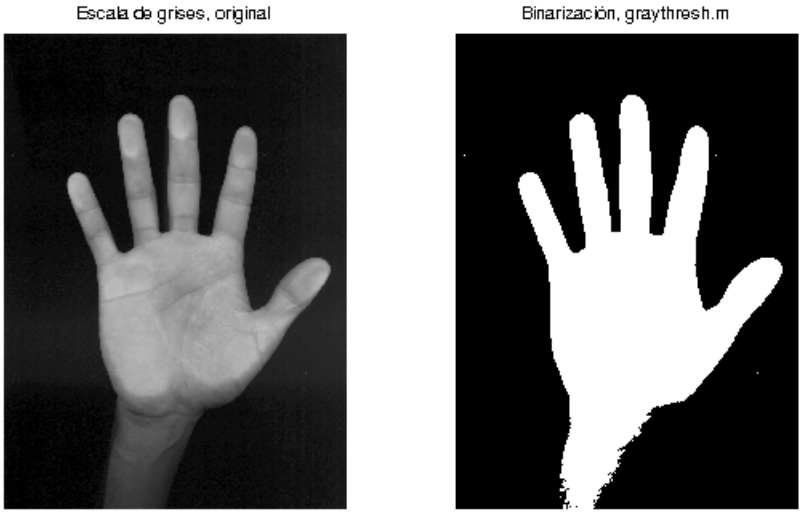

Figura 2.11: Binarización mediante graythresh.m

2.3.2.2 Eliminación de ruido y artefactos

Dadas las características de las muestras, en primer lugar se hace necesario la eliminación del ruido producido en el escaneo, así como algunos objetos flotantes de algunas muestras que parecen ser una referencia para la posición de la mano en el escáner. En segundo lugar, se procede a eliminar la parte del antebrazo que aparece en algunas imágenes, o en su defecto el papel superpuesto que evita su presencia.

Para la eliminación del ruido y de artefactos aislados, tales como la marca de posicionamiento, se procede a identificar el cuerpo de la mano. Dada la calidad de la base de datos, en todos los casos, las manos se encuentran bien orientadas y centradas, por lo que simplemente buscando en el centro de la matriz, tendremos una primera muestra de lo buscado. Para evitar posibles errores, se implementa un sencillo algoritmo, que en caso de no encontrar parte de la mano justo en el centro (se correspondería con un valor 0, blanco) buscaría en los alrededores. A continuación, en la Figura 2.12 se observa la desaparición del ruido y la marca del bolígrafo comentada.

Figura 2.12: Eliminación de ruido y artefactos

Resta la eliminación del antebrazo. Se propone también un proceso dinámico dependiente de cada entrada, basado principalmente en el estrechamiento repentino que se da en la unión de la mano con el brazo. Para buscar el punto donde se produce dicho cambio, se segmenta la mitad inferior de la imagen en 30 partes (20 píxeles aproximadamente) y ahí donde la diferencia de anchos es máxima se toma como zona de corte. Se opta por añadir la mitad de un segmento a la distancia calculada para eliminar posibles irregularidades que pueden darse en la zona de unión por arrugas en la piel o sombras. En la siguiente figura 2.13 se observa el proceso de segmentación (para 15 segmentos) y la elección de la recta de corte, y finalmente la huella de la palma de la mano limpia de cualquier incoherencia.

Figura 2.13: Eliminación del antebrazo

3 Extracción de características

La extracción de características es el siguiente paso en la preparación de los datos para un proceso de aprendizaje automático. Se busca obtener las variables más significativas de cara a que el algoritmo clasificador ofrezca el mejor resultado. Se trata de un proceso largo y que requiere de varias iteraciones hasta conseguir el resultado que mejor satisfaga las necesidades.

3.1 Introducción

Una vez los datos de entrada se encuentran normalizados en cuanto a formato y estructura de los mismos y se han eliminado posibles interferencias de las muestras que puedan alterar el proceso de extracción o clasificación se procede con la extracción de características.

Con imágenes en formato binario (blanco y negro) como las obtenidas y el objetivo de analizar el borde de las mimas como principal característica para el sistema, el siguiente paso lógico es obtener dicho borde como secuencia de puntos. A continuación se recoge el proceso de extracción del borde utilizando dos aproximaciones: obtener los puntos de manera secuencial basándonos en la conectividad de los píxeles de la imagen y obtener los puntos de intersección con un ángulo fijo respecto al origen. El resultado es una secuencia de puntos en sistema de coordenadas cartesianas.

Las secuencias de píxeles obtenidas bien podrían ser aplicadas a la entrada de un clasificador pero se proponen algunas modificaciones que pueden mejorar tanto la eficiencia del sistema, como el grado de acierto final.

Es de especial relevancia el estudio de la cantidad de información que puede ser relevante para el clasificador y en que medida reducirse sin perjudicar el comportamiento del sistema, y cuál es el proceso óptimo para ello.

3.2 Extracción puntos del borde

Se proponen varias alternativas para obtener los puntos que posteriormente representarán mediante alguna transformación las características finales.

A pesar de que los resultados visuales para la extracción basada en la conectividad son aparentemente buenos, tal y como se verá posteriormente en el capítulo de resultados, dicho proceso de obtención de la secuencia del borde no ofrece el mejor comportamiento tanto para las firmas como los contornos de manos, aunque por razones diferentes.

Para el caso de las manos, parece que las diferencias en las métricas obtenidas son muy pequeñas respecto a la gran cantidad de datos. Puede ser entendido debido a que, por ejemplo para un borde constituido por aproximadamente 500 puntos (pudiendo ser más o menos, pero un número elevado), los mismos varían para el tamaño de la imagen dado en ordenes de magnitud de cientos de píxeles a lo largo de todo el contorno, pero la diferencia entre muestras para distintos individuos son sólo de unas decenas de píxeles. Será necesario buscar alguna alternativa que magnifique dicha diferencia.

Para el caso de las firmas, dada la aletoriedad de las mismas, existen casos en los que el borde se extiende por secciones convexas y cóncavas de distinta longitud. A través del proceso de obtención del borde mediante la conectividad de píxeles, se obtienen tramos más largos o más cortos para las mismas partes de la firma entre distintas iteraciones, lo que se traduce en un desplazamiento de la envolvente acumulativo. Así por ejemplo una protuberancia al inicio en sólo una muestra de las firmas, provocará que todo el contornos de vea desplazado en base a la longitud de la misma, lo que directamente limitará el proceso de identifcación.

Dadas estas limitaciones se propone, principalmente para el caso de la firma, un sistema basado en la obtención de puntos del borde en base a incrementos de ángulos constantes. Ello evitará que cambios significativos locales afecten al resto de los puntos del borde.

En primer lugar se propone extraer los puntos del borde basados en la conectividad de los píxeles del mismo.

Una segunda aproximación propone la extracción de forma constante a través de la intersección del borde con rectas infinitas y ángulo incremental variable.

Por último se plantea para el caso particular de la mano obtener medidas de los anchos de los dedos dados los buenos resultados planteados en [10][11].

3.2.1 Recorrido del borde por conectividad de píxeles

El concepto de conectividad en los píxeles de una imagen binaria se recoge en la siguiente figura 3.1.

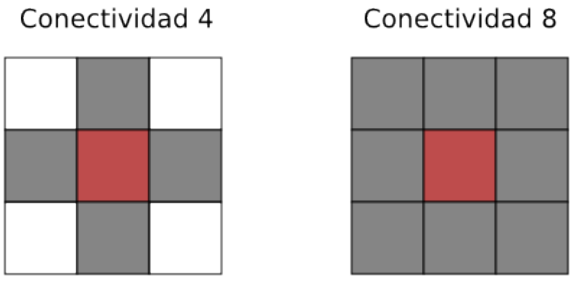

Figura 3.1: Estructura de conectividad 4 u 8

Se pueden considerar conectividad 4 u 8 para identificar que dos píxeles están conectados entre sí.

Es necesaria una nueva etapa de procesamiento que permita obtener el borde como un línea de ancho de 1 píxel. En el siguiente diagrama (Figura 3.2) se recoge el proceso utilizado.

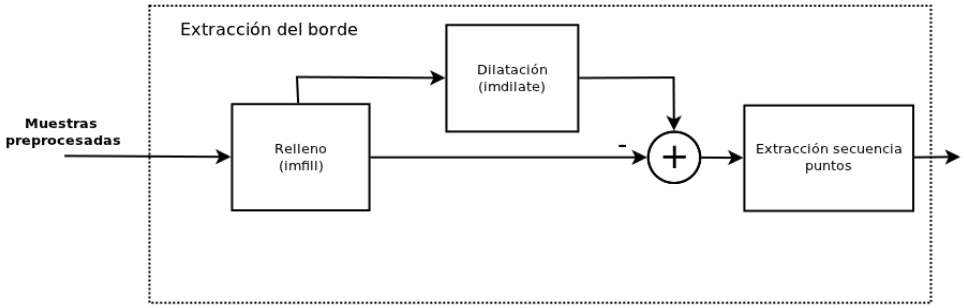

Figura 3.2: Diagrama de extracción del borde

Se recurre a la función imfill de la Image Processing Toolbox de Matlab para rellenar los huecos del objeto. Luego se aplica una dilatación mediante imdilate. Se realiza una sustracción de la imagen dilatada y sin dilatar, consiguiendo un objeto esquelético que delimita el borde del objeto en cuestión.

Por último, se procede a la secuencialización del borde. Se presentan 2 opciones: una implementación propia y el uso de la función bwtraceboundary.

- Función propia (f_puntos_consecutivos). Se trata de un algoritmo que recibe una imagen del borde cerrado del objeto, y da a la salida la secuencia de puntos ordenados aplicando una conectividad 8. También toma como parámetros de entrada un punto inicial para comenzar la secuencia y se elige una dirección para el segundo punto. El borde que se pasa a la función es obtenido por el siguiente proceso (Ilustracción 1.11):
- Función bwtraceboundary. Permite obtener el borde de un objeto macizo. Como entrada recibe el objeto relleno mediante imfill para evitar ambigüedades en el algoritmo. También requiere, punto inicial, dirección inicial, sentido de obtención de la secuencia y conectividad. La salida, es la secuencia ordenada de puntos del borde. Para las pruebas posteriores se utilizará una conectividad 8 y sentido de recorrido en contra de las agujas del reloj.

Los resultados para firmas y manos se muestran a continuación figuras 3.3 y 3.5 respectivamente.

El desempeño de la función implementada es bastante aceptable. Se notan algunas diferencias, puesto que la función f_puntos_consecutivos, por la naturaleza del código diseñado, aunque se indique conectividad 8, siempre busca primero en la vecindad con conectividad 4. Ello se traduce en que para algunas situaciones, el camino recorrido es más largo, y el tamaño de la secuencia borde final, mayor.

El principal inconveniente de la implementación propia es el tiempo que requiere. Mientras que la función interna de la Toolbox no demora nunca más de una décima de segundo. El código diseñado suele tender a un tiempo de ejecución cercano al segundo. Pueden ser dos las razones de esta gran diferencia de casi 10 veces más tiempo: código poco optimizado para f_puntos_consecutivos y código precompilado de las funciones internas.

Dadas las conclusiones anteriores, para las sucesivos cálculos del borde de la figura se recurrirá a la función interna bwtraceboundary de Matlab.

3.2.1.1 Recorrido puntos para firmas

En la figura 3.3 se muestran unas imágenes comparativas para la extracción de puntos de la firma recorriendo el borde. Como se ha mencionado los resultados son semejantes entre sí.

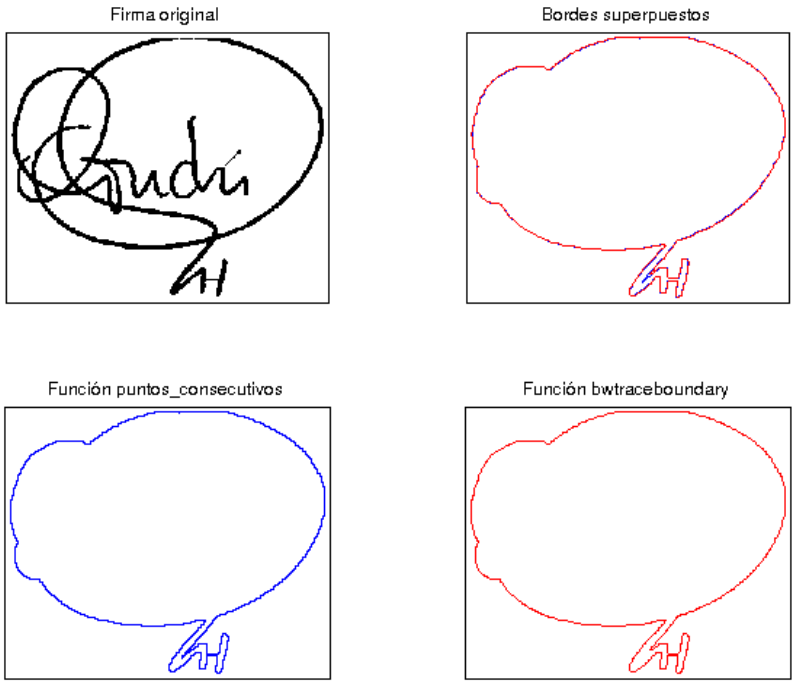

Figura 3.3: Comparación entre f_puntos_consecutivos y bwtraceboundary

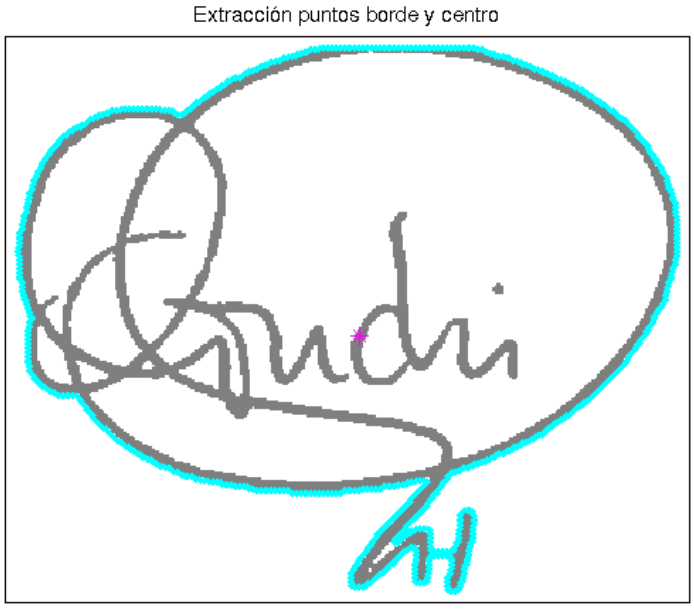

Figura 3.4: Puntos del borde extraídos ordenadamente

Por último, se muestra la representación de los puntos extraídos junto con el centroide de los mismos.

3.2.1.2 Recorrido puntos mano

En la figura 3.5 se representan unas imágenes comparativas para la extracción de puntos del contorno de la mano recorriendo el borde con la implementación propia y la función bwtraceboundary.

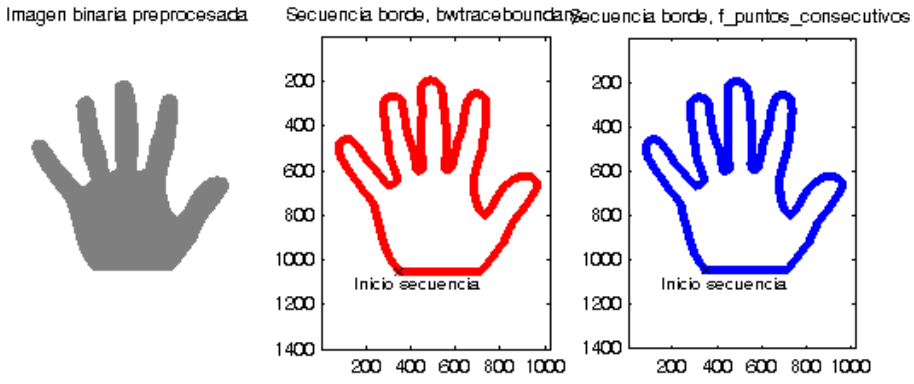

Figura 3.5: Comparación f_puntos_consecutivos y bwtraceboundary (manos)

A continuación se muestra el resultado de obtener los puntos de la mano mediante el proceso descrito.

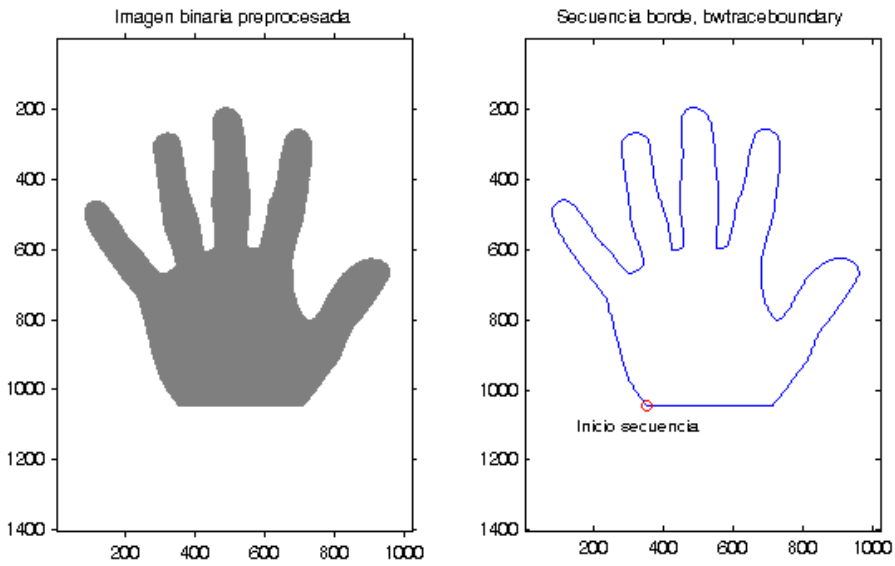

Figura 3.6: Extracción del borde de la mano

3.2.2 Extracción del borde por intersección de rectas con ángulo incremental constante

El proceso se basa en la proyección de sucesivas rectas con origen en el centroide (centro de masas) de la imagen y un ángulo incremental constante. Para cada proyección se obtienen las intersecciones o solapamientos de la recta con los puntos de la firma. De todos los puntos de intersección se selecciona el más alejado del centro, que corresponde con el contorno.

En la siguiente figura (3.7) se recogen algunos pasos del citado proceso para obtener un total de 20 puntos.

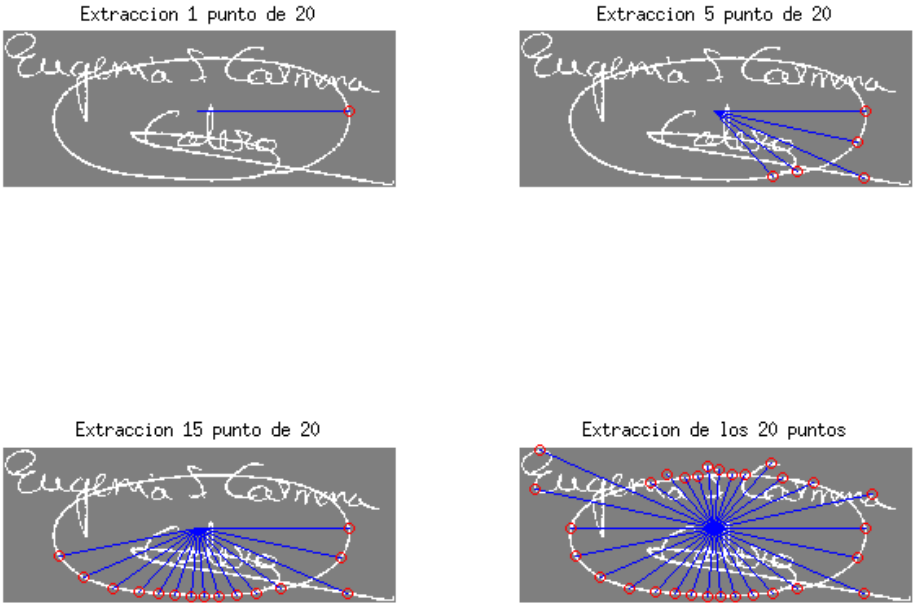

Figura 3.7: Proceso de extracción de puntos del contorno para un ángulo fijo

El resultado de de los puntos extraídos mediante esta aproximación se puede ver en la figura 3.8

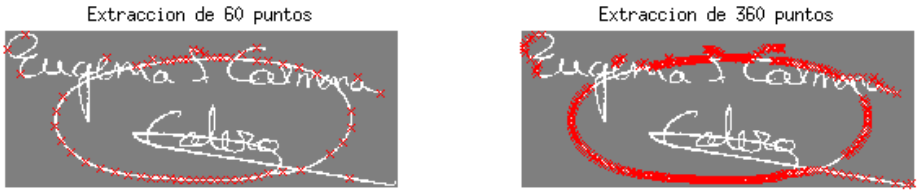

Figura 3.8: Puntos extraídos mediante el proceso de intersección con ángulo fijo para una firma

Para las muestras de la mano se realiza el mismo proceso de extracción de puntos, se puede ver el resultado en la figura 3.9, aunque se anticipa que el resultado del uso de estas características supone un peor comportamiento que en el caso de puntos fijos. En

la variable aleatoria que condiciona el método es la separación de los dedos y la posición de la mano, que entre muestra y muestra para un mismo individuo es ligeramente diferente.

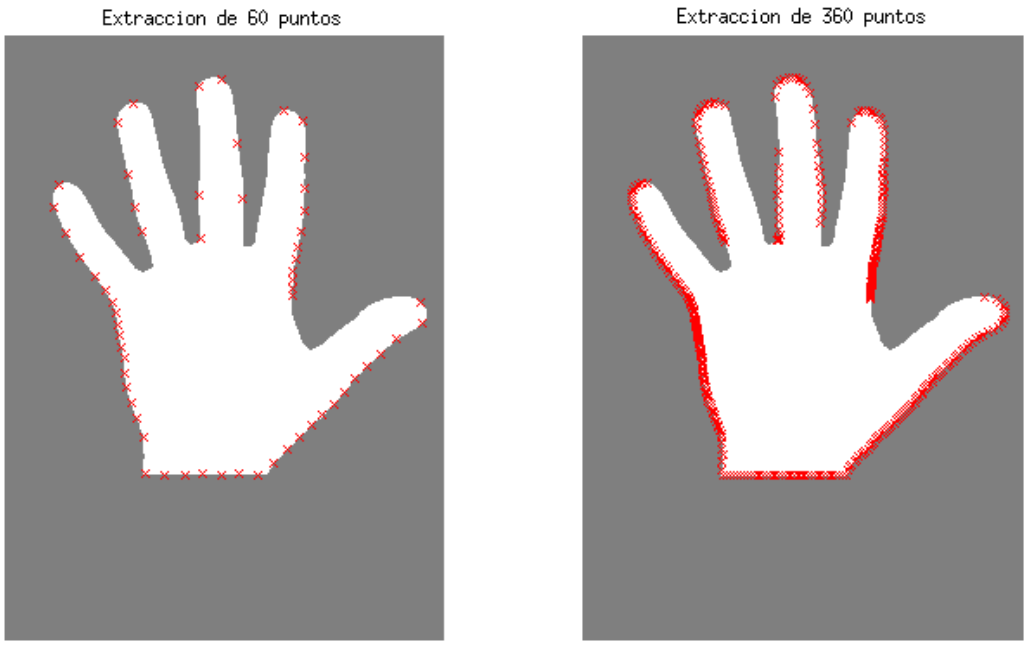

Figura 3.9: Puntos extraídos mediante el proceso de intersección con ángulo fijo para la mano

3.2.3 Extracción del largo y anchos de los dedos

Para el caso particular de la mano, ha sido necesaria la búsqueda de alternativas a obtener el grueso de todos los puntos del borde. Se ha mencionado con anterioridad que las diferencias entre unos individuos y otros en las formas de la mano es bastante pequeña, de unos pocos milímetros frente al tamaño total de la mano que puede llegar a unos 20 cm. Tomando como característica principal el borde la variabilidad entre muestras pasa prácticamente desapercibida para el clasificador.

Revisando el buen trabajo llevado a cabo en [10] reutilizamos la idea de obtener el ancho de los dedos como puntos característicos añadiendo también el largo.

Es para ello necesario obtener el borde a través del proceso descrito en el apartado 3.2.1.2 y buscar máximos y mínimos en las variaciones angulares.

Se opta por usar la variación del radio ρ obteniendo los máximos y mínimos que identifican los 5 dedos y los espacios interdigitales. En la figura 3.10 se puede ver que los dedos y el espacio entre los mismos viene muy bien diferenciado en coordenadas polares.

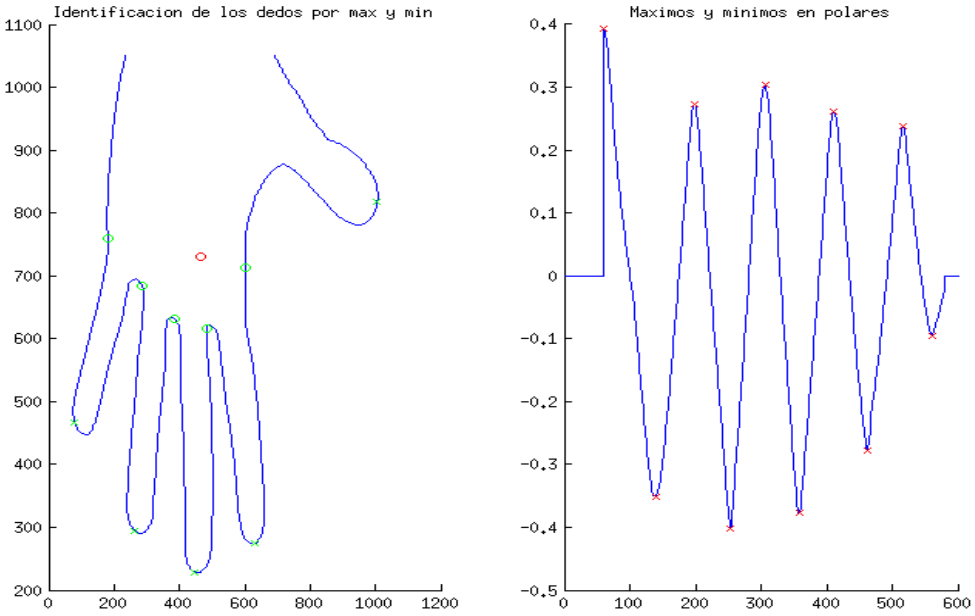

Figura 3.10: Obtención de puntas de los dedos y espacio entre dedos por máximos y mínimos en coordenadas polares

Se ha decidido descartar el dedo gordo porque el cálculo de sus dimensiones requeriría un procesamiento adicional al aplicado para el resto de dedos y aunque no sería complicado los resultados obtenidos para el resto de dedos es satisfactorio.

Sobre los puntos mínimos obtenidos a través de las coordenadas polares se realiza un ajuste únicamente para los puntos extremos de los dedos índice y meñique con el fin de aproximarlos mejor al inicio de los mismos. Se usa como referencia la prolongación de los puntos mínimos entre los dedos de los dedos corazón y anular.

En la figura 3.11 se puede observar la corrección realizada para el dedo índice a la izquierda y para el dedo meñique a la derecha. El resultado da un valor mínimo para el inicio de ambos dedos más realista. Se indica resaltado el tramo del borde donde se realiza la búsqueda de los puntos de corte, en pos de optimizar el proceso.

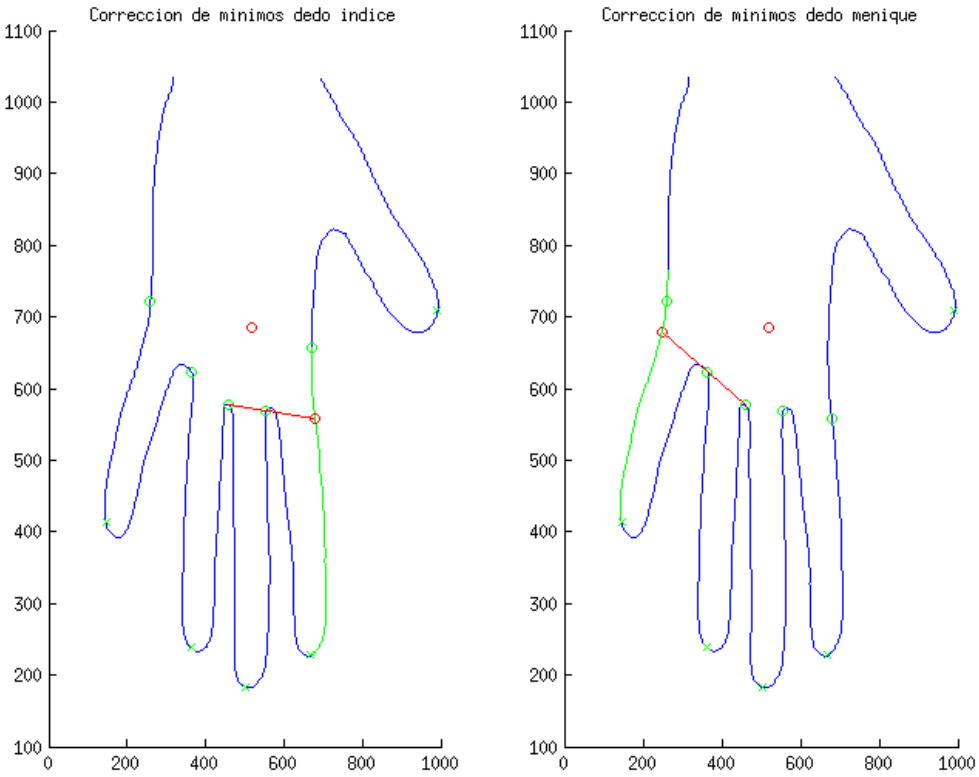

Figura 3.11: Corrección de valor mínimo para dedos índice y meñique

Por último se procede a obtener los anchos a partir de rectas perpendiculares a la línea longitudinal que recorre el dedo por el medio. Se establece a través del punto medio de los dos mínimos adyacentes y el máximo local para un dedo particular. Esta recta longitudinal es la medida tomada como el largo de cada dedo. El proceso de obtención de la recta longitudinal que divide el dedo a la mitad no es perfecta y dependiendo de las distintas fisionomías así como los grados de apertura de los distintos dedos puede variar, pero en base a los resultados obtenidos se acepta esta solución.

En la figura 3.12, en la imagen de la izquierda se tiene el largo y los 10 anchos obtenidos para el dedo índice. A la derecha se muestra el resultado de los largos y anchos obtenidos para los 4 dedos en cuestión.

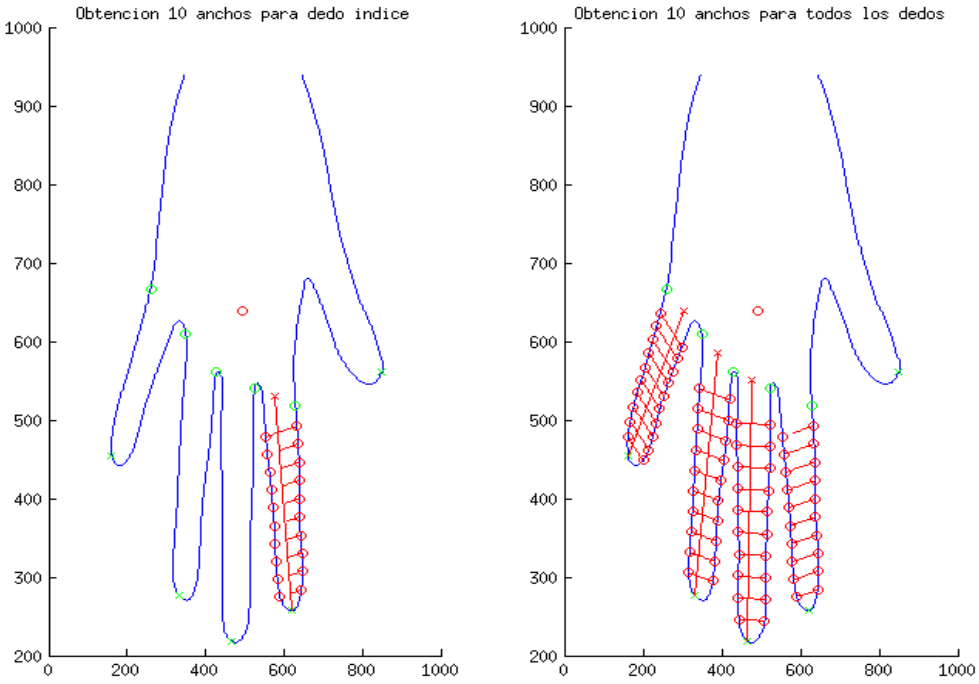

Figura 3.12: Proceso de obtención de 10 anchos por dedo

3.3 Sistemas de referencia

3.3.1 Sistema referencia Imagen, coordenadas cartesianas

La representación característica de una imagen es una particularización del sistema de coordenadas cartesianas. La gran salvedad está en que el origen se sitúa en la esquina superior izquierda, y las coordenadas [x] crecen hacia la derecha, y las [y] hacia abajo. El eje y se encuentra invertido. Además, las abscisas y ordenadas se restringen a los naturales dentro de un rango determinado, el tamaño de la imagen.

El borde procesado y obtenido en el punto anterior se sustenta en el sistema de referencia mencionado. Se trata de una matriz de tamaño (N,2), donde N determina el número de puntos totales del borde. La codificación adoptada es, para cada punto, los valores [fila,columna] que se corresponden con las coordenadas [y,x]. Además, la secuencia se encuentre ordenada, es decir, que los puntos consecutivos son adyacentes entre ellos, puesto que el borde ha sido extraído de forma continua.

A modo de ejemplo se presenta a continuación (Figura 3.13) una muestra de la base de datos de firmas, interpretada a través del sistema de referencia característico de una imagen y el sistema cartesiano natural.

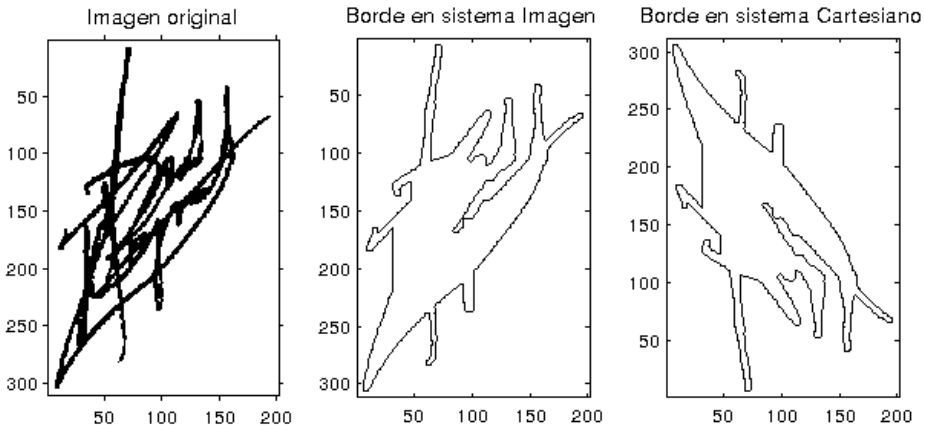

Figura 3.13: Sistemas de representación: Imagen vs. Cartesiano

Es necesario resaltar, que en los sucesivos ejemplos ilustrativos, se muestran las imágenes acorde a su representación natural. Se realiza así para que resulte más sencillo orientarse en la imagen. Por tanto, las coordenadas cartesianas se presentan con

el eje y invertido, y las coordenadas polares con el ángulo θ en sentido de las agujas del reloj, contrario al convenio establecido.

3.3.1.1 Coordenadas cartesianas centradas en origen

Una primera transformación, que desvincula el objeto tratado (firma o mano) respecto de su posición inicial en la matriz, consiste en centrar las coordenadas cartesianas respecto del centro de masas (x_c,y_c) o, más correctamente centroide. Dado que se trata de una imagen binaria, con densidad uniforme de valor 1, el centroide coincide con el centro de masas.

$$\bar{x} = x - x_c \tag{3.1}$$

$$\bar{y} = y - y_c \tag{3.2}$$

En la figura 3.14 se aprecia como las coordenadas ahora se encuentran centradas en el origen.

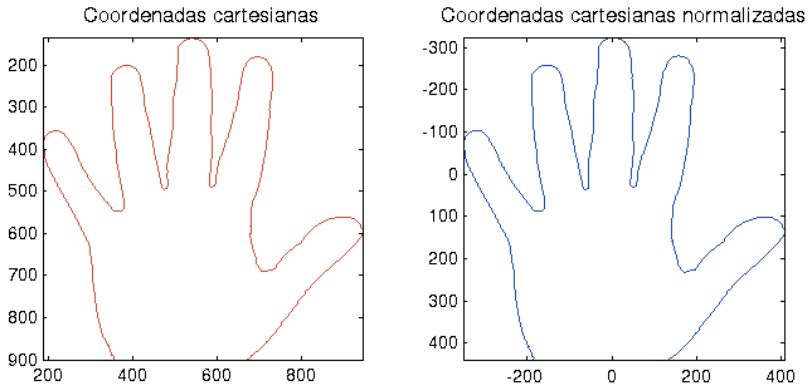

Figura 3.14: Coordenadas cartesianas y cartesianas normalizadas

3.3.2 Coordenadas polares

Sobre las coordenadas [x,y] centradas en el origen, se lleva a cabo un transformación a coordenadas polares.

La secuencia de puntos queda identificada por [θ,r], dada mediante la siguiente relación:

$$r = \sqrt{x^2 + y^2} \tag{3.3}$$

$$\theta = \arctan\left(\frac{y}{x}\right) \tag{3.4}$$

3.3.2.1 Coordenadas polares normalizadas

Una segunda abstracción, de tamaño en este caso, se consigue mediante la normalización del la distancia radial. Así dos objetos iguales pero con tamaños dispares, serán traducidos y representados de forma equivalentes una vez transformados al sistema polar normalizado.

La normalización del radio viene dada por:

$$r = \frac{r}{r_{max}} \tag{3.5}$$

En el siguiente ejemplo, a la derecha tenemos las coordenadas polares normalizadas. En este caso concreto el valor de normalización es r_{max} = 476.881 unidades.

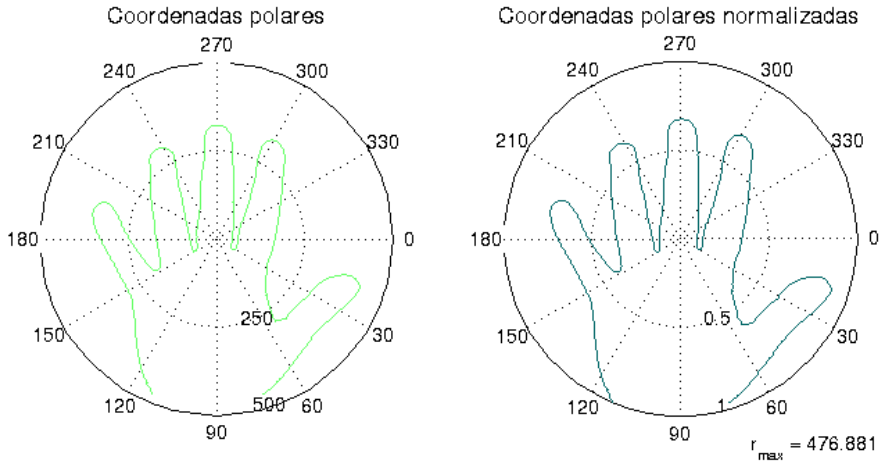

Figura 3.15: Coordenadas polares normalizadas

3.3.3 Coordenadas angulares

El sistema de coordenadas angulares está basado únicamente en medidas angulares, de acuerdo a la propuesta [9]. En la ilustración 3.16, para una secuencia de puntos cualesquiera, dado un punto P_i se proponen los ángulos candidatos $[\alpha_i,\beta_i]$ como coordenadas angulares de dicho punto.

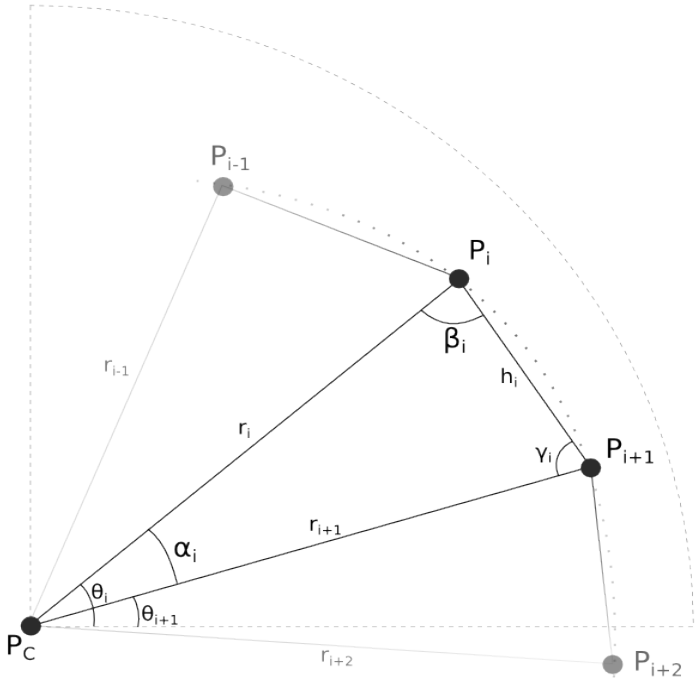

Figura 3.16: Esquema de Coordenadas Angulares

La relación para los ángulos α (P_i,P_c,P_{i+1}) y β (P_c,P_i,P_{i+1}) a partir de las coordenadas polares normalizadas se desarrollan a continuación.

Para el triángulo (P_c,P_i,P_{i+1}), el ángulo α_i se obtiene como la diferencia de la coordenada polar θ.

$$\alpha_i = \theta_i - \theta_{i+1} \tag{3.6}$$

Dos detalles importantes a tener en cuenta:

- α_i toma valores negativos o inversos en los pasos por cero que deben ser corregidos. Realizando un módulo 2*π se corrige el valor.
- α_i puede tomar valores negativos cuando se produce un cambio de sentido en la secuencia de puntos, es decir, $\theta_{i+1} > \theta_i$. El valor es

mantenido, pero para la resolución del triángulo se toman valores absolutos.

Para la obtención del ángulo β_i se procede mediante el Teorema del Seno.

Teorema del Seno

$$\frac{r_i}{\sin(\gamma_i)}=\frac{r_{i+1}}{\sin(\beta_i)}=\frac{h_i}{\sin(\alpha_i)} \tag{3.7}$$

Dónde,

$$\sin(\gamma_i)=\sin(\pi-\alpha_i-\beta_i)=\sin(\alpha_i+\beta_i) \tag{3.8}$$

desarrollando el seno de la suma de ángulos,

$$\sin(\alpha_i+\beta_i)=\cos(\alpha_i)\sin(\beta_i)+\sin(\alpha_i)\cos(\beta_i) \tag{3.9}$$

sustituyendo en el Teorema del Seno y operando.

$$\frac{r_i}{\cos(\alpha_i)\sin(\beta_i)+\sin(\alpha_i)\cos(\beta_i)}=\frac{r_{i+1}}{\sin(\beta_i)} \tag{3.10}$$

$$\frac{r_i}{r_{i+1}}=\frac{\cos(\alpha_i)\sin(\beta_i)}{\sin(\beta_i)}+\frac{\sin(\alpha_i)\cos(\beta_i)}{\sin(\beta_i)} \tag{3.11}$$

$$\frac{r_i}{r_{i+1}}=\cos(\alpha_i)+\sin(\alpha_i)\cot(\beta_i) \tag{3.12}$$

Despejando β_i

$$\beta_i=\operatorname{arccot}\left(\frac{r_i}{r_{i+1}}*\sin(\alpha_i)-\frac{\cos(\alpha_i)}{\sin(\alpha_i)}\right) \tag{3.13}$$

Dado que la representación directa de las coordenadas angulares no es posible, se ha implementado una función de transformación inversa, que realiza el cálculo las coordenadas polares nuevamente.

En el siguiente ejemplo se obtienen dos gráficas en el sistema polar para la misma muestra de mano. La primera se corresponde con las coordenadas polares obtenidas

mediante las transformación desde cartesianas normalizadas. En la segunda, se ha aplicado el cambio a coordenadas angulares y posteriormente, la transformación inversa.

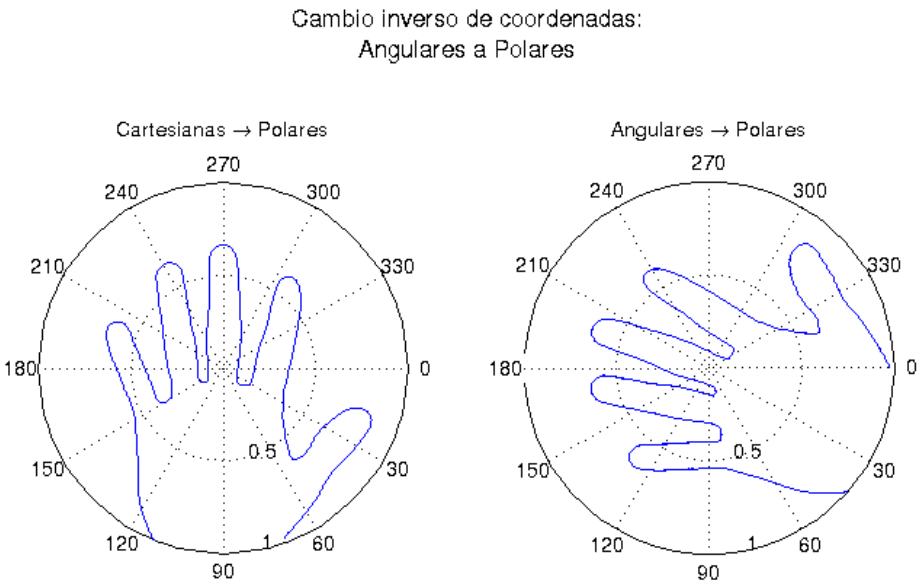

Figura 3.17: Coordenadas angulares de una muestra mano

La segunda gráfica muestra una rotación respecto a la primera, puesto que se le da al primer valor de la secuencia un ángulo θ igual a cero. Si el valor de dicho ángulo fuese conservado, las transformaciones a coordenadas angulares son sin ningún tipo de pérdidas.

Para el caso de las firmas (Figura 3.18)

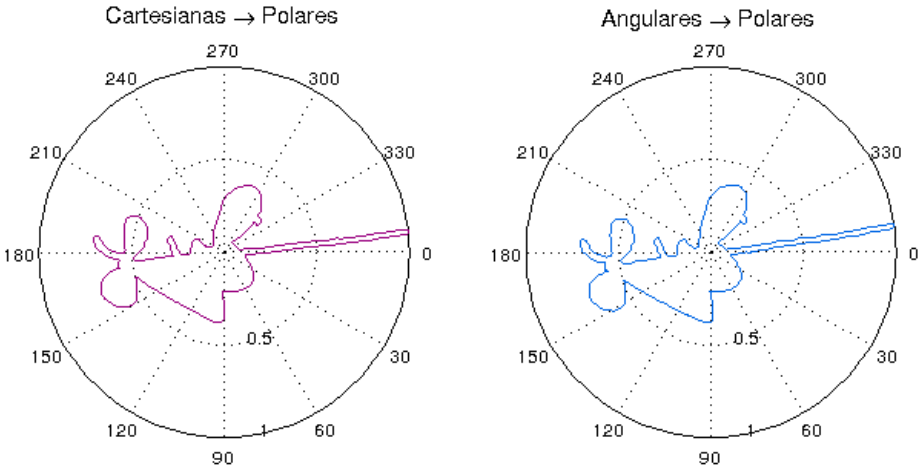

Figura 3.18: Coordenadas angulares de una muestra firma

Para este último ejemplo, la rotación sufrida, es casi inapreciable, pues el primer punto de secuencia se ha tomado casi en $\theta_1 = 0$.

3.4 Puntos del sistema

Una decisión crucial para un sistema de aprendizaje automático es el volumen de información que se lleva a la entrada del mismo. La efectividad está determinada por dicho flujo de datos, pues existe un valor óptimo que evita por un lado la eliminación de información determinante, y por otro, el procesamiento de información redundante, que poco influye.

Para el proyecto presente, dado que los vectores de características recogen la secuencia anterior de coordenadas angulares, la decisión de qué puntos son importantes y cuáles irrelevantes, es tratada a continuación con varias posibilidades: selección de puntos equidistantes, selección de puntos en esquinas...

3.4.1 Puntos equidistantes

La solución más directa al diezmado de los vectores de características pasa por una selección uniforme del conjunto de puntos.

Para una tamaño final deseado, y con la longitud del vector dada, se realiza un muestreado de la secuencia original con una paso de número de puntos dado por :

$$paso = \frac{Longitudvector}{Longitudnueva} \tag{3.14}$$

Véase el siguiente ejemplo de la base de datos de manos (Ilustración 3.19).

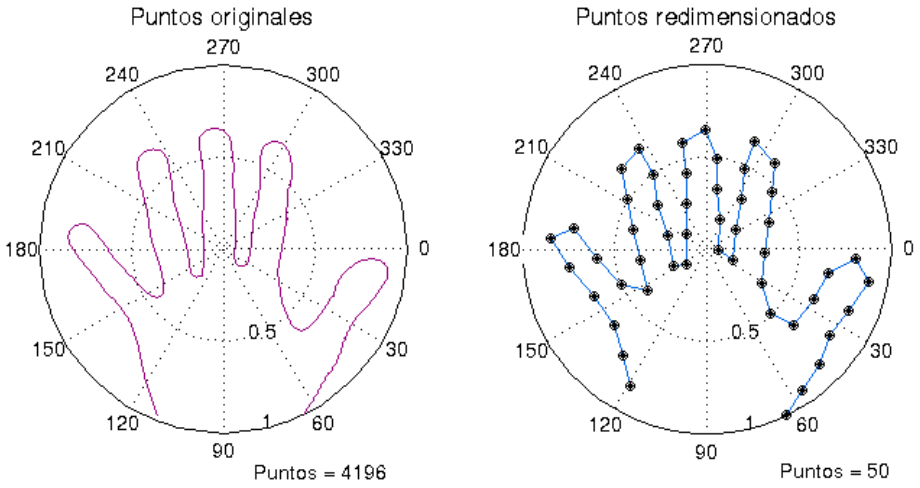

Figura 3.19: Redimensión uniforme. mano

Para un total de 4196 puntos iniciales que delimitan el borde, si se muestrea con un paso de longitud 86 unidades, se obtienen 50 puntos finales. El resultado, apreciable a simple vista, es una pérdida notable de la definición.

Para el caso de una firma (Figura 3.20).

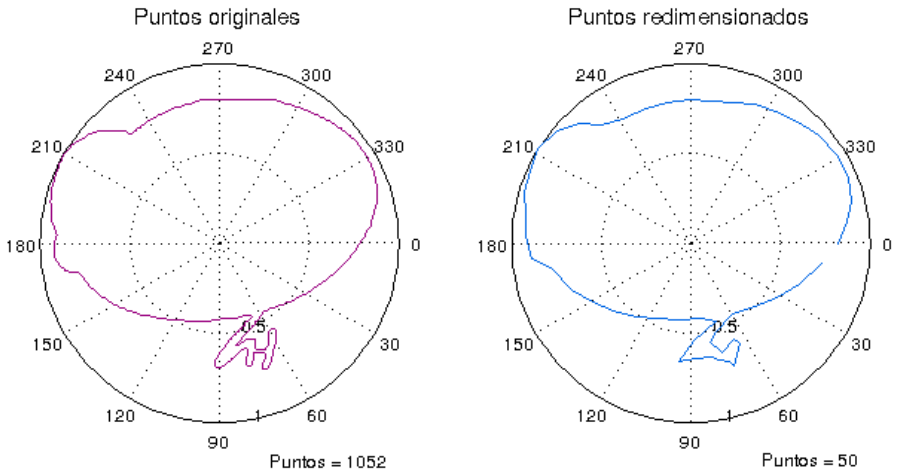

Figura 3.20: Redimensión uniforme. firma.

Para conseguir una nueva longitud de 50 unidades, es necesario, en este caso un paso de 24. También se aprecia la notable pérdida de información.

4 Clasificación

Tras el tratamiento de los datos y la extracción de características se dispone de las entradas para el sistema biométrico correctamente adaptadas.

A continuación se recoge el proceso de clasificación, centrándose en los métodos desarrollados para el presente estudio: los Modelos Ocultos de Markov y las Máquinas de Vectores Soporte, además de la combinación de los mismos mediante el Kernel de Fisher.

4.1 Clasificadores

Los algoritmos clasificadores, funciones predictoras o clasificadores en general son el bloque fundamental de los sistemas de aprendizaje automático supervisado, mediante los cuales el conjunto de datos puede ser catalogado o etiquetado a través de la clase a la que pertenecen. Es tarea de dicho algoritmo determinar para nuevas muestras la etiqueta o clase, para lo cuál es necesario un proceso previo de aprendizaje o entrenamiento. Así, la tarea fundamental consiste en inferir una función a partir de datos de entrenamiento supervisados (etiquetados), conociéndose como proceso de aprendizaje o entrenamiento. El algoritmo de entrenamiento supervisado debe por tanto analizar el conjunto de entrenamiento y proporcionar una función de predicción para nuevas entradas. Para el caso como el estudiado, dónde la salida debe ser una clase determinada, es decir, una etiqueta, la función de predicción se conoce como clasificador.

A continuación (Figura 4.1) se presenta un esquema del proceso de creación de un modelo aprendizaje automático basado en un clasificador, con las etapas de entrenamiento y su posterior evaluación.

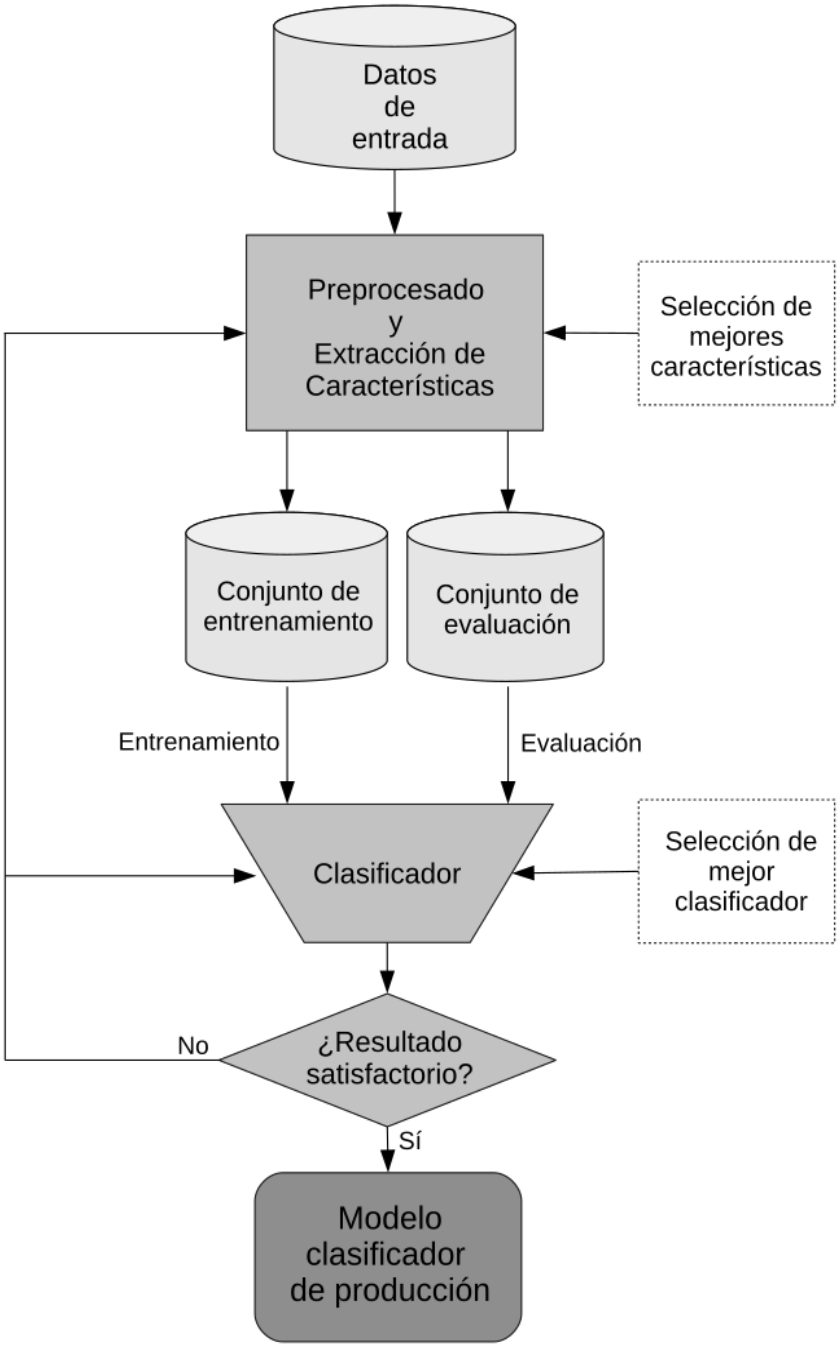

Figura 4.1: Proceso de entrenamiento de sistema supervisado

Se identifican en el esquema las etapas de análisis, preprocesado y extracción del conjunto de entrenamiento, revisadas en capítulos anteriores. El punto más relevante es la elección del modelo y por tanto del algoritmo de aprendizaje a emplear.

Son muchas las técnicas desarrolladas que desempeñan la función de predicción:

- Algoritmos de base lógica como los árboles de decisiones o el aprendizaje de conjunto de reglas.

- Técnicas basadas en el perceptrón cómo las Redes Neuronales Artificiales (En inglés Artificial Neural Networks, ANN),
- Métodos estadísticos como Redes Bayesianas (Naive Bayes) y Modelos Ocultos de Markov.
- Métodos espaciales como las Máquinas de Vectores Soporte separan muestras a través de hiperplanos.

Se pueden catalogar los algoritmos o modelos predictivos [15] en base al sistema aprendizaje al que recurren, teniendo:

- Modelos generativos: recurren a la creación de un modelo que simula la generación real de los datos. La función inferida intenta describir el flujo de generación de datos del sistema real. En este grupo se incluyen los Modelos Ocultos de Markov, al igual que Redes Bayesianas.
- Modelos discriminativos: basan su aprendizaje en los datos en sí mismo para establecer “fronteras” de decisión sobre los mismos. No hacen ninguna suposición respecto del sistema real que ha generado los datos. En este grupo tendríamos las Máquians de Vectores Soporte así como árboles de decisión y redes neuronales.

En la sucesión del presente capítulo se recoge una introducción a los sistemas de clasificación utilizados en este trabajo: los Modelos Ocultos de Markov y las Máquinas de Vectores Soportes.

También se revisa el Kernel de Fisher, una transformación de los datos que permite hacer uso de ambos clasificadores para maximizar los resultados.

4.2 Clasificador basado en Modelos Ocultos de Markov

4.2.1 Introducción

Los Modelos Ocultos de Markov, son una adaptación de los Modelos de Markov estándar (Cadenas de Markov), que buscan extender el concepto de Markov a aquellos casos en los que las observaciones no son intrínsecas al estado, sino una función probabilística del mismo.

Una Cadena o Modelo de Markov es un proceso estocástico discreto estructurado mediante un conjunto de estados que proporcionan una salida establecida (es decir, cada estado proporciona una única salida) y donde la transición entre estados viene dado por una función de probabilidad La particularidad de estos modelos concretos se halla en la denominada propiedad de Markov, que establece que la probabilidad de transición entre estados es únicamente determinada por los estados actual y siguientes. Esta propiedad puede expresarse del siguiente modo.

Para una secuencia de variables aleatorias, X_1, X_2,..., X_N, donde los diferentes valores que pueden adoptar X_i determina el conjunto de estados o espacio de estados. La probabilidad de tránsito a un estado futuro, dado un estado presente viene dada, según la propiedad de Markov por:

$$P\left[X_{t+1}=x \mid X_1=x_{1,} X_2=x_2,\ldots,X_n=x_n\right]=P\left[X_{t+1}=x \mid X_n=x_n\right] \quad (4.1)$$

Para un Modelo de Markov, de puede establecer una nomenclatura como la que sigue.

Sea el conjunto de N estados $S_1,S_2,...,S_N$ con q_t designando el estado en el instate t (t=1,2,...). La probabilidad de transición entre estados viene dada por los coeficientes a_{ij} de tal forma que:

$$a_{ij}=P\left[q_t=Si \mid q_{t-1}=S_j\right] \;\; i,j=1,2,\ldots,N \quad (4.2)$$

Además, la distribución del estado inicial viene dada por:

$$\pi_i=P\left[q_1=Si\right] \;\; i=1,2,\ldots,N \quad (4.3)$$

Un ejemplo de diagrama de Markov es mostrado en la siguiente figura (Figura 4.1). Se puede apreciar que la salida viene enteramente determinada por el estado, por lo que el estado es claramente visible.

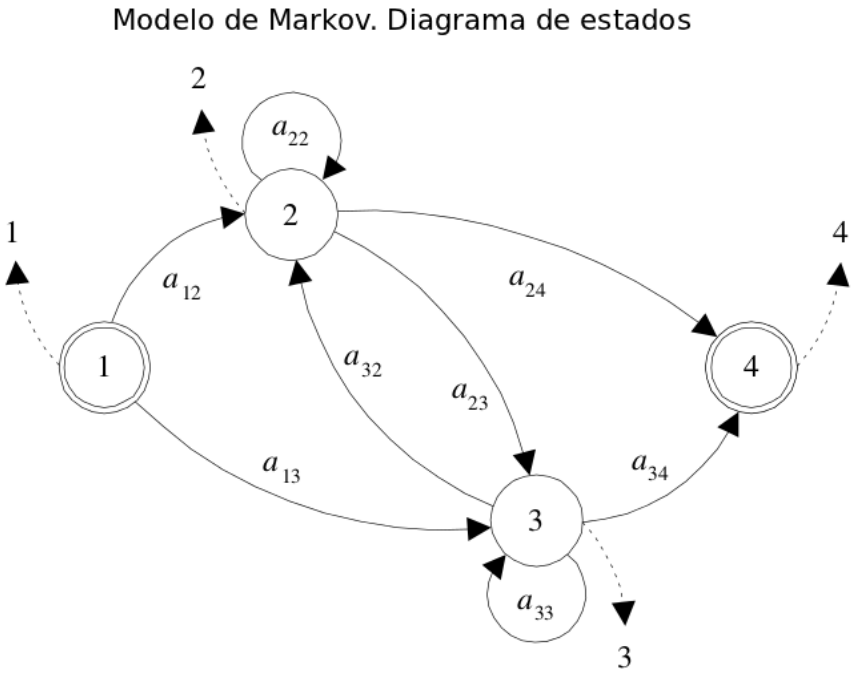

Figura 4.1: Ejemplo Modelo de Markov (MM)

Los Modelos Ocultos de Markov (MOM), al contrario que los anteriores, no ofrecen información clara y directa del conjunto de estados que se recorren. En cambio, la salida sigue una distribución de probabilidad distinta para cada estado, por lo que para una salida dada, se podrían obtener la probabilidad de las secuencias de estados que pueden proporcionar dicha salida. Además, una de las posibles secuencias poseerá probabilidad máxima, y puede ser elegida sobre las otras.

4.2.2 Descripción Modelos Ocultos de Markov

Para la descripción de los MOM, a la nomenclatura del modelo estándar hay que añadir las distribuciones de probabilidad de las salidas del sistema en función del estado presente. Para un alfabeto de salida compuesto por M símbolos, $v_1, v_2, ..., v_M$, se establece la probabilidad de símbolo para un estado j, como:

$$bi(k) = P\left[O_t = v_k \mid q_t = S_i \right] \quad i = 1,2,\ldots,N; \quad k = 1,2,\ldots,M \tag{4.4}$$

Se puede resumir por tanto que la estructura de un MOM consta:

- N estados. $S = S_1,S_2,...,S_N$
- M símbolos. $V = v_1,v_2,...,v_M$
- Distribución de probabilidad de transición de estados. $A = a_{ij}$
- Distribución de probabilidad de símbolo de estados. $B = b_j (k)$
- Distribución de probabilidad inicial de estados. $\pi = \pi_i$

Y el modelo queda completamente caracterizado por los parámetros anteriores: MOM (S, V, A, B, π), siendo común la siguiente denotación: λ (A, B, π).

A continuación, en la figura 4.2 se muestra un ejemplo de un diagrama de Modelos Oculto de Markov. Cuenta con 5 estados y dos posibles salidas para cualquier estado, '0' o '1'.

Cabe destacar que la distribución de probabilidad de transición puede determinar 2 modelos distintos: modelo ergódico donde es posible realizar una transición a cualquier otro estádo o un modelo izquierda-derecha donde se establece una limitación en la transición de estados ya sea a estados adyacentes o por niveles de adyacencia. Esta segunda definición de los modelos permite simplificar el encontrar un MOM que mejor describa un conjunto muestral.

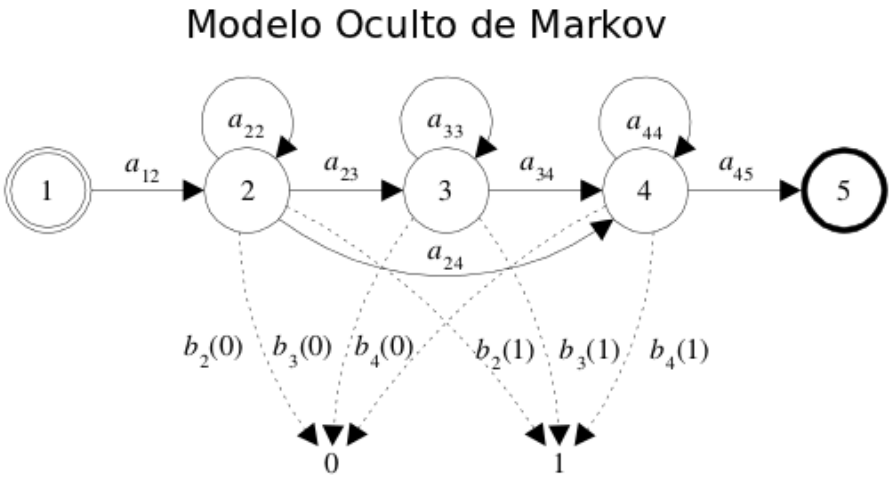

Figura 4.2: Ejemplo Modelo Oculto de Markov (MOM)

La inclusión de los Modelos Ocultos de Markov, en un sistema biométrico, radica en la capacidad que presentan dichos sistemas para adoptar un proceso de aprendizaje que posteriormente permite predecir entre las distintas clases.

El aprendizaje o entrenamiento de un MOM se define como: para una o varias observaciones, el proceso de búsqueda de la mejor caracterización λ (A, B, π) que

pueden dar dichas salidas. La resolución del problema pasa por conseguir estimar los parámetros del modelo que ofrecen un máximo de la función de verosimilitud para la secuencia de observación. No existe un algoritmo manejable para resolver el problema, y otros que se plantearán, pero se recurre a algoritmos buscan máximos locales de verosimilitud de una forma eficiente, por ejemplo, Baum-Welch.

La resolución de crear un sistema basado en MOMs se puede acotar a buscar solución a 3 problemas básicos [16].

1. Problema de evaluación. Para un MOM dado, λ (A, B, π), y una secuencia de observaciones, $O=O_1,O_2,...,O_T$, cuál es la probabilidad de que las observaciones hayan sido generadas por el modelo. Hallar P[O| λ].
2. Problema de decodificación. Para un MOM dado, λ (A, B, π), y una secuencia de observaciones, $O=O_1,O_2,...,O_T$, cómo elegimos la secuencia de estados $S = q_1,q_2,...,q_T$, que sean óptimos en alguna forma de caracterización.
3. Problema de entrenamiento. Mencionado con anterioridad, busca resolver el problema de optimización de un MOM, para que una observación sea máxima. Maximizar P[O| λ].

Una descripción detallada de la resolución de los problemas mencionados puede encontrarse en la sección Apéndices.

La resolución de los problemas es una aplicación directa en la identificación o reconocimiento.

4.2.3 Aplicación de los Modelos Ocultos de Markov

La aplicación de MOM en sistemas biométricos se realiza a través de la estructura 1-frente-resto (En inglés 1-versus-others) que establece la creación de una clasificador por clase. Cada clasificador es entrenado con los vectores de característica de una clase única.

Para la aplicación de los MOM como clasificador se recurre a la implementación desarrollada en el departamento Grupo de Procesado de Señales de la ULPGC [17].

La librería ofrece una implementación para Matlab de MOM discretos con las siguientes características:

- MOM tipo Bakis (izquierda-derecha).
- Algoritmo cuantificador LBG (Linde, Buzo & Gray).

- Algoritmo Baum-Welch para estimar λ (A, B, π).
- Algoritmo adelante-atrás para el proceso de entrenamiento.
- Algoritmo Viterbi para obtener la secuencia de estados más probable.

4.3 Clasificador basado en Máquina de Vectores Soporte

4.3.1 Introducción

Las Máquinas de Vectores Soportes, MVS (En inglés Support Vector Machine, SVM), son modelos de aprendizaje basados en la idea de un margen que divide los puntos correspondientes a dos clases distintas, concretamente, un hiperplano que separa a cada lado las posibles clases y al mismo tiempo maximiza la distancia entre dicho plano y la totalidad de puntos, conocidos como vectores soporte.

Por tanto se trata de un modelo en primera instancia para dos únicas clases, aunque fácilmente extensible para problemas multiclase a través del uso de múltiples clasificadores.

4.3.2 Definición

El caso más sencillo de una Máquina de Vectores Soporte es en el que los datos de entrada son separables linealmente, son los modelos lineales de vectores soporte:

Para un conjunto de vectores

$\{(x_1 , y_1),..., (x_n , y_n)\}$ donde $x_i \in IR_d$ e $y_i \in \{-1, 1\}$ para $i = 1, \cdots, n$

Se dice separable linealmente si existe algún hiperplano en IR_d que separa los vectores $X = \{x_1 ,..., x_n \}$ con etiqueta $y_i = 1$ de aquellos con etiqueta $y_i = -1$.

En la siguiente figura se muestra una serie de vectores correspondientes a dos etiquetas, que son separables linealmente.

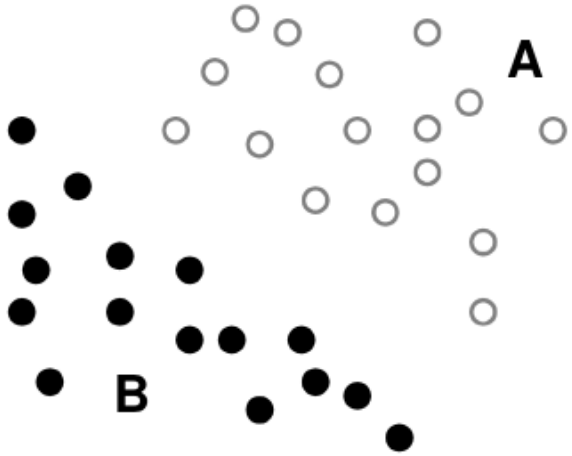

Figura 4.3: Ejemplo conjunto de vectores separables linealmente

Existe por tanto, un hiperplano descrito por,

$$\pi: \ w \cdot x + b = 0 \ , \tag{4.5}$$

La MVS busca entre todos los hiperplanos posibles aquel que maximice la distancia entre los dos conjuntos establecidos. Para un hiperplano dado (w,b)

$$x_i \cdot w + b \geq +1 \quad \forall y_i = +1 \tag{4.6}$$

$$x_i \cdot w + b \leq -1 \quad \forall y_i = -1 \tag{4.7}$$

Se puede reescribir como,

$$y_i \left(x_i \cdot w + b \right) - 1 \geq 0 \quad \forall i \tag{4.8}$$

Los vectores etiquetados como +1 que cumple la igualdad anterior, pertenecen al hiperplano:

$$\pi_1: \ x_i \cdot w + b = 1 \ , \tag{4.9}$$

Con vector normal w y distancia perpendicular al origen de coordenadas |1-b| / ||w|| siendo ||w|| la norma euclidea de w. De igual manera los vectores etiquetados como -1 que cumplen la igualdad pertenecen al hiperplano:

$$\pi_2: \ x_i \cdot w + b = -1 \tag{4.10}$$

También con vector normal w y distancia perpendicular al origen |-1-b| / ||w||. De este modo los hiperplanos π_1 y π_2 son paralelos y la separación entre ambos es: 2 / ||w|| y no hay vector del conjunto de entrenamiento limitado por ellos.

Es natural elegir los hiperplanos que mayor separación presentan y en consecuencia minimizar ||w||:

$$\min_{w \in \mathbb{R}^d} \frac{1}{2}\|w\|^2 \tag{4.11}$$

Un ejemplo de solución al problema de distancia máxima entre hiperplanos en dimensión IR^2, es el siguiente (Figura 4.4):

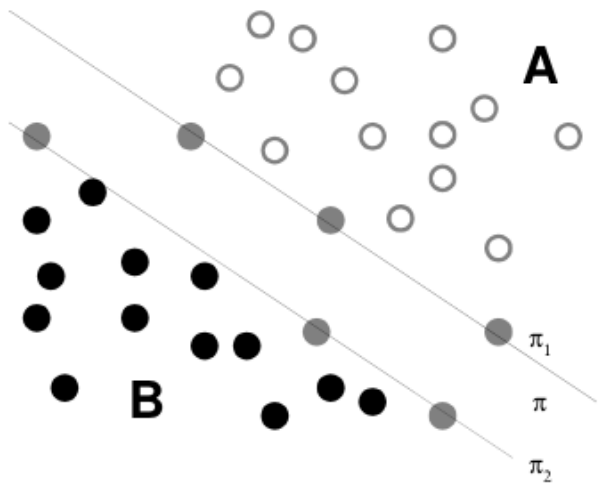

Figura 4.4: Ejemplo solución MVS lineal con hiperplanos π_1 y π_2 con distancia máxima

Para la resolución del problema se recurre a los multiplicadores de Lagrange. Un análisis analítico del proceso puede hallarse en el Anexo B de este documento, así como el estudio de los casos en el que los datos no son linealmente separables [18].

Para el problema descrito sólo se han considerado 2 posibles clases: y_1 e y_2, pero la puede ser extensible a los casos en que existe más de 2 etiquetas. Sea por tanto el conjunto de etiquetas posibles $\{\theta_1,..., \theta_l\}$ con $l > 2$ y sin una relación de orden que las defina. Sea Z el conjunto de entrenamiento:

$$Z_k = \left\{ \left(x_i, y_i \right) \ / \ y_i = \theta_k \right\} \quad (4.12)$$

4.3.3 Aplicación de las Máquinas de Vectores Soporte

Como se ha mencionado la estructura inherente de las MVS es de función biclasificadora, en la que un único modelo de MVS es capaz de discernir entre 2 clases. Existen modelos que intentan extender el concepto a máquinas multiclasificadoras [22], construyendo una función global que considera todas las clases a la vez.

Las MVS biclasificadoras recurren a transformar el conjunto de entrenamiento en un conjunto de L biparticiones, en las que se desarrolla la función de clasificación de una SVM. Se obtienen $f_1,...,_f$L clasificadores dicotómicos (biclasificadores). Posteriormente se fusionan todas las funciones, para dar como salida final una de las L posibilidades.

- Máquinas 1-v-r (denominadas así por sus iniciales en inglés one-versus-rest, una contra el resto). Son SVM dónde cada función clasificadora parcial fi, enfrenta a los vectores de una clase frente al resto de vectores de las restantes clases.
- Máquinas 1-v-1 (iniciales de one-versus-one, uno contra uno). Máquinas de vectores soporte donde cada función clasificadora parcial fij, enfrenta los vectores de la clase θi contra los de la clase θj, sin considerar el resto.

En este estudio se recurre a las implementación SVM-light gratuito para usos científicados y académicos [23].

4.4 Transformación mediante Kernel de Fisher

4.4.1 Introducción

Las ventajas de los modelos generativos como el caso de los HMM son patentes dadas las cualidades probabilísticas de las muestras de entradas además, se tiene las particularidades aportadas por los HMM de poder sustraer cierta información oculta. Los métodos discriminativos también presentan ventajas como clasificador, pues establecen claros límites de definición y suelen presentar buenos resultados para entradas bien definidas. Una buena aproximación sería un clasificador que aunara ambos métodos, y ello es posible gracias al Kernel de Fisher [20][21], que extrae información de los procesos de generación en los Modelos Ocultos de Markov, y los aplica a un discriminante, siendo el caso de las Máquinas de Vectores Soporte.

4.4.2 Descripción

La idea fundamental reside en obtener una función kernel a partir del modelo generativo dado. Es de interés analizar, no ya la relación entre una entrada y su posterior estimación, sino la diferencia entre un par de entradas. Para capturar el proceso generativo en una métrica de relación entre muestras se recurre al espacio del gradiente del modelo. El gradiente de la verosimilitud respecto a un parámetro, describe como el parámetro contribuye a la generación del mismo.

Considerando un modelo generativo, de tipo paramétrico $P(X|\theta)$, $\theta \in \Theta$, que describe una variedad de Riemann M_Θ descrita por una métrica local dada por la información de Fisher:

$$I=E_X\left[U_X U_X^T\right] \tag{4.13}$$

Y la denominada puntuación de Fisher (En inglés, Fisher score):

$$U_X=\nabla_\theta \log P\left(X \mid \theta\right) \tag{4.14}$$

Ésta última proyecta una muestra de entrada en un vector contenido en el espacio de de M_Θ, denominándose mapeo o proyección de la puntuación de Fisher. El gradiente, permite obtener la dirección de máxima variación en la función logarítmica de verosimilitud para la muestra X sobre la variedad de Riemann.

El gradiente natural es obtenido del gradiente mencionado como:

$$\Phi_X = I^{-1} U_X \tag{4.15}$$

Se denomina la proyección X → ΦX como proyección natural de las muestras en vectores de características. El Kernel natural queda definido como el producto interno de estos vectores de características relativo a la métrica local de Riemann.

$$K\left(X_i, X_j\right) = \Phi_{X_i}^T I \Phi_{X_j} = U_{X_i}^T I^{-1} U_{X_j} \tag{4.16}$$

Es el llamado Kernel de Fisher.

4.4.3 Aplicación del Kernel de Fisher

El Kernel de Fisher es aplicado sobre el modelo de Markov entrenado o estimado, concretamente sobre los valores de B (probabilidad de emisión de símbolos).

Se obtiene un nuevo conjunto de vectores que se utilizan como entrada para el clasificador de MVS.

5 Obtención de resultados

En este capítulo se recogen los resultados de la fase de evaluación de un proceso de "Machine Learning" como el presente identificador biométrico. A través de dicha etapa es posible obtener una valoración del funcionamiento del todo el sistema y determinar por tanto si es una solución viable o es necesario volver a analizar los pasos anteriores.

5.1 Introducción

Como se ha mencionado con anterioridad el sistema biométrico se cataloga como un sistema supervisado discreto cuya salida es el etiquetado dentro del conjunto de individuos. Una vez realizado el entrenamiento descrito en el capítulo anterior es necesario valorar los resultados, si son favorables el sistema estará listo para su despliegue en producción, de lo contrario es necesario volver a revisar las etapas anteriores y buscar soluciones y alternativas.

Se recurre a usar un subconjunto de las muestras disponibles apartadas del conjunto de entrenamiento, que permitirán verificar si las predicciones realizadas son correctas puesto que para cada muestra se conoce la clase a la que pertenece.

El porcentaje de datos que se destinan para el test del modelo depende principalmente de la cantidad de muestras disponibles. Revisando el conjunto de datos para las firmas y manos, contamos con 24 y 10 muestras respectivamente para cada individuo. Para las pruebas conjuntas será necesario agruparas en pares por lo que se limitará a 10 muestras totales por individuo. Para estos valores se ha considerado el 50% para cada conjunto, la mitad de muestras para entrenamiento y la otra para test. La separación de los datos se realiza de forma aleatoria para cada iteración.

Para poder medir la viabilidad de un modelo es necesario establecer algún criterio para el error. Para los sistemas discretos es bastante común la medida del porcentaje de error de clasificación o su contraparte el porcentaje de acierto.

Se define sencillamente como:

$$Error\,de\,clasificacion = \frac{\Sigma\, clasificacion\, no\, acertada}{\Sigma\, muestras\, test} \tag{5.1}$$

Y el acierto

$$Porcentaje\,de\,acierto = (1 - error\,de\,clasificación)\, x\, 100\,\% \tag{5.2}$$

Por último, para evitar posibles valores irreales marcados por una alta desviación del sistema para un conjunto de test particular, se realizan repeticiones redefiniendo aleatoriamente el conjunto de entrenamiento y test en cada iteración. El número de iteraciones propuestos es 3 para la pruebas iniciales y para las pruebas finales se realizarán 10 repeticiones.

Por tanto el resultado para cada test mostrado en este capítulo vendrá dado por la media de acierto y su desviación típica.

Es también de especial utilidad el uso de la denominada Matriz de Confusión que permite identificar las clases concretas dónde se producen más errores y llevar a cabo acciones en las etapas anteriores (dentro del proceso iterativo del aprendizaje automático) para remediarlo.

La estructura de la Matriz de Confusión es la mostrada en la siguiente tabla para una población de sólo 2 clases.

		Predicción	
		Clase 1	Clase 2
Clase real	Clase 1	Clase 1 estimada correctamente	Clase 1 estimada como clase 2
	Clase 2	Clase 2 estimada como clase 1	Clase 2 estimada correctamente

Tabla 5.1: Estructura matriz de confusión para 2 clases

Se muestran de forma enfrentada las clases verdaderas en las filas de la matriz y las clases que el clasificador da a la salida como predicción en las columnas. Si el clasificador fuera perfecto y acertara siempre, la matriz de confusión sería una matriz diagonal. Es muy útil para identificar casos que se alejan del comportamiento medio o estándar, por ejemplo, un conjunto de muestras que por el proceso de preprocesado o extracción de características han quedado corruptas y producen un resultado casi aleatorio en la clasificación.

En la figura 5.1 se expone un ejemplo de Matriz de confusión para un caso real de este proyecto en el que se ha conseguido un acierto de alrededor del 80%. Se pueden apreciar algunos puntos algo más brillantes fuera de la diagonal de la imagen indicando que varias muestras han producido el mismo error de clasificación.

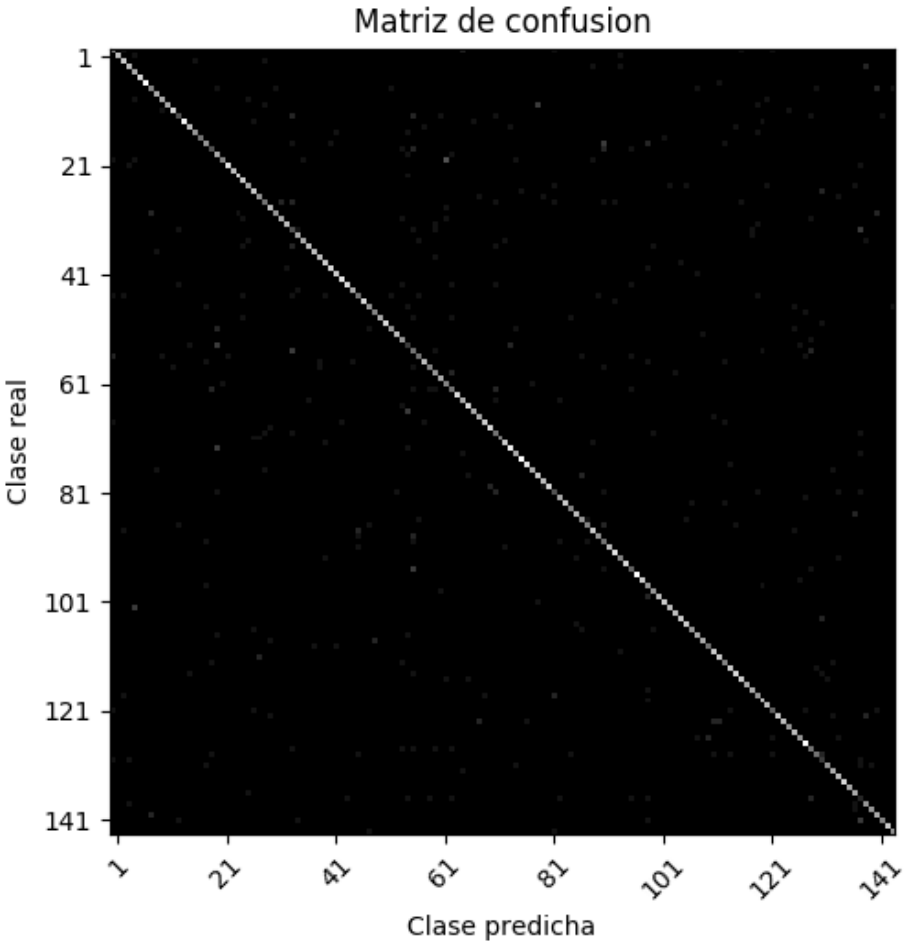

Figura 5.1: Ejemplo de Matriz de Confusión para 144 clases

En los siguientes aparatados se recogen sucesivas iteraciones de evaluación para las distintas características comentadas en el Capítulo 3. En primer lugar se realizan pruebas de los sistemas por separado para firmas y manos y posteriormente se muestran resultados sel sistema combinado.

5.2 Resultados sistemas independientes

Se resumen a continuación los resultados obtenidos para los sistemas biométricos independientes para manos y firmas, usando los diferentes métodos propuestos para la extracción de características y probando diferentes clasificadores.

5.2.1 Resultados base de datos manos usando Modelos Ocultos de Markov

Se analiza en los próximos experimentos la eficiencia del sistema biométrico sólo para la base de datos de manos y haciendo uso MOM. Se realiza para el experimento un barrido de estados sobre el MOM en la búsqueda del óptimo.

5.2.1.1 Selección de puntos mediante recorrido del borde.

Se recogen a continuación los resultados para el clasificador MOM haciendo uso de los puntos del borde obtenidos a través del proceso de conectividad de píxeles y realizando transformaciones a polares y angulares.

Coordenadas angulares

Estados / Puntos	40	50	60
60	26.04 % ± 1.68	28.55 % ± 2.10	28.72 % ± 0.92
80	30.60 % ± 2.78	32.11 % ± 3.31	31.88 % ± 1.15
100	33.44 % ± 2.80	36.23 % ± 3.87	32.03 % ± 1.17
200	32.90 % ± 1.04	35.50 % ± 2.03	34.74 % ± 2.92

Tabla 5.2: Resultados manos selección mediante conectividad, angulares (MOM)

Coordenadas polares

Estados / Puntos	40	50	60
60	20.22 % ± 0.61	22.82 % ± 0.85	21.35 % ± 2.21
80	23.26 % ± 1.71	17.53 % ± 0.49	16.23 % ± 0.85
100	21.87 % ± 0.24	18.14 % ± 1.10	13.19 % ± 2.21
200	16.58 % ± 0.61	18.23 % ± 3.43	19.18 % ± 0.12

Tabla 5.3: Resultados manos selección mediante conectividad, polares (MOM)

Los resultados indican que el recorrido del borde de forma directa a través de la conexión de puntos no es una alternativa viable para la extracción de características.

5.2.1.2 Anchos y largos de los dedos de la mano

En este caso se recogen los experimentos para los vectores de características de anchos y largos de los dedos de la mano.

Estados / Anchos	30	40	50
5	93.52 % ± 3.72	95.09 % ± 0.92	94.31 % ± 3.54
10	96.34 % ± 1.76	96.67 % ± 0.55	96.20 % ± 2.44
15	96.83 % ± 2.60	**97.52 % ± 2.21**	97.31 % ± 2.96
20	96.66 % ± 1.64	97.46 % ± 1.74	97.04 % ± 3.16

Tabla 5.4: Resultados manos largos y anchos de los dedos (MOM)

El mejor valor para el sistema independiente de la mano haciendo uso de MOM es 97.52 % ± 2.21.

5.2.2 Resultados base de datos firmas usando Modelos Ocultos de Markov

De igual forma se recogen en este apartado los experimentos para la base de datos de mano junto con MOM.

5.2.2.1 Selección de puntos mediante recorrido del borde. Coordenadas polares

Estados / Puntos	40	50	60
80	65.06 % ± 1.41	65.41 % ± 0.55	64.71 % ± 2.63
100	64.50 % ± 2.20	64.34 % ± 0.92	65.36 % ± 1.35
150	61.24 % ± 0.67	64.06 % ± 1.35	63.02 % ± 1.10
200	54.64 % ± 2.51	59.37 % ± 0.24	61.49 % ± 1.88

Tabla 5.5: Resultados firmas selección mediante conectividad, polares (MOM)

Coordenadas angulares

Estados / Puntos	40	50	60
80	61.02 % ± 1.96	58.68 % ± 2.57	57.20 % ± 0.36
100	68.27 % ± 1.78	65.40 % ± 2.03	63.39 % ± 1.49
150	69.96 % ± 1.35	74.26 % ± 1.53	70.35 % ± 3.62
200	72.39 % ± 0.98	73.26 % ± 0.31	72.52 % ± 0.55

Tabla 5.6: Resultados firmas selección mediante conectividad, angulares (MOM)

5.2.2.2 Selección puntos mediante ángulo θ constante

Coordenadas polares

Estados / Puntos	35	45	55	60
60	57.00 % ± 1.18	56.56 % ± 3.38	57.54 % ± 0.98	55.27 % ± 0.30
80	59.43 % ± 3.02	58.53 % ± 1.99	59.83 % ± 1.30	59.31 % ± 0.70
100	59.58 % ± 1.48	62.70 % ± 2.56	61.63 % ± 1.02	62.12 % ± 1.18
200	53.62 % ± 0.41	61.81 % ± 1.05	63.91 % ± 2.01	64.61 % ± 0.73

Tabla 5.7: Resultados firmas selección mediante θ constante, polares (MOM)

Coordenadas angulares

Estados / Puntos	35	45	55	60
60	61.21 % ± 2.10	61.57 % ± 1.70	64.81 % ± 1.87	63.54 % ± 0.93
80	71.09 % ± 3.48	70.32 % ± 1.98	68.49 % ± 2.13	72.43 % ± 0.74
100	76.04 % ± 1.35	76.27 % ± 0.98	74.54 % ± 0.99	71.81 % ± 4.01
200	81.97 % ± 1.44	84.17 % ± 2.51	84.84 % ± 1.30	**85.82 % ± 1.01**

Tabla 5.8: Resultados firmas selección mediante θ constante, angulares (MOM)

El mejor resultado para el sistema independiente de firmas haciendo uso de MOM es 85.82 % ± 1.01.

5.2.3 Resultados base de datos manos usando Máquinas de Vectores Soporte

5.2.3.1 Selección de puntos mediante recorrido del borde.

Se recogen a continuación los resultados para el clasificador MVS haciendo uso de los puntos del borde obtenidos a través del proceso de conectividad de píxeles y realizando transformaciones a polares y angulares.

Tipo / Puntos	Coordenadas angulares	Coordenadas polares
60	72.55 % ± 0.02	5.93 % ± 0.01
80	76.90 % ± 0.02	9.77 % ± 0.01
100	76.53 % ± 0.01	13.43 % ± 0.02
200	77.59 % ± 0.02	31.02 % ± 0.03

Tabla 5.9: Resultados manos selección mediante conectividad (MVS)

Los resultados indican que el recorrido del borde de forma directa a través de la conexión de puntos podría ser válido para coordenadas angulares pero no así para las coordenadas polares que ofrecen resultados muy bajos.

5.2.3.2 Anchos y largos de los dedos de la mano

En este caso se recogen los experimentos para los vectores de características de anchos y largos de los dedos de la mano usando el clasificador MVS.

Anchos	Resultado
5	100 %
10	100 %
15	100 %
20	100 %

Tabla 5.10: Resultados manos anchos y largos (MVS)

Las MVS ofrecen un 100% para el sistema independiente de la mano.

5.2.4 Resultados base de datos firmas usando Máquinas de Vectores Soporte

5.2.4.1 Selección puntos mediante recorrido del borde

Los resultados de la siguiente tabla muestran el experimento con la secuencia de puntos del borde obtenidos mediante conectividad a través de una representación en coordenadas angulares y polares haciendo uso de Máquina de Vectores Soporte Lineales.

Tipo Puntos	Coordenadas angulares	Coordenadas polares
60	30.34 % ± 0.02	46.66 % ± 0.01
80	30.50 % ± 0.01	46.66 % ± 0.01
100	30.73 % ± 0.02	47.13 % ± 0.02
200	29.96 % ± 0.03	48.13 % ± 0.02

Tabla 5.11: Resultados firmas selección mediante conectividad (MVS)

5.2.4.2 Selección puntos mediante ángulo θ constante

A continuación se recogen los resultados para los puntos de la firma obtenidos mediante un ángulo incremental constante y representados también en coordenadas polares y angulares. Se recurre en este caso a un clasificador MVS.

Tipo Puntos	Coordenadas angulares	Coordenadas polares
60	70.90 % ± 0.01	79.73 % ± 0.02
80	74.03 % ± 0.01	80.87 % ± 0.01
100	76.41 % ± 0.02	81.44 % ± 0.01
200	78.34 % ± 0.02	**82.55 % ± 0.01**

Tabla 5.12: Resultados firmas selección mediante θ constante (MVS)

Las MVS dan un resultado máximo 82.55 % ± 0.01 para el sistema de firmas independiente haciendo uso de coordenadas polares.

5.2.5 Resultados base de datos manos usando Modelos Ocultos de Markov y Máquina de Vectores Soporte (Fisher)

Para los experimentos de clasificador combinado se han descartado los puntos secuenciales del borde de la mano por su mal comportamiento. Se recurre por tanto a las métricas de largos y anchos como característica principal.

El experimento del modelo combinado de MOM junto a MVS a través del Kernel de Fisher arroja un resultado de 100% estable para todos los distintos valores de anchos tomados (5, 10, 15 y 20).

5.2.6 Resultados base de datos firmas usando Modelos Ocultos de Markov y Máquina de Vectores Soporte (Fisher)

De igual forma que se recurre a los mejores vectores obtenidos hasta el momento para las firmas, estos son los conformados por la selección de puntos a través de un θ incremental constante.

El resultado es nuevamente del 100% para todas las muestras analizadas para los vectores de longitud 60, 80, 100 y 200.

5.3 Resultados sistema de muestras biométricas combinadas

Se recogen en este apartado los experimentos del sistema conjunto a través de la combinación de vectores de características de firmas y manos.

Se recurren a las muestras con mejores resultados de los apartados anteriores.

Para la base de datos de firmas se toman los puntos obtenidos mediante un ángulo θ constante en representación angular y una longitud de vector de 200 puntos.

Para la base de datos de manos se toman los anchos y largos de los dedos. Por cada dedo se utilizan 15 anchos.

5.3.1 Resultados usando Modelos Ocultos de Markov

A través del clasificador MOM se obtienen los siguientes resultados para las muestras biométricas fusionadas.

Estados	35	45	55	60
Manos + Firmas	89.60 % ± 1.56	91.34 % ± 0.89	91.22 % ± 1.77	92.69 % ± 2.08

Tabla 5.13: Resultados sistema biométrico combinado MOM

5.3.2 Resultados usando Máquinas de Vectores Soportes

Para el experimento con el clasificador MVS se usan las mismas muestras y se obtiene un resultado medio del 100% para las sucesivas iteraciones.

5.3.3 Resultados usando Modelos Ocultos de Markov y Máquina de Vectores Soporte (Fisher)

De igual forma, la combinación de muestras en el clasificador combinado de MOM + MVS (Fisher) también arroja un 100% de acierto en el proceso de identificación.

5.4 Resultados reduciendo el porcentaje de entrenamiento

A fin de comparar los modelos de los apartados anteriores que han mostrado resultados del 100%, se procede a reducir el porcentaje de entrenamiento y comprobar cuál presenta un mejor comportamiento.

Se revisan por tanto en este apartado el sistema biométrico de la mano junto con los modelos MVS y MOM + SVM (Fisher), así como el sistema biométrico conjunto a través del MOM + SVM (Fisher).

De forma similar al apartado anterior se selecciona un conjunto de vectores de entrada correspondiente a los mejores resultados de los apartados anteriores. Para la base de datos de firmas se toman los puntos obtenidos mediante un ángulo θ constante en representación angular y una longitud de vector de 200 puntos. Se usan para los experimentos de MOM 60 estados ocultos, aunque como se ha revisado la variabilidad vista de 35 a 60 estados es mínima. Para la base de datos de manos se toman los anchos y largos de los dedos. Por cada dedo se utilizan 15 anchos.

El porcentaje de entrenamiento es reducido del 50% utilizado en los experimentos de apartados anteriores hasta el 20%.

5.4.1 Resultados base de datos manos usando Máquina de Vectores Soporte, entrenamiento reducido

En la siguiente tabla se muestran los resultados del Sistema biométrico de manos haciendo uso de Máquinas de Vectores Soporte. Una vez se reduce el porcentaje de entrenamiento se comprueba que el rendimiento del sistema se reduce notablemente.

Entrenamiento	Resultado
40%	98.80 % ± 0.01
30%	96.23 % ± 0.01
20%	88.11 % ± 0.03

Tabla 5.14: Resultados manos usando MVS, entrenamiento reducido

5.4.2 Resultados base de datos manos Modelos Ocultos de Markov y Máquina de Vectores Soporte (Fisher), entrenamiento reducido

A continuación se recoge el experimento de reducir el set de entrenamiento también para el sistema biométrico de la mano haciendo uso Modelos Ocultos de Markov combinados con Máquinas de Vectores Soporte a través del Kernel de Fisher.

Se aprecia que de igual manera el resultado baja del 100% aunque de forma más lenta que en el caso anterior.

Entrenamiento	Resultado
40%	99.92 % ± 0.01
30%	99.54 % ± 0.11
20%	98.93 % ± 0.22

Tabla 5.15: Resultados manos usando MOM + MVS (Fisher), entrenamiento reducido

5.4.3 Resultados sistema combinado Modelos Ocultos de Markov y Máquina de Vectores Soporte (Fisher), entrenamiento reducido

Por último se muestra en este experimento el sistema de muestras conjuntas de firmas y manos, haciendo uso de MOM + MVS (Fisher).

En este caso al reducir el conjunto de entrenamiento al 40% el sistema sigue ofreciendo un rendimiento del 100% de acierto. Si se sigue reduciendo al 30 y 20 % el sistema reduce su capacidad pero en comparación con los casos anteriores ofrece unos resultados ligeramente superiores.

Entrenamiento	Resultado
40%	**100 %**
30%	99.90 % ± 0.02
20%	99.88 % ± 0.01

Tabla 5.16: Resultados conjunta usando MOM + MVS (Fisher), entrenamiento reducido

6 Demo

Se incluye en este capítulo una propuesta demo para el sistema biométrico desarrollado. Se trata de una sencilla aplicación Android conectada a un servidor donde se implementan los sistemas independientes y combinados vistos en los capítulos anteriores. A través de una interfaz es posible seleccionar muestras para identificarlas en los distintos sistemas y dar una respuesta de identificación satisfactoria o errónea.

6.1 Estructura de la demo

La demo consta de dos bloques independientes: aplicación con interfaz de usuario Android y servidor con los clasificadores configurados previamente. Se conectan a través de una pequeña API HTTP desarrollada mediante el framework Python Flask para establecer la comunicación necesaria.

6.1.1 Servidor Python con Flask

Flask es un micro-framework escrito en lenguaje Python que ofrece un buen punto de partida para pequeñas aplicaciones web o servicios ligeros.

La librería Flask incluye un servidor local de desarrollo, con ciertas limitaciones pero que para los objetivos de la demo es suficiente.

La aplicación web se basa en 3 clasificadores configurados y entrenados con el 50% de las muestras disponibles. De esta forma el servidor debe ofrecer a la aplicación cliente las muestras disponibles para el proceso de evaluación (identificación).

El desarrollo de la aplicación en servidor ha requerido crear una API que consta de dos servicios:

- Obtener individuos y muestras disponibles para cada modelo. Los clasificadores entrenados exponen las muestras de los distintos individuos que no han sido usados en el proceso de entrenamiento. A través de un parámetro se indica el modelo que se está solicitando: “manos”, “firmas” o “combinada”. El servicio devuelve una lista con los posibles individuos que contienen a su vez una lista de muestras. Las imágenes están alojadas en el servidor como recursos estáticos.
- Evaluar una muestra de un individuo usando uno de los 3 modelos definidos. Requiere de tres parámetros en al petición del cliente: modelo, individuo e iteración de la muestra. La respuesta del servicio es el valor de identificación predicho por el modelo.

6.1.2 Aplicación con interfaz de usuario Android

Se ha desarrollado una aplicación nativa con el objetivo de ser ejecutada en un sistema operativo Android. Se cubren sólo diseños para móvil siendo posible ejecutarla en dispositivos con otras resoluciones o densidades de pantalla diferentes pero sin encontrarse correctamente adaptadas a tales pantallas.

6.1.2.1 Estructura técnica de la aplicación

La aplicación está construida sobre el lenguaje de programación Java, tomando como base el SDK de Android y desarrollada haciendo uso del IDE Android Studio, herramientas que simplifican enormemente el desarrollo de una aplicación.

La interfaz de usuario consta de las siguientes pantallas: inicio, principal, menú lateral, lista de individuos por modelo, lista de iteraciones por individuo, pantalla de resultados, ajustes.

6.1.2.2 Pantalla de inicio y pantalla principal

La pantalla de inicio es una interfaz sin funcionalidad que bloquea la navegación de usuario durante la carga de la aplicación. En el presente caso incluye una animación que mejora la experiencia de usuario.

La pantalla principal es la primera pantalla que alcanza el usuario al abrir la aplicación cuando se encuentra correctamente cargada. En este caso sólo es una pantalla informativa.

En la figura 6.1 se muestran las pantallas de inicio y principal.

Figura 6.1: Pantallas de inicio y principal

6.1.2.3 Pantalla menú lateral

Pantalla que permite la navegación entre distintas secciones. Para la demo se ha dado la posibilidad de navegar a tres secciones principales, correspondiéndose con los sistemas biométricos independientes y el sistema biométrico combinado.

A continuación se presenta una captura del menú lateral de navegación.

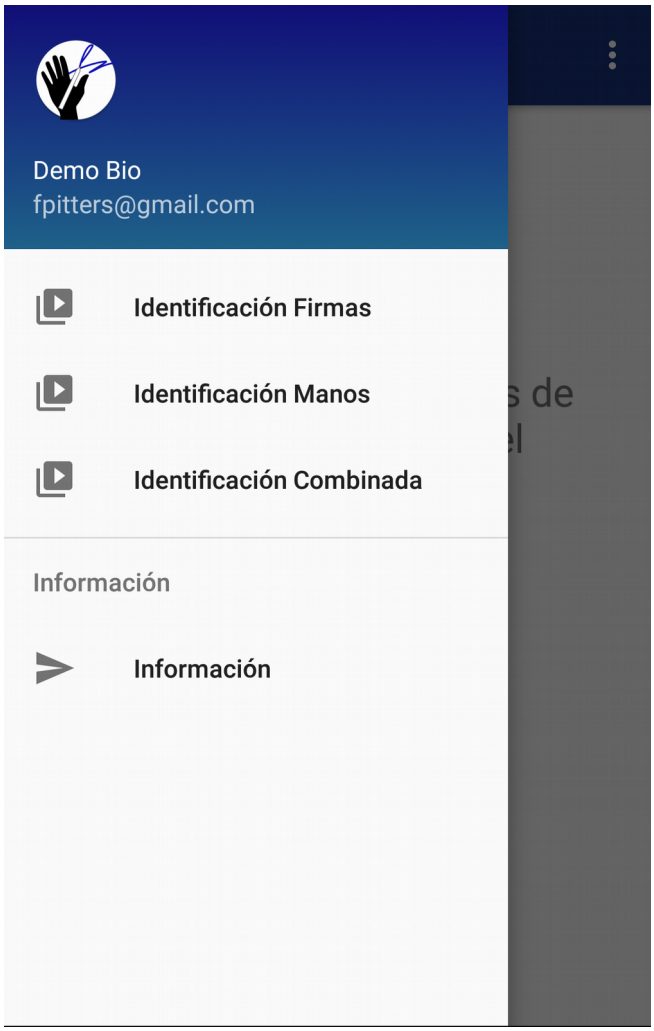

Figura 6.2: Pantalla menú lateral

6.1.2.4 Pantallas de selección de individuo y muestra

Para los tres sistemas biométricos se ofrecen las mismas pantallas de navegación que permiten en primera instancia seleccionar un individuo dentro de los disponibles para un proceso de identificación. A continuación se presenta para la persona seleccionada las posibles muestras para elegir una con la que realizar el test. Las interfaces se adaptan a los distintos contenidos, así para firmas y manos se utiliza una estructura grid con distintas alturas y para el modelo combinado se presenta una lista con la composición de la palma de la mano a la izquierda y la firma a la derecha.

En la figura 6.3 se puede ver un ejemplo de la pantalla de selección de individuos para manos (izquierda), la pantalla de selección de muestras para firmas e individuo 5 (centro) y por último la pantalla de selección combinada para el individuo 2 (derecha).

Figura 6.3: Pantallas de selección de individuos y muestras

6.1.2.5 Pantalla de evaluación y resultados

Se trata de la interfaz presentada al usuario durante el proceso de identificación en el servidor y el posterior resultado.

A continuación, en la figura 6.4, se ofrecen ejemplos de un resultado satisfactorio en el que la muestra de la mano ha sido correctamente identificada y una identificación incorrecta para una muestra combinada.

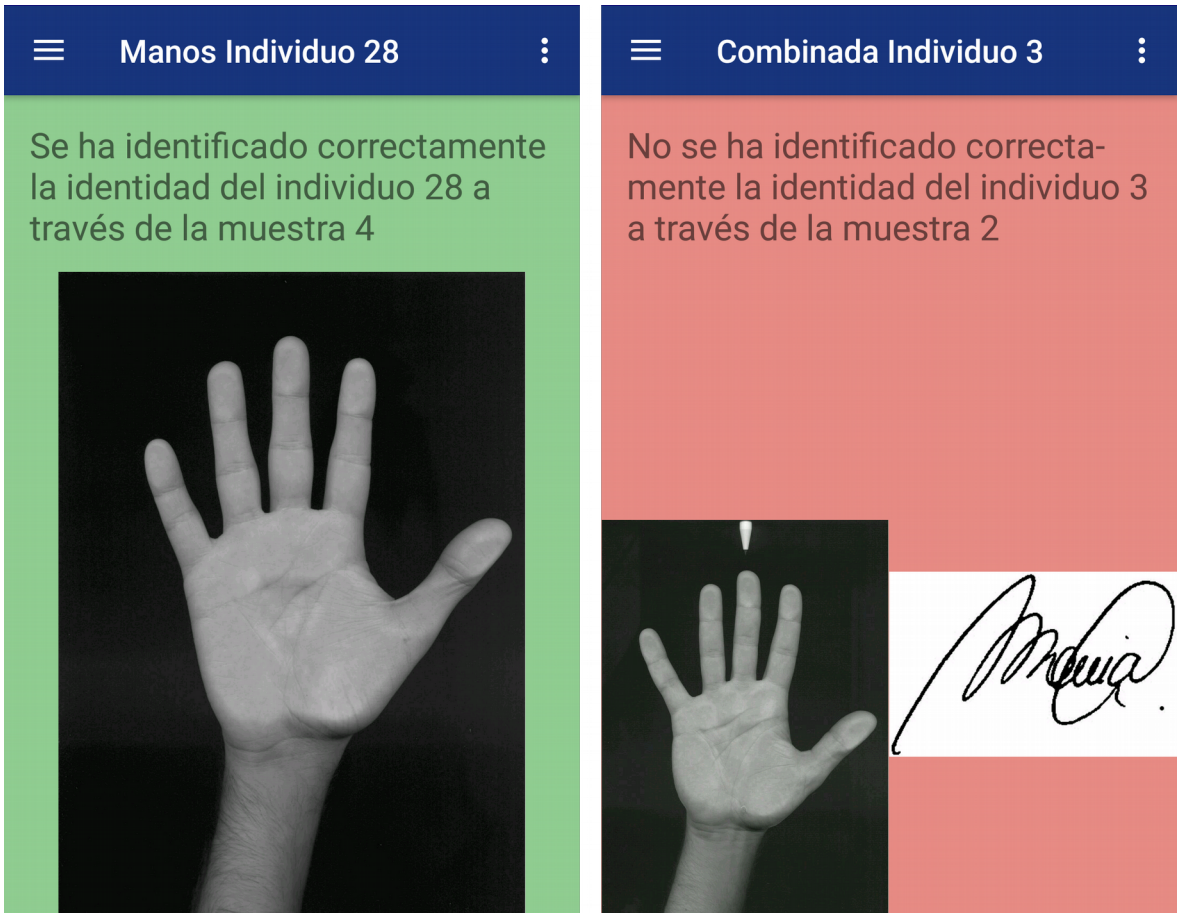

Figura 6.4: Pantalla de evaluación y resultado

7 Conclusiones

Se recoge en este capítulo un análisis sobre los datos obtenidos en los distintos experimentos realizados. Se revisa también la viabilidad del sistema para ser implantando en un entorno de producción y las modificaciones necesarias. Por último se revisan ideas no desarrolladas que podrían conformar mejoras sobre el sistema construido.

7.1 Análisis de resultados

Dados los resultados obtenidos se puede resumir que el objetivo del presente trabajo de estudiar y desarrollar un sistema de identificación biométrico para muestras fusionadas de firmas manuscritas y la forma de la palma de la mano se ha cumplido. Los resultados han sido satisfactorios y en general competentes respecto a otros valores vistos en las referencias visitadas para este trabajo.

A través de los experimentos realizados en el apartado 5.4 reduciendo el porcentaje de entrenamiento se ha obtenido que el sistema biométrico de medidas combinadas de firmas y manos es el más robusto a la hora de disminuir las muestras de entrenamiento, manteniendo un 100% de acierto para un conjunto de entrenamiento del 40% y dando un rendimiento superior para conjuntos de entrenamiento inferiores. Se concluye por tanto que el sistema biométrico de muestras fusionadas ofrece un comportamiento mejor frente a los sistemas independientes al reducir el conjunto de entrenamiento.

A continuación se ofrece una tabla resumen de los mejores resultados para cada sistema.

Sistema / Clasificador	Manos	Firmas	Combinado
MOM	97.52 % ± 2.21	85.82 % ± 1.01	92.69 % ± 2.08
MVS	100%	82.55 % ± 0.01	100%
MOM + MVS	100%	100%	100%

Tabla 7.1: Resumen de mejores resultados

Es patente en los datos expuestos que el mejor comportamiento viene dado por el clasificador compuesto por MOM y MVS a través del Kernel de Fisher. Para los los tres sistemas: base de datos de manos, base de datos de firmas y características combinadas, ofrece un resultado del 100%.

Es importante destacar, por una lado, para la base de datos de manos de forma independiente se podría considerar el uso único de MVS que también ofrece un 100% de acierto en identificación y el rendimiento en cuanto a tiempos de ejecución es mejor que haciendo uso del sistema combinado. Por otro lado se aprecia una disminución del acierto para el caso de MOM y las muestras combinadas. Una posibilidad abierta a un desarrollo futuro sería analizar otras opciones para la combinación de clasificadores [24], como pudiera ser a través de un sistema de votaciones o métodos “boosting”.

Analizados los resultados y de cara a una implementación final del sistema biométrico de medidas combinadas de la mano y la firma se propone el clasificador conjunto de Modelos Ocultos de Markov y Máquinas de Vectores Soporte junto con el Kernel de Fisher.

7.2 Sistema de producción

Los resultados recogidos en el apartado anterior certifican que los sistemas biométricos independientes y combinados estarían preparados para ser implantados en un sistema de producción. Si bien sería conveniente analizar las debilidades de los sistemas implementados y prever posibles inconvenientes. Se enumeran a continuación una serie de pequeñas mejoras o potenciales problemas que deberían ser visitados de cara a la puesta en producción:

- Test de código. Sería conveniente reforzar la implementación del código a través de test unitarios o test funcionales para encontrar potenciales errores que pueden permanecer ocultos con las pruebas ya realizadas.
- Aseguramiento de la calidad (o mejor conocido como Quality Assurance (QA) en inglés). Para el análisis de calidad es conveniente disponer de un equipo dedicado que lleve a cabo un conjunto de pruebas establecidas para el sistema final. Aunque no siempre es viable tener un equipo independiente, la creación de un protocolo de QA es recomendable para evitar la fuga de errores a producción.
- Mejoras en rendimiento. La implementación en lenguajes de código interpretado ofrece numerosas ventajas: sencillez, código de alto nivel, tiempos reducidos de desarrollo, etc. Pero pueden convertirse en un cuello de botella en flujos de ejecución porque el rendimiento de los mismo es menor frente a un código compilado y optimizado para la arquitectura del procesador. Sería por tanto recomendable traducir los algoritmos implementados o utilizados a un lenguaje que pueda ofrecer mejor comportamiento como puede ser C/C++.

7.3 Líneas futuras

Por último se analizan posibles mejoras del sistema que pudieran ser implementadas en posteriores fases con el fin de obtener mayor funcionalidad:

- Sistema embebido en dispositivo móvil. La demo realizada para el presente trabajo ofrece una implementación del sistema en un servidor dando acceso al

mismo a través de servicios web HTTP para una interfaz creada para la plataforma Android. Aunque esta estructura ha sido muy común en la última década dada la limitada capacidad de procesamiento de los dispositivos portables, en lo últimos años han mejorado lo suficiente como para ser capaces de ejecutar los algoritmos de forma embebida en tiempos ajustados. De esta forma permitiría el funcionamiento del sistema incluso en casos en que no exista conexión a Internet.

- Escalado del sistema. Se propone revisar el comportamiento del sistema frente a un número mucho mayor de clases de entradas, es decir, individuos. Se han utilizado para el sistema combinado muestras de 144 personas diferentes obteniendo un acierto total en todos los casos, pero es probable que a medida que se escala el sistema a un número mayor de personas, los resultados se vean mermados. Si bien se estima que el uso de muestras diferentes debe propiciar un mejor comportamiento sería conveniente analizar los límites del conjunto.
- Sistema Online a través de cámara fotográfica y pantalla táctil. El uso del sistema desarrollado puede ser extendido a un sistema online en tiempo real a través del uso de los sensores que podemos encontrar en los móviles de hoy en día. De esta forma se propone la cámara para la obtención de la muestra de la mano y la pantalla táctil para obtener el trazo de una firma. Sería necesario un desarrollo de procesado de datos más cuidadoso pues las entradas podrían contener más ruido y por tanto sería más complicado obtener los vectores finales.

Bibliografía

1. Denice Lauber, “Biometrics – A Brief Overview”, SANS Institute 2003.
2. Burghardt T. “A Brief Review of Biometric Identification”. Technical Report, Visual Information Laboratory, Bristol, 2012.
3. Joseph N. Pato and Lynette I. Millett, Editors. “Biometric Recognition: CHALLENGES AND OPPORTUNITIES” Washington, DC: The National Academies Press. DOI:https://doi.org/10.17226/12720, 2010

   Última visita 6 Mayo 2017: https://www.nap.edu/read/12720/chapter/1
4. T. Burghardt. “Visual Animal Biometrics”, PhD Thesis, University of Bristol, 2008.
5. Blase Ur, Fumiko Noma, Jonathan Bees, Sean M. Segreti, Richard Shay, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, “I Added ‘!’ at the End to Make It Secure. Observing Password Creation in the Lab”, Symposium on Usable Privacy and Security, pág. 123 - 140. 2015.
6. Davy Cielen, Arno D. B. Meysman, Mohamed Ali, “Introducing Data Science – Big Data, Machine Learning and more, using Python tools”, 2016.
7. Henrik Brink, Joseph W. Richards, Mark Fetherolf, “Real World Machine Learning”, 2017.
8. Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani, “An Introduction to Statistical Learning with Applications in R”, 2013.
9. Briceño J.C., Travieso C.M., Ferrer M.A., Alonso J.B., Vargas F. “Angular Contour Parameterization for Signature Identification”. In: Moreno-Díaz R., Pichler F., Quesada-Arencibia A. (eds) Computer Aided Systems Theory - EUROCAST 2009. Lecture Notes in Computer Science, vol 5717. Springer, Berlin, Heidelberg
10. José Aridane Rodríguez Rodríguez, “Clasificación biométrica de la mano basada en información geométrica”, Proyecto Fin de Carrera ULPGC, 2010.
11. B. Mathivanan , Dr. V. Palanisamy , Dr. S. Selvarajan, “Multi Dimensional Hand Geometry Based Biometric Verification and Recognition System”, 2012.

12. Ali Karouni, Bassam Daya, Samia Bahlak, “Offline signature recognition using neural networks approach”, Procedia Computer Science, Volume 3, 2011, pág 155 - 161, ISSN 1877-0509, http://dx.doi.org/10.1016/j.procs.2010.12.027.

13. K.R. Radhika, M.K. Venkatesha and G.N. Sekhar, “Off-Line Signature Authentication Based on Moment Invariants Using Support Vector Machine”, Journal of Computer Science 6 (3): 305-311, 2010.

14. Doroz, R., Porwik, P., Para, T. and Wrobel, K, “Dynamic signature recognition based velocity changes of some features”, Int. J. Biometrics, Vol. 1, No. 1, pág. 47 - 62.

15. Andrew Y. Ng., Michael I. Jordan, “On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes”,

    Última visita 5 de Mayo de 2017: http://papers.nips.cc/paper/2020-on-discriminative-vs-generative-classifiers-a-comparison-of-logistic-regression-and-naive-bayes.pdf.

16. Rabiner, L.R. “A tutorial on hidden Markov models and selected applications inspeech recognition”, Proceedings of the IEEE, Vol.77, Issue 2, pág. 257 - 286, Febrero 1989.

17. Sébastien David in collaboration with Miguel A. Ferrer, Carlos M. Travieso and Jesús B. Alonso., “GpdsHMM Toolbox, user’s guide”, versión 0.0. Última visita 5 de Mayo de 2017: http://www.gpds.ulpgc.es/download/toolbox/gpdshmm/gpdsHMM%20toolbox%20for%20matlab%20v1.0.pdf

18. Burges, Christopher J.C., "A Tutorial on Support Vector Machines for Pattern Recognition" Data Mining and Knowledge Discovery, Vol. 2, pág. 121 - 167, Enero 1998.

19. Cristianini, Nello, and John Shawe-Taylor, “An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods”, Cambridge, UK: Cambridge University Press. 2000

20. Martin Sewell, “The Fisher Kernel: A Brief Review”, Research Note UCL DEPARTMENT OF COMPUTER SCIENCE, 20 Enero 2011.

21. Tommi Jaakkola and David Haussler, “Exploiting Generative Models in Discriminative Classifiers”. Advances in Neural Information Processing Systems 11, pág. 487 - 493, 1998.

22. Koby Crammer and Yoram Singer, “On the algorithmic implementation of multiclass kernel-based vector machines.” J. Mach. Learn. Res. 2 , pág. 265 - 292, March 2002.

23. T. Joachims, Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999. Última visita 5 Mayo de 2017: http://svmlight.joachims.org/

24. Tulyakov, Sergey, and Jaeger, Stefan, and Govindaraju, Venu, and Doermann, David. “Review of Classifier Combination Methods”, Machine Learning in Document Analysis and Recognition, pág 361 - 386, 2008

Anexo A : Extensión Modelos Ocultos de Markov

A continuación se detalla una resolución analítica a los problemas definidos [16] necesarios para su implantación en los sistema de aprendizaje automático.

A.1 Problema de evaluación

La resolución directa del problema de evaluación, hallar P[O|λ], pasa por calcular la suma de todas las posibles secuencias de estados que cumple con la secuencia de observación.

El cálculo procedería.

Para una secuencia de estados $S = q_1,q_2,...,q_T$, la probabilidad de la secuencia es:

$$P\left[S \mid \lambda \right] = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \ldots a_{q_{T-1} q_T} \tag{A.1}$$

Para una secuencia de observaciones $O=O_1,O_2,...,O_T$ dada para la particular secuencia de estados anterior:

$$P\left[O \mid S,\lambda\right] = \prod_{t=1}^{T} P\left[o_t \mid q_t,\lambda\right] \;/\; b_{q_t}\left(o_t\right) = P\left[o_t \mid q_t,\lambda\right] \tag{A.2}$$

Quedando la probabilidad de una secuencia O de observaciones para un modelo λ (A, B, π):

$$P\left[O|\lambda\right] = \sum^{S} P[S|\lambda]\; P\left[O|S,\lambda\right] \tag{A.3}$$

El cálculo a través de la ecuación A.3 es impracticable por tener una dependencia exponencial con el número de observaciones. Por ello se recurre al algoritmo adelante-atrás (También llamado avance-retroceso o forward-backward en inglés) que resuelve el problema de una forma más eficiente.

Algoritmo avance

Considerando la variable $\alpha_t(i)$ como la probabilidad de observar la $o_1,o_2,...,o_T$, y estar en el instante t en el estado i:

$$\alpha_t\left(i\right) = P\left[o_{1,}\, o_{2,\ldots}, o_t, q_t = i \mid \lambda\right] \tag{A.4}$$

El cálculo hacia adelante, o de avance de la probabilidad de una secuencia de observaciones es:

1. Inicialización

$$\alpha_1\left(i\right) = \pi_i b_i\left(o_1\right) \quad 1 \le i \le N \tag{A.5}$$

2. Recurrencia

$$\alpha_{t+1}(j)=\left[\sum_{i=1}^{N}\alpha_t(i)a_{ij}\right]b_j(o_{t+1})\quad t=1,2,\ldots,T-1\ ,\ 1\le j\le N \tag{A.6}$$

3. Terminación

$$P\left[O|\lambda\right]=\sum_{i=1}^{N}\alpha_T(i) \tag{A.7}$$

Algoritmo retroceso

Considerando la variable $\beta_t(i)$ como la probabilidad de observar la $o_{t+1},o_{t+2},\ldots,o_T$, desde el instante t+1 hasta T cuando el estado en t es i:

$$\beta_t(i)=P\left[o_{t+1},o_{t+2},\ldots,o_T,q_t=i\mid\lambda\right] \tag{A.8}$$

El cálculo hacia atrás, o de retroceso, de la probabilidad de una secuencia de observaciones es:

1. Inicialización

$$\beta_T(i)=1\quad 1\le i\le N \tag{A.9}$$

2. Recurrencia

$$\beta_t(i)=\sum_{j=1}^{N}a_{ij}\beta_{t+1}(j)b_j(o_{t+1})\quad t=T-1,T-2,\ldots,1\ ,\ 1\le i\le N \tag{A.10}$$

3. Terminación

$$P\left[O|\lambda\right]=\sum_{i=1}^{N}\beta_1(i)\pi_i b_i(o_1) \tag{A.11}$$

A.2 Problema de decodificación

El problema de decodificación busca esclarecer la información oculta de un MOM dado, es decir los estados partícipes para una secuencia de observación generada. No existe una respuesta concreta a, ¿cuál es la secuencia de estados correcta para un modelo λ(A,B,π), y una secuencia de observaciones, $O=O_1,O_2,...,O_T$, dados?. En consecuencia, se recurre a algún criterio que en la práctica permita resolver el problema de la mejor forma posible. Se suelen describir en la literatura como los estados "óptimos" o que mejor "describen" la observación dada.

Un posible criterio y el más extendido en uso, consiste en hallar la secuencia de estados (camino, o "path" en inglés) que hacen máxima la probabilidad P[Q|O, λ] para cada estado individual.

Para el procedimiento se define la siguiente variable, $\delta_t(i)$, probabilidad del mejor camino hasta el estado i conocidas las t primeras observaciones, función que se evalúa para todos los estados e instantes de tiempo.

$$\delta_t(i) = \max_{q_1, q_{2,\ldots}, q_{t-1}} P[q_1, q_{2,\ldots}, q_t = i, o_1, o_{2,\ldots}, o_t|\lambda] \tag{A.12}$$

Que de forma inductiva:

$$\delta_{t+1}(i) = [\max_j \delta_t(i) a_{ij}]\, b_j(o_{t+1}) \tag{A.13}$$

Además es necesario definir una nueva variable que almacene el argumento que hace máxima la ecuación A.13. Ésta es, $\psi_t(j)$.

El algoritmo es como sigue,

1. Inicialización

$$\delta_1(i) = \pi_i b_i(o_1) \quad 1 \le i \le N \tag{A.14}$$

$$\psi_1(i) = 0 \tag{A.15}$$

2. Recurrencia

$$\left[\delta_{t-1}(i) a_{ij}\right] b_j(o_t) \quad 2 \le t \le T \quad , \quad 1 \le j \le N ¿ \tag{A.16}$$

$$\left[\delta_{t-1}(i) a_{ij}\right] \quad 2 \le t \le T \quad , \quad 1 \le j \le N ¿ \tag{A.17}$$

3. Terminación

$$P^* = \max_{1 \le i \le N} [\delta_T(i)] \tag{A.18}$$

$$q_T^* = \operatorname*{argmax}_{1 \le i \le N} [\delta_T(i)] \tag{A.19}$$

4. Reconstrucción de la secuencia de estados más probable (camino)

$$q_t^* = \psi_{t+1}(q_{t+1}^*) \quad t = T-1, T-2, \ldots, 1 \tag{A.20}$$

A.3 Problema de entrenamiento

El tercer problema característico en los MOM se centra en el ajuste de los parámetros del modelo, es decir, la probabilidad de transición A, la probabilidad de símbolo por estado B y la probabilidad de estado inicial π, con el fin de maximizar la probabilidad de observación. No existe un procedimiento analítico para resolver el presente problema pero es posible elegir unos parámetros del modelo λ(A,B,π) tal que P[O|λ] sea un máximo local. Se recurre para ello al algoritmo iterativo de Baum-Welch, o al uso de técnicas basadas en el gradiente.

Se describe a continuación el método de Baum-Welch como aproximación para resolver el problema. El propósito es conocer:

- Número esperado de transiciones desde el estado i en O.
- Número esperado de transiciones desde el estado i al estado j en O.

Se define la variable $\xi_t(i,j)$ como la probabilidad de estar en el estado i en el instante t y en el estado j en el instante t+1, dado una observación O y el modelo λ.

$$\xi_t(i,j) = P\left[q_t = S_i, q_{t+1} = S_j | O, \lambda\right] = \frac{P\left[q_t = S_i, q_{t+1} = S_j, O | \lambda\right]}{P\left[O|\lambda\right]} \tag{A.21}$$

Introduciendo las variables de los procesos avance-retroceso queda como,

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{P\left[O|\lambda\right]} = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)} \tag{A.22}$$

Se define también $\gamma_t(i)$ como la probabilidad de estar en el estado i en el instante j.

$$\gamma_t(i)=\sum_{j=1}^{N}\xi_t(i,j) \tag{A.23}$$

Realizando un sumatorio en las ecuaciones anteriores para los instantes de tiempo t = 1,..., T-1 se llega a las estimaciones buscadas.

- Número esperado de transiciones desde el estado i en la observación O:

$$\sum_{t=1}^{T-1}\gamma_t(i) \tag{A.24}$$

- Número esperado de transiciones desde el estado i al estado j en la observación O:

$$\sum_{t=1}^{T-1}\xi_t(i,j) \tag{A.25}$$

Reestimación del modelo: proceso iterativo

El proceso iterativo es el siguiente:

1. Se define un MOM inicial, aleatoriamente o de forma determinada.
2. Se realiza el cálculo de las transiciones y símbolos emitidos más probables según el modelo inicial escogido.
3. Se estima un nuevo modelo. Para la secuencia de observaciones dada el modelo tendrá una probabilidad mayor que el anterior.

Se repite el proceso de entrenamiento hasta que no se aprecie mejora entre un modelo y el siguiente estimado.

La estimación de los parámetros es como sigue:

$$_i=\gamma_1(i)\quad 1\le i\le N\ \pi \tag{A.26}$$

$$\bar{a}_{ij}=\frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)}\quad 1\le i,j\le N \tag{A.27}$$

$$\bar{b}_j(o_k) = \frac{\sum_{t=1, o_t = o_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)} \quad 1 \le j,k \le N \tag{A.28}$$

Anexo B : Extensión de Máquina de Vectores Soporte

Se recoge a continuación un análisis analítico del proceso obtención del hiperplano separador en las Máquinas de Vectores Soporte de tipo lineal y no lineal.

B.1 Resolución hiperplano Máquina de Vectores Soporte lineales

Para una MVS lineal con clases separables se definen los vectores para 2 clases dadas como

$\{(x_1 , y_1),..., (x_n , y_n)\}$ donde $x_i \in IR_d$ e $y_i \in \{-1, 1\}$ para i = 1, · · · , n

Para la que existe un hiperplano descrito por que maximiza la distancia de los vectores soporte,

$$\pi: \ w \cdot x + b = 0 \ , \tag{B.1}$$

Y por tanto mínima ||w||:

$$\min_{w \in \mathbb{R}^d} \frac{1}{2} \|w\|^2 \tag{B.2}$$

Para la resolución del problema se recurre a los multiplicadores de Lagrange. La función objetivo queda como:

$$L_P(w,b,\alpha_i) \ = \ \frac{1}{2}\|w\|^2 \ - \ \sum_{i=1}^{n} \alpha_i \left(y_i \left(x_i \cdot w + b \right) - 1 \right) \tag{B.3}$$

La función objetivo es convexa describiendo un problema de tipo cuadrático y los vectores que satisfacen las restricciones forman un conjunto convexo. Por esto último, es posible la resolución del problema dual asociado al problema Primal (razón del índice "P"). Se buscaría maximizar la función $L_P(w,b,\alpha_i)$ respecto de las variables duales α_i sujeta a las restricciones impuestas para que los gradientes de L_P con respecto a w y b sean nulos y además (α_i >= 0, i = 1,...,n). Esto se resume en las siguientes condiciones, equivalentes para ambas formulaciones:

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i \ , \ \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{B.4}$$

Sustituyendo los límites anteriores obtenemos la siguiente función objetivo:

$$L_D(w,b,\alpha_i) \ = \ \sum_{i=1}^{n} \alpha_i \ - \ \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i x_j \tag{B.5}$$

Los puntos de entrenamiento que proporcionan un multiplicador de Lagrange $\alpha_i > 0$, son denominados vectores soporte y se encuentran en alguno de los hiperplanos. El resto de puntos tienen $\alpha_i = 0$. En consecuencia, los vectores soporte, son los puntos críticos para el proceso de entrenamiento. El resto de puntos, son sumamente irrelevantes, pues no condicionan la definición de los hiperplanos separadores.

Condiciones de Karush–Kuhn-Tucker

Para el problema primal quedan definidas como:

$$\frac{\partial}{\partial w_v} L_P = w_v - \sum_{i=1}^{n} \alpha_i y_i x_{iv} = 0 \quad v = 1, \ldots, d \tag{B.6}$$

$$\frac{\partial}{\partial b} L_P = -\sum_{i=1}^{n} \alpha_i y_i = 0 \tag{B.7}$$

$$y_i \left(x_i \cdot w + b \right) - 1 \geq 0 \quad \forall i \tag{B.8}$$

$$\alpha_i \geq 0 \quad \forall i \tag{B.9}$$

$$\alpha_i \left(y_i \left(x_i \cdot w + b \right) - 1 \right) = 0 \quad \forall i \tag{B.10}$$

Las condiciones KKT, son para problemas convexos, como el caso de las MVS, condiciones necesarias y suficientes para que w, b y alfa sean solución. La resolución del problema de MVS es equivalente a encontrar la solución a las KKT conditions.

Como aplicación inmediata, mientras que w es explícitamente determinada en el proceso de entrenamiento, el umbral b no lo es. Sin embargo es fácilmente deducible a través de la condición de complementariedad (Ec. B.10). Tomando cualquier i para el que α_i != 0, y calculando b. Es común en la práctica, el cálculo de b para todos los vectores soportes y el promediado entre todos ellos.

El problema de entrenamiento de una MVS ha sido transcrito como un problema de optimización con unos límites más manejables.

Datos no separables

Se dice que el conjunto es no separable cuando existen vectores etiquetados en una clase dentro de la región etiquetada como clase opuesta, de forma que no puede existir un hiperplano que delimite ambos conjuntos. El problema de optimización presentado

anteriormente no encuentra solución en estos casos, encontrándose con que la función objetivo, L_D, crece arbitrariamente sin límite.

A continuación, en la Figura B.1, se muestra un ejemplo de un conjunto de entrada con datos no separables.

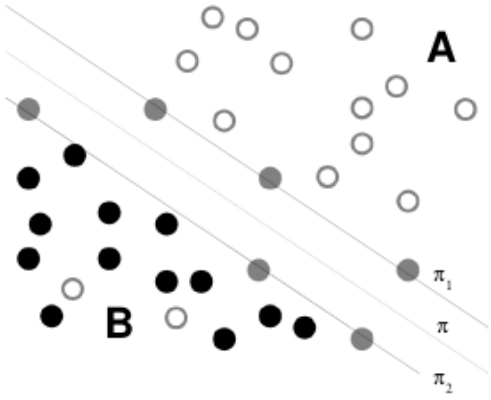

Figura B.1: Ejemplo de muestras no separables linealmente.

La respuesta a este problema pasa por introducir una variable ξ de holgura siendo las nuevas restricciones:

$$x_i \cdot w + b \geq +1 - \xi_i \quad \forall y_i = +1 \tag{B.11}$$

$$x_i \cdot w + b \leq -1 + \xi_i \quad \forall y_i = -1 \tag{B.12}$$

$$\xi_i \geq 0 \quad \forall i \tag{B.13}$$

Para que se de un error, la correspondiente ξ_i debe ser mayor que la unidad, siendo por tanto $\Sigma_i \xi_i$ un límite superior del número de errores. Se introduce en la función objetivo a minimizar del problema anterior: $||w||^2/2$ por $||w||^2/2 + C(\Sigma_i \xi_i)k$. C es un parámetro a elección para cada diseño, correspondiendo una C mayor a una mayor penalización de los errores. Sigue siendo un problema convexo para cualquier entero positivo k. Si k=1,2 es además un problema cuadrático. Y por último la elección de k=1 presenta la ventaja que ni ξ_i ni los correspondientes multiplicadores de Lagrange aparecen en el planteamiento del problema dual a maximizar,

$$L_D(w, b, \alpha_i) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i x_j \tag{B.14}$$

Sujeto a las restricciones,

$$0 \leq \alpha_i \leq C \quad , \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{B.15}$$

La solución es igualmente:

$$w = \sum_{i=1}^{N_s} \alpha_i y_i x_i \tag{B.16}$$

Dónde NS es el número de vectores soporte. La única diferencia con el caso en que los conjuntos si son separables es que α_i tiene un límite superior de C.

El problema primal es:

$$L_P \; = \; \frac{1}{2}\|w\|^2 \; - \; \sum_{i=1}^{n} \alpha_i \left(y_i \left(x_i \cdot w + b \right) - 1 \right) \; - \; \sum_{i=1}^{n} \mu_i \xi_i \quad , \tag{B.17}$$

los μ_i son los multiplicadores de Lagrange introducidos para obligar la positividad de las variables de error introducidas, ξ_i .

Las condiciones KKT para el problema original:

$$\frac{\partial}{\partial w_v} L_P = w_v - \sum_{i=1}^{n} \alpha_i y_i x_{iv} = 0 \quad v = 1, \ldots, d \tag{B.18}$$

$$\frac{\partial}{\partial b} L_P = - \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{B.19}$$

$$\frac{\partial}{\partial \xi_i} L_P = C - \alpha_i - \mu_i = 0 \tag{B.20}$$

$$y_i \left(x_i \cdot w + b \right) - 1 + \xi_i \geq 0 \quad \forall i \tag{B.21}$$

$$\xi_i \geq 0 \quad \forall i \tag{B.22}$$

$$\alpha_i \geq 0 \quad \forall i \tag{B.23}$$

$$\mu_i \geq 0 \quad \forall i \tag{B.24}$$

$$\alpha_i\left(y_i\left(x_i \cdot w+b\right)-1+\xi_i\right)=0 \quad \forall i \tag{B.25}$$

$$\mu_i \xi_i=0 \tag{B.26}$$

Del mismo modo se recurre a las ecuaciones B.25 y B.26 para determinar el valor de b.

Solución de una MVS lineal

El problema de minimización busca determinar un hiperplano separador óptimo:

$$\pi \equiv f\left(x;w,b\right)=\langle w,x\rangle+b=0 \tag{B.27}$$

Dónde se busca estimar los parámetros:

$$w\in\mathbb{R}^d \quad y \quad b\in\mathbb{R} \tag{B.28}$$

Se puede considerar el siguiente conjunto de funciones sobre el que se busca la solución al problema:

$$F=\left\{ f\left(x;w,b\right)=\langle w,x\rangle+b, \;\; w\in\mathbb{R}^d, y\in\mathbb{R}\right\} \tag{B.29}$$

Y

$$\min_{w\in\mathbb{R}^d} L_P\left(x;w,b\right)=\min_{f\in F} L_P\left(x;f\right) \tag{B.30}$$

Y esto último puede ser considerado como un riesgo regularizado por lo que:

$$\min_{w\in\mathbb{R}^d} L_P\left(x;w,b\right)=\min_{f\in F} R_{reg}[f] \tag{B.31}$$

Por otro lado, dado que una vez determinado el valor óptimo de w, haciendo uso de las condiciones KKT se obtiene el parámetro b, el objetivo en esencia del problema se centra en el vector de parámetros w. Dicho vector w se obtiene mediante una combinación lineal de los vectores del conjunto de entrenamiento:

$$w=\sum_{i=1}^{N_s} \alpha_i y_i x_i \tag{B.32}$$

Y la función objetivo quedaría:

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i \langle x_i, x \rangle + b \tag{B.33}$$

B.2 Resolución hiperplano Máquina de Vectores Soporte no lineales: kernels

La aplicación de los métodos descritos a clases de funciones no lineales no es inmediata y requiere la inclusión de algunos conceptos que se describen a continuación.

En la ecuación B.33 se observa que la solución al problema de clasificación depende del producto escalar de los vectores de entrada: x_i y x_j . Se propone ahora transformar los datos a algún otro espacio Euclideo H, mediante un mapeo como el siguiente:

$$\Phi: X \subset \mathbb{R}^d \rightarrow H \tag{B.34}$$

De modo que en lugar de considerar los vectores $\{x_1, \ldots, x_n\}$ se tomarán los vectores transformados $\{\Phi_1,...,\Phi_n\}$

Para el problema de clasificación estudiado los nuevos vectores formarán parte de la solución a través del producto escalar definido en el nuevo espacio H:

$$\langle \Phi(x_i), \Phi(x) \rangle \tag{B.35}$$

Entonces, si existiera una función que se denominará núcleo (En inglés Kernel) de modo que :

$$k: HxH \rightarrow \mathbb{R} \tag{B.36}$$

$$k(x,x') = \langle \Phi(x), \Phi(x') \rangle_H = \Phi(x) \cdot \Phi(x') \tag{B.37}$$

Sólo es necesario conocer la función núcleo para resolver el problema sin la necesidad de contar con la función Φ explícitamente.

La solución transformada viene dada por:

$$w = \sum_{i=1}^{N_{SV}} \alpha_i y_i \Phi(s_i) \tag{B.38}$$

Ahora bien, replanteando la función objetivo se tiene:

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i \langle \Phi(s_i), \Phi(x) \rangle + b \tag{B.39}$$

Y finalmente introduciendo la definición de núcleo:

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i k(s_i, x) + b \tag{B.40}$$

Una importante consecuencia es que la dimensión del espacio característico no afecta a los cálculos ya que la única información necesaria se encuentra en la matriz de orden n x_n : $\{k(x_i,x_j)\}$, i,j=1,..,n. (Matriz de Gram).

Anexo C : Propuesta de publicación IEEE

Se recoge en este Anexo una propuesta de publicación para un congreso IEEE con el mismo desarrollo contenido en este trabajo.

Biometric identifier based on hand and hand-written signature contour information

Fernando A. Pitters-Figueroa, Carlos M. Travieso
Signals and Communications Department
IDETIC - University of Las Palmas de Gran Canaria
Las Palmas de Gran Canaria, Spain
fpitters@gmail.com, carlos.travieso@ulpgc.es

Malay Kishore Dutta
Amity University
Noida, India
malaykishoredutta@gmail.com

***Abstract*—The present work presents a biometric identifier system using the combination of two different features: hands shape (finger lengths and width) and hand-written signature contour. Signature database contains 300 different signers with 24 signatures and the hand database has 144 owners with 10 images. The study covers three different classifiers: Hidden Markov Models (HMM), Support Vector Machines (SVM) and a combination of both using the Fisher Kernel. Systems are evaluated separately and in conjunction, giving in each case 100% of identification success rate for the combined classifier. The combination of features gives better results when reducing the training set than the independent systems.**

Keywords—biometrics; identification; hand shape; hand-written signature; classification; pattern recognition;

I. INTRODUCTION

Identifying a person into a secured system is a delicate process, cause of concern since the beginning of the computers era. Given the enormous impact of Internet and the largely accepted cloud services it has become of special relevance of the analysis and improvement of current identification process to overcome the existing security issues [1][2]. The most extended identification technique is still the user/password pair used for almost every online system. Several additional methods to provide access to an user have appeared and a few are also of common on our day to day use: personal identification document, electronic certificates, phone text messages with pin codes. However during the past two decades biometric technology has broken in as a feasible alternative to the common identification system.

Biometric science is focused on measuring biologic data from living beings and obtain statistical results that may allow catalog species though common patterns, analyze potential diseases and its consequences or get unique measures from an individual. Some biometrics measurements provide distinctive enough properties that make them suitable for a recognition system.

There are several biometric systems under use on production environments that certify their success. Just to mention a few of the most popular: face recognition, fingerprint or iris scanners, DNA analysis. The biometric metrics can be categorized in two groups: biological system, based on physiological data, and behavioral system, based on information learned or acquired.

A biometric system relies on data science for the recognition process, particularly on machine learning algorithms, also known as prediction functions or classifiers. For the process of discovering the actual owner for a given data sample, biometric systems make use of supervised algorithms, where each data sample belongs to a class, an individual.

On this study, an identification system based on biometric information of the hand and fingers shape and handwritten signature contour is proposed. The metrics are analyzed separately and then combined.

II. METHODS

Classification models proposed for the study are: Hidden Markov Models, Support Vector Machines and a combination of both with the Fisher transformation.

A. Hidden Markov Models

Hidden Markov Models (HMM) are an adaptation of standard Markov Models or Markov Chains, that extends the stochastic process to the case where the output observations are based on a probabilistic function of the state and not the state itself as in the Markov Chains [3][4].

For a set of N states $S_1,S_2,...,S_N$ with q_t indicating the state on the instant t (t = 1,2,...). The transition probability between states is given by:

$$a_{ij} = P\left[q_t = S_i \mid q_{t-1} = S_j \right] \quad i,j = 1,2,..,N$$

The initial states distribution is:

$$\pi_i = P\left[q_1 = S_i \right] \quad i = 1,2,...,N$$

The model output consists on M symbols, $V_1,V_2,...,V_M$, and the probability distribution of an output for each state is

$$bi(k) = P\left[O_t = v_k \mid q_t = S_i \right] \quad i = 1,2,..,N; \quad k = 1,2,..,M$$

Then the model can be summarized as:

- N states. $S = S_1,S_2,...,S_N$
- M symbols. $V = V_1,V_2,...,V_M$
- State transition probability distribution. $A = a_{ij}$
- Symbol emission probability distribution. $B = b_j(k)$
- Initial state probability distribution. $\pi = \pi_i$

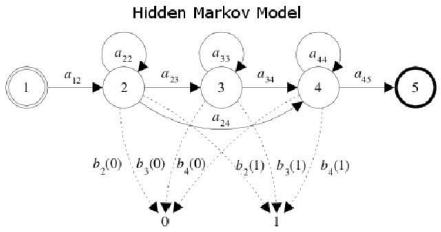

Fig. 1.Example of Hidden Markov Model with 5 states and 2 symbols.

Finally the model is fully described as HMM (S, V, A, B, π), with the following notation broadly adopted on the literature: λ (A, B, π).

Fig. 1 shows an example of HMM with 5 states and 2 different outputs '0' o '1'.

The application of the HMM as predictive function requires the resolution of three different problems [4].

1. Evaluation problem. For a given HMM, λ (A, B, π), and a sequence of observations, $O=O_1,O_2,...,O_T$, which is the probability that the observation has been generated by this model, i.e. resolution of P[O| λ].

2. Decodification problem. For a given HMM, λ (A, B, π), and a sequence of observations, $O=O_1,O_2,...,O_T$, what is the state sequence $S = q_1,q_2,...,q_T$, that are optimal.

3. Training problem. Optimization of a given HMM, λ (A, B, π), maximizing P[O| λ].

B. Support Vector Machines

Support Vector Machines are models based on the idea of margin dividing the data points for different classes, precisely, an hyperplane that separates on each side two different labels and at the same time that maximizes the margin between the plane and the set of points, known as vectors, for each class.

For a set of vectors: $\{(x_1 , y_1),..., (x_n , y_n)\}$ where $x_i \in IR^d$ and $y_i \in \{-1, 1\}$ for $i = 1, \cdots, n$

They can be linearly separated if it does exists an hyperplane in R^d that separates the vectors $X = \{x_1 ,..., x_n \}$ with label $y_i = 1$ from the others with labels $y_i = -1$.

$$\pi: \quad w \cdot x + b = 0$$

The SVM tries to find among all the possible hyperplanes the one that maximizes the distance between both class sets. So an hyperplane defined as (w,b) has to satisfy:

$$x_i \cdot w + b \geq +1 \quad \forall\ y_i = +1$$

$$x_i \cdot w + b \leq -1 \quad \forall\ y_i = -1$$

Or,

$$y_i\left(x_i \cdot w + b\right) - 1 \geq 0 \quad \forall\, i$$

The vectors for label +1 that satisfy the equality above belongs to the hyperplane:

$$\pi_1: \quad x_i \cdot w + b = 1$$

With normal vector w and distance to the origin |1-b| / ‖w‖ being ‖w‖ the euclidean norm of w. In similar way, the vectors labeled as -1 that satisfy the equation are:

$$\pi_2: \quad x_i \cdot w + b = -1$$

Also with normal vector w and perpendicular distance to the origin $|-1-b| / ||w||$.

The hyperplanes π_1 y π_2 are parallel and the distance between them is 2 / ‖w‖. Minimizing ‖w‖ will produce a maximum separation between the above hyperplanes:

$$\min_{w \in \Re^d} \frac{1}{2}\|w\|^2$$

The minimum can be obtained using Lagrange Multipliers method with the following expression

$$L_P\left(w,b,\alpha_i\right) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n}\alpha_i\left(y_i\left(x_i \cdot w + b\right) - 1\right)$$

On the Fig. 2 there is an example of the hyperplane π that maximizes the distance between the vectors of the two classes and the planes π_1 y π_2 that limit the regions.

Support Vector Machines also covers the cases where the data is not linearly separable. In those cases the definition has to be modified in order to be able to find the minimum ‖w‖. Further analytic definition can be found on [5]. It is also possible to resolve the problem looking for non-linear hyperplanes, using the called Kernel Method or Kernel Trick. The Kernel is a function that projects the input data into a higher dimensional space so a non-separable set can become linearly separable.

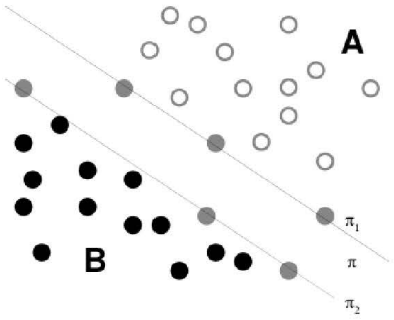

Fig. 2. Example of linear SVM with the hyperplane π that maximizes the separation between classes and hyperplanes π_1 y π_2 containing the support vectors.

C. *Fisher Transformation*

It is a Kernel Method that allows to merge the benefits of the models defined: HMM and SVM [7]. Generative models like HMM try to find a function that simulates the generation of the data and can process data of variable length. On the other hand, discriminative models are known to provide better results.

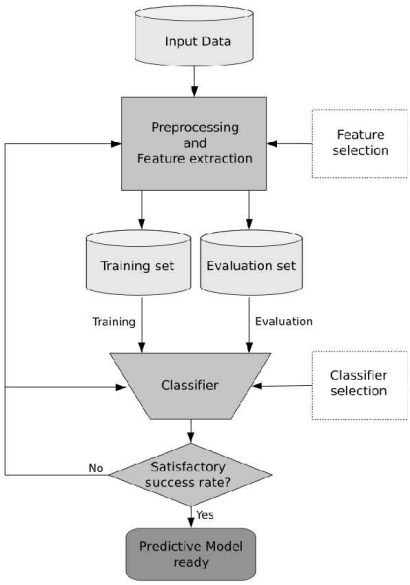

Fig. 3. Diagram for creating a supervised machine learning classifier.

The idea behind Fisher kernel is that similar objects induces similar log-likelihood gradients in the parameters of a generative model P(X) [6].

For a generative parametric model defined as P(X|θ), θ a set of parameters.

The Fisher score is defined as:

$$U_X = \nabla_\theta \log P(X \mid \theta)$$

And the Fisher kernel is defined as

$$K(X_i, X_j) = U_{X_i}^T I^{-1} U_{X_j}$$

Where I is the Fisher information matrix.

III. Experimental Methodology

Building a biometric system based on a supervised machine learning classifier follows the following diagram in Fig. 3.

It is an iterative process where data is firstly transformed into a suitable data format and preprocessed, removing possible errors or noise on the samples. Feature extraction is a key step, where input raw data is transformed into the best representation vectors that will feed the classifier. For training purpose and later evaluation, the input dataset is split into two sets. Then a classifier or a combination of them is trained with the training set and tested with the evaluation set. If the result suffices the research scope the system is ready for a production environment, going for a new iteration of the process if it needs to be improved.

A. *Datasets and preprocessing*

Two databases are used for each biometric system, hands and signatures, provided by the GPDS (Digital Signal Processing Group of Las Palmas de Gran Canaria University).

1) *Handwritten signatures Database*

It is composed by 300 signers, each one with 24 genuine signatures. The scanned images are stored on BMP format on variable size and RGB (8 bits) quality. The images are black and white: a variable width signature trace in black over a white background.

The whole data set is converted to a lossless TIFF format, reducing the size around 5% of the initial one. The images are cleaned removing some noise, small groups of pixels probably introduced during the scanning process.

Fig. 4 below shows a random selection of samples for the signatures database.

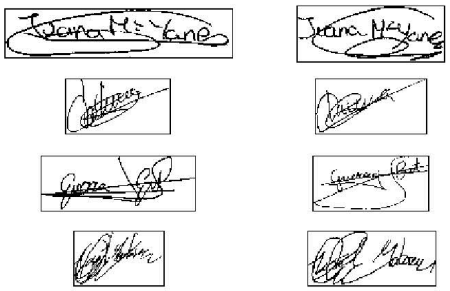

Fig. 4.Random samples from signature database

2) Hands Database

The hand database contains 10 scanned right hand images for 144 different people. The images have a resolution of 1403x1021, on 256 gray-scale and stored on JPEG format.

For the purpose of the study where only the hand contour is analyzed, the source images are binarized with a dynamic gray level selection (Otsu's method). Several images include some artifacts and noise from the binarization process is present. Hence the images are cleaned up removing relative small size objects and the remaining part of the forearm using the narrowest width of the wrist as a hand limit process. Several possible features were extracted, improved and tested to review their feasibility as final approach.

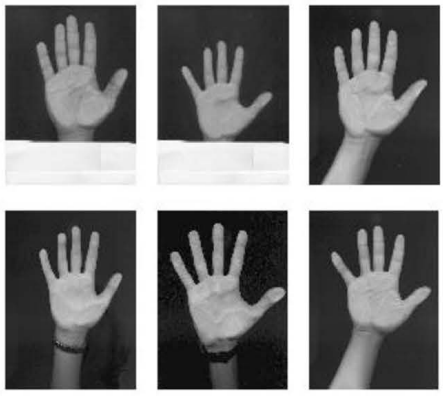

Fig. 5.Random samples from hands database

1) Features extraction for signatures.

Signatures contain a high degree of randomness on each sample that make the contour of the sample to vary significantly. Variations on the trace, that are completely normal on the signing action, produces a displaced border sequence. The approach proposed for signatures is based on border extraction at a fixed step, what makes each point non-dependent on the rest of the trace. Projecting lines from the mass center of the signature with a constant incremental θ the intersection points are collected and the farthest point is considered for the final vector.

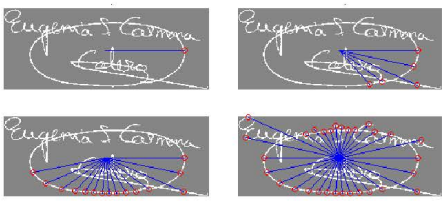

Fig. 6.Example of extraction of key points for signatures based on fixed θ .

Using the Cartesian coordinates does not provided good results. Instead angular transformations perform better. For signatures a transformation to Polar Coordinates [θ, ρ] and then to Angular Coordinates [α, β] as proposed on [8]. The Angular Coordinates follows the diagram of the II

2) Feature extraction for hands.

Hand measures have different intrinsic characteristics. There is no randomness between the samples but the differences from one hand to another are less significant than on signatures case. Therefore its features should maximize that minimal distinction.

Taking the good results from previous studies regarding the hand shape and sizes [9], the finger width and length is used as main feature. There are many ways to get the fingertips and the points between them. Polar coordinates of the hand contour provides a simple option to identify the different finger positions. Once a fingertip point is approximated and the two adjacent points for each finger, a middle line through the finger body can be traced, which gives the finger length, and then perpendicular lines proportionally separated to get the finger width along its full size.

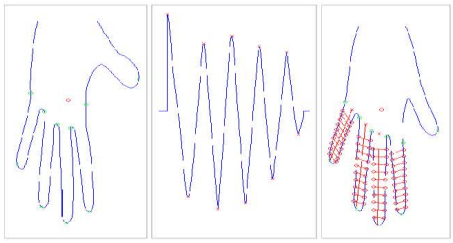

Fig. 7.Conversion to angular coordinates systems [α, β]

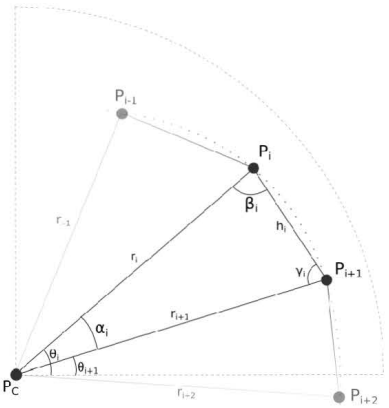

Fig. 8. Example of extraction of finger width and length. On the left the finger key points are overlaying the hand shape. On the middle the polar coordinate ρ with its maximums and minimums. On the right the width of the fingers.

II. Experiments

Initially the system for hands and signatures are evaluated independently and later on combined into a single system.

The classifiers used to evaluate the extracted features and the performance of the whole system are the ones mentioned above: HMM, SVM and HMM chained to a SVM through a Fisher transformation.

HMM and SVM classifiers are set up on the configuration of one versus the other. This implies one classifier per class that is trained to maximize the results for the given label. During evaluation phase each sample is analyzed for all the classifiers and the best result is taken as prediction.

HMM uses Forward-Backward, Viterbi and Baum-Welch algorithms on its implementation. For SVM a linear configuration is used given its satisfactory results.

For the training and evaluation process, the initial data is split into 50% training and evaluation sets. On the last experiment the train set is reduced to compare the strength of each system.

For the purpose of the combined system, the samples of each datasets are matched and labeled with the same class, so the signature database has to be reduced to match the hands' one: 144 classes and 10 samples for each.

The performance measure is done through the success rate as is common for identification systems. Also each test of the system is repeated 5 times to get unbiased results.

III. Results

As mentioned before, experiments for independent system are covered before and then the combined system.

For signatures database, the results are shown on the following table.

Table I. Results for signature dataset

Model	Result
HMM	85.82% ± 1.01
SVM	82.55% ± 1.45
HMM + SVM	100%

The combination of HMM with SVM using Fisher Kernels outperform the other methods. For HMM classifier, a scanning through states from 30 to 80 was done, and the results are similar but giving a slightly better performance for 45 states.

For hands database, the results are shown on the following table.

Table II. Results for hand dataset

Model	Result
HMM	97.52% ± 2.21
SVM	100%
HMM + SVM	100%

For hand features, results are better, getting for SVM and the combined system HMM+SVM a 100% positive identification rate.

The combination of both features is on the next table.

Table III. Results for combined features

Model	Result
HMM	92.69% ± 2.08
SVM	100%
HMM + SVM	100%

Results for the combination of both characteristics reduces the performance of the global system for HMM, but for SVM and HMM+SVM it still identifies all the possible samples during the test.

Table IV. Results reducing training data set

Train (%)	Hands SVM	Combined SVM	Combined HMM + SVM
40%	98.80% ± 0.01	99.92% ± 0.01	100 %
30%	96.23% ± 0.01	99.54% ± 0.11	99.90% ± 0.02
20%	88.11% ± 0.03	98.93% ± 0.22	99.88% ± 0.01

The combination of both features makes the system perform better while reducing the training data set.

IV. CONCLUSIONS

Results confirm that features extracted for each dataset perform slightly better than similar studies [8][9][10][11][12]. The classifier HMM+SVM gives really good results, mainly for a highly variable dataset as is the handwritten signatures.

The feature combination makes the system more robust when the training data set is reduced. In order to improve this combined result an approach of classifier assembling could be used, using the individual results of independent classifiers

The combination of different features should allow the system to scale larger keeping the good performance, a problem that may be reached earlier with a unique characteristic.

REFERENCES

[1] Burghardt T. "A Brief Review of Biometric Identification". Technical Report, Visual Information Laboratory, Bristol, 2012.

[2] Joseph N. Pato and Lynette I. Millett, Editors. "Biometric Recognition: CHALLENGES AND OPPORTUNITIES" Washington, DC: The National Academies Press. DOI:https://doi.org/10.17226/12720, 2010

[3] Rabiner, L.R. "A tutorial on hidden Markov models and selected applications inspeech recognition", Proceedings of the IEEE, Vol.77, Issue 2, pág. 257 - 286, Febrero 1989.

[4] Burges, Christopher J.C., "A Tutorial on Support Vector Machines for Pattern Recognition" Data Mining and Knowledge Discovery, Vol. 2, pág. 121 - 167, Enero 1998.

[5] Cristianini, Nello, and John Shawe-Taylor, "An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods", Cambridge, UK: Cambridge University Press. 2000

[6] Martin Sewell, "The Fisher Kernel: A Brief Review", Research Note UCL DEPARTMENT OF COMPUTER SCIENCE, 20 Enero 2011.

[7] Tommi Jaakkola and David Haussler, "Exploiting Generative Models in Discriminative Classifiers. In Advances in Neural Information Processing Systems 11, pág. 487 - 493, 1998

[8] Briceño J.C., Travieso C.M., Ferrer M.A., Alonso J.B., Vargas F. "Angular Contour Parameterization for Signature Identification". In: Moreno-Díaz R., Pichler F., Quesada-Arencibia A. (eds) Computer Aided Systems Theory - EUROCAST 2009. Lecture Notes in Computer Science, vol 5717. Springer, Berlin, Heidelberg

[9] B. Mathivanan , Dr. V. Palanisamy , Dr. S. Selvarajan, "Multi-Dimensional Hand Geometry Based Biometric Verification and Recognition System", 2012

[10] Ali Karouni, Bassam Daya, Samia Bahlak, "Offline signature recognition using neural networks approach", Procedia Computer Science, Volume 3, 2011, Pages 155-161, ISSN 1877-0509, http://dx.doi.org/10.1016/j.procs.2010.12.027.

[11] K.R. Radhika, M.K. Venkatesha and G.N. Sekhar, "Off-Line Signature Authentication Based on Moment Invariants Using Support Vector Machine", Journal of Computer Science 6 (3): 305-311, 2010.

[12] Guo, J.-M., Hsia, C.-H., Liu, Y.-F., et al.: Contact-free hand geometry-based identification system. Expert Systems with Applications 39(14), 11728–11736 (2012)

Presupuesto

PR. Presupuesto del proyecto

Se recoge en esta sección un desglose del presupuesto calculado para el presente proyecto.

Don Fernando Andrés Pitters Figueroa, autor del presente Proyecto Fin de Carrera, declara que:

El Proyecto Fin de Carrera con título “Identificador biométrico off-line basado en el contorno de la firma y forma de la palma”, desarrollado en la Escuela Superior de Ingeniería de Telecomunicaciones en la Universidad de Las Palmas de Gran Canaria, con una duración de 7 meses, tiene un coste de desarrollo total de 32.422,44 € correspondiente a la suma de las cantidades consignadas a los apartados considerados a continuación.

El autor del proyecto: Fernando Andrés Pitters Figueroa

Las Palmas de Gran Canaria a 4 de Julio de 2017

PR.1 Desglose del Presupuesto

Para la realización del presupuesto se han seguido las recomendaciones del Colegio Oficial de Ingenieros de Telecomunicación (COIT) sobre los baremos orientativos mínimos para trabajos profesionales en 2009 (dichos baremos ya no se encuentran disponibles de forma pública). El presupuesto se ha desglosado en varias secciones en las que se han separado los distintos costes asociados al desarrollo del proyecto. Estos costes se dividen en:

1. Recursos materiales.
2. Trabajo tarifado por tiempo empleado.
3. Costes de redacción del proyecto.
4. Material fungible.
5. Derechos de visado del COIT.
6. Gastos de tramitación y envío.
7. Aplicación de impuestos.

PR.2 Recursos materiales

Se enumeran en este apartado los bienes materiales requeridos para el desarrollo del proyecto acotados en: aplicaciones software para el desarrollo, equipo hardware empleado, sistemas operativos.

Se establece un coste de amortización de 4 años. Para ello, se utilizará un sistema de amortización lineal o constante, en el que se supone que el inmovilizado material se deprecia de forma constante a lo largo de su vida útil. La cuota de amortización anual se calcula usando la siguiente fórmula.

$$Coste\ Anual = \frac{Valor\ de\ adquisición - Valor\ residual}{Vida\ útil}$$

El "valor residual" es el valor teórico que se supone que tendrá el elemento después de su vida útil.

PR.2.1 Recursos software

Las herramientas software utilizadas en este proyecto son:

- Sistema Operativo Windows 10 Home
- Sistema Operativo Lubuntu
- Matlab v7.5
- LibreOffice 5.3
- Git
- Python 2.7
- Flask
- Android Studio

A continuación se procede a obtener los costes de amortización en base al tiempo de utilización. Se supone un coste residual tras los 4 años de 0 euros. Además se omiten todas las herramientas utilizadas con licencia Open Source y sin coste, compatibles en todos los casos con un modelo comercial.

Herramienta software	Coste Total	Valor amortización (7 meses)
Matlab v7.5	3000 €	437,50 €
Windows 10 Home	135 €	19,69 €
Coste acumulado		**457,19 €**

Tabla 1: Recursos software y su coste amortizado

El coste total de los recursos software es de cuatrocientos cincuenta y siete euros y diecinueve céntimos (457.19 €).

PR.2.2 Recursos Hardware

Los recursos hardware utilizados en el desarrollo del proyecto son:

- Ordenador portátil HP: procesador Intel Core i3, 3GB de RAM y 320 GB de disco duro.
- Ordenador portátil Dell: procesador Intel Core i7, 8GB de RAM y 256 GB de disco duro SSD.

Herramienta hardware	Coste Total	Valor residual (4 años)	Valor amortización (7 meses)
Ordenador HP	899 €	150 €	109,22 €
Ordenador Dell	1399 €	250 €	167,59 €
Coste acumulado			**276,81 €**

Tabla 2: Recursos hardware y su coste amortizado

El coste total de los recursos hardware es de doscientos setenta y seis euros y ochenta y un céntimos (276,81 €).

Los recursos materiales totales tienen un coste de setecientos treinta y cuatro euros (734,00 €).

PR.3 Trabajo tarifado por tiempo empleado

Se han invertido 7 meses durante diferentes fases de formación, desarrollo y documentación para poder cumplimentar el presente proyecto. El importe de las horas de trabajo empleadas para la realización del proyecto se calcula siguiendo las recomendaciones del COIT:

$$H = C_t \cdot 74{,}88 \cdot H_n + C_t \cdot 96{,}72 \cdot H_e \text{ €}$$

Donde:

- H son los honorarios totales por el tiempo dedicado.
- H_n son las horas normales trabajadas (dentro de la jornada laboral)
- He son las horas especiales.
- C_t es un factor de corrección función del número de horas trabajadas.

El tiempo destinado a la formación y a la documentación del proyecto no se incluyen dentro de la estimación pues no constituyen parte del desarrollo propio y se trata de un coste añadido al de un desarrollo normal.

Los tiempos invertidos en el proyecto se dividen en los siguientes tramos.

Tareas	Tiempo invertido
Formación	3 semanas (120 horas)
Documentación	4 semanas (160 horas)
Desarrollo	5 meses y una semana (840 horas)

Tabla 3: Horas invertidas por tarea

Por tanto el tiempo de realización del presente proyecto es de 840 horas, dentro de un horario de trabajo normal en días laborables.

De acuerdo a las recomendaciones del COIT, el coeficiente C_t tiene un valor variable en función del número de horas empleadas de acuerdo con la siguiente tabla.

Horas ejecutadas	Factor de corrección C_t
Hasta 36 horas	1,00
De 36 a 72 horas	0,90
De 72 a 108 horas	0,80
De 108 a 144 horas	0,70
De 144 a 180 horas	0,65
De 180 a 360 horas	0,60
De 360 a 540 horas	0,55
De 540 a 720 horas	0,50
De 270 a 1080 horas	0,45
Más de 1080 horas	0,40

Tabla 4: Factor de corrección en función de las horas empleadas

Para el tiempo invertido total de 840 horas, el coeficiente de corrección es de C_t = 0.45, lo que da un cálculo de honorarios totales de veintiocho mil trescientos cuatro euros y sesenta y cuatro céntimos (28.304,64 €).

PR.4 Material fungible

Adicionalmente a los recursos software y hardware se incluyen a continuación una serie de bienes fungibles consumidos durante el curso del presente proyecto.

Material fungible	Valor amortización (7 meses)
Folios	10 €
Tinta impresora	50 €
Encuardernación	15 €
Coste acumulado	**75 €**

Tabla 5: Desglose de materiales fungibles y su coste

Hace un total de material fungible de setenta y cinco euros (75 €).

PR.5 Costes de redacción del proyecto

El importe de la redacción del proyecto se calcula de acuerdo a la siguiente expresión:

$$R = 0.07 \cdot P \cdot C_h$$

Donde:

- P es el presupuesto base del proyecto.
- C_h es el coeficiente de ponderación en función del presupuesto.

El presupuesto base P calculado es de 29.113,64 € como se recoge a continuación.

Definición	Coste
Recursos materiales	734,00 €
Trabajo tarificado por tiempo	28.304,64 €
Material fungible	75 €
Coste acumulado	**29.113,64 €**

Tabla 6: Coste acumulado del presupuesto para cálculo de costes de redacción

Se establece un coeficiente de ponderación de 0.5.

El coste de redacción sin impuestos añadidos es de mil dieciocho euros y noventa y ocho céntimos (1.018,98 €).

PR.6 Derechos de visados del COIT

Los gastos de visado del COIT se establecen mediante la siguiente expresión:

$$V = 0.006 \cdot P \cdot C_v$$

Donde:

- P es el presupuesto del proyecto.
- C_v es el coeficiente reductor en función del presupuesto del proyecto.

El presupuesto P calculado hasta el momento asciende a la suma de los costes acumulados de recursos materiales, material fungible, desarrollo y redacción.

Tiene un valor de:

$$P = 29.113,64\,€ + 1.018,98\,€ = 30.132,62\,€$$

Como el coeficiente C_v para presupuestos de más de 30.050 € y menos de 60.101 €, viene definido por el COIT con un valor de 0,90, el coste de los derechos de visado del proyecto asciende a ciento sesenta y dos euros y setenta y dos céntimos (162,72 €).

PR.7 Gastos de tramitación y envío

Los gastos de tramitación y envío están fijados en seis euros y un céntimo (6,01 €).

PR.8 Aplicación de impuestos

El coste total del proyecto libre de impuestos es de 30.295,34 €. A continuación se desglosa el coste en base a los grupos aplicados y se añade el 7% de IGIC.

Definición	Coste
Recursos materiales	734,00 €
Trabajo tarificado por tiempo	28.304,64 €
Material fungible	75 €
Coste de redacción	1.018,98 €
Derechos de visado	162,72 €
Gastos de tramitación y envío	6,01 €
Subtotal	**30.301,35 €**
7% IGIC	2.121,09 €
Total	**32.422,44 €**

Tabla 7: Desglose presupuesto total

El presupuesto total es de treinta y dos mil cuatrocientos veintidós euros y cuarenta y cuatro céntimos (32.422,44 €).