

Reconocimiento de firmas en dispositivos móviles

Bruno Strasser Marcelo Ortega

6 de diciembre de 2016

1. Introducción

Existen diferentes métodos biométricos utilizados para la identificación y verificación de la identidad de las personas, como las firmas, las huellas dactilares, el reconocimiento facial o de iris, etc. Entre todos estos, el reconocimiento basado en firmas tiene la ventaja de no requerir de la instalación de costosos equipamientos como escáners y de ser utilizado comúnmente: todos los individuos están acostumbrados al uso de las firmas.

El avance de la tecnología móvil hace que cada día sean más las cosas que se pueden realizar desde el dispositivo celular: mensajería, transferencias bancarias, y hasta control remoto de los dispositivos del hogar son algunas de sus posibilidades. Sin embargo, los métodos de identificación del usuario siguen estando mayoritariamente limitados al uso de un nombre de usuario y una contraseña privada. Los modestos avances en incorporar otras tecnologías de identificación están ocurriendo principalmente en el área de reconocimiento facial, para la cual algunos prototipos se están comenzado a implementar principalmente en el sector bancario. La identificación basada en firmas parece haber quedado en el pasado y no parece que se esté adaptando a la progresiva sustitución del papel por formatos digitales de documentos.

Este trabajo se propone una investigación inicial de las posibilidades que ofrece el reconocimiento de firmas en dispositivos móviles, es decir una sistema de identificación en el que el usuario traza en la pantalla táctil de su dispositivo móvil un trazo que identifica como su firma. Un sistema de estas características ofrece facilidad de uso, no necesita de hardware especial como cámaras ni sensores, y por sobretodo es conocido y confiado por las personas.

2. Solución propuesta

2.1. Recolección de datos

Si bien existen algunas bases de datos públicas de firmas, tanto online como offline, todas ellas son de firmas capturadas mediante dispositivos electrónicos como PDAs. Estos métodos de ingreso son diferentes que el que proponemos utilizar en este trabajo, y las firmas producidas en uno u otro tendrán características diferentes. Principalmente, la pantalla táctil de un celular tiene una sensibilidad mucho menor y una superficie mas reducida, por lo que los trazos serán mas pequeños.

Ante esta incompatibilidad de estas bases de datos existentes con la propuesta de nuestro trabajo, producimos una base de datos propia, recogiendo firmas de distintas personas. Las firmas fueron capturadas mediante una aplicación web basada en la librería Signature Pad, la cual permite el ingreso del trazo de la firma y su almacenamiento como lista de puntos.

En la recolección de firmas se le solicitó a 30 personas que repitan su firma 30 veces. Como se dijo anteriormente, las diferencias entre realizar un trazo en la pantalla táctil y en el papel provocaron que la mayoría de los usuarios no pudiera realizar su firma habitual, utilizando en lugar de ella un trazo más pequeño, generalmente la rúbrica de la persona. Contamos con una base de firmas si bien pequeña bastante heterogénea entonces, de firmas capturadas en distintos dispositivos móviles por distintas personas.

2.2. Metodología

El objetivo de este trabajo será entonces evaluar el rendimiento de distintas técnicas de reconocimiento de patrones aplicadas al problema de reconocer firmas capturadas en dispositivos móviles. Para esto, realizamos una partición inicial de nuestra base de datos de firmas recolectadas en dos conjuntos: uno de entrenamiento y otro de prueba. El primero será el conjunto de firmas con el que se realizarán los experimentos con distintos clasificadores y el ajuste de sus correspondientes parámetros para optimizar su desempeño. Una vez concluidos estos experimentos, se procede a evaluar los clasificadores obtenidos en el conjunto de pruebas.

El conjunto de entrenamiento cuenta con 25 usuarios y 20 firmas para cada uno, y el de prueba con 10 nuevas firmas no observadas anteriormente para estos usuarios, y 5 usuarios nuevos con 30 firmas cada uno. No investigaremos en este trabajo la posibilidad de ingresar nuevos usuarios en el sistema, por lo que se buscará que las firmas de los usuarios nuevos sean rechazadas por no estar asociadas a ningún usuario registrado. Esta partición inicial de la base de datos de firmas se realiza una sola vez antes de comenzar con los experimentos, y las firmas de prueba no son utilizadas hasta la fase final de pruebas.

Para la parte experimental, en la que evaluaremos diferentes algoritmos de reconocimiento de patrones y sus resultados, separamos la base de entrenamiento en dos subconjuntos: un conjunto de entrenamiento y uno de validación. De los 25 usuarios y sus respectivas 20 firmas que componen el antes descrito conjunto de entrenamiento, éste queda dividido nuevamente en un conjunto de entrenamiento de 15 firmas para 20 usuarios, y otro de validación con 5 firmas adicionales para estos usuarios y 5 usuarios nuevos con 20 firmas cada uno.

A diferencia de la separación inicial entre las firmas de entrenamiento y las de pruebas, la selección de los usuarios y firmas que compondrán el conjunto de validación es aleatoria y se realiza una nueva para cada experimento. De esta forma, realizando sucesivas repeticiones de un experimento logramos simular el efecto que diferentes nuevas firmas y usuarios podrían tener en él, sin tener que recurrir al conjunto de pruebas que reservaremos para la evaluación final de los resultados.

3. Preprocesamiento de datos

La lista de puntos de la firma j del individuo i la denominaremos como $A_j^i = (a_1^i, a_2^i, \dots, a_{n_{ij}}^i)$ en donde cada $a_k^i = (x_k^i, y_k^i, t_k^i)$ cuyas x_k^i e y_k^i corresponden a las coordenadas del punto en un plano bidimensional, mientras que t_k^i es una marca de tiempo con una precisión de milisegundos (timestamp). A modo de simplificar el problema asumimos que los puntos se encuentran ordenados de forma creciente en función del tiempo t_k^i , es decir, se cumple que si $k < m$ entonces $t_k^i < t_m^i$ para todo i .

El preprocesamiento de los datos fue una de las partes clave del trabajo y en la cuál se identifican varios problemas para resolver. Con el fin de simplificar la notación denominaremos $z_k^i = (x_k^i, y_k^i)$.

3.1. Rotación

Uno de los problemas que nos encontramos es que un individuo puede firmar tomando el dispositivo móvil desde diferentes ángulos produciendo que un mismo garabato se encuentre rotado en diferentes ángulos. Para solucionar este problema, se aplicó una transformación sobre cada firma que consiste en determinar los puntos más alejados de una firma y rotar todos los puntos el ángulo formado por estos puntos. Para realizar esto, se realizó la siguiente transformación:

$$z_k^i := z_k^i * R(\Theta_j^i) \quad \forall k \in \{1, \dots, n_{ij}\}$$
$$R(\Theta) = \begin{vmatrix} \cos(\Theta) & -\sin(\Theta) \\ \sin(\Theta) & \cos(\Theta) \end{vmatrix}$$

En donde Θ_j^i corresponde al ángulo formado por los puntos más alejados y $R(\Theta_j^i)$ la matriz de rotación correspondiente.

3.2. Centrado

Es usual en dispositivos con táctiles con pantallas grandes, que un mismo individuo firme en diferentes lugares de la pantalla, por lo que para que corregir esto, es necesario aplicar un centrado de los datos. Para esto se calculó para cada firma el centro utilizando la fórmula del promedio:

$$C_j^i = \frac{1}{n_{ij}} * \sum_{h=1}^{n_{ij}} z_h^i$$

Una vez obtenido el centro de cada firma se traslada la firma hacia el origen calculando para cada punto de cada firma:

$$z_k^i := z_k^i - C_j^i \quad \forall k \in \{1, \dots, n_{ij}\}$$

3.3. Datos derivados

Una vez que las firmas se encuentran rotadas y centradas, se procede a calcular los datos adicionales correspondiente al porcentaje de avance hasta el punto, porcentaje de tiempo hasta el punto y el ángulo relativo entre puntos (sus vectores) sucesivos.

Para calcular el porcentaje de avance y el tiempo hasta de cada punto, es necesario calcular el largo total D_j^i de la firma y el tiempo total T_j^i de la firma. El largo total de la firma corresponde a la suma de las distancias entre puntos sucesivos dada por la siguiente formula:

$$D_j^i = \sum_{h=1}^{n_{ij}-1} \|x_{h+1}^i - x_h^i\|$$

En cuanto al tiempo total (T_j^i) simplemente alcanza con restar las marcas de tiempo del primer punto con el último, por lo que

$$T_j^i = t_{n_{ij}}^i - t_1^i$$

Antes de hacer las próximas transformaciones, aplicaremos un pequeño ajuste a los valores de t_k^i de la siguiente forma

$$t_k^i := t_k^i - t_1^i \quad \forall k \in \{1, \dots, n_{ij}\}$$

Obtenidos estos valores, el cálculo del los porcentaje de avance (ρ_k^i), el porcentaje de tiempo (w_k^i) y el ángulo relativo (ϕ_k^i) en cada punto se calculan de la siguiente manera:

$$\rho_k^i = \begin{cases} 0 & \text{si } k = 1 \\ \rho_{k-1}^i + \frac{\|z_k^i - z_{k-1}^i\|}{D_j^i} & \text{si } k > 1 \end{cases}$$

$$w_k^i = \begin{cases} 0 & \text{si } k = 1 \\ w_{k-1}^i + \frac{t_k^i - t_1^i}{T_j^i} & \text{si } k > 1 \end{cases}$$

$$\phi_k^i = \begin{cases} 0 & \text{si } k = 1 \\ \arccos\left(\frac{x_{k-1}^i \cdot x_k^i - y_{k-1}^i \cdot y_k^i}{\|z_{k-1}^i\| \cdot \|z_k^i\|}\right) & \text{si } k > 1 \end{cases}$$

3.4. Interpolación

Hasta el momento, tenemos calculado para cada punto $a_k^i = (x_k^i, y_k^i, t_k^i, \rho_k^i, w_k^i, \phi_k^i)$ con x_k^i e y_k^i centrados y rotados. El último problema que debemos resolver es que necesitamos que todas las firmas de todos los individuos tengan un largo fijo. Para esto aplicamos para firma A_j^i una interpolación lineal sobre la lista de puntos a_k^i . Esta interpolación se hace tomando como dimensión de dominio ρ y se obtiene para cada firma $A_j^i = (a_1^i, \dots, a_N^i)$ de largo fijo $N = 75$ en donde $a_k^i = (x_k^i, y_k^i, t_k^i, w_k^i, \phi_k^i)$. Debido a que la dimensión de dominio elegida es ρ el resto de las dimensiones de a_k^i se encuentran en función de ρ , por lo que no es incluida en a_k^i .

4. Datos experimentales

5. Extracción de características

En esta primera serie de experimentaciones decidimos trabajar con un subconjunto de las características con las que contamos para describir cada firma

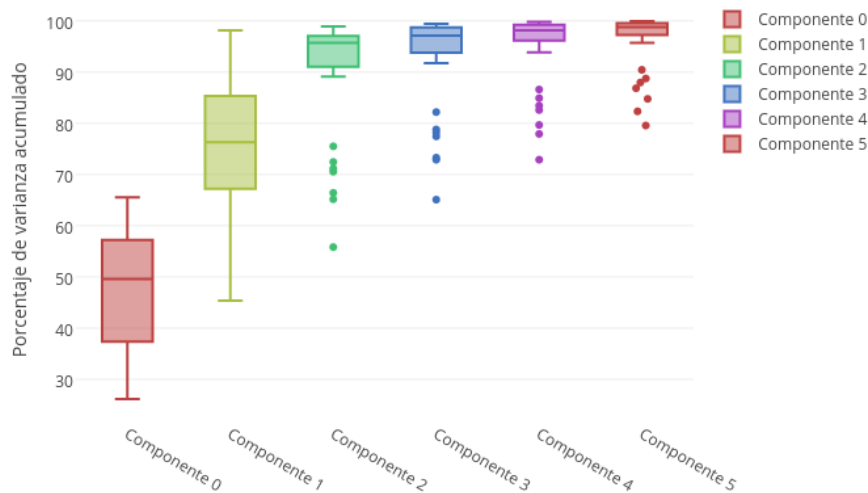


Figura 1: Diagrama de cajas de la varianza acumulada para los primeros 5 componentes principales obtenidos mediante PCA. La línea mayor y menor corresponden al valor máximo y mínimo respectivamente, y la caja abarca los puntos en el segundo y tercer cuartil, separados en el medio por la mediana. Estos resultados fueron obtenidos para 30 ejecuciones.

extraídas durante el proceso de normalización. Con el fin de reducir la dimensionalidad inicial, decidimos representar cada firma como la información de las coordenadas de sus puntos. Esto corresponde a un vector de largo 150, donde los primeros 75 elementos corresponden a la coordenada x que tuvo el trazo a lo largo de su recorrido, normalizado con respecto a la distancia total recorrida (el primer elemento corresponderá al primer punto, y el del medio al punto ubicado en la mitad del largo total del trazo). De la misma manera, los últimos 75 elementos corresponden a las coordenadas y del trazo de la firma.

Con la extracción de características pretendemos reducir aún más la dimensionalidad de nuestro problema, con el fin de hacer mas eficientes (tanto en precisión como en tiempo de cómputo) los clasificadores a utilizar. De las técnicas de extracción de características disponibles, en este trabajo utilizaremos PCA, aunque en otras publicaciones se utilizan también otros métodos como Wavelet Extraction.

Utilizando PCA obtendremos un conjunto de nuevas características con las que representar las firmas, en un espacio que maximice la varianza.

Para determinar el número de componentes principales óptimo a utilizar en esta transformación, encontramos cuanta varianza captura cada uno de estos. Para esto realizamos 30 ejecuciones, en cada una se divide al azar el conjunto de entrenamiento del de validación y se calcula la varianza acumulada por com-

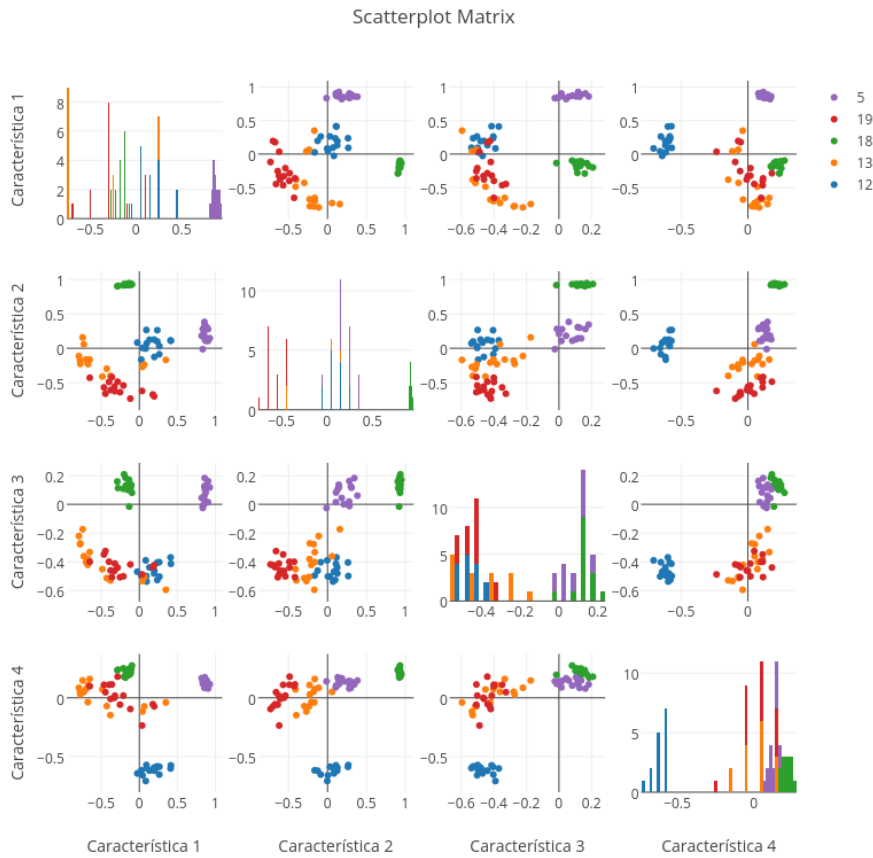


Figura 2: Matriz de diagramas de puntos para 5 firmas, comparadas según los cuatro componentes principales extraídos mediante PCA

ponente. Es de esperar que la varianza por componente sea diferente según las firmas que compongan el conjunto sobre el cual se entrenará la transformación PCA; realizando múltiples simulaciones de conjuntos iniciales pretendemos tener una visión estadística que nos permita elegir una extracción de componentes que minimice la varianza, que sea robusta a las posibles diferentes configuraciones iniciales y por lo tanto a firmas desconocidas que puedan ser incorporadas. El diagrama de cajas de la Figura 5 muestra los resultados obtenidos para estas ejecuciones, se ve allí cómo hay una gran varianza en la varianza acumulada para los dos primeros componentes, pero esta varianza se disminuye considerablemente a partir del cuarto. Basados en esto, decidimos elegir una extracción PCA de los primeros cuatro componentes principales, que capturan más de un 90% de varianza (a excepción de algunos experimentos que registraron outliers).

Con esta extracción de características logramos reducir considerablemente la dimensionalidad de nuestro problema, bajando de las 150 características iniciales a tan solo 4 extraídas mediante PCA. En la Figura 5 se muestra la matriz de

diagramas de puntos para 5 firmas comparadas según estas cuatro características. Aunque es un conjunto de firmas muy pequeño para extraer conclusiones, las clases parecen ser linealmente separables entre sí en este espacio, por lo que es de esperar que un clasificador pueda lograr una gran efectividad en la tarea.

6. Clasificación

Para nuestro problema de reconocimiento de firmas, la tarea de clasificación consiste en poder asignar a cualquier firma el usuario del sistema a cual pertenece, o rechazar la autenticación si la firma ingresada no es similar a la firma recordada para ningún usuario. Para esto, el clasificador es entrenado con un conjunto de usuarios, cada uno de los cuales será una clase, y sus firmas las instancias que componen esa clase. El caso del rechazo se modela como una clase más, la clase 'Rechazo', con la cual se clasificará las firmas que no correspondan a ningún usuario, y el sistema negará así la autenticación.

Las clases correspondientes a cada usuario pueden ser descriptas por sus instancias antes observadas (las firmas presentes en el conjunto de entrenamiento). Sin embargo, no sucede lo mismo para la clase Rechazo, que si bien se puede definir como el conjunto de firmas que no corresponden a ningún usuario, representar todas estas firmas para poder entrenar al clasificador a reconocerlas no resulta práctico ni conveniente. Volviendo a la Figura 5, esto sería el equivalente a representar una cantidad de firmas tal que su representación abarque todos los espacios donde no hay instancias de las firmas de los cinco usuarios.

Incluso si se tuviera la capacidad de representar una a una todas estas firmas que se quiere rechazar, se necesitaría determinar de una vez y de forma totalmente arbitraria en que punto las diferencias entre la firma ingresada y las registradas para un usuario son tales que ya no se consideran ingresos satisfactorios sino que son rechazadas. Lo conveniente sería en todo caso poder variar este umbral de exigencia en el reconocimiento, y determinarlo en base a que tan bien se desempeñe en los datos experimentales.

Por esto, si bien como se vió en la Figura 5 las clases parecen ser linealmente separables y un clasificador convencional lograría un alto porcentaje de acierto, no podría lidiar de forma eficiente con el problema del rechazo de firmas.

En las siguientes secciones se presentarán dos soluciones posibles al problema de la implementación de un clasificador de firmas. En primer lugar se presentará una clasificación por vecinos más cercanos (3NN), que por ser un algoritmo intuitivo, de sencilla implementación y que logra resultados razonables, servirá de línea base contra la cual comparar los resultados obtenidos por el otro clasificador presentado a continuación, una implementación con redes neuronales.

Como se mencionó anteriormente, la autenticación o el rechazo de una firma dependerá de el umbral de exigencia que se maneje. Cuanto más bajo este umbral, más diferencias entre la firma ingresada y las anteriormente registradas se permitirá, lo cual resultará en firmas erróneamente clasificadas y mayor vulnerabilidad a las falsificaciones (no cubriremos este punto en este trabajo). Por otro lado, si el umbral es alto, el sistema ofrecerá mayor seguridad, pero será menos capaz que lidiar con la variabilidad intra-usuario, pudiendo rechazar las firmas de un usuario aunque sean verídicas.

Los resultados obtenidos por los clasificadores dependerán siempre de cómo se defina este umbral, por lo que para poder comparar entre ellos describiremos

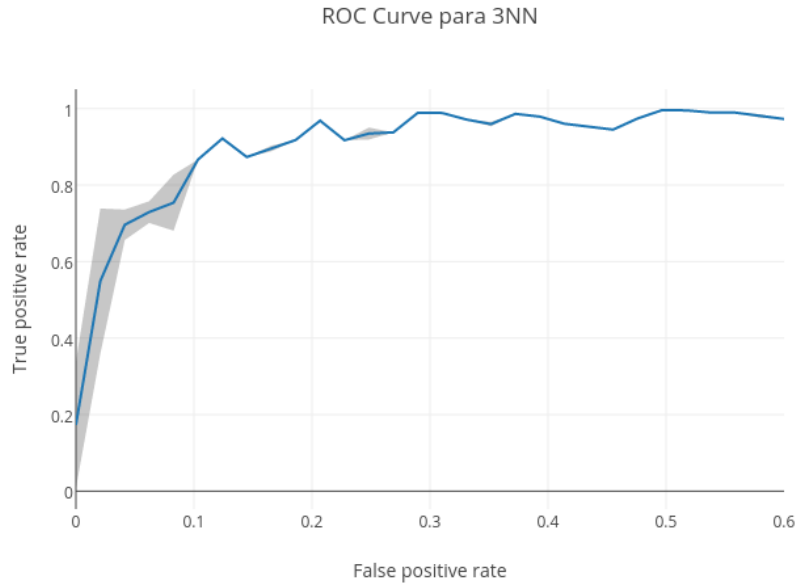


Figura 3: Curva ROC obtenida mediante la ejecución en distintos conjuntos de validación. El sombreado indica la varianza estadística que hubo entre las 30 ejecuciones.

su desempeño en función de sus posibles valores. Utilizando una curva ROC, podremos ver como se desempeña cada clasificador conforme varía el valor del umbral. Tanto para calcular esta curva ROC como en otras menciones, definiremos los valores de TP (verdadero positivo), FP (falso positivo), TN (verdadero negativo) y TP (verdadero positivo) de la siguiente manera:

TP = Firmas clasificadas correctamente como pertenecientes a un usuario.

FP = Firmas asignadas a un usuario pero que deberían de haber sido rechazadas o asignadas

TN = Firmas correctamente rechazadas.

FN = Firmas rechazadas pero que deberían de haber sido asignadas a un usuario.

6.1. Clasificación por 3NN

Para resolver el problema del rechazo, se implementa una versión de 3NN modificada. Para clasificar una instancia, se mide primero la distancia euclideana a sus tres vecinos más cercanos. Si para los tres la distancia es mayor a el umbral de exigencia, la firma es rechazada. En caso contrario, la clase asignada se decide en base a sus tres vecinos mas cercanos, ponderando cada uno por el inverso de la distancia que los separa.

En la Figura6.1 se muestra la curva ROC obtenida evaluando distintos valores del umbral en la partición de validación, repitiendo el experimento 30 veces para cada valor. La variación estadística del valor en cada punto es representada con una sombra gris, y si bien varía mucho para los valores de FPR mas

cercanos al cero, los resultados se estabilizan a partir de aproximadamente 0.15 de FPR, donde se obtiene 0.9 de TPR.

6.2. Clasificación por MLP

La existencia de la clase de rechazo y la imposibilidad de definirla mediante instancias hace que el problema no pueda ser resuelto mediante un clasificador implementado con un Perceptrón Multicapa. En su lugar, la clasificación por MLP es realizada mediante el uso de un Regresor. A cada clase se le asigna un número, y se entrena el Regresor con las instancias de entrenamiento y para cada una de ellas el número asociado a la clase objetivo. Mediante backpropagation, el Regresor logra converger a una función que pueda dar valores tan cercanos como se quiera a estos números. El MLP tiene 4 nodos de entrada, correspondientes a los cuatro componentes extraídos mediante PCA, y un nodo de salida que devuelve el valor resultado de la regresión.

Para clasificar una firma, se toma la salida del Regresor MLP y se busca la clase cuyo valor más se aproxima a esta salida. Si la salida no se aproxima menos que el umbral de tolerancia a ninguna clase existente, entonces la firma es rechazada. El valor de salida del regresor se interpreta entonces como la confianza en que la nueva instancia observada pertenezca a una clase.

En la Figura 6.2, 6.2 y 6.2 se muestran las curvas ROC para ejecuciones con MLPs de dos capas de 100-50 nodos (una primer capa de 100 y otra de 50), 50-25 y 25-15 respectivamente. No significa una diferencia de desempeño significativa conforme aumenta el número de nodos en la capa oculta, si bien en las ejecuciones con capa oculta 100-50 la varianza entre diferentes ejecuciones es menor que en los otros casos.

Esta implementación en base a un regresor MLP tiene un desempeño mucho peor que la clasificación mediante vecinos más cercanos. Para 3NN, se lograba un 0.9 de ratio de verdaderos positivos (TPR) para un ratio de 0.1 falsos positivos (FPR). Con el regresor MLP, obtener un ratio similar requeriría 0.8 de FPR, una capacidad de rechazar correctamente mucho menor.

6.3. Clasificación por MLP individual

Una posible forma de mejorar los resultados obtenidos con el uso de un Regresor MLP para el problema de clasificación de firmas es mediante el uso de múltiples clasificadores, de forma que cada uno esté especializado en reconocer las formas de un usuario en particular. Si la imposibilidad de brindar suficientes instancias de la clase Rechazo son el impedimento para utilizar un clasificador único, tal vez se obtengan mejores resultados con múltiples clasificadores, a los cuales será más fácil entrenar para reconocer la clase Rechazo, que corresponderá a todas las firmas que se tenga de otros usuarios que no sean el correspondiente al clasificador en cuestión.

Se utilizan para la implementación entonces K clasificadores MLP diferentes, cada uno con una capa de entrada de cuatro nodos (correspondientes a los cuatro componentes principales extraídos mediante PCA) y una capa de salida de un nodo. A cada red neuronal se la entrena para asociar las firmas de un usuario con la clase "Aceptación", y a las firmas de otros usuarios con la clase "Rechazo". Una vez entrenados, para evaluar una firma se la valida contra todos

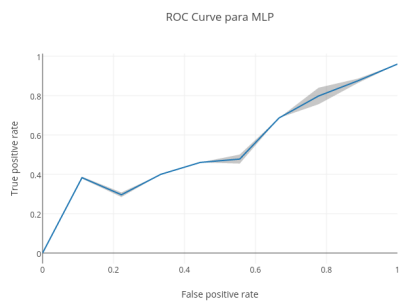


Figura 4: Curva ROC para MLP con capa oculta de 100-50

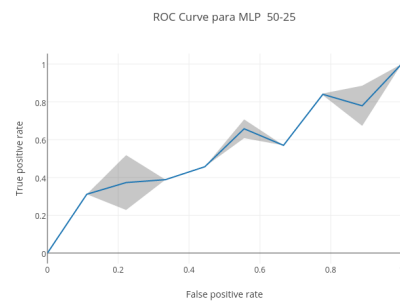


Figura 5: Curva ROC para MLP con capa oculta de 50-25

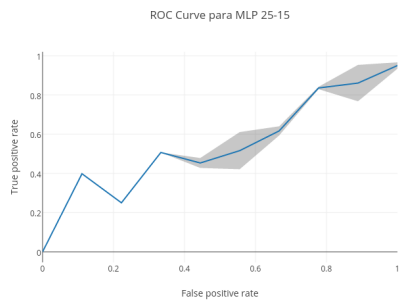


Figura 6: Curva ROC para MLP con capa oculta de 25-15

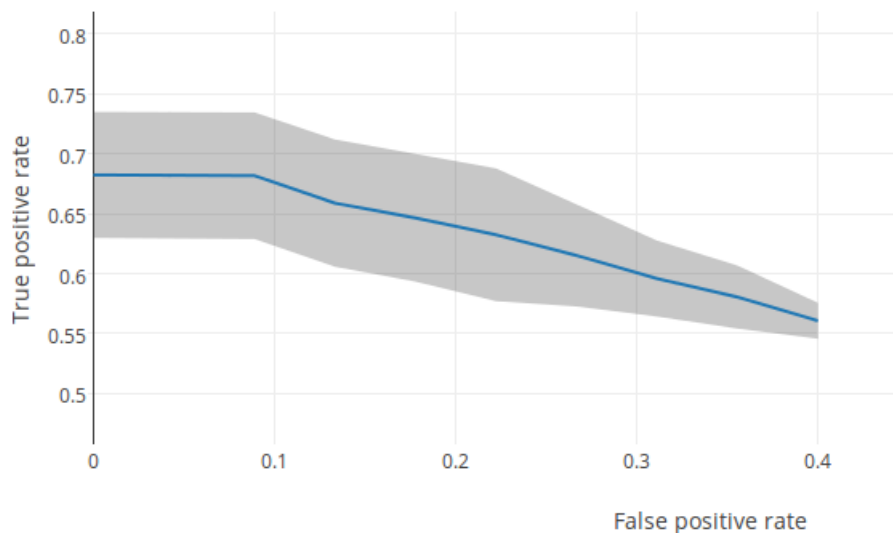


Figura 7: Curva ROC obtenida cuando se utiliza diferentes clasificadores cada uno especializado en reconocer las firmas de un usuario en particular.

los clasificadores existentes. Si solo se obtiene la clase "Aceptación" para un clasificador, entonces se la autentifica como perteneciente a su usuario asociado. Si todos los clasificadores devuelven Rechazo, si la firma es aceptada por más de uno, entonces es Rechazada".

Para evaluar el rendimiento de esta implementación se realizaron 100 ejecuciones, en cada una se realiza una partición aleatoria del conjunto de validación sobre el cual se evalúa el clasificador, y se registran los valores del FPR y TPR obtenidos. En la Figura 6.3 se grafican estos valores. Para estas ejecuciones se utilizó en los clasificadores dos capas ocultas de 20 y 10 nodos respectivamente, aunque similares resultados se obtuvieron en pruebas con más o menos nodos. La forma de la curva obtenida no corresponde a la de una curva ROC, el aumento del FPR no se corresponde con un aumento del TPR. La implementación tampoco cuenta con un umbral de exigencia al reconocimiento que se pueda variar, sino que la variación de los TPR y FPR obtenidos en las diferentes ejecuciones es el producto de diferentes conjuntos de validación en los que se obtenían mejores o peores resultados, que resultaban a los clasificadores más o menos "difíciles" de reconocer según el entrenamiento previo. Por lo tanto, no existe un umbral de rechazo que suponga una compensación entre FPR y TPR, la variación de estos factores está determinada por la dificultad intrínseca al problema, diferente para cada partición de entrenamiento/validación y por lo tanto diferente en cada ejecución.

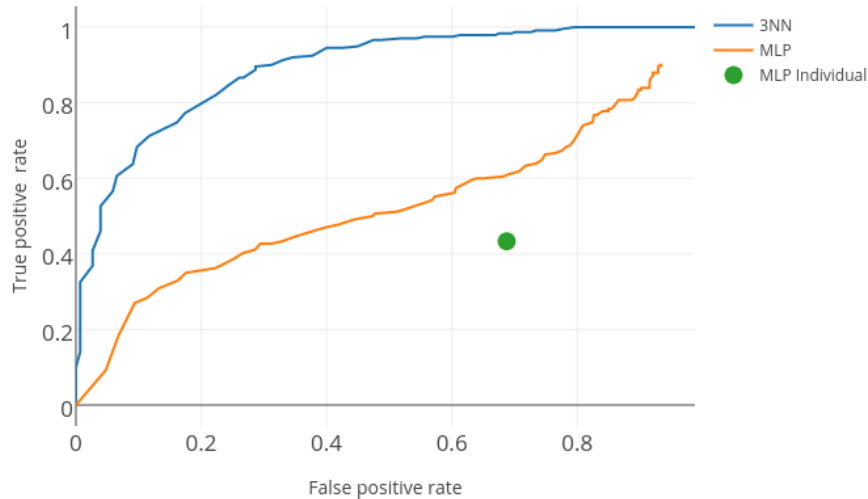


Figura 8: Curva ROC obtenida por cada clasificador en el conjunto de pruebas. El MLP individual corresponde solamente al punto verde, ya que su desempeño no puede ser variado mediante un umbral de tolerancia.

6.3.1. Evaluación final

En esta etapa final se procedió a validar el comportamiento esperado para los clasificadores implementados, y comprobar que las observaciones sobre su desempeño obtenidas en las ejecuciones de pruebas mantienen su validez en el conjunto de pruebas final.

Los resultados finales se muestran en la Figura 6.3.1. Se entrenó cada uno de los clasificadores en la totalidad del conjunto de entrenamiento (es decir, sin realizar una partición de validación) y se evaluó su desempeño en el conjunto de pruebas. A diferencia de los experimentos anteriores, en los que al realizar múltiples ejecuciones con distintas particiones de validación aleatorias los resultados en cada una variaban ligeramente, los resultados para estas pruebas finales no tienen varianza estadística, ya que el desempeño del clasificador está determinado por el conjunto de entrenamiento y el de prueba, que son siempre los mismos.

Estos resultados de prueba ciertamente comprueban las observaciones realizadas en la fase experimental, y el desempeño de la implementación con Vecinos Mas Cercanos es mejor a la obtenida con el Regresor MLP. Para cualquier FPR aceptado, 3NN ofrece siempre un TPR mayor que MLP, lo que lo hace preferible en todos los casos. El desempeño de la implementación mediante una red neuronal experta en cada usuario es aún peor, ofreciendo el menor TPR para un FPR de aprox. 0.7.

7. Conclusión

Se presentaron tres técnicas de reconocimiento de patrones diferentes para la clasificación de firmas en dispositivos móviles. Una basada en Vecinos Cercanos, que clasifica una instancia según sus 3 vecinos mas cercanos ponderados por distancia, o la rechaza si la distancia que la separa de sus vecinos es mayor que la especificada en un umbral de tolerancia arbitrario.

Las otras implementaciones propuestas se basaban en el aprendizaje automático mediante redes neuronales Perceptrón Multicapa. La segunda implementación utiliza un Regresor MLP que se entrena asignando a todas las firmas de un mismo usuario el mismo número. A la hora de clasificar una firma, se le asigna la clase de usuario cuyo número se aproxime lo suficiente según un umbral de tolerancia al número devuelto por el Regresor, o se rechaza la firma si no existe ninguna clase que cumpla esta condición.

La tercer implementación también utiliza redes neuronales Perceptrón multicapa, pero esta vez como un clasificador. Se cuenta con un clasificador MLP por cada usuario, el cual se entrena para aceptar las firmas que le pertenecen y para rechazar todas las que se tienen para los otros usuarios. A la hora de clasificar una nueva firma, se consulta a todos los MLP de los usuarios existentes. Si solo uno acepta a la firma, entonces se autentifica como ese usuario. Si no hay un consenso y más de un clasificador acepta a la nueva instancia o si todos la rechazan, entonces la firma es rechazada.

Tanto en los resultados obtenidos en la fase experimental como en las pruebas finales, el desempeño de la implementación basada en el clasificador por Vecinos más Cercanos fue estrictamente superior a la obtenida por las otras implementaciones. Para cualquier umbral de tolerancia que se defina, el clasificador 3NN siempre obtendrá mejores resultados, reconociendo correctamente una mayor cantidad de usuarios y rechazando incorrectamente una menor cantidad de usuarios que los otros clasificadores.

Los buenos resultados obtenidos por el clasificador 3NN se pueden explicar por sus propias características. La clasificación por vecinos más cercanos se basa en almacenar todas las instancias observadas durante el entrenamiento, y a la hora de clasificar una nueva instancia, se compara su grado de similitud con las ya observadas. Si se asemeja lo suficiente (calculando este grado de similitud según la distancia que las separa en el espacio de su representación) a las instancias ya observadas para alguna clase, entonces se la clasifica con esa clase. De esta forma, se cuenta para la clasificación con toda la información recogida durante el entrenamiento.

Por otro lado la red neuronal almacena la información recogida durante el entrenamiento en los coeficientes de los nodos de sus capas. Al no poder simplemente mantener en memoria todas las observaciones anteriores y tener que comprimir en sus nodos la información que se pueda extraer de ellas para tener éxito en predecirlas correctamente, la red neuronal logra aprender conceptos relativos al problema en sí.

De los resultados obtenidos podemos concluir que la red neuronal no logró adquirir la capacidad de resolver satisfactoriamente el problema del reconocimiento de firmas. Al haberse entrenado y evaluado con un conjunto de firmas y de usuarios relativamente pequeño, los resultados obtenidos por el clasificador de vecinos mas cercanos fueron mejores simplemente porque contaba con toda la información previa disponible a al hora de la clasificación. De la misma forma, el

entrenamiento no fue suficiente para que la red neuronal logre aprender conceptos relativos al problema del reconocimiento de firmas en sí y no a simplemente "memorizar" las firmas que pertenecen o no al sistema.

El problema principal que se debió enfrentar a la hora de resolver el problema de la clasificación fue el rechazo de firmas, que impedía utilizar los algoritmos típicos de clasificación. Otra forma posible de enfrentar este problema hubiera sido almacenando para cada usuario una representación de su "firma promedio", la obtenida por ejemplo mediante la extracción de componentes principales mediante PCA. Una red neuronal podría entrenarse para la tarea de, dadas dos firmas, reconocer si el grado de similitud entre ambas es tal que se pueden considerar como pertenecientes al mismo usuario. De esta forma, y con una base de firmas suficientemente grande, la red lograría absorber información relativa a la comparación de similitud en firmas, como la variación entre el largo de sus trazos, las curvas que la componen, etc.

De todos modos, los resultados obtenidos son aceptables para un primer acercamiento al problema, y demuestran la viabilidad de utilizar firmas trazadas en la pantalla táctil como método de identificación de usuarios.