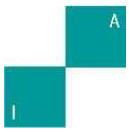
Inteligencia Artificial 43(2009), 1-15
doi: 10.1441/ia.v13i43.1020



INTELIGENCIA ARTIFICIAL

http://erevista.aepia.org/

Extracción de Características Mediante Vectores Comunes Discriminantes Extendidos con Kernel [1]

Katerine Díaz-Chito, Francesc J. Ferri, Wladimiro Díaz-Villanueva
Dept. d'Informàtica. Universitat de València
Avda. Vicent Andrés Estellés, s/n
46100 Burjassot (València) Spain
{katerine.diaz, francesc.ferri, wladimiro.diaz}@uv.es

Resumen En este documento se propone un método de representación y clasificación basado en una extensión del método de los vectores comunes discriminantes con kernel en el que se reinterpreta el espacio nulo de la matriz de dispersión intra-clase para obtener las características discriminantes. El objetivo de la extensión es conseguir que la representación sea más robusta frente a diferentes tipos de variabilidad de los datos y en consecuencia se consiga un mayor poder de generalización. Además el método se evalúa para distintos tamaños del conjunto de entrenamiento, y se compara con el método original en su versión lineal y no lineal así como con otra extensión anterior del método lineal. Para la evaluación empírica, se consideran algunas bases de datos de rostros comúnmente utilizadas pero con ruido añadido, así como una base de datos de objetos públicamente disponible y una base de datos de dígitos escritos a mano. Los experimentos realizados muestran que el método propuesto siempre iguala o supera a los demás métodos considerados con respecto al porcentaje de acierto, incluso cuando se añaden a los datos grandes cantidades de ruido.

Palabras clave: Vectores Comunes Driscriminantes. Extracción de Características. Truco del kernel. Análisis discriminante lineal.

1 Introducción

Recientemente, han surgido numerosos métodos de extracción de características basados en subespacios aplicados a tareas de minería de datos, bioinformática, pronósticos financieros, control, seguridad, etc. Entre los métodos lineales más populares está el Análisis de Componentes Principales (PCA, *Principal Components Analysis* [18]) y el Análisis Discriminante Lineal de Fisher (FLD, *Fisher's Linear Discriminant* [10]). PCA es un método no supervisado usado normalmente para reducir la dimensionalidad del espacio de características preservando al máximo la variabilidad en el espacio original. FLD es un método supervisado que busca una transformación lineal que maximiza la dispersión entre clases y minimiza la dispersión entre miembros de una misma clase.

Cuando la dimensionalidad del espacio de muestras es muy alta o incluso mayor al número de muestras en el conjunto de entrenamiento, muchos métodos de extracción de características fallan o presentan serias dificultades en su aplicación. A esta situación se la suele etiquetar como el problema del número

[1]Trabajo parcialmente subvencionado por FEDER, y el Programa de ayudas FPI del Ministerio de Educación y Ciencia Español a través de los proyectos TIN2009-14205-C04-03, DPI2006-15542-C04-04, y Consolider Ingenio 2010 CSD2007-00018.

ISSN: 1988-3064(on-line)

de muestras de entrenamiento pequeño (*small sample size problem* [11]). Muchos métodos han sido propuestos para resolver este problema en este contexto, como por ejemplo en el caso del FLD [2, 7, 13, 24].

En [6] Cevikalp *et al.* proponen un nuevo método supervisado llamado Método de Vectores Comunes Discriminantes (DCV, *Discriminative Common Vector Method*) en el contexto específico del número de muestras de entrenamiento pequeño, que esta basado en el criterio discriminante lineal de fisher modificado [2, 3].

Los autores muestran empíricamente que el método DCV logra superar significativamente en términos de la tasa de reconocimiento, eficiencia y estabilidad numérica a otros métodos basados en subespacios para el problema específico del reconocimiento de caras a partir de imágenes convenientemente alineadas.

El método DCV, al igual que otros métodos de extracción de características basados en subespacios, ha sido extendido al caso no lineal mediante el truco del kernel que consiste en la proyección implícita del problema a un espacio de mayor dimensión o incluso de dimensión infinita donde es aplicado el discriminante lineal.

La versión no lineal del método DCV ha sido propuesta por Cevikalp *et al.* en [5],denominándose como el Método de Vectores Comunes Discriminantes con kernel (KDCV, *Discriminative Common Vector with Kernel*). Tanto para DCV como para KDCV, las suposiciones subyacentes para el uso de los vectores comunes garantizan siempre un error cero sobre el conjunto de entrenamiento usado. Sin embargo, no hay garantías sobre la probabilidad de error.

Recientemente en [21] se ha planteado una extensión del método DCV (RCV, *Rough Common Vector*) en el que se reinterpretan las bases del método y se relajan, en cierta medida, las suposiciones. Con ello se consiguen dos cosas. Por un lado, desaparece la limitación (del método lineal) de que no puede haber más muestras que dimensiones en el conjunto de entrenamiento. Y por otro lado, se consigue una mejor capacidad de generalización del método en un abanico más amplio de problemas.

En este documento, se propone la versión kernel del método RCV, o dicho de otra forma, una extensión del método KDCV conservando la misma línea en la que ha sido ya propuesto para el caso lineal RCV, con el objetivo de estudiar la capacidad de generalización del método presentado respecto a los métodos basados en vectores comunes, frente a un determinado rango de problemas (con cierta variabilidad), así como observar su comportamiento frente a diferentes tamaños del conjunto de entrenamiento.

El documento se estructura de la siguiente manera. En la Sección 2 se describe la idea de los vectores comunes discriminantes y su correspondiente extensión lineal o método RCV. La Sección 3 propone el concepto del método propuesto. La descripción de las bases de datos empleadas, los experimentos realizados y los resultados obtenidos se presentan en la Sección 4. Finalmente la Sección 5 expone las conclusiones y el trabajo futuro. El Apéndice A muestra los pasos detallados para implementar el método propuesto.

2 Vectores Comunes y Subespacios Nulos

Sea un conjunto de entrenamiento centrado $X = \{x_1^1, x_2^1, \ldots, x_{N_1}^1, x_1^2, \ldots, x_{N_C}^C\}$ con M muestras, C clases y N_j muestras en la clase j, $j = 1, \ldots, C$ y $\sum_j N_j = M$. Las matrices de dispersión inter e intra-clase, S_B y S_W, se definen a partir de X de la forma usual [8].

Dado un determinado conjunto de entrenamiento etiquetado, el método de los vectores comunes discriminantes (DCV) consiste en proyectar el problema original en el subespacio nulo de la matriz S_W. En [6] se demuestra que de esta manera y siempre que el rango de S_W sea mayor que M, todos los elementos de X de una misma clase se proyectan sobre un único vector que recibe el nombre de vector común de la clase j. De esta manera además, se optimiza el criterio de Fisher modificado y por tanto el resultado es equivalente al del método PCA+Null Space [13].

Una forma de caracterizar el subespacio nulo de S_W es mediante sus vectores propios correspondientes a valores propios nulos. Por otro lado, el resto de vectores propios (correspondientes a valores propios no nulos) forman una base del subespacio rango de S_W que es complementario del anterior [8].

La idea del método RCV [21] consiste en reinterpretar el subespacio nulo de S_W, para lo cual se introduce el concepto de subespacio pseudonulo o α-nulo de la siguiente manera.

Sean $\{v_1, \ldots, v_k, \ldots, v_d\}$ los vectores propios de S_W y $\{\lambda_1 \leq \ldots \leq \lambda_k \leq \ldots \leq \lambda_d\}$ sus valores propios normalizados organizados de forma ascendente. d es la dimensionalidad del espacio original, y k es el máximo entero que cumple $\alpha \geq \sum_{i=1}^{k} \lambda_i$, donde $\alpha \in [0, 1)$.

El espacio de representación original se divide en dos subespacios complementarios dados respectivamente por $\{v_1, v_2, \ldots, v_k\}$ y $\{v_{k+1}, v_{k+2}, \ldots, v_d\}$. Si $\alpha = 0$, el primero de estos subespacios es el subespacio nulo de S_W y la correspondiente proyección constituye la base del método DCV. En cambio, para valores (en principio pequeños) mayores que cero, se obtiene un espacio nulo extendido en el que las clases se proyectan formando pequeñas nubes (de tamaño creciente en función de α) en lugar de vectores comunes. La proyección en este espacio α-nulo es la base del método RCV [21]. La constante α del método RCV indica la cantidad de varianza añadida al subespacio nulo de S_W.

La proyección en el subespacio α-nulo de S_W lleva implícito el cálculo de un vector común por cada clase que tiene como objeto servir para diseñar un clasificador por distancia mínima (para asignar la clase más probable a cualquier muestra de test que se proyecte en el mismo subespacio) [6]. En el caso $\alpha = 0$, cualquier muestra de entrenamiento proyectada sirve para obtener el vector común, pero si $\alpha > 0$ la opción que se sugiere en [21] es la de proyectar el vector medio de cada clase y definir éste como vector común.

Cuando $\alpha = 0$, la proyección final lleva a un espacio de como mucho $C - 1$ dimensiones lo cual no es cierto en el caso $\alpha > 0$ aunque, como en la referencia original [21] esto se puede forzar considerando sólo los vectores medios de cada clase en lugar de todo el conjunto de entrenamiento. Diferentes opciones a la hora de proyectar los datos de entrenamiento y test en el espacio α-nulo han sido ya empíricamente evaluadas en [9].

3 Vectores Comunes Discriminantes Extendidos con Kernel

El método DCV ha sido extendido al caso no lineal mediante el llamado truco del kernel [5]. En este contexto, sea ϕ una aplicación no lineal que proyecta el problema en un espacio generalmente de muy alta dimensión en el que se plantea la aplicación de los algoritmos lineales originales.

A partir de ϕ y del conjunto de entrenamiento X, se define su versión proyectada como $\Phi = [\phi(x_1^1)\ \phi(x_2^1)\ \ldots \phi(x_{N1}^1)\ \phi(x_1^2)\ \ldots \phi(x_{Nc}^C)]$.

En este nuevo espacio se definen también las matrices inter e intra-clase S_B^ϕ y S_W^ϕ, respectivamente. Además, la matriz de dispersión total viene dada por $S_T^\phi = S_B^\phi + S_W^\phi$.

A partir de la observación de que la proyección sobre el espacio nulo de S_W^ϕ es equivalente a la proyección sobre la intersección de éste con el subespacio rango de S_T^ϕ, se pueden calcular los vectores comunes en dos pasos: una primera proyección sobre el subespacio rango de S_T^ϕ (que no es más que la aplicación del análisis de componentes principales con kernel o KPCA [20]) seguida de una proyección sobre el subespacio nulo de S_W^ϕ pero una vez transformada según KPCA.

El cálculo es factible dado que tanto los vectores transformados como las diferentes matrices de dispersión se pueden calcular explícitamente a partir de la matriz kernel y de sus valores y vectores propios [20].

$$\tilde{K} = U \Lambda U^T$$

En la anterior expresión $\tilde{K}$ es la versión centrada de la matriz kernel, $K = \Phi^T \Phi$.

Después de este primer paso en el que se aplica KPCA y se obtiene la expresión para la matriz de dispersión intra-clase proyectada, $\tilde{S}_W^\phi$, el problema queda reducido a buscar la proyección en el subespacio α-nulo de ésta última.

Sean $\{\tilde{v}_1, \ldots, \tilde{v}_k, \ldots, \tilde{v}_M\}$ los vectores propios normalizados de $\tilde{S}_W^\phi$ y $\{\tilde{\lambda}_1 \leq \ldots \leq \tilde{\lambda}_k \leq \ldots \leq \tilde{\lambda}_M\}$ sus valores propios normalizados organizados de forma ascendente, donde k es el máximo entero que cumple $\alpha \geq \sum_{i=1}^{k} \tilde{\lambda}_i$.

Al igual que en el caso lineal, si se elige $\alpha = 0$ se obtiene el método KDCV original. Pero con valores de α crecientes se añade variabilidad al subespacio nulo y se obtiene una versión no lineal del método RCV.

En particular y una vez fijado α, se define la matriz de proyección $\tilde{V} = [\tilde{v}_1, \ldots, \tilde{v}_k]$.

En el caso del método KDCV ($\alpha = 0$) obviamente se cumple que

$$\tilde{V}^T \tilde{S}_B^\Phi \tilde{V} = \tilde{V}^T \tilde{S}_T^\Phi \tilde{V} \tag{1}$$

por lo que resulta sencillo calcular una proyección a un subespacio de $C-1$ dimensiones mediante la descomposición

$$\tilde{V}^T \tilde{S}_T^{\Phi} \tilde{V} = \tilde{V}^T \Lambda \tilde{V} = Y \tilde{\Lambda} Y^T$$

Con los anteriores cálculos, el método KDCV utiliza como matriz de proyección (junto con la matriz de kernel), $U\Lambda^{-1/2}\tilde{V}Y$.

Lamentablemente, la igualdad de la ecuación 1 no es cierta para $\alpha > 0$ y si se usa la misma matriz de proyección, la dimensión del espacio final no estará limitada a $C-1$ y crecerá rápidamente con α.

Para obtener un resultado acorde con el método RCV y limitar la dimensión de la proyección final a $C-1$ se debe descomponer la matriz $\tilde{V}^T \tilde{S}_B^{\phi} \tilde{V}$ cuya expresión es ligeramente más complicada. Esto equivale a considerar sólo los vectores medios de cada clase en el espacio de alta dimensión implícitamente definido por ϕ.

$$\tilde{V}^T \tilde{S}_B^{\Phi} \tilde{V} = Y \tilde{\Lambda} Y^T$$

Calculando así los vectores propios Y, la matriz de proyección queda definida como en el método KDCV.

Los vectores comunes se calculan proyectando los vectores medios de cada clase. En el método original ($\alpha = 0$), es suficiente proyectar una muestra de entrenamiento cualquiera, aunque el vector medio daría también el mismo resultado.

Las proyecciones en el nuevo espacio se llevan a cabo de la forma usual mediante la correspondiente matriz kernel que contiene los productos escalares entre las muestras de entrenamiento y los vectores que se quiere proyectar.

En el nuevo espacio se calcula la distancia euclídea de cada elemento que se quiere clasificar a cada uno de los vectores comunes y se aplica la regla de la distancia mínima. Alternativamente, en el caso $\alpha > 0$ se pueden proyectar *todas* las muestras de entrenamiento y usar un clasificador por el vecino más próximo para tener en cuenta la variabilidad de cada clase en el nuevo espacio [9].

4 Experimentos y Resultados

Estos experimentos se realizan con el objetivo de observar la tasa de acierto del método RKDCV en función de la varianza añadida al subespacio de proyección final y al número de muestras de entrenamiento. Además se compara RKDCV con el método lineal no extendido DCV, así como con el no lineal ni extendido KDCV y con la versión lineal extendida RCV, para deducir si tiene o no mayor capacidad de generalización al clasificar las bases de datos empleadas. También se realiza un análisis del tiempo aproximado de entrenamiento de los métodos utilizados.

Las pruebas se han diseñado siguiendo el tipo de experimentación realizada en los trabajos originales donde se proponen los métodos [6, 21, 5].

Puesto que el objetivo es el estudio del método RKDCV se han elegido bases de datos especificas que cubren diversos dominios. Además, en algunas de ellas se ha introducido artificialmente ruido para forzar el método.

La evaluación se realiza clasificando dos bases de datos de rostros llamadas AR NG y ALL NG, así como una base de datos de objetos llamada Coil-40/30, y para Nistgray10 que es una base de datos de dígitos escritos a mano. En todas las bases de datos, excepto en ALL NG, se cumple el caso del número de muestras de entrenamiento pequeño.

Los experimentos se han repetido para varios tamaños relativos del conjunto de entrenamiento usado con respecto al total de muestras disponibles, con objeto de estudiar la dependencia de los métodos en relación a este parámetro. En particular, 1/10, 1/8, 1/5, 1/3, 1/2 y 2/3 del total de muestras disponibles ha sido usado como entrenamiento, repitiéndose estos experimentos 10, 8, 5, 3, 2 y 3 veces, respectivamente. Los resultados de esta validación cruzada repetida en cuanto a la tasa de acierto y al tiempo de entrenamiento serán los respectivos valores promedio.

En la práctica, es interesante conocer el comportamiento de los métodos con un número reducido de muestras de entrenamiento ya que, obviamente, todos los métodos se aproximan al óptimo cuando este tamaño crece. Por lo tanto, en las siguientes subsecciones se mostrarán con más detalle los resultados para

un valor representativo del tamaño relativo del conjunto de entrenamiento. En este caso, se comprueba que el comportamiento de los métodos sigue siendo aceptable aunque empiezan a haber diferencias apreciables.

Los resultados de los experimentos realizados se obtienen a partir de la distancia euclídea entre las características obtenidas del conjunto de entrenamiento y las características del conjunto de prueba. El valor del parámetro α, que añade la varianza al subespacio de proyección final, se incrementa en intervalos de 0.05 desde 0 hasta 0.5.

El kernel empleado para los métodos basados en subespacios no lineales es de tipo gausiano, porqué es el normalmente utilizado en tareas de clasificación de patrones, como en [4, 5, 16, 20].

Para las bases de datos de rostros el parámetro de proximidad del kernel, σ, se ha variado en el intervalo de 20 a 500 y un buen resultado se ha obtenido cuando $\sigma = 100$. Para la base de datos de objetos el intervalo de valores considerado es de 20 a 1400 y un resultado aceptable se ha obtenido con $\sigma = 1000$. El rango de valores de σ considerado en la base de datos de dígitos escritos a mano es de 10 a 500, seleccionando $\sigma = 20$ como apropiado.

4.1 Experimentos usando AR NG

AR NG es una base de datos de imágenes de rostros en niveles de gris con valores de intensidad entre 0 y 1, que se originó al seleccionar 20 clases de la base de datos AR Face [15]. De cada clase se escogieron 14 muestras sin oclusiones que se normalizaron y, se les añadió ruido gausiano con valores de varianza de 0, 0.02, 0.04, 0.06 y 0.08. Obteniendo 70 muestras por clase que presentan cambios de expresión, iluminación y ruido. Las imágenes de tamaño inicial 768x576 son reducidas a un tamaño de 40x40 por un análisis previo. La figura 1 enseña la degradación de una muestra en AR NG.

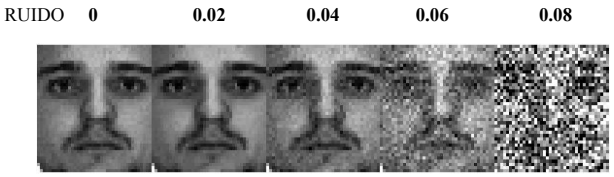


Figure 1: Ejemplo de la degradación de las imágenes en la base de rostros AR NG.

Para examinar en detalle el método RKDCV, respecto a la tasa de acierto en función de la varianza añadida al subespacio de proyección final de AR NG y del número de muestras de entrenamiento, se gráfica la figura 2. El tamaño de los conjuntos de entrenamiento empleados no supera la dimensionalidad del espacio original.

Se observa que, evidentemente y como era de esperar, a mayor muestras de entrenamiento tanto KDCV ($\alpha = 0$) como RKDCV ($\alpha > 0$) mejoran. Pero, para el mismo número de muestras de entrenamiento el método RKDCV obtiene mejores resultados al incrementar la varianza añadida al subespacio de proyección final, excepto si la tasa de acierto de KDCV es lo suficientemente alta (0.99) como es el caso de 47 muestras de entrenamiento por clase (teniendo presente que el número de muestras es de 70 por clase).

Los incrementos más significativos en las tasas de acierto de RKDCV respecto a KDCV ocurren cuando el conjunto de entrenamiento no es muy grande, como es el caso de 7, 9 y 14 muestras por clase (7.93%, 4.37% y 3.21%, respectivamente) con tasas de acierto de 0.8, 0.86 y 0.95, respectivamente.

Los resultados obtenidos en esta prueba sugieren que incrementar la varianza del subespacio de proyección en el método RKDCV es eficiente siempre y cuando el número de muestras de entrenamiento no sea muy elevado ya que el método KDCV ($\alpha = 0$) presenta tasas de acierto altas saturando el sistema, lo cual no permite conseguir tasas de acierto mayores.

La figura 3 muestra el comportamiento de RKDCV al incrementar la varianza añadida al subespacio de proyección final de AR NG respecto al método extendido RCV y a los no extendidos DCV y KDCV, para un número aceptable de 23 muestras de entrenamiento por clase. La tasa de acierto de DCV no es visible en está figura para observar mejor las diferencias presentes entre RKDCV y RCV.

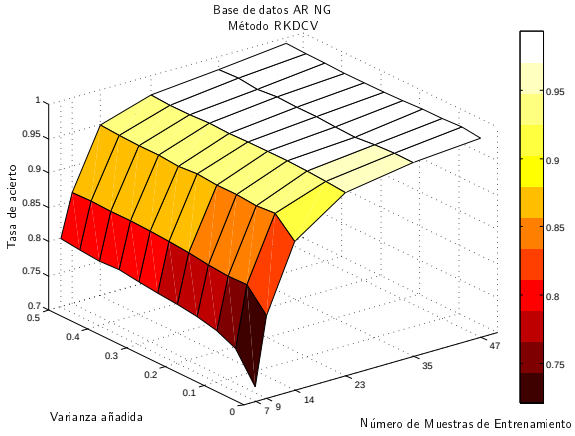


Figure 2: Tasa de acierto en función del número de muestras de entrenamiento y la varianza añadida al subespacio de proyección final de AR NG, para el método RKDCV.

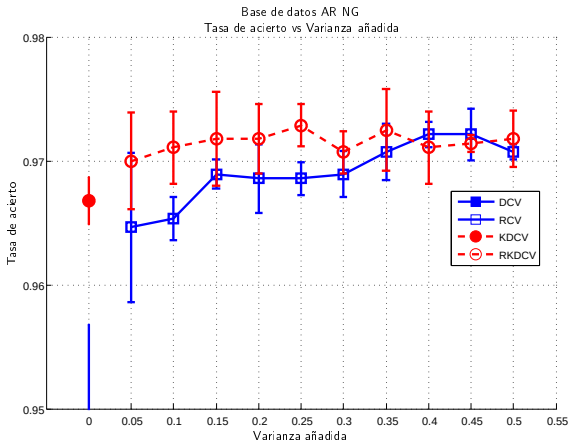


Figure 3: Tasa de acierto en función de la varianza añadida al subespacio de proyección final de AR NG, para 23 muestras de entrenamiento por clase.

Al observar la figura 3 se evidencia que la diferencia entre RKDCV y los otros métodos es mayor cuando la varianza añadida es de 0.25, proporcionando un 2.47%, 0.43% y un 0.61% de mejora respecto a DCV, RCV y KDCV, respectivamente. Cuando la varianza añadida es superior a 0.4, la tasa de acierto de RKDCV es similar a la de RCV.

4.2 Experimentos usando ALL NG

La base ALL NG resulta de la elección aleatoria de 10 muestras sin oclusiones por clase de las bases de rostros AR Face [15], ORL [19], Yale [12] y Umist [23], obteniendo 95 clases. Cada clase de ALL NG tiene 50 muestras, de las cuales 10 no tienen ruido añadido y 40 presentan ruido gausiano añadido (generadas a partir de las 10 imágenes originales normalizadas de cada sujeto). Las características del ruido añadido son idénticas a las de AR NG

Las imágenes son de tamaño 40x40 y presentan cambios de expresión, iluminación, pose y ruido. La figura 4 expone algunas muestras de ALL NG.

Con esta base de datos, significativamente más grande que la anterior, se estudia el comportamiento

de los métodos a medida que el número total de muestras crece respecto a la dimensionalidad. Concretamente, para 17 o más muestras por clase se cumple que la dimensión del espacio es menor que el número total de muestras.

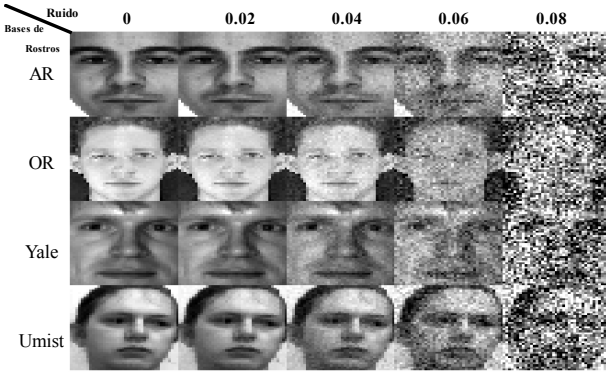


Figure 4: Ejemplo de la degradación de las imágenes en las bases de rostros AR, ORL, Yale y Umist.

Al dibujar en la figura 5 la tasa de acierto en función de la varianza añadida al subespacio de proyección final de ALL NG para el problema del número de muestras de entrenamiento pequeño y sin este problema se observa que, al igual que en AR NG (sección 4.1), al haber más muestras de entrenamiento tanto KDCV ($\alpha = 0$) y RKDCV ($\alpha > 0$) obtienen mejores resultados. Además el método RKDCV supera a KDCV al incrementar la varianza del subespacio de proyección final.

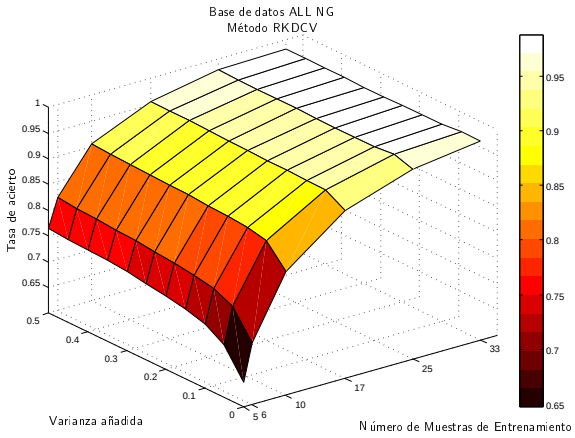


Figure 5: Tasa de acierto en función del número de muestras de entrenamiento y la varianza añadida al subespacio de proyección final de ALL NG, para el método RKDCV.

Es importante recalcar que el método extendido propuesto es aplicable independientemente de la relación entre la dimensión del espacio original y el número de muestras de entrenamiento total.

La tasa de acierto de RKDCV en relación con la de KDCV, presenta cambios más significativos en el caso de muestra pequeña (menores que 17 muestras por clase) obteniendo un 11.68%, 9.95% y 6.64% más de aciertos. Para el tamaño del conjunto de entrenamiento más grande, se obtiene también mejoras.

En la figura 6 se representa el comportamiento de los métodos en función de la varianza añadida al subespacio de proyección final, considerando 17 muestras de entrenamiento por clase. El método DCV no se aplica porque la dimensionalidad del espacio original es menor al número de muestras de entrenamiento total.

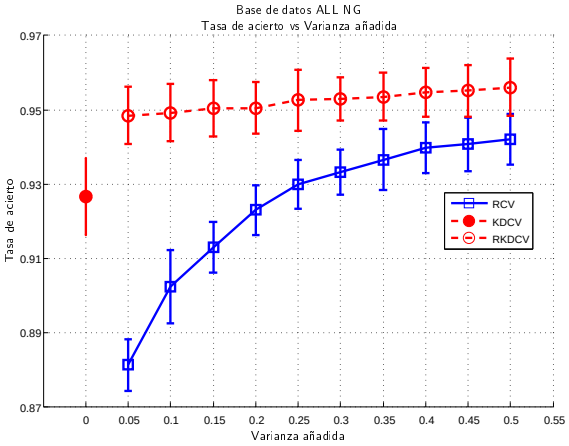


Figure 6: Tasa de acierto en función de la varianza añadida al subespacio de proyección final de ALL NG, para 17 muestras de entrenamiento por clase.

Como se aprecia en la figura 6, el método RKDCV destaca al clasificar ALL NG con una tasa de acierto de 0.96 ($\alpha = 0.5$) respecto a 0.94 para KDCV y 0.93 para RCV. Las diferencias entre los métodos utilizados son más evidentes que en el caso AR NG.

4.3 Experimentos usando Coil-40/30

Coil-40/30 es una base de datos de objetos en niveles de gris que procede de Coil-100 [17], y está constituida por 40 clases seleccionadas al azar entre las 100 clases de la base de datos original. Por cada clase hay 30 imágenes seleccionadas aleatoriamente entre las 72 originales. Las imágenes de tamaño inicial 128x128 son reducidas a un tamaño de 40x40. La figura 7 muestra un objeto por clase de la base de datos Coil-40/30.

Figure 7: Clases de la base de datos Coil-40/30.

Al gráficar en la figura 8 la tasa de acierto en función del número de muestras de entrenamiento y la varianza adicionada al subespacio de proyección final es evidente que al aumentar la varianza la tasa de acierto mejora, incluso para un número de muestras de entrenamiento significativo (en este caso 15 y 20).

En la figura 9 se observa la superioridad de RKDCV sobre DCV, RCV y KDCV al elevar la varianza del subespacio de proyección final de Coil-40/30, para 10 muestras de entrenamiento por clase. A diferencia

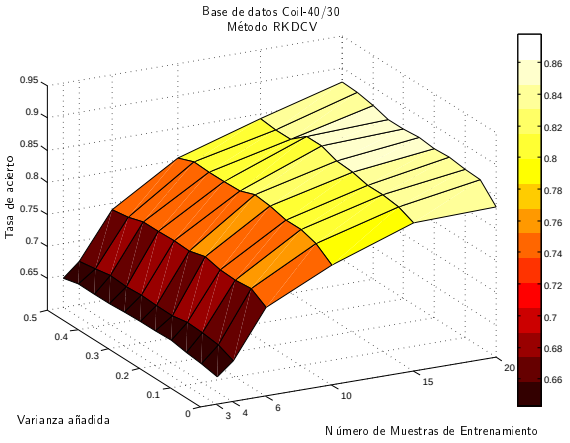


Figure 8: Tasa de acierto en función del número de muestras de entrenamiento y la varianza añadida al subespacio de proyección final de Coil-40/30, para el método RKDCV.

de las bases de rostros, el comportamiento del método RCV no iguala ha KDCV.

También se nota que la tasa de acierto más alta al clasificar Coil-40/30 es 0.82 que corresponde al método RKDCV cuando la varianza añadida al subespacio de proyección final es de 0.25. Para este valor de varianza añadida RKDCV es 8.37%, 4.70%, y 3.41% superior a DCV, RCV y KDCV, respectivamente.

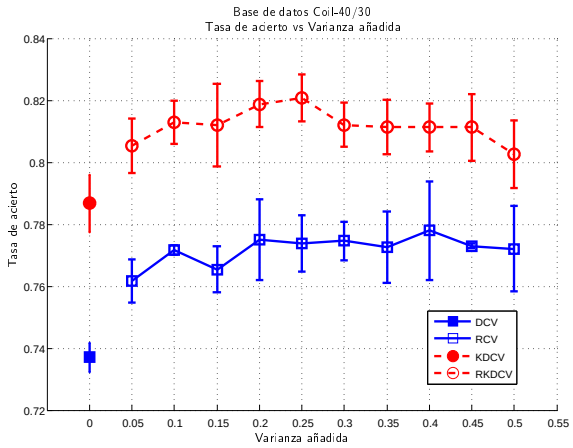


Figure 9: Tasa de acierto en función de la varianza añadida al subespacio de proyección final de Coil-40/30, para 10 muestras de entrenamiento por clase.

4.4 Experimentos usando Nistgray10

Nistgray10 es una base de datos en niveles de gris de dígitos de 0 a 9 escritos a mano (ver Figura 10), la cual deriva de la base de datos Nist32 (de tamaño 32x23) disponible en prtools [22]. Esta conformada por 10 clases, de cada clases hay 100 muestras seleccionadas al azar de entre el total. Las imágenes binarias han sido convertidas en grises mediante el uso de la transformación de distancia [1, 14].

Para forzar el comportamiento del método RKDCV al clasificar la base de datos Nistgray10 se intercambia el 10% de las etiquetas disponibles en el conjunto de entrenamiento, sin favorecer a una clase u a otra .

Figure 10: Algunas muestras de la base de datos Nistgray10.

Al aumentar la varianza del subespacio de proyección para un mismo número de muestras de entrenamiento, la tasa de acierto de RKDCV crece. Ver Figura 11.

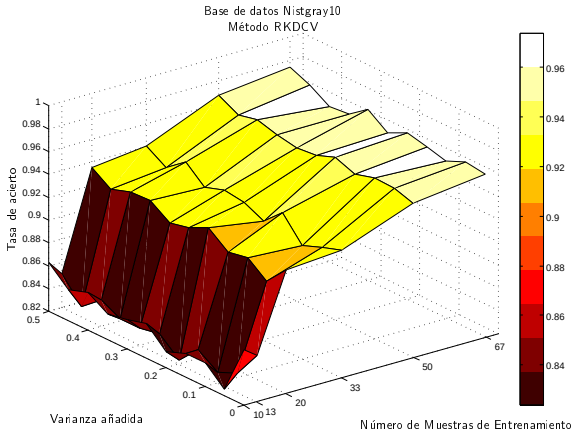


Figure 11: Tasa de acierto en función del número de muestras de entrenamiento y la varianza añadida al subespacio de proyección final de Nistgray10, para el método RKDCV.

La Figura 12 ilustra el comportamiento del los métodos RKDCV, KDCV, RCV y DCV al incrementar la varianza del subespacio de proyección utilizando 50 muestras de entrenamiento por clase. Se observa que los métodos que emplean kernel son superiores a los métodos lineales utilizados en esta prueba. Al igual que en la subsección anterior RCV no supera ni igual a al método propuesto RKDCV.

Los resultados obtenidos muestran que entre los métodos considerados, RKDCV presenta un comportamiento eficiente.

4.5 Análisis del tiempo de entrenamiento

En un sistema de clasificación además de la tasa de acierto de los métodos aplicados también es importante el tiempo de entrenamiento de los mismos. Por ello en esta sección se presenta una estimación de los tiempos de RKDCV de los casos anteriores.

El ordenador empleado en las pruebas tiene un procesador Dual Core AMD Opteron(tm) Processor 270, Velocidad de 1,000.00 MHz, 4 núcleos y openSUSE 11.0 (x86_64) como sistema operativo.

La figura 13 muestra el tiempo de entrenamiento del método RKDCV en función de la varianza añadida al subespacio de proyección final de AR NG, ALL NG, Coil-40/30 y Nistgray10 empleando 23, 17, 10 y 50 muestras de entrenamiento por clase respectivamente, los cuales son valores representativos considerados en las secciones anteriores.

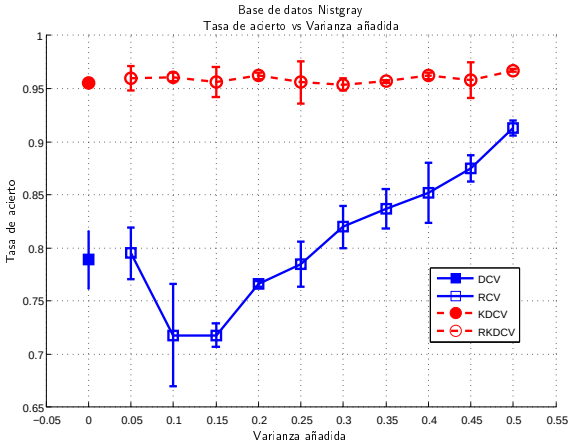


Figure 12: Tasa de acierto en función de la varianza añadida al subespacio de proyección final de Nistgray10, para 50 muestras de entrenamiento por clase.

Se observa, como era de esperar, que al aumentar la varianza añadida al subespacio de proyección final de cada base de datos utilizada, el tiempo de entrenamiento crece puesto que el tamaño de la base ortonormal del espacio α-nulo de $\tilde{S}_W^{\Phi}$ es directamente proporcional al valor del parámetro que incrementa la varianza. En consecuencia, el computo de la matriz de proyección final es un poco más costoso de realizar.

También se nota que el tiempo de entrenamiento de RKDCV depende en mayor medida del número de muestras de entrenamiento total, puesto que el tamaño de la función kernel está dado por este. Así, el tiempo de entrenamiento de ALL NG es el más alto con 1615 muestras de entrenamiento, seguido de Nistgray10 con 500, AR NG con 460 y Coil-40/30 con 400 muestras.

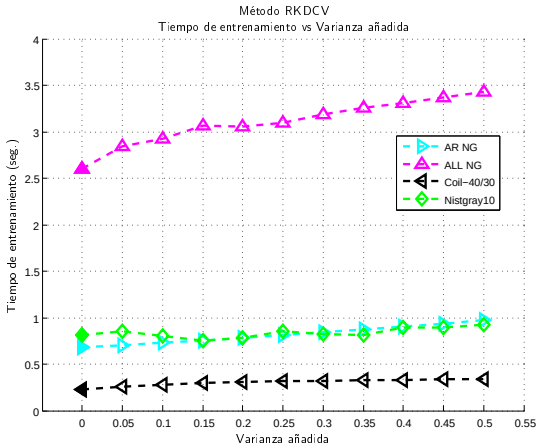


Figure 13: Tiempo de entrenamiento del método RKDCV en AR NG, ALL NG, Coil-40/30 y Nistgray10, para 23, 17, 10 y 50 muestras de entrenamiento por clase, respectivamente.

5 Conclusiones y trabajo futuro

El propósito de este documento ha sido analizar el comportamiento del método propuesto RKDCV, para diferentes tamaños del conjunto de entrenamiento al incrementar la varianza añadida al subespacio de proyección final de algunas bases de datos de rostros comúnmente utilizadas pero con ruido añadido, así como una base de datos de objetos públicamente disponible y una base de datos de dígitos de 0 a 9 escritos a mano. Además se compara RKDCV con los métodos DCV, RCV y KDCV, al clasificar las bases de datos empleadas.

Se ha escogido como punto de partida el comportamiento del método RKDCV al incrementar la varianza del subespacio de proyección para diferentes tamaños del conjunto de entrenamiento. Para completar el análisis se han comparado los resultados relativos a los métodos investigados con respecto a la tasa de acierto, para cada una de las bases de datos consideradas. También se ha realizado un pequeño análisis del tiempo de entrenamiento de RKDCV al aumentar la varianza del subespacio de proyección final, solo para dar una idea de las estimaciones de los tiempos.

Como primera conclusión de este documento se observa, que adicionar varianza al subespacio de proyección final no afecta negativamente la tasa de acierto de los métodos extendidos y que por el contrario la mejora en la mayoría de los casos.

Si en el sistema no se ha añadido varianza y la tasa de acierto es grande (0.99), la varianza añadida al subespacio de proyección final no contribuye a obtener mejores resultados, ya que RKDCV no suple las carencias del método con kernel no extendido (KDCV) cuando el sistema esta saturado. Si la tasa de acierto no es demasiado alta el sistema se puede mejorar añadiendo varianza a este subespacio.

Respecto al tiempo de entrenamiento de RKDCV se advierte que al aumentar el valor del parámetro que adiciona varianza este también crece, pero el incremento no es significativo en relación a la tasa de acierto y depende más del número de muestras de entrenamiento que se usa.

A partir del desarrollo del trabajo realizado y de acuerdo a los resultados obtenidos, se concluye que para las bases de datos utilizadas, el método propuesto RKDCV presenta mejoras en cuanto a la tasa de acierto de los métodos DCV, RCV y KDCV.

Respecto al trabajo futuro, se intentará desarrollar el método en torno a la maximización de la dispersión total del espacio reducido, lo cual conlleva a estudiar las dimensiones involucradas en las diferentes matrices de proyección. Además se experimentará con otras medidas de distancia a partir de las características discriminantes de entrenamiento y las de prueba.

Apéndice A. Vectores comunes discriminantes extendidos con kernel. Implementación

El algoritmo empleado en el método de vectores comunes discriminantes extendidos con kernel, RKDCV, para un conjunto de entrenamiento centrado $\{x_1^1, x_2^1, \ldots, x_{N_1}^1, x_1^2, \ldots, x_{N_C}^C\}$ con M muestras, C clases y N_j muestras en la clase j, se resume paso a paso de la siguiente forma:

Paso 1: Computar la matriz kernel K y centrarla.

La matriz kernel $K \in \mathbb{R}^{(MxM)}$ esta dada por

$$K = \Phi^T \Phi = (K^{ij})_{\substack{i=1,\ldots,C \\ j=1,\ldots,C}}$$

y cada matriz $K^{ij} \in \mathbb{R}^{(N_i x N_j)}$ es definida como

$$\begin{aligned} K^{ij} &= (k_{mn}^{ij})_{\substack{m=1,\ldots,N_i \\ n=1,\ldots,N_j}} = \langle \phi(x_m^i), \phi(x_m^j) \rangle \\ &= k(x_m^i, x_n^j)_{\substack{m=1,\ldots,N_i \\ n=1,\ldots,N_j}} \end{aligned} \tag{2}$$

Donde $k(\cdot)$ representa la función kernel, que normalmente es un kernel gausiano de la forma

$$K^{ij} = exp\left(-\frac{\|x_m^i - x_n^j\|^2}{2\sigma^2}\right)_{\substack{m=1,\ldots,N_i \\ n=1,\ldots,N_j}}$$

y σ es el parámetro de proximidad del kernel.

Calculada la matriz kernel se centra a partir de

$$\tilde{K} = K - 1_M K - K 1_M + 1_M K 1_M \tag{3}$$

con $1_M = (1/M)_{(MxM)}$.

Paso 2: Proyectar el conjunto de entrenamiento en $R(S_T^\Phi)$, calculando la matriz diagonal Λ de valores propios no nulos y la matriz U de vectores propios normalizados asociados a Λ de la matriz kernel centrada $\tilde{K}$.

$$\tilde{K} = U \Lambda U^T \qquad \in \quad \mathbb{R}^{(MxM)} \tag{4}$$

En [5] se dan las expresiones para las nuevas matrices de dispersión total, $\tilde{S}_T^\Phi$ (que no es necesaria), e intra-clase, $\tilde{S}_W^\Phi$.

$$\tilde{S}_W^\Phi = \Lambda^{-1/2} U^T \tilde{K}_W \tilde{K}_W^T U \Lambda^{-1/2} \tag{5}$$

donde $\tilde{K}_W = (K - 1_M K)(I - G)$.

I es una matriz identidad de tamaño (MxM) y $G = diag[G_1, \ldots, G_C] \in \mathbb{R}^{(MxM)}$ es una matriz diagonal con $G_j \in \mathbb{R}^{(N_j x N_j)}$ como una matriz con todos sus elementos iguales a $1/N_j$

La nueva matriz de dispersión intra-clase, esta dada por

$$\tilde{S}_B^\Phi = \Lambda^{-1/2} U^T \tilde{K}_B \tilde{K}_B^T U \Lambda^{-1/2} \tag{6}$$

con $\tilde{K}_B = (K - 1_M K)(H - L)$, $H = diag[\mu_1, \ldots, \mu_C] \in \mathbb{R}^{(MxC)}$ como una matriz diagonal donde cada $\mu_j \in \mathbb{R}^{(N_j x 1)}$ es un vector con todos sus elementos iguales a $1/\sqrt{N_j}$ y $L = [l_1, \ldots, l_C] \in \mathbb{R}^{(MxC)}$ es una una matriz donde cada $l_j \in \mathbb{R}^{(Mx1)}$ es un vector con todos sus elementos iguales a $\sqrt{N_j}/M$.

Paso 3: Agrupar en $\tilde{V}$ los vectores propios normalizados asociados a los valores propios de $\tilde{S}_W^\Phi$ bajo la condición de α.

$$\tilde{V} = [\tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_k] \tag{7}$$

tal que $\alpha \geq \sum_{i=1}^{k} \tilde{\lambda}_i$, $\{\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_k \leq \ldots \leq \lambda_M\}$ donde k es máximo.

Paso 4: Remover el espacio nulo de $(\tilde{V}^T \tilde{S}_B^\Phi \tilde{V})$,

$$\tilde{V}^T \tilde{S}_B^\Phi \tilde{V} = Y \tilde{\Lambda} Y^T \tag{8}$$

y calcular la matriz de proyección final $\tilde{W}$ dada por

$$\tilde{W} = U \Lambda^{-1/2} \tilde{V} Y \tag{9}$$

donde $\tilde{W}$ tiene $C - 1$ vectores de proyección.

Paso 5: Computar el RKDCV de cada clase.

$$\Omega_{rkdcv}^j = (U \Lambda^{-1/2} \tilde{V} Y)^T \overline{\tilde{K}^j} \tag{10}$$

con $\overline{\tilde{K}^j}$ como el vector medio del conjunto de entrenamiento de la clase j en $\tilde{K}$.

Para proyectar el conjunto de prueba se calcula la matriz de kernel

$$K^p = k(x_m^i, x_p)_{\substack{i=1,\ldots,C \\ m=1,\ldots,N_i \\ p=1,\ldots,M_p}}$$

y se proyecta con

$$\Omega_p = (U \Lambda^{-1/2} \tilde{V} Y)^T (K^p - K 1'_M - 1_M K^p + 1_M K 1'_M) \tag{11}$$

donde $1'_M = (1/M)_{(MxM_p)}$.

Para finalizar la clasificación se calcula la distancia euclídea entre el RKDCV de cada clase y las muestras de test proyectadas, asignando la muestra de test a la clase donde la distancia es mínima.

References

[1] J. Arlandis, J.-C. Perez-Cortes, and R. Llobet. Handwritten character recognition using the continuous distance transformation. volume 1, pages 940–943, 2000.

[2] P.N Belhumeur, J.P. Hespanha, and D.J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997. doi: 10.1.1.10.3247.

[3] Y. Bing, J. Lianfu, and C. Ping. A new lda-based method for face recognition. In *ICPR '02: Proceedings of the 16th International Conference on Pattern Recognition (ICPR'02)*, number 1, pages 68–171, 2002.

[4] H. Cevikalp, M. Neamtu, and A. Barkana. The kernel common vector method: A novel nonlinear subspace classifier for pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 37(4):937–951, 2007. doi: 10.1109/TSMCB.2007.896011.

[5] H. Cevikalp, M. Neamtu, and M. Wilkes. Discriminative common vector method with kernels. *IEEE Transactions on Neural Networks*, 17(6):1550–1565, 2006. doi: 10.1109/TNN.2006.881485.

[6] H. Cevikalp, M. Neamtu, M. Wilkes, and A. Barkana. Discriminative common vectors for face recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(1):4–13, 2005. doi: 10.1109/TPAMI.2005.9.

[7] L.-F. Chen, H.-Y.M. Liao, M.-T. Ko, J.-C. Lin, and G.J. Yu. A new lda-based face recognition system which can solve the small sample size problem. *Pattern Recognition*, 33(10):1713–1726, 2000. doi: 10.1016/S0031-3203(99)00139-9.

[8] O.R Duda, P.E. Hart, and G.D. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.

[9] K. Díaz-Chito, F.J. Ferri, and W. Díaz-Villanueva. An empirical evaluation of common vector based classification methods and some extensions. In N. da Vitoria Lobo et al., editor, *Structural, Syntactic and Statistical Pattern Recognition. Lecture Notes in Computer Science. Vol. 5342*, pages 977–985. Springer-Verlag, 2008.

[10] R. Fisher. The use of multiple measures in taxonomic problems. *Ann. Eugenics.*, 7:179–188, 1936.

[11] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 2 edition, 1990.

[12] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001. doi: 10.1109/34.927464.

[13] R. Huang, Q. Liu, H. Lu, and S. Ma. Solving the small sample size problem of lda. In *ICPR '02: Proceedings of the 16th International Conference on Pattern Recognition (ICPR'02)*, number 3, pages 29–32, Washington, DC, USA, 2002. IEEE Computer Society.

[14] Z.M. Kovacs and R. Guerrieri. Computer recognition of hand-written characters using the distance transform. *Electronics Letters*, 28(19):1825–1827, Sept. 1992. doi: 10.1049/el:19921164.

[15] A.M. Martinez and R. Benavente. The ar face database. Technical Report 24, Computer Vision Center CVC), 1998.

[16] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.R. Müler. Fisher discriminant analysis with kernels. In *IEEE Workshop on Neural Networks for Signal Processing*, 1999.

[17] S.A. Nene, S.K. Nayar, and H. Murase. Columbia object image library (coil-100). Technical Report CUCS-006-96, Department of Computer Science, Columbia University, 1996.

[18] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572, 1901.

[19] F.S. Samaria and A. Harter. Parameterisation of a stochastic model for human face identification. In *WACV94*, pages 138–142, 1994.

[20] B. Schölkopf, A. Smola, and K.R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. Technical Report 44, Max Planck Institute for Biological Cybernetics, Tübingen, Germany, 1996.

[21] A. Tamura and Q. Zhao. Rough common vector: A new approach to face recognition. In *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, pages 2366–2371, 2007.

[22] Ferdi van der Heijden, Robert P.W. Duin, Dick de Ridder, David M.J. Tax, and John Wiley & Sons. *Classification, parameter estimation and state estimation - an engineering approach using Matlab.* Wiley, 2004.

[23] H. Wechsler, P.J. Phillips, V. Bruce, F.S. Fogelman, and T.S. (eds) Huang. Face recognition: From theory to applications. *NATO ASI Series F, Computer and Systems Sciences*, 163:446–456, 1998.

[24] H. Yu and J. Yang. A direct lda algorithm for high-dimensional data with application to face recognition. *Pattern Recognition*, 34:2067–2070, 2001. doi: 10.1016/S0031-3203(00)00162-X.