Práctica: Análisis de Componentes Principales (PCA)

Fecha límite de entrega 16 de Abril de 2017

Instrucciones:

- En todos los ficheros que comiencen por **ApellidosNombre** deberás sustituir **ApellidosNombre** por tus apellidos y nombre.
- Deberás subir a *prado2* un único fichero de la forma **ApellidosNombreGuionPCA.zip**. Dicho fichero contendrá todos los ficheros que contenía GuionPCA habiendo reemplazado ApellidosNombre por tus apellidos y nombre. La fecha límite de entrega es el 16 de Abril de 2017.
- Observa que todos los ficheros que comienzan con **ApellidosNombre** requieren que realices modificaciones en ellos.

Introducción

En esta práctica comenzaremos describiendo el funcionamiento de las PCA con un ejemplo sencillo (sección 1) y a continuación las utilizaremos para encontrar una representación de dimensión baja de imágenes de una cara (sección 2). Todo el material necesario para hacer la práctica lo tienes en **GuionPCA.zip**.

Análisis de Componentes Principales. Ejemplo sencillo

En este ejercicio usaremos PCA para realizar reducción de la dimensionalidad con un ejemplo sencillo (de 2D a 1D) para que entiendas como funciona. El guión **ApellidosNombreERRD_pca.m** lo utilizaremos en esta parte de la práctica. Deberás completar el código de las funciones que iremos indicando.

1.1 Base de datos de ejemplo (sección 1 del guión ApellidosNombreERRD_pca.m)

Para ayudarte a entender cómo funciona PCA vamos a empezar con una base de datos bidimensional que tiene una dirección con una variación grande y otra con una variación pequeña. La sección 1 del guión **ApellidosNombreERRD_pca.m** lee y dibuja los datos de entrenamiento (Figura 1).

En situaciones reales puedes querer reducir, por ejemplo, datos de 256 a 50 dimensiones o números mucho más elevados, pero nuestro ejemplo permite su visualización

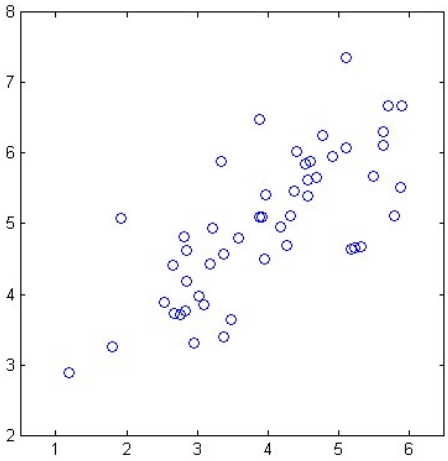

Figura 1. Base de Datos de ejemplo

1.2 Implementando PCA (sección 2 del guión ApellidosNombreERRD_pca.m)

En esta parte del ejercicio implementarás PCA. PCA tiene dos pasos: primero calcular la matriz de covarianza muestral y luego usar la función SVD para calcular los autovectores $U_1,U_2,...,U_M$. Estos vectores corresponderán a las principales componentes de variación en los datos.

Antes de utilizar PCA, es importante normalizar primero los datos restándole la media de cada rasgo y escalando cada dimensión de forma que tengan el mismo rango. En **ApellidosNombreERRD_pca.m** esta operación la realiza la función **featureNormalize.**

Una vez normalizados los datos ejecutamos debes completar el código de **ApellidosNombrepca.m** para calcular las componentes principales. Primero, debes calcular la matriz de covarianza de los datos (Sigma). No te confundas, aquí X tiene los ejemplos por filas mientras que la teoría hemos escrito los ejemplos como columnas.

Una vez calculada la matriz de covarianza debes calcular su descomposición por valores singulares. Puedes ejecutar la orden **[U,S,V]=svd(Sigma)**, donde U contendrá las componentes principales (autovectores de la matriz de covarianza de los datos) y S una matriz diagonal.

Si has completado **ApellidosNombrepca.m** correctamente, el guión **ApellidosNombreERRD_pca.m** ejecutará PCA sobre el ejemplo y dibujará las componentes principales que ha encontrado (figura 2). El guión mostrará también la componente principal más representativa (la que acumula mayor varianza). Debería salirte [0.707 0.707] o su negativo.

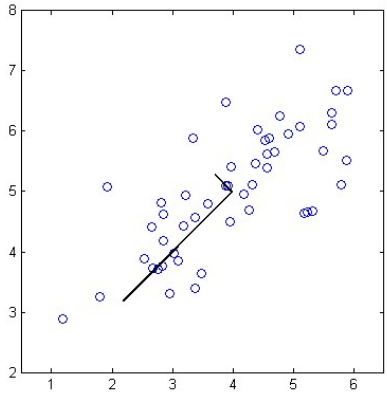

Figura 2: Autovectores de los datos computados

1.3 Reducción de dimensionalidad con PCA (sección 3 del guión ApellidosNombreERRD_pca.m)

Una vez calculadas las componentes principales, podemos utilizarlas para reducir la dimensión de los rasgos proyectando los ejemplos en una dimensión menor $x^{(i)} \rightarrow z^{(i)}$ (en nuestro ejemplo llevándo los datos de 2D a 1D). En esta parte del ejercicio utilizaremos los autovectores (componenetes principales) que nos devolvió **ApellidosNombrepca.m** y proyectaremos los datos en el espacio 1D. Fíjate que con los nuevos rasgos podemos abordar cualquier problema de aprendizaje automático.

Comienza por completar el código de **ApellidosNombreprojectData.m**. Concretamente, dada una base de datos (te recuerdo que los ejemplo van por filas aquí) las componentes principales U y el número de dimensiones a los que deseamos reducir, proyecta cada ejemplo en las K primeras componentes de U. Observa que las primeras K componentes de U vienen dadas por las primeras K columnas de U, es decir, **U_reduce=U(:,1:K).**

Una vez que has completado el código en **ApellidosNombreprojectData.m, ApellidosNombreERRD_pca.m** proyectará el primer ejemplo en la primera dimensión y debería salirte un valor de 1.481.

Una vez que hemos proyectado los datos en un espacio de dimensión inferior, recuperaremos aproximadamente los datos proyectándolos de vuelta en el espacio de mayor dimensión. Tu objetivo es ahora completar **ApellidosNombrerecoverData.m** para proyectar cada ejemplo en Z de vuelta en el espacio original y devolver la aproximación recuperada en **X_rec.** [-1.047,-1.047] debería ser la aproximación del primer ejemplo.

Una vez que hemos completado las dos funciones **ApellidosNombreprojectData.m** y **ApellidosNombrerecoverData.m**, cuando ejecutemos la tercera sección del guión, **ApellidosNombreERRD_pca.m** realizará la proyección y la aproximación de los datos y mostrará como las proyecciones afectan a los datos. En la figura 3, los datos originales aparecen en azul y las proyecciones en rojo. Sólo tenemos información en una dirección.

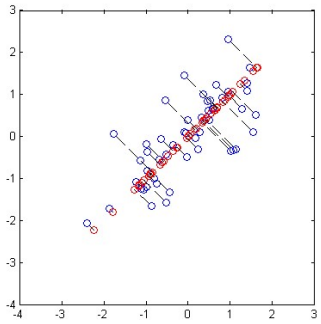

Figura 3: Datos normalizados y proyectados después de PCA

2. Análisis de Componentes Principales. Las caras

En esta parte del ejercicio, ejecutarás PCA sobre imágenes de caras para ver como PCA puede utilizarse paras reducir la dimensionalidad. La base de datos **ERRDfaces.mat** contiene una base de datos X de caras cada una con 32x32 niveles de gris. Cada fila de X contiene una cara (un vector de 1024 rasgos). La sección 4 de **ApellidosNombreERRD_pca.m** cargará y visualizará las primeras 100 caras de imágenes (figura 4).

Figura 4: caras en la base de datos

2.1 PCA sobre caras (sección 5 del guión ApellidosNombreERRD_pca.m)

Para ejecutar PCA sobre la base de datos de caras, primero normalizamos la base restándole la media a la matriz de rasgos X. El guión **ApellidosNombreERRD_pca.m** realizará esta tarea por ti y luego ejecuta el código PCA. Una vez que hayas ejecutado PCA obtendrás las componentes principales de la base de datos. Observa que cada componente principal U es un vector de longitud n (donde en el caso de las caras n=1024) . Podemos por tanto visualizar cada autovector como una matriz 32x32 que es el tamaño de las caras originales. La sección 5 de **ApellidosNombreERRD_pca.m** muestra las 36 primeras componentes que describen la mayor variación. Puedes cambiar el código para ver más o menos autocaras.

Figura 5: 32 primeras autocaras

2.2 PCA sobre caras (sección 6 del guión ApellidosNombreERRD_pca.m)

Ahora que has calculado las componentes principales para la base de datos de las caras, podemos usarlas para reducir la dimensión de la base de datos. Esto permitirá utilizar algoritmos de aprendizaje automático que utilicen menos rasgos (por ejemplo 100 en lugar de los 1024 iniciales). Obviamente esto reducirá el tiempo de aprendizaje. Éste es el objetivo de la sección sexta del guion ERRD_pca.m

2.3 Diferencias (sección 7 del guión ApellidosNombreERRD_pca.m)

Para comprender qué hemos perdido con la reducción de la dimensionalidad podemos recuperar (aproximar) los datos originales usando los datos proyectados. En la sección séptima de **ApellidosNombreERRD_pca.m** podemos comparar los datos originales y su reconstrucción usando PCA

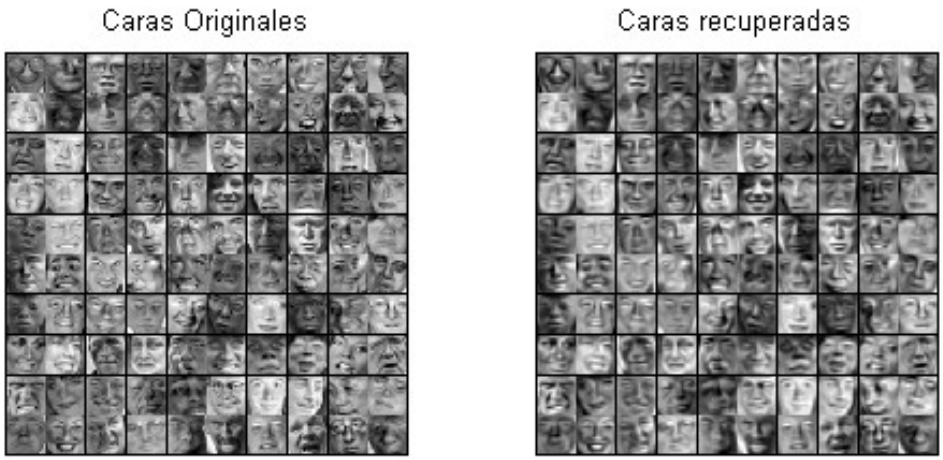

Figura 6: Caras originales y reconstruidas