Aprendizaje Automático

Beltrán Domínguez, Victoria
Ge, Xiangyang

Memoria Práctica 1: Mixturas Gaussianas

En esta práctica, se pretende que el alumno establezca una primera toma de contacto con octave y que además entienda el funcionamiento de las mixturas gaussianas. Primeramente, se presenta resumidamente el funcionamiento de un clasificador multinomial. Seguidamente, se nos presenta el clasificador gaussiano y se nos sugieren varios ejercicios.

El primer ejercicio sugerido (ejercicio 4.1) consiste en realizar un experimento para evaluar el error del clasificador gaussiano sin suavizado en función del número de componentes PCA a las cuales se proyectan los datos originales y representar gráficamente los resultados obtenidos.
Para ello, creamos un archivo llamado *gausspcasuavizado1.m* (adjuntado en la entrega de esta memoria), que dado un conjunto de entrenamiento, un conjunto con las correspondientes clases del conjunto de entrenamiento, un conjunto de test, un conjunto con las correspondientes clases del conjunto de test y un alfa, va probando proyecciones pca para siempre el mismo suavizado, aplicando en cada iteración la gaussiana y guardándose el error para posteriormente, imprimir en pantalla el gráfico resultante (Figura 1).
En esta gráfica, se puede observar que el porcentaje de error disminuye al aumentar el número de componentes PCA hasta cierto umbral (alrededor de 50 en este caso en concreto) y luego se estabiliza a pesar de aumentar el número de componentes.

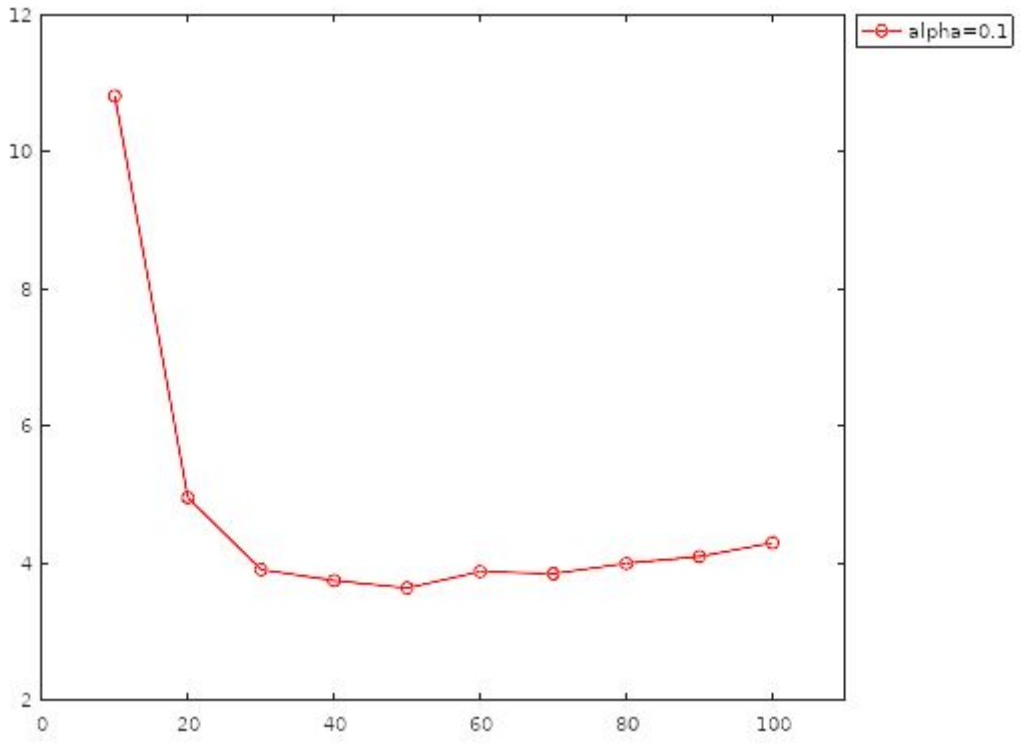

Figura 1- Gráfica dónde X(proyección pca) e Y (%error) con alpha fijo a 1

El segundo ejercicio sugerido (ejercicio 4.2) consiste en realizar lo mismo que en el ejercicio anterior pero probando diferentes valores de alfa.
Para ello, creamos un archivo llamado *gausspcacompleto.m* (adjuntado en la entrega de esta memoria), que dado un conjunto de entrenamiento, un conjunto con las correspondientes clases del conjunto de entrenamiento, un conjunto de test y un conjunto con las correspondientes clases del conjunto de test, va probando proyecciones pca cambiando el valor de alfa, aplicando en cada iteración la gaussiana y guardándose el error para posteriormente, imprimir en pantalla el gráfico resultante (Figura 2).
En esta gráfica en comparación a la Figura 1, observamos que la tasa de error disminuye en función del valor alfa escogido, concretamente cuando mayor alfa, mejores resultados.

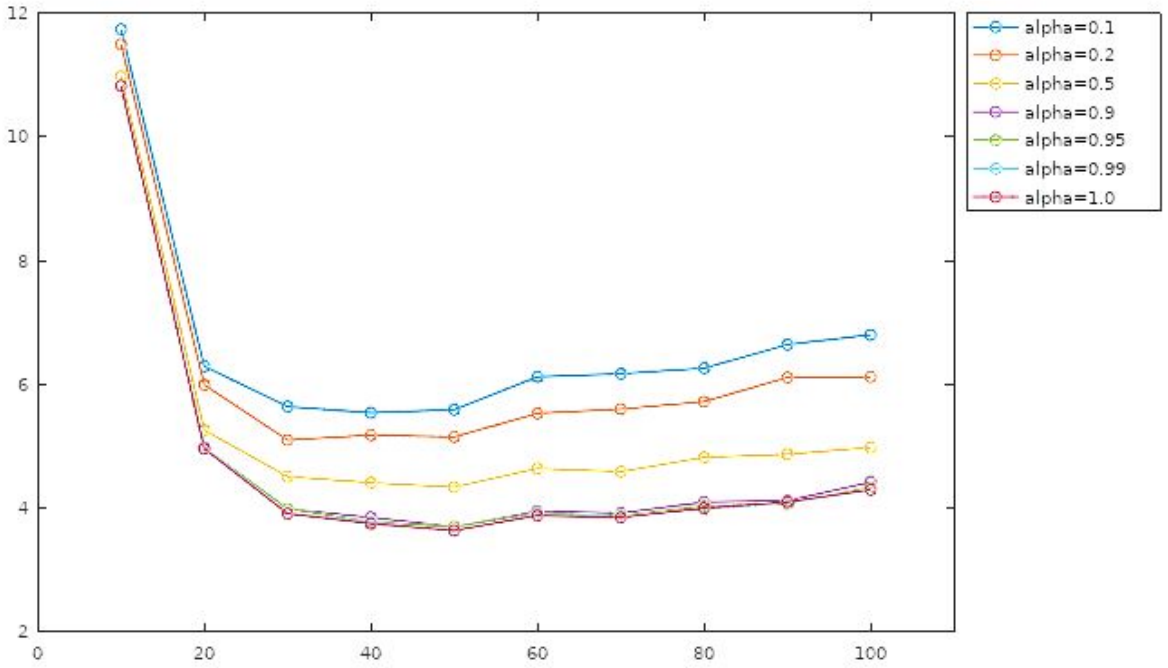

Figura 2- Gráfica dónde X(proyección pca) e Y (%error) con alpha variable

Después de estos ejercicios, se introduce el clasificador basado en mixturas gaussianas y se plantean diversos ejercicios.

En el primer ejercicio, se pide implementar el paso M de la estimación de los parámetros de las mixturas gaussianas. Replicando las fórmulas previamente explicadas en la práctica, las siguientes líneas de código se añaden al archivo *mixgaussian.m*:

```
% M step: parameter update
    pkGc{ic}=sum(zk)/Nc;
    muc=Xc'*zk./sum(zk);
    mu{ic}=muc;
    for k=1:K
     aux=(zk(:,k).*(Xc-muc(:,k)'))'*(Xc-muc(:,k)')/sum(zk(:,k));
     sigma(ic,k)=aux;
          end
```

Entonces, para comprobar el correcto funcionamiento del clasificador de mixturas, probamos a ejecutar mixgaussian.m con sólo un componente y alfa=1, y miramos que el error es el mismo que en una gaussiana normal. La ejecución nos da los siguientes resultados:

```
>> mixgaussian(X,xl,Y,yl,1,1)
 It        oL                  L                    trerr          teerr
--- ----------- ----------- ----- ---------------------------------------
  1       -Inf                -1407.55417       11.77                14.28
  2       -1407.55417         -982.49605              11.77                14.28
  3       -982.49605          -982.49605              11.77                14.28
ans =  14.280

>> gaussian(X,xl,Y,yl,1)
ans =  14.280
```

Como ambos errores son iguales, sabemos que el clasificador de mixturas funciona correctamente.

Finalmente, queremos realizar un experimento donde se evalúa el error de clasificación en función del número de componentes por mixtura dado un número de componentes PCA fijo y un alfa fijo, y representarlo gráficamente.

En *mixgaussian.m*, añadiremos esta línea al paso M para poder suavizar los parámetros:

```
% M step: parameter update
   % HERE YOUR CODE FOR PARAMETER ESTIMATION
        ...
    for k=1:K
     aux=(zk(:,k).*(Xc-muc(:,k)'))'*(Xc-muc(:,k)')/sum(zk(:,k));
     aux2 = alpha * aux + (1-alpha) * eye(D);
     sigma(ic,k)=aux2;
    end
```

Además, creamos una función llamada script.m que dados los sets de training , sus etiquetas, testing y sus etiquetas, devuelve una gráfica (Figura 3) que muestra la tendencia de error respecto al número de componentes

de la mixtura para pca=30. Sabemos que esta gráfica es correcta porque los resultados para 1, 2 y 5 componentes dan el error esperado para la tarea MNIST proyectada a 30 dimensiones (para 1 componente el error es 3.99%, para 2 es 3.72% y para 5 es 2.95%).

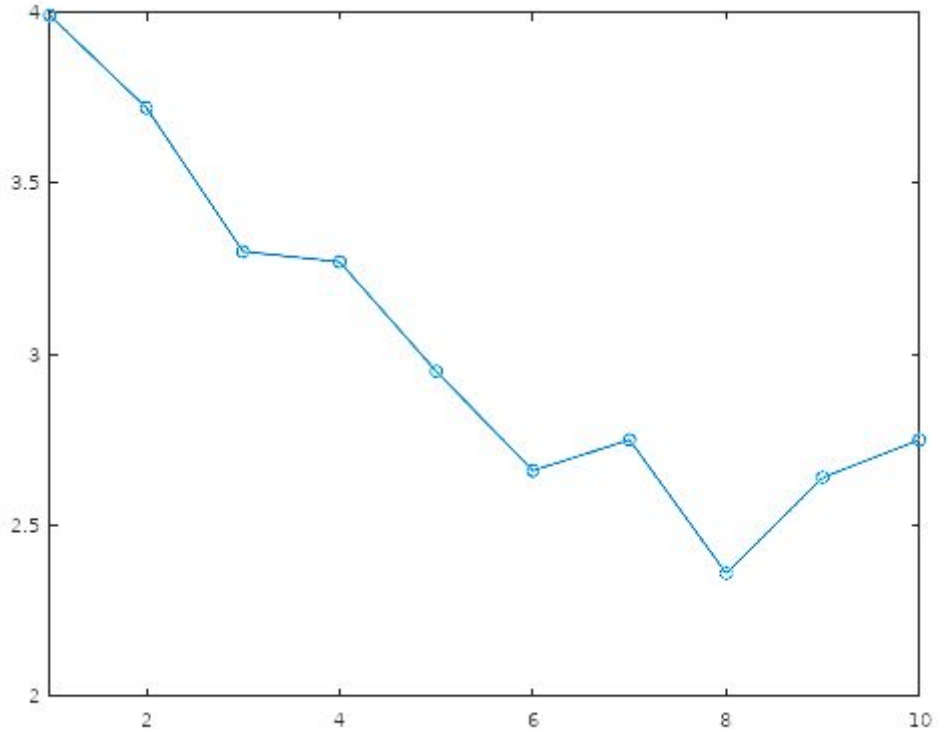

Figura 3- Gráfica donde X(número componentes) e Y (%error) con alpha=0.9 y con los datos proyectados a 30 dimensiones

Además, creamos un script adicional llamado *script2.m* en el que se muestra la tendencia de error con alfa=0.9 donde van variando el número de componentes de la mixtura para diferentes PCA (Figura 4).

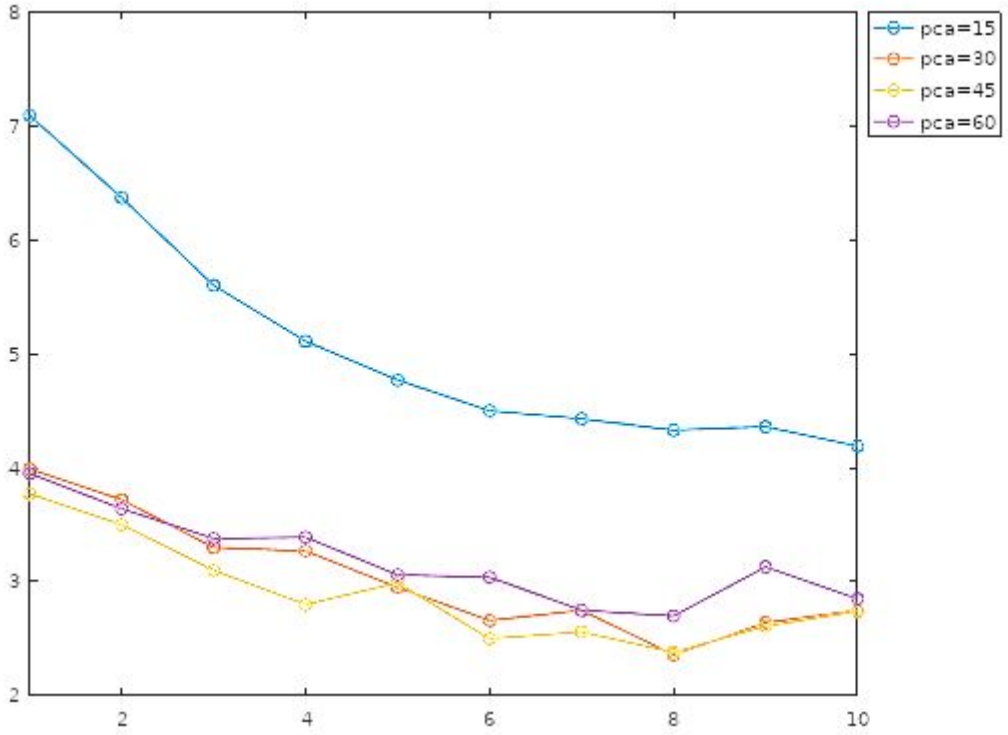

Figura 4- Gráfica donde X(número componentes) e Y (%error) con alpha=0.9 y con los datos proyectados a 15, 30, 45 y 60 dimensiones

Podemos observar que con 15 dimensiones, el error es considerablemente más alto que con 30, 45 o 60. Por lo tanto, es preferible proyectar los datos a 30 dimensiones o más, si no queremos perder información vital. Sin embargo, dado el problema de la dimensionalidad, tampoco nos conviene proyectar a muchas dimensiones, pues como se puede observar con pca=60 el error es mayor que con pca=45.

Además, para poder comprobar mejor los datos, volvemos a ejecutar el script2.m comprobando con más componentes y añadiendo también pca=75 (línea verde).
La primera teoría que confirmamos es que las proyecciones óptimas a pca sin que la maldición de la dimensionalidad nos afecte serían entre 30-60, pues con pca=75 se nos dispara el error.
En cuanto a los componentes, se puede observar que a partir de cierto punto, el error de clasificación aumenta al ser el número de componentes excesivo.

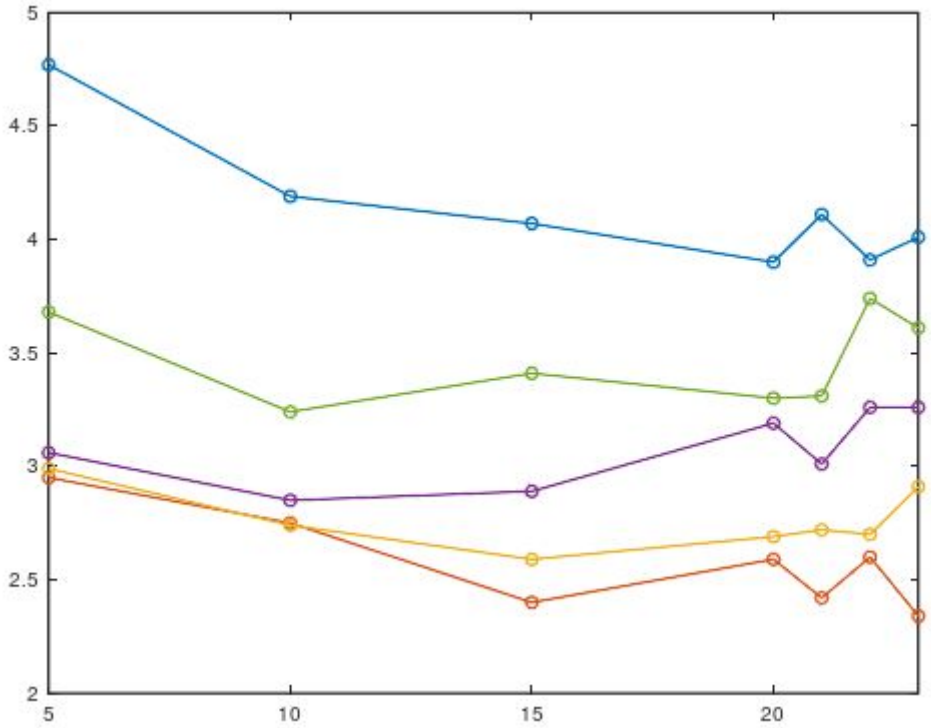

Figura 5- Gráfica donde X(número componentes) e Y (%error) con alpha=0.9 y con los datos proyectados a 15, 30, 45, 60 y 75 dimensiones