

APRENDIZAJE AUTOMÁTICO

Memoria de Mixtura de Gaussianas

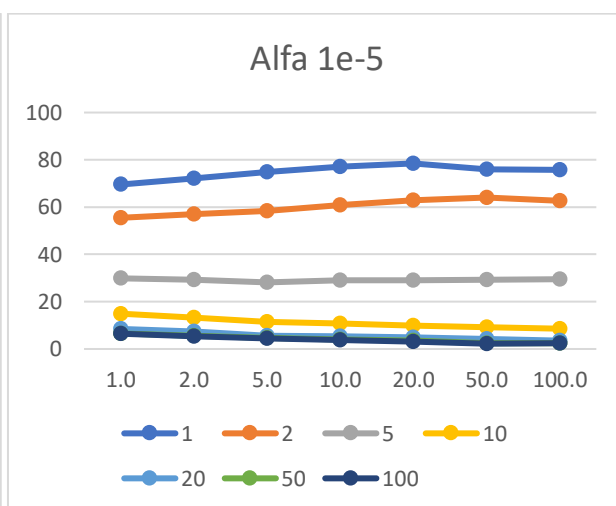
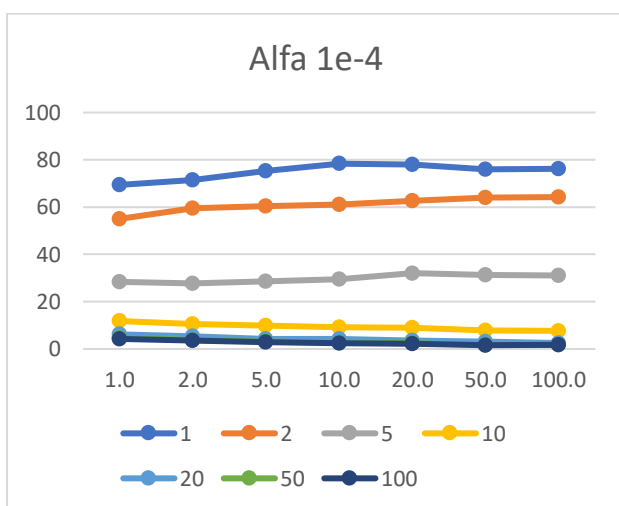
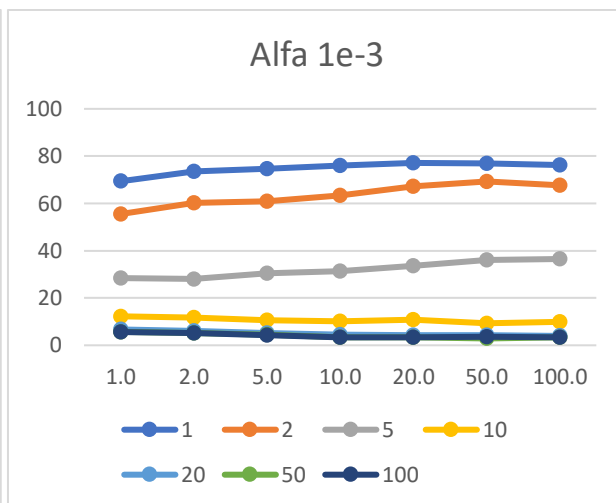
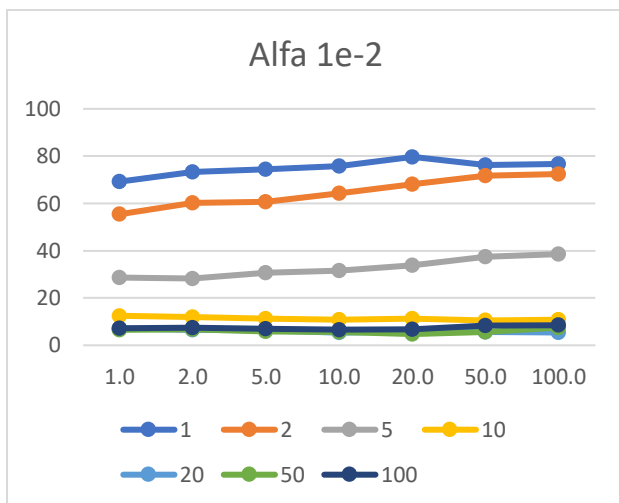
- Fabián Scherle Carboneres
- Lishuang Sun (María)

1. Realizar un experimento para evaluar el error del clasificador en función del número de componentes por mixtura ($K = 1; 2; 5; 10; 20; 50; 100$), para un número de dimensiones PCA variable ($D = 1; 2; 5; 10; 20; 50; 100$) y probando algunos de los valores de alfa de suavizado flat smoothing que mejores resultados han proporcionado en el clasificador gaussiano.

Para realizar dicho experimento creamos un script `pca+mixgaussian-exp.m` al cual le pasamos distintos valores de dimensionalidades para *PCA*, diferentes valores de *k* componentes y los mejores resultados de alfas obtenidos al probar el clasificador gaussiano. Para las alfas partimos de los siguientes resultados de evaluación:

Alfas	Error
9.0e-01	18.683
5.0e-01	13.550
2.0e-01	12.200
1.0e-01	11.967
1.0e-02	10.000
1.0e-03	6.383
1.0e-04	4.267
1.0e-05	6.317
1.0e-06	14.083
1.0e-07	18.933
1.0e-08	19.550
1.0e-09	19.650
#0.0e+00	19.667

Se puede apreciar que para las alfas con valor $1e-2$, $1e-3$, $1e-4$ y $1e-5$ se obtienen los menores porcentajes de error para un clasificador gaussiano. Por lo que los usamos para probar el clasificador de mixtura de gaussianas, dando los siguientes resultados:



2. Utilizar los valores óptimos de los parámetros del clasificador para entrenar y evaluar un clasificador final en los conjuntos oficiales MNIST de entrenamiento y test, respectivamente.

Tomando como base los resultados del apartado anterior, se obtiene como valor óptimo para alfa: $1e-04$, dimensiones: 100 y componentes: 50 (porcentaje de error de 1,6).

Adicionalmente se puede apreciar que en dicha alfa para K componentes con valores [20, 50, 100] y para D dimensiones [50, 100], se obtienen pequeños porcentajes de error. Por lo que usamos dichos valores para evaluar el clasificador:

Dimensiones	Componentes	Error	Intervalos de confianza
50,0	20.0	1,9	[1,632; 2,168]
50,0	50.0	1,87	[1,604; 2,136]
50,0	100.0	1,62	[1,373; 1,867]
100,0	20.0	2,06	[1,782; 2,338]
100,0	50.0	1,92	[1,651; 2,189]
100,0	100.0	1,71	[1,456; 1,964]

Comparando el resultado óptimo obtenido con valor de porcentaje de error de 1,62, con las tasas de error de la MNIST web (<http://yann.lecun.com/exdb/mnist/>) comprobamos que es mejor que el valor aportado por la referencia de “LeCun et al. 1998” con un clasificador cuadrático con 40 PCA y se acerca a la tasa del clasificador de SVM con kernel gaussiano.

3. Modificar el código de aprendizaje del clasificador con mixturas de Gaussianas con el fin de mejorar el error en validación.

Hemos optado por la terminación temprana.

Nuestro propósito es detener el clasificador con mixturas gaussianas en función del avance del error Y en las iteraciones del algoritmo EM.

Definimos una variable auxiliar prevErrY para guardarnos el error de Y de la iteración anterior, en caso de que no se mejore el error, esta variable no se actualiza e incrementamos un contador, el cual si llega a un número máximo pasado por parámetro se detiene el algoritmo.

```
% Código añadido:
if it == 1
    prevErrY = errY;
end
if it > 1 && prevErrY < errY
    contErrY=contErrY+1;
end
if it > 1 && prevErrY > errY
    prevErrY = errY;
    contErrY = 0;
end
if contErrY >= maxIt
    errY = prevErrY;
    break;
end
until ((L-oL)/abs(oL) < epsilon)
end
```

Para probar dicha modificación optamos por usar el valor óptimo obtenido en el apartado anterior, variando el parámetro máximo de iteraciones sin mejora de error Y:

dimension	component	alpha	max it	dv-err	intervalos
50	100	1.0e-04	1	1.600	[1.354,1.846]
50	100	1.0e-04	2	1.600	[1.354,1.846]
50	100	1.0e-04	3	1.600	[1.354,1.846]

A la vista de los resultados podemos apreciar que se ha decrementado el porcentaje de error en un 0.02%, decrementando a su vez el coste temporal del algoritmo.