**Máster de Visión Artificial**

**Asignatura: Reconocimiento de Patrones**

**Práctica 1: Clasificadores generativos**

**Orión García Gallardo**

**DNI: 48330747L**

INTRODUCCIÓN

El objetivo principal de esta práctica es entender los clasificadores paramétricos y no paramétricos. Para ello se evaluará el rendimiento de diferentes clasificadores:

1. Clasificador paramétrico basado en Gaussianas.
2. Clasificador no-paramétrico basado en los k vecinos más próximos (K-NN).
3. Clasificador no-paramétrico basado en histogramas.
4. Clasificador no-paramétrico basado en ventanas de Parzen.

Para realizar esta evaluación se van a suponer únicamente dos dimensiones en el vector de características de los datos. Se utilizarán tres clases equiprobables para evitar descompensación entre ellas y los datos tendrán una distribución gaussiana y conocida. Con el objetivo de optimizar cada clasificador se van a entrenar con conjuntos de 50, 200 y 1000 datos por clase con dos conjuntos de gaussianas diferentes.

Este documento se divide en varias secciones. En la primera de ellas se describe el método desarrollado. A continuación se explica las pruebas realizadas sobre la muestra y se evalúan los resultados obtenidos. Y por último se comenta las conclusiones obtenidas y los flancos que quedan abiertos después de la finalización de esta práctica.

MÉTODO DESARROLLADO

Cada uno de los métodos mencionados tiene algún parámetro que ha de ser introducido previamente, aunque algunos se llamen no paramétricos. Con el objetivo de maximizar el rendimiento de cada clasificador se va a realizar una búsqueda del valor óptimo de los parámetros. La técnica que se va a emplear para realizar esta búsqueda va a ser una validación cruzada de 5 grupos (5-fold). Dicha técnica consiste en dividir los datos de muestra en K subconjuntos. Uno de los subconjuntos se utiliza como datos de prueba y el resto (K-1) como datos de entrenamiento. Dicho proceso se repite durante k iteraciones, donde como conjunto de datos de prueba se selecciona un subconjunto distinto en cada iteración. La elección de qué datos hay en cada subconjunto se va a hacer de manera aleatoria (usando la función *randperm* de MATLAB). El último paso será clasificar un conjunto de datos nuevos para comprobar la eficiencia de clasificación con ese parámetro seleccionado. Con el objetivo de comparar esta eficiencia entre los distintos métodos se va a devolver el error y la matriz de confusión obtenida de evaluar estos nuevos datos.

Clasificador no-paramétrico basado en histogramas.

Este método de clasificación consiste en hacer una división del espacio de trabajo en volúmenes constantes. A continuación se crea un histograma de este espacio de trabajo teniendo en cuenta el número de datos por clase que hay en cada volumen. Para normalizar estos valores se suelen dividir entre el número total de datos de la clase. Es por ello que, para este método, es especialmente interesante que la distribución de datos sea equiprobable entre las clases. Finalmente, dado un nuevo dato perteneciente a un volumen determinado se clasificará en la clase cuyo valor de histograma, o probabilidad, sea mayor en ese volumen.

En este método el parámetro a estimar por validación cruzada será el tamaño de los volúmenes para crear el histograma. Suponiendo conocidos los valores mínimo y máximo de cada dimensión, el problema es análogo al de seleccionar el número de divisiones que vamos a hacer en cada una de estas dimensiones del espacio de trabajo.

En esta práctica se va a acotar los posibles valores de cada dimensión a valores entre 0 y 50. Dicho acotamiento supone que cuando se recibe un dato con valores menor que el mínimo o mayor que el máximo se acumulará a los valores del histograma para los volúmenes del mínimo o máximo respectivamente. Por otro lado, como posibles números de divisiones por dimensión se van a elegir los siguientes:

DIVISIONES\_HISTOGRAMA = 3:30;

La validación cruzada en este caso se encarga de encontrar, entre estos posibles valores en los que se dividen las dimensiones del histograma, él que mejor rendimiento obtenga a la hora de clasificar. Para facilitar los cálculos a realizar y ahorrar memoria, en lugar de crear un histograma de probabilidades por clase, se ha construido un solo histograma donde cada celda contendrá la etiqueta de la clase que mayor probabilidad tenga en ese volumen del espacio de trabajo. Para saber el valor de estas probabilidades es necesario calcular la probabilidad a priori de cada clase, que en el caso de este estudio es igual en cada caso por tener una distribución equiprobable de los datos en cada clase. En caso de empate de probabilidades se clasifica la división con la etiqueta más frecuente entre las en las divisiones vecinas. Dicho algoritmo de construcción de histogramas se emplea tanto cuando se construye a partir del conjunto de datos de entrenamiento en la validación cruzada como cuando se crea para clasificar los datos nuevos y evaluar el clasificador.

Clasificador no-paramétrico basado en los k vecinos más próximos (K-NN).

Este método de clasificación consiste en clasificar cada dato teniendo en cuenta la clase de los k datos más cercanos. El número de vecinos k a tener en cuenta será el parámetro a optimizar, con el objetivo de que la clasificación final sea lo más eficiente posible. Como probar con un número muy grande de posibles valores de k podría ser muy costoso, limitamos esta búsqueda a unos valores concretos:

VALORES\_K\_EN\_KNN = [1, 3, 5, 7, 11, 13, 17, 19, 21, 23, 29, 31];

Sobre estos valores de k se realiza una la validación cruzada según se ha explicado previamente. A continuación se ordenan los valores k según la eficiencia de clasificación para seleccionar el que mejor rendimiento haya obtenido.

Uno de los casos excepcionales de este método se puede dar cuando hay un empate de etiquetas de los k vecinos. Es decir, si k = 5, y se tienen 2 etiquetas de la clase 1, 2 etiquetas de la clase 2 y una etiqueta de la clase 3 no se sabría si clasificar el dato en la clase 1 o 2. Para resolver este problema se han almacenado también las distancias de los k vecinos al dato dado. De tal modo que, en caso de empate, se selecciona la clase del dato más próximo. Se ha escogido este entre los múltiples criterios de desempate que existen porque es sencillo, rápido y fácil de implementar.

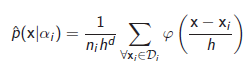
Otro carácter importante a tener en cuenta con este algoritmo es la medida de distancia seleccionada para determinar quiénes son los k vecinos. En este trabajo se ha seleccionado la norma o distancia Euclídea entre dos datos (usando la función *norm* de MATLAB).

Clasificador no-paramétrico basado en ventanas de Parzen

Este método de clasificación consiste en dividir el espacio de trabajo en regiones que no están en posiciones fijas. Usando una función kernel dada, esta técnica aproxima la distribución de un conjunto de entrenamiento a través de una combinación de kernels centrados en los puntos a observar. El ancho de la ventana de estos kernels será el parámetro a optimizar en este algoritmo. Al igual que en los clasificadores anteriores, los valores posibles de ancho de ventana en este algoritmo vendrán determinados de antemano y se corresponderán con:

VALORES\_H\_EN\_PARZEN = [0.5, 1, 2];

La estimación de la función de probabilidad asociada a una determinada clase se calcula usando técnicas de kernel. Esto se traduciría a que, dado un ancho de ventana h, esta función de probabilidad sería:



Donde la aproximación más sencilla de la función sería contar los puntos que están de x a un radio de distancia menor que h, lo cual vendría determinado por:



Sin embargo, es probado que una aproximación que suaviza más la función de probabilidad y, en este caso, da mejores resultados es suponer la gaussianidad de los datos. Esto sería equivalente a definir:



La búsqueda del valor de h óptimo, como en los casos anteriores, se realiza a través de validación cruzada de 5 grupos. En este caso lo que se buscará es el valor de h que satisfaga:



No obstante, es importante remarcar que, para ese trabajo, en este productorio sólo se incluyen los valores de la función de probabilidad en los que la estimación de la clase de un determinado dato de test coincidía con la clase real del dato. Dicho de otro modo, sólo se incluyen los valores en el productorio cuando, para un determinado dato de test su función de probabilidad sea mayor en su clase real. Es por ello que a la hora de elegir el mejor h lo primero que se tendrá en cuenta es la tasa de acierto. En caso de que la mayor tasa de acierto se obtenga por más de un valor se seleccionaría el valor de h cuyo productorio total sea mayor.

EVALUACIÓN

KPERCENTAGES =

0.9800 1.0000

0.9867 3.0000

0.9867 5.0000

0.9800 7.0000

0.9800 11.0000

0.9800 13.0000

0.9800 17.0000

0.9800 19.0000

0.9733 21.0000

0.9733 23.0000

0.9733 29.0000

0.9733 31.0000

Elapsed time is 5.103186 seconds.

Error\_KNN =

0.2140

MatrizConfusion\_KNN =

412 0 263

102 724 50

120 0 829

ans =

K-NN - 50 datos - K = 3

NPERCENTAGES =

0.9733 3.0000

0.9667 4.0000

0.9667 5.0000

0.9867 6.0000

0.9600 7.0000

0.9533 8.0000

0.9867 9.0000

0.9667 10.0000

0.9533 11.0000

0.9467 12.0000

0.9400 13.0000

0.9333 14.0000

0.9200 15.0000

0.8867 16.0000

0.9067 17.0000

0.8800 18.0000

0.8867 19.0000

0.8933 20.0000

0.8800 21.0000

0.8467 22.0000

0.8667 23.0000

0.8333 24.0000

0.8533 25.0000

0.8333 26.0000

0.8200 27.0000

0.8267 28.0000

0.8133 29.0000

0.8267 30.0000

Elapsed time is 2.476161 seconds.

Error\_HIST =

0.2396

MatrizConfusion\_HIST =

629 10 36

129 747 0

276 148 525

ans =

Hist - 50 datos - N = 6

Elapsed time is 2.289770 seconds.

Error\_GAUSS =

0.0672

MatrizConfusion\_GAUSS =

585 0 90

33 811 32

13 0 936

HPERCENTAGES =

0.9933 0.5000 0.0000

0.9867 1.0000 0.0000

0.9867 2.0000 0.0000

Elapsed time is 60.871218 seconds.

Error\_PARZEN =

0.0856

MatrizConfusion\_PARZEN =

586 8 81

46 796 34

45 0 904

KPERCENTAGES =

0.9867 1.0000

0.9917 3.0000

0.9933 5.0000

0.9917 7.0000

0.9933 11.0000

0.9917 13.0000

0.9933 17.0000

0.9917 19.0000

0.9917 21.0000

0.9917 23.0000

0.9900 29.0000

0.9900 31.0000

Elapsed time is 14.912667 seconds.

Error\_KNN =

0.2244

MatrizConfusion\_KNN =

409 0 266

107 742 27

160 1 788

ans =

K-NN - 200 datos - K = 5

NPERCENTAGES =

0.9817 3.0000

0.9867 4.0000

0.9900 5.0000

0.9917 6.0000

0.9867 7.0000

0.9917 8.0000

0.9883 9.0000

0.9917 10.0000

0.9900 11.0000

0.9833 12.0000

0.9817 13.0000

0.9750 14.0000

0.9783 15.0000

0.9867 16.0000

0.9717 17.0000

0.9750 18.0000

0.9717 19.0000

0.9717 20.0000

0.9767 21.0000

0.9667 22.0000

0.9700 23.0000

0.9717 24.0000

0.9750 25.0000

0.9683 26.0000

0.9617 27.0000

0.9633 28.0000

0.9550 29.0000

0.9500 30.0000

Elapsed time is 2.284601 seconds.

Error\_HIST =

0.2800

MatrizConfusion\_HIST =

619 5 51

148 728 0

237 259 453

ans =

Hist - 200 datos - N = 8

Elapsed time is 1.557073 seconds.

Error\_GAUSS =

0.0400

MatrizConfusion\_GAUSS =

662 0 13

33 836 7

46 1 902

HPERCENTAGES =

0.9933 0.5000 0.0000

0.9933 1.0000 0.0000

0.9917 2.0000 0

Elapsed time is 275.444299 seconds.

Error\_PARZEN =

0.0588

MatrizConfusion\_PARZEN =

666 0 9

49 817 10

76 3 870

KPERCENTAGES =

0.9903 1.0000

0.9917 3.0000

0.9923 5.0000

0.9917 7.0000

0.9907 11.0000

0.9913 13.0000

0.9903 17.0000

0.9907 19.0000

0.9903 21.0000

0.9903 23.0000

0.9903 29.0000

0.9897 31.0000

Elapsed time is 181.680958 seconds.

Error\_KNN =

0.1628

MatrizConfusion\_KNN =

406 0 269

69 787 20

49 0 900

ans =

K-NN - 1000 datos - K = 5

NPERCENTAGES =

0.9723 3.0000

0.9790 4.0000

0.9850 5.0000

0.9833 6.0000

0.9873 7.0000

0.9860 8.0000

0.9893 9.0000

0.9817 10.0000

0.9923 11.0000

0.9903 12.0000

0.9883 13.0000

0.9920 14.0000

0.9910 15.0000

0.9887 16.0000

0.9897 17.0000

0.9883 18.0000

0.9853 19.0000

0.9900 20.0000

0.9887 21.0000

0.9850 22.0000

0.9883 23.0000

0.9860 24.0000

0.9867 25.0000

0.9883 26.0000

0.9857 27.0000

0.9853 28.0000

0.9877 29.0000

0.9843 30.0000

Elapsed time is 4.428813 seconds.

Error\_HIST =

0.2840

MatrizConfusion\_HIST =

582 0 93

131 745 0

243 243 463

ans =

Hist - 1000 datos - N = 11

Elapsed time is 1.364079 seconds.

Error\_GAUSS =

0.0160

MatrizConfusion\_GAUSS =

669 0 6

17 857 2

14 1 934

HPERCENTAGES =

0.9923 0.5000 0

0.9937 1.0000 0

0.9913 2.0000 0

Elapsed time is 3141.658577 seconds.

Error\_PARZEN =

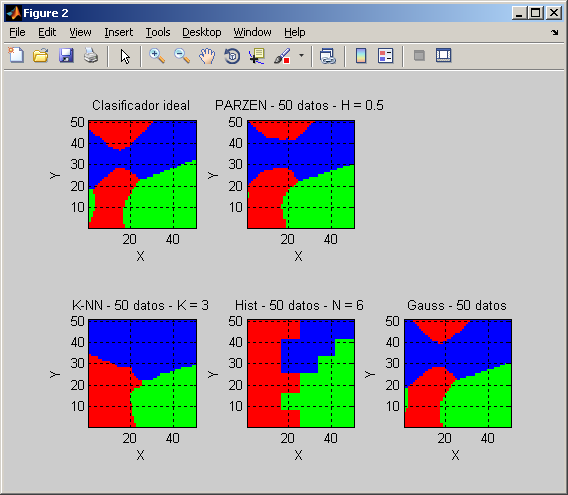
0.0228

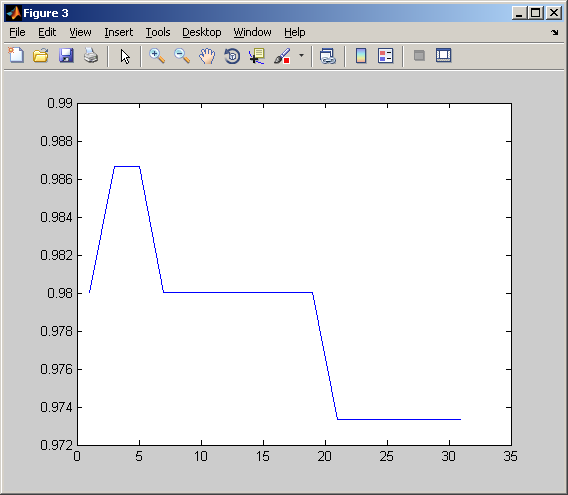
MatrizConfusion\_PARZEN =

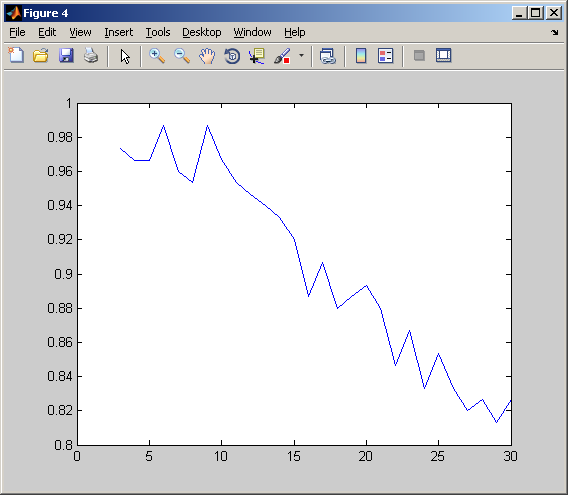
637 2 36

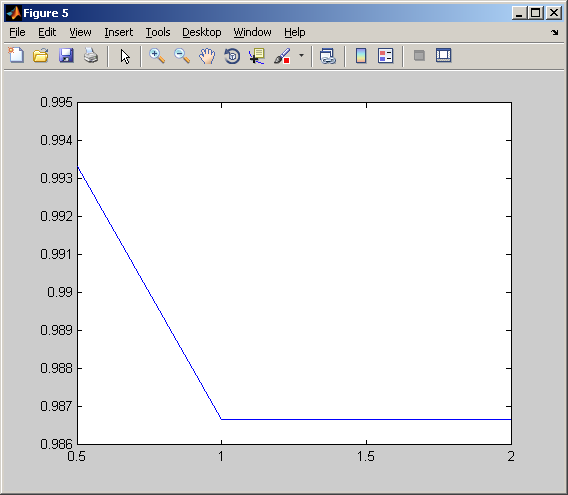
18 857 1

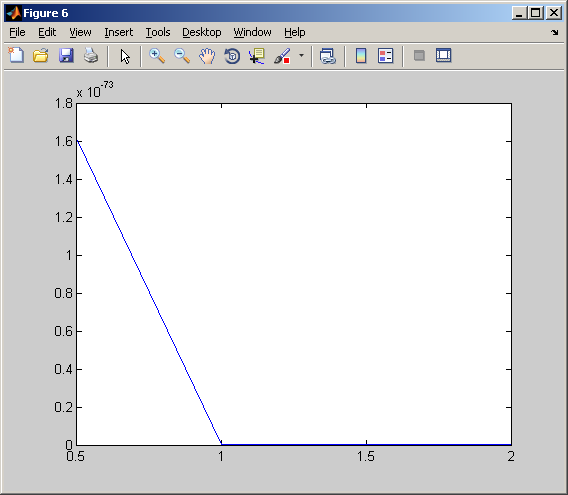
0 0 949

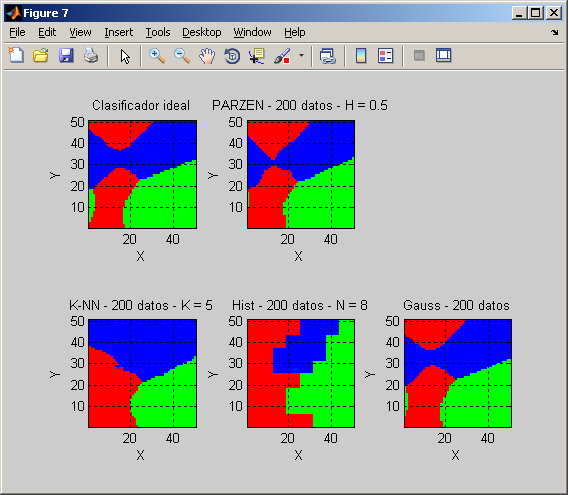


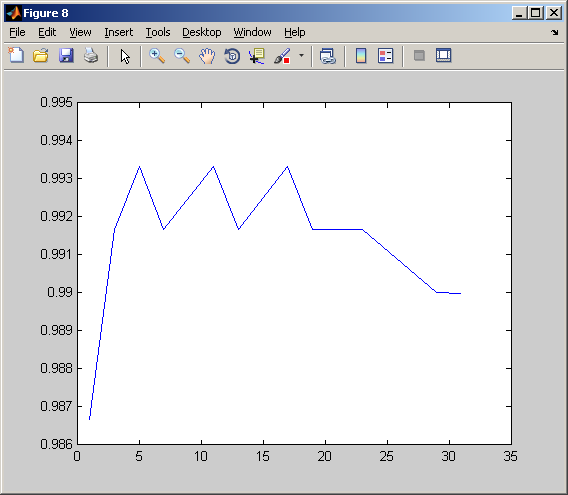


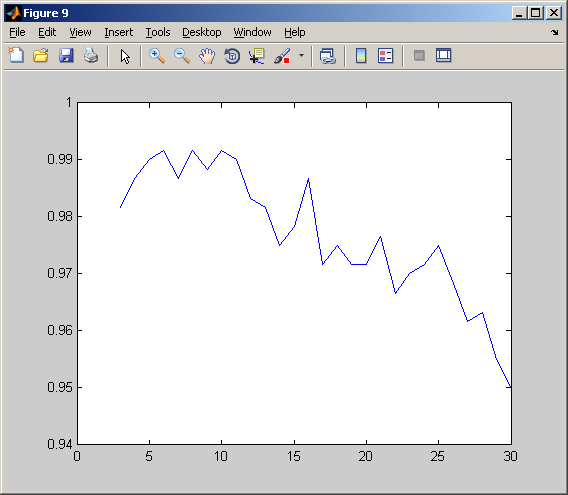


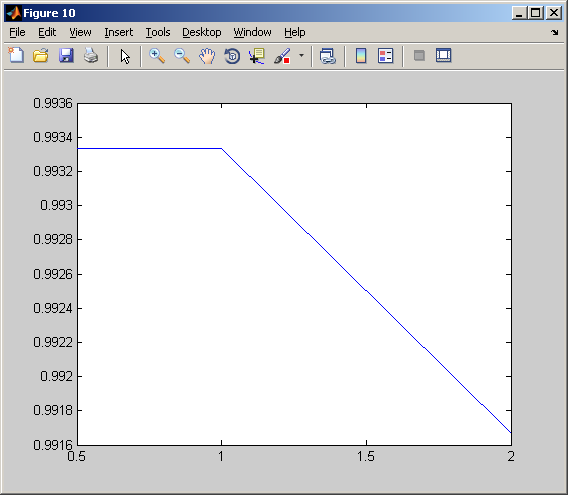


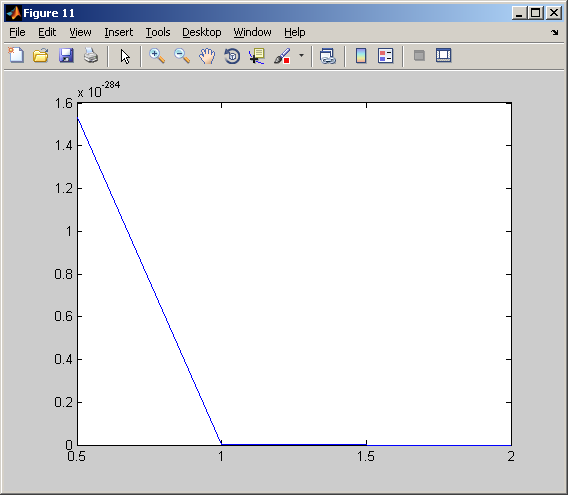


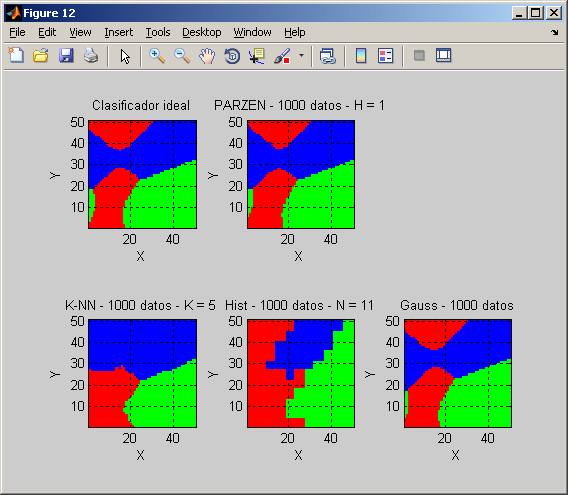


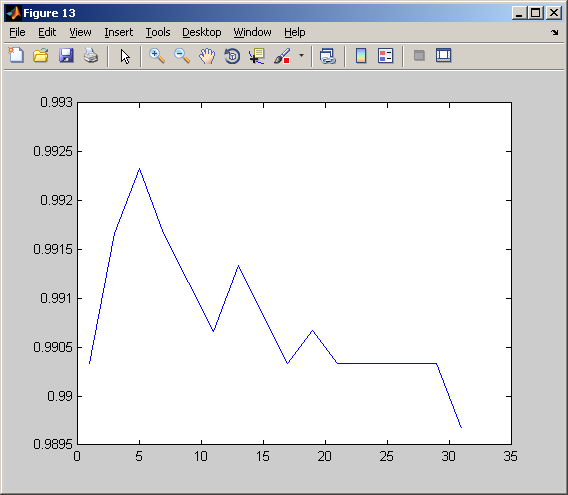


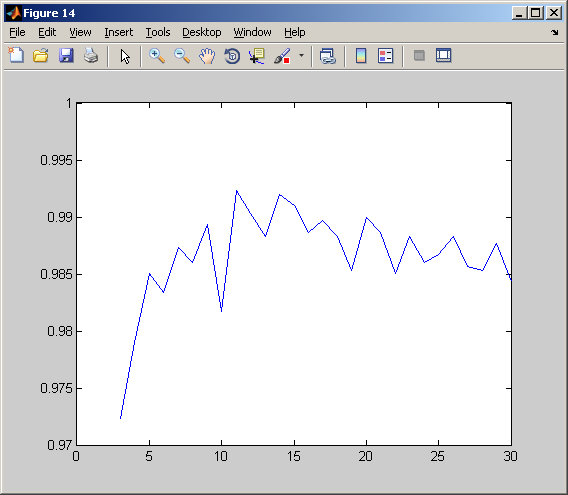


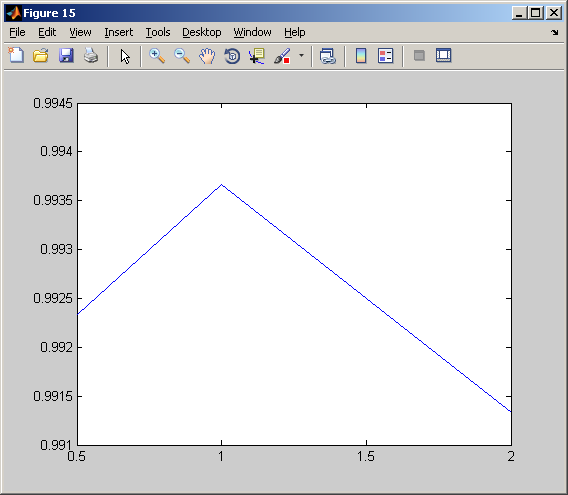


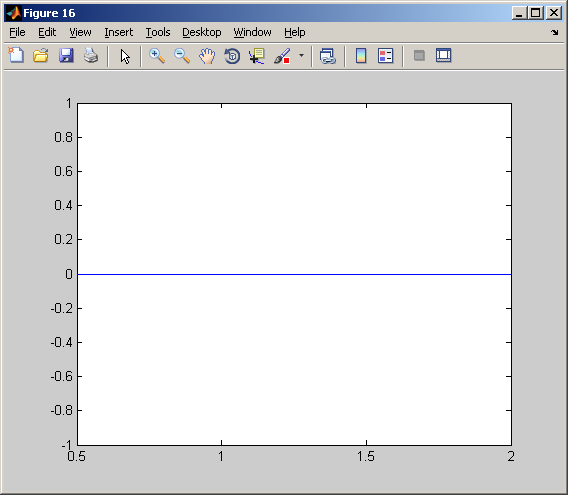












CONCLUSIONES