

Prácticas - Aprendizaje Automático

Alumnos:

- **Luis Alberto Álvarez Zavaleta**
- **David Arnal García**

Introducción

El principal objetivo de este proyecto de prácticas ha sido la implementación y evaluación de diversos clasificadores estudiados en teoría sobre la tarea real de reconocimiento de dígitos manuscritos de *MNIST*.

Este objetivo principal de esta memoria se desglosa en los siguientes objetivos:

- Evaluar el clasificador de mixtura de gaussianas y su interacción con la técnica de reducción de dimensionalidad *PCA* en la tarea *MNIST*.
- Evaluar el clasificador basado en *SVM* en la tarea *MNIST*.
- Evaluar un clasificador basado en redes neuronales en la tarea *MNIST*.

Estos tres puntos han sido los objetivos principales de nuestro proyecto que los hemos ido realizando poco a poco en cada entregable, evaluando así cada clasificador basado en diferentes técnicas, la *mixtura de gaussianas*, la máquina de vectores soporte, así como el clasificador basado en redes neuronales, que nos han permitido poder compararlos con los resultados del *reconocimiento de dígitos manuscritos de MNIST*.

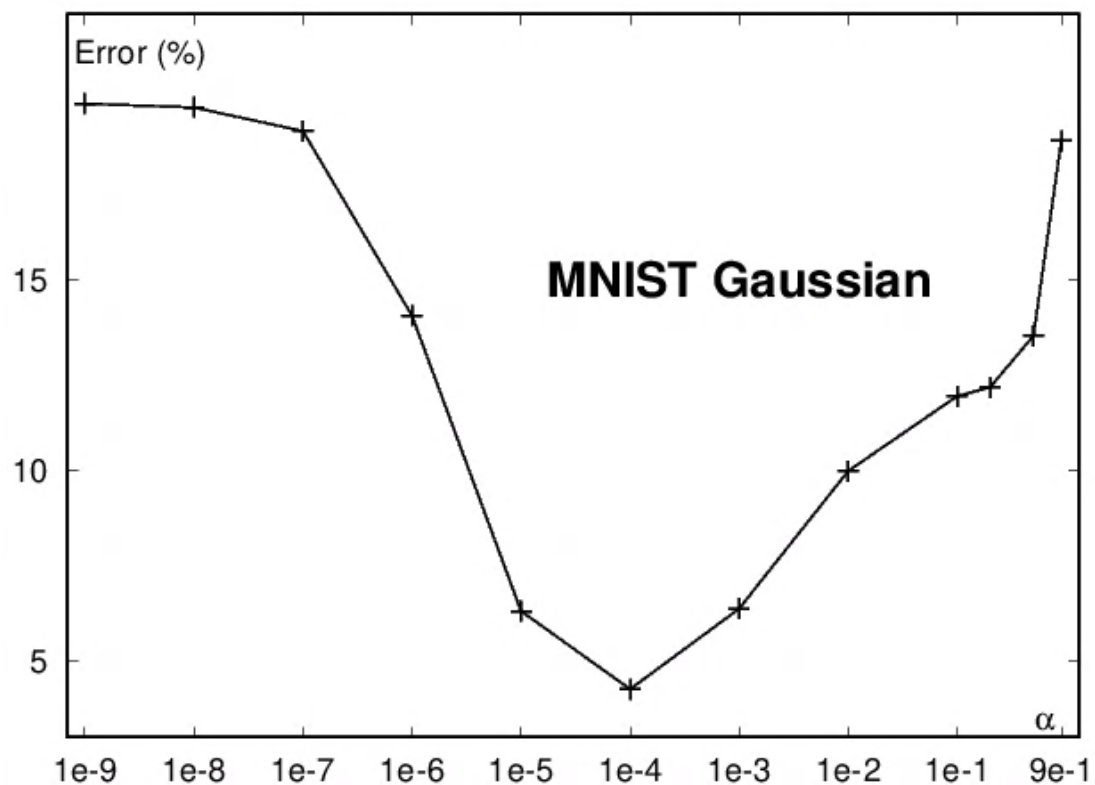
Entregable I:

Mixturas de

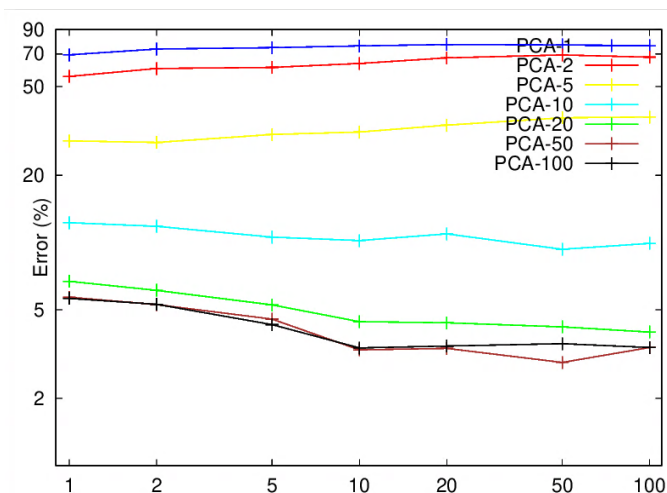
Gaussianas

1.

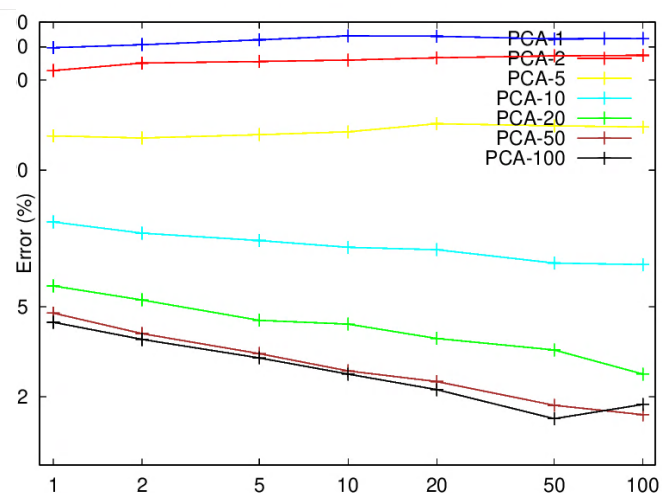
error	9e-1	5e-1	2e-1	1e-1	1e-2	1e-3	1e-4	1e-5	1e-6	1e-7	1e-8	1e-9
alpha	18.683	13.55	12.2	11.967	10	6.383	4.267	6.317	14.083	18.933	19.55	19.65



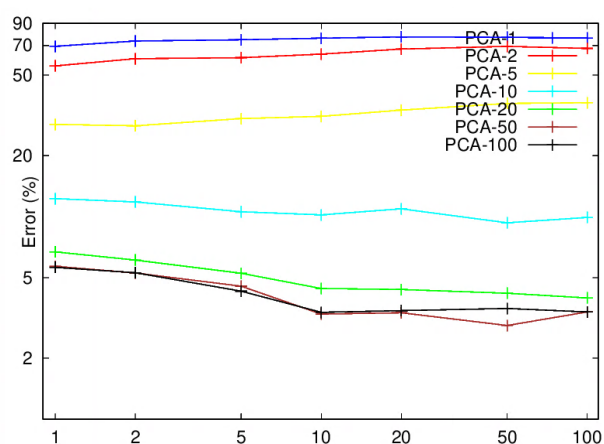
Como podemos observar gracias a la tabla y al gráfico, el valor por *flat smoothing* que mejor resultado da es el **1e-4**, y siendo sus valores más cercanos, el **1e-3** y el **1e-5** los que más se acercan al mejor valor. Los demás valores, ya muestran un error más pronunciado.



$\alpha = 1e-3$



$\alpha = 1e-4$



$\alpha = 1e-5$

Observando las gráficas, podemos concluir que la que mejor resultado obtiene es la de $\alpha = 1e-4$, PCA = 100 y K = 50 con un error de **1.6%**, siendo estos sus parámetros óptimos para el clasificador por mixtura de *gaussianas*. Aunque los valores con PCA = 50 y K = 100 obtienen **1.667%**, por lo que lo tendremos en cuenta para el siguiente apartado.

2.

Alpha	PCA	Ks	Error	Intervalo de confianza
1e-4	50	100	1.62	[1.618, 1.62247]
1e-4	100	50	1.92	[1.917, 1.92269]

Como podemos observar, el mejor resultado lo obtenemos con una reducción de dimensionalidad, $D = 50$, con 100 distribuciones gaussianas, $K_s = 100$, y con un valor de suavizado, $\alpha = 1e-4$.

Además, obtenemos que, con una confianza de 95%, podemos asegurar que su error se encontrará entre el intervalo [1.618, 1.62247].

Comparándolo con el clasificador gaussiano, y el clasificador *PCA + gaussiano* con 200 dimensiones, se puede ver como el resultado es mejor, ya que hemos mejorado nuestro clasificador reduciendo el error en un 2.49% al utilizar las mixturas de *Gaussianas*, y un 1.68% comparado con los resultados de *MNIST*.

Alpha	Error
1e-4 (sin PCA)	4.18
1e-4 ($D = 200$)	4.11

3.

Implementaremos la terminación temprana para el algoritmo EM en el que haremos que termine cuando el error de validación/test no disminuya de una iteración a la siguiente. Para ello, modificaremos *mixgaussian.m* de la siguiente manera:

```
it = 0;
lasterrY = inf;
printf(" It          oL          L  errX  errY\n");
printf(" ---  -----  -----  ----  ----\n");
do
    oL = L;
    L = 0;
    % For each class
    for c = classes'

% Classification of training and evaluation sets and error estimation
    [~, idX] = max(gX');
    errX = mean(classes(idX) != xl) * 100;
    [~, idY] = max(gY');
    errY = mean(classes(idY) != yl) * 100;
    it = it + 1;
    printf("%3d %11.5f %11.5f %5.2f %5.2f\n", it, oL, L, errX, errY);

    if (errY >= lasterrY)
        errY = lasterrY;
        break;
    endif

    lasterrY = errY;
until ((L - oL) / abs(oL) < epsilon)
end
```

Al evaluar esta nueva implementación en los conjuntos oficiales de *MNIST* para los valores óptimos, $PCA = 50$ y $K = 100$, obtenemos los siguientes resultados.

Alpha	PCA	Ks	Error	Intervalo de confianza
1e-4	50	100	1.6	[1.598, 1.6]

Entregable 2:

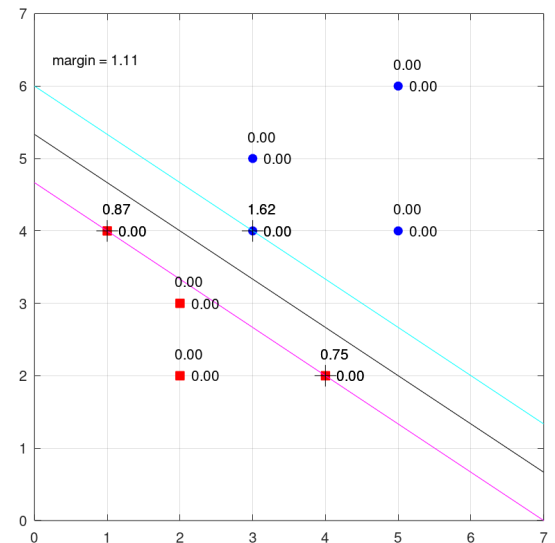
Máquinas de

vectores soporte

Ejercicio 1

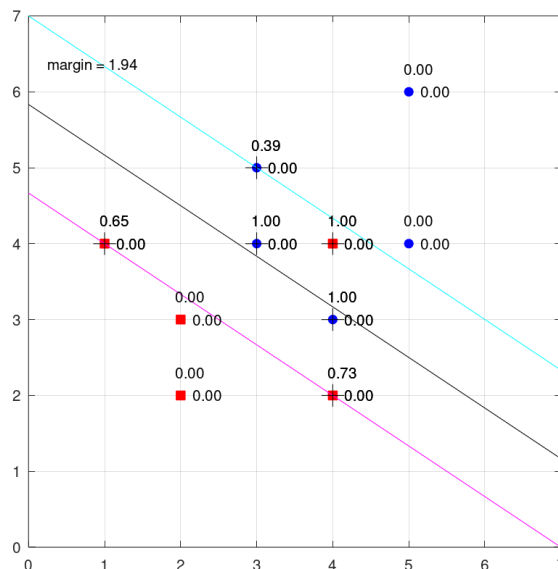
Para el caso separable con (*trSep.dat*, *trSeplabels.dat*) y $C = 1000$:

- Coeficientes de *Lagrange* = [0.87472 0.74989 -1.62461].
- Vector soporte
 - (1, 1) -> 1
 - (2, 1) -> 4
 - (3, 1) -> 3
 - (1, 2) -> 4
 - (2, 2) -> 2
 - (3, 2) -> 4
- Vector de pesos = [-0.99955 -1.49978].
- Umbral de la función discriminante = 7.9987.
- Margen = 1.1097.
- Función discriminante -> $y = -(0.66647)x - (-5.3332)$.



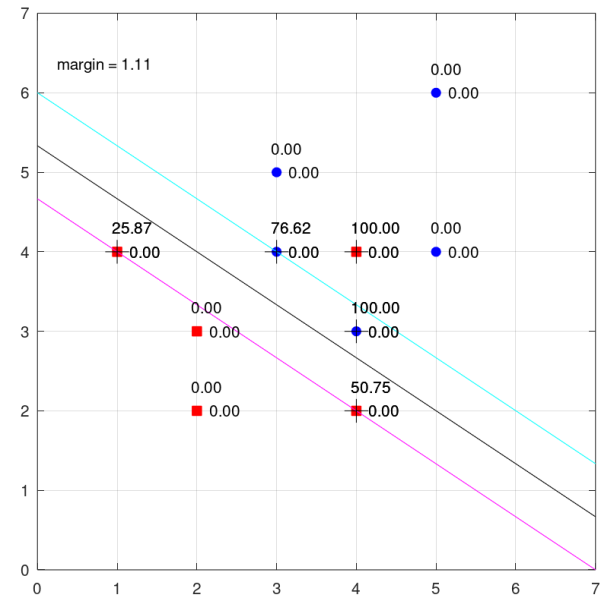
Para el caso no separable con y $C = 1$:

- Coeficientes de *Lagrange* = [0.65306 0.73472 1.0 -1.0 -0.38778 -1.0].
- Vector soporte
 - (1, 1) -> 1
 - (2, 1) -> 4
 - (3, 1) -> 4
 - (4, 1) -> 3
 - (5, 1) -> 3
 - (6, 1) -> 4
 - (1, 2) -> 4
 - (2, 2) -> 2
 - (3, 2) -> 4
 - (4, 2) -> 4
 - (5, 2) -> 5
 - (6, 2) -> 3
- Vector de pesos = [-0.57139 -0.85722].
- Umbral de la función discriminante = 5.0003.
- Margen = 1.9414.
- Función discriminante -> $y = -(0.66657)x - (-5.8331)$.



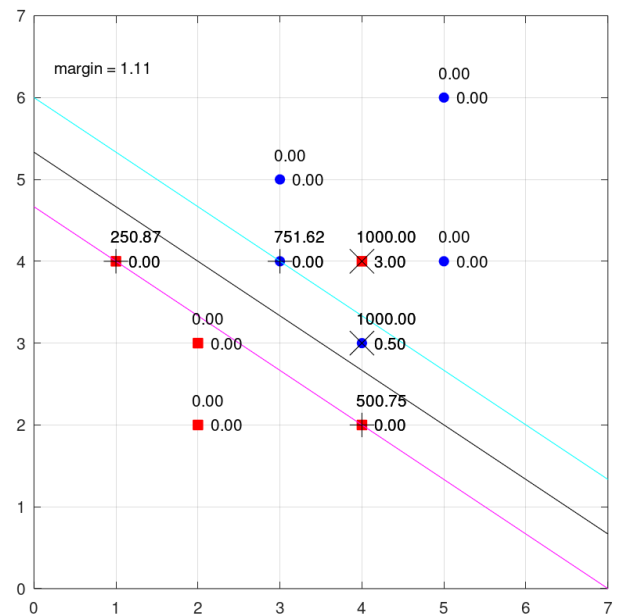
Para el caso no separable con γ y $C = 100$:

- Coeficiente de *Lagrange* = [25.875 50.750 100.0 -76.625 -100.0].
- Vector soporte
 - (1, 1) -> 1
 - (2, 1) -> 4
 - (3, 1) -> 4
 - (4, 1) -> 3
 - (5, 1) -> 4
 - (1, 2) -> 4
 - (2, 2) -> 2
 - (3, 2) -> 4
 - (4, 2) -> 4
 - (5, 2) -> 3
- Vector de pesos = [-0.99955 -1.49978].
- Umbral = 7.9987.
- Margen = 1.1097.
- Función discriminante -> $y = -(0.66647)x - (-5.3332)$



Para el caso no separable con γ y $C = 1000$:

- Coeficiente de *Lagrange* = [250.87 500.75 1000.00 -751.62 -1000.00].
- Vector soporte
 - (1, 1) -> 1
 - (2, 1) -> 4
 - (3, 1) -> 4
 - (4, 1) -> 3
 - (5, 1) -> 4
 - (1, 2) -> 4
 - (2, 2) -> 2
 - (3, 2) -> 4
 - (4, 2) -> 4
 - (5, 2) -> 3
- Vector de pesos = [-0.99955 -1.49977].
- Umbral = 7.9986.
- Margen = 1.1097.
- Función discriminante -> $y = -(0.66647)x - (-5.3332)$.



Ejercicio 2

Ver anexo para visualizar todos los resultados obtenidos por pca+svm-exp.m

PCA = 50

T	C	G	acierto	intervalo
2	100	0.1	0.950000	0.001744
2	100	0.01	0.961667	0.001536
2	100	0.001	0.946667	0.001798
2	100	0.0001	0.921667	0.002150
2	100	1,00E-05	0.906667	0.002328

PCA = 100

T	C	G	acierto	intervalo
2	100	0.1	0.910000	0.002290
2	100	0.01	0.968333	0.001401
2	100	0.001	0.928333	0.002064
2	100	0.0001	0.928333	0.002064
2	100	1,00E-05	0.906667	0.002328

PCA = 200

T	C	G	acierto	intervalo
2	10	0.1	0.860000	0.002776
2	10	0.01	0.968333	0.001401
2	10	0.001	0.933333	0.001996
2	10	0.0001	0.911667	0.002271
2	10	1,00E-05	0.778333	0.003324

Los mejores resultados que hemos obtenido han sido con un *Kernel Gaussiano* (T=2) aplicando $PCA = 50$ con $C = 100$ y $PCA = 100$ con $C = 10$ y un valor *gamma* $G = 0.01$. Por lo que usaremos estos valores para la parte de evaluación , además. Hemos podido observar que, al aplicar PCA para la solución óptima y proyectando a pocas dimensiones, empeora el valor óptimo.

Ejercicio 3

PCA = 100

T	C	G	acierto	intervalo
2	10	0.01	98.370	[98.112 98.618]
2	100	0.01	98.350	[98.100 98.600]

PCA = 200

T	C	G	acierto	intervalo
2	10	0.01	98.280	[98.025 98.535]
2	100	0.01	98.280	[98.025 98.535]

Al ejecutar *pca+svm-eva.m* para un *Kernel Gaussiano* ($T=2$) aplicando $PCA = 50$ o $PCA = 100$ y con $C = 100$ o $C = 10$ y un valor *gamma* $G = 0.01$. Hemos observado que los parámetros que mejor resultado nos ofrecen son $PCA = 100$, $T = 2$, $C = 10$ y $G = 0.01$ con un error de clasificación del 1.63% al compararlo con el de *MNIST*, que es de un 1.4%.

SVM, Gaussian Kernel	none	1.4
----------------------	------	-----

Entregable 3:

Redes neuronales

Ejercicio 1

nº de épocas	Error en entrenamiento	Error en validación
10	0.49%	2.37%
20	0.15%	1.83%
30	0.40%	2.03%
40	0.00%	1.42%
50	0.00%	1.40%
60	0.05%	1.67%
70	0.00%	1.58%
80	0.00%	1.53%
90	0.00%	1.53%
100	0.00%	1.57%
150	0.00%	1.62%
200	0.00%	2.52%
300	0.00%	1.43%
400	0.00%	1.42%
500	0.00%	1.52%

Al utilizar diferentes números de *epochs*, el mejor resultado obtenido a partir de los *epochs* usados ha sido de 50. A partir de este valor, empeora el error de validación por el sobreentrenamiento del clasificador, al ser este nuestro valor optimo, usaremos este parámetro para los siguientes ejercicios.

Ejercicio 2

3-layer NN 300 – 100 – 10

	Error de validación
ReLU	2.48%
Sigmoid	2.05%
Tanh	1.57%
Mish	2.35%

2-layer NN 300 - 10

	Error de validación
ReLU	1.98%
Sigmoid	2.02%
Tanh	1.85%
Mish	2.13%

3-layer NN 500 – 150 - 10

	Error de validación
ReLU	1.35%
Sigmoid	1.87%
Tanh	1.73%
Mish	1.50%

2-layer NN 800 - 10

	Error de validación
ReLU	2.30%
Sigmoid	2.08%
Tanh	1.57%
Mish	1.62%

3-layer NN 500 – 300 - 10

	Error de validación
ReLU	1.40%
Sigmoid	1.83%
Tanh	1.83%
Mish	1.53%

2-layer NN 1000 - 10

	Error de validación
ReLU	1.58%
Sigmoid	1.47%
Tanh	1.88%
Mish	2.10%

Los mejores valores obtenidos han sido con redes neuronales de dos capas. Para este ejercicio hemos tomado algunas de las configuraciones que están disponibles en la web de *MNIST* para que podamos tener una estimación del clasificador comparado con los que hay disponibles en *MNIST*. Hemos observado que, al utilizar más de 3 capas en lugar de 2, los resultados mejoran. La función *ReLU* es la que mejores resultados ha dado, en general, para estas redes de 3 capas.

Ejercicio 3

3-layer NN	Error de entrenamiento	Error de validación	Intervalo de error
500 – 300 - 10	0.17%	1.85%	[1.59, 2.11]

3-layer NN, 500+300 HU, softmax, cross entropy, weight decay	none	1.53
--	------	------

Podemos observar que, para una red neuronal de 3 capas con 500 + 300 *HUD*, aplicando entropía cruzada, hemos obtenido un error de validación de 1.85%, una tasa de error muy parecida a la obtenida a la de la web de *MNIST*, considerando de que además esta usa *weight*, *decay* y *softmax*.

3-layer NN	Error de entrenamiento	Error de validación	Intervalo de error
500 – 150 - 10	0.07%	1.8%	[1.54, 2.06]

3-layer NN, 500+150 hidden units	none	2.95
----------------------------------	------	------

Por otra parte, para una red neuronal de 3 capas con 500 + 150 *HUD*, aplicando entropía cruzada, nos da un error del 1.8%, que comparado con el 2.95% de *MNIST*, podemos observar que nuestro clasificador es un 1% mejor que el obtenido en *MNIST*, aunque en este no se aplica entropía cruzada.

Conclusiones

El error de clasificación con mixturas de *Gaussianas* nos ha dado 1.6%.

Alpha	PCA	Ks	Error	Intervalo de confianza
1e-4	50	100	1.6	[1.598, 1.6]

El error de clasificación con máquinas de vectores soportes nos ha dado 1.63%.

PCA = 100

T	C	G	acierto	intervalo
2	10	0.01	98.370	[98.112 98.618]
2	100	0.01	98.350	[98.100 98.600]

El error de clasificación con redes neuronales nos ha dado 1.8%.

3-layer NN	Error de entrenamiento	Error de validación	Intervalo de error
500 – 150 - 10	0.07%	1.8%	[1.54, 2.06]

Al comparar las diferentes técnicas, la que mejor resultados nos ha dado ha sido la de mixturas de *Gaussianas*. Esto no nos indica que esta técnica sea la mejor, sino que es la que mejor resultado nos ha dado con los parámetros que hemos estudiado. Aunque podríamos haber refinado este resultado explorando más parámetros de las distintas técnicas utilizadas, así como ampliar la búsqueda a la hora de encontrar los resultados óptimos de la fase de entrenamiento con el objetivo de obtener un menor error de clasificación para el problema de *reconocimiento de dígitos manuscritos de MNIST*.

ANEXO

PCA = 50

T	C	acierto	intervalo
0	1	0.925000	0.002108

T	C	D	acierto	intervalo
1	1	1	0.920000	0.002171
1	1	2	0.961667	0.001536
1	1	3	0.950000	0.001744
1	1	4	0.926667	0.002086
1	1	5	0.893333	0.002470

T	C	G	acierto	intervalo
2	1	0.1	0.945000	0.001824
2	1	0.01	0.956667	0.001629
2	1	0.001	0.910000	0.002290
2	1	0.0001	0.773333	0.003350
2	1	1,00E-05	0.123333	0.002631

T	C	G	acierto	intervalo
3	1	0.1	0.503333	0.004001
3	1	0.01	0.916667	0.002212
3	1	0.001	0.888333	0.002520
3	1	0.0001	0.508333	0.004000
3	1	1,00E-05	0.123333	0.002631

T	C	acierto	intervalo
0	10	0.911667	0.002271

T	C	D	acierto	intervalo
1	10	1	0.923333	0.002129
1	10	2	0.951667	0.001716
1	10	3	0.958333	0.001599
1	10	4	0.941667	0.001875
1	10	5	0.941667	0.001875

T	C	G	acierto	intervalo
2	10	0.1	0.950000	0.001744
2	10	0.01	0.960000	0.001568
2	10	0.001	0.920000	0.002171
2	10	0.0001	0.906667	0.002328
2	10	1,00E-05	0.775000	0.003341

T	C	G	acierto	intervalo
3	10	0.1	0.488333	0.004000
3	10	0.01	0.913333	0.002251
3	10	0.001	0.921667	0.002150
3	10	0.0001	0.888333	0.002520
3	10	1,00E-05	0.508333	0.004000

T	C	acierto	intervalo
---	---	---------	-----------

0	100	0.905000	0.002346
---	-----	----------	----------

T	C	D	acierto	intervalo
1	100	1	0.908333	0.002309
1	100	2	0.953333	0.001688
1	100	3	0.958333	0.001599
1	100	4	0.940000	0.001900
1	100	5	0.930000	0.002042

T	C	G	acierto	intervalo
2	100	0.1	0.950000	0.001744
2	100	0.01	0.961667	0.001536
2	100	0.001	0.946667	0.001798
2	100	0.0001	0.921667	0.002150
2	100	1,00E-05	0.906667	0.002328

T	C	G	acierto	intervalo
3	100	0.1	0.503333	0.004001
3	100	0.01	0.888333	0.002520
3	100	0.001	0.930000	0.002042
3	100	0.0001	0.921667	0.002150
3	100	1,00E-05	0.888333	0.002520

PCA = 100

T	C	acierto	intervalo
0	1	0.906667	0.002328

T	C	D	acierto	intervalo
1	1	1	0.923333	0.002129
1	1	2	0.953333	0.001688
1	1	3	0.896667	0.002436
1	1	4	0.518333	0.003998
1	1	5	0.246667	0.003449

T	C	G	acierto	intervalo
2	1	0.1	0.906667	0.002328
2	1	0.01	0.961667	0.001536
2	1	0.001	0.913333	0.002251
2	1	0.0001	0.775000	0.003341
2	1	1,00E-05	0.123333	0.002631

T	C	G	acierto	intervalo
3	1	0.1	0.508333	0.004000
3	1	0.01	0.918333	0.002191
3	1	0.001	0.893333	0.002470
3	1	0.0001	0.508333	0.004000
3	1	1,00E-05	0.123333	0.002631

T	C	acierto	intervalo
0	10	0.888333	0.002520

T	C	D	acierto	intervalo
1	10	1	0.913333	0.002251
1	10	2	0.960000	0.001568

1	10	3	0.958333	0.001599
1	10	4	0.923333	0.002129
1	10	5	0.756667	0.003433

T	C	G	acierto	intervalo
2	10	0.1	0.910000	0.002290
2	10	0.01	0.965000	0.001471
2	10	0.001	0.930000	0.002042
2	10	0.0001	0.906667	0.002328
2	10	1,00E-05	0.776667	0.003333

T	C	G	acierto	intervalo
3	10	0.1	0.505000	0.004001
3	10	0.01	0.906667	0.002328
3	10	0.001	0.923333	0.002129
3	10	0.0001	0.893333	0.002470
3	10	1,00E-05	0.508333	0.004000

T	C	acierto	intervalo
0	100	0.890000	0.002504

T	C	D	acierto	intervalo
1	100	1	0.906667	0.002328
1	100	2	0.958333	0.001599
1	100	3	0.961667	0.001536
1	100	4	0.943333	0.001850
1	100	5	0.933333	0.001996

T	C	G	acierto	intervalo
2	100	0.1	0.910000	0.002290
2	100	0.01	0.968333	0.001401
2	100	0.001	0.928333	0.002064
2	100	0.0001	0.928333	0.002064
2	100	1,00E-05	0.906667	0.002328

T	C	G	acierto	intervalo
3	100	0.1	0.505000	0.004001
3	100	0.01	0.876667	0.002631
3	100	0.001	0.913333	0.002251
3	100	0.0001	0.923333	0.002129
3	100	1,00E-05	0.893333	0.002470

PCA = 200

T	C	acierto	intervalo
0	1	0.901667	0.002383

T	C	D	acierto	intervalo
1	1	1	0.916667	0.002212
1	1	2	0.905000	0.002346
1	1	3	0.391667	0.003906
1	1	4	0.138333	0.002763
1	1	5	0.123333	0.002631

T	C	G	acierto	intervalo
2	1	0.1	0.846667	0.002883
2	1	0.01	0.960000	0.001568
2	1	0.001	0.913333	0.002251
2	1	0.0001	0.776667	0.003333
2	1	1,00E-05	0.123333	0.002631

T	C	G	acierto	intervalo
3	1	0.1	0.526667	0.003995
3	1	0.01	0.923333	0.002129
3	1	0.001	0.895000	0.002453
3	1	0.0001	0.510000	0.004000
3	1	1,00E-05	0.123333	0.002631

T	C	acierto	intervalo
0	10	0.898333	0.002418

T	C	D	acierto	intervalo
1	10	1	0.923333	0.002129
1	10	2	0.965000	0.001471
1	10	3	0.906667	0.002328
1	10	4	0.430000	0.003961
1	10	5	0.165000	0.002970

T	C	G	acierto	intervalo
2	10	0.1	0.860000	0.002776

2	10	0.01	0.968333	0.001401
2	10	0.001	0.933333	0.001996
2	10	0.0001	0.911667	0.002271
2	10	1,00E-05	0.778333	0.003324

T	C	G	acierto	intervalo
3	10	0.1	0.513333	0.003999
3	10	0.01	0.915000	0.002232
3	10	0.001	0.926667	0.002086
3	10	0.0001	0.895000	0.002453
3	10	1,00E-05	0.510000	0.004000

T	C	acierto	intervalo
0	100	0.898333	0.002418

T	C	D	acierto	intervalo
1	100	1	0.900000	0.002400
1	100	2	0.953333	0.001688
1	100	3	0.965000	0.001471
1	100	4	0.906667	0.002328
1	100	5	0.491667	0.004000

T	C	G	acierto	intervalo
2	100	0.1	0.860000	0.002776
2	100	0.01	0.965000	0.001471
2	100	0.001	0.936667	0.001949
2	100	0.0001	0.928333	0.002064

2	100	1,00E-05	0.911667	0.002271
---	-----	----------	----------	----------

T	C	G	acierto	intervalo
3	100	0.1	0.516667	0.003999
3	100	0.01	0.883333	0.002569
3	100	0.001	0.926667	0.002086
3	100	0.0001	0.926667	0.002086
3	100	1,00E-05	0.895000	0.002453