Práctica 4. Máquinas de vectores soporte

Introducción a los modelos computacionales

Ingeniería Informática (Rama Computación)

Escuela Politécnica Superior

Universidad de Córdoba

Curso 2016 – 2017

Autor: Rubén Medina Zamorano - 46068656R

Email: *i32mezar@uco.es*

ÍNDICE

1. INTRODUCCIÓN

Se trata de la cuarta práctica para la asignatura Introducción a los Modelos Computacionales. En ella, procederemos a elaborar una memoria que describa el problema a tratar, analice el pseudocódigo y el funcionamiento del algoritmo utilizado y por último, responderemos a las preguntas necesarias.

2. DESCRIPCIÓN DE LAS MÁQUINAS DE VECTORES SOPORTE

Una máquina de vector de soporte es un método de aprendizaje automático que intenta tomar datos de entrada y los clasifica en una de dos categorías en donde se utilizan datos de entrada y salida.

Dado un conjunto de puntos, subconjunto de un conjunto mayor (espacio), en el que cada uno de ellos pertenece a una de dos posibles categorías, un algoritmo basado en SVM construye un modelo capaz de predecir si un punto nuevo (cuya categoría desconocemos) pertenece a una categoría o a la otra.

Como en la mayoría de los métodos de clasificación supervisada, los datos de entrada (los puntos) son vistos como un vector p-dimensional (una lista ordenada de p números).

La SVM busca un hiperplano que separe de forma óptima a los puntos de una clase de la de otra, que eventualmente han podido ser previamente proyectados a un espacio de dimensionalidad superior.

En ese concepto de "separación óptima" es donde reside la característica fundamental de las SVM: este tipo de algoritmos buscan el hiperplano que tenga la máxima distancia (margen) con los puntos que estén más cerca de él mismo. Por eso también a veces se les conoce a las SVM como clasificadores de margen máximo. De esta forma, los puntos del vector que son etiquetados con una categoría estarán a un lado del hiperplano y los casos que se encuentren en la otra categoría estarán al otro lado.

3. CUESTIONES

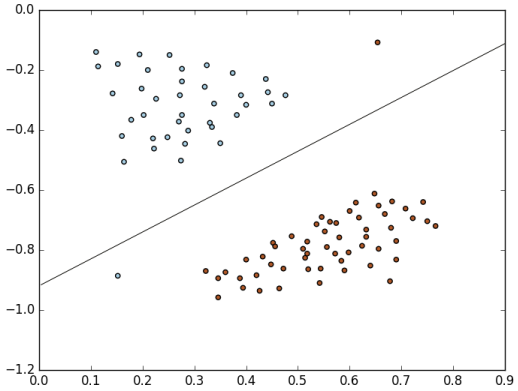
Pregunta [1]: Abre este script y explica su contenido. Podrás ver que se utiliza el primer dataset de ejemplo y se realiza la representación gráfica del SVM. Comenta que tipo de kernel se está utilizando y cuáles son los parámetros de entrenamiento.

El funcionamiento base de este script consiste en abrir un fichero de datos con formato libsvm. Entrenar el modelo de vectores soporte mediante una función rbf con parámetros C=1 y gamma=100, por defecto. En diversos casos, se podrá utilizar un clasificador lineal, y variaremos los parámetros para observar que ocurre.

Primer dataset de ejemplo

Pregunta [2]: Intuitivamente, ¿que hiperplano crees que incurrirá en un menor error de test en la tarea de separar las dos clases de puntos?.

A simple vista, teniendo la nube de puntos representada podemos deducir que el hiperplano más simple y con menos error sería una función lineal. Un hiperplano que divide la zona superior y una inferior con una pequeña pendiente positiva.ç



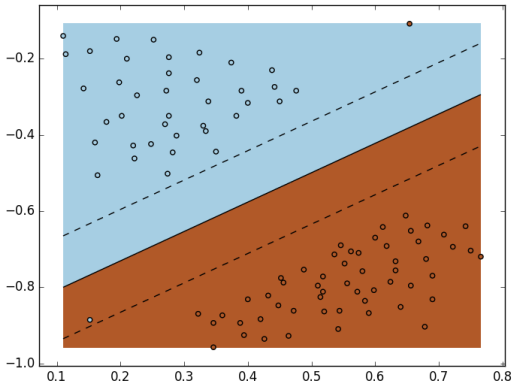
Aunque con una RBF podemos a obtener el 100% de clasificación en train, puede que con el modelo más complejo, en test no funcione bien. Por ello, es mejor utilizar un clasificador lineal obviando los 2 patrones considerados como outlier.

Pregunta [3]: Debes de configurar el script para que se ejecute una SVM lineal (es decir, que el kernel sea lineal).

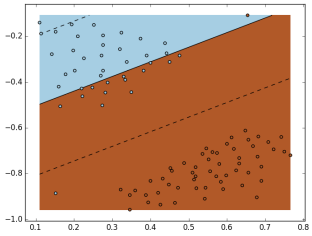
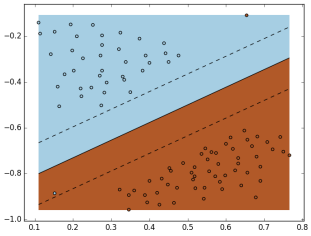
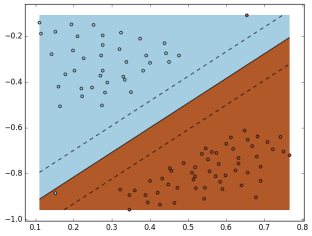
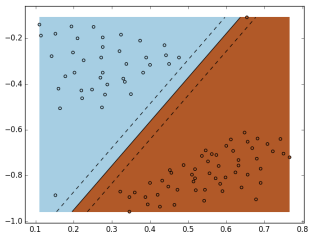
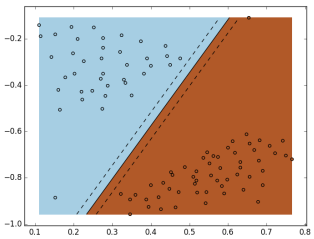
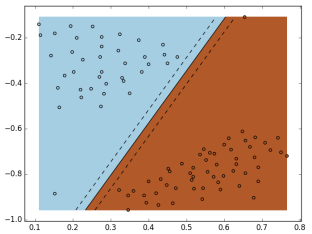
Para ejecutar el script de SVM con función lineal basta con editar la función SVC con el kernel='linear'.

```
svm_model = svm.SVC(kernel='linear',C=1,gamma=100)
```

Podemos ver, que el clasificador lineal es más simple, sin embargo hay 2 patrones que los clasifica incorrectamente. Teniendo en cuenta la complejidad de cada función, es recomendable utilizar la más interpretable, en nuestro caso la lineal.



Pregunta [4]: Modifica el script para que se prueben varios valores de C, en concreto, C ∈{10e−2, 10e−1, 10e0, 10e1, 10e2, 10e3, 10e4}. Observa que sucede, explica porqué y escoge el valor más adecuado.

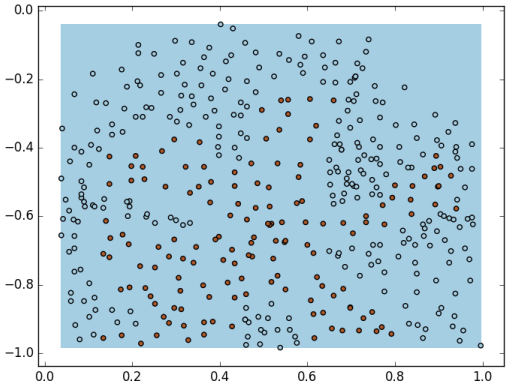
10e-2	10e-1	10e0
		
10e1	**10e2**	**10e3 / 10e4**
		

Se puede apreciar como para valores pequeños de C se tiene un mayor margen, que en los primeros casos no llega a clasificar correctamente, y existen patrones en el margen que se descartarían. Sin embargo, para valores mayores en C, apenas se aprecia el margen y para patrones de test, puede que no clasifique correctamente. Además, con un valor muy elevado de C se estaría considerando una estructura hard, en la que no puede haber patrones en el margen. Entonces, lo recomendable sería **10e0 o 10e-1** con una estructura soft en la que se permita la existencia de patrones en el margen.

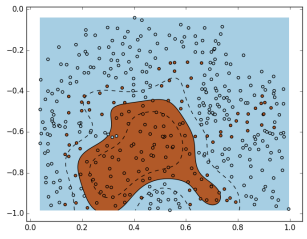
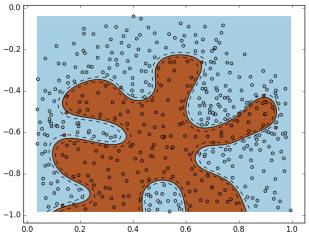
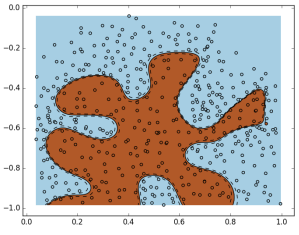
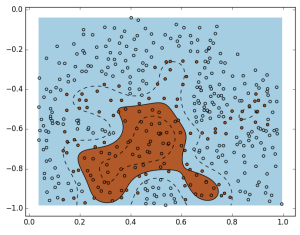
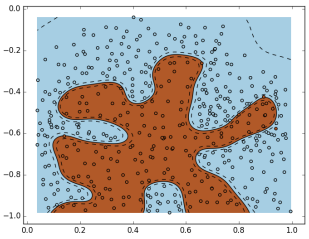
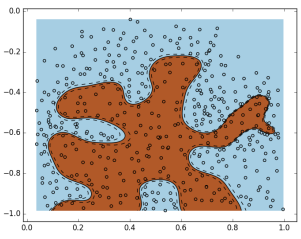
Segundo dataset de ejemplo

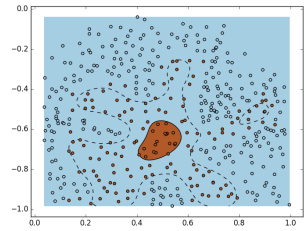
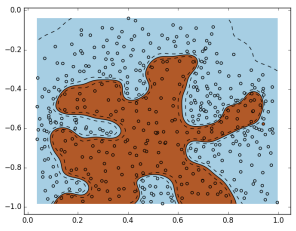
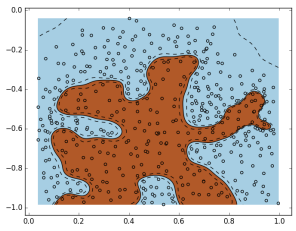
Pregunta [5]: Prueba a lanzar una SVM lineal con los valores para C que se utilizaron anteriormente. ¿Consigues algún resultado satisfactorio en el sentido de que no haya errores en el conjunto de entrenamiento?. ¿Por que?.

Para todos los casos obtenemos la siguiente gráfica, podemos ver que no tiene efecto el modelo lineal. Esto se debe a que los puntos no son separables linealmente, y por tanto se debe utilizar otra función de kernel.



Pregunta [6]: Propón una configuración de SVM no lineal (utilizando el kernel tipo RBF o Gaussiano) que resuelva el problema. El resultado debería ser similar al de la Figura 3. ¿Qué valores has considerado para C y para γ?. Además, incluye un ejemplo de una configuración de parámetros que produzca sobre-entrenamiento y otra que produzca infra-entrenamiento.

γ |C	10e-2	10e0	10e2
50			
100			

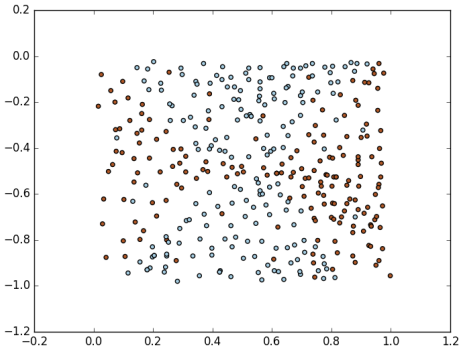
200			

El parámetro gamma indica 1/radio, por lo que si aumentamos este valor, los radios de las funciones RBF disminuyen como se muestra en la tabla. Además, modificando el parámetro C, es inversamente proporcional al margen. Si C es mayor, el margen es menor. Por lo que un término medio sería **gamma=100 y C=10e0**.

Por otro lado, podemos ver que el caso gamma=200 y C=10e-2 tiene infra-entrenamiento, es un mal clasificador. Y por otro lado, el caso con C=10e2 sobre-entrena ya que mete dentro un patrón outlier, que para generalizar no es conveniente.

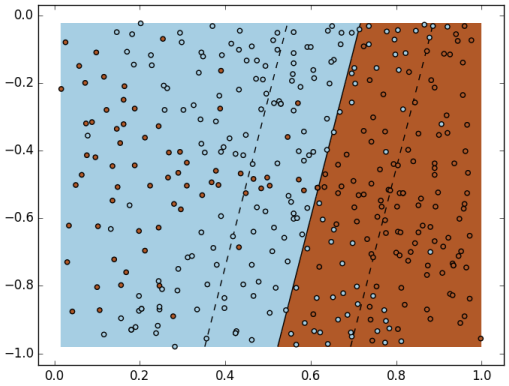
Tercet dataset de ejemplo

Pregunta [7]: En este caso, ¿es el dataset linealmente separable?. A primera vista, ¿detectas puntos que presumiblemente sean outliers?, ¿por qué?.



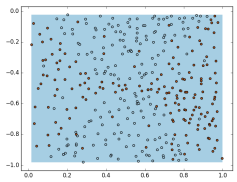
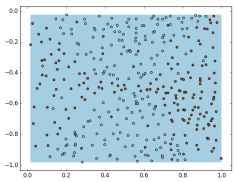
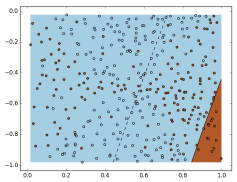
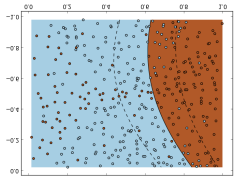
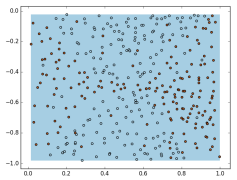
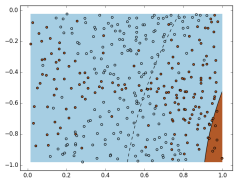
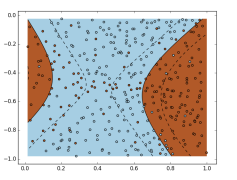
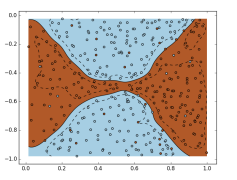
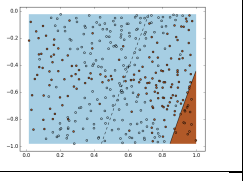
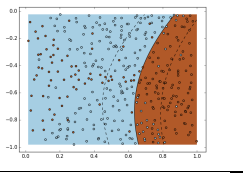
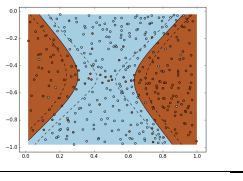
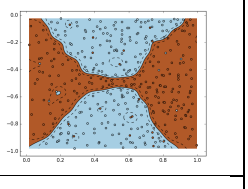
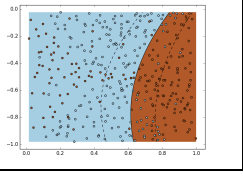
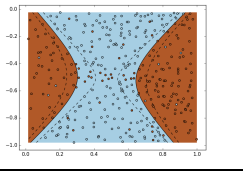
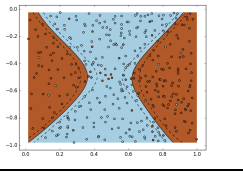
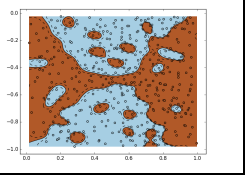
Aparentemente, si visualizamos los patrones como una nube de puntos, no parece ser linealmente separables. Además, existen puntos por encima y por debajo que podrían considerarse outliers ya que se trata de puntos esporádicos metidos en la zona de la clase opuesta.

Aun así, la figura siguiente muestra la clasificación lineal aplicada a este conjunto. Y se puede ver claramente que no clasifica correctamente, ni siquiera con una estructura soft que permita patrones en el margen.



Sin embargo, en la siguiente cuestión, se entrenará con una función de kernel RBF y si producirá mejores resultados para este conjunto.

Pregunta [8]: Lanza una SVM para clasificar los datos, con objeto de obtener un resultado lo mas parecido al de la Figura 5. Ajusta el ancho de kernel en el intervalo $\gamma \in$ {0.02, 0.2, 2, 200}. Ajusta el parámetro de coste en el intervalo C $\in$ {0.02, 0.2, 2, 200}. Establece el valor de los parámetros óptimos. Además, incluye un ejemplo de una configuración de parámetros que produzca sobre-entrenamiento y otra que produzca infra-entrenamiento.

γ |C	0.02	0.2	2	200
0.02				
0.2				
2				
200				

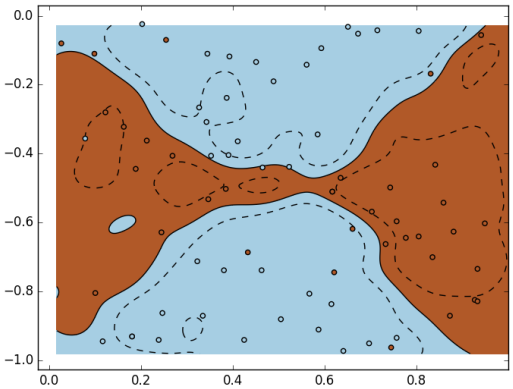
Como se muestra en las anteriores representaciones, al modificar los valores de los parámetros obtenemos diversos entrenamientos, mas o menos adecuados a nuestra base de datos. Por una parte, tenemos los valores gamma=0.02 y C=0.02 que esto indica un radio muy grande, y un margen muy grande, por lo que se obtiene un modelo con infra-entrenamiento. Por otra parte, el modelo de gamma=200 y C=200 ha entrenado demasiado bien, con un radio pequeño y sin apenas margen. Con esto obtenemos un sobre-entrenamiento que no es válido. Así que, a simple vista el modelo más óptimo sería **gamma=0.2 y C=200.**

Interfaz de consola

Pregunta [9]: Vamos a reproducir este proceso en Python. Divide el dataset sintético dataset3.libsvm en dos subconjuntos de forma estratificada, con un 80% de patrones en train y un 20% de patrones en test. Realiza el proceso de entrenamiento especificado anteriormente, utilizando los valores de C y γ obtenidos en el ultimo ejercicio. Comprueba el porcentaje de buena clasificación que se obtiene para el conjunto de test.

Para esto, utilizaremos la función StratifiedShuffleSplit(), que dividirá el conjunto de datos en los dos subconjuntos y posteriormente, utilizaremos el 80% de train para entrenar la máquina de vectores soporte, y el 20% para representarlos en generalización.

```
23  # Dividir patrones en 20% test y 80% train
24  sss = StratifiedShuffleSplit(y, test_size=0.2, train_size=0.8)
25
26  for train_index, test_index in sss:
27      X_train, X_test = X[train_index], X[test_index]
28      y_train, y_test = y[train_index], y[test_index]
```



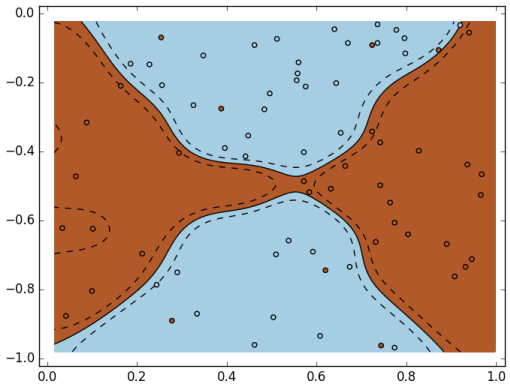
Analizando la figura, tenemos 74 patrones en test, de los cuales 7 están mal clasificados. Por tanto, tenemos un 90% de precisión en test en este caso. De todas formas, como la división train-test es aleatoria, en cada caso dependerá de la semilla utilizada y variará la precisión.

Pregunta [10]: Amplía el código anterior para realizar el entrenamiento del ejercicio anterior sin necesidad de especificar los valores de C y γ. Compara los valores óptimos obtenidos para ambos parámetros con los que obtuviste a mano. Extiende el rango de valores a buscar si es que lo consideras necesario.

Para que los valores de C y γ se estimen automáticamente, será necesario utilizar la función GridSearchCV(), y definir un intervalo para la búsqueda.

```
33  Cs = np.logspace(-5, 7, num=10, base=2)
34  Gs = np.logspace(-5, 7, num=10, base=2)
35
36  svm_model = svm.SVC()
37  #Estima los valores de C y gamma
38  svm_model = GridSearchCV(estimator=svm_model, param_grid=dict(C=Cs,gamm
39  svm_model.fit(X_train,y_train)
40
41  print svm_model.score(X_test, y_test)*100
42  print svm_model.best_params_
```

Después de una prueba experimental, obtenemos la siguiente gráfica con los patrones de test. Y obtenemos un 90% de precisión en la clasificación utilizando los valores estimados para C= 3.17 y gamma=50.79. Anteriormente, los valores fueron 0.2 y 200, que podrían asemejarse a los estimados en esta ocasión. De todas formas, haciendo hincapié en que la elección de los patrones es aleatoria.



Pregunta [11]: ¿Que inconvenientes observas en ajustar el valor de los parámetros "a mano", viendo el porcentaje de buena clasificación en el conjunto de test?

Que mediante la estimación a mano, es más probable que el modelo infra-entrene o sobre-entrene, ya que estamos ajustando los valores a "grosso modo". Por tanto, es mas correcto utilizar una función que recorra las distintas posibilidades y facilite la estimación de los parámetros.

Base de datos digits

Pregunta [12]: En primer lugar, necesitaras convertir los ficheros tipo arff (formato de Weka) al formato de libsvm.

Utilizaremos las siguientes funciones para tratar los ficheros de formato arff, y así poder trabajar con ellos como si fueran libsvm.

```
17  #Cargar dataset
18  dataset = arff.load(open('./BasesDatos/arff/train_digits.arff', 'rb'))
19  data = np.array(dataset['data'])
20
21  #Parsear los datos leidos
22  X_sparse_train = data[:,0:-1]
23  y_train = data[:,-1]
24
25  #Eliminar valores basura y convertir los char a float
26  X_train = np.char.encode(X_sparse_train , encoding="ascii", errors="ignore")
27  y_train = np.char.encode(y_train, encoding="ascii", errors="ignore")
28
29  X_train = np.float32(X_train)
30  y_train = np.float32(y_train)
31
```

Pregunta [13]: Una vez hayas convertido los ficheros, utiliza el script que desarrollaste en el ejercicio anterior para entrenar esta base de datos. Observe el valor de CCR obtenido para el conjunto de generalización y compáralo con el obtenido en prácticas anteriores. El proceso puede demorarse bastante. Al finalizar, toma nota de los valores óptimos obtenidos para los parámetros.

Después de la ejecución, hemos obtenido un valor de precisión en la clasificación de **94.67%** en test y 100% en train. Consideramos más importante en test, que es un buen clasificador con los parámetros estimados C=1.26 y gamma=0.03. A decir verdad, este caso demora demasiado tiempo, anotados 154 segundos.

Pregunta [14]: Localiza donde se especifica el valor de K para la validación cruzada interna y el rango de valores que se han utilizado para los parámetros C y γ. ¿Como podrías reducir el tiempo computacional necesario para realizar el experimento?. Prueba a establecer k = 3, k = 5 y k = 10 y compara, utilizando una tabla, los tiempos computacionales obtenidos y los resultados de CCR en test.

El valor K de la validación cruzada se corresponde con el parámetro cv de la función GridSearchCV(). Además, para reducir el tiempo computacional puede utilizarse el parámetro **n_jobs** que indica el número de trabajos que se ejecutan en paralelo, con -1 se indica que utilice el CPU al completo por tanto menor tiempo computacional tendrá ya que se optimiza el uso.

Estadísticos	K=3	K=5	K=10
C / gamma	3.17/0.03	1.25 /0.03	3.17/0.03
Tiempo Comp. (s)	75	157.7	333.6
CCR Test (%)	94.67	94.67	94.67

Podemos ver, que para cualquier cross-validation con distinta K se obtienen los mismo resultados de clasificación en test, y obviamente el 100% en train. Lo único es que vemos un incremento gradual en el tiempo computacional, y es preferible utilizar K=3.

Base de datos clasificación de spam

Pregunta [15]: Debes entrenar un modelo lineal de SVM con valores C = 10e−2, C = 10e−1, C = 10e0 y C = 10e1. Para ello utiliza un script similar al que usaste en la sección 2.6. Compara los resultados y establece la mejor configuración.

Para este caso, escogeremos el mismo método que utilizamos con el cross-validation 80%test-20%train. La única diferencia en el código se refiere a que en este caso tenemos los dos subconjuntos en dos ficheros distintos, por tanto no sería necesario y podemos cargar ambos ficheros sin utilizar la cross-validación. Aparte, también definimos la función de kernel como **'linear'**. Obtendremos los siguientes resultados para las pruebas con los valores de C.

Estadísticos	C=10e-2	C=10e-1	C=10e0	C=10e1
CCR Test (%)	**98.9**	97.8	97.5	97
Tiempo comp(s)	14.08	11.91	10.9	10.3

Podemos ver, que al aumentar el parámetro C, tarda un poco menos computacionalmente, y pero sobreentrena, perdiendo un poco de precisión. Por tanto, el valor recomendado sería **C=10e-2.**

Pregunta [16]: Para la mejor configuración, construye la matriz de confusión y establece cuales son los correos en los que la SVM se equivoca. Consulta las variables de entrada para los correos que no se clasifican correctamente. Ten en cuenta que para cada patrón, cuando xi es igual a 1 quiere decir que la palabra i-ésima del vocabulario aparece, al menos una vez, en el correo.

A partir de los parámetros anterior, sacamos la siguiente matriz de confusión.

Clase 1="No spam" **Clase 2="Spam"**		**Salidas Obtenidas**	
		Clase 1	Clase 2
Salidas esperadas	Clase 1	684	8
	Clase 2	3	305

Hemos obtenido una precisión de CCR Train=99.8% y CCR Test=98.9% en un tiempo computacional total de 16 segundos.

Posteriormente, vamos a mostrar los índices de los patrones mal clasificados que son

9 – Spam clasificado como no spam

emailaddr, for, group, httpaddr, inform, irish, linux, list, maintain, subscript, un, user.

21 – No spam clasificado como spam

an, and, at, be, by, call, can, contact, email, emailaddr, geek, heaven, httpaddr, if, immedi, is, list, mail, messag, need, net, not, number, of, offic, our, out, repres, respond, return, sf, spamassassin, sponsor, start, talk, the, there, thi, thinkgeek, to, until, welcom, when, will, you, your.

58 – No spam clasificado como spam

action, address, advertis, all, an, and, anti, appl, approv, ar, as, at, awai, back, bank, be, between, bill, bui, busi, by, call, can, card, center, channel, click, code, com, congress, content, continu, control, copyright, cost, custom, cut, daili, deliv, digit, don, down, driver, easi, edit, email, emailaddr, emerg, end, energi, enter, enterpris, error, european, event, everi, exactli, expert, faq, fax, fee, feed, file, find, firm, fnumber, follow, for, franc, free, from, get, give, go, good, grand, grant, group, ha, half, hardwar, headlin, here, high, host, hp, html, httpaddr, ibm, id, imag, in, instruct, intel, intern, internet, into, intro, is, isp, it, job, join, juli, just, keep, kill, known, land, languag, letter, life, like, line, linux, listen, mac, mail, make, maker, media, messag, microsoft, middl, more, ms, need, net, new, next, no, number, of, old, on, onli, out, over, password, pc, person, plan, platform, plug, polit, price, privaci, product, publish, pull, qnumber, realli, reg, regist, remot, reserv, right, road, same, season, secur, see, send, septemb, servic, ship, sign, site, situat, softwar, sort, spec, special, sponsor, staff, stai, standard, start, storag, stream, suffer, sun, sure, system, tablet, take, team, tel, term, that, the, these, thi, ticket, tm, to, todai, told, too, turn, uk, under, unit, unsubscrib, up, updat, url, us, visit, want, war, we, web, will, win, window, with, www, xp, yahoo, you, your.

73 – No spam clasificado como spam

abl, about, abov, accept, access, account, act, addit, address, administr, advertis, advis, affect, affili, ag, again, against, agenc, agent, agre, air, all, allow, also, altern, am, among, an, and, ani, anoth, ar, area, as, associ, assum, at, attempt, author, autom, avail, award, be, becom, befor, begin, behalf, below, between, book, box, brand, broadcast, bug, busi, but, button, by, ca, california, can, cannot, capabl, case, cash, caus, center, centuri, chanc, check, choos, claim, clearli, click, co, collect, com, commit, commun, compani, complet, comput, condit, confirm, connect, corpor, correspond, countri, court, dai, data, date, dear, decis, declar, defin, deliv, depend, describ, direct, directli, director, do, document, doe, dollarnumb, domain, doubl, draw, drive, dure, each, educ, either, electron, email, emailaddr, employe, enter, enterpris, entertain, entri, equal, equip, error, etc, event, except, expens, expir, fair, famili, feder, few, film, final, first, five, follow, for, foreign, form, free, from, full, fulli, game, gener, govern, grand, grant, greater, ground, ha, had, hardwar, hat, have, held, her, hi, him, hold, home, how, ident, if, immedi, improv, in, inc, includ, independ, indic, individu, inform, instal, institut, instruct, integr, internet, is, island, it, judg, kind, late, later, law, legal, like, limit, link, list, live, lo, local, locat, log, loss, lost, made, mai, mail, major, mani, manufactur, matter, meet, member, mention, messag, middl, militari, must, my, name, natur, near, necessari, night, no, non, not, number, numberd, numberth, oblig, occur, of, offer, offic, offici, on, onc, onli, onlin, open, or, order, organ, origin, other, otherwis, out, over, oz, pack, page, paragraph, parent, part, parti, particip, per, period, permiss, person, place, plan, pleas, point, polici, potenti, prevent, print, privaci, prize, problem, process, product, promot, properti, provid, public, purchas, purpos, random, reason, receiv, regard, relat, releas, repli, repres, request, requir, reserv, resid, respect, respons, result, retail, return, right, round, rule, run, same, screen, script, second, secur, select, self, send, servic, set, sign, signatur, sincer, site, size, social, softwar, specif, specifi, sponsor, standard, state, subject, submit, such, take, taken, tax, technic, telephon, ten, term, termin, texa, than, thank, that, the, their, them, these, thi, third, those, time, tm, to, togeth, traffic, transfer, travel, trip, under, unit, up, us, usa, valid, valu, ve, via, visit, voic, wai, we, web, week, where, which, who, whole, whose, will, win, winner, with, within, without, won, write, www, you, your.

147 – No spam clasificado como spam

about, accord, acquir, activ, actual, ad, addit, administr, advertis, advic, advis, after, again, ago, agre, almost, alon, also, am, america, american, among, an, and, ani, announc, annual, anoth, approach, ar, aren, around, articl, artist, as, ask, associ, assum, at, august, averag, awai, back, bank, base, be, becaus, been, befor, below, better, big, billion, bit, board, both, bottom, brand, bring, bui, busi, but, by, cabl, call, can, capit, career, case, cash, caus, charg, check, chief, chip, chri, citi, civil, clearli, client, close, com, commit, compani, complet, comput, conflict, connect, consid, construct, consult, consum, contact, contract, control, copyright, corpor, cost, could, court, creativ, credit, cross, current, dai, david, deal, debt, decad, declin, demand, describ, despit, detail, did, differ, director, discount, discov, discuss, do, doe, dollar, dollarnumb, don, doubl, down, dure, earli, earlier, earn, easili, econom, economi, edg, electron, els, emailaddr, emerg, end, entertain, environ, equip, establish, estim, even, ever, exampl, except, execut, exist, express, extens, face, fall, far, favorit, feder, few, file, financ, financi, find, firm, first, fix, flow, folk, for, form, former, found, friend, from, fund, gener, global, go, goe, good, govern, grant, great, group, grow, guarante, gui, ha, had, half, hand, have, he, head, health, help, hi, high, highli, hold, home, hope, hour, how, httpaddr, huge, human, if, in, includ, industri, inform, initi, instead, interest, internet, interview, into, invest, investor, involv, is, issu, it, jame, jim, job, john, june, just, keep, king, know, knowledg, known, larg, larger, last, late, later, latest, less, let, like, limit, line, littl, lo, loan, long, look, lose, lost, lower, made, mai, mail, make, maker, manag, mani, march, mark, market, master, materi, media, men, million, monei, month, more, most, mr, much, my, name, nation, nearli, necessari, need, network, new, newslett, no, nor, not, note, now, number, numberb, octob, of, off, offer, offic, often, old, on, onc, onlin, oper, opportun, or, organ, other, our, out, over, own, owner, pacif, paid, part, parti, partner, pattern, payment, peopl, percent, perfectli, perform, phone, pick, place, plai, plan, pleas, poor, pop, possibl, potenti, prefer, presid, pretti, price, privat, problem, profit, promis, prove, public, purchas, put, qualiti, quarter, question, quickli, rais, re, reach, realiz, reason, recent, remain, replac, report, repres, result, retail, return, reveal, right, risk, robert, role, sai, said, sale, san, save, search, second, see, seed, seek, seem, seen, self, sell, senior, sent, septemb, serv, servic, shape, share, should, show, sinc, sit, site, six, skeptic, skill, small, so, sold, some, sometim, special, speed, spend, spot, stai, start, step, still, stock, structur, style, success, suffer, suit, system, take, taken, talk, team, technolog, texa, than, that, the, thei, their, them, then, there, these, thi, think, third, those, though, thought, three, through, ticket, time, to, toll, took, top, touch, travel, tri, trip, troubl, trust, turn, two, under, unless, unlik, until, up, upgrad, us, usual, ve, veri, visit, vote, wa, want, war, washington, watch, we, web, week, well, were, west, what, when, where, which, who, whose, why, will, william, with, won, work, worth, would, write, ye, year, york, you, your.

328 – No spam clasificado como spam

address, advertis, all, alwai, and, announc, been, below, bui, close, cnet, com, comparison, custom, dai, download, each, emailaddr, faq, for, format, go, have, help, in, item, juli, let, mail, manag, million, more, my, name, nbsp, new, number, of, on, our, own, price, product, provid, purchas, put, review, sale, save, servic, shop, so, subscript, tech, technolog, that, the, there, thousand, to, todai, unsubscrib, up, valu, ve, visit, you, your.

407 – Spam clasificado como no spam

about, abov, account, address, africa, after, also, although, am, amount, and, ani, ar, as, assist, avoid, awai, be, befor, box, busi, by, call, can, caus, citi, com, command, compani, confidenti, contact, content, countri, current, date, dear, death, democrat, deposit, did, direct, doe, due, dure, effect, emailaddr, expect, expens, famili, father, fax, five, for, former, friend, from, fund, got, govern, have, he, head, help, hi, httpaddr, idea, in, invest, is, it, john, keep, kill, kingdom, know, knowledg, late, life, line, list, long, mail, map, me, militari, million, monei, mr, my, need, not, note, number, of, on, open, or, osdn, out, over, person, plan, pleas, present, presid, privat, profit, prompt, reach, remain, remot, republ, requir, respect, respons, sai, secur, seek, self, should, son, spamassassin, suggest, take, talk, tel, term, that, the, there, these, thi, threat, through, to, train, transact, unit, urgent, us, war, we, while, who, wife, will, with, www, you, your.

526 - No spam clasificado como spam

about, ad, advertis, after, am, an, and, as, at, been, below, bodi, but, by, click, contain, dai, direct, doc, email, emailaddr, except, extra, fals, final, geek, get, ha, have, heaven, here, httpaddr, is, it, list, look, mail, mark, net, next, no, number, off, on, other, posit, razor, see, sender, sf, signatur, so, spam, sponsor, such, talk, the, thi, thinkgeek, to, turn, us, user, welcom, where, which, with, word.

560 – No spam clasificado como spam

as, cours, ham, happi, have, it, late, never, non, number, on, pick, receiv, spam, to, too, up.

842 – No Spam clasificado como spam

about, account, acquir, across, ad, address, advertis, affect, all, allow, almost, american, an, and, ani, anoth, anti, approach, ar, area, as, ask, asset, at, attack, avail, avoid, be, becaus, been, believ, best, better, bill, but, by, california, call, campaign, can, cent, ceo, chanc, channel, chief, choic, choos, come, commerci, commun, compani, congress, constitut, contact, content, could, countri, coverag, dai, decid, defin, delet, differ, direct, directli, do, each, educ, effort, ensur, entir, equal, even, everi, expens, experi, face, fact, fair, featur, feder, few, field, firm, first, for, former, free, from, front, fund, futur, good, great, group, ha, had, have, he, hi, high, highli, him, howev, httpaddr, if, in, individu, inform, instead, internet, is, it, just, know, lack, larg, law, least, less, let, level, light, like, link, list, lo, mai, mail, make, manag, mani, market, media, messag, method, might, mike, million, monei, month, more, most, must, nation, neg, no, not, now, number, of, off, offer, offic, on, onli, onlin, open, opt, or, other, our, out, paid, particularli, peopl, per, percent, perfect, perform, person, phone, plai, point, polit, potenti, practic, prepar, presid, press, problem, provid, public, qualiti, radio, reach, read, receiv, recipi, refin, relev, remain, repli, repres, respons, seek, send, sender, sens, sent, she, shop, should, simpl, social, societi, softwar, some, spam, spamassassin, specif, spot, staff, standard, state, such, support, take, taken, target, technolog, than, that, the, thei, their, them, thi, those, through, time, to, todai, toward, tradit, tri, tv, unsolicit, unsubscrib, up, us, veri, vote, wa, wai, walk, want, war, we, what, when, while, who, whose, will, with, without, won, world, would, written, year.

881 - Spam clasificado como no spam

about, abus, all, also, america, american, amount, an, and, ani, anoth, ar, argument, as, ask, attempt, attent, author, awai, babi, base, basi, be, been, befor, behavior, believ, best, better, black, both, bui, by, car, carri, caus, center, children, choic, choos, citizen, close, collect, com, commentari, commit, commun, confus, contain, cool, copi, could, coupl, cours, current, daili, date, did, direct, do, drive, drug, due, dynam, educ, email, emailaddr, even, ever, exist, expens, explain, ey, fact, femal, for, form, from, further, gener, go, govern, ha, have, heard, hi, higher, home, hous, how, howev, identifi, if, imag, import, in, individu, invest,

is, it, know, known, lack, larg, learn, less, light, like, live, look, mai, make, maker, male, mani, mayb, me, men, might, miss, more, mother, move, my, name, natur, neg, new, no, not, noth, notic, now, number, of, often, on, onli, or, over, own, pai, paid, part, particular, pass, peopl, perhap, place, pleas, poor, popul, prefer, project, purchas, rais, rather, refer, repres, request, reserv, respect, result, said, save, self, sell, shop, side, singl, sit, societi, soon, sort, stori, street, suffer, support, tax, than, that, the, thei, their, them, themselv, there, these, thi, through, to, todai, told, true, truth, util, visit, wa, warn, we, wed, welcom, well, what, when, while, who, with, women, wonder, world, would, www, ye, you, young, your.

Pregunta [17]: Compara los resultados obtenidos con los resultados utilizando una red RBF. Para ello, haz uso del programa desarrollado en la practica anterior. Utiliza solo una semilla (la que mejor resultado obtenga).

Modificamos el script de la práctica3 para que pueda leer los ficheros libsvm, y a partir de ahí trabajaremos de la misma manera, poniendo 100 rbf, y valor eta=0.1. Para un problema de clasificación con rbf del correo de spam, obtenemos la siguiente matriz de confusión:

Clase 1=”No spam” **Clase 2=“Spam”**		**Salidas Obtenidas**	
		Clase 1	Clase 2
Salidas esperadas	Clase 1	685	7
	Clase 2	25	285

Hemos obtenido una precisión de CCR Train=96.33% y CCR Test=97%. Aparte de obtener una precisión similar, vemos que hay más errores en correos de spam que los clasifica como no spam. Esto es positivo, ya que la mayoría de correos “no spam” los clasifica bien, porque de lo contrario estamos perdiendo correos válidos. Posteriormente, vamos a mostrar los índices de los patrones mal clasificados que son:
[9, 21, 36, 90, 117, 208, 264, 306, 309, 328, 386, 412, 413, 422, 541, 586, 620, 638, 652, 676, 730, 771, 791, 811, 832, 869, 870, 881, 939, 958]

4. REFERENCIAS BIBLIOGRÁFICAS

- [1] Algoritmo SVM. Documentación Scikit-learn:
  - http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

- [2] Función StratifiedShuffleSplit. Modulo cross-validation. Documentanción Scikit-Learn:
  - http://scikit-learn.org/stable/modules/generated/sklearn.cross_validation.StratifiedShuffleSplit.html

- [3] Función GridSearchCV. Documentación Scikit-learn:
  - http://scikit-learn.org/stable/modules/generated/sklearn.grid_search.GridSearchCV.html

- [4] Conversión arff files. Documentación Scikit-learn. Stackoverflow:
  - http://stackoverflow.com/questions/27264426/arff-files-with-scikit-learn

- [5] Presentación práctica 4 SVM. Asignatura Introducción al Modelos Computacionales. Moodle 2016/17:
  - http://moodle.uco.es/m1617/mod/resource/view.php?id=66935

- [6] Tema 3. Introducción a los Vectores Soporte. Asignatura Introducción al Modelos Computacionales. Moodle 2016/17:
  - http://moodle.uco.es/m1617/mod/resource/view.php?id=97095