Inés Broto Clemente
Arnau Turch Ferreres

CLASSIFICADOR DE SUPORT VECTORIAL AMB AMPL

1. Introducció

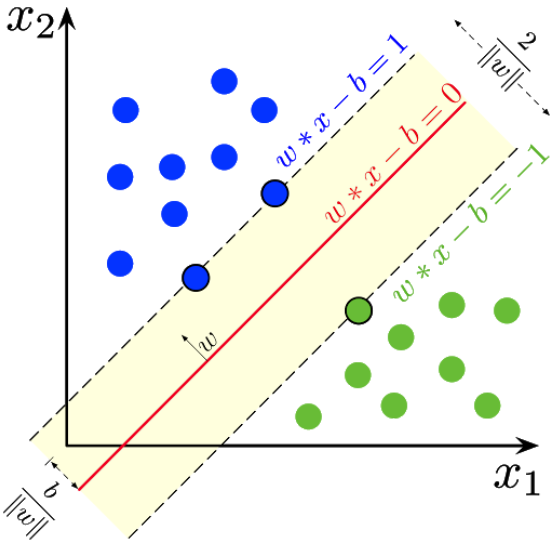
L'objectiu d'aquesta pràctica és implementar un classificador utilitzant tècniques de suport vectorial utilitzant el llenguatge de programació AMPL. Primer de tot es mostrarà el codi utilitzat per optimitzar i solucionar aquest problema, tant el primal com el dual, i seguidament utilitzar diversos conjunts de dades per posar a prova aquest algoritme i comprovar la seva utilitat i correcció a l'hora de classificar.

Utilitzarem tres datasets per tal de poder realitzar aquest estudi, en primer lloc utilitzarem unes dades linealment separables generades automàticament, en segon lloc dades no separables també generades per nosaltres automàticament y finalment posarem a prova aquest classificador amb un problema real.

L'objectiu d'aquest treball és aprendre com és comporta aquest classificador de suport vectorial utilitzant tant la formulació dual com la formulació primal, observar les seves diferències, els avantatges i inconvenients de cadascuna i determinar quin té una millor resposta depenent de com siguin les dades d'entrada.

2. Implementació de les màquines de suport vectorial:

Les màquines de suport vectorial son una eina per resoldre el problema de la classificació d'unes certes dades en dues classes. L'objectiu, per tant, és trobar els dos hiperplans paral·lels que disminueixen l'error de classificació i es troben el més separats possible l'un de l'altre. Per fer això buscarem un hiperplà $w^T x + \gamma = 0$ a partir del qual definirem els dos hiperplans paral·lels $w^T x + \gamma = \delta$ i $w^T x + \gamma = -\delta$ que voldrem separar el màxim possible sense perdre encert de classificació.
D'aquesta manera, si un punt x_i pertany a la primera classe, llavors $w^T x_i + \gamma \geq \delta$ i per contra, si x_i pertany a la segona classe, llavors $w^T x_i + \gamma \leq -\delta$.



Aquest problema sovint és infactible si són tant estrictes pel que fa a la classificació i és per això que afegim unes noves variables s_i que ens permeten cometre errors de classificació. Hem implementat dues màquines de suport vectorial seguint les dues formulacions que hem vist a les classes de teoria:

- **Formulació primal:**

Consisteix en la resolució directa del problema de de les màquines de vectors de suport:

$$\begin{aligned} min\ & \tfrac{1}{2}w^T w + \upsilon \sum_{i=1}^{m} s_i \\ s.a\quad & y_i(w^T x_i + \gamma) + s_i \geq 1 \quad i = 1, ..., m \\ & s_i \geq 0 \quad i = 1, ..., m \end{aligned}$$

on y_i correspon a la classe del punt x_i (1 o -1 per tal d'escriure totes les restriccions dels plans en una sola desigualtat), s_i representa el possible error de classificació del punt x_i (s_i = 0 si el punt queda ben classificat) i ν és el valor de penalització per la classificació errònia, és a dir, quan error estem disposats a permetre a canvi d'obtenir hiperplans més separats.

- **Formulació dual:**

Sabem que a tot problema primal li correspon un de dual que treballa amb els multiplicadors de Lagrange com a variables i sovint ens és útil a l'hora de treballar amb dades que no són linealment separables però l'aplicació d'un Kernel determinat ens en facilita la separació, tal i com veurem al treballar amb el tercer conjunt de dades.

$$\begin{aligned} max\ & \sum_{i=1}^{m} \lambda_i - \tfrac{1}{2}\lambda^T Y K Y \lambda \\ s.a\quad & \sum_{i=1}^{m} \lambda_i y_i = 0 \\ & 0 \leq \lambda_i \leq \nu \quad i = 1, ..., m \end{aligned}$$

On K representa la matriu de Kernel. Aquesta matriu es calcula a partir de les dades i permet transportar-les a un altre espai on possiblement son més fàcils de separar. En general podem definir un kernel com a $K(x,y) = \phi(x)^T \phi(y)$ on la funció $\phi(\cdot)$ transporta les dades a l'espai que més ens interessi. En el cas més senzill, si prenem $\phi(\cdot) = Id$, estarem calculant el producte escalar de les dades per resoldre el problema dual.

Per realitzar aquestes implementacions hem utilitzat el software d'AMPL a través de R (aquest ens ha permès calcular taules de contingència i errors de training i test d'una manera senzilla). Les implementacions dels dos models, així com el script de R s'han lliurat conjuntament amb aquest document.

- **Càlcul de l'error de validació i de test**

Al llarg de la pràctica hem treballat amb diferents bases de dades i pels diferents casos hem avaluat l'error de validació i l'error de test. El primer és una indicació de quan ben classificades han quedat les dades amb les que hem entrenat la nostra SVM tenint en compte que hem permès certs errors de classificació (variable s_i). El segon consisteix en fer funcionar la SVM amb els valors de les variables que hem obtingut de l'optimització amb unes dades diferents, cosa que prova la veritable utilitat de la nostra implementació ja que la idea és que amb una sola optimització aconseguim un classificador prou bo.

Hem calculat aquestes dues mesures a partir de la solució generada per les dues implementacions i això, que en el cas primal és gairebé trivial, no ho és tant si considerem el problema dual. Cal destacar que per evitar trobar molts punts a la regió entre els hiperplans (no-classificats) a l'hora de determinar si un punt pertanyia a una classe o a l'altra hem considerat un únic hiperplà $w^T x + \gamma = 0$ enlloc dels dos hiperplans paral·lels a distància 2δ.
Al obtenir la solució del problema primal ja tenim tota la informació necessària per caracteritzar l'hiperplà separador per tant, només cal veure si a l'avaluar l'expressió $w^T x + \gamma$ per un conjunt de punts x determinats obtenim un valor (per cada x_i determinat) més gran o més petit que zero i classificar cada punt a la classe en funció d'això. Per contra, quan obtenim la solució del problema dual només tenim les λ generades a partir d'optimitzar amb les dades i la funció $\phi(\cdot)$ o amb la matriu K.
Tal i com hem vist a les classes de teoria, podem obtenir els valors de w i γ a partir de les lambdes en el primer cas de manera més o menys senzilla , cosa que utilitzarem per comparar les solucions obtingudes a partir de les dues implementacions. Sabem que $w = (\lambda^T Y A)^T = \sum_{i=1}^{m} \lambda_i y_i \phi(x_i)$ i que $\gamma = \frac{1}{y_i} - \phi(x_i)^T w$ per algun punt que compleixi que $0 \leq \lambda_i \leq \nu$ (que sigui vector de suport). De totes maneres, no sempre coneixerem la funció $\phi(\cdot)$ per poder aplicar-la a les dades crues (x) i trobar el valors de les variables que defineixen el plà. En aquests casos, tal i com hem vist a les lliçons teòriques, l'ús del Kernel té molts beneficis ja que permet obtenir els valors de $\phi(x_i)^T w$ i γ sense necessitat de conèixer la funció $\phi(x_i)$. Podem deduir que $\phi(x_i)^T w = \sum_{i=1}^{m} \lambda_i y_i K(x_i, x)$ i que $\gamma = \frac{1}{y_j} - \sum_{i=1}^{m} \lambda_i y_i K(x_i, x_j)$ per algun punt que compleixi $0 \leq \lambda_j \leq \nu$.
Amb aquestes expressions hem pogut recuperar els valors de les w o de $\phi(x_i)^T w$ i les γ a partir de la solució dual i hem pogut mesurar els errors de validació i de test.

3. Classificador SVM amb dades linealment separables

Un cop ja hem implementat el SVM amb AMPL és hora de posar-lo a prova. En primer lloc utilitzarem dades obtingudes directament del generador “gensvmdat” proveït pel tutor de l’assignatura. Aquest generador ens proporciona un seguit de punts a l’espai juntament amb la classe a la qual pertanyen, cal destacar que alguns estan classificats incorrectament.

3.1. Problema Primal

Els punts utilitzats per entrenar aquest classificador es troben a l'arxiu de text “input_P.dat”, utilitzant la API de AMPL en R ens permet executar l'optimització des del mateix programa i obtenir fàcilment els paràmetres W* i δ* que definirán l’hiperplà de separació dels punts.

Primer de tot executem l’optimització amb 300 punts d’entrenament generats amb la seed 1234, el resultat que ens troba AMPL utilitzant el solver “MINOS” i valor de v = 0.25 és el que surt a continuació:

```
MINOS 5.51: optimal solution found.
212 iterations, objective 70.03023151
Nonlin evals: obj = 227, grad = 226.
```

Els errors de classificació en funció de la v són els següents:

v	0	0.1	0.25	0.5	1	2	5
Error de predicció de training (%)	45.6	10	9.3	8.3	8.6	8	8

Per finalitzar i comprovar el comportament d’aquest classificador amb dades externes al seu entrenament avaluarem l’error amb les dades de test, 200 punts generats utilitzant la seed 8529. Els resultats són aquests:

v	0	0.1	0.25	0.5	1	2	5
Error de predicció de test (%)	50	8	8.5	7.5	8	8	8

3.2. Problema Dual

De la mateixa manera que abans, utilitzarem els 300 punts de training generats prèviament, li passarem al model d'AMPL el valor de les variables nu, y, i la K.

K es el Kernel, en aquesta part del treball $K = AA^t$. Aquesta definició del Kernel fa que solucionar el problema dual sigui equivalent a solucionar el problema primal. En el cas de que les dades no fossin linealment separables, com es veurà més endavant, haurem d'utilitzar una transformació de les dades per enviar-les a un espai on si siguin linealment separables, aquesta transformació s'anomena Φ(x).

Aquesta implementació ens retornarà els valors òptims de lambda, λ*.

Executem l'optimització i el resultat que obtenim utilitzant valor de v = 0.25 és el següent:

```
MINOS 5.51: optimal solution found.
192 iterations, objective 70.03023151
Nonlin evals: obj = 213, grad = 212.
```

Un cop obtenim el valor òptim d'aquesta variable haurem de recuperar els valors que defineixen l'hiperplà de separació: W* i δ*.

Els errors de classificació en funció de la v són els següents:

v	0	0.1	0.25	0.5	1	2	5
Error de predicció de training (%)	45.6	10	9.3	8.3	8.6	8	8

Per finalitzar i comprovar el comportament d'aquest classificador amb dades externes al seu entrenament avaluarem l'error amb les dades de test, 200 punts generats utilitzant la seed 8529. Els resultats són aquests:

v	0	0.1	0.25	0.5	1	2	5
Error de predicció de test (%)	50	8	8.5	7.5	8	8	8

Podem observar que obtenim els mateixos errors de predicció en els dos tipus de problema, aquest fet será explicat més endavant, concretament en el punt 6 d'aquest informe.

Inés Broto Clemente
Arnau Turch Ferreres

4. Classificació de vins amb el classificador SVM

A continuació ens disposem a provar el nostre classificador amb una base de dades que hem obtingut d'internet. Consisteix en un conjunt d'observacions d'anàlisis químics sobre vins que pretenen classificar-los en dos grups diferenciats. Hem obtingut les dades del repositori lliure d'aprenentatge automàtic UCI i corresponen a 130 observacions de 13 variables contínues de caire químic com per exemple la concentració d'alcohol o d'àcid màlic (per més detalls consultar el repositori).

D'una banda buscarem el nostre classificador òptim a través de la modelització primal. Les dades que hem utilitzat es poden trobar a l'arxiu *wine_1.dat* lliurat de la mà d'aquest document. La solució utilitzant el solver MINOS i el valor de υ = 0,5 és la següent:

```
MINOS 5.51: optimal solution found.
196 iterations, objective 2.03101513
Nonlin evals: obj = 213, grad = 212.
Adding  meminc=0.196  to $minos_options might save time.
```

Si repetim l'experiment diferents vegades amb diferents valors de lambda i calculem l'error de validació obtenim els següents resultats:

ν	0	0.25	0.5	0.75	1
Error de predicció (%)	45.39	0,769	0,769	0	0

Per a resoldre el problema dual utilitzarem el mateix procediment de l'apartat anterior, és a dir, calcular la matriu Kernel i juntament amb altres paràmetres optimitzar en funció de λ.

```
MINOS 5.51: optimal solution found.
51 iterations, objective 2.031014508
Nonlin evals: obj = 117, grad = 116.
```

Al solucionar aquest problema hem obtingut els valors òptims de λ però tal i com hem vist a classe, a partir d'aquests valors i la matriu Kernel podem obtenir els valors de $\phi(x_i)^T w + \gamma$ per cada dada i per tant calcular l'error de validació. A més a més, podem estalviar-nos aquest càlcul ja que tal i com hem vist a les lliçons de teoria, el valor de cada λ_i és un indicador de si la observació x_i ha estat correctament classificada o no.

Tal i com podem veure, els errors de predicció al validar el model són els mateixos que havíem observat amb el primal ja que les dades són linealment separables i per tant la solució dels dos problemes és la mateixa.

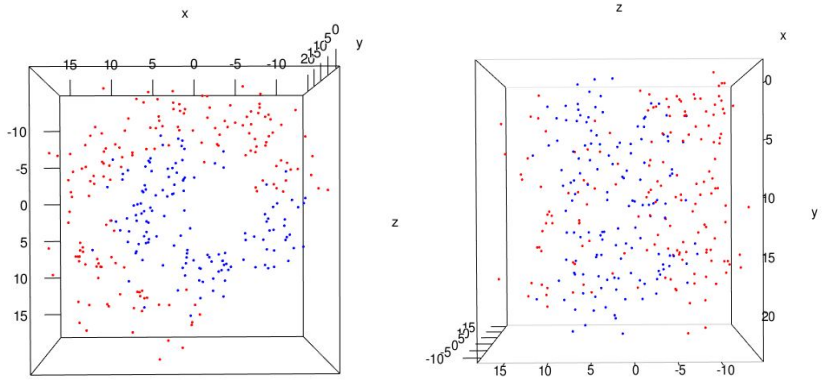
ν	0	0.25	0.5	0.75	1
Error de predicció (%)	45.39	0,769	0,769	0	0

En els dos casos podem observar com amb l'augment del valor del paràmetre ν fa decréixer el percentatge d'error i per tant és convenient definir-lo al voltant d'1.

5. Classificador SVM amb dades linealment no separables.

A continuació realitzarem exactament que en el punt 3 amb la diferència que ara els punts utilitzats per construir el classificador no son separables linealment.

Per tal d'aconseguir aquestes dades hem utilitzat la funció *make_swiss_roll()* i realitzant un plot en 3 dimensions d'aquestes dades observem la seva distribució a l'espai i com clarament no son separables d'una manera lineal.



Aquest problema si l'intentem resoldre com ho hem fet en l'anterior punt, els resultats que obtindrem seràn bastant dolents degut a aquesta distribució dels punts a l'espai, és aquí on entra en joc la funció Φ(x).

v	0	0.1	0.25	0.5	1	2	5
Error de predicció de training (%)	53.6	36	36	36	36	36	36

La funció que utilitzarem per tal de poder transformar l'espai dels punts a un on aquests siguin separables de manera lineal serà el RBF o Kernel Gaussià.

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$$

Si implementem el nostre SVM utilitzant aquesta expressió els resultats són molt millors:

v	0	0.1	0.25	0.5	1	2	5	10
Error de predicció de training (%)	53.6	10.33	8	8.66	8	7.66	6	4.66

A continuació calcularem l'error de test per comprovar com actua aquest SVM utilitzant dades desconegudes utilitzant el Kernel Gaussià:

v	0	0.1	0.25	0.5	1	2	5	10
Error de predicció de test (%)	46.5	22	17.5	15.5	10	39.5	36	32.5

6. Problema Primal vs. Problema Dual

Tal i com ja hem mencionat, a tot problema primal li correspon un problema dual i en el cas que es compleixi el teorema de la dualitat forta, solucionar un dels dos problemes és equivalent a solucionar-ne l'altre.

L'objectiu d'aquest apartat de la pràctica serà comprovar que això és cert recuperant els valors de w i γ a partir de les λ obtingudes al solucionar el problema dual amb els dos primers conjunts de dades per veure si els hiperplans de separació trobats per ambdues modelitzacions coincideixen. Per fer això només ens cal aplicar els resultats hem hem vist a les classes de teoria.

Sabem que $w = (\lambda^T Y A)^T = \sum_{i=1}^{m} \lambda_i y_i \phi(x_i)$. D'aquesta expressió ja podem extreure el valor de w però, a més a més, si tenim en compte que tots els punts correctament classificats però no vinculants donaran lloc a una $\lambda_i = 0,$ el sumatori es pot reduir a sumar només les components de la resta de punts.
Per recuperar la γ podem utilitzar el fet que per tot punt correctament classificat que sigui vector de suport (es trobi sobre el pla de separació) es compleix que $\phi(x_i)^T w + \gamma = 1$. D'aquí podem deduir que $\gamma = \frac{1}{y_i} - \phi(x_i)^T w$ i per tant només cal trobar un punt que sigui vector de suport ($0 \leq \lambda_i \leq \nu$) i aplicar la fòrmula.

Degut a que els dos primers problemes que hem dissenyat treballaven amb dades linealment separables, al calcular els hiperplans a partir de la solució dual hem obtingut els mateixos hiperplans que havíem obtingut al solucionar el problema primal.

- **Dades linealment separables**

```
W dual:  2.55792 2.151925 3.031986 2.292407
W primal:  2.55792 2.151925 3.031986 2.292407

Gamma dual:  -5.070656
Gamma primal:  -5.070656
```

- **Dades d'internet: classificació de vins**

```
W dual:  0.589846 0.5903734 0.8113758 -0.1879901 -0.008920042 -0.02069765 0.2108429 0.1347984
-0.08760371 0.3809319 -0.1152113 0.4161904 0.006196424

W primal:  0.589846 0.5903736 0.8113758 -0.1879902 -0.008920056 -0.02069767 0.2108429 0.1347978
-0.08760368 0.3809318 -0.1152114 0.4161904 0.006196426

Gamma dual:  -13.96099
Gamma primal:  -13.96099
```

7. Conclusió

Utilitzar un SVM per classificar un conjunt en dos classes pot aportar resultats bastant satisfactoris com s'ha vist durant tot aquest treball; per dades linealment separables és equivalent resoldre el problema primal i el problema dual però quan aquestes no són separables linealment la classificació és complica una mica.

Per dades linealment no separables el problema primal no es un bon recurs a utilitzar. En canvi, resoldre el problema mitjançant la formulació dual, i més en concret, utilitzant el Kernel adient, ens aporta resultats molt més satisfactoris. El preu que paguem per tal de millorar la classificació és una complicació en el càlcul del pla de separació i en la interpretació d'aquest.

A més més, hem pogut veure com afecta el valor de ν a l'optimització i en els nostres experiments hem pogut veure que els valors propers a 1 són els que es donen millors resultats. De totes maneres, seria interessant buscar el valor de ν òptim quan es vulgui implementar una SVM amb procediments de validació.