

Mathematical Optimization
JUNE 2020

Support vector classifier in AMPL

PROJECT REPORT

Marc Fuentes i Oncins
Victor Novelle Moriano
Barcelona, UPC - FIB & FME

Index

1.Introducció	1
2. Programes utilitzats per a la resolució del problema	2
2.1 AMPL	2
2.2 Python	3
2.2.1 Funcionament bàsic	3
2.2.2 Requeriments per al seu funcionament	4
3. Anàlisis dels resultats obtinguts	5
3.1 Anàlisis dels problemes tractats	5
3.1.1 SVM Problem (Formulació Primal)	5
3.1.2 SVM Problem (Formulació Dual)	5
3.1.3 RBF Problem	6
3.2 Anàlisis de les dades obtingudes	7
3.2.1 GENSVMDAT	7
3.2.2 Swiss Roll	8
3.2.3 Diabetes Data Set	10
4. Conclusions	12

1.Introducció

L'objectiu d'aquest projecte consisteix en la creació d'una SVM mitjançant AMPL en les seves dues formulacions (Primal i Dual) per a la posterior classificació de punts, la generació dels quals es realitza de manera diferent per tal de comprovar el rendiment de la SVM creada en funció de la generació.

El problema d'optimització primal de classificació mitjançant una *Support Vector Machine* es pot formular com:

$$\begin{aligned} \min_{(\omega, \gamma, s)} \quad & \frac{1}{2} \omega^t \omega + \nu \sum_{i=1}^m s_i \\ \text{s.a} \quad & y_i (\omega^t x_i + \gamma) + s_i \geq 1 \\ & s_i \geq 0 \end{aligned}$$

On w representa la normal de la separació del hiperpla, γ representa la localització respecte a l'origen i s són les variables *slacks*, que ens permeten un cert grau d'error en la classificació de les observacions. En la nostra pràctica, per al cas de la formulació primal i dual disposarem de $\Phi = I$, mentre que per a RBF no coneixem quina és aquesta funció aplicada a l'input space.

Mentre que la formulació dual del problema és:

$$\begin{aligned} \max_{\lambda} \quad & \lambda^t e - \frac{1}{2} \lambda^t Y K Y \lambda \\ \text{s.a} \quad & \lambda^t y = 0 \\ & 0 \leq \lambda \leq \nu e \end{aligned}$$

En aquest cas, la funció dual depèn de λ , la variable dual. Indicar que K representa el kernel, que consisteix en $\Phi(x)^T \cdot \Phi(x)$. Per tant, en un problema dual podem aconseguir la classificació de punts tot i desconèixer el *feature space* (*input space* al qual se li aplica una *mapping function*).

Abans de procedir a l'anàlisi de la SVM així com una anàlisi en profunditat dels resultats obtinguts, considerem necessari proporcionar una breu explicació en relació al programa *.py* que s'adjunta com a part de la pràctica per assegurar un correcte funcionament d'aquest.

2. Programes utilitzats per a la resolució del problema

2.1 AMPL

Per tal de resoldre els problemes formulats prèviament, utilitzem el software *AMPL*. Per a poder realitzar ús d'aquesta eina, és necessari modelar el programa de manera que aquest sigui comprensible per a la màquina per tal que, posteriorment, resolgui el problema amb les dades introduïdes. A continuació presentem els dos models que han estat utilitzats per a la resolució dels problemes, un per a la formulació primal i l'altre pel dual:

Primal

```
# Parameters
param n >= 1, integer;
param m >= 1, integer;
param nu >= 0;

param y {i in 1..m};
param A {1..m,1..n};

#Variables
var gamma;
var w {i in 1..n};
var s {i in 1..m} >= 0;

# SVM primal.

minimize SVM Primal :
    1/2*sum{i in 1..n}(w[i]*w[i]) + nu*sum{j in 1..m}(s[j]);

subject to Restriccion{i in 1..m}:
    y[i]*(sum{j in 1..n}(A[i,j]*w[j]) + gamma) + s[i] >= 1;
```

Dual

```
# Parameters
param m >= 1 integer;
param nu >= 0;

param y {i in 1..m};
param K {1..m,1..m};

#Variables
var landa {i in 1..m} >= 0, <= nu;

#SVM Dual.

maximize SVM_Dual: sum{i in 1..m}(landa[i]) - 1/2*(sum{i in 1..m}(sum{z in 1..m}(landa[i]*y[i]*landa[z]*y[z]*K[i,z])));
subject to Restriccion: sum{i in 1..m}(landa[i]*y[i]) = 0;
```

La sintaxi d'AMPL és força simple. En primer lloc establim quins són els paràmetres del problema, és a dir, aquelles variables que disposen d'un valor fixat i que per tant no han de ser calculades. Posteriorment, indiquem quines són, a diferència dels paràmetres, les variables que han de calcular-se al llarg del problema, així com el seu tipus (en aquest cas

matrius, vectors i escalars) i les seves respectives restriccions. Per últim, es tradueix la formulació del problema presentada a l'apartat 1 a sintaxi comprensible per a *AMPL* indicant la funció a minimitzar/maximitzar i les restriccions que disposem.

Hem d'indicar que per tal que *AMPL* resolgui un problema, a més del model és necessari indicar-li quines són les dades i quin és el solver a utilitzar. Si més no, de tots aquests problemes s'encarrega el següent *software* utilitzat a la pràctica.

2.2 Python

El programa *Python* adjunt *Automatic.py*, s'encarrega, de manera automàtica, de la generació de les dades utilitzades en la classificació, la utilització dels models *AMPL* prèviament presentats i el posterior càlcul del percentatge d'encerts en la predicció.

2.2.1 Funcionament bàsic

El primer pas per procedir en la pràctica consisteix en l'execució del programa. Situant-nos en la carpeta on es troba emmagatzemat, obrim una terminal i executem la següent comanda:

En primer lloc, el programa permet l'elecció a l'usuari de quin problema vol resoldre, disposant com a 3 possibles accions:

- Primal SVM
- Dual SVM
- RBF SVM

Un cop escollit quin és el problema a solucionar, l'usuari indica quina metodologia vol emprar per a la generació aleatòria de punts, disposant de:

- Gensvmat
- Swiss Roll
- Diabetes data set

Font Data set Diabetes: LIBSVM Data: Classification, Regression, and Multi-label
<https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary/diabetes>

on la tercera opció consisteix en una versió netejada (perquè sigui comprensible per a *AMPL*) d'un data set d'internet tal com es demanava en els requeriments de la pràctica. En el cas que les dades a utilitzar no provenguin del *Diabetes data set*, l'usuari ha d'indicar la quantitat de punts a generar, la seed que vol utilitzar per a aquesta generació aleatòria així com el valor

que és vol proporcionar a l'hiper paràmetre *nu*, en funció del grau de penalització respecte als *slacks* que es vulgui realitzar.

Després que els paràmetres necessaris hagin estat introduïts per part de l'usuari, es procedeix a resoldre el problema d'optimització escollit mitjançant AMPL i es mostra per pantalla les variables d'interès, així com quin a estat el percentatge d'ajust aconseguit per a les dades d'entrenament a partir de la solució obtinguda.

Posteriorment, el programa torna a demanar a l'usuari la introducció de nous paràmetres, sent aquests el nombre de punts i la seed desitjades per a la creació dels punts de test. Indicar que la creació d'aquests punts es realitza amb la mateixa metodologia de creació de punts que ha estat escollida per a la creació de les dades d'entrenament. La demanda d'una nova seed té com a principal objectiu assegurar la repetició dels mateixos resultats en cas que els paràmetres introduïts siguin els mateixos.

2.2.2 Requeriments per al seu funcionament

Tot i ser un programa molt útil, ja que permet l'execució automàtica de qualsevol combinació de les opcions disponibles per a l'usuari d'una manera ràpida i còmoda (ja que pot ser executada directament des del terminal) són necessaris uns petits requeriments inicials per assegurar el seu correcte funcionament. Per tal de facilitar el seu *setup*, tots els elements que han de ser modificats per part de l'usuari es troben en la part superior del programa. Tot i que en el codi s'indica mitjançant comentaris com s'ha de realitzar aquesta modificació, adjuntem també una petita explicació.

1. El programa, com ja s'ha indicat, fa ús d' *AMPL* per a la resolució de problemes. Per tant, és necessari que l'usuari indiqui on es troba emmagatzemat en la seva màquina.
2. El programa fa ús de l'executable *gensvmdat* proporcionat pel professor com a possible mètode de generació de dades. Com que l'execució del programa es realitza de manera automàtica és necessari indicar on es troba emmagatzemat l'executable a l'ordinador.
3. El programa realitza ús de diverses llibreries per assegurar la connectivitat entre *AMPL* i *Python*, per a l'ús de funcions matemàtiques, etc. entre altres. Per tant és necessari instal·lar aquestes llibreries prèviament a l'ordinador, sent:

- **`pip install numpy`**
- **`pip install sklearn`**
- **`pip install statistics`**
- **`pip install amplpy`**

Un cop realitzats aquests passos, el programa estarà llest per a funcionar de manera correcta. Indicar que el solver utilitzat en totes les dades proporcionades de la pràctica ha estat *cplex*, tot i que aquest també pot ser modificat canviant la respectiva línia en la part superior del codi. Per últim, tot i que és possible disposar dels arxius distribuïts en diferents carpetes, es recomana disposar de tots els fitxers en la mateixa carpeta, per tal de facilitar la seva consulta.

3. Anàlisis dels resultats obtinguts

3.1 Anàlisis dels problemes tractats

En aquest apartat, realitzarem una anàlisi de cadascuna de les formulacions per a cadascun dels problemes tractats, quines són les seves característiques i com procedir a l'obtenció de paràmetres.

3.1.1 SVM Problem (Formulació Primal)

Com ja sabem, la formulació primal del problema de classificació per a SVM es tracta d'un problema convex. La funció objectiu és convexa (demostració realitzada a classe a l'exercici 2.10), així com la regió factible. Aquesta consisteix en una intersecció de semiplans, els quals analitzats individualment són convexos (la primera restricció és una funció afí $Ax \leq b$, mentre que la segona és clarament convexa), i per tant, la seva intersecció també ho és (demostrat a classe a l'exercici 2.2).

Per tant, a causa de tractar-se d'un problema convex, podem afirmar que si disposem d'un mínim local, tenim al seu torn, un mínim global per a la funció.

3.1.2 SVM Problem (Formulació Dual)

Tal com indica el seu nom, aquest problema consisteix en la versió dualitzada del *SVM formulació primal*. De manera prèvia a la seva resolució, ja sabem que es tracta d'una funció còncava, independentment de si el problema primal és o no convex, gràcies per al teorema de concavitat de $q(\lambda, \mu)$. A més, en el nostre cas, degut a que hem demostrat prèviament que el problema primal és convex, podem garantir que es compleix el *Strong Duality theorem*, que ens indica que el resultat obtingut per la funció dual serà igual al resultat obtingut per la funció primal.

Indicar que per tal d'obtenir la formulació del problema presentada a l'apartat 2.1, s'ha realitzat una simplificació de l'expressió obtinguda mitjançant la dualitat de *Wolfe*, aplicable ja que totes les funcions a més de ser convexes, són diferenciables.

Un cop establert que la solució del problema dual serà igual a la del primal, procedirem a recuperar els resultats per a les variables de la formulació primal a partir de la dual.

Tal com s'ha indicat prèviament, en el cas del problema primal la ϕ utilitzada en aquesta pràctica és la identitat. Com que sabem la funció aplicada a les dades d'entrada, podem ser capaços de calcular els valors dels pesos (w) del problema primal mitjançant la següent fórmula:

$$\omega = \sum_{i=1}^m \lambda_i y_i \phi(x_i)$$

La variable γ pot ser obtinguda, mitjançant:

$$\gamma = \frac{1}{y_{sv}} - \phi(x_i)^t \omega$$

on y_{sv} representa l' y d'un punt que sigui SV ($\lambda_i > 0$).

3.1.3 RBF Problem

El problema RBF consisteix en la resolució d'un problema de SVM Dual, on la matriu de Kernel ve definida per la següent fórmula:

$$K(x, y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}}$$

Si més no, a diferència del cas anterior, la funció ϕ aplicada a les observacions ens és completament desconeguda, aquest fet provoca que no siguem capaços d'obtenir les w del problema primal. Encara no disposar d'aquesta possibilitat, si que serem capaços d'avaluar nous punts, ja que si substituïm en la fórmula, no ens és necessari utilitzar A , sinó que podem treballar amb la matriu de kernel. Per tant, quan treballem aquest problema, únicament proporcionarem el valor de gamma, ja que és l'única variable primal que podem recuperar.

3.2 Anàlisi de les dades obtingudes

Un cop observades les propietats de cadascun dels problemes a resoldre procedim a analitzar els seus resultats per a execucions concretes. Ja que disposem de 3 mètodes de generació de dades i 3 problemes a resoldre, procedirem a analitzar 9 casos, per tal de comprovar totes les possibles combinacions. Així mateix, en cadascuna d'aquestes execucions canviarem el valor de v , per comprovar com es modifica el comportament del SVM.

Indicar també que tal com s'ha indicat en l'apartat 2.2.1, s'avaluarà també la capacitat predictiva del SVM per a dades de test, que no han estat prèviament utilitzades en el seu entrenament.

Per últim, ja que com hem indicat anteriorment el problema dual proporcionarà els mateixos resultats que el problema primal, únicament inclourem una taula per tractar ambdós casos.

3.2.1 GENSVMDAT

Aquest mètode de generació de dades en proporciona observacions 4-dimensionals, linealment separables. Aquestes són classificades en funció de la suma de les seves components. Si aquesta és superior a 2, el punt és classificat com a 1 i, si no, com a -1. Indicar que disposem d'alguns punts aleatòriament distribuïts, que es classifiquen de manera errònia.

Execució realitzada amb 100 punts d'entrenament amb seed 9834 i 50 punts de test amb seed 2323.

Formulació primal/dual

v	0	0,1	0,5	2	5
γ	0	-2,043	-3,393	-5,856	-7,308
w_1	0	0,865	1,682	2,469	3,639
w_2	0	1,086	2,085	3,584	4,285
w_3	0	0,837	1,806	2,736	3,385
w_4	0	0,804	1,555	3,041	3,619
Acc Train	53%	79%	90%	92%	93%
Acc Test	52%	76%	88%	88%	90%
f	0	7,356	25,090	70,54281205	143,837

Com podem comprovar, a mesura que augmentem el valor de v , es produeix al seu torn un augment del valor de γ així com dels pesos. Això és degut a , que en donar una major importància als *slacks*, la minimització enfront dels pesos adquireix un paper secundari, permetent que augmentin en valor absolut. Per altra banda, per a $v = 0$, no realitzem cap penalització per *slacks*, provocant que la funció a minimitzar sigui únicament la norma al quadrat dels pesos, sent mínima quan els pesos valen 0. Aquest fet provoca clarament una classificació incorrecta de les observacions, ja que aquestes únicament es classifiquen segons els *slacks*.

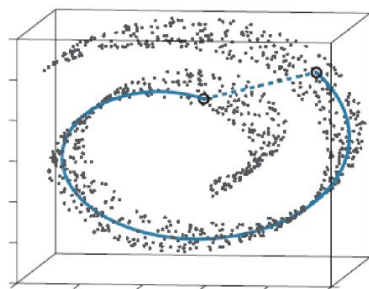
RBF

v	0	0,1	0,5	2	5
γ	1	-0,258	0,157	0,233	0,318
Acc Train	47%	82%	94%	97%	98%
Acc Test	48%	72%	86%	88%	88%
f	0,000	7,753	21,355	44,766	69,334

Tal com es pot comprovar, mitjançant l'ús d'un kernel RBF també aconseguim bons resultats de predicció per a la classificació de les dades. Igual que anteriorment, a mesura que augmentem el valor de v , també es produeix un augment de γ i de f , tot i que aquest no és tan gran com al primal/dual.

3.2.2 Swiss Roll

Aquest mètode de generació de dades proporciona observacions tridimensionals obtingudes que de manera conjunta generen una espiral (braç de gitano). El valor que ens proporciona la funció per a la variable y consisteix en la posició que disposarien els punts en l'eix sobre el qual desenrotllem l'espiral.



Amb l'objectiu d'aconseguir una classificació binària dels punts, en el nostre cas hem decidit realitzar la mitjana d'aquesta variable y , tots els punts que disposen d'un valor inferior són classificats com a -1 i la resta com a 1. L'elecció de la mitjana com límit de separació és totalment arbitrària i un altre criteri podria haver estat escollit (per exemple, la media).

També indicar que a la generació de punts s'ha afegit un lleuger soroll, per tal que no sigui totalment perfecta.

Execució realitzada amb 100 punts d'entrenament amb seed 27 i 50 punts de test amb seed 23.

Formulació primal/dual

ν	0	0.1	0.5	2	5
γ	0	-0,004	-0,002	-0,002	-0,002
w_1	0	0,110	0,110	0,110	0,110
w_2	0	-0.015	-0.015	-0.015	-0.015
w_3	0	-0,051	-0,051	-0,051	-0,051
Acc Train	48%	61%	61%	61%	61%
Acc Test	50%	48%	48%	48%	90%
f	0	7,223	36,114	144,432	361,068

Com podem comprovar, els valors dels paràmetres obtinguts per a $\nu > 0$ són iguals (canvien en el 5è o 6è decimal), mentre que l'única diferència que es produeix és un augment en la funció objectiu. Això és degut al fet que no disposem d'observacions que siguin linealment separables, fet que provoca que no siguem capaços d'obtenir una bona separació mitjançant els pesos, justificant el baix encert. Per contra, en augmentar el valor de ν , estem augmentant el valor pel qual multipliquem els *slacks* provocant un major valor de f .

RBF

ν	0	0,1	0,5	2	5
γ	-1	0,625	0,518	0,604	0.809
Acc Train	48%	78%	94%	98%	97%
Acc Test	50%	70%	86%	90%	88%
f	0,000	7,621	20,931	46,609	71,768

Com podem observar, l'aplicació del RBF suposa una notable millora en el percentatge de classificació correcta, tant en el *training* com en el test. En aquest cas, l'aplicació del kernel gaussià ens permet una modificació de l'espai de les dades, fet que facilita la seva classificació. Respecte a la tendència observem el mateix que per al cas anterior, augmentant el valor de γ i f a mesura que augmenta ν (tot i que per a $\nu = 0,1$ la γ disminueix).

3.2.3 Diabetes Data Set

Amb l'objectiu d'observar la capacitat de predicció del SVM amb dades més “reals” s'ha seleccionat una data base que té com a tema la diabetis. En aquesta, 8 variables explicatives s'han seleccionat per determinar si una persona disposa de diabetis (1) o no (-1). Per tal de reduir les dimensions de la taula, mostrarem únicament el valor dels 4 primers pesos, podent consultant-se la resta, mitjançant l'execució del programa, en la terminal.

Indicar que per a poder disposar d'un set de dades de test, s'ha realitzat una divisió de la base en $\frac{2}{3}$ per a entrenament i $\frac{1}{3}$ per a test.

Formulació primal/dual

v	0	0,1	0,5	2	5
γ	0	6,760	6,886	6,807	-0,002
w_1	0	-0,105	-0,107	-0,106	-0,106
w_2	0	-0,032	-0,032	-0,032	-0,032
w_3	0	0,012	0,012	0,012	0,012
w_4	0	0,001	0,001	0,001	0,001
Acc Train	33,203%	78,125%	78,320%	78,320%	78,320%
Acc Test	38,281%	75,391%	75,391%	75,391%	75,391%
f	0	25,350	126,413	505,256	1262,927

Com podem comprovar, en aquest cas l'SVM disposa del mateix comportament que per a *swiss roll*. L'augment de v permet un augment de l'*accuracy*, encara que arribat un punt, encara augmentar el seu valor les variables es mantenen i no som capaços de millorar les prediccions, augmentant únicament el valor de la funció objectiu.

RBF

v	0	0,1	0,5	2	5
γ	1	0,501	0,031	0,106	0,324
Acc Train	66,797%	72,070%	79,101%	81,836%	84,570%
Acc Test	61,719%	67,578%	72,265%	73,047%	73,828%
f	0,000	31,508	131,594	452,113	1010,194

Com podem comprovar, en aquest cas no es produeix un “estancament” en la predicció, sinó que aquesta continua augmentant a mesura que augmentem el valor de v . Procedim per tant a provar valors superiors per observar el nou comportament de la SVM.

v	10	50	100
γ	0,537	1,155	1,282
Acc Train	86,523%	90,820%	92,187%
Acc Test	71,484%	71,484%	69,531%
f	1824,504	6847,318	11708,441

Podem visualitzar com, a mesura que augmenta el valor de v , també ho fa el percentatge d'encert en les dades d'entrenament, però disminueix per a les dades de test. Aquest és un indicador de què estem realitzant un possible sobre ajustament per a les d'entrenament.

4. Conclusions

Com a principals conclusions podem extreure que la SVM amb $\Phi = I$ proporciona molts bons resultats quan les dades són separables linealment, però quan aquest no és el cas, és incapaç d'establir un bon pla de separació. Per contra, el kernel RBF, per a totes les dades provades en aquesta pràctica, ha funcionat de manera correcta, mitjançant la modificació de l'espai d'entrada ha estat capaç d'aconseguir prediccions força acurades. Aquest fet ens demostra, per tant, que l'elecció d'un kernel o un altre en funció de les dades a tractar resulta crucial per aconseguir un bon rendiment de la SVM.

Per altra banda, cal destacar que tal com s'ha anat observant, el paràmetre ν afecta el rendiment del SVM. Per tant, no es pot escollir un valor trivial sinó que és necessari realitzar estudis per assegurar que l'elecció del kernel i el valor de ν són els més apropiats per assegurar el màxim rendiment teòric.

Des d'un punt de vista més teòric, la pràctica també ens ha ajudat a comprovar el *Strong Duality problem* de manera visual, així com treballar la conversió de formulació primal a dual.