Optimització Matemàtica:
Support Vector Classifier in AMPL

Pau Lozano Garcia
Marina Rosell Murillo
Curs 2019/2020
Professor: Jordi Castro

Índex:

Introducció

L'objectiu d'aquest projecte és implementar les formulacions primals i duals d'una Support Vector Machine que classifiqui en l'entorn AMPL. S'utilitzarà el Kernel lineal per les implementacions primal i dual, i després, el Kernel Gaussià per una altra implementació dual. Compararem la seva eficàcia a l'hora de classificar correctament els diferents datasets que utilitzarem.

Distribució del projecte:

- En primer lloc, aplicarem la nostra implementació de SVM a un conjunt de dades obtingut amb el generador proporcionat "gensvmdat", que genera aleatòriament punts a l'espai de dimensió 4. Aquestes dades no són linealment diferenciables i tenen un factor aleatori.

- Aplicarem, en segon lloc, la implementació de SVM a un conjunt de dades extret d'internet. Les dades escollides per classificar en aquest apartat consisteixen en 2 tipus de llavors de cereals diferents a les quals s'han avaluat un seguit de característiques morfològiques dels grans. I fent servir les seves característiques tractarem de classificar-les amb les SVM amb Kernel lineal i Gaussià.

- A continuació, utilitzarem la nostra implementació de SVM a un dataset diferent que també serà linealment no diferenciable, que obtindrem a partir d'un swiss-roll. S'espera observar millors resultats utilitzant el Kernel Gaussià.

- Tant per les dades obtingudes amb el generador com per les extretes d'internet i per les obtingudes a partir del swiss-roll, es realitzarà una validació dels models resultants de les prediccions amb data sets diferents als de training.

Implementació d'una SVM:

Una Support Vector Machine és una tècnica utilitzada en la presa de decisions i en la classificació de dades en dues classes: +1 i -1. Consisteix en un problema d'optimització de 3 paràmetres formants de l'hiperplà separador de les dades. Amb la següent forma:

$$y(x) \cdot \varphi(x)^{T} \cdot w + \gamma \geq 1$$

L'objectiu és trobar un hiperplà que separi les dades amb el màxim marge possible per tal de tenir les classes ben diferenciades.

El problema d'optimització és el següent:

$$\begin{aligned} \min_{(w,\gamma,s)\in\mathbb{R}^{N+1+m}} \quad & \frac{1}{2} w^{\top} w + \nu \sum_{i=1}^{m} s_i \\ \text{s. to} \quad & y_i(w^{\top}\phi(x_i) + \gamma) + s_i \geq 1 \quad i = 1, \ldots, m \\ & s_i \geq 0 \qquad\qquad\qquad\qquad\quad i = 1, \ldots, m \end{aligned}$$

On la minimització del producte escalar de $w^T w$, maximitza la separació dels dos hiperplans x^T w = a, x^T w = b (augmentant el marge entre els dos grups).
Els valors de la variable s, corresponen a la distància que hi ha entre un punt mal classificat i la classe a la que correspon, per tant, la minimització de:

$$\upsilon \sum_{i=1}^{m} s_i$$

és la minimització de l'error comès en la classificació.
En el cas d'un Kernel Lineal, la funció phi(x) correspon a la identitat, per tant:

$$\varphi(x) = x$$

Naturalment, cal esperar que la solució del problema Primal i Dual quan fem servir el kernel lineal sigui la mateixa.

Per altra banda, també enfocarem el problema des del punt de vista dual:

$$\begin{aligned} \max_{\lambda} \quad & \sum_{i=1}^{m} \lambda_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \lambda_i y_i \lambda_j y_j K_{ij} \\ & \sum_{i=1}^{m} \lambda_i y_i = 0 \\ & 0 \leq \lambda_i \leq \nu \quad i = 1, \ldots, m \end{aligned}$$

On ara lambda és la variable d'optimització i K_{ij} correspon a la funció del kernel, i es calcula de la següent forma:

$$K(x, y) = \varphi(x) \cdot \varphi(y)$$

Quam fem servir el kernel Gaussià el calculem amb

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

Els fitxers dels diferents models utilitzats, amb el Kernel lineal i el kernel Gaussià s'adjunten en la carpeta del projecte com a fitxers apart.

En el cas d'un Kernel Gaussià, desconeixem phi(x), de manera que només podem utilitzar el mètode dual, ja que es pot tractar amb el kernel directament i no amb la funció phi(x) amb el "Kernel Trick".

Generació i obtenció de les dades

En primer lloc, hem generat les dades artificials amb el generador proporcionat, que genera un conjunt de punts aleatoris en 4 dimensions. Aquests punts en l'espai estan classificats a la classe +1 si la suma de les quatre coordenades és major o igual a 2, i a la classe -1 si aquesta suma és inferior a 2. Aquest paràmetre de classe és a la darrera columna de la matriu de dades.
Per fer que les dades siguin linealment no separables hi ha un subconjunt de punts distribuïts aleatòriament que són classificats a la classe contrària a la que els hi tocaria seguint el criteri mencionat, aquests punts són marcats amb un asterisc. La base de dades utilitzada és de 1000 observacions.

El dataset obtingut per internet que hem escollit tracta d'una classificació de llavors de blat de tres varietats diferents: "Kama", "Rosa" i "Canadian" amb 70 mostres de cadascuna. S'han pres diferents mesures de les llavors mitjançant radiografies, i es proporcionen set variables quantitatives: àrea, perímetre, compactesa, longitud del gra, amplada del gra, coeficient d'asimetria i longitud de la ranura del gra; i la varietat a la que pertany cada llavor.
Les dades són extretes d'un estudi que es va dur a terme fent servir llavors de blat procedents de camps experimentals, explorats a l'Institut d'Agrofísica de l'Acadèmia de Ciències Polonesa de Lublin[1].

Per generar les dades del tipus Swiss-Roll hem implementat un codi de python, que adjuntem amb la resta de fitxers, on hem fet servir la funció de python *sklearn.datasets.make_swiss_roll()*. Aquesta funció retorna un conjunt de dades a l'espai tridimensional que tenen forma de "braç de gitano" enrotllat.
Per definir les classes dels punts generats, hem fet servir el paràmetre t que retorna la funció, que correspon a la posició que tindrien els punts si desenrotllèssim les dades. Hem definit que les dades pertanyen a la classe +1 si el paràmetre t és per sobre de la mitjana, i a la classe -1 si és per sota. Per aquestes dades també hem generat 1000 observacions.

[1] M. Charytanowicz, J. Niewczas, P. Kulczycki, P.A. Kowalski, S. Lukasik, S. Zak, 'A Complete Gradient Clustering Algorithm for Features Analysis of X-ray Images', in: Information Technologies in Biomedicine, Ewa Pietka, Jacek Kawa (eds.), Springer-Verlag, Berlin-Heidelberg, 2010, pp. 15-24. https://archive.ics.uci.edu/ml/datasets/seeds#

Modificació dels arxius de dades

1. Amb un programa de R que afegim a la carpeta entregada hem fet les modificacions necessàries per a cada dataset per tal de poder ser tractades en AMPL.

   - Per les dades obtingudes amb el generador, hem eliminat els asteriscs de les dades ja que l'AMPL no els llegeix correctament.

   - Pel cas de les dades de les llavors, donat que tenim tres varietats i la Support Vector Machine implementada només classifica en dues categories, hem eliminat la tercera varietat que correspon a les llavors "Canadian" i tractarem de classificar les de les varietats "Kama" i "Rosa", que corresponen a les classes +1 i -1 respectivament. Així doncs la base de dades amb la que treballarem és de 139 mostres. A més, hem reordenat la base de dades aleatòriament, ja que inicialment estaven ordenades segons la variació, i necessitem que estiguin desordenades per poder fer una bona partició de training i testing.

   - Les dades que provenen del generador de python de Swiss-Roll no necessiten cap modificació específica.

2. A continuació, hem dividit els conjunts de dades en dos grups: un de training amb dos terços de les dades totals, que farem servir per entrenar la SVM i conseguir l'hiperplà de separació (vector w) òptim; i un altre grup de validació, que comporta el terç restant de dades i farem servir més tard per comprovar el correcte funcionament de l'hiperplà de separació generat per la Support Vector Machine.
   A les dades de training hem afegit una columna d'índex per que l'AMPL pugui llegir les dades com una matriu i hem guardat els datasets a arxius amb format .dat. També hem calculat les variàncies de totes les columnes de les diferents bases de dades i hem calculat la mitjana de les variàncies de cada dataset. Aquest paràmetre l'anomenem sigma i forma part del kernel Gaussià.

3. Per últim, pels tres casos, és necessari editar manualment els fitxers de dades de training, que és amb els que implementarem la SVM, abans d'executar el codi d'AMLP, ja que cal declarar els següents paràmetres:
   - m: nombre d'observacions.
   - n: dimensió de les dades.
   - nu: paràmetre de regularització de la SVM.
   - A: matriu de dades.
   - sigma: variància de les dades

L’arxiu de dades que proporcionem a l’entrega ja ha estat modificat i està llest per executar, i la línia de codi que sobreescriu els arxius de dades està comentada per poder executar-lo sense esborrar les modificacions.

- Pel cas de les dades generades amb el generador proporcionat i el Swiss-Roll, el paràmetre m pren per valor 667 (dos terços de 1000).
- Per les dades de les llavors m pren per valor 97. El paràmetre n prendrà per valor 4, 7 i 3 per les dades del generador proporcionat, de les llavors i del Swiss-Roll respectivament.
- El paràmetre de regularització nu l’hem fixat a 5 per tots tres arxius de dades.
- El valor paràmetre sigma depèn de cada dataset i és calculat a l’arxiu R, l’hem arrodonit a un decimal.

Execució de la implementació de la Support Vector Machine a AMPL

Per l'execució a AMPL només cal axecutar l'arxiu "SVM.run" on es carreguen les nou combinacions dels tres datasets de training amb els tres models Support Vector Machine: primal, dual amb kernel lineal i dual amb kernel Gaussià.
Definim el CPLEX com l'algorisme que fem servir per resoldre els problemes d'optimització.

La sortida que obtenim de la crida d'aquest arxiu és la solució del problema d'optimització amb el valor de la funció objectiu minimitzat, i a continuació la precisió d'encert obtinguda sobre les dades de training calculada a partir dels valors de les variables s en el cas primal, i segons els valors de les lambdes en els casos duals.

Hiperplans de separació generats pels SVM amb Kernel Lineal.

- Hiperplà creat a partir de les dades generades per "Gensvmdat":

	w1	**w2**	**w3**	**w4**	**gamma**
primal	4.3755	4.8239	4.9935	4.6075	-9.4121
dual	4.3755	4.8239	4.9935	4.6075	-9.4121

- Hiperplà creat a partir de les dades de les llavors dels cereals:

	w1	**w2**	**w3**	**w4**	**w5**	**w6**	**w7**	**gamma**
primal	-0.2603	-0.8104	0.0994	0.9798	0.8718	-0.5057	-4.1624	32.7539
dual	-0.2602	-0.8104	0.0994	0.9798	0.8718	-0.5057	-4.1624	32.7543

- Hiperplà creat a partir de les dades del swiss-roll:

	w1	**w2**	**w3**	**gamma**
primal	0.0305	-0.0094	-0.124	-0.1558
dual	0.0305	-0.0094	-0.124	-0.1558

Comprovem que els hiperplans de separació del model SVM dual i primal coincideixen, amb petites diferències en algun decimal remot que considerem negligibles. Les gammes també coincideixen.
Amb això, es corrobora la hipòtesi de que els resultats que obtenim amb la SVM primal i SVM dual amb kernel lineal són els mateixos.

Precisió de les SVM:

A continuació, es mostra el percentatge d'encert de classificació obtingut aplicant la Support Vector Machine amb el Kernel lineal o Gaussià a les diferents bases de dades, tant a les de training com a les de testing. Aquests càlculs els hem realitzat des de l'arxiu R segons la fórmula:

$$y(x) \cdot \varphi(x)^T \cdot w + \gamma \geq 1$$

Comptarem com a ben classificades les dades que compleixin la desigualtat.
En els casos dels kernel lineals farem servir $\varphi(x) = x$. En els casos que fem servir el kernel Gaussià donat que no coneixem la funció de transformació $\varphi()$ ni obtenim el paràmetre gamma de la optimització, els haurem de recuperar a partir de les lambdes i fent servir el kernel Gaussià, segons les següents fórmules.

Donat que:

$$w = \sum_{i=1}^{m} \lambda_i \cdot y_i \cdot \varphi(x_i) \qquad \text{i} \qquad K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j)$$

Obtenim que:

$$\varphi(x)^T \cdot w = w^T \cdot \varphi(x) = \sum_{i=1}^{m} \lambda_i \cdot y_i \cdot \varphi(x_i) \cdot \varphi(x) = \sum_{i=1}^{m} \lambda_i \cdot y_i \cdot K(x_i, x)$$

$$\gamma = \frac{1}{y_i} - w^T \varphi(x_i) = \frac{1}{y_i} - \sum_{j=1}^{m} \lambda_j \cdot y_j \cdot \varphi(x_j)^T \cdot \varphi(x_i) = \frac{1}{y_i} - \sum_{j=1}^{m} \lambda_j \cdot y_j \cdot K(x_i, x)$$

Precisió de la SVM amb Kernel lineal:

	GENSVMDAT	LLAVORS	SWISS-ROLL
Training	68%	90%	29%
Testing	70%	89%	27%

Precisió de la SVM amb Kernel Gaussià:

	GENSVMDAT	LLAVORS	SWISS-ROLL
Training	80%	95%	98%
Testing	74%	61%	92%

Conclusions:

Un cop finalitzats tots els càlculs extraiem les següents conclusions:

1. **Kernel Lineal**

   S'observa clarament com l'error de training i de validació és similar, però varia molt en funció de la base de dades sobre la qual apliquem la SVM:

   - Quan les dades no son linealment separables però la funció amb que s'han generat és lineal (les generades amb l'executable "gensvmdat") veiem com la precisió és raonablement bona i gira al voltant del 70%.

   - Per a les dades que hem proposat nosaltres, les de les llavors, hem obtingut uns resultats molt positius, la SVM ha estat capaç de classificar correctament en el 90% dels casos. Creiem que això és degut a que les llavors tenen característiques morfològiques prou diferents entre elles i que ens permeten classificar-les amb facilitat.

   - Per últim, respecte a les dades que provenen del swiss-roll, era d'esperar que la predicció fos molt dolenta, perquè la funció amb que s'han generat les dades no és gens lineal (té forma d'espiral), per tant és molt difícil que un hiperplà pugui classificar bé les dades si aquestes no s'han modificat prèviament.

   Així doncs, creiem que el Kernel Lineal pot ser adequat en casos concrets en que les dades segueixen una distribució lineal o quasi lineal, perquè és prou fiable i a més és molt simple, de manera que els càlculs son ràpids.
   No obstant, quan les dades segueixen funcions més complexes, no és un mètode adequat.

2. **Kernel Gaussià**

   S'observa que el comportament de la SVM és similar quan utilitzem el kernel RBF o el kernel lineal sempre i quan la distribució de les dades sigui lineal o quasi lineal. En canvi, quan les dades segueixen una funció clarament no lineal (Swiss-Roll) el comportament de la SVM amb RBF és molt millor i assoleix una precisió molt bona, ja que la funció phi que forma el kernel és capaç de transformar les dades de forma que siguin linealment separables. El cas del Swiss-Roll és un exemple clar d'això: si el kernel és capaç de desenrotllar les dades, aquestes són linealment separables i no hi ha aleatorietat com en el cas de les dades provinents del generador.

Hem observat que el paràmetre sigma del kernel RBF té un paper força important en la optimització, ja que no a totes les bases de dades ens ha anat bé amb el mateix valor:

- Si les dades tenen una variància baixa, com és el cas de les dades que hem obtingut amb el generador i les dades de les llavors, un valor de sigma baix anirà bé per resoldre el problema d'optimització.

- En canvi, si es fa servir una sigma baixa per unes dades amb variància més alta, com és el cas de les dades del Swiss-Roll, el resultat que s'obté de l'optimització és que quasi tots els punts són Support Vectors, ja que la majoria de les lambdes prenen valors entre 0 i nu. Això comporta que la precisió de training és molt alta, pràcticament del 100% d'encert, però en canvi la precisió de testing és summament baixa, al voltant de 25%.
  Amb tot això, creiem que es tracta d'un problema de sobreajust, i que per tant, és necessari augmentar el valor del paràmetre sigma.

- Per altra banda, quan es fa servir una sigma excessivament alta, la precisió de training baixa considerablement, i pràcticament no hi ha cap punt que sigui Support Vector.

Per això, hem utilitzat la variància de les dades de cada dataset com a paràmetre sigma, i tots ells han donat resultats molt bons tant al training com al test.