ÎNVĂȚAREA UTILIZÂND CONCEPTUL DE „SUPPORT VECTOR”

Asist. ing. drd. Daniel MORARIU

Asist. ing. drd. Gabriel CUREA

(daniel.morariu@ulbsibiu.ro, gabriel.curea@ulbsibiu.ro)

UNIVERSITATEA „LUCIAN BLAGA” SIBIU,

FACULTATEA DE INGINERIE „HERMANN OBERTH”,

CATEDRA DE CALCULATOARE ŞI AUTOMATIZĂRI

Abstract

Algoritmul de clasificare bazat pe conceptul SV (support vector) este considerat unul dintre cei mai interesanți ai teoriei de învățare statistică. Se pleacă de la un concept foarte simplu de grupare a noului obiect la clasa față de care este cel mai apropiat (cu care este similar). Ideea algoritmului constă în a găsi un spațiu de reprezentare al datelor de intrare în care acestea să fie liniar separabile şi găsirea unui *hiperplan optim* care separă cel mai bine cele două clase. De obicei calcularea hiperplanului este imposibilă şi de aceea se merge pe ideea de a calcula granițele de separare ale acestui hiperplan numite *margini* cu o anumită eroare. In continuare ne vom referi şi vom analiza doar clasificarea în două clase.

1. Introducere

Una din problemele fundamentale ale teoriei învățării este următoarea: presupunem că se dau două clase de obiecte, apoi avem de-a face cu un nou obiect care trebuie să îl atribuim uneia din cele două clase. Această problemă ar putea fi formalizată după cum urmează. Se dau datele empirice:

$$(x_1, y_1),...,(x_m, y_m) \in \chi \times \{\pm 1\} \tag{1}$$

Unde, χ este o mulțime nevidă din care sunt luate *datele de antrenament* x_i.

Ideea în clasificare este de a generaliza datele de intrare astfel încât clasificarea să funcționeze şi pe date necunoscute. Într-un caz generalizat, dându-se informații noi *x*, vrem să „clasificăm" cât mai bine ieşirea corespunzătoare, adică, să alegem o ieşire *y* astfel încât (*x,y*) să fie într-o oarecare măsură „similar" cu exemplele de antrenament. Pentru a măsura similaritatea datelor considerăm o funcție de forma $k : \chi \times \chi \to \Re \qquad (x, x') \mapsto k(x, x')$ care pentru două date de intrare x şi x' întoarce un număr real ce caracterizează similaritatea lor. Un caz simplu de măsură a similarității este produsul scalar definit astfel $\langle x, x' \rangle := \sum_{i=1}^{N} [x]_i [x']_i$, unde prin $[x]_i$ notăm intrarea i a vectorului x.

Un algoritm simplu de clasificare, pentru care presupunem că datele noastre sunt într-un spațiu prehilbertian $\mathcal{H}$ înzestrat cu produs scalar, clasificarea unui nou obiect pe baza similarității acestuia, constă în măsurarea distanței în acest spațiu. Ideea de bază a algoritmului este de a atribui unei noi date de intrare clasa cu media cea mai apropiată.

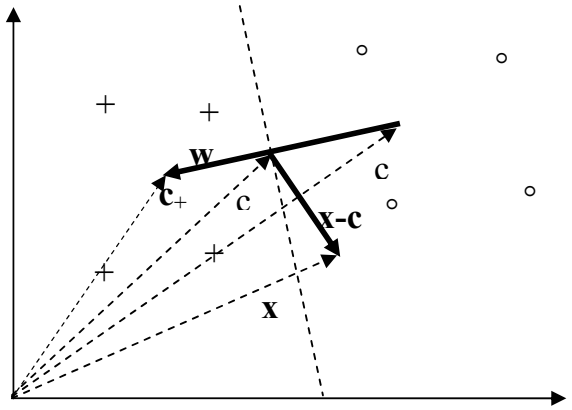

Figura 1

În Figura 1 am prezentat un astfel de exemplu de clasificare unde c_+ şi c_- reprezintă mediile celor două clase (centrele de greutate), iar **c:=(c₊ + c₋)/2**. Calculăm clasa lui **x** verificând dacă vectorul diferență **x-c** are un unghi mai mic decât $\pi/2$ cu vectorul **w := c₊ - c₋** dintre mediile claselor. Aceasta conduce la o formulă care poate fi exprimată în termeni de produs scalar:

$$y = \operatorname{sgn}\langle(\mathbf{x}-\mathbf{c}), w\rangle = \operatorname{sgn}\langle(x-(c_+ + c_-)/2), (c_+ - c_-)\rangle = \operatorname{sgn}\left(\langle x, c_+\rangle - \langle x, c_-\rangle + b\right) \quad (2)$$

unde sgn este funcția semn iar *b* reprezintă ofsetul şi are valoarea $b := \frac{1}{2}(\|c_-\|^2 - \|c_+\|^2)$ iar $\|x\| = \sqrt{\langle \mathbf{x}, \mathbf{x}\rangle}$.

Observăm că, (2) induce o graniță a deciziei care are forma unui „hiperplan"; adică, o mulțime de puncte care satisfac o constrângere exprimată sub formă de ecuație liniară. Dacă renunțăm la centrele claselor şi rescriem expresia (2) în termeni de date de intrare x_i, utilizând nucleul k pentru a calcula produsele scalare $(k : \chi \times \chi \to \Re\ (x, x') \mapsto k(x, x'))$ obținem o *funcție de decizie* mult mai generală pe baza căreia se realizează clasificarea noilor paternuri [1] .

$$y = \operatorname{sgn}\left(\frac{1}{m_+}\sum_{\{i|y_i=+1\}} k(x, x_i) - \frac{1}{m_-}\sum_{\{i|y_i=-1\}} (x, x_i) + b\right) \quad (3)$$

Dându-se câteva puncte x, etichetarea se calculează simplu maximizând expresia (3). Simplificat funcția de decizie ar fi conform [1]:

$$y = \operatorname{sgn}\left(\sum_{i=1}^{m} \alpha_i k(x, x_i) + b\right) \quad (4)$$

Expasiunea lui y corespunde separării hiperplanelor din spațiul trăsăturilor caracteristice. În acest sens, α_i poate fi considerat *reprezentarea duală* a vectorului normal la hiperplan. Ambii clasificatori din (3) se bazează pe exemple în sensul în care nucleul este calculat pe baza datelor de antrenament; adică, unul din argumentele nucleului este o dată de antrenament. Un punct de test este clasificat prin compararea lui cu toate punctele de antrenament care apar în (4) cu o pondere diferită de 0.

2. Conceptul de „Support Vector"

Tehnica de clasificare pe care o abordăm în acest articol derivă din (4) şi se bazează în special pe selecția datelor de intrare pentru calculul nucleului şi alegerea ponderilor α_i care sunt plasate pe nucleele individuale în funcția de decizie. Nu va mai fi cazul ca toate datele de antrenament să apară în calcularea

nucleului. În reprezentarea în spațiul trăsăturilor caracteristice, această afirmație se rezumă la studiul vectorilor normali **w** ai hiperplanului care pot fi reprezentați ca şi combinații liniare generale(de ex. cu coeficienți neuniformi) ai datelor de antrenament. De ex. vrem să ştergem influența patter-nurilor care sunt foarte depărtate de granița de decizie, fie că ne aşteptăm ca ele să influențeze foarte puțin generalizarea funcției de decizie, fie că vrem să reducem costul de calcul a evaluării funcției de decizie. Hiperplanul va depinde doar de o submulțime de patternuri de antrenament numit *support vector*.

Pentru a putea proiecta un algoritm de învățare al cărui grad de eficiență statistică să poate fi controlată, acesta trebuie să vină cu o clasă de funcții care pot fi calculate. Aşa cum este prezentat în [3], se consideră în $\mathcal{H}$ hiperplane de forma ‹**w,x**›+b=0 unde **w** □$\mathcal{H}$ şi b □ □

(5)

corespunzând funcțiile de decizie

$$f(\mathbf{x}) = \mathrm{sgn}(\langle \mathbf{w},\mathbf{x} \rangle + b) \qquad (6)$$

şi se propune un algoritm de învățare pentru probleme separabile (numite uneori şi probleme *liniar separabile*), prin construirea lui f din datele de intrare. Acesta se bazează pe doi factori: primul este că din toate hiperplanele care separă date, există un *hiperplan optimal unic*, distins prin marginea maximă de separație dintre orice punct de antrenament şi hiperplan, acesta fiind o soluție a:

$$\underset{w\in\mathcal{H},b\in\Re}{\text{maximize}} \min\left\{\|\mathbf{x}-\mathbf{x_i}\| \,\middle|\, \mathbf{x}\in\mathcal{H}, \langle \mathbf{w},\mathbf{x}\rangle + b = 0, i = 1,...,m\right\} \qquad (7)$$

iar al doilea, capacitatea de separare a claselor descreşte cu incrementarea marginilor. Deci există argumente teoretice care susțin performanțele bunei generalizări a hiperplanului optim. În cazul de față, trebuie să calculăm vectorul normal care conduce la cea mai mare margine. Pentru a construi hiperplanul optimal, trebuie să rezolvăm

$$\underset{\mathrm{w}\in\mathcal{H},\mathrm{b}\in\Re}{\text{minimize}}\ \tau(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 \qquad (8)$$

cu condiția $y_i(\langle \boldsymbol{w},\boldsymbol{x}_i \rangle + b) \geq 1$ pentru toți i=1,...,m. (9)

Aceste constrângeri asigură că $f(\mathbf{x}_i)$ va fi +1 pentru $y_i = +1$ şi -1 pentru $y_i = -1$. Expresia „≥1” din partea dreaptă a constrângerii fixează eficient scalarea lui **w**. De fapt, orice alt număr pozitiv o va face. Să încercăm acum să obținem intuitiv de ce trebuie să fie minimizată lungimea lui **w** (Figura 2), ca în (8). Dacă ||**w**|| ar fi 1, atunci partea stângă din (9) va egala distanța de la x_i la hiperplan (conform (7)). În general, trebuie să împărțim $y_i(\langle \mathbf{w},\mathbf{x}_i \rangle + b)$ prin ||**w**|| pentru a transforma aceasta în distanță. De acum încolo, dacă putem satisface (8) pentru toți i=1,..,m cu un **w** de lungime minimă, atunci marginea totală va fi maximizată.

Funcția τ din (8) este numită *funcția obiectiv*, în timp ce (9) sunt numite *inegalitățile de constrângere*. Împreună, acestea formează aşa numita *problemă de optimizare cu restricții*.

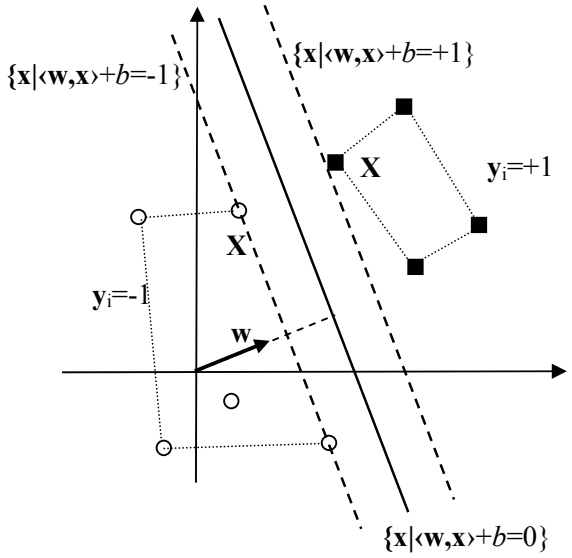

Figura 2

3. Calcularea marginii hiperplanului

În practică, hiperplanul de separare nu este întotdeauna cea mai bună soluție a problemei de clasificare. După toate, o valoare aberantă în datele de intrare, de exemplu o valoare care este etichetată greşit, poate afecta crucial hiperplanul. Ideea naturală este de a găsi un hiperplan care conduce la *minimul* de erori de antrenament. Soluția este de a găsi o serie de constrângeri pe baza cărora să se poată determina acei SV care stau pe marginile hiperplanului de

separare. Utilizând multiplicatorii Lagrange şi condiţiile Karush-Kuhn-Tuker (KKT) (conform [1]) rezultă următoarele constrângeri:

$$\frac{\partial}{\partial b} L(\mathbf{w}, b, \alpha) = 0 \quad si \quad \frac{\partial}{\partial \mathbf{w}} L(\mathbf{w}, b, \alpha) = 0 \tag{10}$$

conducând la $\sum_{i=1}^{m} \alpha_i y_i = 0 \quad si \quad w = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$ (11)

unde, $L(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{m} \alpha_i (y_i (\langle \mathbf{x_i}, w \rangle + b) - 1)$ (12)

Vectorul soluţie este o expansiune în raport cu o submulţime de date de antrenament, anume acele date cu α_i diferit de zero, numite vectori suport. Alături de condiţiile KKT,

$\alpha_i [y_i (\langle \boldsymbol{x_i}, \boldsymbol{w} \rangle + b) - 1] = 0$ pentru toţi i=1,...,m (13)

vectorii suport stau pe marginea hiperplanului. Toate exemplele de antrenament rămase (x_i, y_i) sunt irelevante: restricţiile lor $y_i (\langle \boldsymbol{x_i}, \boldsymbol{w} \rangle + b) - 1 \geq 1$ (conform 8) pot fi foarte bine eliminate şi nu apar în expansiunea (11). Deoarece hiperplanul (conform figuri 2) este complet determinat de către paternurile cele mai apropiate de el, soluţia nu ar trebui să depindă de celelalte exemple.

4. Trucul nucleu

Am explicat mai sus care este modalitatea de calcul a hiperplanului optim în cazul în care datele de antrenament sunt liniar separabile. Dacă acestea nu sunt liniar separabile atunci ideea învăţării prin conceptul de SV este să treacă datele de antrenament din spaţiul de intrare într-un spaţiu de dimensiune mai mare (în care datele devin liniar separabile) utilizând o funcţie de Φ: $x_i \rightarrow \mathbf{x_i}$. Aceasta permite o margine decizională neliniară în spaţiul de intrare. În toate calculele de mai sus datele $\mathbf{x_i}$ nu trebuie să coincidă cu datele de intrare putând foarte bine să fie rezultatul mapării datelor de intrare x_i într-un spaţiu de trăsături de o dimensiune mai mare. Prin utilizarea funcţiei nucleu (care mai este numită şi trucul nucleu) $k(x, x_i) := \langle \mathbf{x}, \mathbf{x}_i \rangle = \langle \Phi(x), \Phi(x_i) \rangle$ este posibil să calculăm

hiperplanul de separație fără a găsi efectiv funcția de trecere în spațiul caracteristicilor.

Trucul nucleu poate fi aplicat de vreme ce toți vectorii caracteristici apar în produsele scalare. Vectorul ponderilor (conform (10)) devin atunci o extensie a spațiului caracteristicilor şi de obicei Φ nu va mai corespunde imaginii unui spațiu vectorial de intrare singular. Obținem funcțiile de decizie de forma

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{m} y_i \alpha_i \langle \Phi(x), \Phi(x_i) \rangle + b \right) = \operatorname{sgn}\left(\sum_{i=1}^{m} y_i \alpha_i k(x, x_i) + b \right) \quad (14)$$

şi programul trebuie să calculeze:

$$\underset{\alpha \in \Re^m}{\text{maximize}}\, W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \quad (15)$$

cu condiția α_i , $\alpha_j \geq 0$ pentru toți i=1,..., m şi $\sum_{i=1}^{m} \alpha_i y_i = 0$ (16)

Folosind nucleele în locul produselor scalare, clasificatorul pe baza marginii optimale a fost schimbat într-un clasificator de înaltă performanță. Un exemplu de nucleu pentru care s-au obținut rezultate foarte bune este nucleul polinomial $k(x, x') = \langle x, x' \rangle^d$ sau nucleul Gaussian $k(x, x') = \exp(-\frac{\|x - x'\|^2}{2\sigma^2})$

5. Concluzii

În acest articol am prezentat o variantă generală a unui algoritm de învățare utilizând conceptul „Support Vector” în cazul în care există doar două clase. De obicei aceşti algoritmi diferă prin valorile apriori care trebuiesc luate în considerare. Acest algoritm poate fi generalizat pentru o clasificare în mai multe clase, astfel se construieşte o mulțime de clasificări binare $f^1,...,f^M$ pentru antrenarea separat a unei clase față de restul claselor şi se combină acestea pentru a face clasificarea în mai multe clase. Această variantă, deşi dă rezultate bune, este criticată pentru distribuirea mai degrabă a problemelor nesimetric şi pentru că implică o rezolvare euristică. O altă variantă este de a antrena clasificarea pentru fiecare pereche posibilă de clase. Deşi aceasta sugerează un

timp de antrenament mai vast, în probleme individuale avem apare un timp semnificativ mai mic, iar dacă algoritmul de antrenament depinde de dimensiunea mulțimii de antrenament, de fapt este posibil să salvăm timp.

În continuare autorii îşi propun să dezvolte aplicații ale acestor algoritmi în cazul extrageri de cunoştințe din date nestructurate.

Mulțumiri

Acest articol a fost elaborat în cadrul programului de doctorat urmat de autori la Universitatea „Lucian Blaga" din Sibiu (conducător ştiințific prof. univ. dr. ing. Lucian VINȚAN).

Autorii îşi exprimă întreaga gratitudine companiei SIEMENS AG, CT IC, MUENCHEN Germania – în special domnului vicepreşedinte Dr. H.C. mat. Hartmut Raffler – pentru sponsorizarea programului de doctorat precum şi pentru sugestiile profesionale extrem de utile.

Bibliografie

[1] B. Scholkopf and A. Smola, "Learning with Kernels, Support Vector Machines, Optimization and Beyond", MIT Press, 2002, pag. 189-220

[2] C. Nelso and J. Shawe-Taylor, "An introduction to Support Vector Machines and other kernel-based learning methods", Cambridge university Press, 200, pag. 93-122

[3] V. Vapnik şi A. Lerner, „Pattern recognition using generalized portaint method. Automotion and Remote control", Berlin, 1099, pag. 230-240

[4] S. Stuart, P. Norvic, "Artificial intelligence, a modern approach", Prentice Hall, 1995, pag. 588-593