Calcul Numeric

Laboratorul 4.

Metode kernel. Clasificare

1 Problema de clasificare

Clasificarea semnalelor presupune identificarea unei reguli care descrie modul în care semnalele din fiecare clasă se deosebesc de celelalte. Figura 1 prezintă un set de date cu două clase.

Proprietățile semnalelor diferă de la o clasă la alta, astfel că spațiul descris de date poate fi separat printr-o dreaptă, astfel încât semnalele din prima clasă să se regăsească de o parte a dreptei, iar cele din a doua clasă de celalaltă parte. Există mai multe posibile astfel de drepte, deci este nevoie de stabilirea unor criterii pentru a o alege pe cea mai convenabilă. O dată stabilită această regulă, orice semnal nou, despre care nu se cunoaște clasa, poate fi clasificat evaluând situarea acestuia față de dreaptă. Dacă semnalele au mai mult de 2 dimensiuni, separarea spațiului va fi dată de un (hiper-)plan și nu de o dreaptă, însă problema este echivalentă.

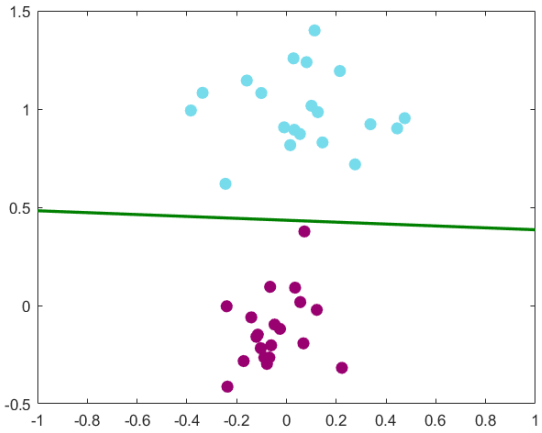

Figura 1: Clasificare binară.

		Actual Class	
		Positive (**P**)	Negative (**N**)
Predicted Class	Positive (**P**)	True Positive (**TP**)	False Positive (**FP**)
	Negative (**N**)	False Negative (**FN**)	True Negative (**TN**)

Figura 2: Matricea de confuzie.

Prima etapă - ce corespunde stabilirii regulii, se numește *antrenare* și se realizează folosind un set de date etichetat: pe lângă proprietățile fiecărui semnal, acesta este caracterizat și de o etichetă (*label*) care semnalează cărei clase aparține. Pentru un set de date $\boldsymbol{Y} \in \mathbb{R}^{m \times N}$, adică o colecție de N semnale de dimensiune m, etichetele reprezintă un vector de N elemente cu valori întregi în intervalul $[1, C]$, unde C este numărul de clase.

Observați că separarea spațiului semnalelor folosind o dreaptă (sau un hiperplan, pentru semnale de dimensiune $m > 3$) este utilă atunci când $C = 2$, ceea ce corespunde problemei de clasificare *binară*. În cazul în care sunt mai multe clase, metoda poate fi utilizată într-o abordare *one-versus-rest*, anume în care semnalele de un tip sunt separate de cele aparținând restului de $C - 1$ clase.

1.1 Metrici de performanță

Pentru a evalua performanța unui clasificator, cel mai des folosit criteriu este acuratețea clasificării. Este utilizat un set de date noi, conținând N_{test} semnale, pentru care se estimează valoarea etichetelor, iar rezultatul se compară cu valorile reale ale etichetelor

$$accurațe = \text{numărul de predicții corecte/numărul de predicții totale} \quad (1)$$

unde numărul de predicții totale este chiar dimensiunea setului de test, N_{test}.

O analiză mai amănunțită se poate realiza utilizând matricea de confuzie. Aceasta reprezintă o matrice de dimensiune $C \times C$, în care semnificația liniilor ține de valorile reale, iar cea a coloanelor de valorile estimate pentru fiecare clasă. Figura 2 prezintă o matrice de confuzie pentru problema de clasificare binară: elementele de pe diagonala principală reprezintă numărul de semnale clasificate corect (din fiecare clasă), în timp ce elementele de pe diagonala secundară pe cele clasificate greșit.

2 Funcții kernel

Problema descrisă anterior reprezintă cazul cel mai simplu, în care datele sunt perfect separabile: niciun punct mov din Figura 1 nu se regăsește de partea albastră a planului. În practică puține probleme sunt atât de simple, cel mai

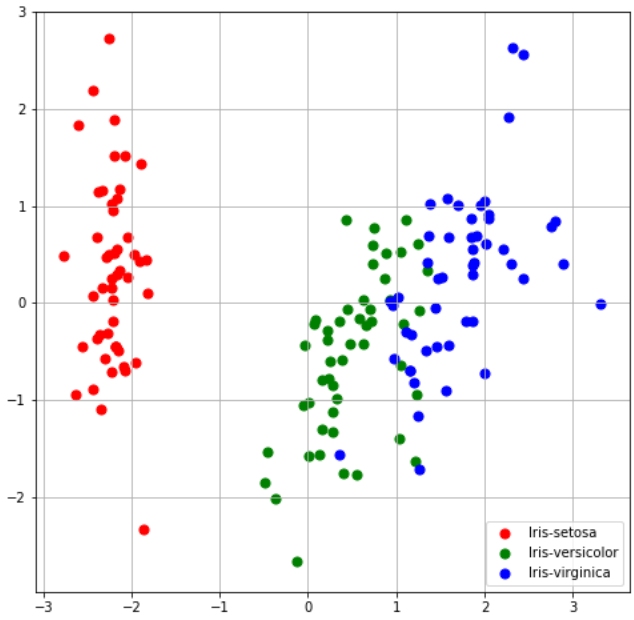

Figura 3: Baza de date de flori după aplicarea PCA.

adesea spațiul semnalelor nu este perfect separabil. Pentru a ilustra o astfel de situație, folosim baza de date Iris din laboratorul precedent, disponibilă la `https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data`.

Aceasta conține $N = 150$ măsurători reprezentând dimensiunile unor flori ce se încadrează în 3 tipuri: Iris Setosa, Iris Versicolor, Iris Virginica. Pentru fiecare floare sunt măsurate lungimea și lățimea petalei și lungimea și lățimea sepalei, deci dimensiunea unui semnal este $m = 4$. Prin urmare, nu putem vizualiza spațiul semnalelor. În laboratorul trecut ați folosit algoritmul PCA pentru a reduce dimensiunea semnalelor la $m = 2$, deci putem folosi această reprezentare pentru a explora problema. Cele 2 dimensiuni vor fi acum abstracte, nu vor avea (neapărat) un corespondent direct în proprietățile fizice ale semnalelor și vor simplifica problema, însă ne putem face o idee despre cât de ușor se pot separa semnalele în clase diferite.

Figura 3 ilustrează transpunerea setului de date într-un spațiu de dimensiune $m = 2$ utilizând PCA. Florile Iris Setosa au proprietăți semnificativ distincte față de celelalte 2 clase, deci se poate găsi cu ușurință o dreaptă care să separe această clasă de restul. Însă distincția între Iris Versicolor și Iris Virginica nu este atât de evidentă. Nu putem găsi o dreaptă care să separe perfect semnalele din cele două clase. Dacă aceasta se întâmplă în spațiul de dimensiune redusă, ne așteptăm ca observația să fie valabilă și în spațiul original al semnalelor.

Totuși, există posibilitatea de a găsi un plan de separare, dacă în loc să scădem dimensiunea semnalelor (cum am făcut la PCA), o creștem. Figura 4

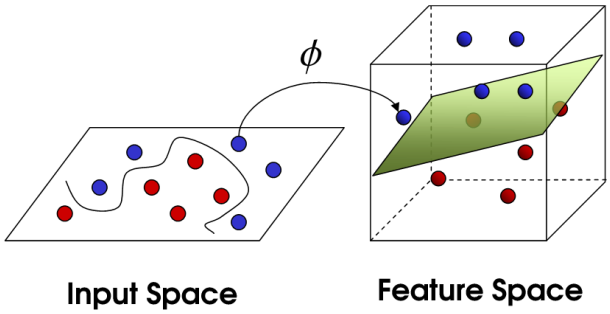

Figura 4: Maparea semnalelor într-un spațiu de dimensiunea mai mare.

prezintă o astfel de situație.

Prin urmare, pentru a putea folosi metoda de clasificare descrisă până acum, e necesar ca întâi datele să fie proiectate într-un spațiu de dimensiunea $m_k > m$. Acest lucru poate fi realizat utilizând funcții kernel.

Printre cele mai des utilizate funcții kernel amintim
Kernel liniar: $K(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^\top \boldsymbol{y}$.
Kernel polinomial: $K(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^\top \boldsymbol{y} + r)^n$, cu $r \geq 0$ și $n \geq 1$.
Kernel Gaussian (Radial Basis Function - RBF): $K(\boldsymbol{x}, \boldsymbol{y}) = e^{-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{2\sigma^2}}$, cu $\sigma > 0$.

3 Support Vector Machine

Metoda Support Vector Machine presupune găsirea hiperplanului ce separă 2 clase maximizând distanța de la fiecare punct la hiperplan. Din acest motiv, soluția se va numi hiperplanul de margine maximă și este definit astfel

$$\boldsymbol{w}^\top \boldsymbol{Y} - b = 0 \tag{2}$$

unde $\boldsymbol{w} \in \mathbb{R}^m$ reprezintă normala la hiperplan, iar $b \in \mathbb{R}$.

Maximizarea marginii dintre hiperplan și semnale se realizează minimizând $\|\boldsymbol{w}\|$. Estimarea etichetei unui semnal se face evaluând plasarea acestuia față de hiperplan, deci a valorii $sign(\boldsymbol{w}^\top \boldsymbol{Y} + b)$.

Figura 5 ilustrează alegerea hiperplanului.

SVM poate fi folosit și împreună cu trucul kernel. În acest caz, în locul semnalor $\boldsymbol{Y}$ se va folosi valoarea kernel-ului evaluată pentru fiecare semnal, $K(y)$.

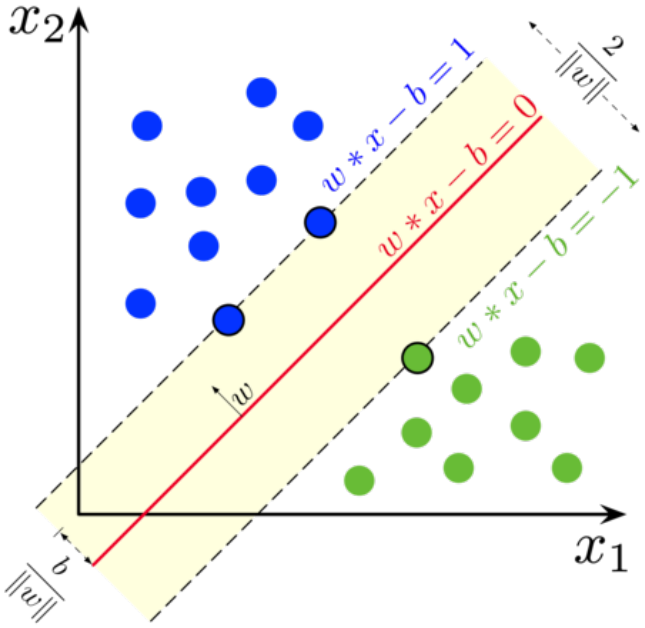

Figura 5: SVM - algerea hiperplanului.

4 Ghid Python

Pentru a rezolva exercițiile din laboratorul de astăzi, aveți nevoie de bibliotecile `numpy`, `random` și `pandas` și modulul `sklearn.svm`.

Pentru a amesteca în mod aleator elementele unui vector v, folosiți funcția `random.shuffle(v)`. Rezultatul se va suprascrie în v.

Pentru a selecta aleator un număr de k elemente dintr-un vector v, folosiți `random.sample(v, k)`.

Pentru a obține diferența dintre două mulțimi de numere, s_1 și s_2, utilizați `np.setdiff1d(s1,s2)`.

Pentru a antrena algoritmul SVM pe setul de date *data*, utilizați următoarele instrucțiuni

```
clf = svm.SVC()
clf.fit(data, labels)
```

Kernel-ul implicit este de tip RBF. Îl puteți schimba folosind sintaxa `svm.SVC(kernel=tip)`, unde *tip* poate lua valorile `'liniar'`, `'polynomial'`, `'rbf'`, `'sigmoid'`.

Puteți folosi rezultatele antrenării pentru a estima etichetele unui nou set de date folosind instrucțiunea `estimated_labels = clf.predict(new_data)`.

5 Exerciții

1. Utilizați algoritmul SVM pentru a clasifica semnalele din baza de date Iris.

   (a) Creați 2 seturi de date: unul de antrenare și unul de test. Setul de antrenare conține $N_{train} = 100$ semnale extrase în mod aleator din fiecare clasă de flori. Setul de test conține restul de $N_{test} = 50$ semnale.

   (b) Antrenați algoritmul SVM pe setul de antrenare folosind un kernel liniar.

   (c) Testați performanța algoritmului pe setul de test, calculând eroarea de clasificare și matricea de confuzie.

   (d) Refaceți experimentul utilizând de data asta un kernel de tip RBF.

Surse imagini

Figura 2:`https://www.ml-science.com/confusion-matrix`.
Figura 4: `https://towardsdatascience.com/the-kernel-trick-c98cdbcaeb3f`.
Figura 5: `https://en.wikipedia.org/wiki/File:SVM_margin.png`.