Support Vector Machines

Michele Di Nanni, mat. 7291871
Corso di Modellizzazione Statistica, prof. M. Bilancia

Introduzione

L'algoritmo **Support Vector Machines** è uno degli algoritmi più utili, efficienti ed importanti, appartenenti alla categoria degli algoritmi di *machine learning* supervisionati. L'ambito applicativo più frequente è quello dell' elaborazione del linguaggio naturale (*NLP, natural language processing*), del riconoscimento vocale, delle immagini e della *computer vision*, quest'ultima nota anche con il termine *visione artificiale*. Esso è utilizzato sia per scopi di classificazione, che di regressione. L'idea alla base è quella della presenza di una non separabilità dei dati in modo lineare, col conseguente obbiettivo di costruire un separatore, ovvero un iperpiano di separazione **ottimale**, che ci permetta di dividere al meglio i dati presenti nel *dataset* in classi. Si cerca per prima cosa un **iperpiano linearmente separabile** e qualora ve ne fossero più di uno, si andrebbe a cercare quello contenente il *margine* più alto, al fine di migliorare l'accuratezza del modello. Tuttavia, se tale iperpiano non esiste l'**SVM** utilizza una **mappatura non lineare** per trasformare i dati di *training* in uno spazio di dimensionalità maggiore(spazio delle variabili). In tal modo, i dati di due classi potranno essere sempre separati da un iperpiano, scelto per la suddivisione dei dati.

1.1 Iperpiano di separazione ottimale e separabilità lineare

Ciò che vogliamo trovare nelle **SVM** è quell' iperpiano, che nel caso di due classi sarà proprio una retta, che meglio classifica e, quindi, separa i nostri dati. Teoricamente, potremmo avere un numero infinito di rette (e quindi anche di iperpiani) che separino le istanze dei dati di training. L'obbiettivo è proprio trovare quella retta(iperpiano) che sia **ottimale**, generando il più piccolo errore di classificazione su dati di test. Ciò che vorremmo, pertanto, è che i nostri dati siano il più lontano possibile dalla retta(iperpiano), pur restando nella parte corretta, cioè quella di appartenenza a quella specifica classe.

Partendo dal caso in cui abbiamo *due classi* con etichette -1 e $+1$, consideriamo il campione $\chi = \{\mathbf{x}^t, \mathbf{r}^t\}$ dove:

1. $\mathbf{x}^t$ è l'insieme dei dati di training
2. $\mathbf{r}^t$ sono le etichette ottenute, ovvero gli output

$$r^t = \begin{cases} +1, & \text{se } \mathbf{x}^t \in C_1 \\ -1, & \text{se } \mathbf{x}^t \in C_2 \end{cases}$$

L'obbiettivo è quello di trovare $\mathbf{w}$ e w_0 tale che:

$$\mathbf{w}^T\mathbf{x}^t + w_0 \geq +1 \quad per \quad r^t = +1$$

$$\mathbf{w}^T\mathbf{x}^t + w_0 \leq -1 \quad per \quad r^t = -1$$

In altre parole, si vuole ricercare quella combinazione lineare per cui, se la etichetta di classe è $+1$, allora la combinazione dovrà essere maggiore o al più uguale a $+1$; viceversa, se l'etichetta è -1, allora occorrerà

trovare quella combinazione lineare che sia minore o al più uguale a -1. Potremmo comunque pensare di "fondere" queste ultime disequazioni, nella seguente:

$$r^t(\mathbf{w}^T\mathbf{x}^t + w_0) \geq +1$$

cioe, se abbiamo una risposta pari a $+1$ ci riferiremo ad una regione, altrimenti ci riferiremo al *complementare* della regione i cui campioni hanno l' etichetta $+1$. Il vettore $\mathbf{w}$ è il cosiddetto *vettore dei pesi* che avrà norma unitaria, cioè $||\mathbf{w}|| = 1$.

Diamo adesso delle definizioni, utili per comprendere al meglio il concetto di *iperpiano di separazione ottimale*:

Def. **Vettori di supporto**

I vettori di supporto sono gli esempi di training **più vicini** all'iperpiano. Questi punti dipendono dal dataset che analizziamo e, pertanto, qualora fossero rimossi o modificati, la posizione dell'iperpiano di divisione verrebbe alterata. A tal punto, possiamo dire che costituiscono *gli elementi critici* del dataset.

Def. **Margine**

Il margine è la definito come la distanza *minima* fra l'iperpiano di separazione e i vettori di supporto. È fondamentale chiarire che a metà di questa distanza viene tracciato l'iperpiano, o la retta nel caso in cui abbiamo due classi. La dimensione del margine massimo è:

$$\frac{1}{||\mathbf{w}||} + \frac{1}{||\mathbf{w}||} = \frac{2}{||\mathbf{w}||}$$

Def. **Iperpiano di separazione ottimale**

L' iperpiano di separazione ottimale è quell'iperpiano che massimizza il margine.

Pertanto, l'equazione $\mathbf{w}^T\mathbf{x} = 0$ definirà il limite di decisione, noto anche col termine **decision boundary**; mentre $\mathbf{w}^T\mathbf{x} = -1$ definisce l' *iperpiano negativo*, ovvero la regione "negativa" e l'equazione $\mathbf{w}^T\mathbf{x} = +1$ definisce l'*iperpiano positivo*, ovvero la regione "positiva". Osserviamo queste considerazioni appena fatte all'interno dell'immagine seguente:

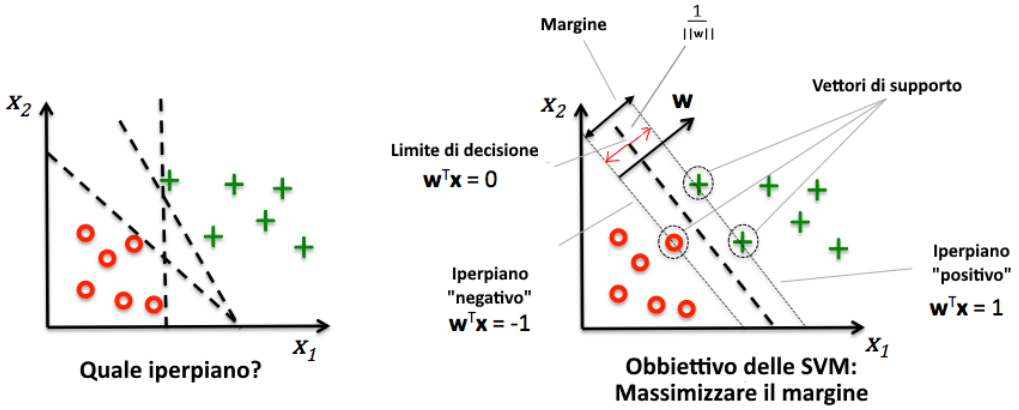

Figure 1: Support Vector Machines: idee di base

Come possiamo evincere dalla *Fig.*1, il concetto alla base delle **SVM** è, quindi, quello di trovare l'iperpiano ottimale che crei il margine più grande fra le istanze di training appartenenti alla classe -1 ed alla classe $+1$.

Il problema può essere ricondotto ad un problema di *ottimizzazione*, in cui vogliamo trovare:

$$\max_{\mathbf{w},w_0,\ ||\mathbf{w}||=1} \rho \qquad t.c. \quad r^t(\mathbf{w}^T\mathbf{x}^t + w_0) \geq \rho, \ \ \forall t$$

dove ρ indica il valore da massimizzare, cioè il margine. Tuttavia, poichè ci sono un numero infinito di soluzioni che possiamo ottenere, al fine di ottenere una soluzione unica, poniamo $\rho||\mathbf{w}|| = 1$ e perciò, per massimizzare il margine, occorre minimizzare $||\mathbf{w}||$, quindi il problema diventa:

$$\min_{\mathbf{w},w_0} \frac{1}{2}||\mathbf{w}||^2 \qquad t.c. \quad r^t(\mathbf{w}^T\mathbf{x}^t + w_0) \geq +1, \ \ \forall t$$

Nota: Abbiamo posto $\frac{1}{2}$ nella definizione del valore da minimizzare e la norma al quadrato, in quanto per poter calcolare tale minimo occorrerà il calcolo della derivata ed in tal modo, derivando appunto, quella frazione scomparirà perchè sarà moltiplicata per l'esponente della norma.

Dunque, il problema rientra nel contesto dei problemi di ottimizzazione quadratici, che possono essere risolti direttamente andando alla ricerca di $\mathbf{w}$ e w_0. Il parametro di complessità è d, ovvero il numero di *features* presenti nel dataset. Per poter trovare il miglior iperpiano, possiamo convertire il problema facendo in modo che il parametro di complessità sia N, ovvero il numero di *istanze di training.*

1.1.1 Risoluzione del problema di minimizzazione

Il problema può essere risolto con l'ausilio del metodo dei moltiplicatori di Lagrange, il quale è un metodo analitico che ci permette di trovare massimi e minimi vincolati.

Per questo poniamo:

$$\begin{aligned} L_p \quad &= \quad \frac{1}{2}||\mathbf{w}||^2 - \sum_{t=1}^{N} \alpha^t [r^t(\mathbf{w}^T\mathbf{x}^t + w_0) - 1] \\ &= \frac{1}{2}||\mathbf{w}||^2 - \sum_{t=1}^{N} \alpha^t r^t(\mathbf{w}^T\mathbf{x}^t + w_0) + \sum_t \alpha^t \end{aligned}$$

Dove la parte relativa alla sommatoria è il vincolo a cui sottoponiamo il nostro problema. La soluzione è data dalla minimizzazione di $\mathbf{w}$ e di w_0 e dalla massimizzazione di $\alpha^t \geq 0$, il che corrisponde al trovare il *punto di sella.*

Svolgendo i prodotti e riscrivendo la norma otteniamo:

$$= \frac{1}{2}(\mathbf{w}^T\mathbf{w}) - \mathbf{w}^T \sum_t \alpha^t r^t \mathbf{x}^t - w_0 \sum_t \alpha^t r^t + \sum_t \alpha^t \qquad (*)$$

Per poter procedere, occorre sia calcolare le derivate rispetto a $\mathbf{w}$ e w_0, ponendole uguali a zero, sia considerare che $\alpha^t \geq 0$. Ovvero:

$$\frac{\partial L_p}{\partial \mathbf{w}} = 0 \qquad \Rightarrow \qquad \mathbf{w} = \sum_t \alpha^t r^t \mathbf{x}^t \qquad (1)$$

$$\frac{\partial L_p}{\partial w_0} = 0 \qquad \Rightarrow \qquad \sum_t \alpha^t r^t = 0 \qquad (2)$$

Da (1) e da (2) abbiamo ottenuto due risultati fondamentali; procediamo quindi a fare le opportune sostituzioni all'equazione contrassegnata con $(*)$, otteniamo:

$$L_d = -\frac{1}{2}(\mathbf{w}^T\mathbf{w}) + \sum_t \alpha^t$$

$$= -\frac{1}{2}\sum_t \sum_s \alpha^t \alpha^s r^t r^s (\mathbf{x}^t)^T \mathbf{x}^s + \sum_t \alpha^t$$

che vogliamo massimizzare rispetto ad α^t, coi vincoli che $\sum_t \alpha^t r^t = 0$ e che $\alpha^t \geq 0$, $\forall t$. La risoluzione è data da metodi di ottimizzazione quadratica. La dimensione dipende da N, ovvero dalla dimensione del campione, e non da d, rispettivamente la *dimensionalità* dell'input. La soluzione deve soddisfare le condizioni di Karush-Kuhn-Tucker (vedi appendice), che includono (1), (2), $\alpha^t \geq 0$ e:

$$\alpha^t [r^t (\mathbf{w}^T \mathbf{x}^t + w_0) - 1] = 0 \quad \forall t$$

Possiamo osservare che:

- Se $\alpha^t = 0$, allora $\mathbf{x}^t$ non è sul confine del margine, tuttavia
- Se $\alpha^t > 0$, allora $r^t(\mathbf{w}^T \mathbf{x}^t + w_0) = 1$, ovvero, $\mathbf{x}^t$ è sul confine del margine

Per questo motivo, i vettori $\mathbf{x}^t$ tali che $\alpha^t > 0$ sono proprio i **vettori di supporto**. Il discriminante, cioè la retta(o l'iperpiano) trovato è chiamato *macchina a vettore di supporto*(nota con l'acronimo **SVM**).

Una **SVM** tiene conto delle istanze che sono vicine al limite(*boundary*), scartando quelle che si trovano all'interno. Usando questa idea di classificazione, possiamo pensare di utilizzare un classificatore più semplice prima di far lavorare la **SVM**, al fine di filtrare una grande parte di tale istanze, facendo decrescere, perciò, la complessità computazionale nella fase di ottimizzazione della **SVM**.

Durante la fase di *testing*, andremo a calcolare $g(x) = \mathbf{w}^T \mathbf{x} + w_0$ e sceglieremo in base al segno di $g(x)$: se $g(x) > 0$ allora sceglieremo la classe $C1$, altrimenti la classe $C2$, cioe:

$$sign(g(x)) = sign(\mathbf{w}^T \mathbf{x} + w_0) = \begin{cases} +1 & \rightarrow C_1 \\ -1 & \rightarrow \mathbf{C}_2 \end{cases}$$

Riassumiamo queste considerazioni fatte finora, nel caso in cui ci troviamo ad operare con dataset di due classi, dove abbiamo la separabilità lineare, nella $Fig.2$:

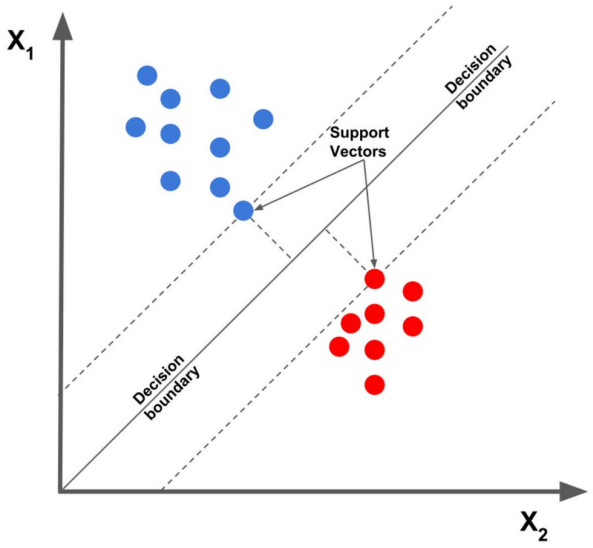

Figure 2: Problema a due classi con separabilità lineare

1.2 Non separabilità lineare

Finora abbiamo considerato il caso in cui vi fosse una separabilità lineare fra i dati, tuttavia, nella maggior parte delle applicazioni reali questo non accade e quindi abbiamo bisogno di trovare una alternativa che ci permetta di ricondurci al caso lineare.

Se le due classi non sono, dunque, *linearmente separabili*, il che vuol dire che non esiste alcun iperpiano che ci permetta di dividere i dati, andremo alla ricerca di un certo iperpiano che presenti l'errore minimo di errata classificazione. In questo caso, possiamo ammettere che alcuni vincoli enunciati precedentemente siano violati e abbiamo necessità di definire le cosiddette "variabili deboli", che denoteremo da ora in poi come variabili **slack**, $\xi = (\xi^1, \xi^2, ..., \xi^t)$ $t.c.$ $\xi^t \geq 0$, che consentono la classificazione errata di qualche punto e che codificano la deviazione del margine. Pertanto, il vincolo diventa:

$$(\mathbf{w}^T\mathbf{x}^t + w_0) \geq 1 - \xi^t \quad se \quad r^t = +1, \quad \forall t$$

$$(\mathbf{w}^T\mathbf{x}^t + w_0) \leq 1 + \xi^t \quad se \quad r^t = -1 \quad \forall t$$

Che possono essere riassunti in:

$$r^t(\mathbf{w}^T\mathbf{x}^t + w_0) \geq 1 - \xi^t \quad \forall t$$

Ci troviamo dinnanzi a due differenti tipi di deviazione del margine: una istanza può trovarsi sul lato sbagliato ed essere non classificata correttamente; oppure, può essere nel lato corretto ma trovarsi sul margine, cioè, non sufficientemente lontana dall'iperpiano. Pertanto, se $\xi^t = 0$ allora avremo corretta classificazione; invece se $0 < \xi^t < 1$, l'istanza $\mathbf{x}^t$ è classificata correttamente ma nella zona del margine; se $\xi^t > 1$, l'istanza $\mathbf{x}^t$ non è classificata correttamente. Osserviamo queste considerazioni nella $Fig.3$

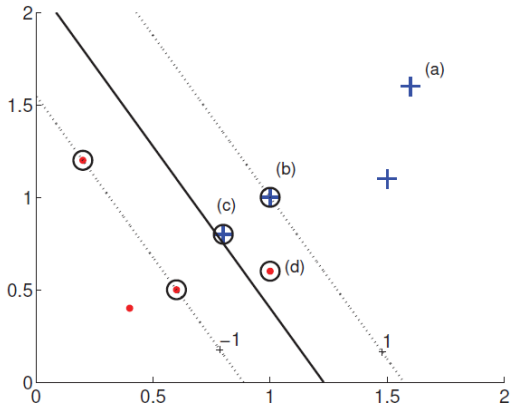

Figure 3: Non linearità

Possiamo notare di come l'istanza (a) sia classificata *correttamente*, per questo motivo $\xi^t = 0$, ovvero $r^t g(\mathbf{x}^t) > 1$ e dunque molto lontana dal margine. L'istanza (b) si trova nella zona corretta, quindi $\xi^t = 0$, ma è sul margine, mentre l'istanza (c) si trova nel lato corretto ma è all'interno del margine, dunque non sufficientemente lontana da esso $(0 < \xi^t < 1)$. Infine, l'istanza (d) è classificata in modo erroneo, pertanto $\xi^t > 1$.

A questo punto definiamo il *numero di classificazioni errate* come $\#\{\xi^t \geq 1\}$ e il ***soft error*** come la somma delle variabili slack su dati di training, cioè $\sum_t \xi^t$.

Occorre modificare la funzione di costo, in modo da penalizzare le variabili slack che non sono a 0, introducendo una costante positiva C che misura il *trade-off* tra la complessità dello spazio delle ipotesi e il numero di esempi non-separabili: in pratica è una misura degli errori commessi, la quale misura non solo i punti mal

classificati(tali che $\xi^t \geq 1$), ma anche quelli presenti sul margine($0 < \xi^t < 1$) al fine di ottenere una migliore generalizzazione. Quindi avremo che:

$$L_p = \frac{1}{2}||\mathbf{w}||^2 + C\sum_t \xi^t \quad t.c. \quad \xi^t \geq 0,\ r^t(\mathbf{w}^T\mathbf{x}^t + w_0) \geq 1 - \xi^t \ \ \forall t$$

L'obbiettivo è calcolare il minimo di L_p, ovvero occorre calcolare:

$$\min L_p = \min_{\mathbf{w},w_0} \frac{1}{2}||\mathbf{w}||^2 + C\sum_t \xi^t \quad t.c. \quad \xi^t \geq 0,\ r^t(\mathbf{w}^T\mathbf{x}^t + w_0) \geq 1 - \xi^t \ \ \forall t$$

1.2.1 Risoluzione del problema di minimizzazione

A questo punto, occorre risolvere il problema di minimo, allo stesso modo di come è stato operato nella sezione 1.1.1

Appendice

Kuhn-Tucker conditions