MAXIMAL MARGIN CLASSIFIER

Il mio lavoro è incentrato sullo studio del metodo SVM, un metodo di classificazione binaria spesso utilizzato come migliore opzione “out-of-the-box” data la sua versatilità.

Nella sua formulazione più semplice, il MAXIMAL MARGIN CLASSIFIER, la classificazione è ottenuta dividendo lo spazio in due attraverso un IPERPIANO, un insieme di punti che rispettano l’equazione , dove p è il numero di dimensioni, ottenendo così un sottospazio di punti per i quali l’equazione assume valori maggiore di zero ed uno per i quali il valore è minore di 0. Se un data point è correttamente classificato, la distanza dall’iperpiano e la class label avranno lo stesso segno, quindi il loro prodotto sarà maggiore di 0. Inoltre, tale quantità è proporzionale alla fiducia che si ha nella classificazione prodotta.

Siccome due scatter di punti osservati possono essere separati da infiniti diversi iperpiani, ottenibili da rotazioni o spostamenti infinitesimali di questa regola decisionale, bisogna selezionarne uno solo: l’iperpiano considerato come il migliore in questo metodo è quello che presenta la maggiore distanza dai punti di entrambe le classi, detto IPERPIANO SEPARATORE OTTIMO con il relativo MARGINE, ovvero la distanza dai data points più vicini delle due classi. Alcuni punti, detti SUPPORT VECTORS, si troveranno esattamente sul margine e saranno gli unici ad influenzarlo. Ciò significa che eventuali outliers per una o più delle direzioni non andranno ad influenzare la regola decisionale dal momento che questa è collocata nella regione interna della nuvola di punti.

SUPPORT VECTOR CLASSIFIER

Per come è individuato il margine, quanto detto finora vale solo in presenza di classi perfettamente separabili da una decision boundary lineare, situazione raramente osservata nella realtà che andrebbe a scoraggiare l’utilizzo di questo metodo. Allontanarsi dalla ricerca di un iperpiano che separa perfettamente le classi permette di conferire al modello una maggiore robustezza dal momento che la possibilità di avere poche osservazioni sul lato sbagliato dell’iperpiano viene controbilanciata dalla capacità di classificare le rimanenti osservazioni di training anche quando le classi non sono linearmente separabili.

Per questo motivo si ricorre solitamente ad una diversa formulazione, detta SUPPORT VECTOR CLASSIFIER, basata su un SOFT-MARGIN, appunto un margine meno stringente che può essere violato da alcune osservazioni. La flessibilità o rigidità attribuita al margine dipende da un iperparametro C, una sorta di budget da spendere per permettere ai data points di violare il margine o anche l’iperpiano. Tale budget viene distribuito tra delle variabili dette di SLACK che rappresentano quindi la quantità di budget C spesa da ogni singola osservazione per poter violare la regola decisionale.

Come è possibile vedere in figura, il valore C controlla l’ampiezza del margine che è direttamente collegata alla quantità di errori che si vanno a commettere: valori grandi di C portano il margine ad essere molto ampio e quindi ad ammettere molti errori; più il valore di C si riduce, più l’ampiezza del margine diminuisce e minore è il numero di errori di classificazione ammessi. Trattandosi di un iperparametro, non esiste un valore “corretto” e sempre applicabile per C: il valore più adatto ai dati in questione deve essere individuato mediante CROSS VALIDATION.

SUPPORT VECTOR MACHINES & KERNEL FUNCTIONS

Nonostante il soft-margin del support vector classifier sia molto versatile, potrebbe generare un eccessivo numero di errori di classificazione quando i dati sono separabili esclusivamente da decision boundaries fortemente non lineari, non approssimabili da un semplice iperpiano. Il vero problema qui è che la dimensione p dello spazio in cui ci si trova potrebbe essere troppo piccola per evidenziare il confine che separa le classi.

L’idea per far fronte a questo problema, su cui è basato e che caratterizza il metodo SVM, consiste nel proiettare i dati in uno spazio di dimensione maggiore, cercare l’iperpiano separatore ottimo in tale spazio e riportare i dati nello spazio di partenza in cui questa separazione potrà essere arbitrariamente non lineare.

Questa trasformazione è condotta mediante l’utilizzo di funzioni KERNEL e l’intera procedura è detta appunto KERNEL-TRICK. Un kernel è una funzione tale che presi in input due vettori, ne calcola una trasformazione che li porta in uno spazio di dimensione diversa, ne calcola il prodotto interno e produce in output uno scalare proporzionale alla similarità tra i due. Facendo questi calcoli “on-the-fly”, non c’è la necessità di salvare in memoria i dati trasformati e questo rende il metodo particolarmente conveniente.

Le funzioni kernel più utilizzate nel metodo SVM sono il kernel lineare, il kernel polinomiale ed il kernel RBF.

Il KERNEL LINEARE è semplicemente pari al prodotto interno tra i due vettori ed il suo punto di forza è la capacità di considerare come simili osservazioni molto distanti in termini di distanza euclidea dal momento che si dimostra che tale quantità è proporzionale alla cosine similarity, misura di similarità che considera simili vettori che formano angoli piccoli.

Si parla effettivamente di SUPPORT VECTOR MACHINE quando il support vector classifier è usato assieme ad una funzione kernel non lineare come i kernel polinomiale o RBF.

Il kernel POLINOMIALE dipende da un iperparametro d che controlla effettivamente la complessità della decision boundary prodotta: d=1 non è altro che il kernel lineare mentre con d>1 i dati vengono proiettati in uno spazio di dimensione maggiore utilizzando dei polinomi di grado d. Essendo legato alla complessità del modello, bisogna fare attenzione al valore utilizzato dal momento che un modello troppo complesso potrebbe generare overfitting e perdere capacità di generalizzazione.

Per quanto riguarda, il kernel RADIAL BASIS FUNCTION invece, guardando la sua espressione è possibile individuare la sua relazione con la distanza euclidea tra le due osservazioni, ed il fatto che presenta una forma “bell-shaped”, il che le conferisce anche il nome di gaussian kernel. L’iperparametro è una costante positiva che determina una maggiore non linearità della decision boundary al crescere del suo valore, ed è inversamente proporzionale all’ampiezza della curva a campana: se le osservazioni sono vicine, la loro distanza euclidea sarà piccola e di conseguenza sarà grande, il che indica elevata similarità; viceversa, se le osservazioni sono lontane, la distanza sarà grande e quindi sarà piccolo, indicando quindi bassa similarità. La forma di questa funzione kernel le permette di assumere un comportamento molto locale ad ogni osservazione per determinare la sua classe di appartenenza, andando a considerare solo i data points più vicini e generando così delle decision boundaries fortemente non lineari.

DATI e RISULTATI

L’applicazione del classificatore SVM che ho condotto consiste nel confronto delle performance dei tre kernel appena descritti utilizzando dati riguardanti la previsione dello stato di bancarotta di imprese polacche. L’analisi esplorativa del dataset ha rivelato alcune particolarità che ho dovuto poi gestire per poter ottenere risultati utilizzabili.

In primo luogo, la variabile di risposta, che indica se un’impresa ha dichiarato bancarotta nell’anno successivo a quello in cui sono stati registrati i valori delle features, è un variabile binaria fortemente sbilanciata. Lo sbilanciamento tra classi è un problema che può invalidare completamente i risultati di una procedura di learning. Per questo motivo, ho condotto ricampionamento usando il metodo SMOTE che prevede la creazione di data points artificiali per la minority class, in modo da bilanciare la rappresentazione delle classi.

Per quanto riguarda i predittori invece, ognuna delle 64 variabili del dataset è calcolata come combinazione aritmetica di una serie di misure economiche. Siccome alcune features condividono le stesse misure, i dati presentano una forte struttura di correlazione: data anche l’elevata dimensionalità dei dati, sono andato a creare una versione ridotta del dataset mediante la riduzione basata sulle componenti principali, eliminando la maggior parte della ridondanza dovuta alla correlazione tra i predittori.

Dopo questi step di pre-processing, ho condotto l’addestramento di diversi modelli SVM, basati su diverse specificazioni degli iperparametri delle varie funzioni kernel. Siccome le classi sono sbilanciate ed il metodo SMOTE è implementato nella fase di training del modello, ho preferito usare la curva ROC e la sua area AUC come metriche di performance dal momento che queste sono più consone al caso delle classi sbilanciate rispetto ad esempio all’indice di accuratezza ACC che va a dare lo stesso peso agli errori di primo e secondo tipo.

Come è possibile vedere dal grafico, i risultati non sono sorprendenti: il kernel RBF produce il modello con AUC maggiore, seguito dalle tre specificazioni di kernel polinomiale ed infine da quello lineare. Inoltre, le performance di quasi tutte le specificazioni sembrano declinare leggermente all’aumentare del parametro di costo C (il che è ragionevole dal momento che si consentono più errori).

Ancora, guardando la forma della vera e propria curva ROC si vede come quella del kernel RBF sia la più vicina alla perfetta classificazione, producendo l’area più grande.