Relazione Intelligenza Artificiale

Heidi Garcia Canizares

17 giugno 2019

1 Introduzione

L'algoritmo **Support Vector Machine** (SVM) è uno degli strumenti più utilizzati per la classificazione di pattern. Invece di stimare le densità di probabilità delle classi risolve direttamente il problema di interesse (che considera più semplice), ovvero determinare le superfici decisionali tra le classi (classification boundaries). Date due classi di pattern multidimensionali linearmente separabili, tra tutti i possibili iperpiani di separazione, SVM determina quello in grado di separare le classi con il maggior margine possibile. Il margine è la distanza minima di punti delle due classi nel training set dall'iperpiano individuato.

Con SVM, però, è posssibile applicare il *kernel trick* nel caso in cui abbiamo dei dati non linearmente separabili. Questa tecnica consiste nel mappare il set di dati non linearmente separabili in uno spazio dimensionalmente superiore in cui possiamo trovare un iperpiano in grado di separare i campioni. In questo esercizio verrà utilizzato il kernel **"Radial Basics Function (RBF)"** definito dalla formula (1):

$$K(x_i, x_j) = e^{-\gamma(x_i - x_j)^2} \tag{1}$$

2 One-class SVM

Molte applicazioni richiedono di decidere se una nuova osservazione appartiene alla stessa distribuzione delle osservazioni esistenti, oppure se sia diversa (è un valore anomalo). Spesso questa capacità viene utilizzata per pulire set di dati reali. Nell'**Outlier detection** i dati di training contengono valori anomali definiti come osservazioni lontane dalle altre. I rilevatori di outlier cercano quindi di adattarsi alle regioni in cui i dati di training sono più concentrati, ignorando le osservazioni devianti.

In questo esercizio tratteremo l'applicazione dell'algoritmo **One-Class SVM** su due problemi di outlier detection scelti tra quelli presenti sul sito http://odds.cs.stonybrook.edu

3 Package impiegati

Pandas: Libreria per il caricamento e salvataggio di formati standard per dati tabellari, quali CSV (Comma-separated Values), usato nel presente esercizio.
Matplotlib: Libreria per la realizzazione di grafici estremamente potente e flessibile.
Seaborn è una libreria per la visualizzazione di dati basata su matplotlib. Fornisce un'interfaccia di alto livello per disegnare grafici statistici attraenti e informativi.
Scikit-learn: Libreria open source di apprendimento automatico per il linguaggio di programmazione Python. Contiene gli algoritmi One-Class SVM e SVC.
NumPy: Estensione open source del linguaggio di programmazione Python, che aggiunge supporto per vettori, matrici multidimensionali e di grandi dimensioni e con funzioni matematiche di alto livello con cui operare.

4 Utilizzo dei dati

I dati sono divisi in due: X è una matrice di dati multidimensionali e y è un vettore di labels (**1 = outliers, 0 = inliers**). A questi dati viene applicato l'algortimo **sklearn.svm.OneClassSVM**. Questo algoritmo **non supervisionato** di outlier detection permette di stimare il supporto di una distribuzione ad alta dimensione. L'implementazione è basata su libsvm.

Gli algoritmi di anomaly detection basati sull'apprendimento non supervisionato imparano cosa è normale e quindi applicano un test statistico per determinare se un dato specifico è un'anomalia. Molte volte si preferisce

questo approccio perché gli algortimi basati sull'apprendimento supervisionato lavorano male su dataset molto sbilanciati.

4.1 Satellite Dataset

Il dataset **"Satellite"** presenta 2036 dati outlier contro i 4399 dati inlier. Prima di lavorare con i dati, essi sono stati normalizzati usando la funzione *scale()*.

Si è cercato di **stabilizzare l'errore di validazione** entro un intervallo di valori di gamma: (0.0000015,0.00015). L'errore minimo è di 0.3595959595959596, associato a un valore di gamma intorno allo 0.00005. Ciò può essere osservato in Figura 1.

A tal proposito si è applicato anche l'algortimo **SVC** per confrontare le performance tra un algoritmo di apprendimento non supervisionato, come può essere OneClass SVM, e uno supervisionato come SVC. In questo caso i dati vengono divisi in dati di train e di test. In questo caso l'errore di validazione si stabilizza nell'intervallo del parametro gamma: (0, 0.0005). Con l'SVM diminusce l'errore minimo a 0.07174556213017746 associato al valore di gamma intorno allo 0.00045. L'accuratezza, nel caso del OneClassSVM, è del 64%, mentre per l'SVC sale all'85%.

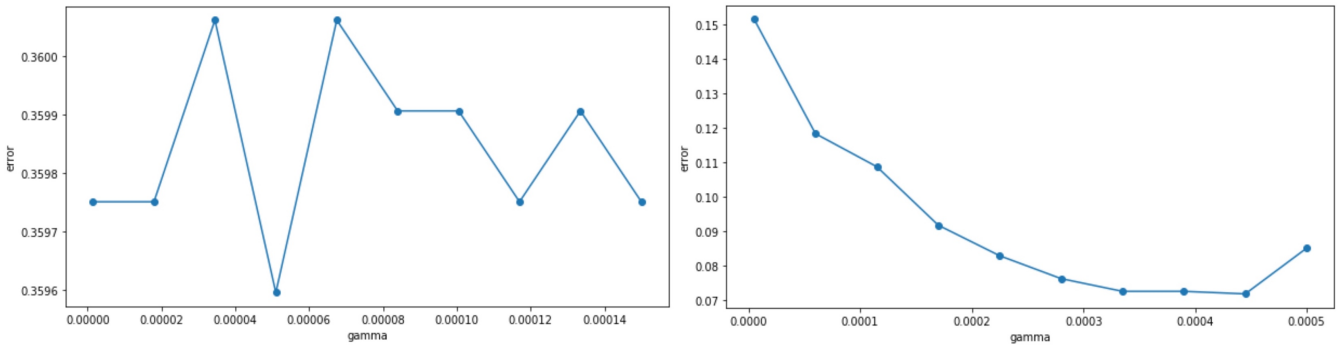

Figure 1: Parametro gamma OneClassSVM vs SVC (C=1)

Un altro parametro importante preso in considerazione nel caso dell'algortimo supervisionato è stato *C*. Il parametro *C* indica all'ottimizzazione SVM quanto si vuole evitare di classificare erroneamente ogni esempio di allenamento. È stata cercata la tupla ottima (γ, C) alla quale fosse associata l'errore minore: a (0.001, 5.7) corrisponde un errore di 0.069. Ciò, insieme ad altri valori per i parametri gamma e C, si osservono chiaramente nel grafico 3D in Figura 2.

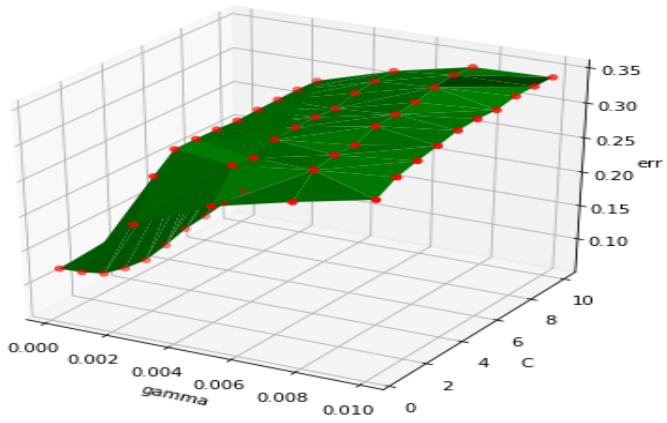

Figure 2: Grafico gamma, C ed errore relativo all'SVC

La differenza tra le matrici di confusione relative ai due algoritmi può essere osservata in Figura 3.

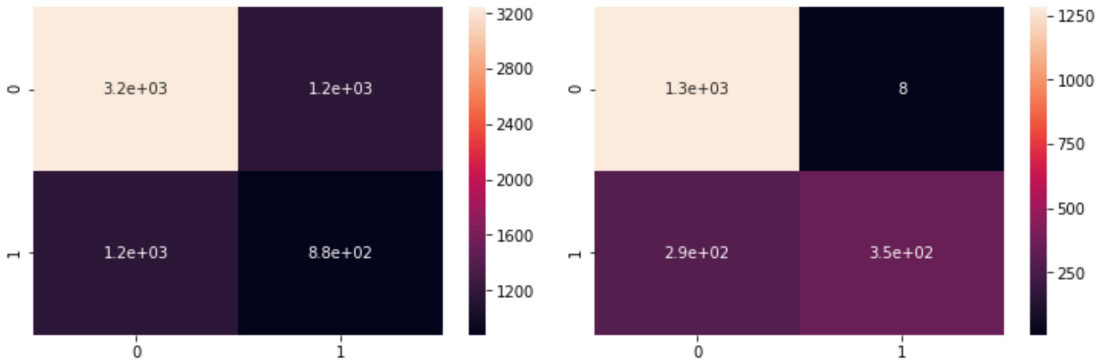

Figure 3: Matrici di confusione OneClassSVM vs SVC

È stato usato l'algortimo t-SNE per ridurre la dimensione dei dati e poterli visualizzare in un semplice grafico 2D. **t-Distributed Stochastic Neighbor Embedding (t-SNE)** è una tecnica non lineare e non controllata, utilizzata principalmente per l'esplorazione e la visualizzazione di dati ad alta dimensione. In termini più semplici, t-SNE permette di visualizzare come i dati siano disposti in uno spazio ad alte dimensioni. In Figura 4 e 5 si può vedere la differenza tra entrambi gli algortimi usati in questo esercizio. Il primo grafico corrisponde ai valori reali, mentre il secondo corrisponde ai valori predetti.

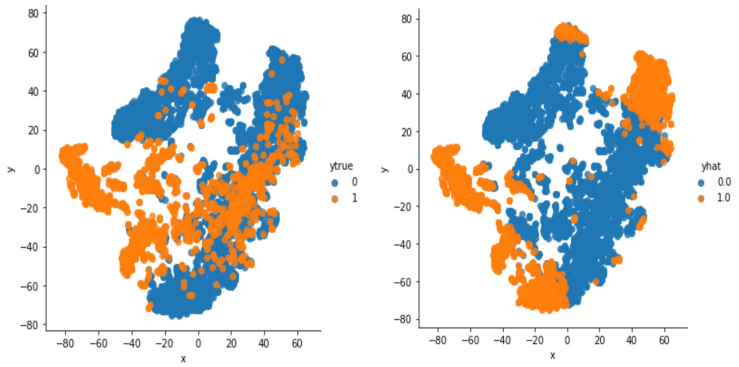

Figure 4: Disposizione dei dati secondo il OneClassSVM

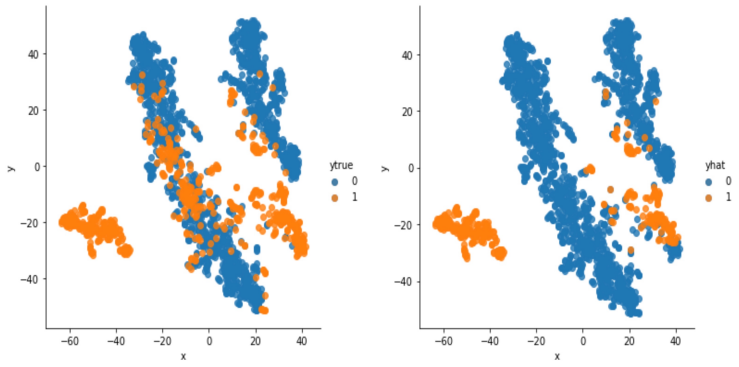

Figure 5: Disposizione dei dati secondo l'SVC

4.2 Shuttle Dataset

Il dataset **"Shuttle"**, invece, presenta 3511 outlier contro i 45586 inlieri. Essi vengono normalizzati usando sempre la funzione *scale()*. Anche su questo dataset si è cercato di **stabilizzare l'errore di validazione** entro un intervallo di valori di gamma: (0.0000015;0.00015). L'errore di validazione minimo in questo caso è di 0.05786504267063164, associato a un valore di gamma intorno allo 0.00005. Ciò può essere osservato in Figura 6.

A tal proposito si è applicato anche l'algortimo **SVC** per confrontare le performance tra un algoritmo di apprendimento non supervisionato e uno supervisionato. In questo caso l'errore di validazione si stabilizza nell'intervallo di gamma (0; 0.0005). Con l'SVM diminusce l'errore minimo a 0.004073319755600768 associato ai valori di gamma nell'intervallo (0.00005, 0.005). L'accuratezza, nel caso del OneClassSVM, è del 94%, mentre per l'SVC sale al 99%.

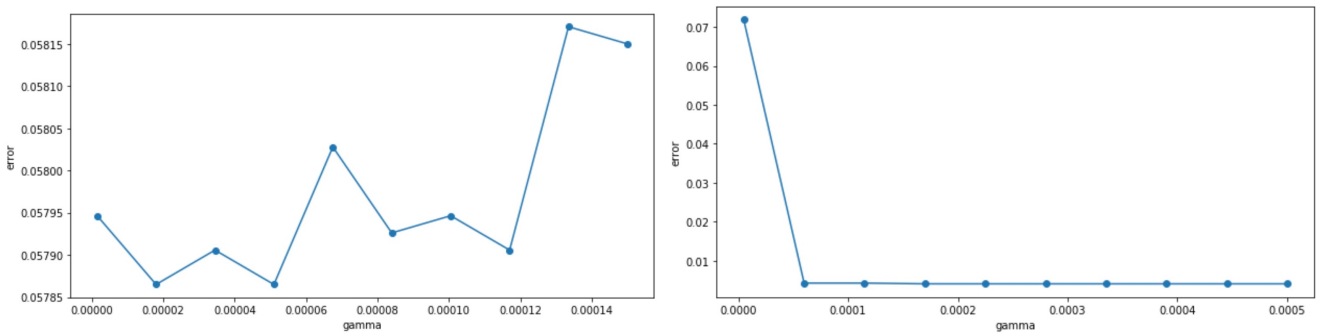

Figure 6: Parametro gamma OneClassSVM vs SVC (C=1)

Anche qua è stato preso in considerazione il parametro *C*. È stata cercata la tupla ottima (γ, C) alla quale fosse associata l'errore minore: a (0.001, 5.7) corrisponde un errore di 0.069. Si riporta il grafico 3D in Figura 7.

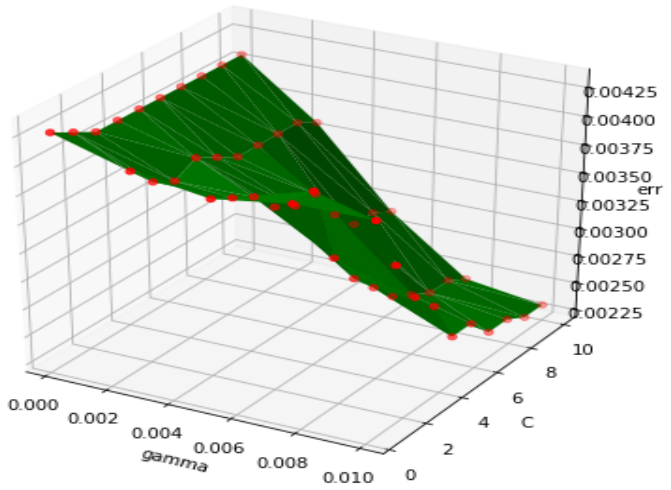

Figure 7: Grafico gamma, C ed errore relativo all'SVC

La differenza tra le matrici di confusione relative ai due algoritmi può essere osservata in Figura 8.

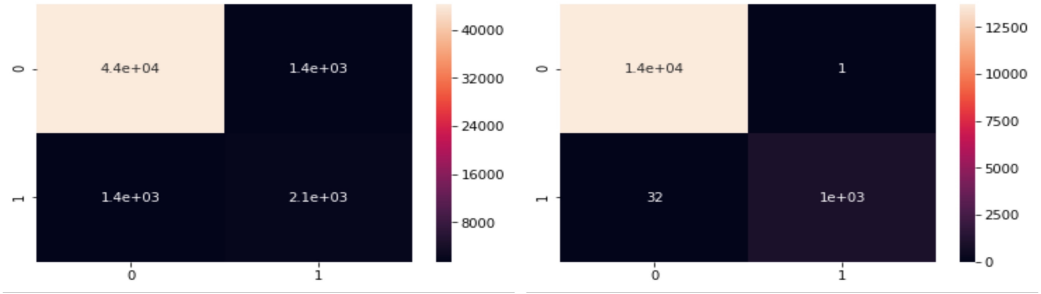

Figure 8: Matrici di confusione OneClassSVM vs SVC

Anche in questo caso è stato usato l'algortimo t-SNE per ridurre la dimensione dei dati e poterli visualizzare in un semplice grafico a 2D. In Figura 9 e 10 si osserva la differenza tra entrambi gli algortimi usati in questo esercizio. Il primo grafico corrisponde ai valori reali, mentre il secondo corrisponde ai valori predetti.

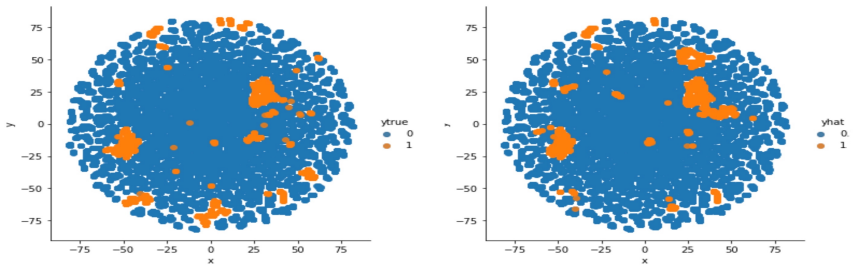

Figure 9: Disposizione dei dati secondo il OneClassSVM

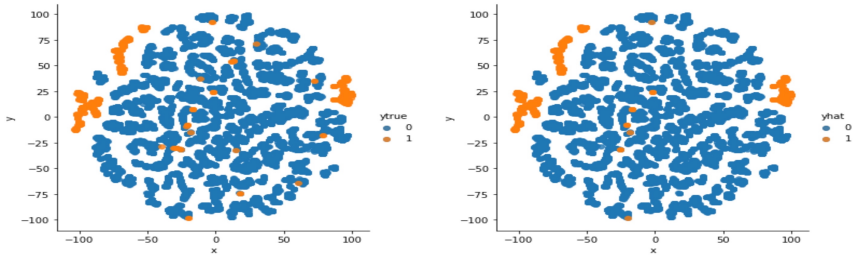

Figure 10: Disposizione dei dati secondo l'SVC

5 Conclusioni

Nei due dataset trattati, l'algoritmo supervisionato lavora meglio rispetto a quello non supervisionato. Di solito, però, un algoritmo di apprendimento supervisionato non da ottimi risultati di fronte ad anomalie "nuove", cioè, le tipologie di outlier che non ha incontrato durante l'apprendimento. Quindi si preferisce un margine di accuratezza più basso (come può essere il 94% nel caso dello Shuttle) con la garanzia di avere un algortimo funzionante anche di fronte ad outlier sconosciuti.