

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA
SEDE DI CESENA

Scuola di Ingegneria ed Architettura
Corso di Laurea in Ingegneria Elettronica e Telecomunicazioni per
l'Energia

ALGORITMI DI MACHINE LEARNING PER IL RICONOSCIMENTO VOCALE

Elaborato in
Elaborazione Numerica dei Segnali LM

Tesi di Laurea di:
ELIA FAVARELLI

Relatore:
Prof. Ing.
ANDREA GIORGETTI
Correlatore:
Chiar.mo Prof. Ing.
MARCO CHIANI

SESSIONE II
ANNO ACCADEMICO 2016–2017

PAROLE CHIAVE

Machine Learning

Voice Recognition

Perceptron

Logistic Regression

MATLAB

Indice

Introduzione	1
1 Algoritmi di Machine Learning	3
1.1 Algoritmi di Classificazione	3
1.2 Classificazione Multiclasse	5
1.2.1 One-versus-the-rest classifier	5
1.2.2 One-versus-one classifier	6
1.2.3 Multi-discriminant function	6
2 Conclusioni	7
Elenco Figure	9
Bibliografia	11

Introduzione

L'obiettivo di questo elaborato è quello di implementare diversi algoritmi che sfruttino alcune fra le più innovative tecniche di classificazione mediante l'approccio del *Machine Learning*.

Per *Machine Learning* si intende una particolare classe di algoritmi in grado di “apprendere” da un set di dati di *training* e successivamente di effettuare delle predizioni su un nuovo set di dati [1]. In questa categoria ricadono un gran numero di algoritmi fra cui anche il *Perceptron* e la *Logistic Regression* che verranno ampiamente discussi nel corso dei capitoli successivi.

Nella prima parte dell'elaborato verranno indagate le caratteristiche e la struttura di questo tipo di algoritmi, ponendo particolare attenzione ai meccanismi di *learning* che consentono, sotto alcune condizioni, di effettuare una buona classificazione delle misure successivamente sottoposte all'algoritmo.

Si procederà dunque con un confronto fra i 2 algoritmi sopracitati, tenendo conto di alcune *features* di interesse quali il tempo di elaborazione e l'accuratezza della classificazione, che può essere stimata con strumenti classici della *teoria della stima* come ad esempio la *matrice di confusione*.

Per concludere si calerà il tutto in un problema reale ovvero la distinzione di 2 voci. L'obiettivo ultimo sarà dunque quello di riconoscere e distinguere in tempo reale 2 interlocutori che parlano fra loro, a valle di una preventiva fase di *learning* in cui gli interlocutori comunicano separatamente e dichiaratamente.

Capitolo 1

Algoritmi di Machine Learning

Gli *Algoritmi di Machine Learning* trovano interessanti applicazioni in svariati campi e consentono di realizzare classificatori (di cui parleremo in questo elaborato) ma anche stimatori, riconoscitori di pattern e predittori.

Il fattore comune a tutti gli algoritmi di *learning* è proprio la fase di apprendimento.

Durante questa fase l'algoritmo conosce la classe di appartenenza di una misura, o il pattern che deve riconoscere, o il parametro che deve stimare e setta alcuni parametri al fine di minimizzare l'errore commesso durante la stima.

Al termine di questa fase l'algoritmo non conosce più il parametro a priori e cerca di stimarlo al meglio sulla base dell'esperienza acquisita durante la fase di training.

1.1 Algoritmi di Classificazione

Concentriamoci quindi sul sottogruppo di *algoritmi di machine learning* utilizzati per effettuare la classificazione, il cui obiettivo è quello di assegnare una realizzazione alla classe ad essa più affine e più in particolare di algoritmi di classificazione *lineari*, in cui le separazioni fra le varie classi sono di tipo lineare (figura 1.1).

Gli algoritmi di classificazione lineare sono molteplici e si è deciso di concentrare l'analisi su due di essi in particolare:

- Perceptron;
- Logistic Regression.

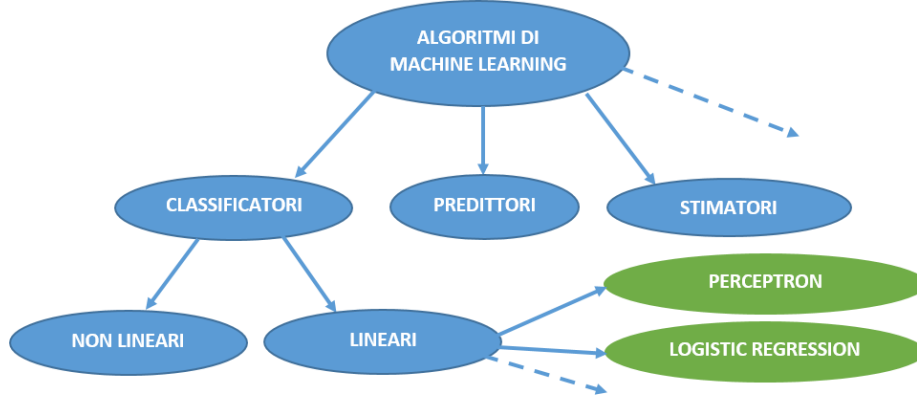


Figura 1.1: Tassonomia degli algoritmi di Machine Learning

Definiamo di seguito alcuni parametri, vettori e matrici che saranno utili durante la trattazione degli algoritmi:

- x:** Vettore contenente le *features* estratte dalla misura effettuata (ad esempio energia, ritardo, fase, ampiezza massima, ecc.). Il vettore risulterà essere m-dimensionale, con una dimensione per ogni *feature* e costituirà un punto di training (o analogamente una realizzazione);
- X:** Matrice contenente tutti i punti di training, sarà di dimensioni $N \times M$ con N numero di punti di training;
- t:** Vettore target contenente le associazioni fra i punti di training e la classe di appartenenza, fornito all'algoritmo durante la fase di learning. Il vettore conterrà tanti elementi quante sono le misure utilizzate per la fase di *learning* (m-dimensionale);
- p:** Elemento contenente la classe di appartenenza predetta;
- w:** Vettore contenente i pesi calcolati dall'algoritmo in modo da minimizzare l'errore di classificazione, anch'esso di dimensione N ;

Definiamo inoltre la funzione discriminante utilizzata per la classificazione al termine della fase di learning, che assume la seguente scrittura:

$$y = \mathbf{w}^T \mathbf{x} + \mathbf{w}_0 \quad (1.1)$$

con \mathbf{w}_0 termine che tiene conto dell'offset dei punti di learning e \mathbf{w}^T vettore dei pesi \mathbf{w} trasposto. Il termine \mathbf{w}_0 può essere omissso se si effettua una regolazione dell'offset del problema.

1.2 Classificazione Multiclasse

È doveroso sottolineare come nel caso di classificazione a K classi (con $K > 2$) la trattazione si complichì e spesso non sia possibile utilizzare algoritmi che risultavano applicabili nel caso di classificazione a 2 classi (è il caso ad esempio del *Perceptron*).

In letteratura [2] sono riportati vari approcci per estendere il caso di trattazione binaria al caso di K classi, di seguito ne vediamo riportati alcuni, di cui solo 1 risulta però essere efficace.

1.2.1 One-versus-the-rest classifier

Questo approccio prevede di scomporre il classificatore a K classi in $K - 1$ classificatori a 2 classi e per ogni classe, verificare se la realizzazione \mathbf{x} in ingresso è più affine alla classe k -esima o a tutte le altre (ecco perchè il nome *one-versus-the-rest*). Purtroppo agendo in questo modo alcune regioni dello spazio delle realizzazioni rimangono non classificate, ciò significa che se una realizzazione dovesse cadere all'interno di tale regione non si potrebbe assegnare a nessuna classe (figura 1.2).

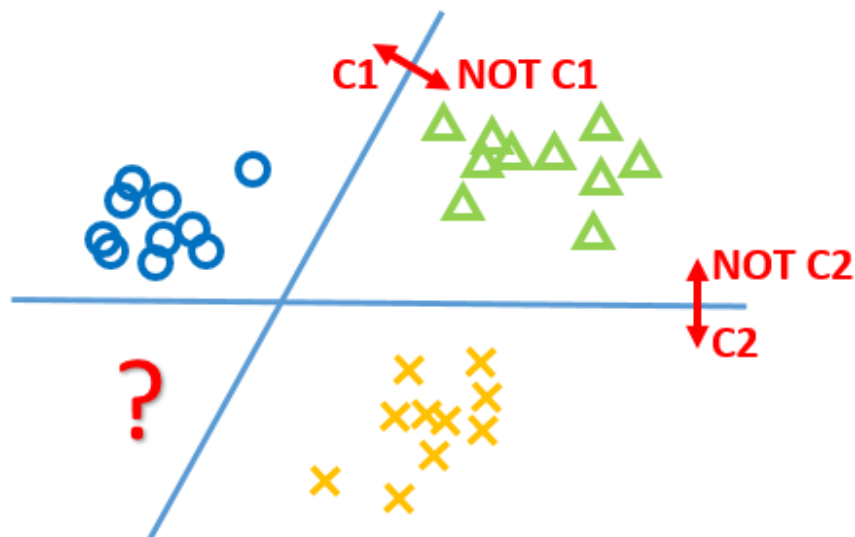


Figura 1.2: One-verus-the-rest problem

1.2.2 One-versus-one classifier

In questo caso si considerano invece tutte le possibili coppie di classi e si realizza un classificatore a 2 classi per ciascuna coppia di esse. Agendo in questo modo però, oltre ad avere un numero molto elevato di classificatori a 2 classi nel caso di K elevato (le possibili coppie di K classi sono $K(K-1)/2$), si ha anche il problema della sovrapposizione di più classi in alcune regioni dello spazio delle realizzazioni, in tali aree non sarà dunque possibile risalire alla classe di appartenenza della realizzazione (figura 1.3).

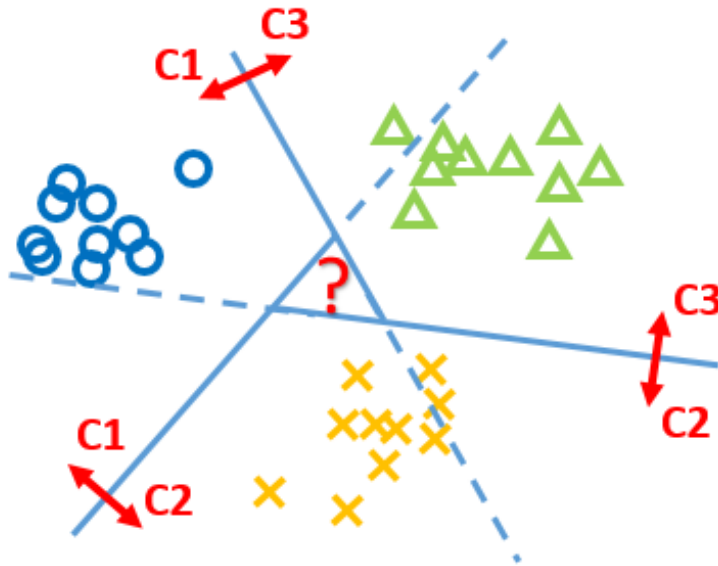


Figura 1.3: One-versus-one problem

1.2.3 Multi-discriminant function

L'approccio che consente di eliminare ambiguità o regioni di spazio non classificate è quello di utilizzare una funzione discriminante per ciascuna classe (y_k), dopodiché si cerca la classe k che massimizza y_k così facendo si trova la classe più affine alla realizzazione \mathbf{x} ovvero si assegna in maniera esclusiva ogni punto dello spazio ad una classe. Le separazioni delle varie classi saranno quei punti nello spazio delle realizzazioni ove y_k ha valori uguali per almeno 2 valori di k (confine fra 2 classi).

Capitolo 2

Conclusioni

Con un'accuratezza del 95% si può affermare che il riconoscitore funziona discretamente, al fine di valutare più a fondo le prestazioni di questa applicazione si potrebbero effettuare in futuro più test variando il tempo di osservazione e il numero di training point per valutare come la distribuzione reale delle realizzazioni infici sull'accuratezza del classificatore.

Per aumentare l'accuratezza si può inoltre agire sulla funzione errore, definendo il problema di minimizzazione non più su una funzione di tipo *sigmoide* ma su di una funzione ricavata “ad hoc” nel caso di distribuzione ad esempio alla *Rayleigh* (la *Logistic Regression* può essere utilizzata per una qualunque funzione appartenente alla classe delle esponenziali di cui la *Rayleigh* fa parte) [3].

Per esplorare ulteriormente lo spazio degli algoritmi di machine learning si può inoltre estendere lo studio al caso di più classi, ricercando soluzioni ottime nel caso di 3 o più utenti, oppure sempre nel caso di 2 classi esplorare la categoria di algoritmi non lineari per trovare la soluzione ottima per ogni tipo di distribuzione.

Elenco delle figure

1.1	Tassonomia degli algoritmi di Machine Learning	4
1.2	One-versus-the-rest problem	5
1.3	One-versus-one problem	6

Bibliografia

- [1] Andrew Ng: Video Course [Online]

<https://www.coursera.org/learn/machine-learning/lecture/Ujm7v/what-is-machine-learning>

- [2] Christopher M. Bishop: *Pattern Recognition and Machine Learning*, Springer-Verlag Ed.,(2006)
- [3] K.Sithamparanathan and A. Giorgetti, *Cognitive Radio Techniques: Spectrum Sensing, Interference Mitigation and Localization*, Boston, USA: Artech House Publisher, 2012.

