UNIVERSITÀ DEGLI STUDI DI MILANO
FACOLTÀ DI SCIENZE E TECNOLOGIE

DIPARTIMENTO DI INFORMATICA
GIOVANNI DEGLI ANTONI

Corso di Laurea triennale in
Informatica

ANALISI DEI DATI PER PROBLEMI DI MEDICINA LEGALE

Relatore: Prof. Dario Malchiodi
Correlatore: Prof. Anna Maria Zanaboni

Tesi di Laurea di:
Alessandro Beranti
Matr. Nr. 855489

ANNO ACCADEMICO 2018-2019

Questo lavoro è dedicato a tutti gli studenti

"Io studio,
ma studiate pure voi,
che se studio solo io non serve a un c. . . o"

– Gli scarabocchi di Maicol & Mirco

"No tale is so good
that it can't be spoiled
in the telling"

– Proverbio

Ringraziamenti

Questa sezione, facoltativa, contiene i ringraziamenti.

Indice

Capitolo 1

Introduzione

Questo documento ha una duplice funzione: da un lato mostra un esempio completo di elaborato finale redatto in LaTeX e conforme allo standard PDF/A, e dall'altro contiene suggerimenti e risposte a domande frequenti poste dagli studenti. Se ne raccomanda, pertanto, un'attenta lettura.

Capitolo 2

Machine Learning

Il termine machine learning, o apprendimento automatico in italiano, si riferisce alla capacità dei computer di apprendere e agire senza essere programmati esplicitamente. Gli strumenti di machine learning si occupano di dotare i programmi della capacità di "apprendere" e adattarsi agli input forniti Al giorno d'oggi siamo circondati da tecnologie basate sull'apprendimento automatico:

- software che rilevano lo spam a partire dai nostri messaggi e-mail
- i motori di ricerca che imparano a fornirci i migliori risultati possibili
- le transazioni con carta di credito sono protette da un software che impara a rilevare le frodi
- Le fotocamere digitali imparano a rilevare i volti
- le applicazioni di assistenza personale intelligenti sugli smartphone imparano a riconoscere i comandi vocali
- addestrare i veicoli per guidare senza assistenza
- applicazioni scientifiche come la bioinformatica, la medicina e l'astronomia

2.1 Come funziona il Machine Learning

Nel Machine learning vengono usati metodi statistici che utilizzano l'esperienza per migliorare le prestazioni di algoritmi o fare previsioni più accurate. La qualità e la dimensione dei dataset, (collezione di dati), raccolti o resi disponibili utilizzati nel processo sono fondamentali per il successo e l'accuratezza delle previsioni fatte.

Nel processo possiamo distinguere diversi aspetti:

- Domain set: raccolta arbitraria di dati, X. Questo è l'insieme di oggetti che si desidera etichettare
- Label set: generalmente si usano un set di etichette del tipo {0, 1} che rappresentano la presenza o l'assenza della caratteristiche si sta cercando
- Training data: è l'input dato in pasto al computer
- Algoritmo: scelto in base ai dati in input, è la regola che viene usata dal sistema per classificare e in generale predire la caratteristica
- Errore di generalizzazione: è la probabilità che non venga predetta l'etichetta corretta

Una categorizzazione dei compiti del machine learning si ha quando si considera l'output desiderato del sistema che può essere di diversi tipi:

- classificazione, i classificatori separano i dati in due o più classi. Quando fornisco un esempio al classificatore, l'algoritmo mi restituisce la classe a cui potrebbe appartenere. Esistono due tipi di classificazione:
  - binaria, quando le etichette sono soltanto due
  - multiclasse se le etichette sono tre o più
- regressione, usata per predire un valore continuo, per esempio il prezzo di una casa date la dimensione e metratura
- clustering, un insieme di input viene diviso in gruppi in modo che i singoli elementi siano simili agli altri punti dello stesso insieme e diversi dagli elementi degli altri. Diversamente da quanto accade per la classificazione, i gruppi non sono noti prima, rendendolo tipicamente un compito non supervisionato.

I compiti dell'apprendimento automatico vengono tipicamente classificati in quattro categorie, a seconda della natura del "segnale" utilizzato per l'apprendimento o del "feedback" disponibile al sistema di apprendimento. Queste categorie, anche dette paradigmi, sono:

- Machine learning supervisionato
- Machine learning non supervisionato
- Machine learning per rinforzo
- Machine learning semi-supervisionato

2.1.1 Machine Learning con apprendimento supervisionato

Attravero l'apprendimento supervisionato cerchiamo di costruire un modello partendo da dei dati di addestramento etichettati, con i quali cerchiamo di fare previsioni su dati non disponibili o futuri. Con il termine "supervisione" si intende quindi che nel nostro insieme dei campioni (o dataset), i segnali di output desiderati sono già noti poiché precedentemente etichettati.

Esistono molti algoritmi per svolgere apprendimento supervisionato, durante il tirocinio ho avuto modo di usare:

- Support Vector Machine
- Decison Tree Classifier
- Random Forest Classifier
- Gaussian Naive Bayes
- Linear Discriminant Analysis
- Multi-Layer Perceptron Classifier

2.1.1.1 Support Vector Machine

Il Support Vector Machine è un algoritmo di apprendimento automatico supervisionato che può essere usato sia per scopi di classificazione che di regressione, ha la sua massima efficacia in problemi di classificazione binaria anche se può essere utilizzato anche per problemi di classificazione multiclasse.

L'SVM è basato sull'idea di riuscire a trovare un iperpiano che divida al meglio un set di elementi in due classi distinte, definiamo alcuni concetti chiave:

- Iperpiano: nel caso in cui sia abbiano solo due dimensioni spaziali x_1 e x_2, l'iperpiano è raffigurato come una linea che separa un insieme di dati. Nel caso in cui le dimensioni siano 3, l'iperpiano è raffigurato come un piano, vedi immagine 1. Con più di 3 dimensioni viene definito "iperpiano".
- Support Vector: chiamati vettori di supporto in italiano, sono i punti che si trovano piu vicini all'iperpiano che divide i dati.
- Margine: è la distanza tra i vettori di supporto di due classi diverse. A metà di questa distanza viene tracciato l'iperpiano. (immagine da inserire)

Il Support Vector Machine ha l'obiettivo di identificare l'iperpiano che divide meglio i vettori di supporto in classi, per fare ciò esegue due step:

- Cerca un iperpiano linearmente separabile che separa i valori di una classe dall'altra. Nel caso in cui ne esista più di uno cerca quello con il margine più alto tra i vettori di supporto in modo da migliorare l'accuratezza del modello
- se l'iperpiano cercato non esiste, Support Vector Machine usa una mappatura non lineare per trasformare i dati di allenamento in una dimensione superiore. In questo modo, i dati di due classi possono sempre essere separati da un iperpiano, che sarà scelto per la suddivisione dei dati.

Un iperpiano linearmente separabile è un iperpiano in cui è semplice distinguere due classi. Nella seguente immagine è visibile come sia possibile disegnare un numero infinito di linee rette per separare i diversi elementi. Il problema è trovare quale tra le infinite rette risulti ottimale, ossia quella che generi il minimo errore di classificazione su una nuova osservazione. Per fare ciò dobbiamo avere i nostri elementi il più lontano possibile dal iperpiano pur rimanendo nella zona corretta. (immagine linearmente separabili)

Nel momento in cui vengono aggiunti nuovi dati di test, il modello decide la classe che gli appartiene. (allego foto con esempio massimizzato)

Dato un training set etichettato:

$$(x_1, y_1), ..., (x_n, y_n) \in R^d and\ y_i \in (-1, +1)$$

dove x_i sono le dimensioni del vettore e y_i sono le etichette.

L'iperpiano ottimale è definito come

$$w_0 + w_1x_1 + w_2x_2 + ... + w_nx_n = 0$$

dove w è il vettore di peso, x è il vettore di caratteristiche di input e w_0 è il bias. In sostanza in n dimensioni un iperpiano di separazione è una combinazione lineare di tutte le dimensioni uguagliate a 0. Ragionando a due dimensioni per semplificare il problema abbiamo che

$$w_0 + w_1x_1 + w_2x_2 = 0$$

I punti che stanno sopra l'iperpiano , e che rappresentano un classe , soddisfano la seguente condizione:

$$w_0 + w_1 x_1 + w_2 x_2 > 0$$

mentre qualsiasi punto che si trova sotto l'iperpiano, appartiene all'altra classe, che è soddisfatta dalla seguente condizione

$$w_0 + w_1 x_1 + w_2 x_2 < 0$$

Includendo anche i limiti dei margini delle classi si hanno le seguenti condizioni:

$$w_0 + w_1 x_1 + w_2 x_2 \geq 1, \ if \ y = 1$$

$$w_0 + w_1 x_1 + w_2 x_2 \leq 1 \ if \ y = -1$$

$$y \in (-1, +1)$$

Se il vettore dei pesi è indicato da w e $\parallel w \parallel$ è la sua lunghezza, allora la dimensione del margine massimo è

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

ciè significa cge minimizzando il vettore peso w, avremo margine massimo che determina l'iperpiano ottimale.

Non è però sempre possibile dividere i dati tramite un iperpiano, nella figura sotto vi è un chiaro esempio (allega foto)

Per utilizzare la classificazione tramite iperpiani anche per dati che avrebber bisogno di funzioni non lineari per essere separati, è necessario ricorrere alla tecnica degli spazi immagine (*feature spaces*). Questo metodo, che sta allabase della teoria delle SVM, consiste nel mappare i dati iniziali in uno spaziodi dimensione superiore. Presupponendo quindi $m > n$, per la mappa si utilizza una funzione

$$\phi : \mathbb{R}^n \to \mathbb{R}^m \tag{1}$$

attraverso la funzione ϕ i dati vengono mappati in uno spazio in cui diventano linearmente separabili e in cui sarà possibile trovare un iperpiano che li separi.

La tecnica degli spazi immagine è particolarmente interessante per algoritmi che utilizzano i dati di training x_i solo attraverso prodotti scalari $x_i \cdot x_j$. In questo caso nello spazio $\mathbb{R}^m$ non si devono trovare esplicitamente $\phi(x_i)$ e $\phi(x_j)$ ma basta

calcolare il loro prodotto scalare $\phi(x_i) \cdot \phi(x_j)$. Per rendere semplice questo ultimo calcolo, che in spazi di dimensioni elevate diventa molto complicato, si utilizza una funzione detta kernel che restituisce direttamente il prodotto scalare delle immagini:

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j) \tag{2}$$

Esistono svariati kernel, i più utilizzati sono:

- Lineare: $K(x, y) = x \cdot y$
- Polinomiale: $K(x, y) = (x \cdot y)^d$ oppure $K(x, y) = (1 + x \cdot y)^d$
- Gaussian Radial Basis function: $K(x, y) = exp(-|x - y|^2)/(2\sigma 2)$
- Sigmoid: $K(x, y) = tanh(kx \cdot y - \delta)$

2.1.1.2 Decision Tree Classifier

2.1.1.3 Random Forest Classifier

2.1.1.4 Gaussian Naive Bayes

2.1.1.5 Linear Discriminat Analysis

2.1.1.6 Multi-Layer Perceptron Classifier

2.1.2 Machine Learning con apprendimento Non Supervisionato

Nell'apprendimento senza supervisione, al contrario di quella supervisionata abbiamo dei dati senza etichetta o dati non strutturati. Con queste tecniche siamo in grado di osservare la struttura dei dati e di estrapolare informazioni cariche di significato. In queste tecniche non si può però contare su una variabile nota relativa al risultato o su di una funzione di ricompensa.

Abbiamo due tecniche che ci vengono in aiuto nell'affrontare problemi di apprendimento non supervisionato: il Clustering e la Riduzione della dimensionalità dei dati.

2.1.3 Machine Learning con apprendimento per Rinforzo

Il terzo tipo di apprendimento automatico è l'Apprendimento per Rinforzo. L'obiettivo di questo tipo di apprendimento è quello di costruire un sistema (agente) che attraverso le interazioni con l'ambiente migliori le proprie performance.

Per poter migliorare le funzionalità del sistema vengono introdotti dei rinforzi, ovvero segnali di ricompensa.

Questo rinforzo non è dato dalle label (etichette) o dai valori corretti di verità, ma è una misurazione sulla qualità delle azioni intraprese dal sistema. Per questo motivo non può essere assimilato ad un apprendimento supervisionato.

Attraverso algoritmi che fanno largo utilizzo del Deep Learning, è tornato di moda questo tipo di apprendimento. Potremmo trovare questo tipo di apprendimento ad esempio nell'addestramento di un sistema per il gioco degli scacchi.

Inizialmente le mosse saranno del tutto casuali e senza una logica. Dal momento in cui il sistema riceverà dei feedback positivi, come ad esempio nel caso in cui mangi una pedina avversaria, allora riceverà un peso maggiore e conseguentemente un rinforzo positivo su quell'azione. Contrariamente in caso di azione negativa, il valore dei pesi su quell'azione andrà in decremento.

Conseguentemente a questi rinforzi, il sistema darà maggior peso alle mosse che gli hanno portato maggiori benefici e tenderà a replicare lo stesso comportamento su nuove mosse future.

2.1.4 Machine Learning con apprendimento semi Supervisionato

Può essere visto come un quarto tipo di apprendimento automatico. In questo caso, al contrario dell'apprendimento non supervisionato, abbiamo che di tutti i dati presenti nel training set, solo pochi di essi sono stati etichettati.

Capitolo 3

Dataset

3.1 Iris

3.2 Incidenti Stradali

3.3 Metodi per ridurre la Dimensionalità

3.3.1 PCA

3.3.2 TSNE

Capitolo 4

Esperimenti

4.1 Model Selection

4.1.1 Scelta degli Iperparametri

$$x_i(n) = a_{i1}u_1(n) + a_{i2}u_2(n) + \cdots + a_{iJ}u_J(n)\,. \tag{3}$$

4.1.2 Scalare i dati

4.2 Errore di Generalizzazione

4.2.1 Training

4.2.2 Cross Validation

4.3 Risultati Ottenuti

4.4 Conclusioni

Nelle conclusioni si tirano le somme di quanto realizzato, facendo un riassunto stringato del lavoro svolto. In particolare vanno dichiarati punti di forza e criticità della ricerca effettuata, nonché quali aspetti dello stato dell'arte siano stati superati dal lavoro in oggetto.

Bibliografia

[1] Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, and Chris Rowley. *The LaTeX Companion.* Addison-Wesley, Boston, second edition, 2004.

Progetto sviluppato presso il Laboratorio di Informatica Musicale
`https://www.lim.di.unimi.it`