

Appunti di Fondamenti di Scienza dei Dati

Riccardo Lo Iacono

Dipartimento di Matematica & Informatica
Università degli studi di Palermo
Sicilia
a.a. 2023-2024

Indice.

1	Pipeline di analisi dei dati	2
1.1	Integrazione dei dati	2
1.2	Trasformazione dei dati	2
1.3	Riduzione della dimensionalità	3
2	Stimatori e predittori	4
2.1	MSE e bias-variance decomposition	4
3	Tipologie di machine learning	5
3.1	Apprendimento supervisionato	5
3.2	Apprendimento non-supervisionato	5
4	Riduzione della dimensionalità	6
4.1	SVD: singular value decomposition	6

– 1 – Pipeline di analisi dei dati.

Il processo di analisi dei dati parte banalmente dall'ottenimento stesso dei dati, cosa che può essere effettuata da database, file di log, o altre fonti. In ogni caso quando si lavora con i dati si deve prestare attenzione a due problemi, uno legato alla strutturazione dei dati, l'altro alla numerosità degli stessi.

In generale comunque, prima di operare con i dati si procede ad una fase di pulizia. Questa è atta ad gestire

- *valori nulli o erronei*: è ovvio che la presenza di valori errati possa influire negativamente sul modello;
- *valori duplicati*: se non gestiti potrebbero portare ad un cattivo addestramento;
- *struttura dei dati*: è opportuno uniformare la struttura con cui i dati sono rappresentati.

Una volta effettuata la fase di pulizia, si può passare alla parte principale dell'analisi dei dati: l'estrazione delle feature. Dal un punto di vista strettamente formale, queste rappresentano caratteristiche valorizzabili dei dati, utili alla risoluzione di un qualche problema. Una volta estratte le feature, queste sono adattate per essere utilizzate da un modello di machine learning, ed eventualmente combinate per definire caratteristiche supplementari.

– 1.1 – Integrazione dei dati.

Fase essenziale della analisi dei dati, *l'integrazione* è atta alla gestione di dati provenienti da fonti diverse. In generale, in questa fase ci si occupa di

- uniformare la struttura dei dati;
- gestire le inconsistenze: ossia dati che, sebbene riferiti ad uno stesso evento, sono discordi tra loro.

Nota: si assume che dati provenienti da fonti diverse, avranno anche una rappresentazione diversa.

– 1.2 – Trasformazione dei dati.

Una volta eseguita la fase di integrazione, i dati sono sottoposti ad operazioni di trasformazione utili ad ottimizzare l'addestramento del modello. Tra queste si considerano

- *lo smoothing*: si tratta di un'ulteriore fase di pulizia dei dati, con l'obiettivo di eliminare del rumore eventualmente permasto;
- *l'aggregazione*: i dati sono tra loro combinati, per poter descrivere concetti più generici;
- *la normalizzazione*: per ridurre il numero di valori assunti dai dati, si procede a ridurre l'intervallo che questi possono assumere.

– 1.2.1 – Normalizzazione.

Sebbene esistano decine di normalizzazione diverse, di interesse risultano

- *la normalizzazione max-min*: si tratta di una normalizzazione dipendente dai valore di massimo e minimo dei dati. Nello specifico, assunti v i dati, v' i dati normalizzati, MAX_0 e min_0 rispettivamente i valori di massimo e minimo iniziali, e assunti inoltre MAX e min gli estremi del nuovo range, segue

$$v' = \frac{v - min_0}{(MAX_0 - min_0)}(MAX - min) + min$$

- *la normalizzazione z-core*: è una normalizzazione dipendente dalla media e dalla deviazione standard dei dati. Nello specifico, posti v i dati, v' i dati normalizzati, μ la media e σ la deviazione standard, si ha

$$v' = \frac{v - \mu}{\sigma}$$

- *la normalizzazione decimale*: si normalizza dividendo per una potenza di 10 tale che, a seguito della normalizzazione, i valori cadano nel range $(0, 1)$. Formalmente

$$v' = \frac{v}{10^k}, k = \min \{k \in \mathbb{N} : \|v'\|_2 \leq 1\}$$

Nota: In generale, poiché si preferisce operare con valori in $[0, 1]$, $Max = 1$, $min = 0$.

– 1.3 – Riduzione della dimensionalità.

In generale, quando si opera con i dati, questi risultano essere di grandi quantità. Poiché gestirli tutti risulta complesso, si preferisce molto spesso ridurre al necessario i dati da utilizzare. Per far ciò si utilizza una delle diverse tecniche di riduzione della dimensionalità, tra queste

- *l'aggregazione*: simile a quella descritta nella fase di trasformazione;
- *la selezione degli attributi minimi*: si procede ad estrarre un sottoinsieme dei dati, tali che questi siano sufficienti a rappresentare l'intero insieme. Di questa tecnica si distinguono
 - *la selezione in avanti*: a partire dall'intero set di dati S , si costruisce un sottoinsieme S' estraendo di passo in passo un elemento da S e inserendolo in S' , se e solo se questi migliora la qualità dei dati in S' .
 - *la selezione all'indietro*: da S si rimuove un elemento, se a seguito di una sua eventualmente estrazione, la qualità degli elementi in S risulta massimizzata.

– 2 – Stimatori e predittori.

L'intero processo di machine learning è basato sulla *teoria dell'apprendimento statistico*. Secondo tale teoria, i dati che rappresentano un qualunque fenomeno, possono essere visti come appartenenti ad una qualche distribuzione di probabilità ignota; e per tale motivo si può assumere che essi siano tra loro indipendenti ed equiprobabili, concludendo pertanto che il valore atteso dei dati di addestramento e quello dei dati di training coincida. A seconda del tipo di machine learning adoperato (si veda *Sezione 3*), l'intero processo fa uso di *stimatori* o di *predittori*.

Definizione: uno stimatore, dal punto di vista statistico, è una funzione che, in funzione dei dati, permette di stimare una quantità/funzione interessante dei dati.

Di questi, supposto Θ_n uno stimatore, Θ il valore da stimare, si distinguono

- *gli stimatori polarizzati:* ossia stimatori tali che

$$\mathbb{E}(\Theta_n) - \Theta \neq 0$$

Cioè, lo stimatore commette un certo errore nell'approssimare Θ ;

- *gli stimatori non-polarizzati:* si tratta di stimatori tali che

$$\mathbb{E}(\Theta_n) - \Theta = 0$$

Osservazione. Se si verifica che

$$\lim_{n \rightarrow \infty} \Theta_n = \Theta$$

si dirà che Θ_n è uno stimatore asintoticamente non polarizzato.

Ulteriori caratteristiche degli stimatori sono la correttezza e la coerenza, nello specifico

$$\begin{aligned} \Theta_n \text{ corretto} &\iff \mathbb{E}(\Theta_n) = \Theta \\ \Theta_n \text{ coerente} &\iff \lim_{n \rightarrow \infty} \text{Var}(\Theta_n) = 0 \end{aligned}$$

Definizione: un predittore è un algoritmo della forma

$$y = f(y) + \varepsilon$$

che permette di stimare la funzione f sulla base dei dati di addestramento x .

Segue dalla definizione che, affinché y sia buona, l'errore ε deve essere minimo.

– 2.1 – MSE e bias-variance decomposition.

Si è parlato di stimatori e predittori, rimane dunque da discutere come determinare la “bontà” degli stessi. Sebbene ne esistano altre, la tecnica che risulta di interesse è l'*MSE*. Questi nel caso degli stimatori è definito come

$$MSE(\Theta_n) = \mathbb{E}[(\Theta_n - \Theta)^2] = \underbrace{\mathbb{E}(\Theta_n^2) - \mathbb{E}(\Theta)^2 - 2\mathbb{E}(\Theta_n)\Theta}_{\text{Bias}(\Theta_n)} + \underbrace{\mathbb{E}(\Theta_n^2) - \mathbb{E}(\Theta_n)^2}_{\text{Var}(\Theta_n)}$$

Osservazione. Tale decomposizione prende il nome di *bias-variance decomposition*.

Supposto infine $y = f(x) + \varepsilon$ un predittore, e \tilde{f} l'approssimazione di f sulla base dei dati, si dimostra che

$$MSE(y) = \mathbb{E}\left[\left(y - \tilde{f}\right)^2\right] = \dots = \text{Var}(f(x)) + \text{Var}(\varepsilon) + \text{Bias}(f(x))^2$$

– 3 – Tipologie di machine learning.

Sia assunto

$$y = f(x, z)$$

un algoritmo di machine learning. Da un punto di vista matematico, y deve rappresentare una valutazione statistica della relazione tra l'input e l'output dell'algoritmo. Formalmente, f è un modello di ML e (x, z) i suoi parametri.

– 3.1 – Apprendimento supervisionato.

Si tratta di una tipologia di machine learning in cui i dati di addestramento sono “etichettati”: cioè presentano un campo che descrive la classe di appartenenza degli stessi. Tra questi si distinguono

- gli algoritmi di regressione;
- gli algoritmi di classificazione.

Nota: entrambe le categorie saranno discusse nelle sezioni a seguire.

In generale, si utilizzano quando l'obiettivo è quello di separare in classi i dati. Si osserva però che affinché l'addestramento possa definirsi “buono”, l'algoritmo deve minimizzare l'errore relativo i dati di addestramento: il cosiddetto *training error*, e l'errore relativo i dati di test: il cosiddetto *test/generalization error*. In fine, per quanto detto e quanto discusso in *Sezione 2*, si deve fare attenzione

- all'*over-fitting*: ossia un fenomeno per cui il modello si è troppo adattato ai dati, non riuscendo dunque a generalizzare;
- all'*under-fitting*: fenomeno opposto all'*over-fitting*, è una condizione in cui il modello ha appreso poco dai dati, pertanto non ha le capacità sufficienti a generalizzare.

– 3.2 – Apprendimento non-supervisionato.

È un caso di machine learning in cui i dati non sono “etichettati”. Tra questi, gli algoritmi di interesse sono

- quelli di riduzione della dimensionalità;
- quelli di clustering.

Nota: come per gli algoritmi di ML supervisionato, entrambe le categorie saranno discusse nelle sezioni a seguire.

In tal senso, l'obiettivo è quello di determinare similarità tra i dati.

– 4 – Riduzione della dimensionalità.

Come anticipato in *Sezione 3*, la riduzione della dimensionalità è una tecnica utilizzata dall'apprendimento non-supervisionato. L'idea alla base è quella di ridurre la quantità dei dati che si deve analizzare, riducendoli al minimo. In generale, alla base di una qualsiasi tecnica di riduzione vi è la PCA, nel seguito descritta.

– 4.1 – SVD: singular value decomposition.

Tecnica fondamentale per la riduzione della dimensionalità, permette di decomporre la matrice rappresentanti i dati in tre sotto-matrici. Nel dettaglio, questa è definita come segue.

Definizione: sia $X \in \mathbb{R}^{n,m}$, allora

$$\exists U, \Sigma, V : X = U \Sigma V^T, \forall X \in \mathbb{R}^{n,m}$$

ove $U \in \mathbb{R}^{n,n}$, $\Sigma \in \mathbb{R}^{n,m}$ e $V \in \mathbb{R}^{m,m}$.

Ciò equivale a dire, da un punto di vista numerico, che

$$X = \sum_{i=1}^n \sigma_i u_i v_i^T$$

ove $u_i \in U$, $v_i \in V$ e $\sigma_i \in \Sigma$, $\sigma_i \geq \sigma_j, \forall i < j < n$.

Geometricamente segue che

- Σ è una matrice diagonale, nello specifico è data dagli autovalori di XX^T e $X^T X$;
- rispettivamente, U e V , sono le matrici date dagli autovettori di XX^T e $X^T X$.

Dalla definizione si è detto che la SVD esiste comunque scelta la matrice X . La dimostrazione di ciò è sostenuta dal seguente teorema.

Teorema. Sia $C = X^T X \in \mathbb{R}^{m,m}$, allora C è diagonale, simmetrica e definita positiva.

Dimostrazione: La dimostrazione segue banale dal *Teorema di decomposizione spettrale*. Per esso si ha che $C = V T V^T$, con $T = \text{Diag}(\lambda_1, \dots, \lambda_m)$ e $r = \text{Rank}(X)$. Posto allora $\sigma_i = \sqrt{\lambda_i}$, segue che

$$\Sigma = \begin{pmatrix} \text{Diag}(\sigma_1, \dots, \sigma_r) & 0_{r, (m-r)} \\ 0_{r, (m-r)} & 0_{m-r, m-r} \end{pmatrix}$$

e ponendo inoltre

$$u_i = \frac{X}{\sigma_i} v_i$$

che si dimostrano ortogonali, segue, completando a base

$$U = [u_1 \quad \dots \quad u_r \quad u_{r+1} \quad \dots \quad u_n]$$

In conclusione, il calcolo della SVD si riduce a

- calcolare $C = X^T X$;
- calcolare $C = V T V^T$;
- calcolare $\Sigma = \sqrt{T}$;
- calcolare dalla relazione $XV = U\Sigma$, gli u_i .

Nota: si definiscono i σ_i valori singolari.

– 4.1.1 – Low-Rank approximation.

La Low-Rank approximation è una tecnica appartenente alle cosiddette *budget SVD*, dove per “budget” si intende il fatto che dimensioni delle matrici sono ridotte a un certo k . Più rigorosamente, la low-rank approximation è definita come segue.

Definizione: sia $X \in \mathbb{R}^{n,m}$ e sia $r = \text{Rank}(X)$, se

$$X_r = \sum_{i=1}^r \sigma_i u_i v_i^T$$

è una SVD di X , allora scelto un $k < r$ è possibile definire una SVD ridotta ai primi k termini. Ossia,

$$X_k = \sum_{i=1}^k \sigma_i u_i v_i^T.$$

Si definisce X_k matrice di miglior approssimazione di X di rango k .

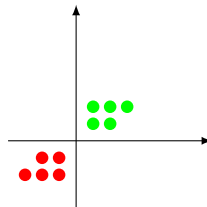
Cioè sostanzialmente, X_k è la matrice di rango k che, fra tutte le matrici di rango k , approssima meglio X .

– 4.1.2 – PCA: principal component analysis.

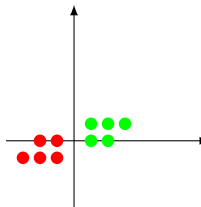
La PCA è una tecnica con funzionalità duplice, essa permette infatti di scoprire strutture a livelli multidimensionale nei dati, e permette inoltre di ridurre la dimensionalità dei dati.

Sostanzialmente, supposta $\Sigma = X^T X \in \mathbb{R}^{n,m}$, invece di considerare tutta la matrice, si considera la versione data dai suoi autovalori. Di questi, siano $\lambda_1, \dots, \lambda_m$, si considera quello che ha il maggior contenuto informativo, e fatto ciò, si procede a ridurre lungo la sua direzione.

Esempio: sia supposto uno scenario come quello in *Figura 1.a*. Eseguendo su di



1.a: Dati base



1.b: Dati riferiti agli autovalori

esso il passaggio agli autovalori (*Figura 1.b*), da cui concludendo la PCA, si ottiene *Figura 2*.

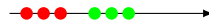


Figura 2: Effetto della PCA

Ricordando che le osservazioni della matrice possono essere visti come vettori di uno spazio n – *dimensionale*, seguono le seguenti definizioni.

Definizione: le nuove coordinate dei vettori corrispondenti le osservazioni, sono definiti *scores*.

Definizione: i pesi che definiscono le componenti principali sono dette *loadings*.