FONDAMENTI DI SCIENZA DEI DATI

APPUNTI A CURA DI: RICCARDO LO IACONO

Università degli studi di Palermo
a.a. 2022-2023

Indice.

– 1 – Introduzione: pipeline di data processing.

Ogni progetto di scienza dei dati segue una pipeline ben definita. Questa parte dall'ottenimento dei dati, dati che possono essere acquisiti da sorgenti diverse: sensori, file di log, ecc. Indipendentemente dall'origine, ai dati sono legati due problematiche, quali

- gestione dei dati strutturati;
- gestione del volume dei dati.

Assunto di aver gestito queste problematiche, si può procedere alle fasi di data-cleaning e di estrazione delle feature. Qui con *data-cleaning* si fa riferimento ad un insieme di tecniche atte a migliorare la qualità dei dati; in tale fase

- si gestiscono i valori nulli: questi se non gestiti porterebbero a un cattivo addestramento del modello;
- si eliminano eventuali duplicati: ciò è atto a prevenire un bias nel modello;
- si ricercano e rimuovono eventuali outliers: è ovvio che se non si procedesse a gestirli questi influenzerebbero negativamente il modello;
- si standardizzano i dati: se si utilizzano dati provenienti da fonti diverse, è necessario che questi abbiano una stessa struttura.

Successivamente si può procedere alla fase di *feature-extraction*: ossia la fase in cui si selezionano quelle caratteristiche valorizzabili dei dati, che possono essere utilizzate per addestrare il modello, ed eventualmente combinate per definire concetti più generali.

– 1.1 – Data Integration.

Ulteriore fase della pipeline di analisi dei dati è quella di integrazione dei dati. Come anticipato, quando si addestra un modello è possibile utilizzare dati provenienti da sorgenti diverse; ciò implica verosimilmente che i dati presenteranno una struttura diversa, rendendo complicato l'addestramento del modello stesso.

Per risolvere tale problematica, nella fase di integrazione ci si occupa di uniformare la struttura dei dati, applicando eventuali trasformazioni, e di gestire eventuali inconsistenze. Con quest'ultime s'identificano dati che, sebbene associati ad uno stesso fenomeno, presentano valori tra loro discordi.

– 1.2 – Data integration.

Nella precedente sezione si è fatto a delle trasformazioni applicabili ai dati. Queste, atte a migliorare la qualità delle informazioni, si distinguono in

- *smoothing:* si tratta di una tecnica che applica un'ulteriore pulizia ai dati, così da rimuovere ulteriore rumore;
- *aggregazione:* i dati sono combinati tra loro così da permettere la descrizione di concetti più generali;
- *normalizzazione:* si riduce il range di valori assunti.

Per quanto riguarda la normalizzazione: esisto varie soluzioni, tra le più utilizzate: la *scalatura decimale*, con la quale si riduce il range tra (0, 1); e la *z-core*, con la quale si sottrae ai dati la media degli stessi, e li si divide per la rispettiva deviazione standard.

– 1.3 – Data reduction.

Poiché generalmente quando si opera con i dati si ha a che fare con quantità dell'ordine dei TB, sebbene ciò possa sembrare un lato positivo, in quanto il modello avrebbe più dati da cui apprendere; ciò risulta eccessivamente lento. Per questa e altre ragioni nella maggior parte dei casi si preferisce ridurre la mole dei dati (*Sezione* (4)).

– 1.4 – Tipologie di dati.

I dati porterebbero essere suddivisi per diversi aspetti, è però di interesse la distinzione tra dati *interdipendenti* e *non-interdipendenti*. Sostanzialmente la differenza è la seguente: nel caso di dati interdipendenti, questi hanno una qualche relazione (**eg:** *peso-altezza*); segue che informazioni in uno o più dati, dipendono o influenzano altri dati. Per i dati non-interdipendenti non si hanno tali relazioni.
Osservazione. È giusto puntualizzare che il concetto di interdipendenza è fortemente legato al problema in esame.

– 2 – Stimatori.

Definizione: uno *stimatore* (dal punto di vista statistico), è una funzione dei dati di cui si è in possesso, che fornisce la migliore approssimazione di una quantità/proprietà a cui si è interessati.

Assunto che Θ_n sia uno stimatore, θ il valore vero, si distinguono due casi

- Θ_n è polarizzato, cioè
$$\mathbb{E}(\Theta_n) - \theta \neq 0$$
Da cui si deduce che lo stimatore commette un certo errore nell'approssimazione.
- Θ_n non è polarizzato, da cui
$$\mathbb{E}(\Theta_n) - \theta = 0$$
Si può inoltre dire che Θ_n è asintoticamente non polarizzato se
$$\lim_{n\to\infty} \mathbb{E}(\Theta_n) = \theta$$
o equivalentemente
$$\lim_{n\to\infty} \mathbb{E}(\Theta_n) - \theta = 0$$

In fine, supposto T_n uno stimatore, τ il valore da stimare, si ha che

$$T_n \text{ è } \begin{cases} \text{corretto} & \iff \mathbb{E}(T_n) = \tau \\ \text{coerente} & \iff \lim_{n\to\infty} \operatorname{Var}(T_n) = 0 \end{cases}$$

– 2.1 – Mean square error.

Dato uno stimatore, sia questi Θ_n, è necessario misurare la "bonta" dello stesso, e, in generale, per farlo si utilizza il *mean square error (MSE)*. Questi, definito come

$$\begin{aligned} MSE &= \mathbb{E}\left[(\Theta_n - \theta)^2\right] \\ &= \operatorname{Bias}(\Theta_n)^2 + \operatorname{Var}(\Theta_n) \end{aligned}$$

ove $\operatorname{Bias}(\Theta_n)^2 = \mathbb{E}(\Theta_n)^2 + \theta^2 - 2\,\mathbb{E}(\Theta_n)\theta$.

– 3 – Tipologie di apprendimento.

Quando si deve addestrare un modello di machine learning, ciò può essere effettuato secondo due "logiche", quali:

- *apprendimento supervisionato*: al modello sono forniti dati di addestramento in cui compaiono delle etichette[1];
- *apprendimento non-supervisionato*: i dati di addestramento sono, per la maggior parte, privi di etichette.

Indipendentemente dal modello di addestramento scelto si deve fare attenzione a due problemi: *over-fitting* e *under-fitting*. Quest'ultimi fanno riferimento, rispettivamente, al caso in cui il modello, apprendendo troppo dai dati di addestramento, non è capace di generalizzare; e al caso in cui, viceversa, non avendo appreso a sufficienza, il modello non ha le capacità minime a generalizzare.

– 3.1 – Apprendimento non-supervisionato.

Si tratta di una tipologia di addestramento alla cui base vi è l'intensione di ricercare delle similarità tra i dati. Di questi, sebbene ne esistano altri si considereranno:

- *la riduzione della dimensionalità* con la quale si tenta di ridurre il numero di osservazioni, ad uno molto minore, cercando di mantenere lo stesso livello di qualità descrittiva;
- *clustering* con la quale i dati sono raggruppati tra loro sulla base di similarità.

Osservazione. Nello specifico nel caso del clustering, sarà analizzato nel dettaglio il *k-mean clustering*.

– 3.2 – Apprendimento supervisionato.

Tipologia di addestramento in cui, sulla base dei dati di addestramento, si cerca di determinare la "classe" classe di appartenenza di nuovi dati. In tale "categoria" rientrano algoritmi che fanno uso

- di *regressione* con cui ai dati di input si assegna un valore numerico;
- di *classificazione* grazie al quale i dati sono divisi sulla base della classe di appartenenza stimata.

Osservazione. Tutto il processo di generalizzazione è di tipo statistico è basato sulla *teoria di apprendimento statistico*[2].

Poiché la scelta della classe è determinata dalle conoscenze apprese, è opportuno che i seguenti errori risultino minimizzati.

- *Training error*, errore relativo ai dati di addestramento;
- *test* o *generalization error*, errore con cui si stima quanto correttamente il modello sia in grado di generalizzare. Nel particolare, fornendo al modello dati mai visti precedentemente, si verifica se questi sono correttamente categorizzati.

[1] Si pensi a queste come a dei campi che definiscono la classe di appartenenza del singolo record.

[2] Si tratta di una teoria secondo la quale tutti i dati possono essere visti come appartenenti ad una qualche distribuzione ignota, da cui si può assumere anche la loro indipendenza. Sulla base di tale ipotesi si può allora assumere che il valore atteso dei dati di addestramento e quello dei dati di test, coincidano.

– 4 – Riduzione della dimensionalità.

Come detto in precedenza, la *riduzione della dimensionalità* è una tecnica utilizzata per addestrare il modello utilizzando un numero ridotto di osservazioni, riduzione che non deve però alterare la qualità dei dati, semmai deve migliorarla. Nel seguito si discuterà la SVD che permette di ridurre la dimensionalità della matrice, e a seguire delle applicazioni della stessa.

– 4.1 – Single value decomposition.

Sia X una matrice che rappresenti un dataset. Sia supposto che il numero di osservazioni in X sia sufficientemente elevato. Allora

$$\exists U, \Sigma, V : X = U \times \Sigma \times V^T$$

con $U \in \mathbb{R}^{n,n}, \Sigma \in \mathbb{R}^{n,m}, V \in \mathbb{R}^{m,m}$.
Osservazione. Si distinguono due tipologie di SVD: la *full* con la quale U, Σ, V hanno le dimensioni sopra riportate; la *economy* nella quale $U \in \mathbb{R}^{n,m}, \Sigma, V \in \mathbb{R}^{m,m}$.

Se considerato un punto di vista numerico, la SVD di una qualche matrice X può essere vista come una combinazione lineare di opportuni vettori. Cioè

$$X = \sum_{j=1}^{k} \sigma_j u_j v_j^T$$

con $u_j \in U, v_j \in V, \sigma_j \in \Sigma$. Si ha inoltre che $\sigma_i \geq \sigma_j, \forall i, j : i < j$.

Geometricamente si ha invece che

- gli elementi della diagonale di Σ, risultano essere le radici quadrate degli autovalori di $X^T X$ e XX^T;
- U è la matrice data dagli autovettori di $X^T X$, mentre V è composta dagli autovettori di XX^T.

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

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

e ponendo inoltre

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

che si dimostrano ortogonali, segue, completando a base

$$U = \begin{bmatrix} u_1 & \cdots & u_r & u_{r+1} & \cdots & u_n \end{bmatrix}$$

– 4.2 – Principal component analysis.

La *principal component analysis* o più semplicemente PCA, è una tecnica che permette di rilevare strutture (bi-, tridimensionali), all'interno dei dati, fornendo informazione su eventuali colinearità. Considerando la PCA in se: sia B una matrice rappresentante un dataset, per la geometria esiste sempre B^T. Se ad ogni colonna di B se ne sottrae la corrispettiva media, si può definire

$$C = \frac{B^T B}{n-1}$$

contenente, per ogni i, j la covarianza tra gli elementi di colonna i e quelli di colonna j. Da ciò, la diagonale di C si comporra unicamente delle varianze.

Si dimostra che con tale costruzione, gli autovettori di C sono ortogonali, autovalori che prendono il nome di *componenti principali*. Si conclude da ciò che se alcuni vettori descrivono bene la varianza della matrice, è possibile utilizzare i relativi autovettori per rappresentare l'intero sistema.

Osservazione. Il calcolo delle componenti principali può essere realizzato dunque con la SVD, questo perche essa fornisce un approccio numericamente robusto.

– 5 – Clustering.

Quando si parla di clustering, si fa riferimento ad un insieme di algoritmi di apprendimento non-supervisionato, principalmente applicati per operazioni di data mining. In generale si parla più precisamente di m-clustering, definito come segue.

Definizione: dato $X \in \mathbb{R}^n$ un vettore di features, si definisce *m-clustering* di X il partizionamento di X in m classi $c_1, \dots, c_m$, tali che

- $c_i \neq \emptyset, i = 1, \dots, m$;
- $\bigcup_{i=1}^{m} C_i = X$;
- $C_i \cap C_j = \emptyset, \forall i \neq j, i, j = 1, \dots, m$.

Analizzando il processo di clustering nel dettaglio, questi può essere sintetizzato nelle seguenti fasi.

1. *quantizzazione delle similarità*: nel quale si definiscono i range entro i quali due diversi record appartengano ad una stessa classe;
2. *criterio di clustering*: si sceglie la funzione che determina le effettive classi;
3. *scelta dell'algoritmo di clustering*;
4. *convalida dei risultati*;
5. *interpretazione dei risultati*.

Osservazione. È opportuno puntualizzare che esistono diverse tipologie di clustering, tutte suddivisibili in *partizionali* o *gerarchici*.

– 5.1 – Misure di qualità.

Ora sebbene gli algoritmi di clustering siano utilizzati per il ML non-supervisionato, è necessario stabilire la qualità dell'algoritmo, e per far ciò si definiscono delle misure relativamente ai cluster. Tali misure si dividono in *interne* ed *esterne*.

Tra le più diffuse ed utilizzate la *sum of square errors (SSE)*, definita come

$$SSE = \sum_{j=1}^{m} \sum_{x \in C_j} dist(x, y_j)$$

con y_j rappresentate della classe C_j.

– 5.1.1 – Misure interne: coesione e separazione.

Si tratta di due misure essenziali, queste infatti stabiliscono, rispettivamente, quanto elementi di uno stesso cluster siano correlati (*coesione*) e, quanto in media gli elementi di due cluster distinti siano distanti.

– 5.1.2 – Misure interne: WSS.

È una misura di tipo strutturale; quel che fa è misurare la "bonta" di ogni cluster, in maniera indipendente dalle altre. In genere può anche essere utilizzata per stimare il numero di cluster necessario.

– 5.1.3 – Stima del numero di cluster.

Per stabilire il numero di cluster si può operare in diversi modi, uno di questi è fornito dalle misure di validazione. Considerando il k-means, il valore ideale di k può essere ottenuto come segue:

- si esegue l'algoritmo per valori di k diversi;
- per ognuno dei k considerati, si calcola WSS;
- si considera il grafico dato dai valori di WSS.

– 5.2 – K-means.

Se $dist\left(X_i, Y_j\right) = \left\|X_i - Y_j\right\|_2^2$, si ha a che fare con il *k-mean* clustering. Con tale algoritmo, l'utente deve solamente fornire i rappresentanti dei cluster, e da esse l'algoritmo procedera a comporli. Nello specifico l'algoritmo procede come segue.

1. per ogni elemento x, si calcola la distanza con ogni rappresentate μ_i definito dall'utente, e si assegna x al cluster il cui rappresentante ha la distanza minore da x;
2. per ogni cluster così formato si calcola il centroide (la media);
3. si ripete il processo finché non vi è convergenza[3].

Circa la complessità dell'algoritmo, questa è $\mathcal{O}(nkdi)$, ove n è il numero di elementi del dataset, k il numero di cluster desiderati, i il numero di iterazioni e d il numero di attributi delle osservazioni.

[3] Si raggiunge la convergenza se vi è una ripetizione di medie, o se si è definito un qualche criterio di arresto.

– 6 – Classificazione.

L'idea alla base della classificazione è molto semplice; supposto S un insieme di dati etichettati, che si ricordano essere dati che definiscono la propria classe di appartenenza, si costruisce un modello di ML tale che, fornendo allo stesso dati non ancora visionati, questo sia in grado di predirne la classe di appartenenza. Sebbene esistano vari algoritmi di classificazione, di interesse risulta essere *support vector machine*, descritto nel seguito.

– 6.1 – SVM: support vector machine.

Considerando la sua versione più elementare (il caso lineare) SVM costruisce un iper-piano della forma

$$\mathbf{w}\mathbf{x} + b = 0$$

con $\mathbf{w} \in \mathbb{R}^n$ e b costante, parametrizzanti l'iper-piano, tale da separare le osservazioni.
Osservazione. L'iper-piano, generalmente, non è unico.

Poiché generalmente l'iper-piano non è unico, come si stabilisce quello da utilizzare? Di norma si tende ad utilizzare quello che massimizza il cosiddetto *margine*: si tratta della distanza tra l'osservazione più vicina all'iper-piano e il limite decisionale dello stesso.

– 6.1.1 – Hard margin SVM.

Da quanto precedentemente detto, posto $\mathbf{x}$ il vettore associato a un punto p, $\mathbf{w}$ il vettore ortogonale all'iper-piano e c la distanza tra $\mathbf{w}$ e l'iper-piano, segue che

- se $\mathbf{w}\mathbf{x} = c$: il punto p sara sovrapposto al limite decisionale, e dunque non sarà classificabile;
- se $\mathbf{w}\mathbf{x} < c$: il punto sara classificato negativamente;
- in ultima istanza, p è classificato positivamente.

Da ciò, risulta fondamentale massimizzare c, e si dimostra che ciò si verifica massimizzando

$$\arg\min_{\mathbf{w},\mathbf{x}} \left\{ \frac{2}{\|\mathbf{w}\|} \right\} \tag{1}$$

– 6.1.2 – Soft margin SVM.

Si osserva però che l'hard margin SVM è applicabile unicamente al caso lineare. Poiché nella realtà tale situazione è estremamente rara, è necessario poter estendere SVM al caso non lineare. Tale estensione è data proprio dal *soft margin*.

Considerando la (1), e ricordando che massimizzare un funzione equivale a minimizzare l'inverso della stessa, si ha

$$\arg\min_{\mathbf{w},\mathbf{x}} \left\{ \frac{\|\mathbf{w}\|}{2} \right\}.$$

Si deve però tenere traccia dell'eventuale errore di classificazione, errore rappresentato dalla funzione ζ. Da cui in conclusione, si deve minimizzare

$$\arg\min_{\mathbf{w},\mathbf{x}} \left\{ \frac{\|\mathbf{w}\|}{2} \right\} + c \sum_{i=1}^{n} \zeta_i$$