

Data Analytics

Ilaria Battiston

Anno scolastico 2018-2019

Contents

1	Introduzione	3
2	Data analytics issues	3
2.1	Istanze, classi e attributi	3
2.2	Data analytics tasks	4
3	Data preprocessing	5
4	Unbalanced data	5
4.1	SMOTE	6
4.2	Tomek Link Method	6
5	Subset selection and evaluation	6
5.1	Feature reduction	6
5.2	Feature extraction	6
5.3	Feature selection	7
5.4	Wrappers	7
6	Evaluation issues	8
6.1	Numerosità del dataset	9
6.2	Affidabilità delle misurazioni	10
7	Network analysis	10
7.1	Proprietà delle reti	11
8	Network centrality	12
8.1	PageRank	14
9	Community detection	15

1 Introduzione

L'analisi dei dati è una disciplina introdotta nel 1935 da *Fisher* (F-test, per capire se due campioni sono originari della stessa popolazione) e in seguito sfruttata per automazione e gestione di dati prevalentemente non strutturati.

Le prime analisi descrittive risalgono alla fine degli anni '70, insieme ai primi software e al linguaggio R, il quale è successivamente stato integrato con tecnologie big data. Le prime modalità di visualizzazione (grafici a torta, istogrammi) nascono per dare una forma ai fenomeni legati al business (**business intelligence**).

Alla fine degli anni '90 questi modelli convergono nel **machine learning**, che permette non solo la creazione di modelli descrittivi ma anche previsioni e prescrizioni in base al contesto.

Negli ultimi decenni Google ha sviluppato strumenti come TensorFlow per poi renderli accessibili al pubblico, e sono nate diverse discipline e nuove applicazioni di data analytics (Google Flu Trends).

Alcuni utilizzi di questo campo sono i database transazionali per i sistemi di raccomandazione, IoT tramite wireless sensor data, o le analisi in ambito medico. Il volume di dati da elaborare è immensa, essendo questi prodotti in tempo reale dagli utenti e dai dispositivi.

Per **data analytics** si intende la generazione di valore da dati per scopi decisionali, cioè la trasformazione dei dati in prodotti. L'intervallo di tempo considerato include presente, passato e futuro, ed è necessaria una profonda comprensione empirica dei modelli.

I big data hanno la caratteristica di essere molto economici e in grande quantità, quindi hanno bisogno di velocità e disponibilità, con query ottimizzate. I sistemi di gestione sono prevalentemente NoSQL.

Più il dato è complesso, più crescono velocità e volume a discapito della varietà (e viceversa).

- Dati strutturati: hanno forma **proposizionale** (tabellare), permettono query con range numerici e matching esatto di stringhe;
- Dati non strutturati: **grafi** (reti) e **testo** libero, i quali compongono circa l'80% del totale dei dati disponibili in un'azienda;
- Dati semi-strutturati: via di mezzo tra strutturati e non. La maggior parte dei dati è libero, ma esistono linee guida per le rappresentazioni (ad esempio HTML).

2 Data analytics issues

2.1 Istanze, classi e attributi

Istanza (oggetto, record): esempio descritto da un numero di attributi.

Attributo (campo, caratteristica): misura gli aspetti delle istanze.

Classe: gruppo di istanze.

Le istanze sono tipi specifici di esempi che devono essere classificati, associati ed eventualmente raggruppati. Possono essere *dipendenti* o *indipendenti* e sono caratterizzate da un numero predefinito di attributi: l'input ai modelli di apprendimento sono istanze contenute in un dataset,

basandosi su assunzioni **IID** (indipendenti e identicamente distribuite, derivano dalla stessa distribuzione).

La rappresentazione in forma proposizionale (tabellare) implica la definizione degli attributi, rimuovendo le relazioni ed esprimendole eventualmente tramite campi e condizioni. Viene considerato solo l'insieme delle osservazioni disponibili ("mondo chiuso").

Il processo di flattening di relazioni per creare un'unica tabella si indica con **proposizionalizzazione**: è possibile con qualsiasi insieme finito, ma causa "biased models" con irregolarità o dati replicati rispetto al modello originale. Le relazioni $1 : n$ sono gestite associando un campo aggiuntivo nel lato 1 oppure con matrici booleane. L'operazione di *join* generalmente causa rappresentazioni non totalmente veritiere (distorsioni), ma che hanno minore complessità computazionale.

Gli attributi sono le *features* che costituiscono lo spazio di rappresentazione dell'input: devono sempre avere lunghezza predefinita, eventualmente si ricorre all'uso di flag. L'esistenza di alcuni attributi (derivabili) può dipendere da altri, e questo potrebbe aumentare la complessità del modello.

Le variabili **nominali** (simboliche, categoriche, discrete) rappresentano una quantità nominale e non hanno relazioni logico-matematiche tra loro.

Le variabili **ordinali** (numeriche) impongono un ordine (numeri, stringhe).

Spesso non è immediata la distinzione tra nominali e ordinali.

Conoscere la natura dell'attributo è essenziale per poter effettuare confronti e avere un criterio per le operazioni, trattare i dati mancanti e gestire le problematiche legate alla qualità dei dati.

2.2 Data analytics tasks

2.2.1 Classification learning

Supervisionato, si occupa della classificazione di campioni predefiniti in classi, secondo approcci di machine learning (regressioni, alberi di decisione, reti bayesiane).

Il modello deve avere buone capacità di generalizzazione per input mai trattati prima, ed è possibile definire modelli di apprendimento in base a regole logiche su rappresentazioni proposizionali. Le regole logiche sono codificate usando *if*.

2.2.2 Clustering

Il clustering serve per identificare gruppi di istanze simili. Gli algoritmi sono non supervisionati: la classe di un esempio non è conosciuta in partenza. Vengono utilizzati per la segmentazione.

2.2.3 Associazione

Modello predittivo non supervisionato con l'obiettivo della comprensione di associazioni: dall'esistenza di un attributo prevedere l'esistenza di un altro.

2.2.4 Predizione numerica

Supervisionato, modelli con un valore target in input: cerca di individuare relazioni tra attributi numerici (regressione).

3 Data preprocessing

I dati solitamente contengono problematiche che devono essere affrontate prima di poterli dare in input al modello: il preprocessing è un'attività fondamentale per individuare rumore, inconsistenze e incorrettezze.

Il processo di **data cleaning** si occupa di rimpiazzamento di valori mancanti e smoothing dei dati rumorosi. Ci sono modelli in grado di gestire per natura i missing values, ma altri hanno necessariamente bisogno della completezza. Non tutti i dati incompleti possono essere sostituiti.

- MCAR (Missing Completely At Random): la distribuzione di un esempio con valori mancanti non dipende da altri attributi;
- MAR (Missing At Random): la distribuzione di un esempio con valori mancanti dipende dagli attributi osservati, non necessariamente mancanti;
- NMAR (Not Missing At Random): la distribuzione di un esempio con valori mancanti dipende da attributi con valori mancanti.

I dati mancanti si possono ignorare, convertire a valori di default o rimpiazzare. Alcune tecniche di sostituzione implicano l'utilizzo della media (per valori continui con distribuzione normale) o della moda (discreti). Un altro modo è k-NN, che associa la classe sulla base della maggioranza degli oggetti vicini.

Un metodo di discretizzazione (smoothing) è il binning: divide il range in N intervalli in base alla media (distribuzioni normali) o alla frequenza (skewed).

4 Unbalanced data

Una delle problematiche che riguardano modelli di apprendimento predittivi e prescrittivi si verifica quando la distribuzione dei campioni di una determinata classe sono molto più frequenti rispetto a un'altra (es. contesto medico). La maggior parte dei campioni sono corretti, ma inutili: non è necessario fare analisi sulla maggioranza, sapendone già il comportamento.

Per bilanciare i dati si usano due tecniche:

- Oversampling: costruzione di un dataset con un numero desiderato di campioni dalla classe di minoranza e uno uguale dalle altre;
- Undersampling: eliminazione di un numero arbitrario di campioni dalla classe di maggioranza, in modo da avere lo stesso numero rispetto alla classe di minoranza.

Questi metodi sono applicabili con due classi, ma anche con un maggior numero. Ci sono algoritmi che si focalizzano su uno o sull'altro, e approcci ibridi che li combinano. Si assume che i dati siano corretti, rappresentativi e in assenza di rumore.

4.1 SMOTE

SMOTE (Synthetic Minority Oversampling Technique) è un algoritmo iterativo di oversampling:

1. Per ogni campione di minoranza, trova i suoi k elementi di minoranza più vicini;
2. Di queste ne sceglie n in modo casuale;
3. Calcola il punto medio tra quello iniziale e ciascuno degli n (Generated Synthetic Instance);
4. Il GSI viene aggiunto al dataset.

4.2 Tomek Link Method

Tomek Link è un algoritmo che si basa sulla frontiera della classe. Un Tomek Link è una coppia di istanze $\langle E_1, E_2 \rangle$ tale che E_1, E_2 appartengano a una classe diversa e non esistano altri esempi E_k più vicini a ognuno di essi.

L'undersampling viene effettuato sui campioni della classe di maggioranza che non sono parte di Tomek Link.

5 Subset selection and evaluation

Questo processo consiste nella selezione di un sottoinsieme di attributi rilevanti per la costruzione del modello. Ciò comporta minore complessità e tempistiche più brevi.

5.1 Feature reduction

La riduzione e la modifica dello spazio di input è indispensabile, soprattutto per i modelli predittivi, per permettere una mappatura dall'input all'output (durante l'apprendimento): il target eredita delle caratteristiche, che non devono essere irrilevanti e ridondanti.

Le risorse computazionali sono ridotte (crescono esponenzialmente al numero delle variabili), e può essere complesso trovare le distribuzioni di probabilità dei campioni, quindi è meglio rimuovere alcuni attributi.

Riducendo le features, si riduce la dimensionalità dello spazio: gli attributi scelti devono essere sufficienti per distinguere i campioni tramite alberi di decisione (valori booleani).

5.2 Feature extraction

La feature selection trasforma attributi esistenti in uno spazio dimensionale minore. Dato un insieme di attributi $x = \{x_i | i = 1 \dots N\}$, si trova una mappatura $y = f(x) : R^N \rightarrow R^M$ con $M < N$ tale che il vettore trasformato preservi la maggior parte delle informazioni o della struttura.

Un mapping ottimale $y = f(x)$ risulterà in una minima probabilità di errori non incrementata, ma non esiste un modo sistematico per generare trasformazioni non lineari.

5.3 Feature selection

La feature selection seleziona un sottoinsieme degli attributi esistenti. Dato un insieme di attributi $x = \{x_i | i = 1 \dots N\}$, si trova un sottoinsieme $x_m = \{x_{i1}, x_{i2}, \dots, x_{iM}\}$ con $M < N$, che ottimizza una funzione obiettivo $J(Y)$.

Questo è necessario nei casi in cui le features siano difficili da ottenere o si voglia estrarre regole significative; è utile quando le unità di misura vengono perse o i dati non sono numerici.

Per trovare i candidati ottimali e i potenziali sottoinsiemi si ricorre al filtering: una ricerca esaustiva implica $\binom{n}{m}$ combinazioni, quindi si utilizzano metaeuristiche.

Criteri di ranking:

- Ranking variabile, ordinamento delle features in base a una funzione di scoring che misura la rilevanza e risulta in una permutazione ordinata;
- Correlazione, usando il coefficiente R di Pearson per m campioni, che misura la similarità e l'approssimazione a una funzione lineare;

$$R(f_i, y) = \frac{\text{cov}(f_i, y)}{\sqrt{\text{var}(f_i)\text{var}(y)}} \in [-1, 1]$$
- Information Gain: classificazione in base all'informazione (entropia) guadagnata da ogni feature e alla sua contribuzione alla riduzione dell'incertezza della classe;

$$IG(C, A) = H(C) - H(C|A)$$
- Subset evaluation tramite CFS (Correlation-based Feature Selection), selezione di g attributi altamente correlati tra loro ma incorrelati con le altre classi;

$$M_s = \frac{k r_{cf}}{\sqrt{k + k(k+1) r_{ff}}}$$
 con k features, r_{cf} media della correlazione tra classi, r_{ff} la media dell'intercorrelazione tra features.

5.4 Wrappers

I wrappers sono metodologie per riconoscere le possibili interazioni tra le variabili, a discapito del tempo computazionale (al contrario dei filtri).

La subset evaluation viene effettuata tramite wrappers, i quali includono politiche per valutare le features rispetto al potere predittivo del modello di apprendimento. Le performances devono soddisfare un determinato livello di qualità per poter accettare il sottoinsieme, e l'obiettivo è la massimizzazione della qualità.

I risultati possono variare a seconda della ricerca nello spazio delle possibili variabili, e l'output è strettamente dipendente dall'approccio utilizzato (paradigma di predizione). Questi parametri devono essere definiti e valutati attentamente, misurando man mano le performance e la robustezza del modello.

Fasi dell'algoritmo:

1. Selezione di un subset di attributi (uno o più);
2. Induzione di un algoritmo di apprendimento (albero di decisione, SVM, ...);
3. Valutazione dell'accuratezza o del risultato in generale;

4. Se essa non è sufficiente, ripetizione del processo.

Il problema della selezione del subset ottimale è NP-hard, quindi si ricorre a strategie metaeuristiche di ricerca. Le classi si dividono in forward e backward (aggiunta o rimozione di features nell'insieme), e la metodologia viene misurata tramite cross-evaluation, inducendo il wrapper sul training set e valutando sul test set.

La tecnica è comunque molto pesante computazionalmente, però permette un utilizzo semplice e universale dei wrappers.

6 Evaluation issues

Lo sviluppo di modelli per la valutazione delle features ha comportato la necessità di una misurazione accurata della predizione, introducendo errori e stime di performances.

Gli errori sul training data non sono buoni indicatori della qualità dell'apprendimento, perché non è possibile prevedere l'output con altri dati (la varianza è alta). I test set per questo sono utili a capire la variabilità dell'output con differenti input.

Problemi più comuni:

- Overfitting, quando il modello è molto accurato (troppo complesso) sui dati di training ma ha scarsi risultati con il testing e incapacità di generalizzazione;
- Underfitting, quando la varianza è nulla e tutti i dati sono interpretati allo stesso modo troppo generico (semplice), i risultati sono scarsi sia nel training che nel testing.

Nel momento in cui i risultati del training iniziano a divergere da quelli del testing, c'è la probabilità di overfitting. Il range ideale ha complessità ridotta del modello con poco scostamento delle curve.

Ci sono diversi metodi di valutazione degli errori:

- Misurazione delle performance, contando quante istanze sono classificate correttamente e in base a ciò calcolare error rate e accuracy;
- Falsi negativi/positivi, da minimizzare;
- Problemi multi-classe, osservando come il modello si comporta in base a ogni classe;
- Precisione, tenendo conto delle singole istanze con recall e self-measure;
- Curve ROC, grafici di veri positivi e falsi negativi al variare della soglia;
- Complessità computazionale.

Ognuno di questi è distinto dall'utilizzo di veri/falsi positivi e negativi, che contribuiscono al calcolo di alcuni indicatori di performance. Le misurazioni distinguono tra actual target values a_1, a_2, \dots, a_n e predicted target values p_1, p_2, \dots, p_n .

$$\text{Mean-squared error: } \frac{(p_1 - a_1)^2 + \dots + (p_n - a_n)^2}{n}$$

$$\text{Mean-absolute error (meno sensibile agli outliers): } \frac{|p_1 - a_1| + \dots + |p_n - a_n|}{n}$$

$$\text{Root ean-squared error: } \sqrt{\frac{(p_1 - a_1)^2 + \dots + (p_n - a_n)^2}{n}}$$

Altre formule utilizzate:

1. Accuracy, $A = \frac{\text{numero di istanze correttamente classificate}}{\text{istanze totali}} = \frac{TP+TN}{TP+TN+FP+FN}$;
2. Precision, $P(k) = \frac{\text{numero di istanze correttamente classificate come } k}{\text{istanze totali classificate come } k} = \frac{TP+TN}{TP+FP}$;
3. Recall, $R(k) = \frac{\text{numero di istanze correttamente classificate come } k}{\text{istanze totali della classe come } k} = \frac{TP+TN}{TP+FN}$;
4. F-measure, $F(k) = \frac{2 \cdot P(k) \cdot R(k)}{P(k) + R(k)}$, distinta da:
 - Macro-average, $\frac{1}{N} \sum_{i=1}^N P(i)$, che consiste nella media di tutte le F-measures di ogni classe;
 - Micro-average, $\sum_{i=1}^N \frac{C_i}{N} P(i)$, per assegnare ad alcune classi una maggiore importanza (dove C è la cardinalità).

6.1 Numerosità del dataset

Generalmente il dataset, se sufficientemente grande, viene diviso in training e testing (solitamente 2/3 training, 1/3 testing) ed essi vengono utilizzati rispettivamente per costruire e valutare il classificatore.

Più aumenta di dimensioni il training set, più migliora il classificatore; più aumenta il testing set, più la stima degli errori è accurata. Dopo la valutazione, i dati vengono uniti per costruire la versione finale.

In mancanza di informazioni a riguardo, si assume che la distribuzione dei parametri sia uniforme, e poi si aggiusta la precisione in base alla performance.

Se il quantitativo di dati è piccolo, si usano strategie di repeated holdout: il processo di training viene ripetuto più volte con diversi sottoinsiemi di attributi, ma ciò causa overlap. A ogni istanza viene assegnato un numero casuale per l'assegnazione.

La cross-validation evita l'overlap dividendo i dati in k subsets (di solito 10) e dividendo ulteriormente essi in testing e training (k -fold), stratificando i gruppi e calcolando stime globali. La stratificazione permette il rispetto della cardinalità di ogni classe distribuendole comunque in modo relativamente uniforme.

Se un'istanza di minoranza viene valutata più volte in testing, la sua originale distribuzione marginale viene persa, e ci sono più previsioni per lo stesso dato.

Altre tecniche meno utilizzate sono quelle di bootstrap, con dati poco numerosi: ogni istanza uò far parte sia del set di training che di testing. Ogni elemento viene scelto più volte, il che è utile in presenza di classi di minoranza. La varianza è ridotta, ma i risultati tendono a essere pessimisti.

Una particolare istanza ha probabilità $1 - \frac{1}{n}$ di non essere selezionata, quindi la probabilità di essere selezionata è:

$$\left(1 - \frac{1}{n}\right)^n \approx e^{-1} = 0.368$$

Questo significa che il training data ha circa il 63.2% delle istanze.

Le curve di learning mostrano come l'accuratezza cambia al variare della dimensione del campione, stabilizzandosi dopo una certa quantità.

Quando ci sono davvero poche istanze si utilizza leave-one-out, una forma di cross-validation che crea tanti folds quante le istanze e ne usa uno solo come testing, iterativamente.

6.2 Affidabilità delle misurazioni

Per confrontare due modelli di misurazione delle performances si utilizzano gli intervalli di confidenza, stimando quanto una misura di errore è vicino alla realtà. Si può affermare (T di Student) che p si trova in un intervallo specifico con un determinato livello di confidenza.

I test di significatività indicano quanta confidenza si può avere che ci sia una differenza tra due risultati di modelli. L'ipotesi nulla in questo caso è che non esistano differenze rilevanti.

Il T -test serve per capire se le medie di due campioni sono diverse, prendendo individui da tutti i sottoinsiemi considerati nella cross-validation. Si ha:

$$\frac{m_x - \mu}{\sqrt{\sigma_x^2/k}}$$

Le medie sono (approssimativamente) normalmente distribuite se il numero di campioni è sufficientemente elevato. La differenza delle medie m_d ha una distribuzione Student con $k - 1$ gradi di libertà. La versione standardizzata di m_d è:

$$t = \frac{m_d}{\sqrt{\sigma_d^2/k}}$$

7 Network analysis

Una rete è un insieme di entità con proprietà simili chiamate nodi, connesse da archi. Questi possono essere reali, dinamici, astratti, ..., ma ogni tipologia può essere rappresentata da un grafo.

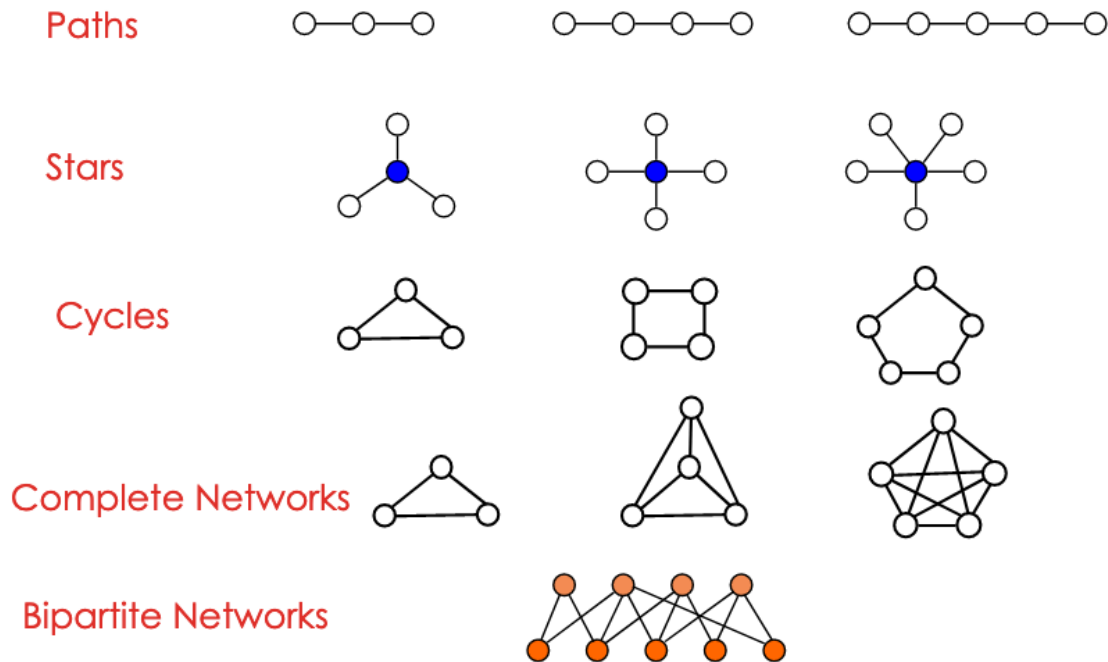
Terminologia:

- Arco diretto (indiretto), ha una direzione (nodo di entrata e di uscita);
- Attributo, caratteristica associata a un arco come il peso, il ranking, la tipologia;
- Multiarco, più archi collegano la stessa coppia di nodi;
- Grafo connesso, esiste un cammino per ogni coppia di nodi.

Capire la struttura della rete è importante per le operazioni di gestione e le analitiche. La rappresentazione è generalmente tramite matrice di adiacenza: A_{ij} è 1 se il nodo i ha un arco che lo collega al nodo j , 0 altrimenti. Un 1 sulla diagonale indica un ciclo.

Un altro modo per rappresentare un grafo è la lista di adiacenza, più veloce se i nodi sono sparsi e per visualizzare i vicini di ciascuno.

Le reti considerate dalle network analytics sono larghe in termini di nodi, hanno collegamenti sparsi e si evolvono dinamicamente.



7.1 Proprietà delle reti

Le reti complesse sono difficili da visualizzare, quindi si ricorre a misure e statistiche descrittive. Queste servono per indicare la distribuzione dei nodi, il raggruppamento o la centralità.

Il primo concetto da considerare è il grado di un nodo, cioè il numero di archi entranti o uscenti. Formalmente, $\text{outdegree}(i) = \sum_{j=1}^n A_{ij}$, $\text{indegree}(j) = \sum_{i=1}^n A_{ij}$ (matrice di adiacenza).

Queste quantità permettono di calcolare il grado totale di un nodo (somma), la media e la distribuzione. $P(k)$ è la probabilità che un nodo casuale abbia grado k (robustezza della rete, diffusione di informazioni). Essa viene calcolata dividendo il numero di nodi con grado k per il totale, e poi normalizzando il valore.

Il numero massimo di archi in una rete di N nodi è:

$$E_{max} = \binom{N}{2} = \frac{N(N-1)}{2}$$

Un grafo con grado massimo è completo, e il suo grado medio è $N-1$.

La distribuzione è essenziale per capire il livello di connessione del grafo: se è bassa, la maggior parte dei nodi hanno pochi link, e qualche nodo ne ha molti. I nodi che hanno molti collegamenti in una rete sparsa si definiscono **hub**, ed è importante capire le motivazioni per cui essi hanno così tanti archi entranti o uscenti.

Legge di Metcalfe: il valore della rete è proporzionale al quadrato del numero dei nodi: più nodi ci sono, più una rete diventa importante (es. fax). Il problema nella realtà è che la maggior parte dei grafi sono sparsi, e gli archi hanno peso differente.

Un cammino è una sequenza di nodi in cui ognuno è adiacente all'altro (connesso da un link), di cui la lunghezza è costituita dal numero di archi. Bisogna tenere conto anche dell'eventuale direzione dei collegamenti, nei grafi diretti. Si hanno:

- Distanza, numero di archi appartenenti al cammino più corto tra due nodi;
- N_{ij} numero dei cammini più corti estratti dalla matrice di adiacenza;
- Visita BFS per trovare i cammini minimi;
- Diametro, massima distanza tra una coppia di nodi;
- Distanza media, per un grafo diretto, la media delle distanze per ogni coppia.

La connessione delle componenti di un grafo può essere determinata osservando la matrice di adiacenza: le parti connesse sono confinate in sottomatrici quadrate.

Il coefficiente di clustering, compreso tra 0 e 1, è la probabilità che due vicini di un nodo abbiano un collegamento. $C_j = 1$ se il grafo è completo.

Più la rete è densa, quindi, maggiore è il coefficiente: ci si trova in contesti con elevato numero di archi. La media può dare un'idea generale del grado di clustering.

8 Network centrality

La conoscenza della struttura di una rete permette di calcolare e catturare particolari caratteristiche, come la centralità in termini di singoli nodi e di intera rete.

Il concetto di centralità è strettamente dipendente dal contesto e dallo scopo, e non è univoco: ogni definizione dà informazioni diverse rispetto alle altre, e si distingue in base alle tipologie di nodo. Un nodo può essere più o meno centrale rispetto a un altro in base al criterio di misurazione (indegree, betweenness, ...).

Ognuna di questi parametri può essere più o meno informativo a seconda delle applicazioni nella vita reale, ma in generale si parla di misure locali. Il confronto tra reti di grandezza differente è possibile normalizzando i valori.

La centralizzazione di una rete mette a confronto il nodo con centralità più alta con tutti gli altri. Per determinarla (per esempio con il grado), viene usata la formula di Freeman:

$$C_D = \frac{\sum_{i=1}^N C_D(n^*) - C_D(i)}{(N-1)(N-2)}$$

Il denominatore rappresenta la più grande somma di differenze ottenibile in una rete analoga (caso limite), mentre il numeratore è il valore pratico della centralità, cioè la differenza normalizzata tra il nodo con grado massimo e tutti gli altri.

Il range varia nell'intervallo $[0, 1]$ dove 1 rappresenta una rete a stella e 0 una rete ad anello: la prima ha un nodo centrale unico con collegamenti solo con esso, mentre la seconda è completamente decentralizzata.

La centralizzazione in base al grado è una misura puramente locale, che può variare molto in base alla struttura del grafo, quindi potrebbe non essere sufficiente a descrivere l'influenza di un nodo sull'intera rete.

Per effettuare analisi in base allo shortest-path si utilizza la betweenness centrality. Naturalmente anche questa è dipendente dalla struttura, e quantifica quanto un nodo può essere un “ponte” nei cammini verso gli altri.

$$C_B(i) = \sum_{j \neq k} \frac{g_{jk}(i)}{g_{jk}}$$

$$C'_B(i) = \frac{C_B(i)}{(n-1)(n-2)/2}$$

Il calcolo è simile a quello con il grado, mentre la seconda formula rappresenta la centralità normalizzata. Considerando un grafo orientato e diretto, il coefficiente di normalizzazione cambia perché è necessario considerare la direzionalità. j può essere uguale a k per tenere conto di tutti gli archi differenti.

La closeness centrality si basa sulla lunghezza media dello shortest path tra un vertice a e tutti gli altri, e rappresenta l'efficienza di esso nello scambio di informazioni.

$$C_C(i) = \left[\frac{\sum_{j=1}^N d(A, j)}{N-1} \right]^{-1}$$

In altre parole, vengono considerate tutte le possibili distanze e divise per il totale dei nodi (normalizzazione con il numero di shortest-path possibili), per poi invertire il valore in modo da rappresentare una distanza breve come una centralità alta.

Di solito, gli indicatori di centralità sono correlati positivamente. Quando uno di essi è più alto rispetto agli altri, la rete ha proprietà interessanti (nodi ego).

La reciprocità è la capacità di una rete diretta di avere doppi link (direzioni opposte) tra i vertici. Questo problema è fondamentale per analizzare la centralità e capire eventuali pattern o errori nelle stime.

La densità di una rete è il rapporto tra il numero di archi esistenti e il numero massimo di archi possibili $n(n-1)/2$. Questo è utile per identificare comunità, cioè gruppi di nodi con densità molto diversa tra di loro rispetto al resto del grafo.

L'attaccamento preferenziale è la maggiore centralità di un nodo rispetto ai vicini (non alla rete intera), ed è un fenomeno che permette di individuare popolarità, qualità e altri attributi delle pagine.

Non tutti i nodi sono equivalenti: un vertice A è importante se è collegato ad altri vertici importanti (PageRank). In questo modo, la vicinanza a nodi importanti dà un peso alla centralità. Questo concetto viene rappresentato con gli autovalori:

$$x_i = \frac{1}{\lambda} \sum_k a_{ki} x_k \implies \lambda x = xA$$

Un nodo con molti collegamenti non ha necessariamente una centralità alta rispetto agli autovalori, perché il peso può essere minimo, e viceversa. La matrice A è di adiacenza, e i coefficienti di centralità corrispondono agli autovettori.

8.1 PageRank

Dividendo il valore della centralità per l'outdegree (nodi uscenti), si evita la trasmissione della enorme centralità di un singolo nodo a tutti quelli adiacenti. Questo concetto, insieme alle catene di Markov, è ampiamente utilizzato dall'algoritmo PageRank.

Si ha, per ogni coppia di nodi, che un arco (i, j) è una referenza, mentre un arco (j, i) è una raccomandazione. Se esiste una referenza, c'è la probabilità che i due nodi siano in qualche modo legati tra di loro (dal punto di vista del contenuto).

Se un nodo ha un rank elevato, avrà più impatto su quelli adiacenti (cede un po' della sua capacità), mentre se un nodo non è centrale quelli adiacenti non vengono penalizzati. Il grafo è diretto e orientato, definendo il coefficiente come:

$$P(i) = \sum_{(i,j) \in E} \frac{P(j)}{O_j}$$

Si effettua una sommatoria per tutti i nodi, rapportando il numero di archi entranti P con il numero di archi uscenti J . In forma algebrica, questa formula è un sistema di n equazioni e variabili dato da $P = A^T P$.

Il grafo del Web rappresentato come catena di Markov ha come archi la probabilità di transizione da una pagina all'altra, e PageRank simula tutti i possibili percorsi di navigazione. Ogni nodo è uno stato, e i link sono le probabilità di transizione.

La distribuzione di probabilità iniziale è uniforme, ma a seconda del contesto cambieranno la matrice di transizione e gli stati. Per simulare il processo, vengono stimati il numero di outlink da ogni nodo per poi calcolare la probabilità di random surfing ($1/O_i$), per poi inserire i valori nella matrice.

Si ha $\sum_{i=0}^n p_0(i) = 1$, e $\sum_{j=1}^n A_{ij} = 1$ assumendo che ogni pagina all'inizio abbia la stessa importanza, e di non avere informazioni aggiuntive. Se questa proprietà è soddisfatta, la matrice è stocastica.

Una catena di Markov definita da una matrice stocastica è caratterizzata da una distribuzione stazionaria (converge a π) se A è irriducibile e aperiodica.

Applicando questo concetto al web, non è possibile avere tutte e 3 le caratteristiche, essendoci pagine non raggiungibili (non hanno archi uscenti), quindi queste vengono rimosse oppure viene assunta una probabilità uniforme. Esiste anche la probabilità di fare salti completamente random all'interno della pagina web.

Il ranking (centralità) di una pagina può essere calcolato risolvendo un sistema di equazioni lineari, formato dalla probabilità di una transizione volontaria sommata al random jump dove le incognite sono le probabilità e la loro somma è 1. La soluzione iterativa del sistema corrisponde agli autovalori della matrice di adiacenza normalizzata.

8.1.1 Catene di Markov

Un processo stocastico è una collezione indicizzata (ordinata) di variabili X_t casuali raccolte da una serie di osservazioni, dove t di solito denota il tempo. Le variabili rappresentano caratteristiche da modellare, e sono discrete o continue.

Uno stato è un numero finito di valori che può assumere X_t , che possono realizzarsi quindi in modo diverso nel tempo. Un processo stocastico soddisfa la proprietà Markoviana se:

$$P(X_{t+1} = j \mid X_0 = k_0, X_1 = k_1, \dots, X_{t+1} = k_{i-1}, X_t = i) = P(X_{t+1} = j \mid X_t = i) \quad \forall t$$

Ogni stato del sistema al tempo t , quindi, dipende solo dallo stato $t - 1$ ed esiste una distribuzione (grafo) di probabilità in grado di rappresentare le variazioni. Le probabilità condizionali si definiscono one-step transition probabilities, e sono stazionarie se la probabilità resta costante.

Un processo stocastico è una catena di Markov a stati finiti se:

1. Ha un numero finito di stati;
2. Gode della proprietà Markoviana al primo ordine;
3. La distribuzione di probabilità è stazionaria;
4. Esiste un set (vettore) di probabilità iniziali per ogni stato.

Per avere una descrizione completa è necessario specificare le probabilità di transizione attraverso una matrice, che rappresenta la probabilità di passare da uno stato i a uno stato j . Si ha π come distribuzione delle probabilità iniziali (uniforme, di solito). Elevando la matrice alla n , cioè rappresentando i cambiamenti nel tempo, il risultato tenderà a essere stazionario al crescere di n .

9 Community detection

Non esiste una definizione formale di comunità, ma il concetto si riferisce alla struttura (aspetto topologico e relazionale) di sottocomponenti dense di un grafo. Talvolta sono considerati anche gli attributi dei nodi, per effettuare inferenze sulla loro appartenenza.

I dati sono in formato collegato con attributi discreti, e le applicazioni si basano sulle proprietà dei grafi.

Scoprire come funzionano le comunità e identificarle permette di capirne il funzionamento e le relazioni tra individui con le stesse caratteristiche. Anche i diversi gruppi interagiscono tra di loro, e possono essere correlati, fornendo informazioni sullo schema globale.

Le reti sociali online sono rappresentate da nodi che comunicano strutturalmente e condivisioni o produzioni di contenuti simili. Questa proprietà distingue la tipologia di rete rispetto alle altre, e viene utilizzata per il sentiment analysis. Predirre quali vertici potrebbero essere connessi (link prediction) è uno dei potenziali usi.

A seconda del risultato cercato, ci sono due distinzioni di comunità: disgiunte o overlapping, con nodi che appartengono a uno o più sottoinsiemi.

Ci sono quattro diverse famiglie di algoritmi:

- Node-centric community, in cui ogni nodo deve soddisfare proprietà. Serve per identificare cliques, raggiungibilità e grado;
- Group-centric community, in cui tutto il gruppo deve soddisfare proprietà (es. densità). Viene usato per garantire un livello di coesione su grafi di dimensioni ridotte;

- Network-centric community, in cui si considera l'intera rete da suddividere in n (prefissato) insiemi disgiunti. L'obiettivo è il partizionamento in base a similarità, spazio latente (spettro) o massimizzazione della modularità;
- Hierarchy-centric community, organizzando la rete secondo una struttura gerarchica.

Una clique è un insieme di nodi in cui ognuno è adiacente agli altri (es. triangolo). La ricerca può essere incentrata su quelle di dimensione massima o su tutte in generale, e il brute-force è impraticabile: si stabilisce una grandezza minima k , quindi un grado dei nodi almeno pari a $k - 1$.

Questo approccio ha delle limitazioni, come la scarsa frequenza di cliques anche in grafi complessi. La rimozione degli archi rischia di tagliare potenziali gruppi.

Un altro criterio è la raggiungibilità in k step (archi), per individuare una rete in cui la maggiore distanza tra ogni coppia di nodi sia minore o uguale a k .

Una quasi-clique è una sottostruttura con i nodi che soddisfano alcune caratteristiche come la densità minima γ :

$$\frac{2|E_s|}{|V_s|(|V_s| - 1)} \geq \gamma$$

Il problema per trovare γ è di ottimizzazione, cioè stabilire quanti nodi possono appartenere a una quasi-clique tramite algoritmi greedy.

La similarità è definita dalla tipologia di interazioni: due nodi sono strutturalmente equivalenti se sono connessi agli stessi vicini. Il concetto può essere esteso dal primo ordine, essendo molto severo e poco probabile.

Le similarità vengono individuate applicando le formule (coseno sul continuo, Jaccard sul discreto) ai singoli vettori della matrice di adiacenza.

$$\cos(\theta) = \frac{A \cdot B}{\|A\| \cdot \|B\|} \quad J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Per le reti più grandi vengono utilizzati gli algoritmi di clustering, di cui il più famoso è k -means: esso trova centroidi e comunità in rappresentazioni sferiche nello spazio geometrico, ma fallisce con densità o forme diverse.

La matrice di adiacenza può essere modificata in modo da avere suddivisioni più chiare, oppure i nodi centrali saranno stabiliti in base al grado e non solo il peso degli archi (matrice Laplaciana).

Spectral è un ulteriore algoritmo che consiste nella ricerca dei gruppi tramite pre-processing, computazione di autovettori e autovalori e organizzazione in possibili community.

Alcune connessioni si possono creare grazie a un fattore casuale, e non danno informazioni significative quindi dovrebbero essere rimosse nello studio dei gruppi. Una rete con link motivati è strutturalmente molto diversa da una costruita con criteri random.

Il concetto di modularità $[-1, +1]$ definisce appunto questa differenza.

$$\frac{1}{2E} \sum_C \sum_{i \in C, j \in C} \left(A_{ij} - \frac{d_i d_j}{2E} \right) z_{ij}$$

Vengono considerate tutte le possibili coppie di nodi in relazione alla probabilità di una connessione random rispetto alla presenza o meno di un arco. $z_{ij} = 1$ se i nodi sono nella stessa community.

Più è ampia la differenza tra la probabilità casuale e la matrice di adiacenza, più una relazione tra vertici è forte (non è dovuta al caso). La modularità globale è pari a 0 quando tutti i nodi sono all'interno di un unico cluster, e se ne calcola lo spettro.