DATA ANALYTICS

2022/2023

Sommario

PRESENTAZIONE CORSO

Modalità d'esame composta da progetto e orale.

Svolgimento del progetto:

1- Sceglierai tra uno dei possibili domini applicativi (finance, sensor networks, social networks, etc.).
2- Sceglierai eventuali sorgenti esterne di dati per arricchire/integrare i datasets forniti durante il corso.
3- Implementerai workflow di analytics per estrarre insights dai dati, fare previsioni e prendere decisioni.
4- Visualizzerai insights e previsioni
5- Esporrai la tua interpretazione dei dati, la progettazione degli analytics e gli insights ottenuti.

Orale con 4 domande di teoria.

Il progetto da fino ad un massimo di 24 punti, le domande danno +2 o -2 punti.

INTRODUZIONE AL CORSO

L'analisi dei dati è uno strumento molto importante ed utile al giorno d'oggi, infatti è possibile ottenere numerose applicazioni pratiche dall'analisi dei dati tra cui:

- Suggerimenti su show che possono interessarti su Netflix
- Individuare pattern comuni dei tentativi di frode (Sicurezza informatica)
- Sistemi di valutazione dei prodotti e soddisfazione del cliente di Facebook o Amazon
- Trouble shooting automatico

L'analisi dei dati si basa sull'utilizzo di grandi quantità di dati che vengono chiamate Big Data. Reperire grandi quantità di dati è piuttosto semplice dato che il nostro mondo è permeato da una infinità di dati che possono essere creati dagli utenti stessi che utilizzano un applicativo, da delle ricerche scientifiche, dall'internet of things o più in generale di tutto quello che avviene in rete di cui ovviamente si può tenere traccia.

I Big Data nonostante la loro estrema utilità hanno anche delle caratteristiche complesse da gestire che non sono trascurabili durante il loro utilizzo:

- Il volume dei dati grezzi
- La velocità con cui questi dati cambiano nel tempo
- La varietà dei dati messi a disposizione
- La qualità dei dati messi a disposizione

Definiamo alcuni concetti della Data Analytics:

Si dice Business Intelligence una collezione di approcci per raccogliere, salvare, analizzare e fornire accesso a dei dati per fornire agli utenti delle informazioni che consentono di eseguire decisioni di mercato basate su fatti e su dati analizzati.

Si dice Analytics quel processo scientifico atto al trasformare i dati in informazioni utili ad eseguire decisioni utili alla azienda oppure ad implementare dei modelli di ML in grado di eseguire previsioni guidate da questi dati in ingresso.

Si dice Data Analytics quella scienza che fa uso dei computer, delle statistiche, del ML e delle interazioni uomo-macchina per ottenere, pulire, integrare, analizzare, visualizzare e interagire con grandi quantità di dati e trasformarli in dei data products.

L'obiettivo principale della Data Analytics è dunque quello di trasformare i dati in data product.

Facendo un confronto tra i tradizionali database e la data analytics possiamo dire che:

1- Il valore dei dati è incredibilmente prezioso nei DB mentre nella data analytics è di basso costo
2- Il volume nei dati nei DB è modesto mentre nella data analysis è di enormi dimensioni
3- I DB vengono usati principalmente per i contenere dati su banche, dipendenti, censimenti, ecc.. mentre nella data analytics si trattano dati riguardanti i click su un sito, i tweet delle persone, log dei GPS, ecc..
4- Le priorità da avere in un DB sono avere consistenza, verificabilità e possibilità di correggere gli errori; mentre nella data analysis le priorità riguardano la velocità, la disponibilità e la ricchezza delle query di ricerca dati
5- Le proprietà che devono essere garantite nei DB sono transazionali e le proprietà ACID (Atomicity, Consistency, Isolation, Durability), mentre nella data analytics abbiamo le proprietà del CAP (Consistency, Availability, Partition Tolerance) theorem ed eventualmente la consistenza
6- Il linguaggio usato per creare i DB principalmente è SQL mentre per la data analytics si usano NoSQL, MongoDB, Cassandra, ecc…

Se volessimo invece confrontare la Business Intelligence con la data analytics potremmo dire che la prima di occupa di interrogare il passato e i suoi dati, mentre la seconda interroga il passato, il presente e il futuro.

Per concludere confrontando la data analytics con il ML possiamo dire che:

- Il ML sviluppa un modello di apprendimento mentre la data analytics ne esplora, modifica ed eventualmente ibrida diversi modelli
- Il ML individua proprietà matematiche dei modelli mentre la data analytics comprende le proprierà empiriche dei modelli
- Il ML migliora le sue prestazioni su dei dataset di dimensione ridotta e con dei dati puliti, mentre la data analysis implementa dei metodi che fanno uso di grandissime quantità di dati non necessariamente puliti

In conclusione possiamo dire che a differenza del ML la data analytics una volta concluso il suo processo porta a visualizzare delle informazioni nascoste e consente l'esecuzione di azioni specifiche.

TIPI DI DATI E MISSING FEATURES

Quando si va a definire una pipeline di analytics bisogna sempre tenere in considerazione la tipologia di dati che si ha a disposizione che possono essere: strutturati, semi-strutturati e destrutturati. Dato che al giorno d'oggi ogni oggetto che ci circonda produce una grande quantità di dati, abbiamo la certezza che non riceveremo mai dei dati strutturati come quelli utilizzati in ML ovvero delle tabelle di dati. Da un recente studio sulla crescita esponenziale dei dati possiamo notare come la maggior parte dei dati reperibili al giorno d'oggi siano di tipo semi-strutturato o destrutturato. Bisogna anche porre particolare attenzione sui tipi di analytics che possono essere: descriptive, prescriptive, predictive.

L'obiettivo della data analytics è quello di definire dei processi che non siano triviali che ci permettono di identificare dei pattern che siano comprensibili, utili, validi e che ci permettano di prendere delle decisioni consapevoli.

TIPI DI DATI

Definiamo con più precisione le varie tipologie di dati che si possono avere:

- Dati strutturati: si presentano in forma tabellare struttura tipica vista nel ML
- Dati non strutturati: tipicamente identificati con del testo libero che richiede query più sofisticate e keyword-based
- Dati semi strutturati: hanno al loro interno una struttura in qualche modo gerarchica navigabile e consultabile

I dati non strutturati e semi strutturati saranno oggetto del corso in quanto i file di testo (non strutturati) e i grafi (semi strutturati) sono i tipi di dati che maggiormente vengono prodotti e sono molto difficili da gestire.

TIPI DI DATA ANALYTICS

Le diverse tecniche di analisi dei dati che abbiamo a disposizione sono:

- Modelli descrittivi: modelli di analisi di strutture rappresentate tramite grafi e reti in grado di identificare cosa è successo alla struttura analizzata
- Modelli diagnostici: identificano perché si è sviluppato un dato fenomeno (causalità)
- Modelli predittivi: modelli che ci consentono di fare una previsione sulla base della analisi di dati passati
- Modelli prescrittivi: modelli che ci dicono come un certo fenomeno potrebbe svilupparsi come ad esempio dei modelli che si occupano di giocare una partita a scacchi, quindi prende delle decisioni sulle osservazioni che esegue sull'ambiente in tempo reale

I dati in forma tabellare con i quali siamo abituati a lavorare in gergo si definiscono in forma proposizionale e questi modelli presuppongono che vi sia una indipendenza e identica distribuzione delle istanze. Questo vuol dire che tutti i dati in forma tabellare utilizzati per i modelli ML fino ad ora siano considerati indipendenti e che quindi non esistano relazioni tra gli elementi del dataset e che questi dati siano identicamente distribuiti.

La trasformazione di dati non strutturati in dati strutturati in forma tabellare introduce una distorsione, in quanto ad esempio molto spesso gli oggetti non sono indipendenti tra di loro.

Quindi quando si lavora con un DB si deve eseguire una operazione di proposizionalizzazione o di flattening che, attraverso delle Join si riconduce le varie istanze ad un'unica tabella. Tuttavia questa operazione produce un dataset distorto con delle capacità predittive minori rispetto ad un dataset contenente le relazioni tra gli oggetti che lo compongono.

GESTIONE DELLE PROBLEMATICE DEI DATI

Le principali problematiche sui dati in forma strutturata sono:

1- Dati mancanti: quali sono le tipologie di dati mancanti e quali sono le strategie che si possono adottare per il dato mancante
2- Dimensionalità: tecniche per la riduzione dello spazio di input
3- Dati sbilanciati

DATA PRE-PROCESSING

Nel mondo reale i dati di cui dobbiamo fare uso sono sempre rumorosi, incompleti e inconsistenti. Se i dati di questa natura non vengono processati prima di essere utilizzati si ottengono delle previsioni errate o comunque non precise.

I principali task che si possono eseguire in fase di pre-processing per far fronte a questo tipo di problemi sono:

- Data cleaning: tecniche che si occupano di dati tabellari risolvendo i problemi di mancanza dati, dati rumorosi, dati inconsistenti ed infine identificare la presenza di outliers ovvero degli elementi estranei agli elementi normalmente presenti nella tabella
- Data integration: tecniche utilizzate per integrare i dati presenti su diverse tabelle o DB
- Data transformation: tecniche utilizzate per la normalizzazione o aggregazione di dati
- Data reduction: tecniche che si occupano della selezione o trasformazione degli attributi

MISSING VALUES

Iniziamo con la data cleaning definendo cosa si intende per valore mancante, ovvero una cella vuota all'interno di una rappresentazione proposizionale. Il motivo per il quale abbiamo dei dati mancanti è fondamentale per capire se è ragionevole rimpiazzarlo o meno; un dato può mancare se ad esempio non si era prevista la rilevazione di tale dato, oppure per un malfunzionamento nel sensore atto a recepire tale dato oppure ancora nel caso in cui il dato sia stato cancellato. I dati mancanti generano tre tipologie di problemi:

1- Perdita di efficienza: al aumentare dei dati mancanti minore sarà la solidità delle nostre previsioni
2- Complicazioni: non tutti gli algoritmi di analisi dei dati sanno gestire i dati mancanti
3- Bias: rimpiazzare dei dati in maniera inopportuna o rimpiazzarli nel caso in cui non debbano essere rimpiazzati introduciamo una distorsione nel dato

Abbiamo tre tipologie di dati mancanti a cui conseguono diverse strategie per rimpiazzare tali dati:

1- Missing Completely At Random (MCAR): quando la distribuzione di un esempio avente un valore mancante per un attributo non dipende né dai dati osservati né da altri dati mancanti

2- Missing at Random (MAR): quando la distribuzione di un esempio per cui osserviamo un dato mancante dipende dai dati osservati ma non dipende da altri dati mancanti
3- Not Missing at Random (NMAR): quando la distribuzione di un esempio per cui osserviamo un dato mancante dipende dai dati mancanti

Rispetto a queste tre tipologie di dati mancanti possiamo mettere in atto diverse strategie risolutive; possiamo agire generalmente in tre modi:

1- Ignorare le istanze/attributi con i dati mancanti: rimuoviamo dal dataset le istanze o gli attributi che contengono il dato mancante
2- Convertire i dati mancanti in dei nuovi valori: usare un valore speciale per identificare il valore mancante tramite ad esempio NA
3- Metodi imputazionali: metodi che vanno a sostituire il valore mancante con dei dati ottenuti facendo delle considerazioni sulla rimanente parte del dataset.

I metodi di imputazione possono essere applicati alle tipologie di dati mancanti che rientrano in MCAR e MAR, mentre non sono adatte per NMAR.

MOST COMMON (MC) VALUE

Il primo metodo di imputazione è il Most Common Value (MC) che generalmente si applica facendo una media tra i valori possibili per tale attributo mancante se questo è continuo, se invece è discreto tale dato viene rimpiazzato con il valore più frequente (la moda matematica). Questa strategia non è applicabile se avessimo degli intervalli di valori. Per poter applicare questa tecnica si assume che ogni attributo sia generato da una distribuzione normale. Questa tecnica nel ML si presta maggiormente nel caso di un tipo di apprendimento non supervisionato.

CONCEPT MOST COMMON (CMC) VALUE

Una seconda tecnica adottabile si chiama Concept Most Common Value (CMC). Si può definire come una versione migliorata di MC; infatti va a rimpiazzare il valore mancante con media o valore più frequente utilizzando solo le istanze appartenenti alla stessa classe. Per applicare questa tecnica si assume che la distribuzione per un attributo di tutte le istanze di una stessa classe sia normale. Questa tecnica si presta all'utilizzo all'interno dell'apprendimento supervisionato del ML in quanto fa riferimento alla classe di un attributo per determinarne il dato mancante.

K-NEAREST NEIGHBOUR IMPUTATION

Un'altra tecnica di imputazione più raffinata delle precedenti è il K-Nearest Neighbour. Questa tecnica si basa sul KNN del ML che va ad analizzare le K istanze più vicine all'istanza in oggetto per determinarne la classe. Allo stesso modo la tecnica di imputazione va a utilizzare la media o il valore più frequente delle K istanze vicine per identificare quale valore sostituire all'interno del dato mancante.

NOISY DATA

Molto spesso i dati che abbiamo a disposizione sono rumorosi e per gestire questa problematica esistono delle tecniche di smoothing/binning o di discretizzazione che si occupano per prima cosa di ordinare i dati e li divide in dei bin che sono dei contenitori. Una volta fatto ciò si passa alla fase di smoothing dove utilizzando la media, mediana, limiti dell'intervallo si raffinano i dati.

Una prima strategia è quindi quella della semplice discretizzazione come tecnica di binning. Le tecniche adoperate in questo tipo di approccio sono le Equal-width e Equal-depth partitioning. La prima tecnica divide l'insieme dei valori in N intervalli di uguale dimensione, tuttavia è soggetta a problematiche derivate da outlier che sballano il confine tra un intervallo e l'altro. La seconda tecnica suddivide l'insieme in N intervalli contenenti lo stesso numero di campioni.

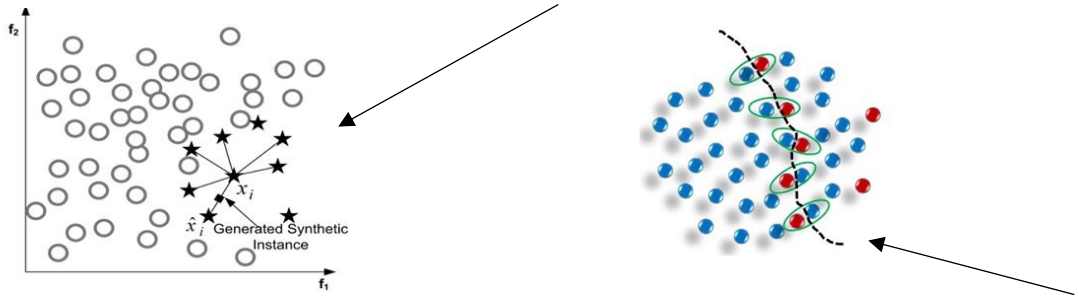
UNBALANCED DATA

Quando ci si trova in un contesto in cui non abbiamo una distribuzione delle classi bilanciata la maggior parte dei modelli ML si trova in seria difficoltà. Quando abbiamo due classi o più l'approccio generalmente applicato per bilanciare nuovamente il dataset si occupa di andare a bilanciare la componente di training e non quella di test in quanto lo sbilanciamento in fase di test serve per verificare l'efficacia del modello creato. Ci sono due tecniche principali di bilanciamento dei dati:

1- Oversampling: aggiunge delle istanze alle classi di minoranza per bilanciare il dataset
2- Undersampling: rimuove delle istanze dalle classi di maggioranza fino a bilanciare il dataset

I principali metodi di bilanciamento del dataset sono:

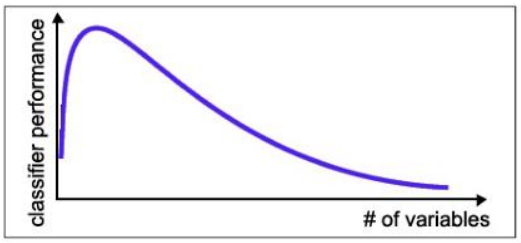
- Metodi baseline: sfrutta il random over-sampling e il random under-sampling, quindi va a selezionare casualmente delle istanze da aggiungere o rimuovere per bilanciare il dataset
- Metodi di over-sampling: tra le varie tecniche quella più famosa è quella di Smote. Il Synthetic Minority Oversampling Technique (SMOTE) si occupa di aggiungere sinteticamente delle istanze di minoranza andando a selezionare una istanza di minoranza, successivamente si scelgono i K elementi più vicini e si crea una nuova istanza della classe di minoranza ottenuta da un calcolo della media tra l'istanza e un suo vicino



- Metodi di under-sampling: tra le varie tecniche quella più famosa è quella di Tomek links. Un tomek link è una coppia di due istanze di due classi diverse (minoranza e maggioranza) dove non esiste un altro esempio nel dataset tale per cui la sua distanza dalla istanza di minoranza sia minore. In sostanza stiamo andando a creare le coppie di istanze di minoranza e maggioranza che sono al limite tra le due possibili classificazioni, ovvero quelle più difficili su cui fare una classificazione. Queste coppie vanno a definire la frontiera di classe e la tecnica di under-sampling va a rimuovere le istanze della classe di maggioranza appartenenti a questa frontiera
- Combinazione di metodi over e under sampling

FEATURE REDUCTION

Nello sviluppo di modelli di ML spesso si fa uso della riduzione dello spazio di input, in quanto avere tante feature non sempre implica un miglior potere predittivo. Nella gestione dello spazio di input tipicamente abbiamo un numero di istanze su cui fare apprendimento che è predeterminato e tendenzialmente si ha un incremento delle performance predittive all'aumentare del numero di variabili solo fino ad un certo punto, poi diventa inutile avere altre istanze. Questo concetto viene chiamato Curse of Dimensionality che individua un numero ottimale di variabili per ottenere le performance massime oltre il quale si inizia a perdere efficienza.



Nei modelli standard si utilizzano generalmente due tecniche che sono la feature selection e la feature extraction. Le prime vanno a selezionare un sottoinsieme delle feature esistenti senza fare alcuna trasformazione dei dati, mentre le seconde trasformano le feature esistenti in delle feature appartenenti ad uno spazio dimensionale inferiore.

Le tecniche di feature selection partono da un insieme di attributi e ne determinano una funzione di mappatura tale per cui il nuovo spazio degli attributi mantenga una più elevata quantità di informazioni rispetto al dominio originale.

FEATURE REDUCTION - FILTRI

Tra le varie tecniche di feature selection abbiamo i filtri; questi sono costituiti da due elementi principali ovvero una strategia di ricerca che si occupa di individuare un sottoinsieme di attributi candidati e la funzione che valuta la qualità dei candidati. Per la strategia di ricerca o andiamo a fare una ricerca esaustiva che comporta il valutare tutte le possibili combinazioni di tutti gli attributi a disposizione oppure si possono adottare delle tecniche euristiche.

Tra i filtri la tecnica più adottata è quella della Variable Ranking, ovvero ordinamento delle variabili che parte da un insieme di attributi e a prescindere dalla strategia di ricerca si occupa di associare un coefficiente a ciascuno degli attributi, ordina questi attributi in base al valore di questo coefficiente e va a selezionare gli attributi o rispetto ad un criterio di soglia oppure scegliere le prime N feature. Per poter stilare una classifica di quali siano gli attributi che ci forniscono il maggior numero di informazioni si fa spesso uso dell'information gain che misura la quantità di informazione guadagnata se si conosce il valore di una determinata feature, assumendo che le feature contribuiscano in termini di rilevanza in maniera indipendente dalle altre feature.

Un'altra tecnica di ranking fa uso della Correlation-based Feature Selection (CFS) che va a considerare non più la rilevanza degli attributi indipendentemente, ma valuta la bontà di un sottoinsieme di attributi rispetto ad una specifica classe e al contempo il fatto che questi attributi siano inversamente correlati ad altre classi (es. attributi fortemente correlati a classe A ma non correlati a classe B e C).

FEATURE REDUCTION - WRAPPERS

L'altra grande famiglia di tecniche di feature selection è quella dei wrapper che vanno a selezionare un sottoinsieme di attributi che sia particolarmente significativo rispetto ad un algoritmo di apprendimento specifico; quindi usa un algoritmo di apprendimento per valutare quanto è stata buona la selezione degli attributi. Il processo di selezione all'interno di un wrapper

parte da un sottoinsieme di attributi e si induce un training dell'algoritmo di ML sulla base di questo sottoinsieme e si valuta la capacità predittiva di questo sottoinsieme e se il modello risultante soddisfa i nostri criteri allora lo si seleziona, altrimenti si prova con un nuovo sottoinsieme. Per evitare il problema di eseguire una ricerca esaustiva andando a valutare ogni possibile sottoinsieme si fa uso delle euristiche che adottano due tecniche:

- Forward selection: inizia da un insieme vuoto degli attributi e ne aggiunge uno alla volta
- Backward selection: inizia dall'intero insieme degli attributi e ne toglie man mano

NETWORK ANALYSIS

Il dato strutturato in forma tabellare è il tipo di dato meno frequente che troviamo nel mondo reale in quanto i dati generalmente sono in relazione tra di loro e quindi tipicamente vengono rappresentati con delle strutture che manifestano delle relazioni. Per rappresentare queste relazioni in generale si fa uso delle reti basate sui grafi. All'interno delle reti vi sono diversi elementi di base che le compongono:

- Nodi: rappresentano un insieme generale di entità che hanno delle proprietà e che in qualche modo sono in relazione tra loro. I nodi possono avere degli attributi che ne descrivono delle caratteristiche e in quel caso il grafo prende il nome di attributed graph
- Archi: rappresentano i collegamenti fra i vari nodi e possono essere di varia natura come ad esempio possono essere dei collegamenti reali (ponti che collegano isole a terra ferma) possono essere dinamici (gli spazi aerei) oppure astratti. Gli archi possono essere diretti o non diretti; i primi sono gli archi che hanno una direzione, mentre i secondi o non hanno direzione oppure sono bidirezionali. Gli archi possono essere anche caratterizzati da degli attributi come ad esempio dei pesi, ranking o tipi. Possiamo avere anche dei multiarchi ovvero molteplici archi che collegano gli stessi nodi.

Ci sono diversi tipi di reti che differiscono tra loro per diverse caratteristiche:

- Connettività: un grafo si dice connesso se ha solo una singola componente oppure si dice connesso se esistono diverse componenti disgiunte
- Forma: i grafi possono avere la forma ad albero dove non vi sono delle connessioni cicliche tra i nodi, oppure possiamo avere dei grafi contenenti dei cicli che ci fanno percorrere la stessa strada più volte oppure ancora possiamo avere una struttura a stella

I grafi possono essere rappresentati in diversi modi che sono utili in diversi scenari:

- Matrice di adiacenza: va a costruire una matrice di dimensione NxN dove N sono i nodi, inserendo 1 se esiste un arco che collega due nodi
- Edge list: elenca tutte le possibili coppie di nodi
- Liste di adiacenza: elenca tutti gli elementi adiacenti ad un nodo andando a rappresentare il "vicinato" di un nodo

PROPRIETA STRUTTURALI DELLE RETI

Nel caso di reti di piccola dimensione consultare un grafico in maniera diretta risulta essere molto utile, ma quando si trattano reti di grandi dimensioni l'osservazione del grafico non ci porta ad alcuna conclusione soddisfacente. Per far fronte a questo problema ci sono diversi gruppi di misurazioni quantitative in grado di descrivere e confrontare le reti come ad esempio:

- Distribuzione di grado
- Diametro nel Clustering
- Misure di centralità dei nodi

PROPRIETA STRUTTURALI DELLE RETI – NODE DEGREE

La prima statistica descrittiva di un grafo riguarda il grado dei nodi che identifica il grado di incidenza degli archi su un determinato nodo preso in considerazione; nel caso di grafi non diretti abbiamo un grado unico in quanto non abbiamo direzionalità nell'arco ma nel caso di grafi diretti bisogna distinguere due concetti:

- Outdegree: rappresenta il totale degli archi uscenti dal nodo verso gli altri. Graficamente andiamo a prendere la riga della matrice di adiacenza e vediamo quanti 1 ci sono
- Indegree: rappresenta il totale degli archi entranti (incidenza) su un determinato nodo. Analogamente graficamente andiamo a prendere la colonna della matrice di adiacenza e vediamo quanti 1 ci sono

Il grado totale di un nodo all'interno di un grafo diretto è dato dalla somma di indegree e outdegree.

Oltre al grado associabile a ciascun nodo abbiamo anche un average degree che identifica il grado medio all'interno di un grafo; quindi nelle reti non dirette avremo un unico average degree, mentre in quelle dirette avremo l'averege outdegree e average indegree.

PROPRIETA STRUTTURALI DELLE RETI – DEGREE DISTRIBUTION

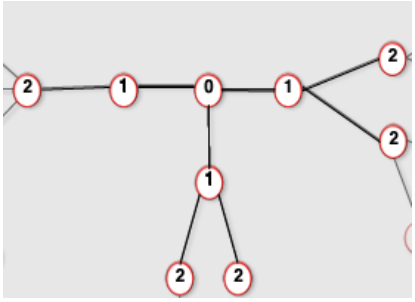
La distribuzione di grado misura la probabilità che scegliendo casualmente un nodo all'interno di un grafo questo abbia grado K, questo dato può essere utile per individuare dei fenomeni della rete come ad esempio la robustezza.

In generale non basta solo guardare l'average degree ma bisogna rapprotarlo alla distribuzione di grado per non trarre conclusioni errate. Si può dire che l'average degree individua il numero di link medi che possiedono i nodi, mentre con la distribution vediamo quanti nodi sul totale dei nodi disponibili hanno un certo numero di link (quindi magari vediamo che la maggior parte dei nodi ha pochi link e ci sono dei nodi con tantissimi link che identificano degli hub)

PROPRIETA STRUTTURALI DELLE RETI – DIAMETRO E AVERAGE PATH LENGHT

Per definire altre statistiche descrittive di una rete è necessario introdurre il concetto di percorso e distanza.

Il percorso è una sequenza di nodi che si può attraversare in cui ogni nodo è adiacente al successivo.



La distanza tra due nodi all'interno di un grafo è definita come il numero di archi che collega due nodi; se due nodi sono disconnessi la distanza è pari ad infinito. Per individuare la distanza all'interno di un grafo si usa la strategia di visita BFS che parte dal nodo di partenza 0 e identifica i nodi ad esso adiacenti con il valore 1 poi reitera il processo e identifica con altri valori gli altri nodi fino a finirli.

Possiamo quindi ora definire i concetti di diametro e average path lenght. Il diametro è la distanza massima tra qualunque coppia di nodi (considerando sempre lo shortest path tra questi) presente nel grafo. L'average path lenght per un grafo diretto è la misura media degli shortest path che collegano tutte le coppie di nodi di una rete.

Ritornando al discorso di connettività possiamo dire che i grafi con connessioni dirette sono fortemente connessi dove ogni coppia di nodi è collegata a qualunque altra coppia di nodi e viceversa, mentre debolmente connessi se togliamo la direzionalità dei collegamenti (ovvero basta che siano collegati).

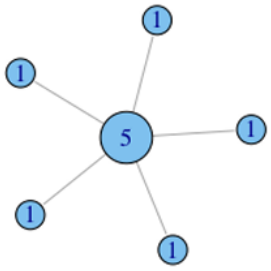
Tra i vari indicatori legati al comportamento dei grafi abbiamo il coefficiente di clustering, questa misura il grado per cui i vicini di un nodo scelto si collegano tra di loro. Il coefficiente di clustering ha un valore che varia tra 0 e 1, dove se si ha un nodo collegato ai suoi vicini ma questi non sono collegati tra di loro allora si ha coefficiente 0, se invece tutti i vicini sono collegati tra loro si ha coefficiente pari ad 1. Il coefficiente di clustering di un grafo individua la densità locale di un grafo; tanto più è elevato il numero di interconnessioni rispetto ai nodi di un grafo tanto più elevato sarà il suo coefficiente di clustering. Il valore del coefficiente di clustering di un intero grafo è pari alla media di tutti i coefficienti di clustering dei nodi del grafo stesso. Nella pratica possiamo dire che il coefficiente di clustering individua la probabilità che presi a caso due nodi vicini di un nodo principale del grafo questi siano collegati tra loro.

NETWORK CENTRALITY E MISURE DI CENTRALITA'

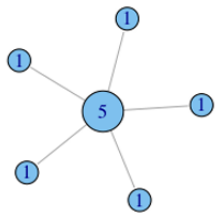
All'interno dei grafi sono presenti delle centralità, ovvero degli elementi che possono essere più significativi di altri in vari aspetti. Le misure presentate in precedenza vanno ad analizzare un comportamento presente in tutto il grafo, ma all'interno delle reti sono presenti dei nodi più rilevanti di altri e questa rilevanza può essere misurata con metriche diverse. Le quattro misure di centralità oggetto del corso sono: in-degree, out-degree, betweeness e closeness.

CENTRALITA' E CENTRALIZZAZIONE

Il concetto più semplice di centralità è la centralità di grado che è data dalla sommatoria degli archi incidenti per un determinato nodo. La centralità di grado mette in risalto l'influenza che un nodo ha sui suoi vicini. È possibile anche andare a calcolare la centralità di grado normalizzata andando a dividere ciascun grado del nodo (il numero di archi che si collegano al nodo) rispetto al valore massimo possibile stimabile di grado all'interno di una rete (N-1).

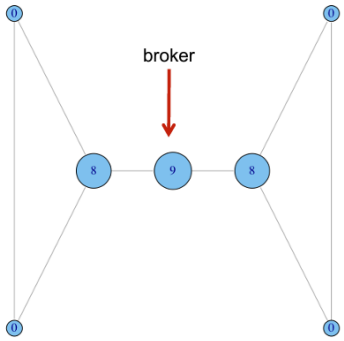


Oltre alla centralità di grado legata ad un singolo nodo è utile introdurre il concetto di centralizzazione che misura quanta variabilità c'è tra la centralità dei nodi presenti all'interno della rete. Per misurare la centralizzazione si utilizza la centralizzazione di Freeman che calcola la differenza tra il valore massimo di centralizzazione della rete e la centralità di tutti i nodi, dividendo il risultato per la più grande centralità possibile nella rete meno la seconda più grande centralità possibile nella rete. In altre parole possiamo dire che la centralizzazione sia la misura di quanto centrale è il nodo più centrale tra i nodi di una rete rapportato a quanto centrali sono gli altri nodi della rete stessa.



$C_D = 1.0$

Non sempre la centralità di grado ci fornisce informazioni esaustive, quindi ci sono ulteriori misure di centralità come ad esempio la centralità betweeness. Questa misura di centralità evidenzia la capacità di un nodo di svolgere la funzione di ponte tra due diverse parti della rete. Per misurare questa centralità si calcola quante coppie di nodi sarebbe necessario attraversare per raggiungere un qualunque altro nodo nella rete con il numero minore di salti possibili (shortest path). La betweeness quindi va a contare il numero di volte che un nodo svolge la funzione di ponte sullo shortest path che collega due nodi della rete. Questa misura può essere eventualmente normalizzata dividendo per il numero di coppie di nodi possibili escludendo il nodo di cui si sta misurando la betweeness.



La terza misura di centralità che analizziamo è la closeness; questa metrica indica quanto un nodo è vicino al centro della rete o se è un nodo periferico. Da questa misura si può capire quanto velocemente (efficienza) un nodo può raggiungere gli altri nodi della rete. La misura calcola lo shortest path medio che esiste tra un nodo e tutti gli altri nodi presenti nel nostro grafo. Anche questa misura può essere normalizzata dividendo per N-1.

Oltre alle misure appena presentate si può andare a misurare la centralità tramite l'utilizzo di autovalori e autovettori; questi sono stati chiamati in causa in quanto è utile andare a calcolare l'apporto di rilevanza che un nodo trasmette ad altri nodi. Si assume che un nodo è importante se è linkato con altri nodi che sono importanti. Questa misura è importante perché se un nodo riceve molti link ma questi non sono provenienti da nodi importanti, tale nodo avrà un alto valore di coefficiente di centralità ma un basso valore di rilevanza calcolato dagli autovettori e autovalori.

PAGE RANK

Un famoso tipo di misura della centralità è pagerank centrality; questo algoritmo veniva utilizzato per calcolare il coefficiente di importanza di una pagina web in funzione di una query di ricerca. Pagerank si basa sull'assunzione che un arco che collega in maniera diretta due nodi sia una raccomandazione che un nodo i dà rispetto ad un altro nodo j, quindi se i due nodi si puntano a vicenda allora la probabilità che questi due nodi siano relazionati tra di loro è molto più alta rispetto alla probabilità che non siano collegati tra loro. Quindi l'importanza (coefficiente di pagerank) di un nodo è influenzata dall'importanza dei nodi che puntano verso di lui. I nodi che possiedono un alto coefficiente di pagerank avranno un valore di importanza di voto più alto quando andranno a referenziare (ovvero puntare con un arco) un altro nodo.

Il coefficiente quindi viene inizialmente calcolato come la somma di tutte le referenze che questo nodo riceve moltiplicato per il coefficiente di ranking dei nodi che referenziano e fratto la capacità di disperdere il coefficiente di ranking dei nodi che referenziano. In altre parole l'importanza di un nodo è data dai voti che riceve dagli altri nodi e dalla loro importanza.

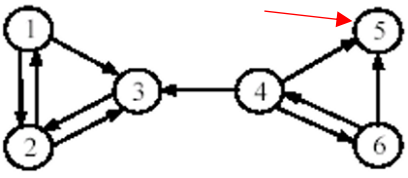
Il calcolo del pagerank può essere computato da un algoritmo iterativo e la sua soluzione corrisponde agli autovalori della matrice di adiacenza normalizzata. L'unica condizione che deve essere posta al fine di ottenere una convergenza è che 1 è il valore più alto di autovalore e che P sia il principale autovettore.

PAGE RANK – MARKOV CENTRALITY

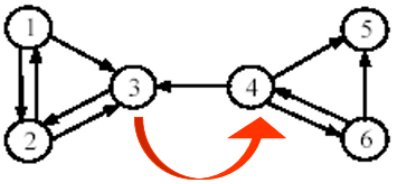
Sfruttando i processi Markoviani il processo di stima di importanza di un nodo viene visto come un problema di navigazione; in questo processo ogni nodo viene visto come uno stato del sistema e ogni arco rappresenta la transizione che permette ad un utente di passare da uno stato ad un altro, in sostanza senza andare a calcolare autovalori e autovettori i processi di markov simulano una navigazione del web per identificare quanto uno stato viene attraversato durante la navigazione degli utenti. Questo tipo di navigazione viene chiamato random surfing e si basa sulla probabilità che un utente ha di passare da un nodo di partenza a degli altri nodi. Nel definire questo processo di navigazione randomica si genera una matrice stazionaria che identifica lo stato in cui il sistema non si modifica più (quindi ho simulato tutti i possibili salti che un utente può fare durante la navigazione) e per garantire il raggiungimento di tale risultato convergente è necessario che si garantisca:

- Stocasticità della matrice
- Irriducibilità della matrice
- Aperiodicità della matrice

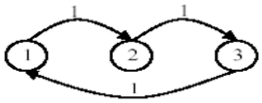
Nei grafi del web tuttavia tali condizioni non sono sempre verificate.



Se la matrice non è stocastica vuol dire che abbiamo pagine che vengono referenziate da altre pagine ma che a loro volta non referenziano nessuna altra pagina. Per risolvere tale problema o si rimuove il nodo o si aggiunge un coefficiente piccolo che collega tale nodo a tutti gli altri.



Se la matrice non è irriducibile vuol dire che non esiste per ogni coppia di nodi un percorso che permette di collegare i due nodi scelti a caso.



Se la matrice non è aperiodica vuol dire che esiste un ciclo che deve essere attraversato ogni volta che si vuole raggiungere un nodo.

Per risolvere entrambi questi due problemi (aperiodicitià e irriducibilità) si va ad aggiungere un arco da un nodo che va verso tutti gli altri nodi assumendo che ci sia una probabilità di transizione molto piccola.

In conclusione la parte fondamentale di tutte queste misure messe a disposizione per misurare la centralità riguarda il saper interpretare le misure e sapere anche quando utilizzare in base al problema che ci stiamo ponendo. Riassumendo si ha che:

- Centralità di grado: misura che identifica quanti nodi un certo nodo riesce a raggiungere in maniera diretta
- Betweeness: misura quanto è probabile che un nodo sia nel percorso principale utilizzato da un qualsiasi nodo della rete per raggiungerne un altro
- Closeness: misura la capacità in termini di velocità che un nodo ha di raggiungere un qualsiasi altro nodo della rete
- Autovalori/autovettori: misura quanto un nodo sia ben collegato a tutti gli altri nodi della rete

RECIPROCITA' E DENSITA'

Oltre alle misure di centralità e centralizzazione di una rete esistono altre misure che possono descrivere le caratteristiche di una rete in modo descrittivo, tra cui la reciprocità e la densità.

La reciprocità identifica il numero di relazioni reciproche che ho all'interno di un grafo. Matematicamente è il rapporto tra il numero totale di archi reciproci che si possono osservare in una rete rispetto al numero totale di archi della stessa rete. Il coefficiente di reciprocità ci dà informazioni su quanto bene una informazione circola dentro la rete in quanto maggiore è il numero di archi reciproci, minore è la probabilità di interrompere il flusso informativo della rete.

La densità di una rete è data dal rapporto del numero di archi presenti all'interno di una rete su il numero massimo possibile di archi presenti all'interno di una rete di stesse dimensioni. Nel caso di grafi non diretti il numero massimo di archi possibili è quello che permette di collegare ogni nodo con un altro nodo della rete, mentre per i grafi diretti avendo la direzionalità questo numero raddoppia perché ci deve essere anche la reciprocità. Con questa misura possiamo capire quanto ben connessa è la nostra rete in generale e una rete perfettamente connessa viene chiamata clique.

COMMUNITY DETECTION

Quando si parla di community spesso si intende l'aspetto sociale legato ad un gruppo di persone; in generale si può dire che una community è un insieme di nodi che ha delle relazioni forti tra i nodi e che questi siano legati e interagiscano tra di loro frequentemente. Uno dei task più importanti che si fanno quando si studiano i grafi è quello di identificare le community in quanto i nodi membri di una community in generale interagiscono fortemente tra di loro e hanno delle relazioni forti. Individuare una community ci consente di inferire caratteristiche dei nodi appartenenti ad una community come ad esempio:

- Condividere la funzionalità: i nodi di un gruppo sono legati ad un certo aspetto funzionale
- Studiare interazioni tra gruppi/community
- Inferire il valore di eventuali nodi mancanti appartenenti allo stesso gruppo
- Prevedere connessioni non osservate

Definire il concetto di community all'interno di una rete può essere soggettivo ma in genere si identificano community in dei gruppi di nodi interconnessi tra di loro, oppure dei nodi densamente collegati tra loro. Possiamo distinguere in generale anche le community disgiunte o overlapping nel caso in cui vi siano dei nodi in comune appartenenti a due o più community differenti.

Vi sono diversi algoritmi in grado di andare ad individuare le community e questi possono essere distinti in 4 famiglie di approcci:

1. Community Node-Centric: ogni nodo all'interno di un gruppo deve soddisfare una determinata proprietà
2. Community Group-Centric: si considerano le connessioni all'interno di un gruppo e tale gruppo deve soddisfare una determinata proprietà (quindi non necessariamente ogni nodo deve soddisfarla)

3. Community Network-Centric: partiziona la rete in dei set disgiunti
4. Community Hierarchy-Centric: costruisce una organizzazione gerarchica delle community

NODE CENTRIC COMMUNITY

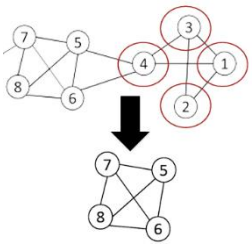
Negli approcci Node-Centric le possibili proprietà che i nodi devono soddisfare sono:

- Completa mutualità
- Raggiungibilità dei nodi
- Garanzia del grado dei nodi

Quando si parla di completa mutualità si osservano le clique, ovvero un sottogruppo composto da tre o più nodi i quali sono tutti adiacenti tra loro. Per identificare le community basate su clique ci sono due approcci:

1. Cercare la clique massimale ovvero la clique che ha il più grande numero di vertici
2. Cercare tutte le clique massimali ovvero tutte quelle clique che non fanno parte di una clique più grande

Entrambi questi approcci se calcolati sarebbero quasi impossibili da processare in quanto sarebbe necessario eseguire troppe verifiche, quindi si può utilizzare un processo a forza bruta che riduce la complessità del problema introducendo un pruning che consiste nell'andare a cercare la clique di dimensioni maggiori o uguali a K. Settando quindi un valore K si rimuove dalla ricerca tutti i nodi con archi minori o uguali di K e, inoltre, se dispongono di archi, questi saranno eliminati andando quindi a modificare il numero di archi di altri nodi rendendoli quindi potenzialmente di grado inferiore a K. Nell'immagine abbiamo un esempio dove K=3.



Per quanto riguarda la proprietà di raggiungibilità possiamo dire che tale proprietà è rispettata se un qualunque nodo di un gruppo è raggiungibile in K salti. In generale conoscere la raggiungibilità è utile quando si ha un vincolo di tempo dato per raggiungere un determinato nodo. La prima soluzione adottata per definire la raggiungibilità di un gruppo è la k-clique; questa tecnica identifica un sottografo massimale la cui massima distanza geodesica tra coppie di nodi è minore o uguale di K. La seconda tecnica utilizzabile è il k-club, ovvero una sottostruttura di diametro minore o uguale a K.

Si può concludere che tutte le tecniche appartenenti alle community Node-Centric sono generalmente utilizzate per reti di piccole dimensioni in quanto al crescere della dimensione del grafo tendenzialmente i nodi al suo interno saranno più sparsi rendendo così più difficile l'identificazione delle clique.

GROUP CENTRIC COMMUNITY

Se le tecniche di community detection node-centric si concentrano sull'individuare community tramite l'analisi dei singoli nodi, le strategie group-centric sono meno restrittive e vanno ad individuare delle community che rispettano una determinata proprietà a livello di gruppo e non più a livello di singolo nodo.

Un primo approccio group centric è quello di andare ad identificare dei sottografi chiamati quasi-clique; queste strutture pongono un determinato valore di soglia che se viene superato da un gruppo di nodi li considera delle quasi-clique. Se la densità interna al gruppo è maggiore di una

certa soglia allora tale gruppo viene considerato una quasi-clique. Andare a fare il calcolo esatto della densità risulterebbe una operazione troppo onerosa, quindi sono introdotti degli algoritmi greedy per individuare la massima quasi-clique; in generale si parte dal nodo di grado più alto e ne si verifica l'intorno di nodi che hanno grado maggiore che potenzialmente possono andare a comporre la massima quasi-clique e così via scendendo di grado.

NETWORK CENTRIC COMMUNITY

Gli algoritmi di community detection definiti come network-centric si pongono come obiettivo principale quello di essere meno stringenti a livello di proprietà da soddisfare da parte dei nodi di un sottografo e conseguentemente sono anche degli approcci meno onerosi.

Prima di parlare degli algoritmi è utile introdurre il concetto di equivalenza strutturale. Due nodi vengono definiti strutturalmente equivalenti se sono connessi allo stesso insieme di vicini. Impostare un algoritmo di ricerca per individuare questa caratteristica è complicato in quanto l'equivalenza strutturale è un concetto molto restrittivo in quanto difficilmente si incontra all'interno di reti sparse. Per far fronte a ciò si approssima l'equivalenza strutturale utilizzando il concetto di similarità vettoriale calcolandola dalla matrice di adiacenza; con questo approccio la similarità dei nodi può essere misurata in due modi:

- Cosine Similarity: particolarmente utile per i grafi pesati e non eccessivamente sparsi
- Jaccard Similarity: particolarmente utile per i grafi non pesati con rappresentazione booleana

Ora che è stata definita l'equivalenza strutturale si può passare con l'analisi degli algoritmi utilizzati per individuare i gruppi di nodi strutturalmente simili (community). Analizzando la matrice di adiacenza e calcolando la similarità con Cosine o Jaccard si potrà quindi applicare l'algoritmo K-Means Clustering per identificare le community. Il funzionamento del K-Means è lo stesso del ML quindi si parte da un insieme di K punti definiti centroidi e iterativamente si calcola l'appartenenza di tutti i nodi ai centroidi più vicini per poi ricalcolare la posizione dei centroidi. Questo tipo di approccio va a definire delle community egocentric, ovvero delle community basate sulle caratteristiche di un nodo e ricercando le stesse nei nodi vicini.

Nel caso in cui non si volesse individuare delle community egocentriche vi sono le tecniche di spectral clustering che non utilizzano più la matrice di adiacenza come punto di partenza ma utilizzano invece una rappresentazione del grafo di similarità. Successivamente utilizzando gli autovalori e autovettori determinano le componenti sottostanti fondamentali dai quali sarà possibile identificare le community che ad un livello superiore non sarebbero neanche visibili. In generale quindi lo spectral clustering sfrutta la rappresentazione dei dati in uno spazio dimensionale più piccolo di quello di partenza all'interno del quale è più semplice individuare le community. L'approccio basato su clustering spettrale quindi si articola in tre fasi:

1. Pre-processing: si costruisce la matrice che rappresenta il dataset
2. Decomposition: si computano autovalori e autovettori da cui definire uno spazio di rappresentazione di dimensione minore
3. Grouping: si applicano gli algoritmi di partizionamento su questo sottoinsieme più facile da gestire

Un altro approccio per individuare le community tramite approccio network centric è quello che si basa sulla massimizzazione della modularità che va a misurare quanto i gruppi di un grafo si distanziano e quindi individuare dei possibili grafi generati in modo random. Tanto più un grafo o una componente del grafo si allontana da un equivalente generato in modo random (è difficile che un grafo reale sia simile ad un grafico generato a random) tanto più sono sicuro che quelle relazioni siano forti e quindi effettivamente debbano appartenere alla stessa community.

La modularità quindi va a misurare l'interazione di un gruppo rispetto alle connessioni che si creano in maniera casuale all'interno di un gruppo. Per misurare la quantità di connesioni random (randome expected connection) tra due nodi si moltiplica il grado dei due nodi e il risultato lo si divide per il doppio degli archi dell'intera rete; una volta individuato tale valore è possibile verificare quanto la matrice di adiacenza sia distante da questo valore per identificare la modularità. Quando si ha una differenza molto grande tra la matrice di adiacenza e la random expected connection si può dire che la relazione tra i due nodi presi in considerazione è molto forte, al contrario quando c'è vicinanza tra matrice di adiacenza e random expected connection allora la relazione che esiste tra le coppie di nodi diventa debole e, conseguentemente, tali nodi non saranno collocati nello stesso gruppo.

HIERARCHY CENTRIC COMMUNITY

L'ultima famiglia di algoritmi atti ad individuare le community sono quelli che si basano sulla creazione di una gerarchia all'interno della rete. L'obiettivo è quindi quello di creare una struttura gerarchica delle community basandosi su due principali tipologie di approcci: agglomerativi e divisivi. Gli approcci divisivi partono considerando il grafo nella sua totalità e poi lo dividono in porzioni che identificano le community, mentre gli approcci agglomerativi partono dai singoli nodi e li uniscono in base alla similarità. Il grafo ottenuto da questi approcci (dendogramma) risulta avere altezza variabile e l'altezza rappresenta una misura di distanza chiamata intra-cluster distance ovvero la distanza che intercorre tra gli elementi all'interno di un cluster. Quindi più siamo in basso più i nodi saranno vicini tra loro, più siamo in alto più saranno lontani. Ovviamente per reti molto grandi vi sarà un dendogramma molto alto ed individuare le community sarebbe computazionalmente oneroso, quindi si opta per una operazione di pruning (definendo una soglia di intra cluster distance) dell'albero ad una altezza designata. Dopo aver eseguito il taglio del dendogramma si possono identificare le community con i primi rami che partono dal taglio e tutti i relativi sottorami.

L'approccio divisivo è un approccio più efficiente rispetto a quello agglomerativo e si occupa di andare a dividere il grafo intero in dei sottografi via via composti da sempre meno nodi. Un diverso approccio divisivo è quello della edge-betweeness che identifica il numero di shortest path che esiste tra qualunque coppia di nodi che passa da un arco specifico (stesso discorso di node-betweeness ma per gli archi). In questo modo si identificano quegli archi che fungono da "ponte" tra una community e l'altra. Uno degli algoritmi che fa uso della edge-betweeness è quello di Girvan-Newman che opera con i seguenti step:

1. Misura l'edge-betweeness di tutti gli archi del grafo
2. Rimuove l'arco con il valore più altro di betweeness
3. Misura nuovamente la betweeness degli archi che sono stati influenzati dalla rimozione dell'arco del punto precedente
4. Itera i primi 3 step finchè tutti gli archi sono rimossi

VALUTAZIONI DI COMMUNITY DETECTION

In conclusione si può dire che non esiste un metodo ottimale per individuare le community, ma ogni approccio può essere più o meno utile in base ai vincoli del dominio e alle necessità che si ha. L'ultimo processo per concludere l'argomento riguarda la valutazione della qualità delle community ottenute tramite gli algoritmi. In genere si lavora o facendo uso della ground truth o non facendone uso.

Quando si vuole eseguire una valutazione delle qualità tramite ground truth, quindi abbiamo una conoscenza pregressa generale di come le community dovrebbero presentarsi, i passaggi da eseguire sono i seguenti:

- Generare la matrice di assegnazione degli argomenti (strutturata con veri positivi, falsi positivi, veri negativi e falsi negativi) per poter valutare quanto bene l'algoritmo abbia eseguito l'assegnazione
- Fare le misure di valutazione della qualità tipiche degli algoritmi di ML, ovvero Precision, Recall, F-measure

Per calcolare le misure di qualità delle community dobbiamo identificare:

- True Positive: oggetti simili che sono assegnati alla stessa community
- True Negative: oggetti diversi tra loro sono assegnati a community differenti
- False Negative: oggetti simili assegnati a community diverse
- False Positive: oggetti diversi assegnati alla stessa community

Quando NON abbiamo a disposizione una ground truth possiamo misurare:

- Silhouette media
- Distanza media inter-cluster/community
- Somiglianza media intra-cluster/community

A prescindere dal come calcolare questi valori è necessario considerare prima di fare i calcoli che cosa ci interessa sapere veramente; se ad esempio non ci interessa che le community siano distanti tra loro ma che internamente ci sia un alto grado di similarità non ha senso usare silhouette e inter-cluster distance, ma possiamo solo usare la intra-cluster distance.

ASSORTATIVITY AND DYNAMICS

Prima di parlare di assortatività bisogna partire da una domanda: gli hub all'interno della nostra rete si collegano con altri hub o no? Assumiamo che ogni nodo si possa legare in modo randomico con qualunque altro nodo della rete, in tal caso la probabilità che ciò accada è pari a: $k*k'/2E$ dove k e k' sono i due nodi e 2E è il doppio di tutti gli archi nella rete. Per identificare la discriminante che ci consente di identificare se un hub si possa collegare ad un altro hub o meno si deve fare uso del concetto di assortatività e disassortatività. In funzione della tipologia dei legami che si creano dentro una rete si può avere:

- Reti assortative: se gli hub si legano con altri hub
- Reti disassortative: se gli hub si legano con nodi di grado inferiore
- Reti neutrali: se i nodi si legano ad altri nodi seguendo un criterio randomico

Per poter misurare il livello di assortatività di una rete si fa uso della correlazione di grado:

- Se la correlazione di grado si sviluppa sulla diagonale principale, allora si può dire che i nodi con grado simile tendono a legarsi tra loro, quindi siamo di fronte ad una rete assortativa
- Se la correlazione di grado si sviluppa sulla antidiagonale, allora si può dire che i nodi con grado dissimile tendono a legarsi tra loro, quindi siamo di fronte ad una rete disassortativa
- Se la correlazione di grado è simmetrica, allora si può dire che i nodi si legano tra loro in modo randomico, quindi avremo una rete neutrale

Per definire quindi se una rete è assortativa/disassortativa/neutrale allora bisogna misurare la correlazione di grado. Per fare ciò si usa la matrice di correlazione di grado che individua la probabilità di trovare un nodo con grado i e j alla fine di un link scelto in maniera casuale. La matrice è quindi costruita contenendo gli elementi eij che rappresentano il numero di archi all'interno di una rete che collegano vertici di grado i con vertici di grado j. In sostanza faccio una matrice avente per righe e colonne i possibili gradi e vedo quanti nodi sono collegati tra di loro e che grado hanno.

CALCOLO RETE ASSORTATIVA/DISASSORTATIVA/NEUTRALE

La probabilità che esista un nodo di grado k alla fine di un arco preso a caso sia: qk= k*pk/<k> dove k è il grado, pk è la probabilità di grado e <k> è il grado medio. Sapendo ora sia come strutturare una matrice di correlazione e la probabilità appena enunciata possiamo fare dei confronti per identificare delle possibili distorsioni. All'interno di una rete che non presenta correlazioni di grado (rete neutrale) si avrà che la matrice di correlazione sarebbe equivalente a: ejk= qj * qk quindi che la matrice di correlazione sia uguale al prodotto delle probabilità dei due nodi qj e qk. Se la matrice di correlazione generata da questa equazione è diversa allora vuol dire che si sta manifestando una distorsione rispetto alla expected random connection ejk= qj * qk. Quindi per misurare se siamo di fronte ad una rete assortativa o disassortativa si misura la differenza che c'è tra la matrice di correlazione e la formula di expected random connection con la formula ejk – qj * qk avendo cosi:

- Se la differenza è positiva allora abbiamo una rete assortativa
- Se la differenza è negativa allora abbiamo una rete disassortativa
- Se la differenza è zero allora abbiamo una rete neutrale

È possibile normalizzare la formula andando a dividere per il valore massimo ottenibile, ovvero la deviazione standard ottenendo così un valore che viene definito R che ha valore compreso tra -1 e 1:

- Se R=0 allora abbiamo una rete neutrale
- Se R<0 allora abbiamo una rete disassortativa
- Se R>0 allora abbiamo una rete assortativa

In genere le reti che definiscono una relazione sociale tendono ad essere reti assortative, mentre le reti biologiche e tecnologiche tendenzialmente sono reti di tipo disassortativo.

GIANT COMPONENT NELLE RETI ASSORTATIVE E DISASSORTATIVE

Individuare l'assortatività ci fornisce informazioni per quanto riguarda la giant component, infatti il grado di correlazione è strettamente legato al processo di creazione o distruzione della giant component.

Per quanto riguarda la creazione della giant component possiamo dire che:

- Nelle reti assortative la giant component emerge ad un istante temporale in cui la rete ha un grado medio pari ad 1 in quanto denota la presenza di hub di grado alto che cercano di fare da aggregatori
- Nelle reti disassortative la giant component emerge quando il grado medio è maggiore di 1 in quanto in questo tipo di rete gli hub si legano più facilmente agli spook e quindi fanno più fatica a creare la giant component.

Per quanto riguarda la distruzione della giant component possiamo dire che:

- Se abbiamo una rete assortativa e andiamo a rimuovere un hub questo causerebbe pochi danni alla struttura in quanto sono presenti diversi hub legati all'hub eliminato che sopperiscono alla sua mancanza
- Se abbiamo una rete disassortativa e andiamo a rimuovere un hub questo causrebbe molti danni in quanto gli hub tendono a legarsi con gli spook e togliendo l'hub toglieremmo il collegamento tra l'hub e tutti i nodi circostanti

Da questo possiamo andare a misurare il livello di resilienza delle reti andando a simulare la rimozione dei nodi o degli archi all'interno della rete. Se rimuoviamo i nodi abbiamo un site percolation, mentre invece se rimuoviamo gli archi abbiamo una bond percolation. Lo smembramento della giant component all'interno di una rete disassortativa richiederà più tempo rispetto a quello impiegato per smembrare una rete assortativa in quanto servirà togliere più nodi prima di riuscire ad avere un grado medio prossimo ad 1.

SOCIAL MEDIA ANALYTICS

L'analisi delle informazioni sotto forma di testo è la parte fondamentale di questo corso. Le principali piattaforme da cui possiamo reperire tali dati in forma testuale sono i social che vengono definiti in questo caso microblogs, ovvero delle piattaforme dove si condivide il proprio pensiero sotto forma di un breve testo. I fenomeni di social media analytics che prevedono l'analisi di linguaggio naturale sono sempre più importanti e utilizzati per diversi scopi online come ad esempio la brand reputation, la quality of life oppure più in generale per prevedere l'andamento del mercato.

Vedremo ora i vari passaggi necessari per andare a trasformare le informazioni espresse in forma testuale, non strutturata in conoscenza, in un elemento strutturato. L'obiettivo, quindi, sarà passare ad esempio da un tweet di un utente a delle informazioni che possono essere rappresentate in uno spazio strutturato. Alcune tra le varie informazioni che possono essere estratte dai tweet sono:

- Emozioni
- Polarità
- Entità a cui si riferisce il tweet
- Topic/argomento
- Lingua e linguaggio utilizzato

Le componenti base che verranno utilizzate per eseguire la social media analytics sono:

- Opinion holder: individuo che per mezzo di una piattaforma online esprime un proprio pensiero su un determinato oggetto

- Object: oggetto sul quale viene espressa una opinione
- Aspect: aspetti trattati quando si va ad esprimere un'opinione
- Opinion: opinione ed annessa emozione espressa riguardante all'intero messaggio
- Connessioni tra gli utenti: utenti collegati tra loro possono condividere le stesse opinioni

Quando si lavora con i social media dunque si fa un lavoro di Natural Language Processing (NLP). Questa tecnica è strutturata idealmente per gestire degli input ben formati che sono:

- Grammaticamente corretti
- Senza errori di scrittura
- Frasi in una sola lingua
- Frasi che utilizzano un linguaggio formale

Il problema è che quando si va ad analizzare i dati all'interno di un social media come FB o insta i parametri delle frasi sono completamente diversi dal concetto ideale appena esposto. I social media sono degli insiemi di tecnologie e strumenti della comunicazione volti a creare, scambiare e condividere su internet dei contenuti multimediali. Come possiamo facilmente intuire vi sono diverse difficoltà nell'applicare l'analisi dei dati posti all'interno di un social media rispetto ai dati con le caratteristiche sopra presentate. La difficoltà principale che si deve affrontare quando si esegue la NLP è come rappresentare le stringhe che identificano le parole e come possiamo distinguere i vari contesti in cui compaiono le parole. La composizione di un testo si basa sul principio di composizionalità, ovvero il principio tale per cui le frasi vengano composte in modo tale che il senso delle parole venga definito dal contesto della frase stessa. Un'altra notevole difficoltà che si incontra durante il processo di NLP è il rumore, ovvero dati non significativi. Infine un'ultima difficoltà del linguaggio naturale la identifichiamo nella ambiguità che può essere di diverso tipo:

- Ambiguità morfo-sintattica: ogni singola parola può appartenere ad una classe di parte del discorso diversa, ad esempio una parola che può essere sia un avverbio che un nome e bisogna capire quale dei due è. In questo caso grazie al Part-of-Speech tagging andiamo a dare un tag alle componenti della frase che identifica quale accezione di senso devono assumere.
- Ambiguità strutturale: la struttura della frase è ambigua sintatticamente, ovvero l'intera frase può assumere dei significati diversi in base all'interpretazione. Per risolvere questo problema si fa uso della Parse Tree Disambiguation, una tecnica che genera degli alberi che collegano ogni elemento della frase con una relazione di tipo sintattico generando una struttura di dipendenza sintattica il più verosimile possibile.
- Ambiguità semantica: ambiguità nel riconoscimento delle entità all'interno del testo e distinguere quali sono entità di interesse e quali no. Per risolvere questi due problemi si usano gli algoritmi di Named-Entity Recognition. Altri problemi sematici si possono riscontrare quando una entità può avere più significati (come Frozen il film o la canzone) e per disambiguare il significato di tali entità si usano le tecniche di Named-Entity Linking che associano le entità ad una descrizione presente all'interno di una base di conoscenza.
- Ambiguità emozionale: quando si lavora coi social capire la polarità non è semplice perché spesso non ci sono frasi esplicite di gradimento di un prodotto, infatti l'analisi di testo per individuare dei pareri non esplicitamente espressi viene chiamata Sentiment Analysis.

Sempre sul campo dell'emozione bisogna notare anche la necessità di disambiguare le espressioni ironiche che potrebbero indirizzare la polarità della frase in maniera sbagliata.

Nei prossimi due capitoli si vede nel dettaglio queste tecniche appena elencate partendo da quelle di sentiment analysis per poi arrivare all'entity linking e recognition.

SENTIMENT ANALYSIS E IRONY DETECTION

Uno degli obiettivi della analisi del linguaggio naturale nei documenti generati online dagli utenti dei social media è quello di riuscire ad andare ad interpretare il linguaggio naturale per identificare all'interno di tali documenti la presenza di messaggi soggettivi che possono denotare delle opinioni personali che associano una polarità positiva, negativa o neutrale all'intero contenuto del documento e, infine, comprendere quale sia il target verso cui è volto questo tipo di contenuto.

Per prima cosa andremo a trattare i modelli che ci consentono di affrontare il problema della ambiguità emozionale, ovvero i modelli che ci consentono di capire se:

- Un messaggio è oggettivo o soggettivo
- Quale sia la polarità del messaggio
- Riuscire a capire se un messaggio contiene dentro di sé delle emozioni

Prima di vedere le tecniche in sé dobbiamo dare alcune definizioni di base:

- Un testo oggettivo è un frammento testuale che riporta delle informazioni fattuali, mentre un testo soggettivo esprime delle credenze, delle opinioni.
- La polarità positiva e negativa sono dei messaggi di testo che contengono all'interno degli elementi atomici appartenenti a specifiche classi del lingauggio (verbi, avverbi, aggettivi) tipicamente positive o negative. Per la polarità neutrale invece abbiamo un messaggio misto che contiene parti di elementi tipicamente positivi e parti di elementi tipicamente negativi
- Un messaggio contiene una opinione esplicita se nel messaggio c'è una affermazione soggettiva che contiene al suo interno una polarità
- Un messaggio che contiene una opinione implicita se nel testo del messaggio ci sono affermazioni oggettive, ma che implicano una opinione
- Un testo ironico è una espressione comunicativa che esprime un parere opposto a quello che si è effettivamente scritto
- Le emozioni sono delle reazioni a degli stimoli che denotano un coinvolgimento personale riguardo tale stimolo; vi sono due modelli emozionali utilizzati per le tecniche di NLP ma noi useremo solo quello di Plutchik che è un modello gerarchico di emozioni riprestabile graficamente con un fiore di diversi colori

Il processo di sentiment analysis si occupa di andare ad estrapolare da un set di informazioni reperite dai social media tutti questi dati. In particolare il processo SMA (sentiment analysis) si articola nelle seguenti fasi:

1- Collezione dei dati
2- Rappresentazione degli stessi in modo tale che siano significativi per i modelli computazionali che li tratteranno
3- Indurre i modelli di classificazione che trovano la polarità, le emozioni e la ironia

COLLEZIONE DEI DATI

Per quanto riguarda la collezione dei dati nei social possiamo o collezionare i dati in tempo reale (on-line) oppure offline. Il primo metodo è molto utile ad esempio per la politica, per l'immissione di nuovi prodotti sul mercato, ecc. in quanto va a prendere i dati da oggi in poi, mentre il metodo offline va ad osservare i dati del passato che ci consentono di eseguire analisi di mercato a posteri ad esempio.

RAPPRESENTAZIONE DEI DATI – BAG OF WORDS

Il passo successivo riguarda l'andare ad individuare una rappresentazione che sia trattabile dal calcolatore e che sia rappresentato in maniera efficace il significato del testo. Il passaggio che si fa con i dati non strutturati è il passaggio da stringa a dato quantitativo che ci consente di misurare e di indurre i modelli di apprendimento.

La più semplice rappresentazione dei dati che possiamo avere è quella in cui si costruisce una matrice chiamata Document Term Matrix; all'interno di questa matrice vengono inserite nelle righe tutte le parole univoche presenti all'interno del testo andando così a formare il vocabolario comune, mentre nelle colonne vengono inserite tutte i messaggi che si stanno trattando. In sostanza nella matrice ci saranno degli 1 se nella frase viene menzionato un termine del vocabolario comune, altrimenti si mette 0. Questo tipo di rappresentazione viene chiamato bag of words e implica che ogni parola sia indipendente dalle altre cosa non vera nella realtà e quindi datata come tecnica.

Quando si lavora con il linguaggio naturale ci sono delle tecniche di pre-processing che si occupano della normalizzazione del linguaggio, ovvero si riconduce il linguaggio naturale ad una sua rappresentazione in forma canonica e si vanno a rimuovere tutti gli elementi atomici poco descrittivi. Alcune tecniche di pre-processing dei dati sono:

- Rimozione delle stopwords (articoli, congiunzioni, ecc.)
- Rimozione di numeri e punteggiatura
- Stemming ovvero prendere una parola e ricondurla alla sua radice (gatto e gattino in gatt)
- Lemming ovvero riportare la parola alla sua forma canonica (ragazzo e ragazzi riassunti in ragazzo)

Nei social spesso troviamo all'interno dei testi altri elementi importanti che dobbiamo identificare con precisione in quanto ci forniscono degli indicatori, come ad esempio:

- Parole allungate (belooooooo)
- Onomatopee (bleah)
- Slang (grz)
- Acronimi (LOL)
- Emoji (☺)

RAPPRESENTAZIONE DEI DATI – WORD2VEC

I nuovi modelli introdotti negli ultimi 10 anni vanno a sopperire alla mancanza di dipendenza delle parole presente nel modello bag of words usato in precedenza; questi modelli sono i modelli neurali che si basano su tecniche di deep learning. L'obiettivo dei modelli di questa tipologia come Word2Vec è quello di rappresentare una parola come un vettore che ne va a denotare il senso.

Il modello Word2Vec viene utilizzato per andare a creare delle rappresentazioni distribuite delle parole e va ad ovviare a due problemi della bag of words:

- La bag of words dà valore 0 quando non è presente un vocabolo del vocabolario all'interno di un messaggio ma questo rende sparsa la rappresentazione dei dati se abbiamo dei vocabolari grossi e conseguentemente tanti 0 per ogni frase
- Due parole dello stesso significato (auto e macchina) nel modello bag of words vengono rappresentate con due elementi distinti nel vocabolario quando in realtà non lo sono

La rappresentazione distribuita dei modelli come il Word2Vec ha due grandi varianti che vanno a sistemare i problemi appena elencati:

- Skip-gram: modello che dato un insieme di frasi va ad iterare sulle parole di ciascuna delle frasi a disposizione e cerca di utilizzare la parola corrente per prevedere quale sia la parola adiacente (sia la precedente che successiva)
- CBOW: modello che dato un insieme di frasi va ad iterare sulle parole di ciascuna delle frasi a disposizione e cerca di utilizzare le parole adiacenti per prevedere quale sia la parola che sta al centro

Il modello Skip-gram parte dalla rappresentazione di ogni singola parola con un vettore che si chiama One-hot-vector, ovvero un vettore composto da tutti 0 tranne un 1 che rappresenta la parola in corrispondenza del vocabolario. Questo vettore One-hot viene poi fornito in input ad una rete neurale a singolo strato che si occuperà di fornire una rappresentazione della parola contenuta nel vettore in funzione di tutte le parole che ci sono prima e che ci sono dopo; infatti come output da questa rete si otterrà un vettore che contiene per ogni parola del vocabolario la probabilità che prendendo in modo random una parola vicina questa sia una parola del vocabolario. Quindi è un vettore contenente le probabilità che una parola scelta a caso da quelle del vocabolario sia vicino alla parola che sto considerando.

Per fare ciò Word2Vec struttura una rete neurale collegata con un singolo layer nascosto e che ha una funzione di attivazione lineare. La dimensione del vettore di input è pari al numero di parole del vocabolario. La dimensione dello strato nascosto viene selezionata in base alla dimensione del vettore di partenza che è un multiplo di 128. La dimensione del layer di output è uguale alla dimensione del layer di input. Il layer di output rappresenta la probabilità che scegliendo a caso una parola vicino alla parola che stiamo codificando con la rete neurale questa sia una delle parole del nostro vocabolario. Quindi data una parola del vocabolario quale sia la probabilità che adiacente a lei ci sia una qualsiasi altra parola del vocabolario.

Ogni parola del vocabolario quando si applica Word2Vec viene data in pasto alla rete neurale che trasforma il vettore one-hot in un vettore denso.

PREDIZIONE DEL SENTIMENT

Una volta definito il modo di rappresentare i dati bisogna andare a vedere come effettivamente fare la previsione del sentiment e delle emozioni in generale. Ci sono 4 tipi di approcci per andare ad effettuare delle previsioni sul sentiment:

1- Approcci basati sui lessici

2- Approcci supervisionati
3- Approcci semi-supervisionati
4- Approcci non supervisionati

APPROCCI BASATI SUI LESSICI

La soluzione più semplice si basa sui metodi basati sui lessici, i quali per classificare come positivo, negativo o neutrale un testo si appoggiano a dei lessici generici o specifici che sono di fatto degli elenchi di parole di cui si conosce la polarità. Questi approcci quindi processano il testo e vanno ad individuare se ciascuna delle parole che compongono il testo appartiene ad un lessico di parole positive, negative o neutrali e se la maggioranza di parole appartiene ad un lessico positivo allora il testo intero sarà considerato positivo. Quando sono presenti in pari occorrenze termini appartenenti a lessici positivi e negativi allora il testo viene classificato come testo neutrale. Come già detto ci sono moltissimi lessici sia generici che specifici, ma soprattutto specifici perché il lessico ad esempio delle recensioni amazon sulle biciclette sarà estremamente diverso dal lessico delle recensioni dei film.

Oltre ai lessici è presente una risorsa lessicale chiamata Sentiwordnet, ovvero un database lessicale che contiene tutti i synset (i sensi delle parole) ognuno dei quali è annotato con un grado di positività, negatività e neutralità. Graficamente la polarità di una parola è rappresentata come un punto in un triangolo i cui vertici rappresentano il grado massimo di positività, negatività e neutralità. Sentiwordnet si basa su Wordnet che è una risorsa lessicale che contiene la definizione di verbi, avverbi, nomi e aggettivi rappresentanti lo stesso concetto che vengono raggruppati per sinonimia.

Dato che le parole possono assumere diversi significati ognuno dei quali con una diversa polarità, per andare a determinare all'interno di una frase quale sia la sua effettiva polarità ci sono due approcci principali:

1- Approccio a forza bruta: utilizzare la risorsa lessicale e andare per maggioranza riguardo le sue possibili polarità. Se una parola ha 7 accezioni di polarità positiva e 2 negativa allora la si considera per maggioranza positiva sempre a prescindere dal suo effettivo significato nel contesto. Ovviamente questo approccio è poco preciso.
2- ?

APPROCCI SUPERVISIONATI

Il metodo tradizionale per affrontare la sentiment analysis è quella di vederla come un approccio supervisionato. Questo tipo di approccio applica dopo una fase di collezione e pre-processing dei dati uno tra i vari modelli di ML studiati per andare a classificare correttamente emozioni e sentiment.

Quando si parla di linguaggio naturale sia per i modelli basati su lessici sia su apprendimento supervisionato prestare moltissima attenzione alle negazioni che sconvolgono la polarità di una frase che potenzialmente potrebbe essere positiva. In generale ci sono due approcci utilizzati per il riconoscimento delle negazioni: o si nega la polarità degli aggettivi oppure negare tutte le parole nell'intorno della negazione. L'approccio più semplice per far fronte alle negazioni prevede che venga aggiunto un NOT ad ogni parola che sta tra la negazione e il successivo primo segno di

punteggiatura, quindi una volta fatto ciò se ad esempio si lavora con i lessici quando si trova una parola della frase all'interno del vocabolario si faccia un reverse della polarità, mentre se abbiamo un modello basato su apprendimento supervisionato si cambia la rappresentazione della parola ovvero avremo la rappresentazione della parola originale e la sua rappresentazione anteceduta dal NOT.

Una volta fatta questa distinzione negli approcci supervisionati si passa all'applicazione del modello di apprendimento con training set e test set per poi andare ad ottenere un modello di classificazione adatto alle nostre esigenze.

APPROCCI SEMI-SUPERVISIONATI

Il vantaggio degli approcci semi supervisionati rispetto agli altri è quello che hanno bisogno di una quantità minima di dati etichettati (ovvero sapere la polarità di alcune parole) e sfruttando le informazioni di contesto si va ad arricchire le informazioni ottenute dai dati etichettati.

A tal proposito in generale si va ad arricchire i lessici rendendoli più specifici per il nostro caso d'uso tramite l'operazione di Bootstrap. Un semplice approccio per andare ad applicare il bootstrap è quello di andare ad accoppiare gli aggettivi che sappiamo avere una polarità positiva o negativa con un AND e cercare la loro presenza congiunta in un motore di ricerca come Wordnet. Ad esempio fair AND legitimate sono entrambi di polarità positiva ma se sapessimo solo che fair è positivo potremmo andare a supporre che anche legitimate lo sia perché accoppiato tramite AND ad un termine di polarità positiva come fair. Allo stesso modo, aggettivi congiunti con la congiunzione BUT identificano che la polarità dei due aggettivi sia opposta (fair BUT brutal). Una volta creati gli accoppiamenti in AND e BUT tramite la consultazione di una risorsa web si va ad applicare un algoritmo di clustering su grafo per distinguere la positività e negatività, in modo tale da andare ad allargare il lessico iniziale.

Un altro approccio legato all'apprendimento semi supervisionato è quello che si basa sul concetto di Pointwise Mutual Information. Questo approccio è costituito da tre passaggi fondamentali:

1- Estrarre delle phrasal lexical da un testo, ovvero una sottoporzione dell'elemento sentence costituita da specifici elementi appartenenti a specifiche parti del discorso
2- Stima della polarità delle phrasal lexical
3- Definire la polarità media delle frasi

Per eseguire una stima della polarità si definisce come parola positiva una parola che occorre molte volte con un'altra parola positiva definita seme che scegliamo noi (es. "excellent"), allo steso modo definiamo una parola di polarità negativa quella parola che occorre più volte con il seme negativo scelto da noi (es. "poor"). Per andare a misurare questo valore di co-occorrenza si fa uso della mutua informazione andando a misurare tra le parole della frase la loro pointwise mutual information (PMI) che va a misurare quanto un elemento x e y co-occorrono in una frase rapportandoli alla possibilità che questi possano essere indipendenti. In sostanza va a calcolare la probabilità che c'è di vedere due parole insieme nella stessa frase fratta la probabilità di vedere le due parole in due frasi diverse. Questo discorso si può applicare anche per i phrasal lexical che se messi insieme vanno a comporre la frase intera e perciò si dovrà calcolare il PMI del phrasal lexical rispetto a quanto è vicino alla parola seme positiva e negativa e poi fare una differenza tra i due risultati.

APPROCCI NON SUPERVISIONATI

Gli approcci non supervisionati sono i più complessi della lista per andare ad eseguire operazioni di sentiment analysis; questo tipo di apprendimento è utilizzato per l'aspect-based sentiment analysis, ovvero degli approcci in grado di individuare all'interno di un documento quali sono gli aspetti specifici percepiti positivamente e negativamente di un oggetto. Per rendere l'idea possiamo dire che in uno stato FB composto da più frasi riguardanti lo stesso argomento, possiamo avere delle porzioni di stato che esprimono un concetto di polarità positiva e porzioni che esprimono polarità negativa, quindi l'aspect-based sentiment analysis si occupa di andare ad identificare per ogni aspetto trattato in un documento quali sia il sentiment correlato e conseguentemente quale sia la sua polarità (es. aspetto batteria sentiment negativo, aspetto schermo sentiment positivo).

Per eseguire questo tipo analisi vi sono due modelli non supervisionati di tipo generativo basati sul concetto di Latent Dirichlet Allocation che sono:

1- Joint Sentiment Topic (JST)
2- Aspect Sentiment Unification Model (ASUM)

Il modello LDA è un modello generativo e della sua spiegazione non si capisce un cazzo lo trovi ad 1h05m di Sentiment analysis 2.

Il modello JST è una estensione del modello LDA in quanto va ad identificare oltre al topic, già identificato da LDA, anche il sentiment. Ad esempio quindi con la parola "CASA" verranno rilevati differenti topic tra cui arredamento, famiglia, edilizia ecc.. ai quali a loro volta viene assegnata una probabilità di polarità positiva/negativa. Il modello JST produce in output un elenco di topic con la relativa polarità; avremo quindi una serie di topic caratterizzati da parole con polarità positiva e una serie di topic caratterizzati da parole con polarità negativa.

Il modello ASUM è simile a JST e ha lo stesso compito, ma a differenza di JST assume che un documento sia costituito da M frasi, dove ogni frase può descrivere un aspetto. Questo modello quindi va a raffinare il modello di JST in quanto associa documento, parola e frase ad un sentiment e topic.

NAMED ENTITY EXTRACTION, LINKING AND DISAMBIGUATION

Vediamo ora le tecniche di Named-entity recognition, linking e disambiguazione; tutti argomenti trattati nel capitolo 9 in particolare quando si trattava delle ambiguità di tipo semantico che si possono incontrare quando ci si approccia al Natual Language Processing (NLP).

La named entity recognition risolve il problema di riconoscere dimensioni di entità note e di disambiguare i vari possibili significati di queste entità selezionando il significato più opportuno per il contesto nel quale tale entità si trova tramite una operazione di link della entità con una descrizione presente all'interno di una base di consocenza.

Le principali problematiche che si riscontrano durante il processo NLP quando si vuole fare disambiguazione e linking delle entità sono:

1- I testi sono molto brevi e rumorosi in quanto possono contenere errori di battitura (tipico dei social) hastag non identificati, polisemia ecc.
2- Problematice Out Of Vocabulary (OOV): tale problematica si verifica quando abbiamo delle entità menzionate nel testo di cui non ne sappiamo il significato in quanto non presente nel vocabolario
3- Problematice Out Of Knowledge Base (OOKB): problema che si verifica quando nuove entità emergono nei contesti social per cui tale entità non è presente all'interno di una base di conoscenza

NAMED ENTITY RECOGNITION

Il problema di name entity recognition è un task dell'analisi del linguaggio che ha l'obiettivo di segmentare il testo in frammenti e di classificarli in un insieme predefinito di classi/etichette.

Uno dei primi modelli atti al riconoscimento delle entità è il Conditional Random Fields. Questo è un modello grafico probabilistico indiretto e discriminativo che ha l'obiettivo di imparare in fase di training a massimizzare la probabilità condizionata di avere una certa sequenza di etichette condizionata ad una sequenza di osservazioni, ovvero per ogni token viene associata una specifica etichetta. Esemplificando la definizione appena data possiamo dire di avere una serie di parole alle quali corrispondono una etichetta specifica (la parola Jhon ha categoria persone ecc.) verranno quindi mandati in input una sequenza di parole e si riceverà come output le relative etichette. È importante sottolineare come questo modello consideri la dinamica ovvero ogni parola dipende dalla parola che c'è prima (dipendenza markoviana).

Per andare a massimizzare la probabilità di avere una sequenza di etichette data una sequenza di osservazioni è necessario considerare la relazione che esiste tra la parola e la relativa etichetta e la relazione che esiste tra etichette adiacenti. Quindi questa probabilità si compone in una somma di due sotto operazioni chiamate feature functions:

1- State feature function: funzione che va a valutare lo stato al variare del tempo di una certa etichetta assegnata ad una certa parola (es. la parola Messina può essere associata una volta ad una etichetta persona e una volta ad una etichetta città)
2- Transition feature function: funzione che va a modellare la transizione di stato tra etichette consecutive, ovvero tornando all'esempio di prima va a valutare la probabilità di passare dall'etichetta persona all'etichetta città tra etichette adiacenti

L'obiettivo quindi per andare a massimizzare la probabilità di avere una sequenza di etichette data una sequenza di osservazioni è quello di calcolare i pesi di queste due feature function. Questi pesi vengono calcolati durante la fase di training andando a massimizzare la log-likelihood di un dato training set.

La fase successiva è la fase di inferenza è quella fase che dato un testo va ad etichettare ogni parola del testo con una delle classi possibili. Quindi l'obiettivo è quello di trovare una sequenza di etichette tale per cui la probabilità della sequenza condizionata alle osservazioni dei dati di test massimizzano tale probabilità. Per fare inferenza è possibile utilizzare la tecnica del shortest path:

tale tecnica mette una sequenza di parole e tutte le possibili etichette a loro assegnabili all'interno di un grafo di trelling nel quale sarà necessario individuare il shortest path percorribile per raggiungere la fine della sequenza di parole andando a massimizzare la probabilità a posteriori della sequenza di parole condizionate dalle etichette.

NAMED-ENTITY LINKING

Una volta etichettata una sequenza di testo con le procedure precedentemente menzionate, sappiamo che ogni parola è un'entità che può avere diverse accezioni di significato, quindi si ha la necessità di disambiguare tali entità. Il Conditional Random Fields può fornire una interpretazione iniziale riguardo alle entità ma non sempre il CDR interpreta correttamente il significato di ogni entità.

Per far fronte a questo problema si passa alla procedura di Named-Entity Linking; questa procedura ha l'obiettivo di associare a ciascuna menzione (significato di una entità di una frase) una risorsa presente in una base di conoscenza ottenendo così una netta disambiguazione il significato dell'entità.

Le Named entity prodotte da un CRF (Conditional Random Fields) hanno una menzione chiamata surface form che dovrà essere associata ad una entità all'interno di una base di conoscenza; per fare ciò si sfruttano le etichette presenti all'interno della base di conoscenza (RDF Label) e si cerca un match tra la menzione ottenuta dal CRF con le etichette RDF. In sostanza è come se facessimo una ricerca di un termine su internet e vedessimo tutti i possibili risultati ottenuti. Una volta identificati i potenziali candidati si va a misurare quanto dista la Named Entity rispetto alle descrizioni (abstract) dell'entità presenti nella base di conoscenza. L'ultimo elemento da considerare per completare questa operazione di match è la popolarità della risorsa, ovvero dare un peso alla popolarità di una descrizione rispetto alle altre in una base di conoscenza (es. se cerco Obama sicuramente è più probabile sia Barak rispetto a Michelle). Infine si sceglie il candidato che ottiene più punteggio rispetto al match di label, abstract (descrizioni) e popolarità.

NAMED-ENTITY DISAMBIGUATION

Un problema importante è legato al riconoscimento del senso di una parola (es. Milan la città o la squadra) identificato come Word Sense Disambiguation (WSD).

Ci sono diversi approcci per risolvere il problema della disambiguazione del linguaggio naturale e appartengono a tre grandi famiglie:

1. Knowledge-Based Disambiguation: approccio basato su basi di conoscenza come dizionari, vocabolari ecc. che vengono utilizzati per capire se una parola in un testo assume un significato o un altro
2. Supervised Disambiguation: approccio che necessita di un insieme di dati di training etichettati di cui sappiamo il significato per sviluppare un modello di classificazione che riesce ad associare il significato più probabile
3. Unsupervised Disambiguation: approccio che non fa uso né di una base di conoscenza né di dati etichettati

KNOWLEDGE-BASED DISAMBIGUATION

GlI approcci di Knowledge-based disambiguation fanno utilizzo di risorse lessicali definite Machine Readable Dictionaries (MRD) che sono dei dizionari espressi in formato digitale come ad esempio l'Oxford dictionary, oppure fanno uso di dizionari di sinonimi oppure ancora fanno uso di reti semantiche come Wordnet. Questi approcci sfruttano le definizioni presenti nei dizionari per capire quale senso assuma una parola; per ogni parola il dizionario fornisce una lista di sensi con le loro definizioni e contesto di utilizzo di tale senso e, ovviamente, tali informazioni possono essere utilizzate per il processo di disambiguazione.

Per sfruttare queste risorse per la disambiguazione si fa uso di alcuni algoritmi tra cui quello di Lesk. Questo algoritmo identifica il senso delle parole in un contesto utilizzando una sovrapposizione di definizioni presenti nel dizionario. Il suo funzionamento si articola nelle seguenti fasi:

1. Data una parola di un testo recupera tutte le definizioni dei sensi della parola nel dizionario (questo processo viene eseguito anche per le altre parole del testo)
2. Determina la sovrapposizione di tutti i possibili sensi tra le definizioni di tutte le parole del testo combinate tra loro (es. le due parole pine e cone vengono combinate e si cercano tutte le sovrapposizioni di sensi possibili della combinazione delle due parole)
3. Sceglie il senso con maggiore overlap

L'algoritmo funziona molto bene con due parole in quanto va a vedere i sensi di entrambe le parole, ne determina le possibili combinazioni di senso e identifica la combinazione più adatta (es. Pine cone prenderà il senso di Pine albero e cone pigna e non cono)

Se con questo algoritmo si dovessero considerare più di due parole sarebbe un problema in quanto le combinazioni aumentano a livello esponenziale rendendo così il calcolo non computabile. Per far fronte a questo problema esiste una versione semplificata dell'algoritmo di Lesk; tale algoritmo va a misurare la sovrapposizione tra la definizione dei sensi di una parola rispetto al suo contesto. In pratica va ad identificare il senso corretto di una parola considerandola singolarmente. Il processo si articola nelle seguenti fasi:

1. Dato un testo prendo una singola parola della frase (non più tutte le possibili combinazioni di parole) e ne recupero le definizioni nel dizionario
2. Determino la sovrapposizione nel dizionario e la sua occorrenza all'interno della frase, ovvero confronto la descrizione del dizionario con l'intera frase della parola in analisi e vedo quante parole sono presenti in entrambe

UNSUPERVISED DISAMBIGUATION

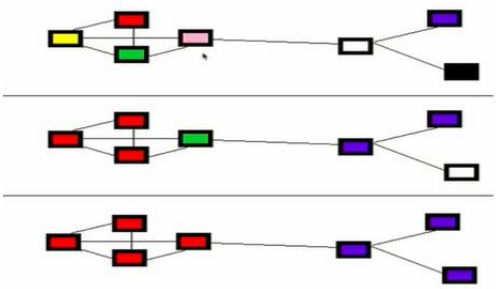
Per quanto riguarda invece gli approcci non supervisionati gli unici elementi che vengono utilizzati per disambiguare sono le parole che fanno parte della stessa frase della parola che devo disambiguare. Questo approccio si rende quindi indipendente dalla limitazione delle basi di conoscenza e della etichettatura; perciò possiamo definire la procedura come Word Sense Discrimination (WSD). Questo nome viene attribuito in quanto andando a cercare il senso di una parola in base al contesto in cui essa si trova, quindi il procedimento è di discriminazione tra significati e si articola a livello generico come segue:

1. Prendo una parola di cui voglio sapere il significato e prendo tutte le frasi possibili dove compare tale parola
2. Si individuano le co-occorrenze di tale parola con le altre parole delle frasi che denotano il contesto in cui compare una parola
3. Si generano dei cluster individuati dalle co-occorrenze che identificano i possibili sensi di tale parola

Andando a fare un esempio di quanto appena spiegato se abbiamo la parola Milan andiamo a vedere le co-occorrenze e vediamo che spesso abbiamo la parola calcio oppure la parola Pioli, quindi possiamo generare un cluster dove il significato della parola Milan è quello della squadra di calcio. Allo stesso modo per altre co-occorrenze possiamo generare altri cluster come quello dove Milan è una città.

CHINESE WHISPERS

Uno degli approcci che si occupa di Word Sense Discrimination è il Chinese Whispers. Questo approccio lavora su grafi di co-occorrenze non dirette e non pesate, dove ogni collegamento tra un nodo e un altro denota la co-occorrenza in una frase. Una volta generato il grafo si assegna ad ogni nodo una classe random dove la classe rappresenta il senso di tale parola. Inizialmente avremo quindi che il numero delle classi è uguale al numero delle parole nel grafo. Successivamente viene processato ogni nodo in ordine random e lo assegna alla classe che lo collega al nodo che ha più link (a parità di collegamenti sceglie a random). Infine viene iterato questo passaggio finché non si hanno più cambiamenti di classe.



Questo metodo è molto semplice e veloce da eseguire e soprattutto non prevede la selezione del numero di cluster K da cui partire, tuttavia il modello non è deterministico in quanto la soluzione varia in base a quale nodo viene scelto per primo randomicamente e tra latro ci possono essere casi in cui la soluzione non converge e si itera all'infinito.

DISCRIMINAZIONE BASATA SU GRAFI

Un diverso approccio sempre basato sui grafi che va a migliorare la soluzione proposta dal Chinese Whisperers si articola nelle seguenti fasi:

1. Raccoglie tutte le frasi in cui compare una determinata parola
2. Pre-processa tutti i documenti di testo rimuovendo stopword e facendo un lemming
3. Costruisce il grafo di co-occorrenza inserendo ogni parola come nodo e collegandole tra loro se c'è co-occorenza e pesando ogni collegamento in base a quante volte si verifica la co-occorenza
4. Applica una tecnica di clustering per individuare i possibili sensi della parola. Questa tecnica passa dal grafo ad uno spazio metrico delle parole che individua per ogni parola il suo vicinato composto dalle altre parole collegate con degli archi ad essa. Successivamente calcola la distanza pesata di Jaccard sul grafo utilizzando il vicinato appena individuato per cercare una corrispondenza tra vicinato e il grafo delle parole. In

sostanza si usa Jaccard per vedere se delle parole che NON sono direttamente collegate tra di loro con un arco possano essere considerate appartenenti allo stesso cluster di senso di parola; per fare ciò prende due parole e vede quanto i loro vicinati si sovrappongono. Infine si generano i cluster in modo aggregativo che parte da una parola con ricorrenza più elevata e presa un'altra parola se la distanza di Jacard tra i due vicinati delle parole non supera una certa soglia allora le due parole appartengono allo stesso cluster, se invece lo supera le parole appartengono a due differenti cluster.

5. Osservando le parole appartenenti ad ogni cluster appoggiandosi ad una base di conoscenza andiamo ad identificare il senso

Questo processo quindi in maniera non supervisionata va ad identificare i cluster senza assegnarli un senso e individua tale senso andando a consultare a posteriori una base di conoscenza.