****

**UNIVERSITÀ DEGLI STUDI DI CATANIA**

**DIPARTIMENTO DI INGEGNERIA ELETTRICA ELETTRONICA E INFORMATICA**

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**GIOVANNI FAUSTO**

**LINK PREDICTION USING GEOMETRIC DEEP LEARNING**

**TESI DI LAUREA**

**­­­­­­­­­­­­­­­­­­­­­**

**ANNO ACCADEMICO 2019-2020**

**INDICE**

[Introduzione 5](#_Toc59436191)

[Capitolo 1 8](#_Toc59436192)

[Introduzione a link prediction 8](#_Toc59436193)

[1.1. Definizione del problema 11](#_Toc59436194)

[1.2. Storia 13](#_Toc59436195)

[1.3. Approcci e metodi 13](#_Toc59436196)

[1.4. Stato dell’arte 16](#_Toc59436197)

[1.5. Applicazioni 19](#_Toc59436198)

[Capitolo 2 21](#_Toc59436199)

[Link prediction e Machine learning 21](#_Toc59436200)

[2.1. Algoritmi di link prediction tradizionali 23](#_Toc59436203)

[2.2. Metodi basati sulla somiglianza 24](#_Toc59436204)

[2.3. Approcci probabilistici e statistici 28](#_Toc59436205)

[2.4. Metodi algoritmici 29](#_Toc59436206)

[2.5. Metodi di preelaborazione 31](#_Toc59436207)

[Capitolo 3 34](#_Toc59436208)

[Implementazione 34](#_Toc59436209)

[3.1. Introduzione 34](#_Toc59436211)

[3.2. Geometric deep learning 40](#_Toc59436212)

[3.3. Graph attention network 44](#_Toc59436213)

[3.4. L’architettura Gat 49](#_Toc59436214)

[3.5. L’architettura del modello 51](#_Toc59436215)

[3.6. Esperimenti 52](#_Toc59436216)

[3.7. Modelli usati ed approccio 54](#_Toc59436217)

[3.8. Risultati 57](#_Toc59436218)

[Considerazioni conclusive 71](#_Toc59436219)

# 

# Introduzione

L’argomento che è stato affrontato in questa tesi è un problema conosciuto come link prediction. Il link prediction è un tema molto studiato negli ultimi anni, in quanto può essere applicato in diversi ambiti scientifici.

In generale, l’obiettivo che c’è alla base di queste tecniche è prevedere un collegamento, che si potrebbe verificare in futuro, tra due entità.

Dato che è stato affrontato in diversi ambiti, queste due entità potrebbero essere qualunque cosa. Ad esempio, pensiamo ad una persona che voglia acquistare un qualsiasi oggetto attraverso un sito web di e-commerce. Il sistema potrà raccomandarle un qualche tipo di acquisto sulla base di quelli fatti in precedenza. Questi meccanismi sono alla base di quelli che vengono chiamati reccomendation system, il cui scopo principale è prevedere i futuri link nella rete persone-prodotti facendo tesoro della storia passata e del profilo delle persone.

Altre entità potrebbero essere le persone. Infatti, se pensiamo alle ormai consolidate reti sociali on-line, che si sono formate ultimamente anche grazie ai social network, la tecnica del link prediction è uno dei meccanismi basilari per il loro funzionamento, in quanto consente di suggerire nuove potenziali ‘amicizie’ cercando di prevedere quale persona è più probabile che possa diventare una nuova conoscente. Un altro ambito del link prediction sono le interazioni proteina-proteina, ossia interazioni per via di reazioni biochimiche; anche in questo caso si cerca di prevedere i loro collegamenti.

Quindi, possiamo affermare che anche grazie all’evolversi delle reti create dai social network, delle reti nell’ambito scientifico, come ad esempio quelle relative alle citazioni, ma anche delle reti naturali, come quelle che esistono nelle interazioni tra proteine, il tema del link prediction è molto attuale e meritevole di ulteriori studi.

In questa tesi, il nostro obiettivo è quello di presentare un modello che faccia proprio questa predizione, cercando di ottenere i migliori risultati per quanto riguarda l’accuratezza che viene raggiunta. Per fare ciò il modello è stato testato su svariate reti di diverse dimensioni e caratteristiche.

Abbiamo affrontato il problema in tre capitoli, introducendo le varie problematiche legate al tema della tesi, gli ambiti dove è possibile applicare questa tecnica e le diverse metodologie usate negli anni, per poi spiegare come siamo arrivati al modello utilizzato all’interno di questo studio e concludendo con la descrizione degli esperimenti realizzati e dei risultati ottenuti.

Più precisamente nel primo capitolo ci siamo concentrati su quella che è una illustrazione del problema del link prediction, i vari ambiti applicativi e lo stato dell’arte, descrivendo quella che può essere definita come una tassonomia dei vari approcci di link prediction che esistono in letteratura.

Nel secondo capitolo ci siamo concentrati per lo più sull’aspetto del link prediction nell’ambito del machine learning, riportando gli approcci, i metodi, e i diversi algoritmi trattati in letteratura.

Nell’ultimo capitolo ci siamo concentrati su quelli che sono gli aspetti implementativi, inquadrando prima i vari processi che hanno portato al modello che abbiamo utilizzato, le Graph Attention Network, quindi spiegando cos’è il deep learning geometrico, che si differenzia da quello tradizionale per la presenza di dati di tipo geometrico e, quindi, trattando anche quelle che sono le reti neurali nel campo dei grafi.

Successivamente ci siamo concentrati sul modello usato presentando, infine, quelli che sono i risultati ottenuti a seguito di diverse esperimenti su una varietà di modelli.

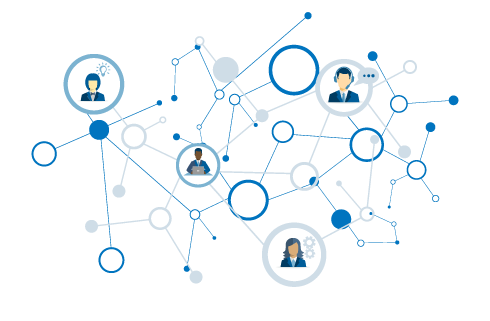
Questo tema, come abbiamo già detto, è nato negli ultimi anni e trova applicabilità in diversi ambiti e, sicuramente, continuerà ad evolversi in quanto lo studio delle reti è, a sua volta, un campo in continua evoluzioni e si prevede che ci saranno tante nuove applicazioni in un prossimo futuro. Una possibile estensione futura della tecnica di link prediction è quella di cercare di realizzare delle previsioni basare solo sulla struttura delle reti (o topologia), senza, quindi, considerare altre caratteristiche specifiche delle reti, come ad esempio particolari feature dei nodi.

**Capitolo 1**

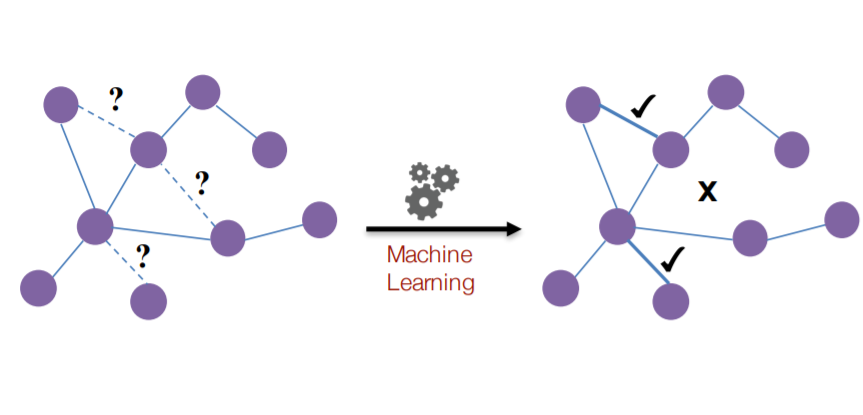
Introduzione a link prediction

Le reti sono un linguaggio generale per descrivere e modellare i sistemi complessi.

Task classici del machine learning nelle reti sono ad esempio: classificazione dei nodi, link prediction, network similarity.



Nella teoria delle network, il link prediction è un problema che riguarda la predizione o meno dell’esistenza di un collegamento, quello che viene chiamato link, tra due entità presenti nella network, che sono chiamati nodi. Il problema di link prediction è un problema di classificazione supervisionato in cui abbiamo coppie di nodi nella rete con due possibili etichette, una che rappresenta la presenza del link e l'altra che rappresenta l'assenza.



Per quanto riguarda questo tema si possono fare diversi esempi e possono essere trattati in diversi modi, in quanto è anche un problema molto diffuso negli ultimi tempi, dato anche dall’enorme crescita ad esempio dei social network, e non di facile risoluzione, infatti, prevedere le interazioni in reti complesse implica affrontare una serie di sfide.

Per cominciare, la maggior parte delle reti del mondo reale mostra comportamenti altamente dinamici e stocastici che possono essere difficili da catturare con modelli di apprendimento automatico. Inoltre, il numero di possibili coppie di nodi, e quindi di istanze del problema, cresce quadraticamente rispetto al numero di nodi della rete. Questa situazione implica la necessità di tecniche altamente scalabili, poiché le reti complesse possono essere composte da migliaia o addirittura milioni di nodi. Inoltre, il problema del link prediction è un problema di classificazione altamente sbilanciato, perché il numero di coppie di nodi non connesse in una rete complessa è solitamente di ordini di grandezza maggiore del numero di coppie connesse, non è raro incontrare reti dove il rapporto dei link non esistenti rispetto quelli esistenti è duecento, o più, volte maggiore. Questo deve essere preso in considerazione dato che alcune tecniche di apprendimento automatico mostrano difficoltà quando si tratta di set di dati sbilanciati.

Infine, una delle principali preoccupazioni nel problema del link prediction è la presenza di incertezza e rumore, poiché l'assenza di un collegamento nella rete non implica l'effettiva assenza di una relazione o interazione tra nodi.

Fortunatamente, nonostante queste difficoltà, la topologia della rete e le caratteristiche di nodi e collegamenti possono essere sfruttate per prevedere questi collegamenti mancanti o futuri con ragionevole precisione in molti casi. In letteratura sono stati proposti diversi modelli di formazione di reti per catturare le dinamiche di formazione dei collegamenti osservate nelle reti del mondo reale.

Le osservazioni empiriche suggeriscono anche che i nodi tendono a formare collegamenti con nodi simili, un fenomeno noto come omofilia.

Nel contesto del link prediction in reti complesse, la definizione di somiglianza dipende dal problema: ciò che può funzionare bene in alcune reti potrebbe funzionare male in altre reti. Da un lato, poiché l'omofilia è stata molto osservata nei social network, dove le persone tendono a formare legami con persone con caratteristiche \ sociodemografiche, comportamentali e intrapersonali simili, la somiglianza potrebbe essere definita in termini di caratteristiche dei nodi. D'altra parte, la somiglianza potrebbe anche essere definita in termini di topologia della rete, come la chiusura triadica, dove è più probabile che due nodi siano collegati se sono vicini nella rete o se sono collegati agli stessi nodi.

Esempi di link prediction includono la previsione dei legami di amicizia tra gli utenti in un social network, la previsione di co-autore in una rete di citazioni e la previsione delle interazioni tra geni e proteine in una rete biologica.

Il link prediction può anche avere un aspetto temporale, dove, dato uno snapshot dell'insieme dei link al momento t, l'obiettivo è prevedere i link al momento t + 1.

Il link prediction è ampiamente applicabile. Nell'e-commerce, il link prediction è spesso un'attività secondaria per consigliare articoli agli utenti, per esempio se visiono un articolo “x” allora verrà consigliato un articolo “y” con delle caratteristiche simili in quanto molto probabilmente sarà presente un link tra i due prodotti.

Nella gestione dei database delle citazioni, può essere utilizzato per la deduplicazione (rimozione dei doppioni o ripetizioni) dei record.

In bioinformatica, è stato utilizzato per prevedere le interazioni proteina-proteina (PPI).

Viene anche utilizzato per identificare gruppi nascosti di terroristi e criminali nelle applicazioni relative alla sicurezza.

Il problema non è facile da affrontare in quanto i toolbox moderni di deep learning sono progettati per sequenze o grid semplici, ad esempio le CNN che sono progettate principalmente per le immagini o le RNN che lavorano principalmente con le sequenze, come ad esempio i testi.

Le reti sono molto più complesse rispetto ai dati sopra citati, ad esempio la struttura topografica è complessa, nessuna località spaziale come le grid, nessun ordine di nodi fissi o punto di riferimento, spesso dinamici e con caratteristiche multimodali.

* 1. Definizione del problema

La teoria dei grafi è una branca della matematica che studia i grafi, che non sono altro che strutture usate per rappresentare le relazioni tra degli oggetti di diversa natura.

Considerando una network G= (V, E), dove V rappresentano i nodi all’interno della rete, e E | V | x | V | rappresenta il set di link veri tra i nodi all’interno della rete. Inoltre, | V | x | V | rappresenta una matrice quadrata che è conosciuta come matrice di adiacenza, questa matrice è simmetrica se ci troviamo nel caso di grafi non diretti.

Si dice che un nodo sia vicino o adiacente ad un altro nodo se entrambi sono collegati da un link, il vicinato di un nodo invece è composto da tutti i vicini di quel determinato nodo.

Nelle reti dirette possiamo distinguere diversi gradi in base ai link, infatti abbiamo i link uscenti, cioè quelli che vanno da un nodo al suo target, quelli entranti, quelli che arrivano al nodo, e quelli totali che sono la somma dei due. Per quanto riguarda i grafi indiretti non c’è questa differenza perché il link vale in entrambe le direzioni.

Oltre a questo, possiamo anche considerare un path, che non è altro che il percorso che collega due nodi passando da dei link intermedi, il path con meno link è considerato il path più breve, la lunghezza del path, infatti, è misurata in base a quanti link sono presenti nel path.

Ci viene fornito l'insieme di nodi V e un sottoinsieme di link veri a cui ci si riferisce come collegamenti osservati.

L'obiettivo del link prediction è identificare i link veri non osservati. Nella formulazione temporale della previsione dei link i collegamenti osservati corrispondono ai collegamenti veri all'istante t e l'obiettivo è inferire l'insieme dei collegamenti veri all'istante t+1.

Di solito ci viene fornito anche un sottoinsieme di link non osservati chiamati link potenziali E', e dobbiamo identificare i veri collegamenti tra questi potenziali collegamenti.

Nella formulazione di classificazione binaria dell'attività di previsione dei link, i potenziali collegamenti sono classificati come collegamenti veri o falsi. Gli approcci di link prediction per questa impostazione apprendono un classificatore Mb che mappa i collegamenti in E' in etichette positive e negative, ad esempio Mb : E' {0,1}.

Nella formulazione della stima della probabilità, i potenziali collegamenti sono associati alle probabilità di esistenza. Gli approcci di link prediction per questa impostazione apprendono un modello Mp che mappa i collegamenti in E' con una probabilità, ad esempio Mp : E' [0,1].

Gli approcci a link singolo apprendono un modello che classifica ogni collegamento in modo indipendente. Gli approcci di previsione strutturati catturano la correlazione tra potenziali collegamenti formulando l'attività come un'attività di previsione di collegamenti collettivi. Gli approcci di previsione dei link collettivi apprendono un modello che identifica congiuntamente tutti i veri link tra l'insieme di potenziali collegamenti.

L'attività di link prediction può anche essere formulata come un'istanza dell'attività di stima del valore mancante. Qui, il grafo è rappresentato come una matrice di adiacenza con valori mancanti. Il compito è completare la matrice identificando i valori mancanti. I metodi basati sulla fattorizzazione matriciale usano comunemente questa formulazione.

* 1. Storia

Il compito del link prediction ha attirato l'attenzione di diverse comunità di ricerca che vanno dalla statistica e la scienza delle reti all'apprendimento automatico e al data mining.

In statistica, i modelli di grafi casuali generativi, come i modelli a blocchi stocastici, propongono un approccio per generare collegamenti tra i nodi in un grafo casuale.

Per i social newtork, Liben-Nowell e Kleinberg hanno proposto modelli di link prediction basati su diverse misure di prossimità del grafo.

Diversi modelli statistici sono stati proposti per il link prediction dalla comunità di machine learning e data mining. Ad esempio, Popescul ha proposto un modello di regressione logistica strutturato che può utilizzare caratteristiche relazionali.

I modelli di probabilità condizionata locale basati su attributi e caratteristiche strutturali sono stati proposti da O'Madadhain. Diversi modelli basati su modelli grafi diretti per la previsione di collegamenti collettivi sono stati proposti da Getoor.

Sono stati proposti anche altri approcci basati su random walk e fattorizzazione di matrice. Con l'avvento del deep learning, sono stati proposti anche diversi approcci basati sull'embedding dei grafi per il link prediction.

* 1. Approcci e metodi

Sono stati proposti diversi approcci di link prediction, inclusi approcci non supervisionati come misure di somiglianza calcolate sugli attributi dell'entità, approcci basati sul random walk e fattorizzazione di matrice e approcci supervisionati basati su modelli grafici e deep learning.

Il problema del link prediction è un problema di classificazione supervisionata, che mira a prevedere la classe effettiva di ciascuna coppia di nodi. In questo problema vengono considerate due classi differenti: esistenza e assenza di link. Pertanto, la metodologia di convalida più comune per le tecniche di link prediction è l'uso di un set di addestramento, un set di test e un set di convalida opzionale. Sebbene sia possibile seguire altri approcci, l'approccio più comune è suddividere l'insieme di collegamenti E per creare questi insiemi. I collegamenti nel set di addestramento devono essere utilizzati dalle tecniche di link prediction per provare a prevedere i collegamenti nel set di test. Il set di convalida viene utilizzato per ottimizzare gli eventuali iperparametri.

Un approccio nella valutazione del task di classificazione è la corss validation, una tecnica di convalida che comprende cicli di valutazione su sottoinsiemi complementari di dati. Nella convalida incrociata k-fold, l'insieme originale di collegamenti E viene suddiviso in k sottoinsiemi disgiunti. Per ogni round di valutazione, i sottoinsiemi k-1 vengono combinati e utilizzati come set di addestramento mentre il sottoinsieme rimanente viene utilizzato come set di test. Questo processo viene ripetuto k volte in modo che ogni sottoinsieme agisca come set di test una volta.

È possibile utilizzare due metodologie per misurare le prestazioni delle tecniche di link prediction.

Il primo è l'approccio comunemente usato nella classificazione tradizionale, in cui il classificatore deve produrre la classe prevista e la valutazione viene eseguita confrontando la classe prevista con la classe effettiva.

Poiché la maggior parte delle tecniche di link prediction di solito produce un punteggio o una probabilità proporzionale alla probabilità di esistenza di un collegamento, un secondo approccio si basa sulla classificazione di tutte le coppie di nodi in base a questi punteggi e sulla restituzione delle prime t coppie di nodi come istanze positive, dove t è uguale al numero di collegamenti nel set di test. Il resto delle coppie di nodi viene restituito come istanze negative, prevedendo che non siano connesse.

Quindi potremmo definire quella che viene chiamata matrice di confusione, dove abbiamo i veri positivi cioè i link che realmente esistono e sono stati previsti come esistenti, e i veri negativi cioè quando prediciamo esattamente la non esistenza del link, mentre gli altri sono falsi positivi e falsi negativi cioè quando prediciamo come vero un link che non esiste o non prediciamo un link che realmente esiste. Da questa matrice possiamo ricavare come metrica di valutazione delle prestazioni l’accuratezza.

Oltre all’accuratezza un’altra metrica utilizzata molto spesso è quella che si ricava dalle note curve ROC, questa metrica è l’area sotto la curva conosciuta come AUC, più si discosta dalla diagonale e migliore è la predizione che effettua il nostro modello, in quanto la diagonale rappresenta una predizione random.

Gli approcci di link prediction possono essere suddivisi in due ampie categorie in base al tipo di rete sottostante:

* approcci di link prediction per reti omogenee
* approcci di link prediction per reti eterogenee.

In base al tipo di informazioni utilizzate per prevedere i collegamenti, gli approcci possono essere classificati come approcci basati sulla topologia, approcci basati sul contenuto e metodi misti.

I metodi basati sulla topologia sostengono ampiamente che i nodi con una struttura di rete simile abbiano maggiori probabilità di formare un collegamento.

I metodi di similarità dei nodi prevedono l'esistenza di un collegamento in base alla somiglianza degli attributi del nodo.

L’embedding dei grafi offrono anche un modo conveniente per prevedere i collegamenti. Gli algoritmi di embedding dei grafi, come Node2vec, apprendono uno spazio di embedding in cui i nodi vicini sono rappresentati da vettori in modo che le misure di similarità vettoriale, come la similarità del prodotto puntuale o la distanza euclidea, rimangano nello spazio di embedding. Queste somiglianze sono funzioni sia delle caratteristiche topologiche che della somiglianza basata sugli attributi. È quindi possibile utilizzare altre tecniche di apprendimento automatico per prevedere gli edges sulla base della similarità vettoriale.

* 1. Stato dell’arte

Il compito impegnativo del link prediction ha attirato molta attenzione da parte della comunità scientifica a causa delle numerose applicazioni che questo problema ha nel mondo reale. I progressi in questo campo hanno portato allo sviluppo di un gran numero di approcci che modellano diversi modelli di connettività.

Il problema del link prediction è stato affrontato seguendo approcci molto diversi. Nella tassonomia che viene proposta in letteratura si possono notare due livelli per classificare l'ampio spettro di tecniche di link prediction esistenti. Nel primo livello, le tecniche sono state classificate come metodi basati sulla somiglianza, metodi probabilistici e statistici, metodi algoritmici e metodi di preelaborazione.

Immagine che contiene testo

Descrizione generata automaticamente

I metodi basati sulla somiglianza si basano sul presupposto che i nodi tendono a formare connessioni con nodi simili. Queste tecniche assumono la forma di una funzione che assegna un punteggio di somiglianza, dove più somiglianza implica più probabilità di esistenza del collegamento, per una coppia di nodi dati in base alla topologia della rete. Tecniche differenti catturano differenti definizioni di somiglianza.

Nel secondo livello della tassonomia, le tecniche basate sulla similarità sono classificate in base alla gamma di informazioni topologiche che prendono in considerazione: approcci locali, che utilizzano solo informazioni dirette di vicinato, approcci globali, che considerano l'intera topologia della rete per stimare la somiglianza tra qualsiasi coppia di nodi e approcci quasi locali, che sono un punto intermedio tra approcci locali e globali.

La tecnica di link prediction locale più basilare, che sorprendentemente funziona molto bene in molti casi, è il conteggio del numero di vicini condivisi tra i nodi. Tuttavia, supporre che tutti i vicini condivisi contribuiscano nella stessa misura è un approccio ingenuo.

L'indice Adamic-Adar e l'indice di allocazione delle risorse penalizzano il contributo di ogni vicino condiviso proporzionalmente al suo grado. Altre tecniche, come l'indice Jaccard o l'indice Hub Promoted, penalizzano la somiglianza dei nodi in modo proporzionale alla quantità di vicini non condivisi.

Le tecniche basate sulla somiglianza globale di solito considerano i percorsi esistenti tra i nodi. L'approccio più semplice sarebbe la lunghezza negata del percorso più breve tra una coppia di nodi. Tuttavia, questo approccio ottiene scarsi risultati nella pratica a causa della mancata considerazione dei percorsi indiretti. L'indice di Katz, che può essere calcolato in forma chiusa, risolve questa limitazione sommando l'influenza di tutti i possibili percorsi tra i nodi considerati. Un'altra famiglia di tecniche globali si basa sulla stima della probabilità di raggiungere il nodo target a partire dal nodo sorgente mediante random walk. La random walk con riavvio e propagazione del flusso sono esempi di questo tipo di tecniche.

Infine, gli approcci quasi-locali sono di solito versioni limitate di tecniche globali, come il Local Path Index, fortemente ispirato all'indice di Katz, o Local Random Walks, basato sul random walk limitato in lunghezza.

La categoria dei metodi probabilistici e statistici è costituita da tecniche che assumono una struttura di rete nota a priori con parametri sconosciuti che sono stimati utilizzando l'analisi statistica e la teoria della probabilità. Una volta stimati i parametri più adatti alla rete, il modello ottenuto può essere utilizzato per prevedere la probabilità di esistenza di ogni collegamento non osservato. Ad esempio, il modello di struttura gerarchica presuppone un'organizzazione gerarchica della rete, in cui i nodi hub fungono da ponti tra comunità di nodi altamente connessi. Un approccio diverso, noto come modello a blocchi stocastici, presuppone che la rete sia organizzata in comunità o blocchi di nodi densamente connessi. I parametri di questo modello, che sono le probabilità di collegamento tra i nodi di ogni possibile coppia di blocchi, sono stimati massimizzando la verosimiglianza del modello data una rete. Altri esempi di tecniche in questo gruppo sono il modello di formazione del ciclo, basato sulla stima della probabilità di esistenza di cicli di una certa lunghezza, e il modello di co-occorrenza locale, basato sull'esistenza di nodi rilevanti nei percorsi tra altri nodi.

Poiché il problema del link prediction è un'attività di classificazione supervisionata, un approccio evidente è l'uso di tecniche di apprendimento automatico, come classificatori o metaeuristiche. Questo tipo di tecniche di apprendimento e ottimizzazione sono classificate nella tassonomia come metodi algoritmici. Esistono molti studi sull'applicazione degli algoritmi più diffusi per la classificazione al problema del link prediction. Il problema dello squilibrio di classe e della loro scarsa scalabilità limita la loro applicabilità in molte situazioni. Recentemente, algoritmi di ottimizzazione metaeuristica come algoritmi evolutivi sono stati utilizzati per stimare l'influenza di diversi possibili fattori che guidano le dinamiche della formazione dei collegamenti nelle reti. In questa categoria sono incluse anche le tecniche di fattorizzazione matriciale, ampiamente utilizzate nei sistemi di raccomandazione. Queste tecniche si basano sulla fattorizzazione della matrice di adiacenza della rete al fine di apprendere caratteristiche latenti che spiegano l'esistenza o l'assenza di collegamenti nella rete.

Infine, i metodi di preelaborazione costituiscono l'ultimo gruppo della tassonomia. Questi approcci mirano a ridurre il rumore presente nella rete rimuovendo i collegamenti deboli o spuri. Ad esempio, l'approssimazione di basso rango della matrice di adiacenza può essere utilizzata per rimuovere i collegamenti meno rilevanti pur mantenendo la struttura complessiva della rete. In questa sezione sono incluse anche altre tecniche come i bigram invisibili, basati sull'idea di sostituire i nodi con elementi simili e il filtraggio, basato sull'utilizzo di altre tecniche di link prediction per valutare la forza dei collegamenti e rimuovere quelli più deboli.

* 1. Applicazioni

Il link prediction ha trovato vari utilizzi, ma qualsiasi dominio in cui le entità interagiscono in modo strutturale può trarre vantaggio dal link prediction.

Un'applicazione comune del link prediction è il miglioramento delle misure di somiglianza per gli approcci di filtraggio collaborativo alla raccomandazione.

Il link prediction viene spesso utilizzata nei social network per suggerire amici agli utenti. È stato utilizzato anche per prevedere le associazioni criminali.

In biologia, il link prediction è stato utilizzato per prevedere le interazioni tra le proteine ​​nelle reti di interazione proteina-proteina.

Il link prediction è stato utilizzato anche per dedurre le interazioni tra farmaci e obiettivi utilizzando il link prediction. Un'altra applicazione si trova nella previsione della collaborazione nelle reti scientifiche di co-autore.

La risoluzione delle entità, nota anche come deduplicazione, utilizza comunemente il link prediction per prevedere se due entità in una rete sono riferimenti alla stessa entità fisica. Alcuni autori hanno utilizzato le informazioni di contesto in domini strutturati in rete per migliorare la risoluzione delle entità.

Il link prediction nel contesto degli effetti di rete è stato utilizzato per analizzare le tendenze a diffondersi attraverso le reti e può essere utilizzata per migliorare le strategie di marketing, in particolare il marketing virale.

**Capitolo 2**

Link prediction e Machine learning

Al momento, la maggior parte degli algoritmi di previsione dei link si basa sulla somiglianza tra due entità. Il link prediction si basa sull'evidenza empirica che due entità hanno maggiori probabilità di interagire se sono simili. La somiglianza nelle reti deve essere intesa come un concetto astratto e potrebbe variare tra le reti. Comprendere il dominio rappresentato dalla rete è un passaggio cruciale per definire la somiglianza tra due nodi. Nella maggior parte dei domini, è stato osservato che i nodi tendono a formare comunità altamente connesse. Ciò ha portato alla definizione comune di somiglianza come la quantità di path diretti o indiretti rilevanti tra i nodi.

Una delle principali difficoltà nel link prediction è raggiungere un buon equilibrio tra la quantità di informazioni considerate per eseguire la previsione e la complessità dell'algoritmo delle tecniche necessarie per raccogliere tali informazioni. Poiché le reti effettive sono generalmente formate da centinaia di migliaia o addirittura milioni di nodi, le tecniche utilizzate per eseguire il link prediction devono essere altamente efficienti. Tuttavia, considerare solo le informazioni locali potrebbe portare a previsioni sbagliate, soprattutto in reti molto sparse.

Le informazioni sulla topologia dei social network sono una delle fonti principali per progettare la funzione di somiglianza tra entità. Ma gli algoritmi di link prediction esistenti non applicano sufficientemente le informazioni sulla topologia di rete.

Attualmente, con il rapido sviluppo, i social network online hanno preso sempre più piede nella vita delle persone. Un sacco di sociologia, biologia e sistemi di informazione possono usare le reti per essere descritte, in cui i nodi rappresentano l'individuo e i link rappresentano le relazioni tra gli individui o l'interazione tra gli individui. Pertanto, lo studio di reti complesse è stato il ramo importante di molti campi scientifici.

Il link prediction è strettamente correlato a molte aree. Pertanto, negli ultimi anni sono stati proposti molti algoritmi di correlazione per risolvere il problema del link prediction.

Nella vita reale, gli individui non sono indipendenti l'uno dall'altro. Sono reciprocamente contattati. Se prestiamo attenzione solo agli attributi individuali, ignorando le relazioni tra gli individui, ciò è destinato a influenzare l'accuratezza e la completezza dell'analisi.

Il social network rappresenta la relazione tra entità sociali (come ogni persona, gruppo sociale). L'analisi dei social network si concentra sulla spiegazione dei modelli nascosti e degli effetti di queste relazioni. Si basa su tale presupposto, vale a dire che gli individui nei gruppi sociali sono unità autonome interdipendenti, non indipendenti. Il social network include un insieme di oggetti e relazioni tra loro. Queste relazioni possono essere qualsiasi tipo di relazione sociale, come l'amicizia, l'acquisto di relazioni. I social network possono essere rappresentati dal grafo. Il grafo contiene l'insieme di nodi e l'insieme di edge.

Le proprietà del social network sono effetto small word, effetto scale-free e effetto cluster.

L'effetto Small World è stato prodotto dallo psicologo americano Stanley Milgram nel 1969. Ha scoperto che assegna il nome del destinatario in modo casuale e invia un messaggio ai suoi, e così via, questo messaggio può raggiungere le mani del destinatario in un percorso relativamente breve (circa 6 persone). Il risultato è stato che è stata generata la famosa teoria dei "sei gradi di separazione". Il social network, la rete Internet e la rete di simulazione hanno proprietà small world. Nella rete, l’effetto small word si riferisce al fatto che la distanza media nella rete è molto piccola rispetto alle dimensioni della rete. Vale a dire, ogni coppia di nodi può essere collegata tramite un breve percorso in una rete.

L'effetto scala-free si riferisce al fatto che la maggior parte dei collegamenti dei nodi sono molto piccoli nella rete; solo pochi nodi hanno molti collegamenti. In questa rete, i nodi con alto grado sono chiamati hub (nodo cardine). Il nodo hub domina le operazioni di rete. L'effetto scala-free mostra che la distribuzione dei gradi dei nodi è gravemente irregolare nella rete su larga scala.

L'effetto di clustering del social network si riferisce alla presenza di una cerchia di amici, conoscenti, anelli e altri piccoli gruppi nel social network. Ogni membro del piccolo gruppo si conosce. Questo fenomeno può essere descritto dal grafo, vale a dire, ci sono molti sottografi completamente connessi nei social network.

2. 1. Algoritmi di link prediction tradizionali

Link prediction è un importante campo di ricerca nel data mining. Ha una vasta gamma di scenari. Molte attività di data mining coinvolgono la relazione tra gli oggetti. Il link prediction può essere utilizzato per sistemi di raccomandazione, social network, recupero di informazioni e molti altri campi.

Dato uno snapshot del grafo del social network in un momento G = (V, E) e il nodo v1 e il nodo v2, il link prediction consiste nel prevedere la probabilità del collegamento tra il nodo v1 e il nodo v2. Dalla definizione di link prediction si può vedere che l'attività di link prediction è divisa in due categorie. La prima categoria è prevedere che il nuovo collegamento apparirà in futuro. La seconda categoria è quella di prevedere collegamenti sconosciuti nascosti nello spazio.

La struttura più semplice dell'algoritmo di link prediction si basa sulla somiglianza dell'algoritmo. Qualsiasi coppia di nodo x e nodo y, che abbiamo assegnato a questo nodo è una funzione di similarità (x, y), questa funzione è definita come la funzione di similarità tra i nodi x e y. Quindi ordinando la coppia di nodi in base ai valori della funzione dal più grande al più piccolo, maggiore è il valore della funzione di somiglianza, maggiore è la probabilità del collegamento nei nodi.

La maggior parte delle tecniche di link prediction esistenti lo considerano un problema di classificazione in cui alle coppie di nodi non connessi viene assegnato un punteggio proporzionale alla probabilità di esistenza di un collegamento tra di loro. Di solito viene stabilita una soglia: tutte le coppie con un punteggio superiore alla soglia sono considerate come istanze positive e tutte le coppie al di sotto della soglia sono viste come istanze negative. Questa soglia può essere specificata dall'utente, dipendente dall'applicazione o determinata automaticamente. La selezione automatica della soglia nel link prediction rimane un problema inesplorato. Il problema del link prediction può essere visto come un problema di classificazione binaria per i collegamenti nella rete in cui vengono considerate due classi: positiva o esistenza di collegamento e negativa o assenza di collegamento.

* 1. Metodi basati sulla somiglianza

I metodi basati sulla somiglianza presumono che i nodi tendano a formare collegamenti con altri nodi simili. Questi metodi derivano dall'ipotesi che due nodi siano simili se sono collegati a nodi simili o sono vicini nella rete in base a una data funzione di distanza.

Gli approcci definiscono una funzione s (x, y) che assegna un punteggio noto come similarità per ogni coppia di nodi x e y. Questa misura viene calcolata per ogni coppia interessante di nodi, di solito quelli con collegamenti non osservati tra loro. Le coppie di nodi sono classificate in ordine decrescente in base ai loro punteggi di somiglianza, quindi si suppone che i collegamenti in cima alla classifica abbiano maggiori probabilità di essere presenti nella serie di collegamenti mancanti.

La definizione di similarità non è un compito banale, poiché ha una componente euristica. La funzione di similarità può variare tra le reti anche dallo stesso dominio. Come risultato non sorprendente, è stato proposto un gran numero di metodi basati sulla somiglianza con diverse definizioni di somiglianza. È stato dimostrato empiricamente che la somiglianza tra i nodi può essere definita in termini di proprietà topologiche della rete.

I primi approcci che andiamo a considerare sono quelli locali. Gli approcci basati sulla similarità locale utilizzano informazioni strutturali relative al vicinato dei nodi per calcolare la somiglianza di ciascun nodo con altri nodi della rete. Questi approcci sono più veloci delle tecniche non locali e altamente parallelizzabili. Inoltre, consentono di gestire in modo efficiente il problema del link prediction in reti molto dinamiche e mutevoli come i social network online. Il loro principale svantaggio è che l'uso solo di informazioni locali limita l'insieme di nodi, la somiglianza può essere calcolata per la distanza di due nodi (vicini di vicini). Questo può essere un grosso svantaggio poiché molti collegamenti si formano a distanze maggiori di due in molte reti del mondo reale, specialmente in reti non di piccole dimensioni. Tuttavia, questi metodi hanno mostrato un'accuratezza di previsione molto competitiva rispetto a tecniche più complesse. Va notato che, poiché questi metodi sono limitati a nodi a due hop, la loro complessità temporale è O(vk2f (k)) dove f(k) è la complessità del calcolo della somiglianza tra una coppia di nodi e la loro complessità spaziale è O(vk2).

I vicini comuni sono la tecnica locale più semplice. La somiglianza tra due nodi è definita come il numero di vicini condivisi tra entrambi i nodi. Ha senso presumere che, se due persone condividono molte conoscenze, è più probabile che si incontrino rispetto a due persone senza contatti comuni. Diversi studi hanno confermato questa ipotesi osservando una correlazione tra il numero di vicini condivisi tra coppie di nodi e la probabilità di essere collegati. Nonostante la sua semplicità, questa misura funziona sorprendentemente bene sulla maggior parte delle reti del mondo reale e supera approcci molto complessi.

Un’altra tecnica è quella dell'indice Adamic-Adar (AA), questa misura di somiglianza, inizialmente proposta da Lada Adamic e Eytan Adar, aveva lo scopo di misurare la somiglianza tra due entità in base alle loro caratteristiche condivise. Tuttavia, il peso di ciascuna caratteristica è logaritmicamente penalizzato dalla sua frequenza di comparsa.

L'indice di allocazione delle risorse, questo indice è motivato dal processo di allocazione delle risorse che si svolge in reti complesse. Modella la trasmissione di unità di risorse tra due nodi non collegati x e y attraverso i nodi di vicinato. Ogni nodo di vicinato ottiene un'unità di risorsa da x e la distribuisce equamente ai suoi vicini. La quantità di risorse ottenute dal nodo y può essere considerata come la somiglianza tra entrambi i nodi. Va notato che questa misura è fortemente correlata ai vicini comuni e all'indice Adamic-Adar.

L'allocazione delle risorse in base alle interazioni dei vicini comuni è motivata dal processo di allocazione delle risorse in cui ogni nodo invia un'unità di risorsa ai suoi vicini. Tuttavia, questo metodo prende in considerazione anche il ritorno di risorse nella direzione opposta.

Dopo gli approcci locali andiamo ad attenzionare quelli che sono gli approcci globali. Gli indici globali basati sulla somiglianza utilizzano le informazioni topologiche dell'intera rete per assegnare un punteggio a ciascun collegamento. Questi metodi non si limitano a misurare la somiglianza tra due nodi distanti. Tuttavia, la loro complessità computazionale può renderli irrealizzabili per reti di grandi dimensioni e la loro parallelizzazione può essere molto complessa, specialmente in ambienti distribuiti dove la topologia completa della rete potrebbe non essere conosciuta da ogni agente computazionale. Nonostante esibiscano complessità temporali molto diverse, la loro complessità spaziale è O(v2), poiché devono memorizzare un punteggio per ogni coppia di nodi.

Il percorso più breve negato è una misura di somiglianza del grafo di base che richiede di calcolare il percorso più breve tra una coppia di nodi. I percorsi più brevi possono essere calcolati in modo efficiente con l'algoritmo di Dijkstra, che ha una complessità O(e\*log(v)) utilizzando una rappresentazione dell'elenco di adiacenza della rete e una struttura di dati heap per la sua coda di priorità.

Poiché i percorsi più brevi devono essere calcolati per ogni nodo della rete, la complessità temporale complessiva di questo metodo è O(ev\*log(v)). La precisione della previsione del percorso negativo è scarsa anche se confrontata con la maggior parte dei metodi locali. Altri metodi, basati su più percorsi, ottengono risultati notevolmente migliori. Questo fatto illustra l'importanza di considerare i percorsi indiretti nelle tecniche di link prediction.

Random walk, dato un grafo e un nodo iniziale, supponiamo di selezionare a caso un vicino di questo nodo e di spostarci su di esso; quindi, ripetiamo questo processo per ogni nodo raggiunto. Questa catena di Markov di nodi selezionati casualmente è nota come random walk sul grafo. La random walk è stata introdotta dal matematico Karl Pearson ed è stata applicata per descrivere molti processi stocastici in molti campi come l'economia, la fisica o la biologia. Se definiamo come il vettore di probabilità di raggiungere qualsiasi nodo iniziando una random walk dal nodo x, la probabilità di raggiungere ciascun nodo può essere approssimata iterativamente.

Il tempo medio di viaggio (ACT) è definito come il numero medio di passi che un random walker, che parte dal nodo x, impiega per raggiungere un nodo y per la prima volta e tornare a x. Se il numero di passaggi necessari per raggiungere il nodo y a partire dal nodo x in una random walk è indicato con m (x, y) (noto anche come tempo di battuta), è possibile definire il valore ACT c (x, y) tra entrambi i nodi come n(x,y)=m(x,y)+m(y,x).

Dopo aver visionato gli approcci globali e quelli locali passiamo a quelli quasi locali. Recentemente sono emersi metodi quasi locali per trovare un equilibrio tra misure locali e globali. Gli approcci quasi locali sono efficienti da calcolare quasi quanto i metodi locali, ma considerano anche informazioni topologiche aggiuntive, come fanno i metodi globali. Non tengono conto della somiglianza tra qualsiasi coppia arbitraria di nodi nella rete, ma non sono nemmeno limitati ai vicini dei vicini. Alcuni metodi quasi locali hanno accesso all'intera rete, ma la loro complessità temporale algoritmica è ancora inferiore alla complessità temporale dei metodi globali. La loro complessità spaziale è O (vk2 + s), dove s dipende da parametri specifici che stabiliscono il numero di iterazioni o la lunghezza dei percorsi considerati.

L'indice di percorso locale (LPI), questo indice è fortemente basato sull'indice Katz ma considera solo un numero finito di lunghezze di percorso.

Le random walk locali sfruttano il concetto di random walk, ma limitano il numero di iterazioni a un numero fisso a priori piccolo *l*. Questo metodo non si concentra sullo stato stazionario quando viene raggiunta la convergenza come altri approcci basati sul random walk.

I metodi basati sul random walk sono troppo sensibili alla topologia della rete in zone distanti. Il metodo del random walk sovrapposto, che si basa sul metodo del random walk locale, è stato proposto per contrastare questo problema rilasciando continuamente il walker al nodo di partenza.

* 1. Approcci probabilistici e statistici

Molti modelli di formazione di reti sono stati descritti con successo in termini di concetti statistici e probabilistici. Questi studi hanno aperto la porta per collegare le tecniche di previsione basate sull'analisi statistica e la teoria della probabilità. Questi approcci di solito presuppongono che la rete abbia una struttura nota. Costruiscono un modello che si adatta alla struttura e stimano i parametri del modello utilizzando metodi statistici. Questi parametri vengono utilizzati per calcolare la probabilità di formazione di ciascun collegamento non osservato. Questi valori di probabilità possono essere utilizzati per classificare i potenziali collegamenti come si fa nei metodi basati sulla somiglianza.

Iniziamo parlando del modello di struttura gerarchica. Alcuni studi dimostrano che molte reti reali sono organizzate gerarchicamente, comprese le reti metaboliche, le reti di interazione delle proteine, i domini Internet e alcuni social network come le reti degli attori. Nelle reti gerarchiche, i nodi con un grado più elevato dovrebbero avere un coefficiente di clustering inferiore rispetto ai nodi con un grado inferiore. I nodi hub collegano debolmente comunità isolate di nodi altamente raggruppati. In questo modo si forma una struttura gerarchica.

Il metodo proposto da Clauset rappresenta una rete strutturata gerarchicamente da un dendrogramma con | V | foglie e | V | - 1 nodi interni. Ogni foglia rappresenta un nodo della rete e ogni nodo interno rappresenta una relazione tra i suoi nodi discendenti nel dendrogramma. Ogni nodo interno n ha una probabilità associata pn, che è uguale alla probabilità di un collegamento tra i nodi di entrambi i rami discendenti da esso. Ogni rete ha più rappresentazioni basate su dendrogrammi a seconda di come sono impostati i nodi interni.

Il modello del blocco stocastico, la maggior parte delle reti non si adatta a uno schema gerarchico. Un approccio più generale consiste nel considerare che i nodi sono distribuiti in comunità o blocchi. La probabilità di formazione del collegamento tra due nodi dipende direttamente dal blocco a cui appartengono. In questo modello, dobbiamo calcolare una partizione M della rete in cui ogni nodo è assegnato a un gruppo o blocco mM.

Il modello di formazione del ciclo si basa sul presupposto che le reti abbiano la tendenza a chiudere i cicli nel loro processo di formazione. Questa ipotesi corrisponde ad altre tecniche come il metodo dei vicini comuni, che conta il numero di cicli di lunghezza tre che cosa si formerebbe se il collegamento valutato esistesse. Questo metodo cerca di acquisire cicli più lunghi estendendo il coefficiente di raggruppamento globale a un coefficiente di raggruppamento generalizzato.

* 1. Metodi algoritmici

Tutti gli approcci presentati nelle sezioni precedenti si basano sul calcolo di un punteggio per ogni collegamento non osservato definendo una somiglianza o una funzione di probabilità. Tuttavia, il link prediction può anche beneficiare di altri approcci algoritmici, tra cui l'apprendimento supervisionato e le tecniche di ottimizzazione. Questi approcci sono stati meno esplorati nella letteratura sul link prediction, ma presentano proprietà interessanti.

Il problema del link prediction può essere affrontato con le classiche tecniche di apprendimento supervisionato. Può essere visto come un problema di classificazione con due classi: esistenza e assenza di collegamento.

Questa è una tecnica molto potente poiché può utilizzare qualsiasi proprietà e misura topologica; o anche qualsiasi altra misura di link prediction come caratteristica. Questo approccio, tuttavia, deve affrontare il noto problema dello squilibrio di classe, poiché quasi tutte le reti reali sono sparse; ovvero, il numero di collegamenti assenti è estremamente superiore al numero di collegamenti esistenti.

Sono stati proposti diversi approcci basati su classificatori. È possibile utilizzare quasi tutti i tipi di classificatori. Alcuni lavori hanno confrontato classificatori diversi tra cui alberi decisionali, macchine a vettori di supporto (SVM), vicini k-più vicini, perceptrons multistrato, reti di funzioni di base radiale, naive bayes e diversi insiemi di questi classificatori. Altri autori hanno ottenuto buoni risultati utilizzando classificatori di foreste casuali. Le foreste casuali sono insiemi di alberi decisionali addestrati sullo stesso set di addestramento ma utilizzando diversi sottoinsiemi delle funzionalità disponibili.

Molti metodi basati su classificatori non classificano possibili collegamenti come metodi probabilistici o basati sulla somiglianza. Questa proprietà rende più difficile il confronto, poiché in questo caso il numero di collegamenti previsti in ciascuna classe non può essere controllato.

La formazione del collegamento è un processo complesso con un gran numero di fattori coinvolti. Tutti gli approcci sono euristici, nel senso che cercano di superare un predittore di base casuale formulando alcune ipotesi sulla formazione dei collegamenti nella rete studiata. Supporre che tutti i collegamenti siano formati dallo stesso meccanismo è una semplificazione eccessiva del problema.

Recentemente è stato proposto un approccio basato su un algoritmo evolutivo. Questo metodo presuppone che diverse euristiche di formazione dei collegamenti possano coesistere e cooperare nella stessa rete. Utilizza una strategia di evoluzione per ottimizzare l'influenza di diversi predittori del collegamento di base, inclusi indici basati sulla somiglianza locale e globale e caratteristiche di somiglianza dei nodi in una rete di risposta reciproca di Twitter. Ogni soluzione individuale o candidata *u* nella popolazione è un vettore w(u) di tanti numeri reali quanti sono i numeri euristici considerati. Ciascuno di questi valori rappresenta il peso o l'influenza dell'euristica nella rete. Ogni candidato rappresenta un predittore di collegamento basato sulla somiglianza caratterizzato da una funzione di somiglianza. L'idoneità di ogni candidato è definita come la precisione ottenuta applicandola su un secondo sottoinsieme di allenamento e su un secondo sottoinsieme di test creato da collegamenti campionati dal set di allenamento originale. Una strategia di evoluzione dell'adattamento della matrice di covarianza (CMA-ES) viene applicata per generare nuovi candidati con una migliore idoneità creando nuove popolazioni di candidati sulla base di combinazioni e mutazioni di quelle precedenti.

I modelli di fattorizzazione matriciale sono stati ampiamente utilizzati nei sistemi di raccomandazione poiché possono estrarre funzionalità latenti o utilizzare funzionalità aggiuntive per eseguire previsioni. Ad esempio, è stato suggerito un metodo di apprendimento delle caratteristiche latenti per il link prediction composto da un vettore latente i per ogni nodo *i*, un fattore di scala Fx,y per ogni collegamento, pesi per le caratteristiche del nodo Wn e un vettore di pesi per le caratteristiche del collegamento l . Inoltre, ogni nodo i ha un vettore di caratteristiche ai e ogni collegamento ha un vettore di caratteristiche x,y. In questo modello, dati i vettori latenti, il fattore di scala e i pesi, viene calcolato un punteggio di previsione per ciascuna coppia di nodi x e y.

* 1. Metodi di preelaborazione

I metodi di preelaborazione sono noti anche come approcci di livello superiore o meta-approcci, poiché sono destinati ad essere utilizzati insieme ad altri metodi. L'obiettivo principale degli approcci di preelaborazione è quello di ridurre il rumore presente nelle reti come collegamenti “deboli” o “falsi”, al fine di migliorare le prestazioni dei metodi sopra descritti.

Come primo metodo analizziamo il metodo low-rank. Questo metodo semplifica la struttura della rete per ridurre il suo rumore utilizzando la matrice di adiacenza "A" rappresentazione del graco risolvendo il problema di approssimazione di rango basso. Questo problema di ottimizzazione cerca di minimizzare una funzione di costo che misura l'adattamento tra la matrice originale e una matrice di approssimazione di rango ridotto. Il rango della matrice approssimata è solitamente impostato su un numero relativamente piccolo. Questo problema può essere risolto algoritmicamente in modo efficiente utilizzando la decomposizione del valore singolare (SVD) della matrice originale. Esistono diverse tecniche per calcolare l'SVD. L'approccio più elementare si basa sul fatto che i valori singolari sono le radici quadrate degli autovalori di AAT. Sfortunatamente, questa tecnica non è pratica per matrici di grandi dimensioni a causa della mancanza di accuratezza numerica.

Un bigram è una sequenza di due elementi adiacenti in una stringa composta da token o parole. La distribuzione di frequenza dei bigram è stata ampiamente studiata in molte applicazioni come la linguistica, il riconoscimento vocale o la crittografia. I bigram invisibili sono bigram validi non osservati in un dato insieme di stringhe. È stato osservato che è probabile che gli stessi token in bigram diversi con distribuzioni di aspetto simili siano intercambiabili e formino bigram invisibili. Ad esempio, se osserviamo bigram "una casa", "la casa", "un albero", "l'albero" e "una macchina", possiamo dedurre che "l'auto" è un bigram invisibile. L'idea di "sostituzione" presentata da bigram invisibili può essere adattata per il link prediction al fine di ridurre il rumore sostituendo un nodo con i suoi nodi più simili.

L’ultima tecnica che analizziamo è quella del filtering, originariamente chiamato clustering. La rimozione dei collegamenti più deboli (di solito, quelli osservati tra i nodi con un numero ridotto o senza vicini condivisi) potrebbe aiutare a migliorare i risultati ottenuti dai metodi di link prediction a causa della riduzione del rumore associato. La maggior parte delle tecniche assegna un punteggio S(x,y) ai collegamenti non osservati, ma possono essere applicate ai collegamenti osservati per stimarne la forza. Pertanto, l'approccio di filtraggio consiste nell'applicare qualsiasi tecnica di link prediction che assegni un punteggio a ciascuna coppia di nodi tra i nodi connessi per pesare il collegamento osservato e rimuovere la frazione di collegamenti più deboli per ottenere una rete ripulita.

**Capitolo 3**

Implementazione

Il link prediction mira a dedurre i collegamenti mancanti o prevedere quelli futuri sulla base di reti parziali attualmente osservate, è un problema fondamentale nella scienza delle reti con enormi applicazioni del mondo reale. Tuttavia, gli approcci di link prediction convenzionali non hanno un'elevata precisione di previsione né sono in grado di rivelare le informazioni nascoste dietro i collegamenti. Per risolvere questo problema, generalizziamo le ultime tecniche di deep learning sui grafi e presentiamo un modello di link prediction bastato sulle Graph Attention Network (GAT).

Invece di apprendere la rappresentazione del nodo con le informazioni sull'etichetta del nodo, il modello utilizza i collegamenti come informazioni supervisionate.

Gli esperimenti sui grafi mostrano che non solo può ottenere l'accuratezza del link prediction allo stato dell'arte, ma anche acquisire le rappresentazioni efficienti dei nodi e la centralità dei nodi classificandoli come sottoprodotti. Sebbene le rappresentazioni siano ottenute senza alcuna informazione sull'etichetta del nodo supervisionato, continuano a funzionare bene nelle attività di ranking dei nodi e di classificazione dei nodi.

1. 1. Introduzione

Molti dati del mondo reale arrivano naturalmente sotto forma di relazioni a coppie, come l'interazione proteina-proteina nella cellula umana, citazioni sui paper nella ricerca scientifica e interazione farmaco-bersaglio nella scoperta della medicina. Questi collegamenti contengono informazioni dettagliate sulle proprietà dei nodi, le strutture di rete e l'evoluzione della rete. Prevedere l'esistenza di una relazione è un compito fondamentale nella scienza delle reti e di grande importanza nella pratica. Per le reti in biologia come la rete di interazione proteina-proteina, la rete metabolica e le reti alimentari, la scoperta e la convalida dei collegamenti richiedono uno sforzo sperimentale significativo.

Invece di controllare ciecamente tutti i collegamenti possibili, il link prediction può aiutare gli scienziati a concentrarsi sui collegamenti più probabili e quindi a ridurre drasticamente il costo sperimentale. Per WWW, social network e reti di citazioni, il link prediction può aiutare a consigliare pagine pertinenti, a trovare nuovi amici o a scoprire nuove citazioni.

I metodi convenzionali di link prediction possono essere suddivisi in diverse categorie.

Gli approcci di link prediction locali fanno previsioni basate sul presupposto che due nodi hanno maggiori probabilità di essere collegati se hanno molti vicini comuni. Questi metodi basati sulla similarità locale sono veloci e altamente paralleli poiché considerano solo la struttura della rete locale. Ma la loro accuratezza di previsione è molto bassa, specialmente quando le reti sono sparse e grandi.

Gli approcci di link prediction globale prendono in considerazione la somiglianza strutturale dell'intera rete, questi metodi hanno una maggiore precisione di link prediction rispetto a quelli locali, ma hanno un problema di elevata complessità computazionale che ne impedisce l'applicazione su grafi che contengono milioni e miliardi di nodi. Esistono anche alcuni approcci probabilistici e statistici che presuppongono l'esistenza di una struttura precedente nota della rete, come le strutture gerarchiche o circolari. Ma questi metodi non possono superare il problema della bassa accuratezza del link prediction. Inoltre, difficilmente possiamo estrarre la struttura di rete nascosta e le proprietà dei nodi dagli approcci di link prediction convenzionali sopra menzionati.

Recentemente, c'è stata un'ondata di algoritmi che cercano di fare link prediction attraverso l'apprendimento della rappresentazione di rete che decodifica automaticamente le informazioni strutturali locali e globali dai grafi. L'idea alla base di questi algoritmi è apprendere una funzione di mappatura che, facendo embedding dei nodi come punti in uno spazio a bassa dimensione Rd con il vettore dei nodi, rappresenta le informazioni estratte dal grafo originale.

La maggior parte dei metodi basati sulla rappresentazione di rete sono basati sul metodo Skip-Gram o sulla fattorizzazione di matrici, come DeepWalk, node2vec, LINE e struc2vec.

Tali algoritmi sono indipendenti dalle attività, le rappresentazioni apprese vengono quindi utilizzate per eseguire attività di apprendimento automatico a valle basate su grafi, come la classificazione dei nodi, il ranking dei nodi e il link prediction. Rispetto a quelli convenzionali, questi metodi basati sulla rappresentazione hanno raggiunto un'accuratezza di link prediction molto più elevata.

Ma hanno ancora diversi inconvenienti, per cominciare, non ci sono informazioni supervisionate durante il processo di formazione, non possiamo valutare la qualità dell’embedding a meno che non si eseguano le attività di machine learning a valle. In secondo luogo, i vettori di rappresentazione dei nodi vengono aggiornati direttamente senza considerare la struttura mutevole dinamica delle reti, le strutture di rete non sono statiche, i nodi e gli edge cambiano rapidamente.

Questi algoritmi non possono assegnare un vettore significativo a un nodo appena aggiunto. Inoltre, il potere espressivo è limitato, perché il processo di embedding è fissato dalla strategia di random walk. Infine, le rappresentazioni possono essere difficilmente estese per l'apprendimento induttivo poiché i vettori di embedding non possono essere trasferiti a grafi simili.

Più recentemente, le tecniche di deep learning basate su reti neurali hanno ottenuto trionfi nell'elaborazione delle immagini e del linguaggio naturale. Ciò stimola le estensioni dei metodi sulle strutture dei grafi per eseguire attività di classificazione dei nodi e previsione dei link convertendo le strutture di rete in rappresentazioni a bassa dimensione. Ad esempio, graph convolutional network (GCN) prende in prestito il concetto di convoluzione dalla rete neurale convoluzionale (CNN) e coinvolge il grafo direttamente secondo la struttura di connettività del grafo.

Successivamente, Velickovic propone graph attentional network (GAT) e ottiene l'accuratezza allo stato dell'arte nel compito di classificazione dei nodi, utilizzando l’apprendimento induttivo e quello trasduttivo su diverse network. Seguendo il meccanismo dell'auto-attenzione, GAT calcola la rappresentazione di ogni nodo combinando i suoi vettori di vicinato in modo adattivo. L'attenzione qui è sui pesi regolabili su diversi nodi vicini che possono essere aggiornati dinamicamente in base agli stati dei nodi all'interno di un quartiere connesso locale.

Tuttavia, quegli algoritmi sopra menzionati e le loro estensioni hanno il problema della scalabilità poiché accettano l’intero grafo come input ed espandono ricorsivamente i quartieri attraverso i livelli. Questa espansione è computazionalmente costosa soprattutto quando i grafi diventano grandi.

A causa della proprietà free scala in molti grafi, quando i nodi hub vengono campionati come vicini di primo ordine, i loro vicini di secondo ordine di solito riempiono rapidamente la memoria che porta al problema del collo di bottiglia della memoria. Questo problema impedisce l'applicazione di GAT e GCN su reti su larga scala.

GrapSAGE cerca di risolvere il problema del collo di bottiglia della memoria campionando un intorno di dimensione fissa durante ogni iterazione, dopodiché esegue un aggregatore specifico sull'estrattore di funzionalità.

La strategia di campionamento in GraphSAGE produce risultati impressionanti prestazioni sul compito di etichettatura dei nodi su diverse reti su larga scala.

FastGCN propone quindi di visualizzare GCN come integrale trasformazioni di funzioni di embedding sottomisura di probabilità. La precisione della classificazione è altamente comparabile con la GCN originale mentre elimina la dipendenza dai dati del test. Tuttavia, la maggior parte dei metodi basati sulla convoluzione del grafo applica principalmente le etichette dei nodi invece dei collegamenti come informazione supervisionata.

Le etichette dei nodi, tuttavia, sono sempre scarse nella maggior parte delle reti reali, ci sono solo diversi tipi di reti che hanno informazioni sull'etichetta del nodo, come la rete interattiva proteina-proteina, reti di citazioni e così via.

Inoltre, i collegamenti, piuttosto che gli attributi del nodo, contengono informazioni molto più ricche sulla struttura e l'evoluzione della rete. Secondo la somiglianza e la teoria della popolarità, i collegamenti all'interno di una rete non solo contengono la somiglianza dei nodi, ma codificano anche le informazioni sulla popolarità dei nodi.

Prendiamo ad esempio la formazione di una rete di citazioni, i giornali tendono a citare letterature che non solo sono più somiglianti per quanto riguarda il contenuto ma hanno anche una maggiore popolarità.

I link contengono molte più informazioni sulla struttura rispetto alle etichette dei nodi. E il potere di rappresentazione è fortemente influenzato dalle informazioni supervisionate, quindi, per la maggior parte del machine learning a valle, attività come il link prediction, la visualizzazione e il rilevamento della comunità, i collegamenti, anziché le etichette dei nodi, dovrebbero essere utilizzati come informazioni supervisionate perché codificano almeno sia popolarità che somiglianza.

Proponiamo un nuovo modello che estende l'architettura GAT da applicare sulla previsione del collegamenti mancanti su varie reti. Adottando il meccanismo di attenzione, non solo possiamo fare previsioni sui collegamenti ma anche apprendere rappresentazioni di nodi significative che possono essere utilizzate per identificare le categorie di nodi, specialmente quando i nodi etichettati sono di riserva.

I pesi attenzionali appresi possono aiutarci a formare grafi diretti e ponderati da quelli non diretti e non ponderati. I pesi attenzionali mostrano anche il potenziale nella valutazione dell'importanza dei nodi e nella misurazione della centralità del nodo.

Nell'originale GAT il modello non può essere applicato direttamente all'attività di link prediction supervisionato per i seguenti motivi.

Primo, la complessità del compito di classificazione è O(N), mentre la complessità del link prediction è O(N2), dove N è il numero di nodi.

Rispetto con la classificazione dei nodi, il link prediction di solito implica un calcolo della caratteristica del nodo molto più grande, che porta al problema del collo di bottiglia nella memoria.

In secondo luogo, il modello GAT originale richiede l'accesso all'intera rete durante la classificazione dei nodi inferenza, ma non possiamo applicare direttamente questa strategia per collegare l'attività di previsione a causa della ben nota proprietà scale-free nelle reti reali, poiché l'espansione del vicinato di un nodo hub può riempire rapidamente gran parte del grafo.

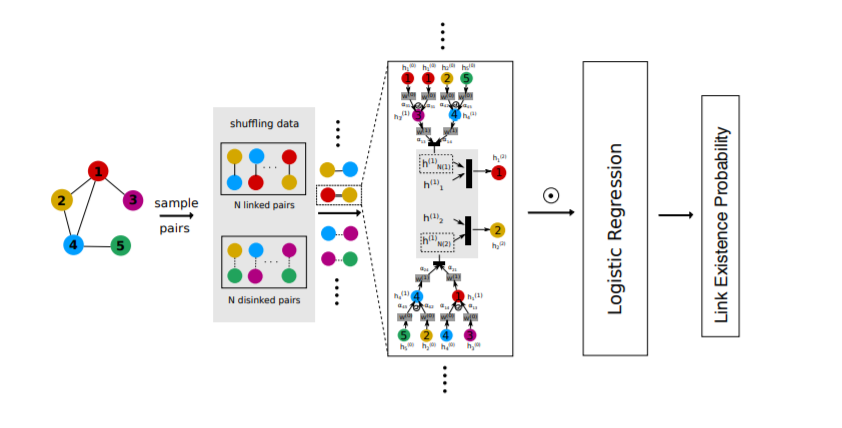
In fine, però sono preferibili mini-batch di grandi dimensioni per ridurre i costi di comunicazione, poiché in pratica possono rallentare il tasso di convergenza poiché la diminuzione della dimensione del mini-batch aumenta tipicamente il tasso di convergenza nel processo di ottimizzazione.

Considerando che, in un modello GAT originale, un piccolo mini-batch di solito coinvolge una grande quantità di nodi, il che riduce il tasso di convergenza e di solito porta a scarse prestazioni nella precisione del link prediction.

Cambiare i vicini in ogni epoca di solito rallenta la convergenza e provoca oscillazioni in fase di training. Il nostro modello è una nuova struttura che combina sia GAT che GraphSAGE, ed è particolarmente progettata per l’attività di link prediction.

Questo modello calcola le rappresentazioni nascoste di ogni nodo attraverso un meccanismo di attenzione condiviso attraverso il suo vicinato. Un gran numero di esperimenti viene implementato su due reti rappresentative.

L’architettura del modello viene rappresentate nella figura sottostante.



Prendiamo come esempio una semplice rete a cinque nodi, verrà considerata qualsiasi relazione di collegamento a due nodi. Descriviamo le relazioni collegate in linee continue e quelle non collegate in linee tratteggiate. Prendiamo quindi il nodo 1 collegato in rosso e il nodo 2 in giallo come esempio di addestramento. Campioniamo prima i nodi 3 e 4 come loro vicini di primo ordine, quindi i nodi 1,2,5 come loro vicini di secondo ordine, i nodi 1,2,5 sono anche i vicini di primo ordine dei nodi 3 e 4.

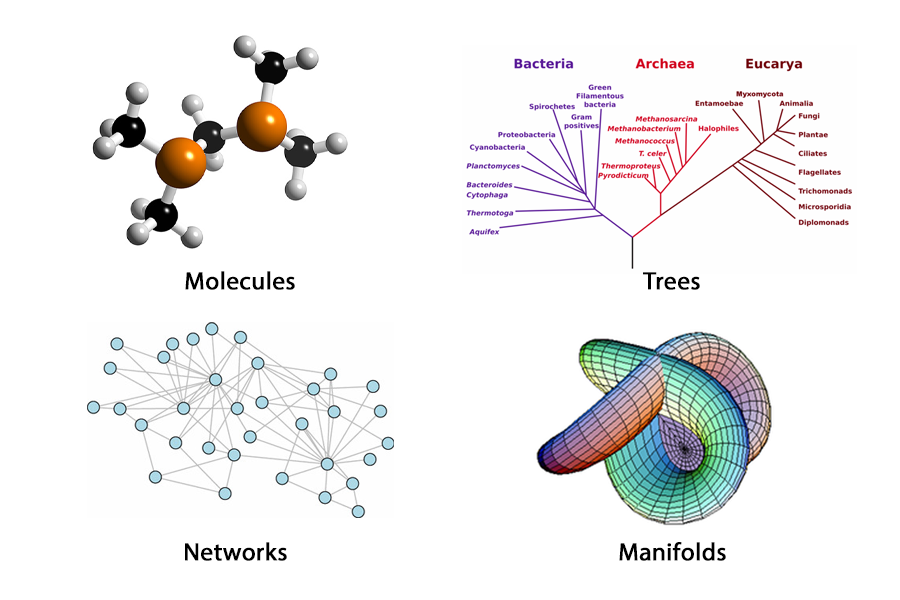
Dopodiché calcoliamo le rappresentazioni vettoriali dei nodi 1 e 2 in base ai loro attributi iniziali e alle caratteristiche iniziali dei loro vicini tramite architettura GAT.

Acquisiamo il vettore dell’edge calcolando il prodotto interno delle rappresentazioni vettoriali di 1 e 2. Infine, viene applicata una funzione di regressione logistica per calcolare la probabilità di esistenza del collegamento.

* 1. Geometric deep learning

La stragrande maggioranza del deep learning viene eseguita su dati euclidei. Ciò include i tipi di dati nel dominio 1-dimensionale e 2-dimensionale. Ma non esistiamo in un mondo 1D o 2D. Tutto ciò che possiamo osservare esiste in 3D e i nostri dati dovrebbero riflettere questo. È quasi ora che il machine learning raggiunga il nostro livello. Immagini, testo, audio e molti altri sono tutti dati euclidei.

I dati non euclidei possono rappresentare elementi e concetti più complessi con maggiore precisione rispetto alla rappresentazione 1D o 2D. Ecco un esempio di dati non euclidei.



Quando rappresentiamo le cose in modo non euclideo, diamo loro un bias induttivo. Ciò si basa sull'intuizione che, forniti i dati di tipo, formato e dimensione arbitrari, è possibile dare la priorità al modello per apprendere determinati modelli modificando la struttura di tali dati. Nella maggior parte delle attuali attività di ricerca e letteratura, il bias induttivo utilizzato è relazionale.

Basandosi su questa intuizione**, Geometric Deep Learning (GDL)** è il campo di nicchia sotto l'ombrello del deep learning che mira a costruire reti neurali in grado di apprendere da dati non euclidei.

Il primo esempio di un tipo di dati non euclideo è un grafo. I **grafi** sono un tipo di struttura dati che consiste di **nodi** che sono collegati con **edge**. Questa struttura di dati astratta può essere utilizzata per modellare quasi tutto. **I grafi ci consentono di rappresentare le singole caratteristiche, fornendo anche informazioni sulle relazioni e sulla struttura.**

Per esempio, concreto di come Graph Learning può migliorare le attività di machine learning esistenti**,** possiamo guardare alle scienze computazionali. Uno dei colli di bottiglia della chimica computazionale, della biologia e della fisica sono i concetti, le entità e le interazioni di rappresentazione. La natura della scienza è empirica ed è quindi il risultato di molti fattori e relazioni esterne.

Ecco alcuni esempi di dove questo è più ovvio: reti di interazione delle proteine, reti neurali molecole, diagrammi di Feynman, mappe cosmologiche.

I nostri attuali metodi di rappresentazione computazionale di questi concetti possono essere considerati "con perdita", poiché perdiamo molte informazioni preziose. Ad esempio, l'utilizzo di una stringa SMILE (simplified-molecular-input-line-entry-system) per rappresentare le molecole è facile da calcolare, ma a scapito delle informazioni strutturali della molecola. Trattando gli atomi come nodi e i legami come edge, possiamo salvare informazioni strutturali che possono essere utilizzate a valle nella previsione o nella classificazione. Quindi, invece di utilizzare una stringa che rappresenta una molecola come input per una rete neurale ricorrente (RNN), possiamo utilizzare il grafo molecolare come input per il suo equivalente geometrico.

Come esempio di come **Geometric Deep Learning ci consente di imparare da tipi di dati mai usati prima,** consideriamo una persona che posa per una fotocamera. I nostri algoritmi attuali, vale a dire Convolutional Neural Networks (CNN), sono piuttosto bravi a prevedere etichette come la persona in posa e / o il tipo di pose data solo da un'immagine 2D. **La difficoltà sorge quando le pose diventano estreme e l'angolo non è più fisso.**Spesso in un'immagine possono esserci vestiti o oggetti che ostruiscono la visualizzazione di un algoritmo, rendendo difficile prevedere la posa. Ora immaginiamo un modello 3D di questa stessa persona che fa pose.

La CNN può ora essere eseguita sull'oggetto 3D stesso piuttosto che su un'immagine 2D dell'oggetto. Invece di imparare da una rappresentazione 2D, che limita i dati a un singolo angolo prospettico, immaginiamo di poter eseguire una convoluzione direttamente sull'oggetto stesso. Analogamente alle CNN tradizionali, il kernel passerebbe attraverso ogni "pixel" rappresentato come un nodo in una nuvola di punti (fondamentalmente un grafico che avvolge l'oggetto 3D).Ogni angolo e fessura sul modello 3D sarà coperto e le informazioni verranno prese in considerazione.

In breve, la differenza tra le CNN standard, rispetto al suo equivalente geometrico, è la previsione dell'etichetta di un oggetto data una sua immagine, rispetto alla previsione dell'etichetta di un oggetto dato un modello 3D di esso. Man mano che la nostra tecnologia di modellazione, progettazione e stampa 3D migliora, si potrebbe immaginare come ciò produrrebbe risultati molto più accurati e precisi.

La nozione di dimensionalità è già comunemente utilizzata nella scienza dei dati e nell'apprendimento automatico, dove **il numero di dimensioni è correlato al numero di caratteristiche / attributi per esempio / punto forniti in un insieme di dati.** Mentre all'inizio le prestazioni degli algoritmi di machine learning aumentano, dopo un certo numero di funzionalità (dimensioni), le prestazioni si stabilizzano. Questo è noto come **la maledizione della dimensionalità.**

Il Geometric Deep Learning non risolve questo problema. Piuttosto, algoritmi come le graph convolutional network riducono le penalizzazioni delle prestazioni sostenute quando si utilizzano tipi di dati che hanno molte caratteristiche, poiché i **dati relazionali sono considerati tramite bias induttivo e non come una caratteristica aggiuntiva.**

La dimensionalità nel geometric deep learning è solo una questione di dati utilizzati nell'addestramento di una rete neurale. I dati euclidei obbediscono alle regole della geometria euclidea, mentre i dati non euclidei sono fedeli alla geometria non euclidea. Infatti, nella geometrica non euclidea vale la frase: il percorso più breve tra due punti non è necessariamente una linea retta.

Un'immagine composta da pixel ha un'idea di sinistra, destra, su e giù. Si può attraversare l'immagine traducendo ricorsivamente una funzione sull'immagine. Questo è esattamente ciò che fa una CNN. In un grafo, tuttavia, non vi è alcuna nozione di sinistra, destra, su o giù. C'è solo un nodo che è connesso a un numero arbitrario di nodi. Un nodo può anche essere connesso a sé stesso.

Le dimensioni nel senso tradizionale di machine e deep learning esistono ancora nell'uso di dati non euclidei per l'addestramento di reti neurali. Ad esempio, è del tutto possibile avere molte caratteristiche del nodo, dove ogni caratteristica è un'altra dimensione. Ma il termine è usato raramente in letteratura per rappresentarlo.

Il machine learning è incentrato sul deep learning, che a sua volta ruotava attorno a una manciata di algoritmi popolari. Ogni algoritmo è specializzato approssimativamente in un tipo di dati specifico. Proprio come le **RNN** sono state create per i dati dipendenti dal tempo e le **CNN** per i dati di tipo immagine, le graph neaural network **(GNN)** sono un tipo di algoritmo di Geometric Deep Learning creato per grafici e reti.

In breve, il campo del Geometric Deep Learning ha dato **tre contributi principali: p**ossiamoutilizzare dati non euclidei, possiamo massimizzare le informazioni dai dati che raccogliamo, possiamo usare questi dati per insegnare algoritmi di machine learning.

* 1. Graph attention network

Nello sviluppo dell’[intelligenza artificiale](https://www.ionos.it/digitalguide/online-marketing/vendere-online/intelligenza-artificiale-di-cosa-si-tratta-e-a-cosa-serve/) il processo di apprendimento è decisivo. Il [machine learning](https://www.ionos.it/digitalguide/online-marketing/analisi-web/machine-learning-cose-lapprendimento-automatico/) (e il [deep learning](https://www.ionos.it/digitalguide/online-marketing/marketing-sui-motori-di-ricerca/deep-learning/) in particolare) vengono utilizzati per addestrare gli algoritmi e quindi insegnare al software a pensare in modo autonomo. Il riconoscimento facciale, ad esempio, si basa su tecnologie di questo tipo. Le **reti neurali artificiali** sono alla base di molti approcci di machine learning: gli algoritmi sono progettati come una rete di nodi, su ispirazione del modello del sistema nervoso umano.

Un esempio sono le reti neurali convoluzionali (CNN) che sono state applicate con successo per affrontare problemi come la classificazione delle immagini, la segmentazione semantica o la traduzione automatica, dove la rappresentazione dei dati sottostanti ha una struttura a grid.

Queste architetture riutilizzano in modo efficiente i loro filtri locali, con parametri apprendibili, applicandoli a tutte le posizioni di input.

Tuttavia, molte attività interessanti coinvolgono dati che non possono essere rappresentati in una struttura a griglia e che invece si trovano in un dominio irregolare. Questo è il caso di mesh 3D, social network, reti di telecomunicazione, reti biologiche. Tali dati possono solitamente essere rappresentati sotto forma di grafi.

Ci sono stati diversi tentativi in ​​letteratura di estendere le reti neurali per gestire grafi strutturati arbitrariamente. I primi lavori utilizzavano reti neurali ricorsive per elaborare i dati rappresentati nei domini dei grafi come grafi aciclici diretti.

Le graph neural network (GNN) sono state introdotte come una generalizzazione di reti neurali ricorsive che possono trattare direttamente con una classe più generale di grafi, ad esempio, grafi ciclici, diretti e non orientati. Le GNN sono una classe di metodi basati sul deep learning in grado di affrontare la natura non euclidea dei grafi, apprendendo automaticamente la topologia di rete preservando le rappresentazioni vettoriali a livello di nodo dalle reti.

Le GNN consistono in un processo iterativo, che propaga gli stati dei nodi fino all'equilibrio, seguito da una rete neurale, che produce un output per ogni nodo in base al suo stato. Questa idea è stata adottata e migliorata utilizzando unità ricorrenti delimitate nella fase di propagazione.

Nelle GNN i nodi raccolgono informazioni dai loro vicini in quanto si scambiano regolarmente messaggi tra loro. In questo modo, la GNN è in grado di apprendere e le informazioni vengono trasmesse e incluse nelle proprietà del rispettivo nodo.

Un particolare modello di GNN sono le Graph Attention Nettwork (GAT), nuove architetture di reti neurali che operano su dati strutturati a grafo, sfruttando i livelli di auto-attenzione mascherati per affrontare le carenze dei metodi precedenti basati su convoluzioni di grafi o loro approssimazioni. Vengono sovrapposti dei layer in cui i nodi sono in grado di contribuire alle feature dei loro vicini, abilitiamo (implicitamente) la specifica di pesi diversi per i diversi nodi in un vicinato, senza richiedere alcun tipo di operazione costosa sulle matrici (come l'inversione) o in base alla conoscenza in anticipo delle struttura del grafo.

GAT è un'architettura basata sull'attenzione per eseguire la classificazione dei nodi dei dati strutturati in grafo. L'idea è di calcolare le rappresentazioni nascoste di ogni nodo nel grafo, prestando attenzione ai suoi vicini, seguendo una strategia di auto-attenzione. L'architettura dell'attenzione ha diverse proprietà interessanti: l'operazione è efficiente, poiché è parallelizzabile tra coppie di nodi vicini; può essere applicato a nodi di grafo con gradi diversi specificando pesi arbitrari ai vicini; e il modello è direttamente applicabile ai problemi di apprendimento induttivo, compresi i compiti in cui il modello deve generalizzare a grafi completamente invisibili.

Pur essendo nate da poco le GAT sono già state usate in diversi studi, ad esempio per la previsione dell'essenzialità genica, infatti l'identificazione di geni / proteine essenziali è un passo fondamentale verso una migliore comprensione della biologia e della patologia umana. Gli approcci computazionali hanno contribuito a mitigare i vincoli sperimentali esplorando i metodi di machine learning e la correlazione dell'essenzialità con le informazioni biologiche, in particolare le reti di interazione proteina-proteina (PPI), per prevedere i geni essenziali.

Tuttavia, le loro prestazioni sono ancora limitate, poiché le centralità basate sulla rete non sono proxy esclusivi dell'essenzialità e i metodi di machine learning tradizionali non sono in grado di apprendere da domini non euclidei come i grafi. È stato proposto quindi un modello basato sulle GAT che è, appunto, EPGAT.

Data la natura del problema di previsione, gli approcci computazionali spesso applicano algoritmi di machine learning, in particolare metodi di apprendimento supervisionato, come parte della soluzione. In questo senso, l'identificazione di geni e proteine ​​essenziali è considerata un compito di classificazione binaria e gli algoritmi costruiscono un modello di previsione basato su caratteristiche legate all'essenzialità di geni e proteine. La maggior parte dei lavori in questo ambito ha applicato algoritmi di machine learning superficiali, tra cui Support Vector Machines (SVM), naive Bayes e alberi decisionali ricorrenti. Questa metodologia ha indubbiamente contribuito a far avanzare la frontiera della conoscenza sui geni e le proteine ​​essenziali.

Tuttavia, mentre i metodi basati sulla centralità richiedono una conoscenza preliminare per creare buone funzioni di punteggio, i metodi di machine learning superficiali richiedono l'input dello specialista per selezionare proprietà biologiche rappresentative come caratteristiche per il problema di apprendimento, il che non è un compito banale, specialmente quando la conoscenza di queste caratteristiche è ancora in evoluzione e potrebbe non essere completamente chiarito.

I metodi di deep learning, al contrario, evitano il passaggio critico di estrarre caratteristiche artigianali nella progettazione dei metodi di machine learning, apprendendo automaticamente le caratteristiche dai dati di addestramento durante il suo funzionamento. Pertanto, il deep learning non richiede conoscenze o presupposti preliminari riguardo alle caratteristiche biologiche rilevanti dell'essenzialità genica. Inoltre, i metodi di deep learning hanno dimostrato prestazioni eccellenti in un'ampia gamma di problemi di previsione e basati sulla rete in bioinformatica.

Per questo è stato proposto un metodo basato su GAT per la previsione dei geni essenziali che è appunto EPGAT. Questa soluzione opera direttamente sulla struttura del grafo utilizzando GAT e incorpora set di dati multiomici, vale a dire profili di espressione genica, informazioni di ortologia e informazioni di localizzazione subcelulare, come caratteristiche del nodo, che vengono utilizzate collettivamente con le reti PPI nel processo di apprendimento del modello. Si preferisce GAT in questi casi poiché i metodi di machine learning tradizionali, di solito, si basano su procedure di pre-elaborazione per analizzare i dati del grafo e generare un set di dati tabulare. Poiché le informazioni sul vicinato sono importanti in un grafo, l'immissione di ogni nodo nel dataset tabulare risultante dovrebbe contenere le caratteristiche dei nodi vicini. Poiché il numero di vicini per ogni nodo varia, il set di dati risultante sarebbe poco maneggevole. Pertanto, sono desiderabili metodi che trattano dati grafi grezzi. Rispetto ai metodi tradizionali di machine learning l’uso nelle GAT in quest’ambito ha ottenuto risultati migliori.

Un altro campo nel quale è stata impiegata la GAT è per il rilevamento di anomalie su serie temporali multivariate. Il rilevamento delle anomalie su serie temporali multivariate è di grande importanza sia nella ricerca di data mining che nelle applicazioni industriali.

Approcci recenti hanno ottenuto progressi significativi in questo argomento, ma rimangono dei limiti. Uno dei principali limiti è che non catturano esplicitamente le relazioni tra le diverse serie temporali, risultando in inevitabili falsi allarmi.

È stato proposto un nuovo framework auto-supervisionato per il rilevamento di anomalie di serie temporali multivariate per affrontare questo problema. Il framework considera ogni serie temporale univariata come una caratteristica individuale e include due livelli di attenzione del grafo in parallelo per apprendere le complesse dipendenze delle serie temporali multivariate sia nelle dimensioni temporali che in quelle delle caratteristiche. Inoltre, l'approccio ottimizza congiuntamente un modello basato su previsioni e un modello basato sulla ricostruzione, ottenendo migliori rappresentazioni di serie temporali attraverso una combinazione di previsione con timestamp singolo e ricostruzione dell'intera serie temporale.

Precedenti studi sul rilevamento di anomalie multivariate su serie temporali hanno fatto progressi fruttuosi. Ad esempio, è stata proposta una rete di codificatore-decodificatore basata su LSTM che modella le probabilità di ricostruzione della serie temporale normale e gli errori di ricostruzione vengono utilizzati per rilevare anomalie da più sensori. Hundman sfrutta LSTM per rilevare anomalie nelle metriche multivariate di serie temporali di veicoli spaziali sulla base di errori di previsione. OmniAnomaly propone una rete neurale ricorrente stocastica, che cattura i modelli normali di serie temporali multivariate modellando la distribuzione dei dati attraverso variabili latenti stocastiche. Tuttavia, nessuno di questi lavori in letteratura ha affrontato il problema di catturare esplicitamente correlazioni multivariate.

È stato proposto un nuovo framework MTAD-GAT (Multivariate Time-series Anomaly Detection via Graph Attention Network), per affrontare i limiti delle soluzioni precedenti. Questo metodo considera ogni serie temporale univariata come una caratteristica individuale e cerca di modellare esplicitamente le correlazioni tra le diverse caratteristiche, mentre le dipendenze temporali all'interno di ciascuna serie temporale vengono modellate allo stesso tempo.

Gli ingredienti chiave in questo modello sono due livelli GAT, vale a dire il livello di attenzione del grafo orientato alle caratteristiche e il livello di attenzione del grafo orientato al tempo. Il livello di attenzione del grafo orientato alle caratteristiche cattura le relazioni causali tra più caratteristiche e il livello di attenzione del grafo orientato al tempo sottolinea le dipendenze lungo la dimensione temporale. Inoltre, forma congiuntamente un modello basato sulle previsioni e un modello basato sulla ricostruzione per una migliore rappresentazione dei dati delle serie temporali. Questo modello mostra dei risultati migliori rispetto alle altre soluzioni proposte in letteratura, grazie all’uso delle GAT all’interno del modello.

* 1. L’architettura Gat

Per cominciare, esaminiamo l’architettura del modello Gat, su cui si basa principalmente il nostro modello.

Nel nostro caso andremo ad utilizzare quella già presente nel tool fornito da Pytorch Geometric.

Gat accetta un insieme di feature del nodo come input,

h = {}, dove , e N è il numero di nodi, F è il numero di feature dei nodi di input.

Usiamo , per indicare l’output della GAT che contiene anche le feature F’.

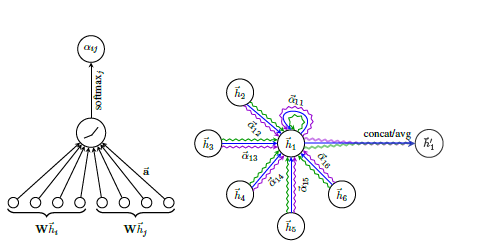
L'obiettivo di GAT è ottenere una potenza espressiva sufficiente per trasferire le funzioni di input in funzioni di output di alto livello.

Per prima cosa applica una trasformazione lineare apprendibile, parametrizzata dalla matrice dei peso, W∈RF’xF per ogni nodo, utilizza quindi una rete neurale feed-forward a strato singolo ∈R2F’ per calcolare il coefficiente di attenzione tra nodi.

Così otteniamo , il nodo *j* è un vicino del nodo *i*, e indica l’importanza delle feature di *j* per *i* tra tutti i vicini di *i*, rappresentati come Ni. Il simbolo || indica la concatenazione mentre l’esponente T indica la trasposta.

Una volta ottenuto il coefficiente di attenzione normalizzato, GAT aggrega le feature dei nodi come una combinazione dei loro vicini, seguito da una funzione sigmoid non lineare .

In fine GAT impiega l’attenzione multi-head per stabilizzare il processo di apprendimento dei coefficienti di attenzione. K indica il numero di head attenzionali, denota i k-esimi pesi attenzionali relativi delle caratteristiche di *j* rispetto *i*. Le caratteristiche di output dei nodi adiacenti sono concatenate o mediate per formare le loro caratteristiche di output finali.



A sinistra c’è il meccanismo di attenzione impiegato dal modello, parametrizzato con un vettore dei pesi , e viene applicata una funzione di attivazione LeakyRelU. A destra invece è illustrato l’attenzione multi-head, in questo caso con head pari a tre per il nodo uno nel suo vicinato. Diversi stili e colori delle frecce indicano calcoli indipendenti dell'attenzione. Le features aggregate di ciascuna head vengono concatenate o mediate.

* 1. L’architettura del modello

Per migliorare l’accuratezza del link predicition, introduciamo il nostro modello, come un encoder-decoder.

Un encoder codifica i nodi in un vettore di rappresentazioni, F’RF’, un decoder che genera un vettore di edge aggregando i vettori dei nodi.

Infine, una funzione score è applicata per valutare la probabilità di esistenza di un link tra i due nodi attraverso il vettore degli edge. Una delle idee chiave consiste nell'apprendere come aggregare le caratteristiche dei nodi in un vettore di edge per l'attività di link prediction.

Il grafo non diretto può essere rappresentato come G = (V, E), dove V denota l’insieme dei nodi e E rappresenta l’insieme degli edge nella network.

Scelti due nodi qualsiasi in modo casuale *i* e *j,* calcoliamo la distanza di Hadamard per rappresentare il vettore degli edge e valutare la probabilità di esistenza di un edge addestrando una funzione di regressione logistica.

Calcoliamo il prodotto di Hadamard tra due vettori di output per generare il vettore degli edge, ottenendo e(i,j)=(, eij è un vettore di dimensione d.

Assumiamo che la probabilità tra due nodi è data dall’output ottenuto applicando la sigmoid al nostro vettore di output contenente il prodotto di Hadamard.

Per quanto riguarda la loss che è usata per fare il train del modello viene presa in considerazione la binary cross entropy (BCE).

Il modello è addestrato in modo tale da predire, non solo l’esistenza dei link, ma anche la loro non esistenza.

Il modello principalmente è composto da tre layer diversi, i primi sono ovviamente i livelli GAT, che si occupano della parte di embedding delle reti, poi un livello che si occupa di calcolare il vettore contenente il prodotto di Hadamard, e infine abbiamo i livelli fully connected composti da livelli lineari con al termine una sigmoid che si occupa di calcolare la probabilità di ogni link in base al vettore di input che riceve. Questi aspetti del modello verranno trattati con maggiore attenzione nei paragrafi successivi.

* 1. Esperimenti

Per valutare le performance del nostro modello e esplorare le potenziali applicazioni abbiamo eseguito diversi esperimenti su due reti molto conosciute nell’ambito delle network.

La prima presa in considerazione è la rete di citazioni Cora che contiene 2708 nodi, 5429 edge, e 7 classi. Ogni nodo presente in Cora è rappresentato da un vettore di 1433 feature, che corrisponde alla rappresentazione di un documento tramite quella che è conosciuta come bag-of-words.

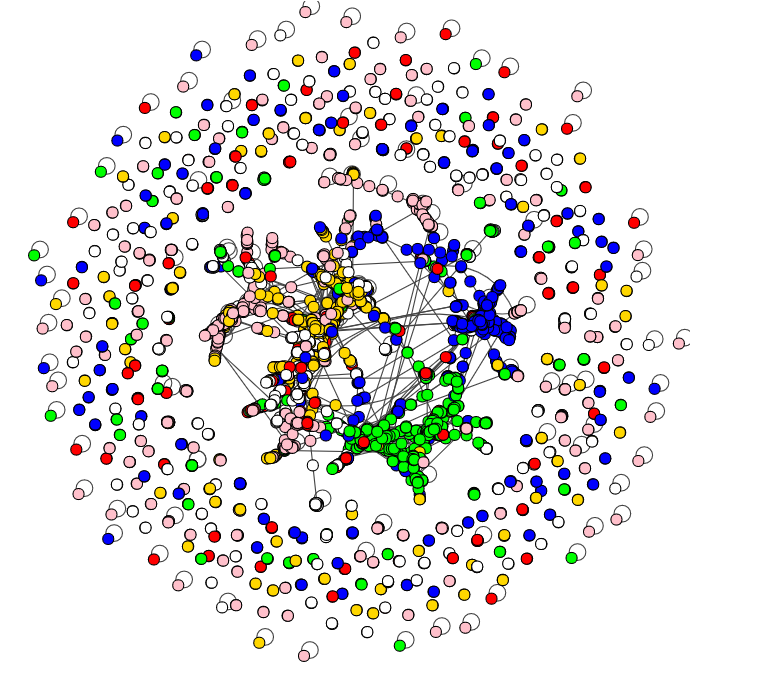
La seconda network presa in esame è CiteSeer che contiene 3327 nodi, 4732 edge, e 6 classi. Qui le feature che rappresentano un nodo sono più del doppio di Cora, infatti abbiamo 3703 attributi estratti dal contenuto dei paper.

Riportiamo sotto la struttura delle network che abbiamo attenzionato.

Immagine che contiene mappa

Descrizione generata automaticamente

Questa è la network Cora, ogni colore indica l’argomento di un relativo paper, più precisamente: Genetic Algorithms in rosa, Reinforcement Learning in bianco, Theory in oro, Rule Learning in arancione, Case Based in verde, Probabilistic Methods in rosso, Neural Networks in blu.



Questo invece è CiteSeer, anche in questo caso i colori indicano nodi di diversa tipologia, Agents in verde, HCI in blu, DB rosa, IR in oro, ML in bianco, AI in rosso.

Come si può notare inizialmente entrambi le reti hanno dei nodi isolati che verranno rimossi quando andremo a considerare solo i relativi LCC.

Per le metriche riguardanti le prestazioni ci siamo basati sull’accuratezza e sulla AUC (l’area sotto la curva ROC), per misurare l’accuratezza nella predizione di un link.

* 1. Modelli usati ed approccio

Per la parte di implementazione abbiamo usato Python e i tools messi a disposizione, ne elenchiamo i principali: pytorch, pytorch geometric e igraph.

Per quanto riguarda i modelli usati per i vari esperimenti, abbiamo usato gli stessi modelli su entrambe le reti per vedere come variavano i risultati.

Abbiamo usato come layer iniziali i livelli GAT che sono forniti nel tool di Pytorch Geometric, ogni livello aveva una funzione di attivazione ELU, mentre nell’ultimo livello non era presente la concatenazione, a differenza dei livelli precedenti.

Per quanto riguarda i livelli fully connected abbiamo usato dei layer lineari con una funzione di attivazione RELU.

Mentre per l’ultimo layer abbiamo usato una sigmoid in modo tale da avere la probabilità di esistenza di un determinato link dati due nodi.

Abbiamo usato il prodotto di Hadamard per confrontare due nodi che erano stati scelti in modo random nella fase di creazione dei dataset di train, validazione e test.

Per quanto riguarda la generazione dei dataset abbiamo applicato lo stesso metodo ad entrambe le network.

Come prima cosa abbiamo considerato solo LCC delle reti, cioè il largest connected componet, in modo tale da rimuovere i nodi isolati che avrebbero portato del rumore nei risultati.

Dopo abbiamo selezionato casualmente dei nodi da questi LCC, per quanto riguarda il grafo di validazione abbiamo considerato il 20% dei nodi, mentre per la fase di test il 10%, stando attenti che fossero set diversi in modo da non avere gli stessi nodi, i restanti invece sono relativi al grafo di train.

Per quanto riguarda i grafi di validazione e test non ci siamo limitati solo a considerare dei nodi random, ma dopo la creazione dei dataset, abbiamo rimosso i link di questi nodi considerati, in modo tale da vedere se anche così il modello riusciva a predire l’esistenza o meno del link.

La creazione del dataset invece è fatta prendendo in considerazione tutti e tre i grafi di train, validazione e test e da questi vendono estratte le features di ogni nodo e gli indici degli edge, inoltre andiamo anche a considerare delle maschere che vengono applicate al vettore in uscita dai livelli GAT e che sono usate per calcolare il prodotto di Hadamard, in quanto servono per sapere il nodo sorgente del link e il nodo target.

Durante la creazione del dataset ci siamo occupati anche del problema del bilanciamento in quanto i link che sono effettivamente reali all’interno delle reti sono una quantità molto ridotta rispetto ai link possibili tra tutti i nodi delle reti. Per ovviare a questo abbiamo considerato alcuni nodi più importanti dai quali abbiamo ricavato i loro link reali, e successivamente abbiamo creato lo stesso numero di link fittizi che non esistono in modo tale da avere un dataset bilanciato, in questo modo abbiamo lo stesso numero sia per quanto riguarda gli edge con etichetta uno, e quindi realmente presenti all’interno della rete, che di edge con etichetta uguale a zero e quindi di edge che non sono reali.

Una volta che abbiamo ottenuto i dataset possiamo eseguire le fasi di addestramento, per far ciò abbiamo usato come ottimizzatore Adam coi seguenti parametri, learning rate=5e-4 e weight decay=5e-4 per un numero complessivo di 100 epoche per ogni modello.

Per quanto riguarda ogni modello è stato generato tenendo conto delle feature di input, che dipendo dalla network (1433 per Cora e 3703 per CiteSeer), delle feature di uscita dai livelli GAT, abbiamo considerato 20 e 40, e ci siamo basati anche sulle Head, da 1 a 4.

Per quanto riguarda i vari livelli GAT non siamo andati oltre i 4 livelli per ogni modello, mentre i livelli lineari non più di 3, in quanto già con questo numero di livelli riuscivamo ad ottenere ottimi risultati, e aumentarli non avrebbe portato ulteriori miglioramenti.

Per quanto riguarda le feature all’interno dei livelli GAT andiamo ad usare un fattore di riduzione che varia, infatti usiamo 1.5, 2, 5 ,10. Questo significa che se ad esempio in ingresso abbiamo circa 1400 feature allora se usiamo un fatto di riduzione di 2 in uscita avremmo circa 700 feature, che corrispondono a quelle in ingresso del livello successivo GAT se presente e se il numero di Head è uguale a 1, altrimenti andremo a moltiplicare le feature in uscita dal livello per il numero di Head e poi potranno andare in ingresso al livello successivo.

* 1. Risultati

Come è stato detto nei paragrafi precedenti sono stati fatti diversi esperimenti su due diverse reti, Cora e Citeseer, con diversi modelli al variare dei loro parametri, con questi esperimenti sono stati ottenuti diversi risultati interessanti per quanto riguarda il link prediction.

I dati che vengono riportati sono le percentuali di accuratezza che abbiamo ottenuto e la AUC per i modelli migliori, quindi quelli con le prestazioni più alte per tutte e due le reti.

Questi risultati vengono anche confrontati con i dati che erano presenti negli studi trovati in letteratura che abbiamo tenuto in considerazione per svolgere la tesi.

Prima di tutto mostriamo i risultati migliori che abbiamo ottenuto su tutte e due le reti.

|  |  |  |
| --- | --- | --- |
| AUC/Accuratezza | Cora | Citeseer |
| Modello | 90/82 | 91/75 |
| GAT | 88/79 | NA |
| DeepLinker | 93/87 | 91/85 |
| GraphSAGE-mean | 89/83 | 90/84 |
| Node2vec | 92/82 | 89/85 |
| LINE | 76/69 | 73/67 |
| RA | 75/41 | 73/32 |

Come si può notare il nostro modello è riuscito ad ottenere ottimi risultati comparabili con i modelli più conosciuti in quest’ambito, ottenendo un AUC di circa il 90% nel modello più prestante.

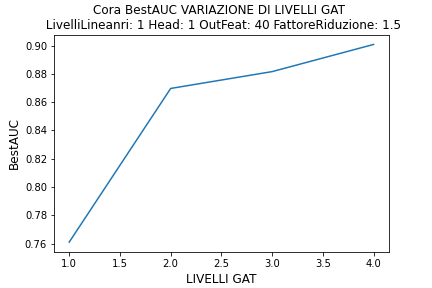
Per mostrare la validità del nostro modello sono state fatte diverse prove al variare dei suo parametri per controllarne la sensibilità, come al solito sempre riferito ad entrambe le reti.

In pratica quello che è stato fatto è stato considerare il modello che ha ottenuto risultati migliori in entrambe le reti, che come verrà mostrato sono modelli con caratteristiche diverse, e poi in questo modello, in entrambi i casi, sono stati fissati tutti i parametri tranne uno, e questo parametro che non è stato fissato è stato fatto variare tra diversi valori per controllare quanto era sensibile alle variazioni il modello.

Per ora concentriamoci sulla prima rete presa in esame che è Cora, successivamente analizzeremo i risultati ottenuti per Citeseer, per poi finire con un loro confronto.

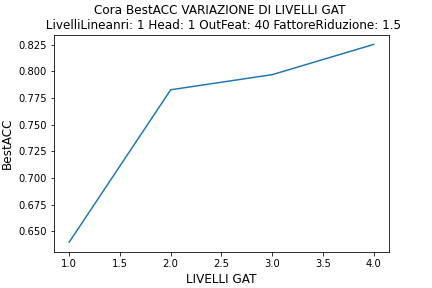
Come detto in precedenza i modelli avevano diversi parametri che vengono riportati per comprenderli meglio, il modello aveva un numero di layer GAT variabili, i nostri esperimenti andavano da un minimo di 1 layer a un massimo di 4, stessa cosa per quanto riguarda i livelli lineari, con un minimo di 1 e un massimo di 3, per quanto riguarda le head i test sono stati effettuati con 1 e 4 head, le feature di output invece erano 20 o 40, mentre i fattori di riduzione erano 1.5, 2, 5 e 10, che come ricordiamo significa che se il fattore di riduzione è 2 e in ingresso abbiamo 1000 features in uscita ne avremo solo 500. Questi parametri valgono come sempre anche per i modelli usati per Citeseer.

Passiamo a quelli che sono i grafici che abbiamo ottenuto per Cora per quanto riguarda l’AUC del modello migliore al variare dei livelli GAT.



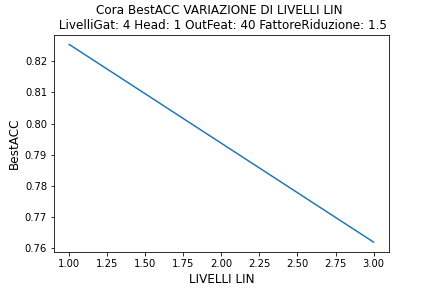
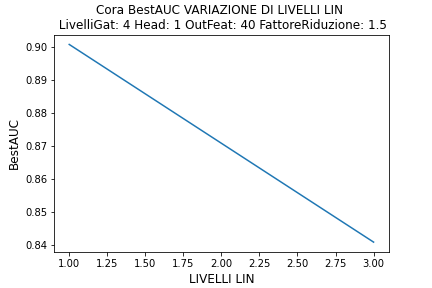
Come si può notare dal grafico il modello che ha ottenuto i risultati migliori è quello con un solo livello lineare, una sola head, 40 feature in uscita dall’ultimo livello GAT e un fattore di riduzione delle feature nei livelli GAT di 1.5.

Il grafico mostra che se il nostro modello è perturbato nei suo livelli GAT e quindi per esempio se li andiamo a diminuire fino ad un solo livello otteniamo un AUC di solo 75%. Quindi in questo modello più livelli GAT abbiamo e meglio performerà.



Qua invece andiamo a considerare quella che è l’accuratezza del modello come si può notare il trend della curva è praticamente identico a quella riferita all’AUC con l’unica differenza che i valori sono di poco più bassi, quindi anche in questo caso più livelli GAT sono presenti all’interno del modello è migliore risulterà l’accuratezza.

Come è stato detto più livelli GAT sono presenti in questo modello e più performa meglio, stessa cosa non si può dire per quanto riguarda i livelli Lineari, infatti sia per quanto riguarda l’AUC e l’accuratezza in entrambi i casi abbiamo una diminuzione.

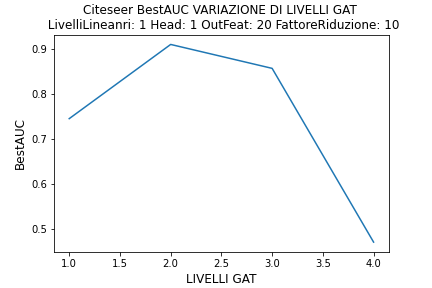


Infatti, come si può vedere dai grafici, passiamo da un AUC del 90% con un solo livello lineare all’85% con tre livelli lineari, stessa cosa per l’accuratezza che passa da 83% a 76%.

Per quanto riguarda le head abbiamo anche in questo caso riscontrato che il loro aumento porta a una diminuzione dei valori di accuratezza e AUC, in maniera molto simile al variare dei livelli lineari.

Ora spostiamo l’attenzione a quelli che sono i migliori risultati ottenuti per quanto riguarda la seconda rete cioè Citeseer.

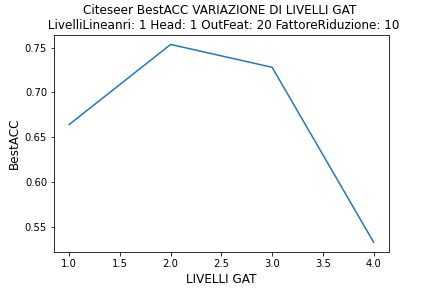
Anche in questo caso analizziamo il modello prima in base alla variazione dei layer GAT.



Come si può notare da questo grafico, il modello migliore è quello con 2 livelli GAT, un solo livello lineare e un solo head, 20 feature in uscita dall’ultimo layer GAT, e il fattore di riduzione massimo cioè 10. Quindi sin da subito notiamo che il modello che performa meglio è decisamente diverso dal modello migliore che era presente in Cora, questo è dato dal fatto che le due reti sono molto diverse tra loto, infatti Citeseer ha decisamente molti più nodi, e molte più feature per ogni nodo, ma ha anche meno link.

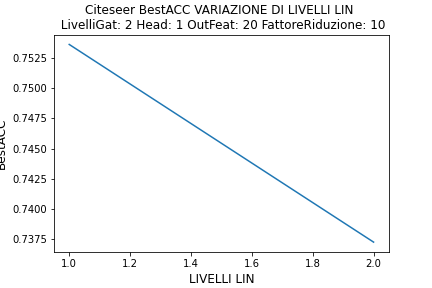
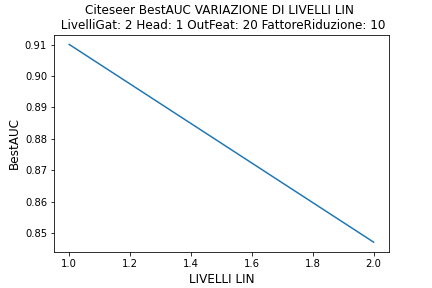
Si può anche notare che qua andare anche oltre ai due layer GAT porta a un peggioramento dei nostri risultati, infatti usando quattro livelli GAT nel modello l’AUC scende fino a circa il 40%, quindi non è sempre detto che aumentare i livelli GAT porti ad un miglioramento.

Viene anche presentato il grafico relativo all’accuratezza che anche in questo caso, come era anche per Cora, è simile a quello dell’AUC.

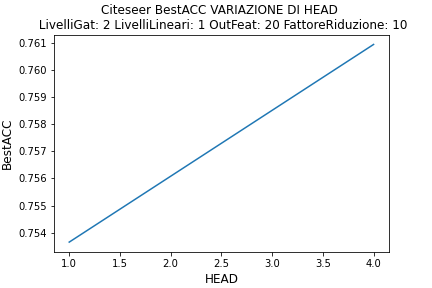
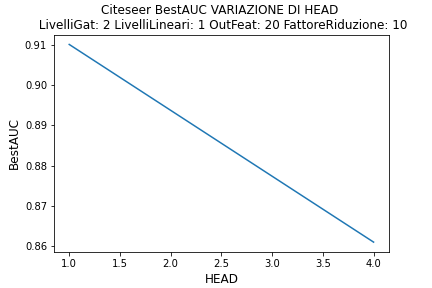


Anche qua più livelli GAT portano a una minore accuratezza.

Ora passiamo al secondo parametro perturbato che sono appunto i livelli lineari. Anche in questo caso come per Cora aumentare i livelli lineari non porta ad un miglioramento.

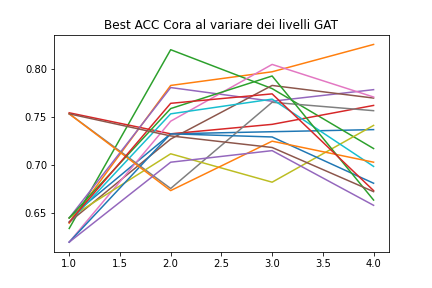
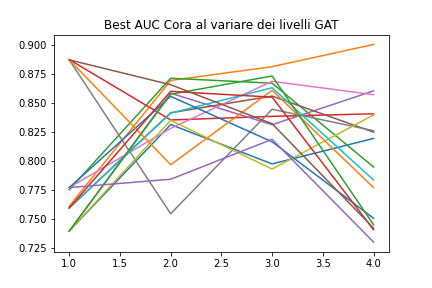


Per quanto riguarda le head abbiamo che il loro aumento causa una diminuzione della percentuale di AUC ma un aumento per quanto riguarda l’accuratezza.

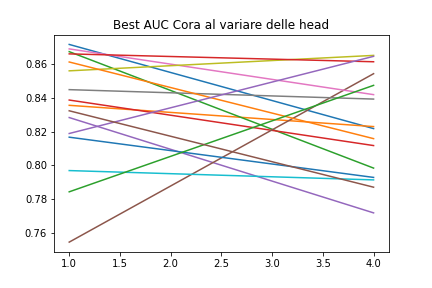


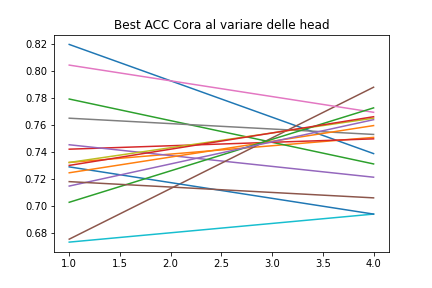
I dati riportati sopra sono relativi a un singolo modello, o meglio un modello per ogni network presa in considerazione, che è stato perturbato nei suoi parametri. Di seguito, invece, vengono mostrati quelli che sono i dati relativi a più modelli considerati nella fase degli esperimenti.

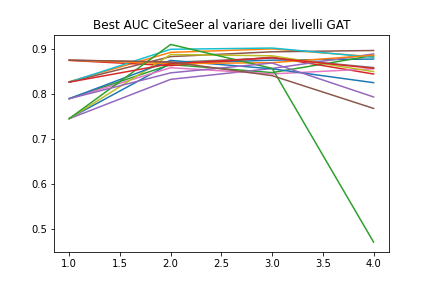
Come prima iniziamo a considerare quelli relativi a Cora.

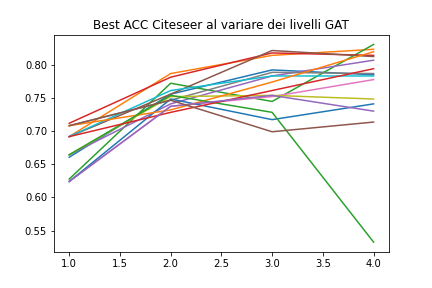


Come si può notare dai grafici considerati, quindi sempre relativi a AUC e accuratezza, otteniamo in media dei buoni risultati se andiamo a considera due o tre livelli GAT nel modello, ma in alcuni modelli, casi rari, succede che anche con un solo livello abbiamo dei buoni risultati, mentre il migliore in assoluto è stato ottenuto con quattro livelli GAT.

Vengono riportati anche i grafici relativi al variare delle head sempre per AUC e accuratezza. Come si può notare qua non si può definire con precisione una tendenza in quanto può succedere che le prestazioni aumentino all’aumentare delle head ma succede anche che diminuiscono.



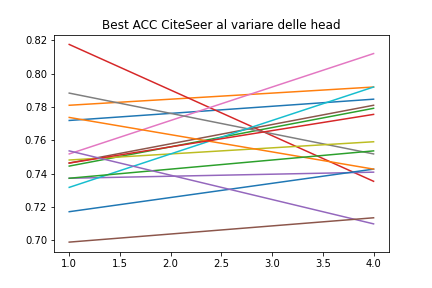
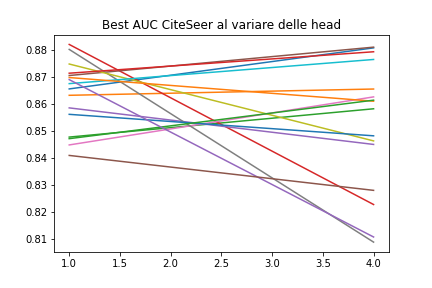
Mostriamo ora gli stessi grafici per Citeseer per far notare una differenza sostanziale per quanto riguarda i risultati ottenuti.



Come si può notare in questi casi l’AUC resta sempre abbastanza elevata anche aumentando i layer GAT, solo in un modello scende drasticamente arrivata a quattro layer, ma è anche il modello che performa meglio quando consideriamo solo due layer GAT.

Il fatto che le prestazioni siano migliori, rispetto a quelle rilevate sulla network Cora, è dovuto al fatto che Citeseer ha il doppio delle feature per ogni nodo ed ha anche più nodi, quindi il modello funziona meglio.

Come per Cora ora mostriamo i grafici relativi al variare delle head.



Anche in questo caso notiamo che il variare delle head non ha una tendenza, perché come prima se aumentano non è detto che accuratezza e AUC aumentino di conseguenza.

# 

# Considerazioni conclusive

Con questa tesi è stata analizzato il problema del link prediction mediante l’uso del deep learning geometrico, più precisamente usando le GAT, per quanto riguarda un problema fondamentale ai giorni nostri e che continuerà ad essere studiato ed analizzato.

È stato posto come obiettivo quello di trovare un modello che riesca a performare la migliore accuratezza nel campo del link prediction, e come mostrano i dati siamo riusciti ad eguagliare, e a volte anche superare, modelli già studiati e consolidati, grazie ai diversi esperimenti, e quindi le diverse combinazioni di modelli, che sono stati testati sulle reti.

Come si sa però questo campo è abbastanza recente, quindi si presta a tantissimi altri studi che potrebbero portare a risultati ancora migliori, e anche a benefici migliore, anche perché non riguarda solo problemi come il recommendation system, ma si pensi per esempio nel campo della medicina per prevedere qualche malattia o per prevedere i collegamenti dei gruppi terroristici e sventare qualcosa di pericoloso e quindi dove ci sono delle vite piuttosto che degli oggetti da comprare.

Potrebbe essere testato anche in futuro su altre reti, differenti sia topologicamente che per le caratteristiche dei nodi, oltre al numero di nodi e dei link.

Uno studio interessante che potrebbe essere fatto in futuro potrebbe riguardare il link prediction senza conoscere le effettive caratteristiche dei nodi e quindi basarsi solo su un aspetto topologico della rete presa in considerazione, e se portasse buoni risultati sarebbe un ottimo traguardo in quanto non sarebbero da considerare tutte le feature dei nodi che portano facilmente al riempimento della memoria, e quindi a un bottleneck che limita l’uso di alcuni modelli in reti molto grandi.

**BIBLIOGRAFIA**

* William L. Hamilton, Rex Ying, Jure Leskovec. Representation Learning on Graphs: Methods and Applications. 10 Aprile 2018
* Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio. Graph attention networks. 4 Febbraio 2018.
* Weiwei Gu, Fei Gao, Xiaodan Lou, and Jiang Zhang. Link Prediction via Graph Attention Network. 29 Oct 2019
* V. Mart\_\_nez, F. Berzal, J.C. Cubero. A survey of link prediction in complex networks. ACM Computing Surveys 49(4): 69:1-69:33, 2017. DOI 10.1145/3012704.
* Joo Schapke, Anderson Tavares, and Mariana Recamonde-Mendoza. EPGAT: Gene Essentiality Prediction With Graph Attention Networks. 19 Jul 2020.
* Hang Zhao, Yujing Wang, Juanyong Duan, Congrui Huang, Defu Cao, Yunhai Tongy, Bixiong Xu, Jing Bai, Jie Tong, Qi Zhang Microsoft, Key Laboratory of Machine Perception, MOE, School of EECS, Peking University. Multivariate Time-series Anomaly Detection via Graph Attention Network. 4 Sep 2020

**SITOGRAFIA**

* <https://en.wikipedia.org/wiki/Link_prediction>
* <https://snap.stanford.edu/proj/embeddings-www/>
* <https://arxiv.org/pdf/1710.10903.pdf>
* <https://arxiv.org/pdf/1910.04807.pdf>
* <https://arxiv.org/pdf/1709.05584.pdf>
* <https://www.hindawi.com/journals/mpe/2013/125123/>
* <https://dl.acm.org/doi/10.1145/3012704>
* <https://iaml.it/blog/geometric-deep-learning-1>
* <https://www.ionos.it/digitalguide/online-marketing/marketing-sui-motori-di-ricerca/graph-neural-network/>
* https://medium.com/@flawnsontong1/what-is-geometric-deep-learning-b2adb662d91d