

UNIVERSITÀ DEGLI STUDI DI PADOVA

CORSO DI LAUREA TRIENNALE IN
INGEGNERIA DELL'INFORMAZIONE

**Analisi di una rete di autori di
pubblicazioni scientifiche**



Laureando:

Pietro Maria NOBILI

Relatore:

Prof.ssa Cinzia PIZZI

Matricola:

1067941

Correlatore:

Dott. Mattia SAMORY

18 Luglio 2018
Anno Accademico 2017/2018

Indice

Abstract	1
Introduzione	1
1 Community detection	3
1.1 Studi esistenti	3
1.2 Metodi usati	3
1.2.1 Girvan–Newman	3
1.2.2 Clauset–Newman–Moore	4
1.2.3 Stochastic Block Model	4
2 Estrazione dati	5
2.1 Struttura database Microsoft	5
2.2 Processo di estrazione	6
2.2.1 Filtro degli autori per affiliation	8
2.2.2 Unione di ID autore in singoli nodi	9
2.2.3 Errori intrinseci al database	15
3 Troubleshooting	17
4 Risultati	19
4.1 V-measure	19
4.2 Analisi dei risultati	21
4.2.1 Dati generati	21
4.2.2 Analisi	22
5 Conclusioni	27

Abstract

È stato generato un grafo delle pubblicazioni del DEI.
Sono stati cercati cluster nel grafo.
Sono stati confrontati con la struttura delle comunità del dipartimento.

Introduzione

Breve descrizione del community detection e della sua importanza, in generale e nel caso particolare delle comunità di autori di pubblicazioni scientifiche.

Presentazione della struttura della tesi.

Capitolo 1

Community detection

1.1 Studi esistenti

Descrizione delle comunità

Descrizione della community detection.

Descrizione della bibliografia attuale sui grafi di coautori.

1.2 Metodi usati

Per partizionare il set di nodi di un grafo in cluster sono stati utilizzati tre algoritmi: l'algoritmo di Girvan-Newman [2] e l'algoritmo di Clauset-Newman-Moore [1], implementati nella libreria Stanford Network Analysis Platform (SNAP) [3], e l'algoritmo *Stochastic Block Model* [6], implementato in *graph-tool* [7].

1.2.1 Girvan–Newman

Girvan e Newman introducono la *Edge Betweenness* come misura di quanto un arco agisca da ponte fra due comunità in un grafo. Ispirandosi alla *Vertex Betweenness*, che valuta il numero di cammini minimi fra coppie di nodi che passano per il nodo in esame, definiscono la centralità di un arco come numero di cammini minimi tra i nodi del grafo che passano per tale arco.

Per suddividere il grafo in comunità, vengono progressivamente eliminati gli archi con centralità massima, dopo aver ricalcolato i cammini minimi modificati dall'eliminazione degli archi.

Il risultato dell'algoritmo è un dendrogramma, da cui si estrapola la struttura delle comunità del grafo.

Ha complessità $O(E^2N)$ nel caso peggiore, dove E è il numero di edge e N è il numero di nodi, che diventa $O(N^3)$ nel caso di un grafo sparso in cui $N \sim E$.

1.2.2 Clauset-Newman-Moore

Girvan e Newman introducono la *modularity* [4] come valutazione della qualità della suddivisione in cluster dei nodi di un grafo. La *modularity* viene definita come la frazione di edge in un grafo che connettono nodi nella stessa comunità meno la frazione di edge che connetterebbero quei nodi se gli archi del grafo fossero casuali ma la struttura in comunità fosse la stessa. Se la frazione di archi all'interno di una comunità non è maggiore di quella prevista in un grafo casuale, la *modularity* è nulla, se il grafo ha una struttura in comunità ben definita, il valore si avvicina a 1.

L'algoritmo sviluppato da Clauset, Newman e Moore utilizza un approccio *greedy* per trovare una suddivisione dei nodi che massimizzi la *modularity*. Partendo da una suddivisione che consiste in un nodo per comunità, ad ogni iterazione si uniscono le due comunità la cui aggregazione comporta il massimo incremento della *modularity*.

L'algoritmo ha complessità $O(ED \log N)$ dove E è il numero di edge, N è il numero di nodi e D è la profondità del dendrogramma che descrive la struttura delle comunità del grafo. La complessità diventa $O(N \log^2(N))$ nel caso di un grafo sparso in cui $N \sim E$, e con struttura delle comunità gerarchica in cui $D \sim \log N$.

1.2.3 Stochastic Block Model

A differenza dei due metodi descritti nelle sezioni precedenti, l'algoritmo del *Stochastic Block Model* segue un approccio euristico al problema del *community detection*. Inoltre questo algoritmo non favorisce necessariamente la struttura delle comunità formate da nodi strettamente connessi tra loro e meno connessi con membri delle altre comunità.

Il modello stocastico a blocchi generalizza la struttura di comunità di un grafo raggruppando i nodi in blocchi e definendo una matrice delle probabilità di esistenza degli archi tra nodi di ciascun blocco.

Il problema del partizionamento viene quindi convertito in un processo statistico di inferenza dei parametri del modello basandosi sul grafo osservato.

Anche questo algoritmo è implementato con una strategia *greedy* che risulta in una complessità $O(N \ln^2(N))$.

Capitolo 2

Estrazione dati

2.1 Struttura database Microsoft

Il database da cui sono stati estratti i dati è il Microsoft Academic Graph [9], che contiene informazioni relative a pubblicazioni scientifiche, autori, istituzioni accademiche, riviste, conferenze e settori di studio. I record presenti nei file forniscono le relazioni tra queste entità.

Il database è composto da undici file di testo che contengono un record per riga, con i campi separati da tabulazione.

Per il lavoro oggetto di questa tesi sono stati utilizzati in particolare quattro file del database, la cui struttura è illustrata nella tabella 2.1.

L'ultimo aggiornamento del database disponibile risale all'agosto 2015.

Tabella 2.1: Struttura del database

Nome file (Numero record)	Campi
Authors.txt (123.017.489)	Author ID Author Name
PaperAuthorAffiliations.txt (325.498.063)	Paper ID Author ID Affiliation ID Original affiliation name Normalized affiliation name Author sequence number
Affiliations.txt (2.719.436)	Affiliation ID Affiliation name
Papers.txt (122.695.085)	Paper ID Original paper title Normalized paper title Paper publish year

	Paper publish date
	Paper Document Object Identifier (DOI)
	Original venue name
	Normalized venue name
	Journal ID mapped to venue name
	Conference series ID mapped to venue name
	Paper rank

2.2 Processo di estrazione

Dal sito di dipartimento (<http://www.dei.unipd.it/lista-docenti>) sono stati estratti i nomi degli attuali afferenti DEI, includendo Docenti, Assegnisti di ricerca, Collaboratori di ricerca e Dottorandi.

Si è reso necessario intervenire manualmente sulla lista per risolvere alcune criticità:

- I caratteri Unicode non sono ben gestiti nel database, a volte si trova la lettera non accentata, altre volte è presente un apostrofo invece dell'accento, in altri casi il carattere è assente del tutto. I nomi che contengono tali caratteri sono stati inseriti in tutte le varianti identificate.
- Afferenti con nomi composti uniti da trattino o con molteplici secondi nomi sono stati inseriti in multiple varianti

La lista ottenuta comprende 379 autori. Dove presente, è stato estratto il Settore Scientifico Disciplinare dell'afferente, che ha fornito la partizione in classi utilizzata come riferimento alla fine dell'elaborazione dei dati.

I nomi propri degli autori sono stati abbreviati in tutte le possibili combinazioni per rispecchiare la struttura del database Microsoft, ed è stato creato un file con la struttura seguente:

Tabella 2.3: PersoneComunitaDEI.txt

```

a a pietracaprina          INF/01 - INFORMATICA
a alberto pietracaprina    INF/01 - INFORMATICA
andrea a pietracaprina     INF/01 - INFORMATICA
andrea alberto pietracaprina INF/01 - INFORMATICA
...
```

Utilizzando questa lista di nomi ed abbreviazioni, sono stati estratti dal file *Authors.txt* le coppie (ID autore, nome autore), che risultano essere 8.135, con una media di 21,5 ID per nome.

Con il set di ID autori ottenuto, sono stati estratti dal file *PaperAuthorAffiliations.txt* i record relativi ai paper, nella forma (ID paper, ID autore, ID affiliation), per un totale di 62.291 paper.

Dalla lista così ottenuta, non necessariamente ordinata, sono stati creati gli edge fra ID autore, che sono poi stati aggregati creando una edge list pesata, come indicato in 2.4

Tabella 2.4: Creazione edge

IDpaper1	IDautore1				
IDpaper1	IDautore2		IDautore1	IDautore2	1
IDpaper1	IDautore3	→	IDautore1	IDautore3	1
IDpaper2	IDautore2		IDautore2	IDautore3	2
IDpaper2	IDautore3				

A partire da questi edge è stato generato il primo grafo che rappresenta la rete degli autori.

Questo metodo di creazione del grafo, che considera le collaborazioni tra autori, riduce a 778 il set di ID autori per un totale di 287 nomi univoci. Il grafo è formato da 1.830 edge con 7.803 di peso totale.

I grafi in questo capitolo sono stati divisi in comunità utilizzando l'algoritmo Modularity di Gephi.

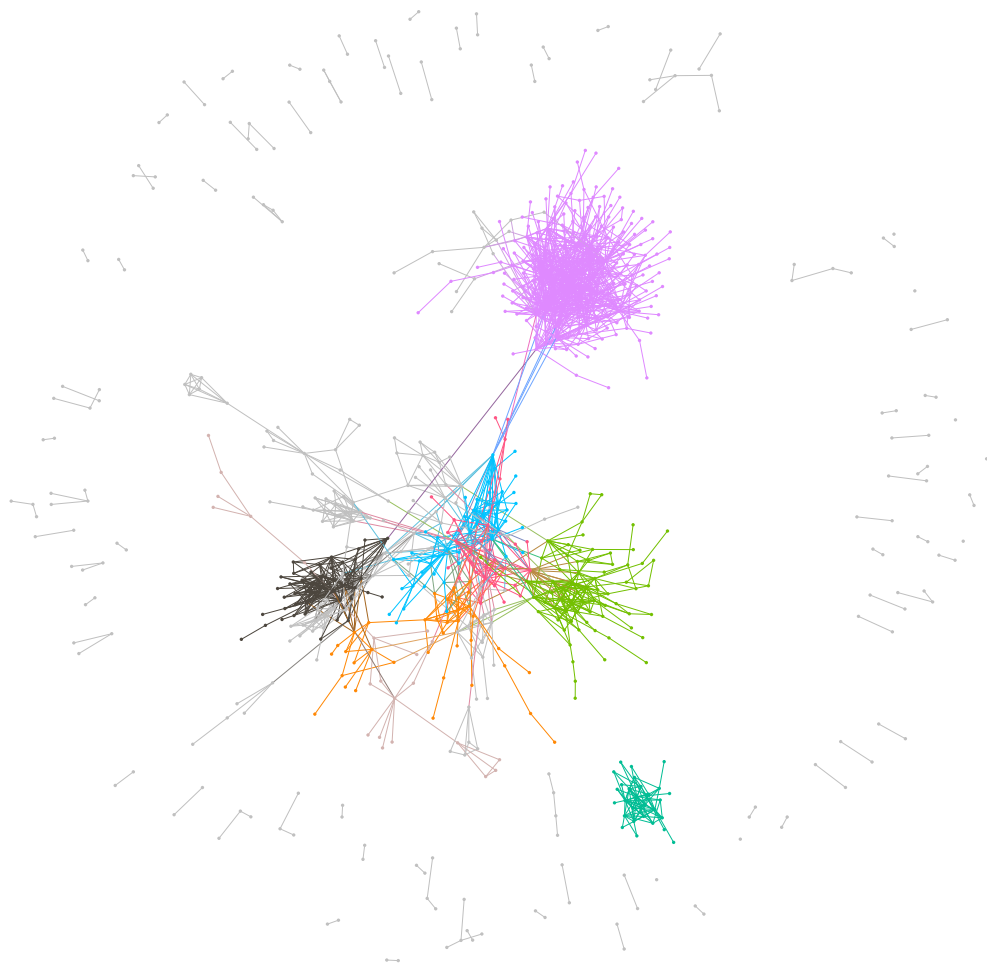


Figura 2.1: Grafo collaborazioni

Analisi della prima estrazione

I due problemi principali di questo processo emergono al momento dell'estrazione degli ID autore dal file *Authors.txt*:

- Vengono selezionati anche gli ID riferiti ad autori omonimi, non affiliati al DEI.
- Ad un singolo autore DEI sono associati più ID autore.

2.2.1 Filtro degli autori per affiliation

Il primo problema è già in parte risolto dal metodo di creazione del grafo, che considera solo gli ID autori che hanno almeno una collaborazione con un altro ID nel set. In questo modo più del 90% degli ID viene filtrato.

Ispezionando manualmente il grafo, si rileva la presenza di due nodi riferiti a "c pizzi", che però sono connessi ad autori con cui la professoressa Pizzi non ha mai collaborato. Analizzando gli ID, le loro collaborazioni e i paper a cui hanno collaborato, si è dedotto che sono ID relativi probabilmente a Carmine Pizzi e Claudia Pizzi,

medici rispettivamente a Bologna e Milano. Questi ID sono stati inclusi perché risultano collaborazioni con altri autori il cui nome è presente nella lista di partenza estratta dal sito di dipartimento.

È stato sviluppato un secondo metodo di estrazione dei dati che esclude gli ID autore se non hanno mai pubblicato un paper a Padova, considerando le informazioni sulle affiliation dei paper, come viene illustrato di seguito.

- Dal file *Affiliations.txt* si estrae la lista delle affiliation in cui risulta nel nome un match all'espressione regolare “*pad(ov|u)a*”.
- Dalla lista di terne (ID paper, ID autore, ID affiliation) si mantengono solo quelle in cui l'ID affiliation compare nel set di affiliation padovane appena estratte.
- Dalle terne selezionate, si estrae un set di ID autori che hanno pubblicato almeno un paper con affiliation padovana.
- Si estraggono i paper scritti da questi ID autore e si procede alla generazione degli edge pesati.

Applicando questo metodo, gli ID autore si riducono a 306, relativi a 201 nomi univoci. Il grafo generato include 850 edge con un peso totale di 5.195.

Gli ID autore riferiti a "c pizzi", non avendo neanche un paper con affiliation padovana, vengono correttamente filtrati.

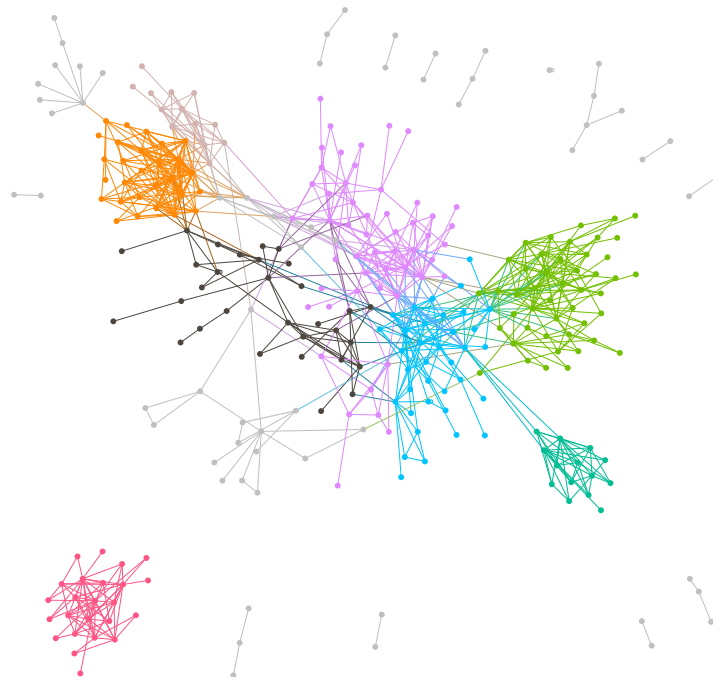


Figura 2.2: Grafo Padovani

2.2.2 Unione di ID autore in singoli nodi

Per risolvere il secondo problema insorto nell'estrazione dei dati, per cui ad una singola persona fisica, effettivamente affiliata al DEI,

possono essere associati più ID autore, sono state seguite due vie alternative fra loro.

Il primo metodo tiene conto solo dei nomi associati ai nodi del grafo, il secondo metodo sfrutta la struttura del grafo per stabilire se due nodi siano riferiti alla stessa persona.

Per nome

I grafi precedentemente generati vengono rielaborati, ipotizzando che nodi con nomi uguali o le cui abbreviazioni sono uguali possano essere considerati un unico nodo.

In questo modo “*a alberto pietracaprina*” viene associato a “*a a pietracaprina*” ma anche a “*andrea a pietracaprina*” come pure a “*andrea alberto pietracaprina*”.

I nodi del grafo in figura 2.1 si riducono da 778 a 199. Gli edge scendono da 1830 a 661, mantenendo lo stesso peso totale di 7.803.

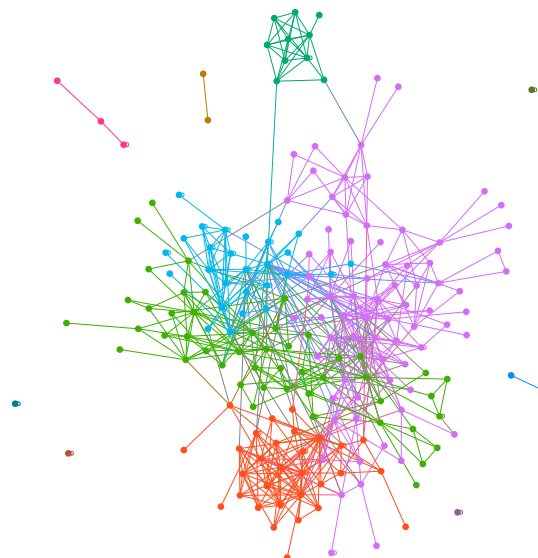


Figura 2.3: Grafo collaborazioni - nodi unificati per nome

I nodi del grafo in figura 2.2 si riducono da 306 a 158. Gli edge scendono da 850 a 463, mantenendo il peso totale di 5.195.

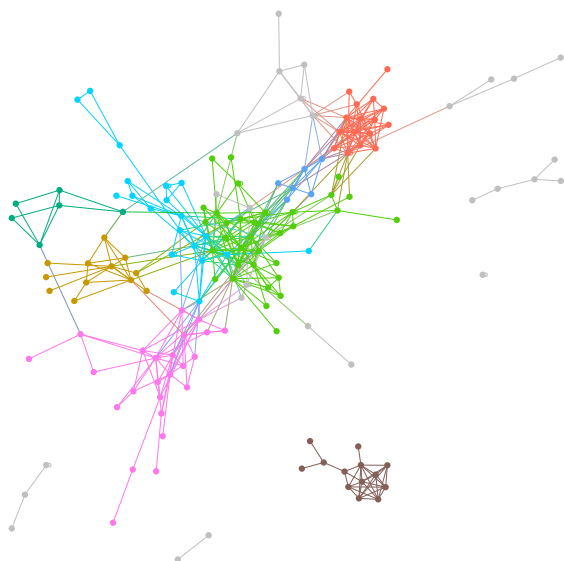


Figura 2.4: Grafo padovani - nodi unificati per nome

Per distanza

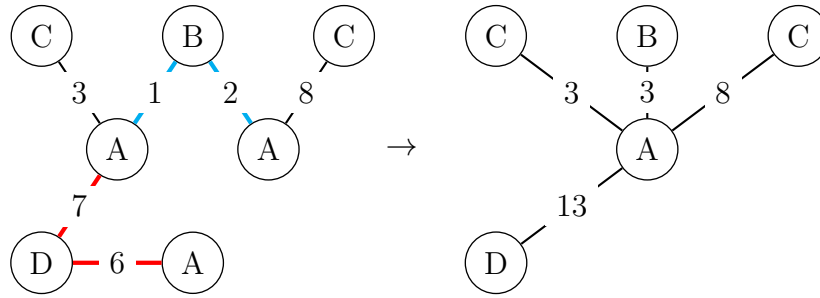
Il metodo che considera i nomi precedentemente descritto introduce una criticità nel caso in cui due autori abbiano un nome che viene abbreviato nello stesso modo. Questo fa sì che più persone vengano assimilate erroneamente in un unico nodo.

Nei dati che sono stati trattati questo succede solo nel caso di “*Mattia Zorzi*” e “*Michele Zorzi*”, che si abbreviano entrambi in “*m zorzi*”.

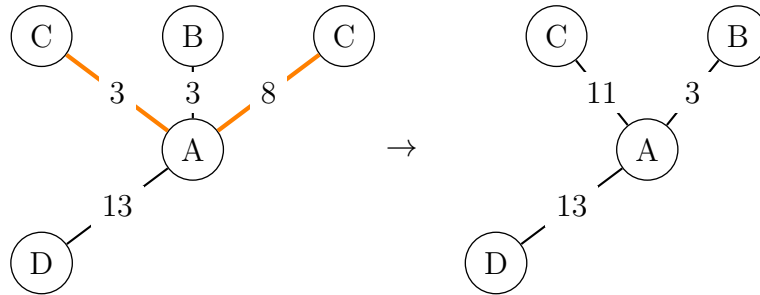
In dataset più ampi oppure relativi a comunità che anche se ristrette presentano una bassa variabilità dei cognomi, il fenomeno dei falsi positivi incide in maniera molto più marcata sulla veridicità del grafo.

Nel secondo metodo sviluppato per unificare i nodi, viene richiesta un’ulteriore condizione per considerare due nodi come relativi alla stessa persona fisica. Oltre a verificare la corrispondenza delle abbreviazioni dei nomi si calcola anche la distanza minima tra i due nodi nel grafo. Se questa distanza è minore di una certa soglia, i due nodi vengono unificati. Questo processo può essere iterato più volte, per sfruttare le informazioni ottenute nei passi precedenti.

Prima iterazione unione nodi con distanza minima minore di 2



Seconda iterazione unione nodi con distanza minima minore di 2



I valori ottimi di soglia e numero di iterazioni sono stati cercati sperimentalmente, ma non è emersa dai risultati una coppia di valori migliore in maniera rilevante rispetto alle altre. Ispezionando manualmente il grafo, si è constatato che ID autore riferiti alla stessa persona risultano generalmente distanti 2. Il procedimento è stato ripetuto tre volte, in modo da sfruttare le informazioni generate nei passi precedenti. Dopo questo numero di iterazioni il grafo raggiunge quasi una situazione di stabilità.

I nodi del grafo in figura 2.1 si riducono da 778 a 336. Gli edge scendono da 1830 a 634, mantenendo lo stesso peso totale di 7.803.

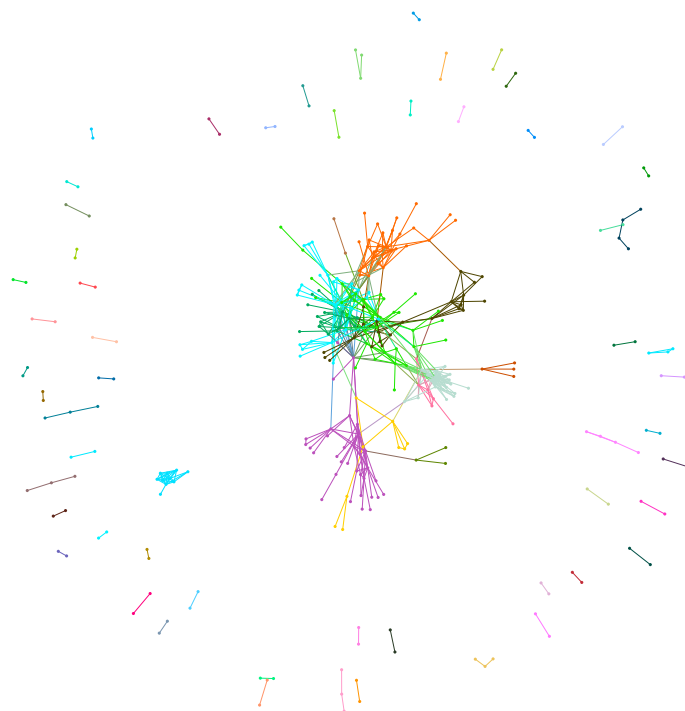


Figura 2.5: Grafo collaborazioni unificati per distanza

I nodi del grafo in figura 2.2 si riducono da 306 a 173. Gli edge scendono da 850 a 439, mantenendo il peso totale di 5.195.

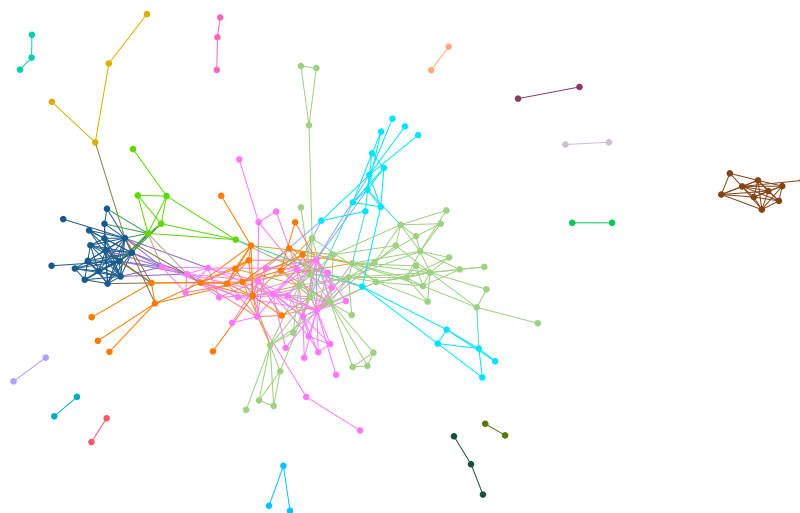


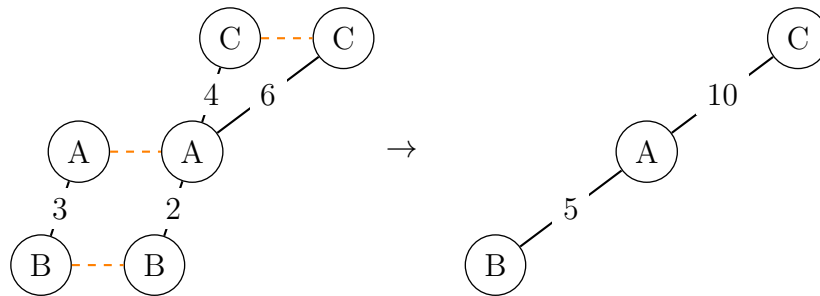
Figura 2.6: Grafo Padovani unificati per distanza

Per nodi adiacenti

Osservando i grafi generati senza unire i nodi, si nota la presenza di un elevato numero di componenti sconnesse formate da 2-4 autori, che sono principalmente effettivi collaboratori tra loro. È stato sviluppato un metodo di deduplicazione dei nodi che si basa sulle collaborazioni tra coppie di autori.

Per ogni edge si estraggono i nomi relativi agli estremi, se due edge hanno gli estremi con i nomi coincidenti, si considerano le coppie di nodi come relative alla stessa persona.

Unione dei nodi per coppie di nodi adiacenti



I nodi del grafo in figura 2.1 si riducono da 778 a 313. Gli edge scendono da 1830 a 615, mantenendo lo stesso peso totale di 7.803.

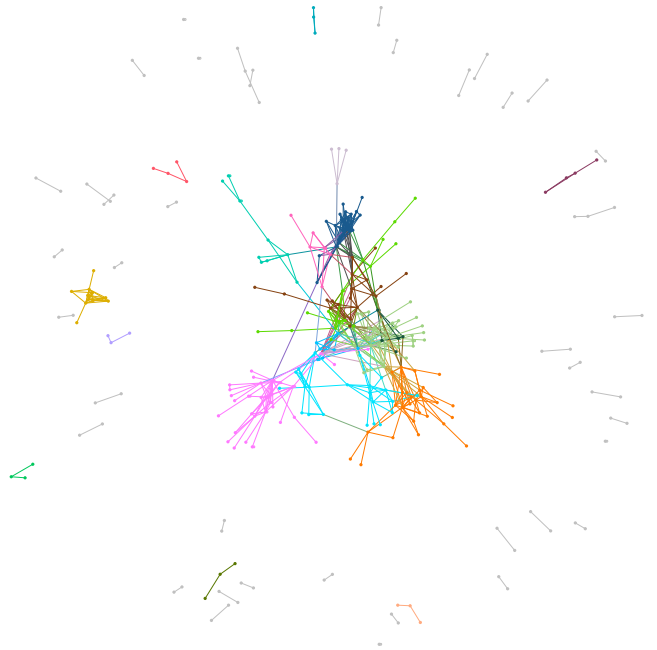


Figura 2.7: Grafo collaborazioni unificati per coppie di nodi adiacenti

I nodi del grafo in figura 2.2 si riducono da 306 a 174. Gli edge scendono da 850 a 453, mantenendo il peso totale di 5.195.

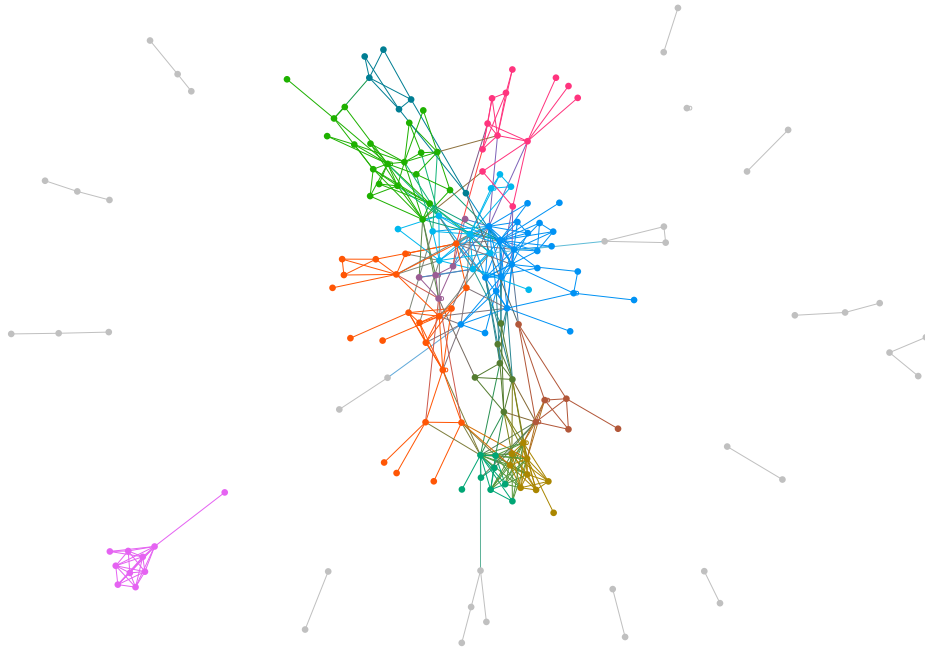


Figura 2.8: Grafo Padovani unificati per coppie di nodi adiacenti

2.2.3 Errori intrinseci al database

Nel corso delle varie iterazioni dell'estrazione dei dati, sono state identificate varie lacune nei record del database, riassunte di seguito

- I record di Paper-Autore-Affiliation non contengono l'ID affiliation in 264.069.014 record su 325.498.062. In particolare per autori con pochi paper, la fase di pruning basata sulle affiliation è molto influenzata da questa mancanza.
- Nei record Paper-Autore-Affiliation, può mancare il record relativo a uno dei coautori.
- Tra i coautori di un paper, possono essere riportati più ID autore riferiti alla stessa persona.
- I nomi con caratteri non ASCII sono trattati in maniera imprevedibile.

Nel prossimo capitolo si propongono dei metodi alternativi di estrazione dei dati per tentare di aggirare questi problemi.

Troubleshooting

In particolare, la lista di partenza include solo gli attuali afferenti DEI, estratti dal sito di dipartimento. Questo comporta la mancanza di professori, non più attivi a Padova, che avevano ruolo di aggregatore di una comunità.

Un metodo generale proposto per identificare anche autori non presenti nella lista degli attuali afferenti DEI è:

- 17

estrarre le terne

5. Eventualmente ridurre il set di autori basandosi sulle affiliation dei loro paper
6. Ripetere i passi 2-5 per un numero predefinito di volte o fino alla convergenza della rete, nel momento in cui non vengono identificati ulteriori coautori

In questo modo si estraggono tutte le comunità relative alle affiliation usate come filtro. Una singola collaborazione con un dipartimento esterno comporta alle iterazioni successive l'inclusione di molti autori di quel dipartimento. Un metodo proposto per risolvere il problema è, alla fine delle iterazioni e della creazione dei cluster, considerare solo quelli che contengono almeno un nome presente nella lista originale: in questo modo dipartimenti esterni, anche se connessi al grafo, non vengono inclusi.

Capitolo 4

Risultati

4.1 V-measure

La *V-measure*, una misura esterna della validità dei cluster basata sull'entropia, presentata da Rosenberg e Hirschberg [8], è stata utilizzata per valutare la qualità delle partizioni dei grafi ricavati con i vari metodi descritti nelle sezioni precedenti.

Come terminologia, si parlerà di due distinte partizioni del set di nodi da analizzare: un set di classi C , considerate la suddivisione reale dei nodi, e un set di cluster K per indicare la partizione generata.

Rosenberg e Hirschberg hanno sviluppato due concetti, l'omogeneità e la completezza, la cui media armonica è la V-measure.

Una partizione di cluster è considerata perfettamente omogenea quando i cluster contengono *solo* membri di una singola classe.

In maniera simmetrica una partizione è perfettamente completa quando un singolo cluster contiene *tutti* i membri di una classe.

Per valutare l'omogeneità di una partizione si considera l'entropia condizionata della distribuzione di classi dato il clustering generato $H(C|K)$. Nel caso di perfetta omogeneità questo valore è nullo in quanto il cluster contiene una singola classe, mentre è massimo e vale $H(C)$ quando la distribuzione delle classi all'interno del cluster è identica alla distribuzione delle classi nell'intero set, per cui il clustering non fornisce alcuna informazione aggiuntiva. Considerando che il valore di entropia è massimo quando la partizione è pessima, si definisce l'omogeneità come

$$h = \begin{cases} 1 & \text{se } H(C) = 0 \\ 1 - \frac{H(C|K)}{H(C)} & \text{altrimenti} \end{cases} \quad (4.1)$$

Analogamente si definisce la completezza come

$$c = \begin{cases} 1 & \text{se } H(K) = 0 \\ 1 - \frac{H(K|C)}{H(K)} & \text{altrimenti} \end{cases} \quad (4.2)$$

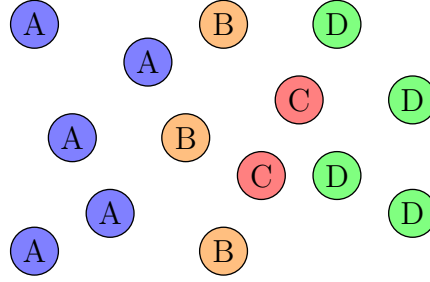
La *V-measure* è definita come la media armonica dei due valori di omogeneità e completezza

$$V = \frac{h \cdot c}{h + c} \quad (4.3)$$

I seguenti diagrammi illustrano graficamente le tre situazioni limite. Le classi sono {A, B, C, D}, i colori rappresentano i cluster.

Clustering perfetto

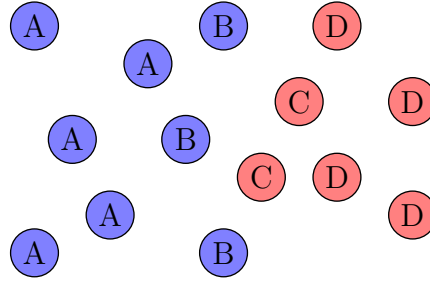
h	1
c	1
V	1



Classe	A	A	A	A	A	B	B	B	C	C	D	D	D	D
Cluster	1	1	1	1	1	2	2	2	3	3	4	4	4	4

Clustering completo ma non omogeneo

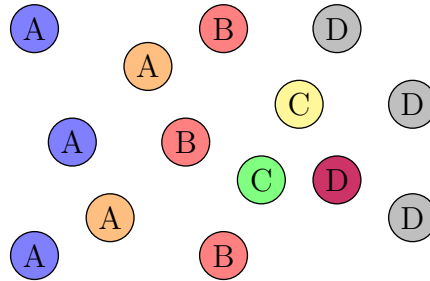
h	0.512
c	1
V	0.677



Classe	A	A	A	A	A	B	B	B	C	C	D	D	D	D
Cluster	1	1	1	1	1	1	1	1	2	2	2	2	2	2

Clustering omogeneo ma non completo

h	1
c	0.727
V	0.842



Classe	A	A	A	A	A	B	B	B	C	C	D	D	D	D
Cluster	1	1	1	2	2	3	3	3	4	5	6	7	7	7

L'algoritmo utilizzato per calcolare la *V-measure* è `homogeneity_completeness_v_measure`, incluso nella libreria *scikit-learn* [5].

4.2 Analisi dei risultati

Per valutare la qualità delle partizioni generate, è stata calcolata la *V-measure* considerando il Settore Scientifico Disciplinare estratto dal sito di dipartimento come classe corretta di appartenenza degli autori.

4.2.1 Dati generati

Nella tabella 4.1 sono indicati i valori di *V-measure* per tutte le possibili combinazioni di metodologia di estrazione paper, unione dei nodi e creazione dei cluster.

Tabella 4.1: Numero di cluster e valore di V-measure delle partizioni generate

Set paper	Unione	Algoritmo di clustering	Num. di cluster	V-measure
Tutti	Nomi	Blockmodel	4	0.51
		Clauset-Newman-Moore	15	0.63
		Girvan-Newman	18	0.57
		Blockmodel GC	6	0.54
		CNM Giant Component	9	0.62
		GN Giant Component	12	0.56
	Distanza	Blockmodel	10	0.36
		Clauset-Newman-Moore	64	0.44
		Girvan-Newman	64	0.48
		Blockmodel GC	6	0.44
		CNM Giant Component	8	0.40
		GN Giant Component	15	0.52
	Edge	Blockmodel	5	0.35
		Clauset-Newman-Moore	56	0.45
		Girvan-Newman	56	0.48
		Blockmodel GC	6	0.47
		CNM Giant Component	8	0.42
		GN Giant Component	15	0.50
Padovani	Nomi	Blockmodel	3	0.40
		Clauset-Newman-Moore	13	0.58
		Girvan-Newman	16	0.62
		Blockmodel GC	3	0.41
		CNM Giant Component	8	0.52
		GN Giant Component	11	0.61
	Distanza	Blockmodel	6	0.48
		Clauset-Newman-Moore	20	0.52
		Girvan-Newman	25	0.64
		Blockmodel GC	4	0.52
		CNM Giant Component	9	0.61
		GN Giant Component	12	0.63
	Edge	Blockmodel	5	0.49
		Clauset-Newman-Moore	20	0.52
		Girvan-Newman	25	0.65
		Blockmodel GC	6	0.60
		CNM Giant Component	9	0.63
		GN Giant Component	12	0.65

Gli stessi dati sono riportati graficamente in figura 4.1, utilizzando le seguenti etichette:

Tu: Tutti; Pa: Padovani

No: Nomi; Di: Distanza; Ed: Edge

Bl: Blockmodel; Cl: Clauset-Newman-Moore; Gi: Girvan-Newman

GC indica l'elaborazione effettuata sulla Giant Component

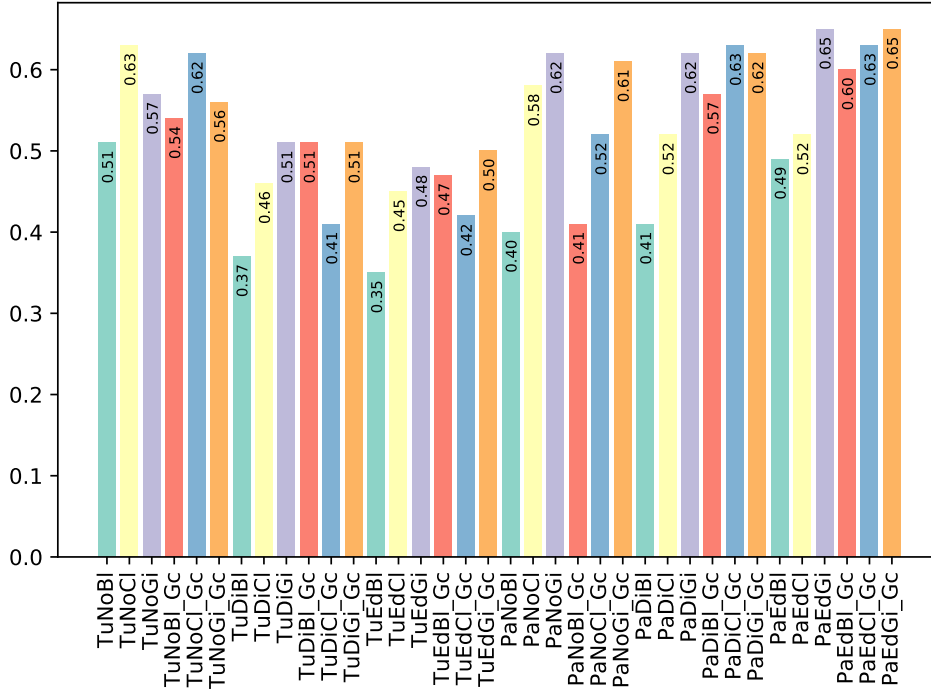


Figura 4.1: Valori di V-measure delle partizioni generate

4.2.2 Analisi

Confronto tra i metodi di estrazione

I valori di *V-measure*, aggregati per metodo di estrazione dei dati ed unione dei nodi, sono riassunti in figura 4.2.

I grafi generati considerando solo i paper scritti da autori con almeno un affiliation padovana hanno un valore medio di *V-measure* di 0.56, lievemente superiore al valore medio ottenuto dai grafi generati considerando tutti i paper estratti, pari a 0.49.

Il valore della *V-measure* del grafo generato considerando tutti i paper estratti ed unendo i nodi per nome è fra i più alti, ma ciò non corrisponde a un grafo ben rappresentativo della rete di autori del DEI, per i motivi illustrati in 2.2.1.

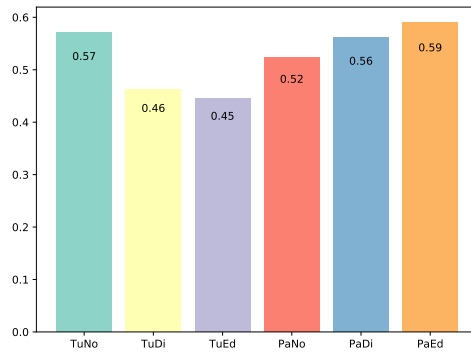


Figura 4.2: Medie dei valori di $V\text{-measure}$ aggregate per metodo di estrazione degli autori e per metodo di unione dei nodi

Confronto tra i metodi di community detection

I valori di $V\text{-measure}$, aggregati per algoritmo di generazione dei cluster, sono riassunti in figura 4.3.

Il metodo Girvan-Newman si è rivelato essere il migliore in termini di $V\text{-measure}$, sia quando applicato all'intero grafo, sia quando applicato solo alla componente centrale dello stesso.

Nel caso del metodo Blockmodel, si nota una differenza tra l'analisi dell'intero grafo e della sua componente centrale, ed è stata rilevata una migliore efficacia del metodo nel secondo caso.

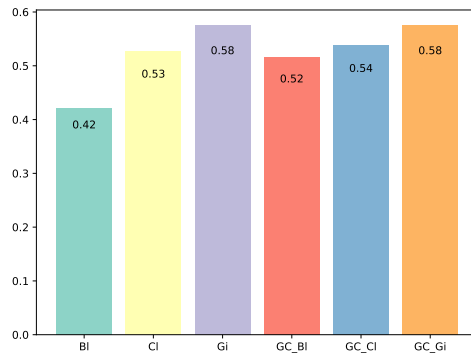
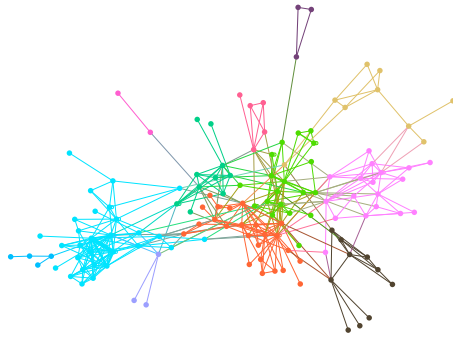


Figura 4.3: Medie dei valori di $V\text{-measure}$ aggregate per algoritmo di generazione cluster

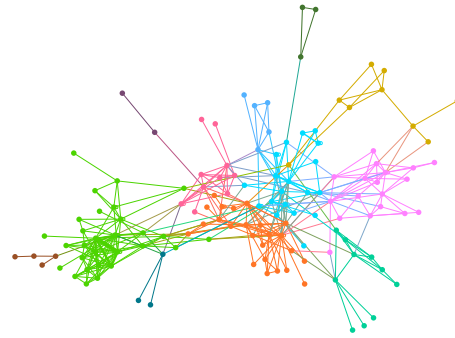
Nelle pagine successive sono riportati i grafi che hanno ottenuto i valori migliori di $V\text{-measure}$

In figura 4.4 sono presenti i grafi relativi ai grafi generati considerando solo gli autori con almeno un'affiliazione padovana, unendo i nodi seguendo la strategia dei nodi adiacenti. In figura 4.5 i cluster degli stessi grafi sono stati aggregati per evidenziare le relazioni tra le comunità del grafo.

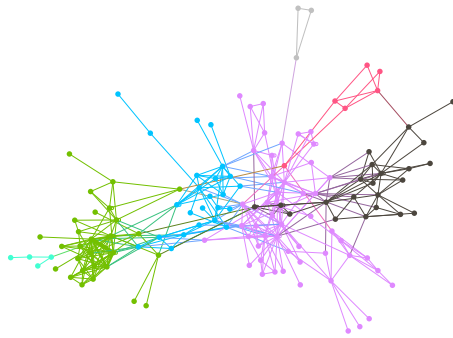
In figura 4.6 sono riportati i cluster ottenuti applicando l'algoritmo di Girvan e Newman alle componenti centrali dei grafi generati in tutti i possibili modi.



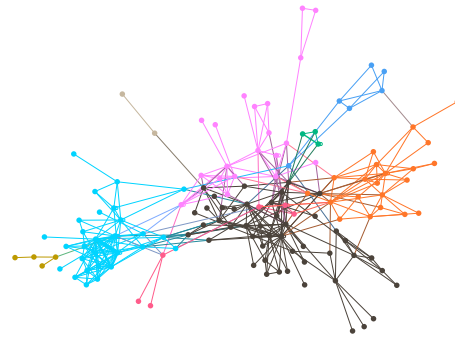
(a) Girvan Newmann



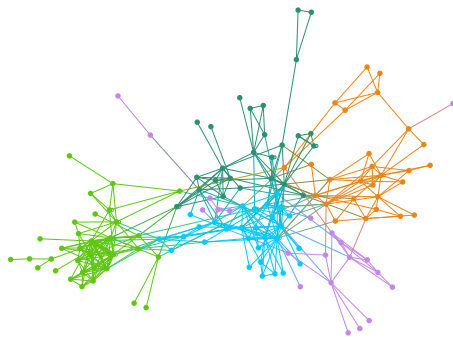
(b) GN - Componente centrale



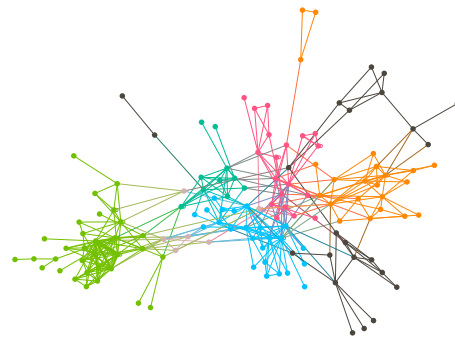
(c) Clauset Newman Moore



(d) CNM - componente centrale

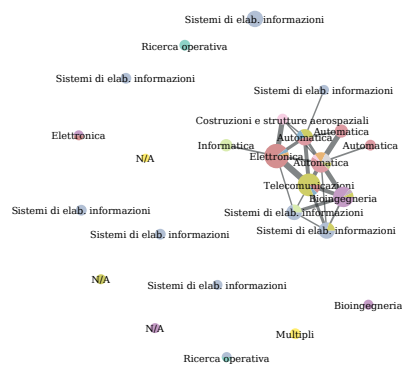


(e) Block Model

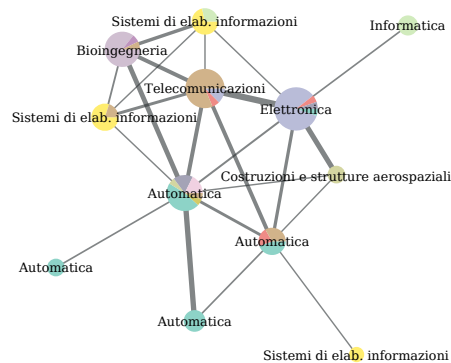


(f) BM - componente centrale

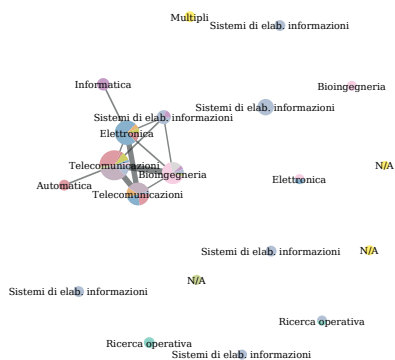
Figura 4.4: Confronto delle suddivisioni in cluster del grafo degli autori con almeno un'affiliazione padovana, dove i nodi sono stati uniti utilizzando la strategia per nodi adiacenti



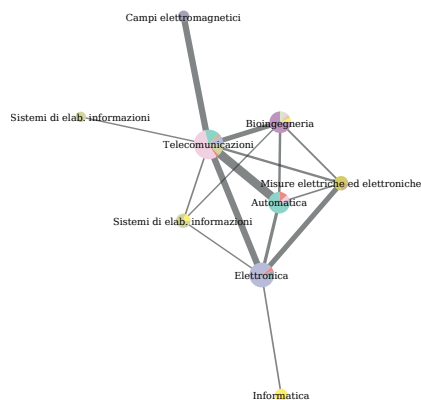
(a) Girvan Newmann



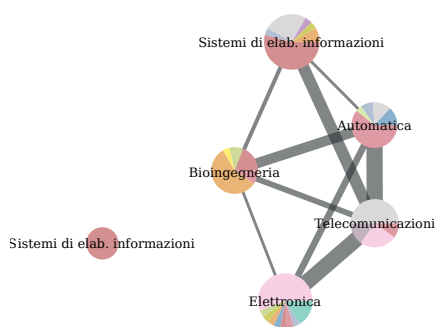
(b) GN - Componente centrale



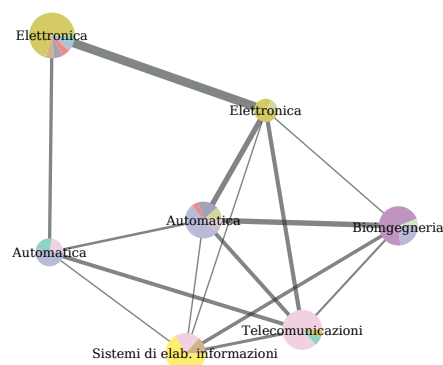
(c) Clauset Newman Moore



(d) CNM - componente centrale

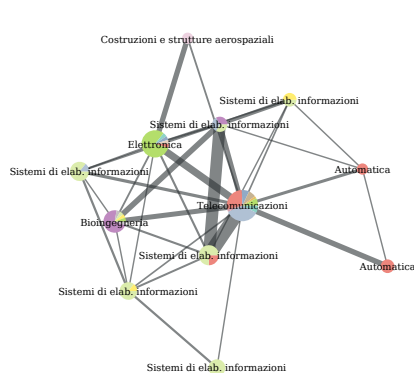


(e) Block Model

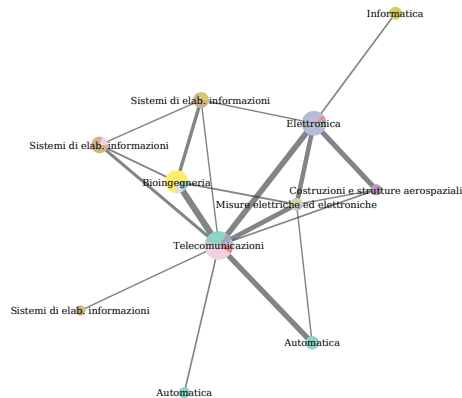


(f) BM - componente centrale

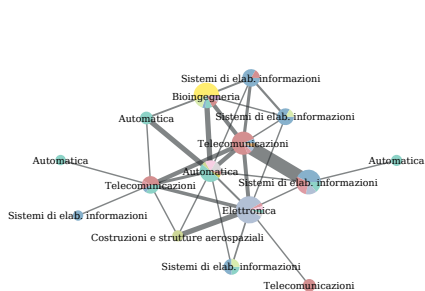
Figura 4.5: Confronto delle suddivisioni in cluster del grafo degli autori con almeno un'affiliazione padovana, dove i nodi sono stati uniti utilizzando la strategia per nodi adiacenti; i cluster sono stati aggregati in nodi, i colori dei nodi indicano la suddivisione in classi dei cluster; l'etichetta corrisponde alla classe predominante nel cluster



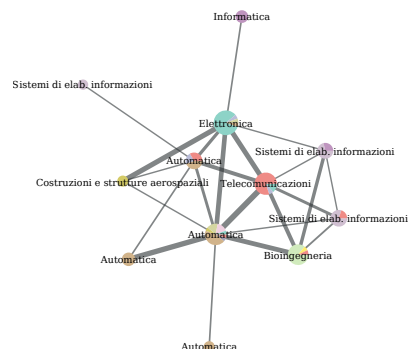
(a) Tutti - uniti per nome



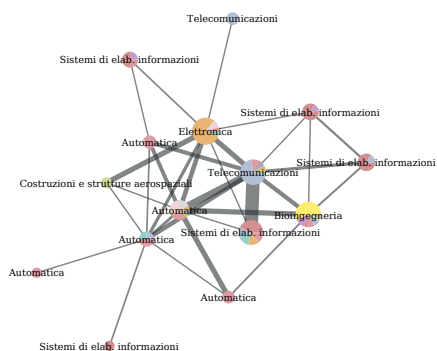
(b) Padovani - uniti per nome



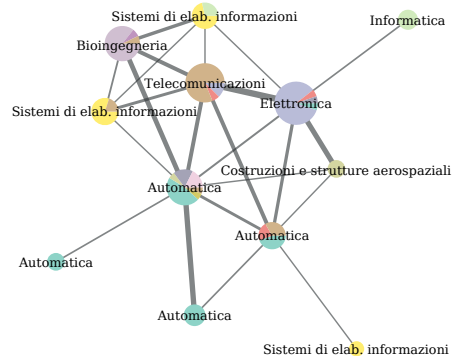
(c) Tutti - uniti per distanza



(d) Padovani - uniti per distanza



(e) Tutti - uniti per nodi adiacenti



(f) Padovani - uniti per nodi adiacenti

Figura 4.6: Confronto delle suddivisioni in cluster generate dall'algoritmo Girvan-Newman applicato alla componente centrale del grafo; i cluster sono stati aggregati in nodi, i colori dei nodi indicano la suddivisione in classi dei cluster; l'etichetta corrisponde alla classe predominante nel cluster

Capitolo 5

Conclusioni

Partendo da una lista di nomi di un dipartimento si può estrarre un grafo che lo rappresenti? Le comunità generate rispecchiano quelle reali?

TODO se vuoi fallo su matematica e mostra un grafo con qualche valore di v-measure

Bibliografia

- [1] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. 70(6):066111, December 2004.
- [2] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002. URL: <http://www.pnas.org/content/99/12/7821>.
- [3] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. <http://snap.stanford.edu/data>, June 2014.
- [4] M. E. Newman and M. Girvan. Finding and evaluating community structure in networks. 69(2):026113, February 2004.
- [5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [6] T. P. Peixoto. Efficient Monte Carlo and greedy heuristic for the inference of stochastic block models. 89(1):012804, January 2014.
- [7] Tiago P. Peixoto. The graph-tool python library. *figshare*, 2014. URL: http://figshare.com/articles/graph_tool/1164194.
- [8] Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. pages 410–420, 01 2007.
- [9] Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June (Paul) Hsu, and Kuansan Wang. An overview of microsoft academic service (mas) and applications. In *Proceedings of the 24th International Conference on World Wide Web, WWW '15 Companion*, pages 243–246, New York, NY, USA, 2015. ACM. URL: <http://doi.acm.org/10.1145/2740908.2742839>.