

POLITECNICO DI MILANO
Corso di Laurea Magistrale in Ingegneria Informatica
Dipartimento di Elettronica, Informazione e Bioingegneria



Un modello generativo spettrale basato su grafi per il mapping semantico di edifici

AIRLab
Laboratorio di Intelligenza Artificiale
e Robotica del Politecnico di Milano

Relatore: Prof. Francesco Amigoni
Correlatore: Ing. Matteo Luperto

Tesi di Laurea di:
Leone D'Emilio, matricola 798842

Anno Accademico 2013-2014

A mia madre

*Di quel poco di vita che mi resta
con piacere darei gli anni migliori
per sapere ciò che ad altri
tu hai detto di me
e questa vita mortale e dell'eterna
ciò che mi spetta
se qualcosa mi spetta
per sapere ciò che da sola
hai pensato di me.*

Gustavo Adolfo Becquer

Sommario

Si consideri un robot mobile autonomo che stia esplorando un edificio scolastico non noto a priori e si assuma che esso abbia già scoperto alcune classi, uffici e bagni. Cosa può inferire il robot circa la presenza e la posizione di altre classi e uffici all'interno dell'edificio scolastico? In questo lavoro di tesi si compie un passo in avanti nel fornire una risposta alla domanda precedente proponendo un sistema basato su un modello generativo che sia capace di rappresentare le strutture topologiche e gli schemi di labeling semantico di edifici e di predire la struttura e lo schema di porzioni inesplorate di questi ambienti. Gli edifici sono rappresentati da grafi non orientati, i cui nodi sono etichettati con i nomi delle stanze e i lati sono collegamenti diretti tra di esse. Data una base di conoscenza iniziale di grafi, l'approccio illustrato, sfruttando l'analisi spettrale, segmenta ogni grafo in sottografi significativi, i quali sono poi raggruppati in cluster in accordo alla loro similarità. Un grafo che rappresenta un nuovo edificio intero o una sua parte è infine generato tramite il campionamento di sottografi dai cluster e il loro collegamento.

Abstract

Consider a mobile robot exploring an initially unknown school building and assume that it has already discovered some classrooms, offices, and bathrooms. What can the robot infer about the presence and the locations of other classrooms and offices in the school building? This thesis makes a step toward providing an answer to the above question by proposing a system based on a generative model that is able to represent the topological structures and the semantic labeling schemas of buildings and to predict the structure and the schema for the unexplored portions of these environments. Buildings are represented as undirected graphs, whose nodes are labeled with names of rooms and edges are physical connections between them. Given an initial knowledge base of graphs, the proposed approach, relying on a spectral analysis of these graphs, segments each graph for finding significant subgraphs, which are then clustered according to their similarity. A graph representing a new building or an unvisited part of a building is eventually generated by sampling subgraphs from clusters and connecting them.

Ringraziamenti

In cinque anni di università succedono una marea di cose: vuoi perché segnano l'ingresso trionfale nel mondo degli adulti con annessi e connessi, vuoi perché si vivono alcune esperienze memorabili. Alla fine di un periodo così importante mi sembra il minimo ringraziare tutte le persone che mi hanno accompagnato e sopportato in questi cinque anni. Cominciamo.

Ringrazio i compagni di corso con cui ho condiviso centinaia di lezioni e pranzi. Un grazie speciale a Gus e ai nostri dialoghi sopra i massimi sistemi, e a Ema, fonte inesauribile di informazioni su videogiochi e serie tv.

Ringrazio il Prof. Amigoni e l'Ing. Luperto che mi hanno seguito in questi mesi, fornendomi preziosi consigli e risolto il mio dubbio di turno. Scrivere una tesi non è affar semplice ma, grazie a loro, ce l'ho fatta.

Ringrazio i miei amici più cari, i Birilli. Questi sette anni hanno confermato il grande valore di questa amicizia: abbiamo condiviso momenti belli e brutti, ma la cosa importante è che siamo ancora qua e spero che ci porteremo dietro quel soprannome ridicolo per un bel pezzo. Ringrazio Orlo e Dade per il supporto datomi in quest'ultimo periodo, avete fatto la differenza.

Ringrazio i miei genitori, per avermi sempre incoraggiato a coltivare i miei interessi e per i sacrifici che hanno fatto per me, non ultimo permettermi di studiare lontano da casa. Li ringrazio per tutto quello che mi hanno trasmesso. Questi ultimi anni ci hanno visto affrontare qualcosa di ben più grave e terribile di un esame andato male, e ringrazio tutta la mia famiglia, tutti gli amici, tutte le persone che mi sono state vicine. Mentre scrivo queste righe è passato un anno esatto dalla scomparsa di mia madre, e io le dedico con tutto il cuore l'impegno di questi ultimi mesi (non dico la tesi che non è troppo poetica a parte forse qualche citazione tra i capitoli); tra tante mancanze da parte mia, almeno questa promessa sono riuscito a mantenerla. Ringrazio tutte le altre persone con cui ho vissuto questi cinque anni, universitari e non: lo spazio è tiranno e non mi concede di nominarle tutte come vorrei, ma questo non mi impedisce di mandare loro un forte abbraccio.

Con affetto, Leone

Indice

Sommario	I
Abstract	III
Ringraziamenti	V
1 Introduzione	1
2 Stato dell'arte	7
2.1 Il mapping semantico	7
2.2 Il concetto di mappa	8
2.2.1 La mappa metrica	9
2.2.2 La mappa di navigazione	10
2.2.3 La mappa topologica	10
2.2.4 La mappa concettuale	11
2.3 La mappa semantica	12
2.4 Usi e applicazioni della mappa semantica	13
2.5 Ottenere una mappa semantica	14
2.5.1 I sensori utilizzati	14
2.5.2 La rappresentazione dell'ambiente	14
2.5.3 La classificazione delle stanze	16
2.6 Verso un utilizzo predittivo dell'informazione semantica	20
2.6.1 Predizione su una base di conoscenza preesistente	23
2.7 Gli approcci al mapping semantico in relazione al lavoro svolto	26
3 La teoria dei grafi	29
3.1 I grafi	29
3.2 La rappresentazione matriciale di un grafo	30
3.2.1 Permutabilità di una matrice	31
3.2.2 La rappresentazione spettrale	31
3.3 Algoritmi noti sui grafi	32

3.4	Un metodo generativo, spettrale e part-based	40
3.5	Un modello generativo sui grafi per predire le parti inesplorate di un edificio	42
4	Un modello generativo di grafi per il mapping semantico	45
4.1	Distinzione tra costruzione del modello generativo e campio- namento	47
4.2	La base di conoscenza del robot	48
4.3	Segmentare i grafi	51
4.3.1	Risolvere il problema spettrale generalizzato	51
4.3.2	Utilizzare il Fiedler value per partizionare il grafo	53
4.3.3	Gestire la ricorsione	55
4.3.4	Memorizzare le connessioni	56
4.4	Raggruppare i sottografi in cluster	58
4.4.1	Identificare i sottografi simili	59
4.4.2	Spectral clustering di sottografi con Ncut	59
4.4.3	Eseguire l'allineamento a livello di cluster	61
4.5	Analizzare le connessioni tra sottografi appartenenti a cluster diversi	61
4.5.1	Estrarre il grado di connettività tra le coppie sottografi	62
4.5.2	Memorizzare le connessioni tra cluster	62
4.6	Generare la nuova configurazione dei cluster	63
4.6.1	Significato della configurazione dei cluster	64
4.6.2	Rappresentazione della configurazione dei cluster e cam- pionamento di Gibbs	65
4.7	Selezionare i sottografi che rappresentano la configurazione dei cluster generata	66
4.8	Connettere insieme i sottografi campionati	67
4.8.1	Determinare i gradi di connettività tra i sottografi	68
4.8.2	Determinare i collegamenti tra i nodi dei sottografi	71
4.9	Ricostruzione del grafo finale	75
5	Il modulo per la generazione di nuovi grafi per il mapping semantico	77
5.1	Le tecnologie utilizzate	78
5.1.1	Ottenere i grafi di input	78
5.1.2	Realizzare il modello generativo in MATLAB	81
5.1.3	Visualizzare i grafi	81
5.2	L'architettura del sistema	81
5.2.1	Il componente Segmentation	84

5.2.2	Il componente Clustering	85
5.2.3	Il componente ConnectionManager	86
5.2.4	Il componente SampleSubgraphs	88
5.2.5	Il componente ReconnectSubgraphs	89
5.2.6	Il modulo generativo GenerateGraphs	90
5.3	Alcune funzioni del modulo	91
5.3.1	Implementare il partizionamento di un grafo	91
5.3.2	Memorizzare le connessioni tra cluster	94
5.3.3	Calcolare i parametri della distribuzione normale multivariata	94
6	Realizzazioni sperimentali e valutazione	97
6.1	I dataset utilizzati	98
6.1.1	Le label delle stanze	100
6.2	Analisi delle fasi principali	103
6.2.1	Analisi della segmentazione dei grafi	104
6.2.2	Analisi del clustering dei sottografi	108
6.2.3	Valutazione del campionamento di Gibbs per le configurazioni dei cluster	111
6.3	La riconnessione dei sottografi	113
6.4	Valutazione complessiva del sistema	116
6.4.1	Le metriche utilizzate	116
6.4.2	Risultati sperimentali	120
6.4.3	Esempio di grafo campionato	124
7	Direzioni future di ricerca e conclusioni	127
7.1	Sviluppi futuri	128
	Bibliografia	135

Elenco delle figure

2.1	Esempio di architettura di un sistema multilivello, tratta da [53]. Le mappe sono disposte dal basso verso l'alto secondo un grado di astrazione crescente.	9
2.2	Esempio di una pianta di un'abitazione con la rispettiva map- pa semantica, tratto da [38]. In basso a sinistra ogni regio- ne spaziale che rappresenta una stanza è evidenziata con un colore. La rappresentazione attraverso un grafo dotato di etichette è mostrata a destra. Le lettere indicano le etichet- te semantiche associate alle rispettive stanze: S per STANZA PICCOLA, M per STANZA MEDIA, B per STANZA GRANDE, C per CORRIDOIO ed E per INGRESSO.	13
2.3	Esempio di ontologia, tratta da [53].	17
2.4	Mappa topologica ottenuta da scansioni laser, tratta da [45].	18
2.5	Architettura di un sistema che utilizza un laser ed una teleca- mera, tratto da [53].	19
2.6	Struttura del modello chain graph di una mappa concettua- le, tratta da [52]. I vertici rappresentano variabili casuali. I lati rappresentano le relazioni tra le variabili casuali. I vertici barrati indicano osservazioni che corrispondono ai rilievi dei sensori.	22
2.7	Mappa semantica di massima verosimiglianza di un ambiente costituito da più edifici nel campus del MIT, tratto da [25]. Le vignette rappresentano input vocali da parte di esseri umani.	23
2.8	Esempio di grafo di un edificio, tratto da [4]. I nodi nel grafo possiedono label che indicano la categoria di quella stanza come cucina ufficio, corridoio.	24
3.1	Esempio di trasformazione di un grafo in un altro attraverso una sequenza S di operazioni di edit, tratto da [73].	33

4.1	Modello della predizione semantica di edifici. Il riquadro evidenzia gli aspetti trattati nel lavoro di tesi.	49
4.2	Esempio su come Ncut privilegia il taglio di A rispetto al taglio di B	55
4.3	Esempi di grafi simmetrici: per ciascuno è indicato il parametro di threshold minimo $\rho_{min}(S_i)$ per cui S_i viene diviso a metà.	57
4.4	Esempio di configurazioni di cluster con tre grafi e quattro cluster, tratto da [73].	65
4.5	Schema della rete Bayesiana per la riconnessione dei nodi di due sottografi. Il significato dei termini è spiegato nel testo.	72
4.6	Due sottografi campionati \hat{s}_1 e \hat{s}_2 . La linea tratteggiata tra i due nodi di label ‘C’ indica una connessione su cui valutare $p(C_{n_C, n_C})$	74
5.1	Schermata del tool per la creazione dei dataset. L’immagine della planimetria rappresenta la scuole secondaria di primo grado del Comune di Berlingo. Il grafo con le label ottenuto manualmente è sovrapposto all’immagine.	79
5.2	Un esempio di file XML contenente le informazioni di un grafo che rappresenta un edificio. In (a) è mostrato il formato per i nodi, mentre in (b) è mostrato il formato per le connessioni.	80
5.3	Diagramma di attività ad alto livello del modulo GenerateGraphs. Ogni colonna corrisponde a uno dei cinque componenti principali.	83
5.4	Esempio di file di testo contenente dati aggregati sulle stanza, tratto da [38]. Ogni riga contiene informazioni su una stanza.	87
5.5	Dipendenze tra i componenti di GenerateGraphs. Gli archi modellano la relazione “l’output di A è richiesto da B”. Si può notare come ConnectionManager e SampleSubgraphs non dipendano tra loro, per cui la loro esecuzione potrebbe anche essere resa parallela.	91
5.6	Diagramma della divisione in sottografi operata da partition. Si può notare come la ricorsione venga gestita in modo iterativo, considerando ad ogni iterazione i grafi parziali temporaneamente ottenuti e memorizzando quali non devono più essere considerati per una suddivisione nelle iterazioni successive.	93

6.1	La classificazione standard delle stanze, utilizzata come riferimento.	100
6.2	La classificazione delle stanze per le SCUOLE.	101
6.3	La classificazione delle stanze per gli uffici.	102
6.4	Esempio di un grafo che rappresenta una scuola (a) e della sua divisione in sottografi (b).	104
6.5	Andamento del numero medio di sottografi in cui è diviso un grafo al variare del parametro di Ncut ρ . In blu sono rappresentate le ABITAZIONI, in verde le SCUOLE e in rosso gli UFFICI.	105
6.6	Andamento della cardinalità media dei nodi dei sottografi al variare del parametro di Ncut ρ . In blu sono rappresentate le ABITAZIONI, in verde le SCUOLE e in rosso gli UFFICI.	106
6.7	Confronto tra il numero medio di nodi e il numero medio di sottografi al variare di ρ . Dall'alto verso il basso si trovano i grafici relativi alle ABITAZIONI, alle SCUOLE e agli UFFICI. . .	107
6.8	Confronto tra il numero medio di sottografi in un cluster (in verde) e il numero cluster (in blu) al variare di ρ_c . Dataset delle SCUOLE.	109
6.9	Esempio di cluster non rappresentativo, ottenuto con $\rho_c = 0.4$. Ad esempio, sono contenuti sia sottografi che rappresentano un insieme di classi (M) collegate a una hall (H), sia sottografi che rappresentano un insieme di segreterie/uffici (Y) collegati a una hall (H).	110
6.10	Esempio di cluster rappresentativo, ottenuto con $\rho_c = 0.8$. Sono rappresentati principalmente gruppi di classi (M) collegate a un corridoio (C).	110
6.11	Esempio di clustering eseguito solo in base alla topologia, ottenuto con $\rho_c = 1.1$. In questo caso la caratteristica strutturale comune a questi sottografi è la struttura <i>a stella</i> . A livello di label vi sono delle differenze significative tra alcuni sottografi.	111
6.12	Un esempio di due grafi ricollegati con $\beta_1 = 0.3, \beta_2 = 0.7$. Si nota che, nonostante gli parametri siano gli stessi, il grafo in (b) sia molto più connesso del grafo (a).	117
6.13	Un esempio di grafo ottenuto tramite un processo di riconnessione incrementale: inizialmente (a) vengono connessi i sottografi principali con $\beta_1 = 0.3, \beta_2 = 0.7$. In seguito (b) i sottografi \hat{s}_i rimasti non connessi sono collegati al sottografo \hat{s}_j per cui il valore di $\hat{c}(\hat{s}_i, \hat{s}_j)$ è massimo (b).	118

6.14	Esempio di un grafo \hat{G} campionato dal dataset SCUOLE. In questo caso la struttura è compatibile con quella di una scuola. I colori dei riquadri identificano un tipo di zona interpretabile in termini di una scuola reale: in nero l'ingresso con l'atrio, in blu le zone per la didattica, in verde la zona amministrativa, in rosso i servizi igienici e in fucsia le zone per conferenze e attività di gruppo.	125
7.1	In (a) è mostrata la mappa che rappresenta il piano terra della Brunswick Training School, a cui è sovrapposto il grafo corrispondente. In (b) è mostrata la parte di mappa nota a un robot in base alle precedenti esplorazioni. I colori dei nodi nel grafo, indicanti le label delle stanze, fanno riferimento allo schema riportato in Figura 6.2.	132
7.2	Un esempio di un passo di predizione. In (a) è mostrato il solo grafo della parte di ambiente nota. In (b) è mostrato un grafo generato a partire dal precedente.	133

Elenco delle tabelle

6.1	Valori medi del numero di sottografi e delle dimensioni dei sottografi per $\rho = 0.5$	107
6.2	Valori delle probabilità che il grado di connessione c_{s_i, s_j} tra due sottografi arbitrari s_i, s_j sia X , avendo effettuato la segmentazione dei grafi con $\rho = 0.5$	115
6.3	Confronto tra il dataset \mathcal{G} e i grafi campionati \hat{G} per la tipologia edilizia SCUOLE.	121
6.4	Confronto tra il dataset \mathcal{G} e i grafi campionati \hat{G} per la tipologia edilizia UFFICI.	122
6.5	Confronto tra il dataset \mathcal{G} e i grafi campionati \hat{G} per la tipologia edilizia ABITAZIONI.	122
6.6	Confronto del numero medio di sottografi $ S $ e del numero medio di nodi per sottografo $ s $ tra il dataset \mathcal{G} e i grafi campionati \hat{G} . Di ogni valore è indicata la media μ e la deviazione standard σ	124

Capitolo 1

Introduzione

L'universo (che altri chiama la Biblioteca) si compone d'un numero indefinito, e forse infinito, di gallerie esagonali, con vasti pozzi di ventilazione nel mezzo, bordati di basse ringhiere. Da qualsiasi esagono si vedono i piani superiori e inferiori, interminabilmente. La distribuzione degli oggetti nelle gallerie è invariabile. Venticinque vasti scaffali, in ragione di cinque per lato, coprono tutti i lati meno uno; la loro altezza, che è quella stessa di ciascun piano, non supera di molto quella d'una biblioteca normale. Il lato libero dà su un angusto corridoio che porta a un'altra galleria, identica alla prima e a tutte. [...] Nel corridoio è uno specchio, che fedelmente duplica le apparenze. Gli uomini sogliono inferire da questo specchio che la Biblioteca non è infinita (se realmente fosse tale, perché questa duplicazione illusoria?) io preferisco sognare che queste superfici argentate figurino e promettano l'infinito...

Jorge Luis Borges, *La Biblioteca di Babele*

Il campo della robotica mobile autonoma è oggetto numerosi studi compiuti negli ultimi anni. L'obiettivo a cui ambisce questa disciplina è la creazione di robot mobili autonomi in grado di aiutare l'uomo nelle attività quotidiane. Uno dei risultati più importanti da raggiungere per ottenere questo scopo è dotare il robot di autonomia, ovvero fornirgli la possibilità di compiere scelte e azioni, relazionandosi con l'ambiente esterno, senza richiedere un continuo intervento umano.

La realizzazione dell'autonomia in un robot richiede innanzitutto una efficace rappresentazione dell'ambiente circostante, attraverso una mappa. Una mappa è infatti utile a un robot per compiere attività elementari come muoversi tra due punti [66], ma non è sufficiente, da sola, per fornire autonomia.

Lo step successivo, oggetto di studio del *mapping semantico*, riguarda la creazione di mappe rappresentanti il significato dell'ambiente, cioè mappe

semantiche [52]. In un ambiente indoor, tipicamente, una mappa semantica assegna delle label a porzioni dell'ambiente. Nel nostro caso, una mappa semantica rappresenta il tipo delle stanze in cui è diviso un ambiente.

L'approccio tradizionale al mapping semantico, in particolare per quanto riguarda la categorizzazione dell'ambiente in stanze e l'identificazione della loro funzionalità, prevede di ricavare le mappe attraverso la classificazione delle informazioni ottenute da sensori quali scanner laser e telecamere. Nel caso di ambienti indoor, questo tipo di approccio presenta alcune problematiche. Un primo limite è relativo all'incertezza legata ad errori delle misurazioni dei sensori, specie se in ambienti stretti e con molti ostacoli; un secondo limite, più strutturale, è relativo al fatto che le mappe semantiche di un ambiente sono ottenute solo dopo che questo è stato interamente visitato, non fornendo al robot alcuna informazione sulle stanze che non ha ancora visitato.

Una situazione particolare, in cui il concetto di autonomia risulta essere importante, riguarda il caso in cui l'ambiente in cui il robot si trova ad operare sia parzialmente o totalmente ignoto. Anche in questo caso, il robot dovrà essere in grado di orientarsi e localizzarsi efficacemente. Una situazione simile non è inusuale per un essere umano, che svolge questo compito in maniera naturale. Una persona, ad esempio, se posta in un edificio scolastico mai incontrato in precedenza, e con uno specifico obiettivo quale raggiungere un bagno o trovare una segreteria, difficilmente visiterà ogni singola stanza incontrata nella sua ricerca, ma si orienterà cercando di giungere direttamente alla propria meta. Questa constatazione apparentemente banale in realtà sottintende un aspetto molto importante, ovvero che alcune caratteristiche degli edifici, conosciute anche se gli ambienti non sono stati visitati precedentemente, permettono un orientamento all'interno di essi.

La maggior parte dei metodi di mapping semantico attuali non considera come un ambiente indoor sia stato costruito dall'uomo per uno specifico uso; ogni edificio possiede infatti una funzione che ne definisce la struttura e ne determina il significato. Il rapporto tra la forma e il significato di un edificio prende il nome di *tipologia edilizia* [40]. Se gli edifici appartenenti a una certa tipologia edilizia presentano delle caratteristiche simili nella loro struttura, queste informazioni possono essere analizzate, codificate in un modello e utilizzate per inferire conoscenza circa altri edifici, non noti a priori, dello stesso tipo. Il modello di una tipologia edilizia vuole essere quindi usato, in questo lavoro di tesi, al fine di fare predizione sulla struttura degli edifici.

Scopo del lavoro di tesi è lo sviluppo di un modello generativo da uti-

lizzare nell'ambito specifico del mapping semantico per la predizione della struttura di edifici non noti a priori.

Un modello generativo determina una distribuzione su un insieme di dati osservati (con cui approssima la distribuzione reale), permettendo, oltre alla modellazione dei dati, il campionamento di nuovi dati con caratteristiche analoghe a quelle dei dati originali. Dal momento che in questo lavoro di tesi una mappa semantica è rappresentata come un grafo non orientato con etichette associate ai nodi, è stato realizzato un modello generativo su grafi basato su queste impostazioni.

I modelli generativi su grafi sono stati studiati meno approfonditamente rispetto ai modelli generativi vettoriali, perché a differenza dei vettori i grafi rendono più difficoltosa la determinazione delle distribuzioni dei campioni. Ciononostante, negli ultimi anni vi sono stati studi sui grafi che hanno portato allo sviluppo di nuovi metodi e algoritmi.

La creazione di modelli generativi su grafi è stata resa possibile soprattutto grazie agli studi nella teoria spettrale dei grafi [14], un ramo dell'algebra che utilizza gli autovalori e gli autovettori delle rappresentazioni matriciali dei grafi per determinarne alcune proprietà e caratteristiche strutturali. Siccome il modello generativo realizzato in questo lavoro di tesi sfrutta alcuni algoritmi e metodi della teoria spettrale dei grafi, può essere considerato un modello generativo spettrale.

Il metodo proposto è basato in parte sul lavoro di [70, 71, 72, 73], con il quale condivide alcuni concetti di base, mentre si distingue negli aspetti legati all'ambito specifico di applicazione. Il modello generativo viene realizzato attraverso una serie di fasi sequenziali, ognuna delle quali si caratterizza per una determinata operazione sui grafi. La costruzione del modello è effettuata su un insieme di grafi di input, che nel caso in esame rappresentano edifici.

La prima fase è rappresentata dalla *segmentazione* dei grafi in sottografi, ovvero dalla suddivisione di ogni grafo in sottografi disgiunti. La segmentazione è effettuata utilizzando il metodo spettrale di Ncut [58]. Le connessioni tra i sottografi vengono memorizzate, e i sottografi ottenuti da ogni grafo sono raccolti in un unico insieme.

La seconda fase consiste nel *clustering* dei sottografi, dove i sottografi simili fra loro sono categorizzati insieme in uno stesso cluster. L'individuazione dei cluster è attuata utilizzando nuovamente il metodo di Ncut sul grafo di affinità. Le connessioni tra i sottografi sono modellate come con-

nessioni tra cluster.

La terza fase svolge il *campionamento* dei sottografi che compongono un nuovo grafo. Per ogni grafo di partenza viene codificato il numero di sottografi appartenenti a ogni cluster, ovvero la configurazione dei cluster in un grafo. Utilizzando un campionamento di Gibbs [22], è generata la configurazione dei cluster di un nuovo grafo, rispetto alla quale sono poi campionati i sottografi dai cluster corrispondenti.

L'ultima fase riguarda la riconnessione dei sottografi sin qui ottenuti in un unico grafo finale. Questa fase consta a sua volta di due passaggi: in un primo momento le connessioni tra i cluster sono approssimate a una distribuzione normale multivariata, dalla quale sono campionate le connessioni tra i cluster nei sottografi selezionati dal campionamento di Gibbs. In un secondo momento, sono stabiliti i nodi attraverso i quali sono collegati i sottografi. Viene utilizzata una rete bayesiana, modellata su alcune proprietà dei nodi in funzione delle label e della topologia dei sottografi. La riconnessione dei sottografi porta alla formazione di un nuovo grafo, che condivide la distribuzione dei grafi campione.

Il modello generativo è stato implementato in un modulo per la generazione di grafi. Il codice è stato scritto nel linguaggio MATLAB.

Il sistema è stato testato su tre dataset contenenti i grafi di tre tipologie edilizie distinte: le SCUOLE, gli UFFICI e le ABITAZIONI.

Alcuni parametri sono stati adattati empiricamente su ogni dataset in modo da contenere entro un certo intervallo il numero e le dimensioni di sottografi e cluster. Sono stati quindi campionati diversi grafi delle tre tipologie.

Il confronto tra alcune metriche misurate nei grafi dei dataset e in quelli campionati ha mostrato che i grafi ottenuti dal modello generativo presentano effettivamente delle caratteristiche comuni a quelli originali. I dataset delle SCUOLE e degli UFFICI hanno mostrato, in particolare, degli ottimi risultati. Da un punto di vista qualitativo, un grafo campionato è interpretabile come rappresentante un edificio reale della stessa tipologia da cui è stato generato.

Il lavoro di tesi svolto si presta essere ulteriormente sviluppato sotto diversi aspetti: un primo tipo di estensione riguarda i dataset, sia aumentando le loro dimensioni, sia attraverso lo studio del modello generativo per altre

tipologie edilizie; un altro tipo di sviluppo fondamentale riguarda l'utilizzo del modello generativo con lo scopo per il quale è stato realizzato, ovvero la predizione della struttura semantica di parti di edifici ignote a priori.

Se il lavoro svolto può essere già utilizzato per affiancare a un dataset reale un insieme di dati ottenuti dal modello, eventualmente impiegati per fare il confronto con un ambiente che un robot sta visitando, un aspetto che dovrà sicuramente essere indagato a fondo riguarda lo sviluppo di un meccanismo che regoli il campionamento in tempo reale a seconda dei risultati dell'esplorazione del robot, in modo da ottenere una convergenza in un tempo ragionevole tra la struttura reale di un edificio ignoto e un grafo campionato dal modello generativo.

La tesi è strutturata come segue.

Nel Capitolo 2 è descritto lo stato dell'arte relativo al mapping semantico nell'ambito della robotica mobile autonoma in ambienti indoor. Vengono mostrati i tipi di approcci principali, ponendo particolare rilievo sulle limitazioni rispetto alla predizione di edifici ignoti. Sono illustrati in maniera più approfondita alcuni metodi basati non esclusivamente sull'uso dei sensori, sviluppati recentemente.

Nel Capitolo 3 vengono introdotti i grafi. Vengono mostrate alcune proprietà caratterizzanti dei grafi, insieme a metodi e tecniche utilizzati per analizzarli, ponendo particolare rilievo alla teoria spettrale dei grafi e ai modelli generativi basati su grafi.

Nel Capitolo 4 viene descritto in maniera dettagliata un modello generativo basato su grafi, realizzato per essere applicato nell'ambito del mapping semantico. L'utilizzo di un modello generativo si pone l'obiettivo di effettuare il campionamento da una distribuzione di grafi, che rappresentano mappe semantiche, in modo che i nuovi dati campionati possano essere impiegati per effettuare predizione semantica sulla topologia e il tipo di stanze di una parte inesplorata di edificio appartenente a una certa tipologia edilizia. Ogni aspetto del modello viene discusso nelle sue possibili varianti e nelle eventuali difficoltà incontrate. Viene inoltre effettuato un confronto con il metodo descritto in [73].

Nel Capitolo 5 viene descritto il modulo che permette, a partire da un insieme di grafi che rappresentano planimetrie di piani di interi edifici appartenenti a una tipologia, la generazione di nuovi grafi con cui ampliare la base di conoscenza di mappe semantiche per un robot mobile autonomo.

Nel Capitolo 6 sono illustrate le prove sperimentali effettuate sull'implementazione del modello generativo. Sono inizialmente descritti i dataset su cui sono stati effettuati gli esperimenti e l'impostazione generale delle prove spe-

rimentali. Successivamente sono mostrati i test effettuati su aspetti specifici del sistema implementato. Infine il sistema è valutato nel suo complesso. La valutazione dei risultati viene fatta sia da un punto di vista qualitativo (mostrando esempi visuali di grafi), sia da un punto di vista quantitativo (utilizzando alcune metriche sui grafi).

Nel Capitolo 7 è riassunta la struttura della tesi e sono tratte alcune conclusioni riguardanti il lavoro svolto. A conclusione del capitolo sono elencati alcuni spunti per sviluppi futuri riguardanti in particolar modo la predizione di porzioni non esplorate di un edificio per la robotica mobile autonoma.

Capitolo 2

Stato dell'arte

Non c'è dubbio che ogni nostra conoscenza incomincia con l'esperienza; da che infatti la nostra facoltà conoscitiva sarebbe altrimenti stimolata al suo esercizio, se ciò non avvenisse per mezzo degli oggetti che colpiscono i nostri sensi e, per un verso, danno origine da sé a rappresentazioni, per un altro, muovono l'attività del nostro intelletto a paragonare queste rappresentazioni, a riunirle o separarle, e ad elaborare per tal modo la materia greggia delle impressioni sensibili per giungere a quella conoscenza degli oggetti, che chiamasi esperienza? Nel tempo, dunque, nessuna conoscenza in noi precede all'esperienza, e ogni conoscenza comincia con questa.

Immanuel Kant, *Critica della ragion pura*

In questo capitolo è descritto lo stato dell'arte relativo al mapping semantico nell'ambito della robotica mobile autonoma in ambienti indoor. Gli approcci al mapping semantico esistenti al momento si differenziano per tipo di rappresentazione di una mappa semantica e modalità attraverso le quali la stessa è ottenuta. Utilizzando questi due criteri di classificazione, illustriamo i maggiori approcci al mapping semantico: la trattazione non intende essere esaustiva, ma evidenzia per ciascun metodo i caratteri principali, le tecniche impiegate e i risultati raggiunti. Alcuni sono confrontati con il nostro lavoro.

2.1 Il mapping semantico

Per un robot mobile autonomo, la possibilità di comprendere lo spazio in cui si opera è un punto di fondamentale importanza. La capacità di un robot di svolgere adeguatamente il proprio compito è infatti determinata dall'efficacia nell'interazione in ambienti spaziali in cui opera. Rendere il modo in cui i robot ragionano sull'ambiente simile a quello degli esseri umani può essere una soluzione per aumentare il grado di autonomia. Nel particolare caso

di ambienti indoor, considerato in questa tesi, ovvero degli edifici, questi ambienti sono costruiti da uomini per essere abitati da uomini [53].

La possibilità di effettuare un parallelismo tra sistemi artificiali (robot) e sistemi biologici (esseri umani) è particolarmente interessante e ha ispirato negli ultimi anni campi come il *bio-inspired computing* [8]. Nel contesto di questo lavoro di tesi può essere utile indagare come gli esseri umani si orientano e interagiscono con ambiente strutturati. Lo studio delle funzioni cognitive umane per rappresentare lo spazio esula dallo scopo di questa tesi, pertanto accenniamo solo ad alcuni risultati in psicologia cognitiva.

Alcune ricerche esplorano la natura inerentemente qualitativa della conoscenza spaziale umana. In accordo con studi sperimentali, viene generalmente assunto che gli umani adottino una rappresentazione parzialmente gerarchica dell'organizzazione spaziale [41, 64]. Le unità elementari di una rappresentazione spaziale qualitativa sono regioni topologiche [15], che corrispondono ad aree spaziali più o meno delimitate. I confini possono essere definiti fisicamente, percettivamente, o possono essere puramente soggettivi. E' stato mostrato che perfino in ambienti naturali privi di chiare barriere fisiche o percettive, gli umani decompongono lo spazio in gerarchie topologiche utilizzando punti di riferimento salienti [26].

Nel caso della robotica, i sensori rappresentano l'elemento analogo ai sensi dell'uomo, in quanto permettono di percepire lo spazio. I dati che un robot ottiene dai sensori devono però essere interpretati [66].

2.2 Il concetto di mappa

Uno dei problemi della robotica mobile autonoma riguarda dunque il modo in cui il robot interpreta e rappresenta le informazioni dei sensori.

Tipicamente, un robot costruisce mappe dell'ambiente utilizzando le informazioni dei sensori per interpretare e rappresentare il mondo. Lo spazio è modellato a diversi livelli di astrazione che variano da mappe metriche di basso livello per la localizzazione e la navigazione del robot, a un livello concettuale che fornisce una decomposizione e categorizzazione semantica dello spazio simile a quella umana. In questi casi, come in [53], si parla di *sistemi multilivello*. In Figura 2.1 è illustrato lo schema di un'architettura di un sistema multilivello.

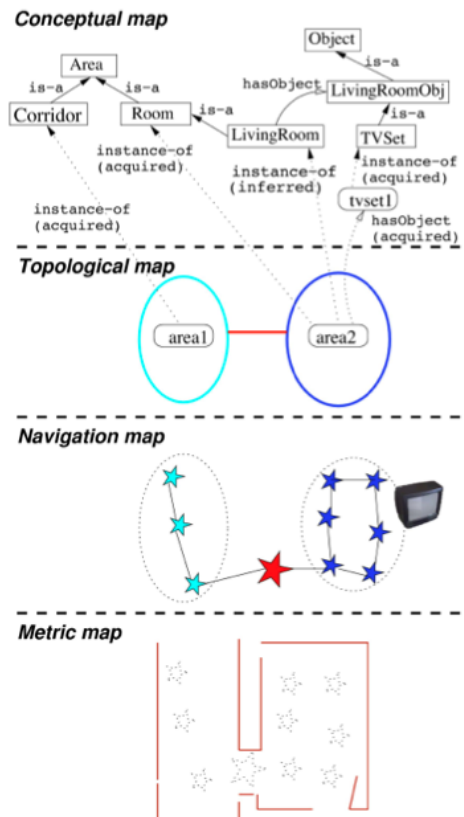


Figura 2.1: Esempio di architettura di un sistema multilivello, tratta da [53]. Le mappe sono disposte dal basso verso l'alto secondo un grado di astrazione crescente.

Descriviamo ora la struttura generale di un sistema multilivello, indicando per ciascuna mappa il tipo di informazione contenuta e le funzionalità che tali informazioni abilitano per il robot.

2.2.1 La mappa metrica

Il livello inferiore del modello spaziale è rappresentato da una mappa metrica, la quale codifica l'occupazione e la non occupazione dell'ambiente.

Una mappa metrica può essere rappresentata in maniera diversa. Una rappresentazione comune è la mappa a griglia, in cui per ogni singola cella si rileva se essa è occupata o meno da un oggetto. Esempi di mappe a griglia si possono trovare in [18, 44, 45, 53, 65].

Una rappresentazione alternativa viene ottenuta rilevando tramite sensori delle feature primitive dello spazio, come punti o segmenti. L'insieme di

queste feature viene poi ancorata a un sistema di riferimento assoluto. La mappa metrica viene aggiornata in maniera incrementale con l'esplorazione dell'ambiente esterno. Un esempio di mapping semantico che utilizza mappe a segmenti si può trovare in [2]. Alcuni riferimenti sulla creazione di segmenti dai punti di una scansione laser si possono trovare invece in [27, 34, 47, 67]. Il metodo tipico utilizzato per questo processo è un raggruppamento dei punti di una scansione in cluster, dove i punti appartenenti ad un cluster vengono poi approssimati con dei segmenti.

Le mappe metriche contengono informazioni spaziali e metriche sulla struttura dello spazio: possono essere utilizzate per la localizzazione del robot o per pianificare una traiettoria, evitando gli ostacoli. Tuttavia, non abbiamo ancora alcun tipo di associazione tra una porzione di spazio e ciò che essa rappresenta: l'unica informazione memorizzata è se in una determinata zona il terreno sia libero o meno, a prescindere da chi o cosa eventualmente la occupi.

2.2.2 La mappa di navigazione

La mappa di navigazione fornisce il livello successivo di rappresentazione, e stabilisce un modello dello spazio libero e della sua connettività, della sua raggiungibilità. E' basata sulla nozione di una roadmap di marcatori dello spazio libero come descritto in [35, 48] e può essere implementata come un grafo i cui nodi sono ancorati alla mappa metrica. Mentre il robot naviga attraverso l'ambiente, un marcatore (o nodo di navigazione) è creato ogni volta in cui il robot ha percorso una certa distanza dal nodo esistente più vicino. I nodi sono connessi da lati seguendo l'ordine in cui sono stati visitati.

2.2.3 La mappa topologica

La mappa di navigazione fornisce la base per ulteriori astrazioni. Una mappa topologica di un ambiente indoor è costruita segmentando il grafo di navigazione in insiemi interconnessi di nodi, rappresentanti delle aree, separati da porte riconosciute (nodi porta). Questo livello di astrazione corrisponde alla segmentazione qualitativa dell'uomo di uno spazio interno in regioni distinte (ad esempio stanze). Una composizione di questo tipo è insita dell'ambiente (un edificio è concepito in termini di stanze e del modo in cui sono disposte), è nota alle persone che lo vivono, ed è così possibile esplicitarla come informazione fruibile anche da un robot. L'informazione semantica estratta e accumulata su intere regioni viene valutata per determinare categorie se-

mantiche appropriate per le aree nella mappa topologica.

È quindi possibile assegnare a ogni singola porzione dell'ambiente una *etichetta semantica* che la rappresenti. In un ambiente indoor, 'stanza', 'corridoio', 'cucina', 'salone' sono alcune delle etichette semantiche che possono rappresentare una caratteristica, la funzione, di una porzione dello spazio. La mappa topologica è dunque rappresentabile come un grafo formato da nodi connessi tra loro, dove ogni nodo rappresenta una stanza e le connessioni tra due nodi indicano che le due stanze da essi rappresentate sono collegate direttamente (ad esempio tramite una porta).

2.2.4 La mappa concettuale

Infine, è possibile un ulteriore livello di astrazione: una mappa concettuale. La mappa concettuale costruisce un'ulteriore interpretazione dell'organizzazione spaziale. Le aree topologiche e le loro categorie formano le entità spaziali base, mentre un reasoner logico è usato per inferire informazioni semantiche a grana più fine per le aree. Il reasoner integra conoscenza su aree e oggetti osservati con una ontologia common-sense di un ambiente indoor. Questa ontologia rappresenta una tassonomia di aree e oggetti e delle relazioni tra essi. Siccome vi è una forte connessione tra oggetti tipici trovati in un'area e la categoria semantica dell'area, questo livello può essere utilizzato anche per vincolare le aspettative su quali oggetti si possono tendenzialmente osservare, posto che il concetto base di un'area sia noto.

Gli ambienti indoor sono realizzati in modo da rendere la loro struttura, l'aspetto generale e il layout spaziale, adatti a svolgere specifiche azioni; i corridoi e le rampe di scale ne sono un esempio. Altre stanze permettono azioni più complesse fornite dagli oggetti che vi sono collocati. Ad esempio, il concetto di salotto si applica a stanze che sono adatte a riposarsi. Il riposo, a sua volta, può essere permesso da certi oggetti, quali divani o televisori. E' quindi possibile combinare informazioni fornite dalla forma geometrica, dall'aspetto generale e dagli oggetti percepiti per fornire una descrizione affidabile dello spazio. La mappa concettuale contiene in definitiva informazioni riferite, oltre che all'ambiente, agli oggetti trovati al suo interno. La relazione tra gli oggetti e l'ambiente in cui si trovano, caratteristica di questo livello di astrazione, permette un ragionamento di tipo simbolico.

Un sistema multilivello come quello appena descritto cerca rendere un robot capace di rapportarsi con l'ambiente che lo circonda attraverso l'astra-

zione e la concettualizzazione progressiva dell'ambiente da un livello a quello superiore [53]. In generale, l'approccio multilivello può essere considerato come una scelta diffusa nel mapping semantico. Un aspetto che vogliamo però sottolineare è che, come ogni tipo di mappa può essere ottenuta e rappresentata in molteplici modi, anche la scelta su quanti e quali livelli di mappe utilizzare sarà una conseguenza delle esigenze applicative specifiche per ogni robot.

2.3 La mappa semantica

Introduciamo ora il concetto di *mappa semantica*, largamente utilizzato in questa tesi.

L'associazione di etichette semantiche, le *label*, a delle porzioni dell'ambiente che rappresentano permette di costituire una mappa semantica, da affiancare alla mappa metrica raffigurante la struttura dell'ambiente stesso. La mappa semantica rappresenta quindi un'associazione tra parti di ambiente e label. Il mapping semantico è il procedimento attraverso il quale si sovrappone una mappa topologica etichettata alla mappa metrica di base, assegnando a ogni nodo, indicante una porzione dell'ambiente, una label indicante la funzione e il significato di tale nodo. Un tipo di mappa come questo, che contiene informazioni di tipo semantico su un edificio, può essere considerato una mappa semantica. Con questi dati a disposizione ci si aspetta di migliorare le performance del robot in alcuni compiti specifici come la localizzazione e l'esplorazione. In Figura 2.2 è riportato un esempio di mappa semantica ottenuta da una pianta di un'abitazione.

È importante notare come la qualità di una mappa semantica sia dipendente dalla qualità della mappa metrica del robot. Se un robot è posto in un ambiente sconosciuto, la mappa metrica non è inizialmente nota: si rendono quindi necessari l'uso un metodo attraverso il quale ottenere la mappa metrica, integrando in modo incrementale le scansioni acquisite in punti diversi dell'ambiente. Questi metodi prendono il nome di Simultaneous Localization and Mapping (SLAM), che rappresenta l'insieme delle attività compiute dal robot per costruire una mappa metrica dell'ambiente e al contempo stabilire la propria posizione all'interno di essa [19, 66].

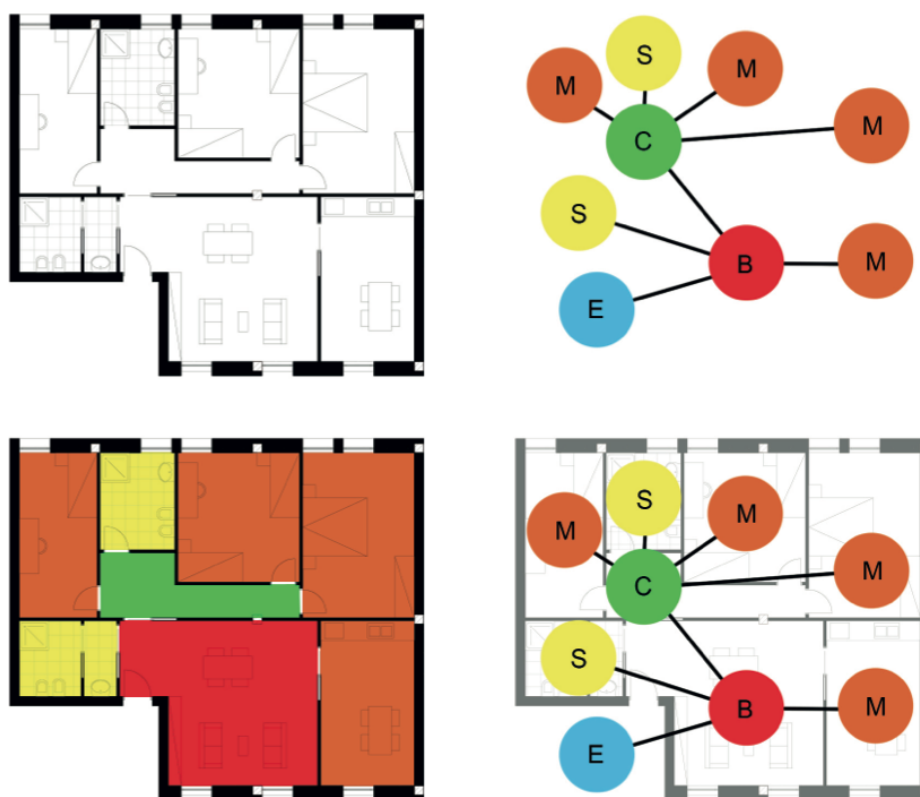


Figura 2.2: Esempio di una pianta di un'abitazione con la rispettiva mappa semantica, tratto da [38]. In basso a sinistra ogni regione spaziale che rappresenta una stanza è evidenziata con un colore. La rappresentazione attraverso un grafo dotato di etichette è mostrata a destra. Le lettere indicano le etichette semantiche associate alle rispettive stanze: S per STANZA PICCOLA, M per STANZA MEDIA, B per STANZA GRANDE, C per CORRIDOIO ed E per INGRESSO.

2.4 Usi e applicazioni della mappa semantica

Prima di scendere nel dettaglio mostrando le diverse scelte tecnologiche e implementative per la costruzione di una mappa semantica, è opportuno elencare alcune delle sue applicazioni pratiche e dei suoi utilizzi più comuni.

- **Path planning:** una mappa semantica può essere utilizzata per ottenere dei percorsi migliori da un dato luogo dell'ambiente a un altro e ridurre il rischio di collisioni [13, 33, 43]. Ad esempio, per raggiungere una determinata stanza è opportuno che un robot cerchi un percorso diretto, evitando di esplorare ricorsivamente tutte le porte che incontra.

- **Esplorazione:** la conoscenza della struttura dell'ambiente e della tipologia di una stanza può permettere ad un robot di decidere quali stanze siano più promettenti da esplorare. Se si ha anche una forma di coordinamento tra i robot, è possibile migliorare notevolmente le performance di esplorazione di una ambiente ignoto, come illustrato in [2, 59, 63]. Ad esempio, dovendo esplorare un ambiente indoor in un tempo limitato, è preferibile percorrere prima i corridoi piuttosto che esplorare tutte le stanze singolarmente.
- **Search and rescue:** essa è un'applicazione in cui una squadra di robot deve esplorare un ambiente ignoto, in condizioni ostili, al fine di ricercare la presenza di eventuali vittime. Vengono utilizzati i vantaggi dovuti sia ad un migliore path planning, sia a una migliore esplorazione [63].
- **Utilizzo domestico:** poiché la casa è un ambiente molto strutturato, darne una conoscenza esplicita al robot può migliorare le sue performance [20]. Il robot può riconoscere la funzione delle singole stanze, sapere quali oggetti trovare in ogni ambiente, con il risultato di una migliore interazione con esso, ad esempio per velocizzare la ricerca di oggetti.

2.5 Ottenere una mappa semantica

Il processo classico che porta da un ambiente indoor, strutturato e non noto a priori, a una mappa semantica dell'ambiente consta di tre passaggi fondamentali che ne influenzano il risultato finale.

2.5.1 I sensori utilizzati

Il primo punto di fondamentale importanza riguarda la scelta del tipo di sensori da utilizzare. Principalmente, in letteratura, ne vengono utilizzati due tipi: scanner laser e videocamera. In alcuni casi i due sensori sono presenti contemporaneamente, come in [54], mentre in molti altri vi è la presenza solo di uno o dell'altro. La scelta dei sensori utilizzati determina il modo in cui viene rappresentata la mappa metrica.

2.5.2 La rappresentazione dell'ambiente

Una volta scelto il tipo di sensore da utilizzare, la decisione seguente riguarda il tipo di rappresentazione che si vuole dare dell'ambiente. Infatti, il robot

riconosce che una data porzione dell'ambiente rappresenta una stanza, e ne attribuisce la funzione, attraverso la ricerca di alcune feature dell'ambiente stesso. Il tipo di feature dell'ambiente, il come vengono ricercate e la loro rappresentazione determinano il modello di stanza che ha il robot. I metodi tipicamente utilizzati per rappresentare il concetto di stanza sono due.

La rappresentazione dell'ambiente tramite landmark

Il primo di questi, l'approccio a landmark, prevede che il robot derivi la sua conoscenza sull'ambiente esterno, utilizzata per creare la mappa semantica, dal riconoscimento di alcuni landmark. Per svolgere questa operazione viene usato come sensore, principalmente, la telecamera. Un landmark è una porzione di ambiente riconoscibile in maniera univoca. Nell'ambito trattato, i landmark possono essere delle feature caratteristiche dell'ambiente esterno o un particolare oggetto che viene riconosciuto. I landmark possono essere ricercati tra un set già definito a priori nel sistema (ovvero il sistema riconoscerà o meno la presenza di una serie di feature che conosce), oppure possono essere derivati dall'ambiente attraverso un metodo probabilistico, come in [30, 31]: il robot così stabilisce autonomamente quali sono le caratteristiche interessanti dell'ambiente e come queste possano essere interpretate. Un esempio di utilizzo di landmark rappresentanti delle caratteristiche dell'ambiente, a cui è stato dato il nome di invarianti, si può trovare in [30]. Un esempio di landmark intesi come un particolare oggetto riconosciuto nell'ambiente è riportato invece in [77].

La rappresentazione dell'ambiente tramite scansioni laser

Nel secondo approccio, invece, il robot riconosce la presenza di una stanza dal raggruppamento dei vari punti, o segmenti, che la rappresentano nella mappa metrica [42]. In questo caso, si parte dalla mappa metrica di basso livello, ottenuta da un sensore laser o dai dati provenienti dalla telecamera, e si cercano di raggruppare i dati in alcune strutture più complesse; queste strutture, coerenti con forme geometriche, caratteristiche spaziali e simmetrie rispetto ad un modello, sono create modo tale che ogni struttura rappresenti una stanza.

Un metodo utilizzato per questo processo di raggruppamento, prevede l'uso di diagrammi di Voronoi generalizzati [13, 33]. Un altro metodo prevede invece l'utilizzo di un modello markoviano nascosto (HMM) per filtrare i risultati delle sequenze di scansioni [56, 74]. Questo tipo di approccio è direttamente legato con il processo di classificazione delle stanze. In alcuni

lavori la divisione dell'ambiente in stanze avviene già a livello delle singole scansioni ottenute dal laser, prima ancora che queste vengano aggiunte alla mappa.

2.5.3 La classificazione delle stanze

Una volta che è stato definito il tipo di rappresentazione dell'ambiente, il processo di classificazione assegna ad ogni singola stanza una label semantica. Questa fase è strettamente dipendente dalle scelte fatte ai livelli precedenti, a seconda dell'utilizzo di un approccio a landmark o a scansioni laser.

Uso di una telecamera e di un'ontologia

Nel primo caso, la fase di classificazione di una stanza avverrà in base ai landmark che sono stati identificati nell'ambiente. Il metodo di classificazione usato prevalentemente con questa configurazione sfrutta delle ontologie e un meccanismo di reasoning automatizzato.

Quest'approccio si fonda sul fatto che si può catalogare un dato ambiente con un'etichetta semantica in base agli oggetti che si trovano al suo interno. Il robot visiterà la stanza e analogamente a quanto farebbe un essere umano "si guarderà in giro con la sua telecamera". Il riconoscimento degli oggetti avviene tipicamente mediante un training eseguito in precedenza, ovvero il robot non è capace di individuare altri oggetti se non quelli per cui è stato addestrato in fase di preparazione. Solitamente, la divisione dell'ambiente in singole stanze non è trattata. L'unica stanza cui deve far riferimento il robot è (implicitamente) quella in cui si trova; di conseguenza non viene dato un concetto di stanza e di divisione dello spazio esplicita come invece accade in altri tipi di lavori. La classificazione della stanza stessa avviene mediante un'ontologia. Un esempio di ontologia è riportato in Figura 2.3.

L'ontologia rappresenta la struttura tipica di ogni ambiente considerato, partendo dalle sue caratteristiche e arrivando a tutti i tipi di oggetti che si possono trovare al suo interno. Il numero e il tipo di oggetti trattati nell'ontologia sono tipicamente decisi a priori e forniti come conoscenza iniziale al robot. In alcuni casi viene data al robot la possibilità di inferire nuova conoscenza a partire dai dati a disposizione. Questo approccio si può trovare in [16, 20, 45, 51, 62, 77].

La telecamera, oltre al riconoscimento degli oggetti all'interno di una stanza, può anche permettere di classificare le stanze in base all'aspetto assunto dall'ambiente, [52].

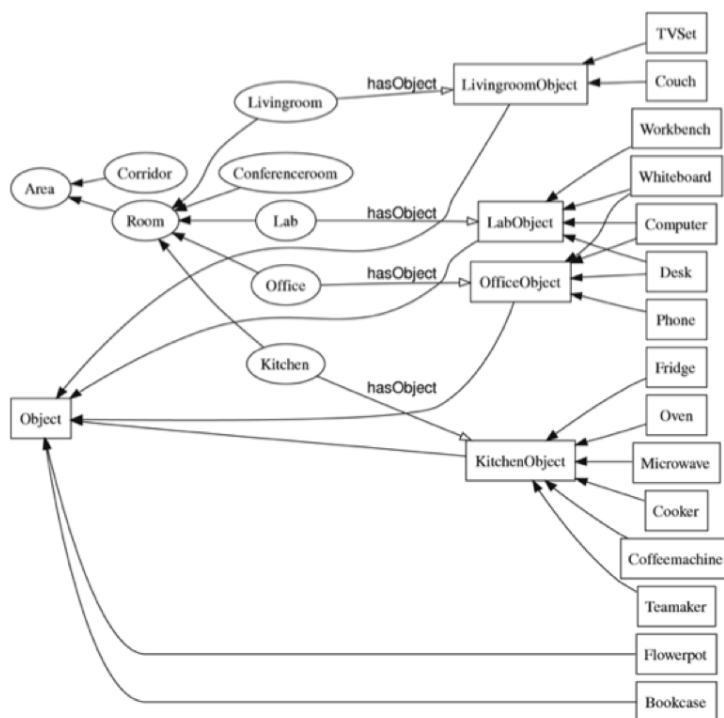


Figura 2.3: Esempio di ontologia, tratta da [53].

L'uso di un'ontologia e di una telecamera presenta alcuni vantaggi e alcuni limiti. Le performance di classificazione risultano essere molto buone per quanto riguarda un ambiente dove possono essere riconosciuti molti oggetti. Al contrario, è meno accurato il riconoscimento di stanze con meno oggetti al loro interno, come un corridoio [54]. Un limite è nel fatto che sia l'ontologia sia gli oggetti riconoscibili sono decisi a priori e non sono ottenuti in maniera automatica. Particolarmente limitante è il riconoscimento di oggetti: ottenere un tool per riconoscere ogni singolo oggetto, che abbia buone performance, è un procedimento complicato e lungo. L'estendere il sistema con nuovi oggetti risulta quindi macchinoso; questo ne limita l'utilizzabilità, siccome il riconoscimento degli oggetti è la fonte primaria di informazioni, dal quale dipendono le prestazioni del sistema. Come evidenziato in [51] l'utilizzo della telecamera è dipendente dalle caratteristiche ambientali, complicandone l'uso in diversi momenti della giornata, a diversi mesi di distanza o in ambienti non favorevoli come possono essere quelli di un'operazione di search and rescue.

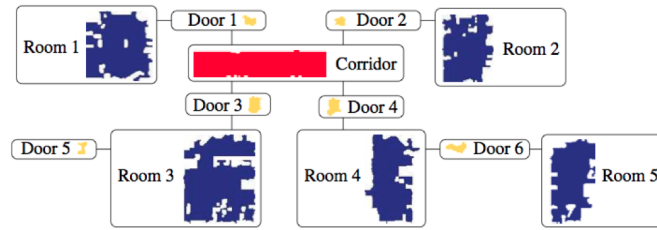


Figura 2.4: Mappa topologica ottenuta da scansioni laser, tratta da [45].

Uso delle scansioni laser

Nel secondo caso, in cui le singole stanze vengono riconosciute direttamente nella mappa ottenuta dallo SLAM attraverso le loro caratteristiche geometriche, il metodo più noto è descritto in [45]. La caratteristica dominante di un classificatore basato sulle scansioni ottenute da uno scanner laser è quella di anticipare la fase di classificazione stessa effettuandola contestualmente alla rappresentazione dell'ambiente. Ciò è dovuto al fatto che il meccanismo utilizzato prevede di etichettare le singole scansioni del laser ogni volta che sono ottenute per poi riportare i risultati sulla mappa metrica. Una volta che la mappa è stata classificata punto per punto in base alle scansioni questa viene divisa in porzioni omogenee, utilizzando metodi diversi. In particolare la divisione tra stanze può essere ottenuta attraverso un modello markoviano nascosto [45], un approccio bayesiano unito allo SLAM [7, 20], oppure attraverso un clustering sullo spettro delle singole scansioni unito a una classificazione usando AdaBoost [10]. In Figura 2.4 è mostrata una mappa topologia ottenuta da scansioni laser.

Uso combinato di un telemetro laser e di una telecamera

E' inoltre possibile combinare i due approcci descritti sopra, utilizzando in parallelo un sistema di classificazione mediante telecamera e ontologia e uno mediante classificazione delle scansioni del telemetro laser. Esempi sono riportati in [21, 46, 54]. L'architettura di un sistema che utilizza entrambi i sensori è riportata in Figura 2.5.

Il risultato comune a questi esempi dimostra come l'uso in contemporanea di una telecamera e di un sensore laser fornisca performance migliori rispetto all'uso di un sensore singolarmente. La telecamera è, difatti, imprecisa in condizioni di luce variabile e restituisce informazioni che si modificano nel tempo, ma riesce a fornire dei dati molto affidabili sulla presenza di oggetti. Il laser invece fornisce dei dati che talvolta non sono sufficienti per

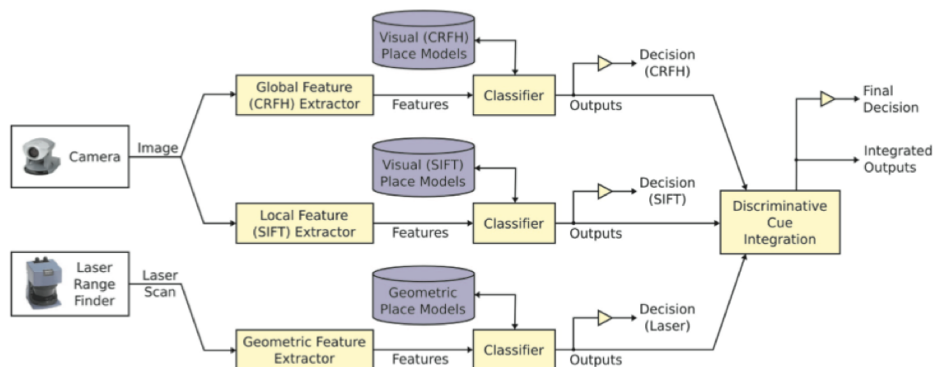


Figura 2.5: Archiettura di un sistema che utilizza un laser ed una telecamera, tratto da [53].

classificare in maniera completa un ambiente ma che hanno il pregio di essere maggiormente affidabili e costanti nel tempo. I limiti dei dati ottenuti dal laser possono essere colmati da un'integrazione di quelli ottenuti dalla telecamera e viceversa, ed è appunto questa caratteristica che viene sfruttata. In particolare, in [54] è mostrato come i risultati migliori si ottengano in un sistema di classificazione multi-livello. Ogni singola stanza riceve tre label semantiche: due da classificatori utilizzando feature derivate dalla telecamera e una facente uso del laser. I tre output in questione vengono ulteriormente classificati in un nuovo livello ottenendo come risultato l'etichetta finale.

Recentemente sono apparsi lavori che utilizzano tre tipi di sensori, integrando a telecamera e scanner laser il Microsoft Kinect. In [5] si mostra che l'uso dei tre sensori permette di ottenere la rappresentazione tridimensionale dell'ambiente con maggiore facilità.

Classificazione basata sulle funzioni di energia

Un approccio alternativo e differente per l'attribuzione delle label, basato sulla massimizzazione di funzioni di energia. L'autore, in [61], propone un sistema per la segmentazione automatica di mappe metriche bidimensionali in unità semantiche, valutando funzioni spaziali basate su caratteristiche quali la connettività (il numero di percorsi tra stanze che incrociano una certa area) o proprietà funzionali (ad esempio la stanza sembra un posto di lavoro). Queste funzioni spaziali sono rappresentate come funzioni di energia, che vengono valutate sui punti della mappa metrica. Una label

semantica è poi assegnata ad ogni punto in accordo alle funzioni di energia. Un passo di raffinamento è effettuato valutando il pattern spaziale delle label semantiche, al fine di assegnare delle label più precise.

2.6 Verso un utilizzo predittivo dell'informazione semantica

Come abbiamo visto nella sezione precedente, vi sono molteplici approcci al mapping semantico che si differenziano per diverse scelte nell'utilizzo dei sensori, nel tipo di rappresentazione dell'ambiente e nel modo in cui classificano le stanze. Tuttavia, questi approcci diventano concettualmente molto simili tra loro se si considera che tutti quanti partono dal presupposto che il fine del semantic mapping sia la stessa classificazione dell'ambiente. Si assume cioè implicitamente che non sia possibile sfruttare le informazioni semantiche per inferire nuove informazioni che non siano ottenibili dai dati acquisiti [20].

Una conseguenza diretta è che, se tali approcci si sono dimostrati efficaci nel classificare stanze di ambienti indoor, incrementando la consapevolezza del robot nei confronti di porzioni dell'ambiente già conosciute, tali approcci non forniscono alcuna conoscenza sulle parti ignote (non visitate) dell'ambiente. Ciò è dovuto al fatto che tutti questi metodi assegnano label sulla base delle letture dei sensori acquisite nell'ambiente corrente e su modelli affinati sui dati dei sensori acquisiti nelle precedenti esplorazioni del robot. L'utilizzo di un modulo di mapping semantico come quelli visti finora può aumentare le performance di un robot in esplorazione, ma le informazioni del mapping semantico sono riferite allo spazio già esplorato. La parte di spazio che maggiormente interessa al robot è, però, quella non ancora esplorata.

In astratto, una mappa semantica è una rappresentazione dell'ambiente che potrebbe essere (almeno parzialmente) disaccoppiata dai dati dei sensori. Una simile situazione può permettere di ragionare anche sulle parti inesplorate dell'ambiente. L'idea che si trova anche alla base del nostro approccio è che sia l'informazione contenuta all'interno delle mappe semantiche a determinare alcune delle caratteristiche dell'ambiente e non viceversa.

Un esempio può chiarire questo concetto. Si ricordi la classificazione di una stanza mediante una telecamera e un'ontologia: in questo caso le etichette semantiche sono assegnate in base agli oggetti che le stanze contengono, in accordo con l'ontologia. Questa considerazione implica che la

funzione e l'uso della stanza sono imposti dagli oggetti che vi sono al suo interno. Un'assunzione di questo tipo può essere vera per un robot, che è un osservatore esterno in un ambiente a lui inintelligibile, ma non è vera per le persone. Le etichette semantiche delle stanze per una persona identificano la funzione posseduta dalle stesse stanze, mentre gli oggetti che vi si trovano sono una diretta conseguenza della funzione di una stanza. Il rapporto causale è quindi ribaltato rispetto a quello rappresentato dall'ontologia: è la categoria di una stanza che caratterizza gli oggetti che vi si trovano all'interno e non gli oggetti al suo interno che determinano l'etichetta corrispondente. Una stanza di un certo tipo, che deve contenere determinati oggetti al suo interno, viene concepita come tale fin dalla sua costruzione. Una cucina, ad esempio, deve poter ospitare un forno e sarà pertanto collocata in un ambiente appositamente predisposto.

Recentemente sono stati proposti miglioramenti che mirano a espandere il classico framework di mapping semantico, provando a sfruttare la natura astratta dell'informazione semantica. In particolare sono apparsi i primi metodi che potrebbero predire e fare inferenza sulla conoscenza semantica relativa alla struttura delle parti inesplorate dell'ambiente. In seguito vengono descritti tre approcci che cercano di predire lo spazio inesplorato.

Framework probabilistico basato su feature eterogenee

Il sistema presentato in [52] è un framework probabilistico di mapping semantico che applica delle label appropriate a sei tipi di feature dell'ambiente (oggetti, porte, forme delle stanze, dimensione delle stanze, aspetto e stanze associate). La mappa semantica è rappresentata come un *chain graph*, mostrato nella Figura 2.6. Un chain graph è una generalizzazione di una rete bayesiana e di Markov Random Fields, e permettono di modellare relazioni sia causali (archi orientati) sia associative (archi non orientati). La struttura è aggiornata runtime in base ai dati percepiti. L'uso di un chain graph permette di considerare le incertezze dei modelli dei sensori e di stimare l'ambiente spaziale sconosciuto.

Il beneficio primario di avere un modello relazionale probabilistico è quindi la capacità di inferire alcuni concetti basandosi esclusivamente sulle loro relazioni con altri concetti anziché sulle osservazioni dirette. Il chain graph può per esempio prevedere l'esistenza di una feature di una certa categoria (come una stanza e la sua label o gli oggetti contenuti in una stanza) nello spazio inesplorato e può estendere la mappa concettuale di conseguenza.

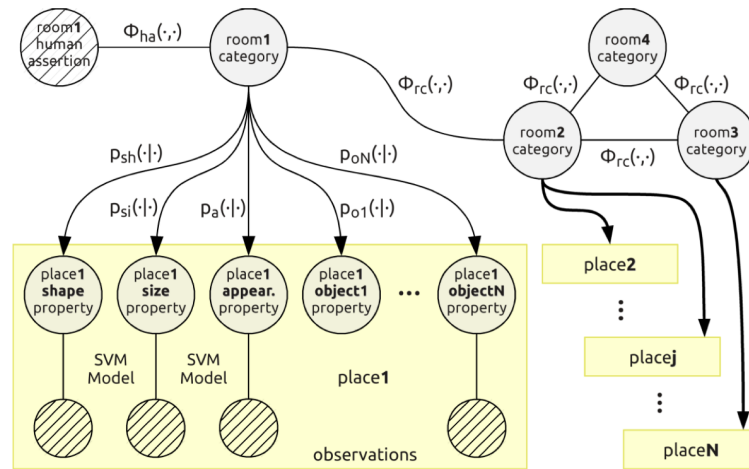


Figura 2.6: Struttura del modello chain graph di una mappa concettuale, tratta da [52]. I vertici rappresentano variabili casuali. I lati rappresentano le relazioni tra le variabili casuali. I vertici barrati indicano osservazioni che corrispondono ai rilievi dei sensori.

Predizione basata su descrizioni in linguaggio naturale

L'approccio riportato in [25] integra le rappresentazioni metriche, topologiche e semantiche con informazioni derivanti da descrittori in linguaggio naturale ottenuti da utenti umani. In questo metodo il grafo semantico viene definito come una tupla contenente le rappresentazioni metriche, topologiche e semantiche dell'ambiente. Su di esso viene calcolata una distribuzione su un insieme di proprietà semantiche associate all'ambiente quali l'aspetto (in base ai rilievi di scanner laser e telecamere) e le label delle stanze. La distribuzione sul livello semantico è mantenuta utilizzando un factor graph. Per ogni stanza vengono mantenute due variabili, una per la categoria di stanza e un'altra contenente le label associate a quella stanza. Ad esempio, una sala conferenza potrebbe avere label multiple come SALA INCONTRI o SALA CONFERENZA.

La mappa semantica viene quindi integrata dalle classificazioni basate su immagini e scansioni laser, basandosi sulla distribuzione di alcune variabili osservate per ogni categoria di stanza. Un secondo passo di integrazione è dato dalle descrizioni in linguaggio naturale da parte di utenti umani. Ognuna di queste descrizioni aggiorna la probabilità di osservare certe label in certe stanze. La label di una stanza è infatti collegata alla categoria della stanza e per ogni categoria è più probabile che vengano generate certe label anziché altre. Ad esempio, se una persona potrebbe descrivere una CAFFET-

TERIA come una TAVOLA CALDA è improbabile che la descriverà come un UFFICIO. Una volta rilevato che una label corrisponde sempre alla stanza a cui è stata assegnata, vengono considerate esatte anche le altre label che vengono rilevate per le stesse stanze. Ciò serve a individuare i sinonimi, evitando così le ambiguità insite nel linguaggio naturale. In Figura 2.7 è mostrato come le descrizioni fornite da persone portano alla formazione di una mappa semantica di massima verosimiglianza. Il risultato dell'approccio descritto è un metodo che, ad esempio, può inferire dalla frase "l'uscita è di fianco alla caffetteria in fondo all'atrio" l'esistenza e la posizione di un'uscita e di una caffetteria.

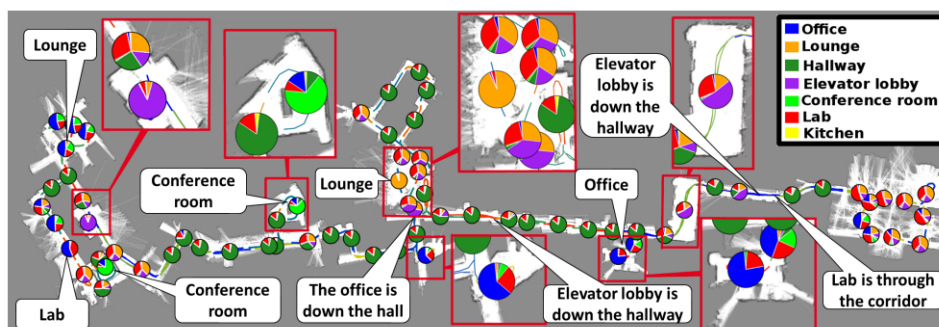


Figura 2.7: Mappa semantica di massima verosimiglianza di un ambiente costituito da più edifici nel campus del MIT, tratto da [25]. Le vignette rappresentano input vocali da parte di esseri umani.

2.6.1 Predizione su una base di conoscenza preesistente

In [4] è introdotto un approccio piuttosto differente, che si concentra interamente sulla predizione della struttura di un ambiente. Gli autori considerano una base di conoscenza di 38,000 stanze (che rappresentano i campus universitari del MIT e del KTH). Ogni piano degli edifici è rappresentato come un grafo, come quello riportato in Figura 2.8, dove i nodi sono stanze le cui label vengono attribuite secondo la loro funzione (classi, dormitori, uffici, ...).

Il problema che gli autori si pongono è il seguente: dato il grafo incompleto di un piano, come predire la categoria della prossima stanza e il modo in cui essa è connessa al grafo corrente o il nuovo percorso più probabile tra le due stanze. La presenza di una base di conoscenza così ampia di grafi ha permesso di analizzare il dataset, esaminando sia le proprietà globali dei piani degli edifici, sia la struttura locale. Elenchiamo ora queste proprietà



Figura 2.8: Esempio di grafo di un edificio, tratto da [4]. I nodi nel grafo possiedono label che indicano la categoria di quella stanza come cucina ufficio, corridoio.

dal momento che risultano utili anche al nostro approccio.

In [69] viene introdotta la proprietà *small-world* per descrivere il tipo di grafi in cui la gran parte dei nodi può essere raggiunta da ogni altro nodo in un piccolo numero di step. E' stato mostrato che grafi naturali come i social network e Internet sono small-world. Se anche gli edifici appartenessero a questa categoria, i metodi di reasoning sullo spazio potrebbero essere migliorati sfruttando queste caratteristiche. Le due proprietà globali dei grafi utilizzate per determinare se i grafi degli edifici siano small-world sono:

- *Lunghezza caratteristica di un percorso:* Viene calcolato il numero medio di nodi, cioè di stanze, che un robot deve visitare per poter spostarsi da una stanza a un'altra. In [4] i risultati dimostrano che la lunghezza caratteristica di un percorso per grafi con un numero di nodi compreso tra 10 e 25 è pari a 1. Ciò significa che una persona o un robot deve attraversare in media un'altra stanza per poter raggiungere qualunque altra stanza nello stesso piano rispetto a dove si trova. In genere questa stanza intermedia è un corridoio. La lunghezza

caratteristica di un percorso cresce poi linearmente con le dimensioni del grafo.

- *Coefficiente di clustering*: indica se all'interno di un grafo sono presenti dei sottogruppi di nodi densamente connessi. Il coefficiente di clustering rappresenta la frazione del numero di triplette chiuse sul numero complessivo di triplette di nodi connessi in un grafo. Nel caso di [4] è stato trovato un coefficiente simile a quello della rete elettrica degli USA, suggerendo una presenza di sottogruppi connessi all'interno dei grafi.

Dal momento che è stata rilevata una lunghezza caratteristica di un percorso bassa e un coefficiente di connettività relativamente alto, gli autori considerano che gli ambienti indoor possano rientrare insieme ad altri grafi naturali nella categoria small-world.

I grafi degli edifici mostrano inoltre due proprietà globali tipiche di grafi naturali. Esse sono:

- *Grafi scale-free*: la probabilità che un nodo abbia k lati incidenti segue la funzione: $P(k) \propto k^{-\gamma}$. Questa distribuzione è rilevabile in grafi molto complessi quali Internet, le citazioni scientifiche, le reti elettriche e le reti sociali [1, 6]. Ciò significa che in molti grafi naturali la connettività locale è indipendente dalle dimensioni del grafo.
- *Statistiche del singolo nodo*: è possibile analizzare il grafo di un edificio a livello dei singoli nodi se si vogliono estrarre delle feature legate in questo caso al tipo di stanza. Ad esempio è possibile osservare che i nodi dei corridoi presentano un numero di lati incidenti maggiore rispetto alle altre categorie di stanza, deducendo in accordo all'intuizione comune che i corridoi fungono da connettori per le altre stanze.

Secondo gli autori di [4], le proprietà che sono state ritrovate nei grafi degli edifici mostrano che le topologie di questi ambienti indoor siano fortemente strutturate. In particolare, il coefficiente di clustering indica la presenza all'interno del piano di un edificio di sottografi molto connessi. Ciò suggerisce che un piano di un edificio consista in alcune parti funzionali, costituite da un raggruppamento di stanze che formano un sottografo. Tali pattern di stanze più frequenti sono estratti usando gSpan [76], un algoritmo usato per estrarre i sottografi più frequenti. Usando questa informazione, il metodo proposto predice sia la tipologia sia le label di stanze non visitate,

data una mappa parziale. La predizione è fatta identificando i sottografi più comuni che si sovrappongono parzialmente alla mappa parziale già esplorata.

2.7 Gli approcci al mapping semantico in relazione al lavoro svolto

In questo capitolo è stata fatta una panoramica degli approcci al mapping semantico attualmente esistenti. Da un punto di vista concettuale, è possibile riassumere i vari lavori svolti suddividendoli in tre macrocategorie:

- Il primo tipo raggruppa tutti i metodi che effettuano mapping semantico elaborando i dati dei sensori del robot. Si tratta in gran parte di approcci multilivello che costruiscono una mappa metrica con i dati dei sensori e al di sopra di questa individuano le regioni topologiche dello spazio e associano una label semantica ad esse. Dal momento che assumono che non esista informazione semantica che non derivi dalle informazioni dei sensori, questi approcci possono essere utilizzati solo per migliorare le prestazioni in aree precedentemente esplorate del robot.
- Il secondo tipo contiene gli approcci che integrano ai dati dei sensori altre fonti di informazione per cercare di predire le parti di edifici non ancora esplorate. Due esempi visti sono [52], che fa inferenza su un modello relazionale probabilistico, e [25] che utilizza descrizioni di persone in linguaggio naturale. In entrambi i casi vengono costruiti modelli probabilistici (rispettivamente tramite chain graph e factor graph) i quali diventano l'input stesso dei metodi per prevedere le stanze ancora inesplorate: vi è una prima forma di disaccoppiamento dell'informazione semantica dai dati sensoriali.
- Il terzo tipo, rappresentato da [4], non usa dati di sensori come fonte diretta di conoscenza per il labeling semantico. Il metodo opera a un livello più alto di astrazione, ragionando direttamente sulle stanze contenute nella base di conoscenza, ottenute indipendentemente dall'uso di robot.

Passare da room-level a floor-level

Quest'ultimo tipo di approccio introduce un concetto molto importante: per poter fare predizione della struttura semantica di un ambiente è necessario

avere un modello che riesca a descrivere tutte le possibili caratteristiche salienti di un edificio. Questo è possibile identificando proprietà locali e globali degli edifici. Il nostro approccio presenta molti punti di contatto con quello di [4], nel modo in cui sfrutta le caratteristiche strutturali ricorrenti degli ambienti che formano la base di conoscenza.

Il modello generativo per il mapping semantico che viene proposto nei prossimi capitoli utilizza una rappresentazione di edifici come grafi: in particolare, vengono utilizzati metodi che fondano i loro presupposti nella teoria spettrale dei grafi, una parte della matematica che applica metodi di algebra spettrale ai grafi. Gli elementi che sono al centro della nostra indagine sono pertanto i grafi; nel Capitolo 3 vengono discusse le basi teoriche e metodologiche su cui si fonda nostro lavoro.

Capitolo 3

La teoria dei grafi

I dati dimostrano in modo convincente che la maggior parte delle nostre azioni è guidata da leggi, schemi e meccanismi che in quanto a riproducibilità e capacità predittiva uguagliano quelli individuati nelle scienze naturali. [...] Seguendo le tracce di queste scoperte arriveremo a considerare i ritmi della vita come segni di un ordine più profondo che caratterizza il comportamento umano, ordine che può essere esplorato, previsto e senza dubbio sfruttato. [...] Più a fondo le esamineremo, più sarà evidente che le azioni umane seguono schemi semplici e riproducibili, governati da leggi di vasta portata. Dimenticate il lancio dei dadi e le scatole di cioccolatini come metafore della vita. Pensatevi come un robot sognante guidato dal pilota automatico e sarete molto più vicini alla verità.

Albert-László Barabási, *Lampi*

In questo capitolo viene data un'introduzione alla teoria dei grafi. In particolare, vengono mostrate alcune proprietà caratterizzanti, insieme a metodi e tecniche utilizzati nell'analisi dei grafi. Le nozioni qui introdotte hanno lo scopo di esplicitare alcuni concetti di base riguardo ai grafi, su cui sarà basata la trattazione dei capitoli successivi di questa tesi.

3.1 I grafi

La *teoria dei grafi* è quel ramo della matematica, della ricerca operativa e dell'informatica che si occupa di studiare i grafi, strutture matematiche discrete attraverso cui è possibile rappresentare una grande varietà di fenomeni, consentendone uno studio in termini quantitativi.

Un grafo è costituito da un insieme di oggetti semplici, detti *nodi* o *vertici*, e da *lati* o *archi* che connettono i nodi.

Un grafo G è formalmente definito come una coppia $G = (N, E)$ in cui:

- $N = \{n_1, n_2, \dots, n_n\}$ è l'insieme dei nodi;
- $E = \{(n_1, n_2), (n_1, n_3), \dots, (n_{n-1}, n_n)\} \subset N \times N$ è l'insieme dei lati.

I lati possono essere orientati o meno a seconda del tipo di relazione che è necessario modellare:

- un grafo G è orientato se $(n_1, n_2) \in E \Leftrightarrow (n_2, n_1) \in E$,
- $(n_1, n_2) \in E \Leftrightarrow (n_2, n_1) \in E$ se $(n_1, n_2) \in E$ indica un arco orientato diretto da n_1 a n_2 .

Una specifica informazione, ad esempio un peso numerico o una label, può essere associata a nodi o lati di un grafo, che prenderà il nome di *grafo pesato*. Più in generale, un grafo può essere usato per rappresentare un insieme di entità, i nodi, e le relazioni, i lati, che intercorrono tra esse.

3.2 La rappresentazione matriciale di un grafo

Uno dei modi più utilizzati per rappresentare un grafo è attraverso una matrice.

Una *rappresentazione matriciale* di un grafo è una matrice \mathbf{G} di dimensioni $|N| \times |N|$, tale che un elemento G_{ij} di questa matrice rappresenta alcune proprietà del lato (n_i, n_j) se $n_i \neq n_j$ o del nodo n_i se $n_i = n_j$.

Il tipo più intuitivo di rappresentazione matriciale di un grafo è la *matrice di adiacenza*, in cui la diagonale principale contiene il peso dei nodi e gli altri elementi contengono il peso dei lati. Una matrice di adiacenza \mathbf{M} è così definita:

$$M(n_i, n_j) = \begin{cases} w_N(n_i), & \text{se } n_i = n_j \\ w_E(n_i, n_j), & \text{se } n_i \neq n_j \text{ e } (n_i, n_j) \in E \\ 0, & \text{altrimenti} \end{cases} \quad (3.1)$$

laddove $w_N(n_i)$ è il peso del nodo n_i e $w_E(n_i, n_j)$ è il peso del lato (n_i, n_j) . Nel caso di grafi non orientati e non pesati, gli elementi sulla diagonale principale sono posti a 1 e gli elementi corrispondenti ai lati presenti nel grafo sono posti a 1.

Se il grafo non è orientato, la matrice \mathbf{M} è simmetrica.

Un altro tipo di rappresentazione matriciale molto ricorrente è la *matrice Laplaciana*, per costruzione semidefinita positiva. Avendo definito *grado di*

un nodo la somma dei lati incidenti a quel nodo, $d_n = \sum_{i=1}^{|N|} W_{in}$, la matrice Laplaciana è così ottenuta:

$$\mathbf{L} = \mathbf{D} - \mathbf{M} \quad (3.2)$$

laddove \mathbf{D} è una matrice diagonale i cui elementi sono dati dal *grado* dei nodi.

$$\mathbf{D}(n, n') = \begin{cases} d_n, & \text{se } n = n' \\ 0, & \text{altrimenti} \end{cases} \quad (3.3)$$

Una rappresentazione alternativa della matrice Laplaciana \mathbf{L} è la sua forma normalizzata \mathbf{N} . In questa rappresentazione, gli elementi vengono scalati in modo che gli autovalori siano compresi nell'intervallo $[0, 2]$, proprietà utile per alcuni algoritmi e risultati teorici [14]. \mathbf{N} è definita come:

$$\mathbf{N} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} \quad (3.4)$$

3.2.1 Permutabilità di una matrice

Dal momento che l'ordine dei nodi in un grafo può non essere significativo in alcuni contesti, con la permutazione degli indici associati ai nodi si ottiene un grafo isomorfo. Chiamando \mathbf{P} la matrice di permutazione dei nodi, allora il grafo $\mathbf{G}' = \mathbf{P} \mathbf{G} \mathbf{P}^T$ ottenuto permutando \mathbf{G} rappresenta lo stesso grafo \mathbf{G} .

3.2.2 La rappresentazione spettrale

Un risultato importante dell'algebra lineare mostra che una qualunque matrice quadrata, simmetrica e reale può essere decomposta in autovalori e autovettori per produrre una *rappresentazione spettrale* [14].

Dal momento che le rappresentazioni matriciali dei grafi discusse precedentemente rientrano in questa categoria, è sempre possibile ottenere autovalori e autovettori di una matrice di un grafo. In particolare, in un grafo non orientato, una matrice di adiacenza simmetrica garantisce che tutti gli autovalori siano reali. L'ambito della teoria spettrale dei grafi [14] ha portato allo sviluppo di molti approcci e applicazioni. Definita \mathbf{G} la rappresentazione matriciale di un grafo, allora la scomposizione di $\mathbf{G} = \mathbf{V} \mathbf{E} \mathbf{V}^T$ in un insieme di autovalori \mathbf{E} e di autovettori \mathbf{V} è la sua *decomposizione spettrale*.

Formalmente, da una rappresentazione matriciale \mathbf{G} si ottengono una matrice $\mathbf{E} = \text{diag}(e_1, e_2, \dots, e_n)$ e una matrice $\mathbf{V} = (\mathbf{v}_1 | \mathbf{v}_2 | \dots | \mathbf{v}_n)$, laddove a ogni autovalore e_i corrisponde l'autovettore \mathbf{v}_i . Se i valori degli autovalori sono ordinati in modo che $e_1 \leq e_2 \leq \dots \leq e_n$, allora la decomposizione spettrale di ogni permutazione della matrice \mathbf{G} , \mathbf{PGP}^T , risulterà nello stesso insieme di autovalori ordinati e colonne di autovettori ordinate. I componenti individuali delle colonne degli autovettori tuttavia varieranno a causa della permutazione.

L'utilizzo della rappresentazione spettrale di un grafo porta diversi vantaggi. Il primo è che fornisce una forma di rappresentazione più idonea al problema dell'ordinamento dei nodi in un grafo.

Inoltre, l'uso della decomposizione spettrale per rappresentare un grafo porta a una maggiore visibilità di alcune proprietà strutturali. Ad esempio, l'autovettore associato al secondo autovalore più piccolo, detto Fiedler value, può essere usato per la segmentazione del grafo poiché pone in evidenza i gruppi di nodi molto connessi tra loro e poco connessi rispetto agli altri nodi. Inoltre, se la rappresentazione originaria è una matrice Laplaciana, allora il numero di autovalori di modulo zero è collegato al numero di componenti non connesse presenti in un grafo.

3.3 Algoritmi noti sui grafi

La capacità di modellare una enorme varietà di situazioni rende i grafi uno strumento particolarmente potente. Negli ultimi anni, in particolare, vi è stato un grande sviluppo nella cosiddetta *network theory*, quella parte della teoria dei grafi che si occupa di studiare le reti sociali, le reti biologiche, Internet, e in generale tutti i sistemi che presentano un elevato grado di interconnessione. Per questo motivo, lo sviluppo di algoritmi efficienti che operino sui grafi è di fondamentale importanza.

In questa sezione vengono descritti alcuni algoritmi sui grafi che sono stati utilizzati all'interno del lavoro di tesi. Non si pretende pertanto di svolgere in questa sede uno studio esaustivo dello stato dell'arte riguardante i grafi, bensì si cerca di mostrarne alcuni aspetti specifici. Il sistema sviluppato in questa tesi utilizza questi metodi, integrandoli in un modello generativo.

Inizialmente vengono discussi gli algoritmi di similarità e di allineamento dei grafi. Viene poi introdotto il campo della segmentazione dei grafi che considera come un grafo può essere decomposto in un insieme di sottografi

nella maniera più logica. Infine, vengono discussi gli approcci per costruire modelli generativi di grafi.

Misurare la similarità tra grafi

Un problema comune in pattern recognition è la misurazione della similarità fra due grafi. Ad esempio, in un problema di ricerca all'interno di database di dati strutturati, è opportuno riuscire a eseguire delle query in un grande database computando la similarità tra la struttura di una query e le strutture contenute in un database. Questo tipo di informazione può essere usato per computare la similarità tra due grafi. In questa sede ci limiteremo a discutere una misura semplice e intuitiva di similarità fra grafi.

La *distanza di edit* tra grafi [57, 12] è un approccio che cerca di misurare la similarità tra due grafi trasformando un grafo nell'altro. Questa trasformazione può essere espressa da un *edit path* che descrive una sequenza di operazioni che possono essere applicate a un grafo per alterarne la struttura. Tali operazioni includono l'inserimento e la cancellazione di nodi e lati, e la sostituzione di un nodo o di un lato. Ad ogni operazione è assegnato un costo e la somma dei costi delle operazioni all'interno di un edit path determina il costo totale di applicazione di tale edit path a un grafo. Un esempio di trasformazione di un grafo attraverso operazioni di edit è mostrato in Figura 3.1. La distanza di edit tra due grafi è formalmente definita come l'edit path di costo minimo. Il problema di trovare il path di edit di costo minimo è NP-completo.

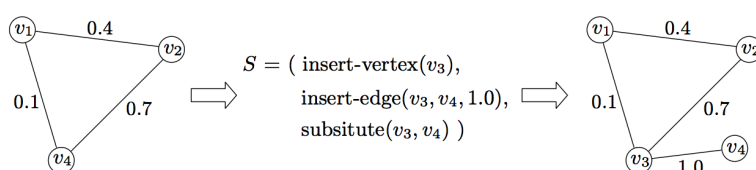


Figura 3.1: Esempio di trasformazione di un grafo in un altro attraverso una sequenza S di operazioni di edit, tratto da [73].

L'idea di usare la distanza di edit come misura di similarità tra i grafi compare frequentemente in letteratura, e sono stati sviluppati alcuni metodi per migliorare la sua usabilità. Ad esempio, in [11] è stata stabilita una relazione tra la distanza di edit e il massimo comune sottografo sotto una certa funzione di costo. Il *massimo comune sottografo*, detto anche $mcs(G_1, G_2)$, è un sottografo sia di G_1 sia di G_2 tale per cui non esiste un altro sottografo comune a entrambi con un numero maggiore di nodi. La funzione di costo è

attribuita come segue: gli inserimenti e le cancellazioni di nodi hanno costo 1, le sostituzioni di nodi o lati identici hanno costo 0 ma sostituzioni differenti hanno costo ∞ e infine l'inserimento o la cancellazione di nodi ha costo 0.

Nel definire questa funzione di costo, gli autori considerano solo grafi completi.¹ Questo semplifica la dimostrazione e non porta a perdite di generalità. Un grafo qualunque può essere mappato su un grafo completo ponendo in quest'ultimo a zero i pesi dei lati che non compaiono nel grafo standard. In questo modo l'inserimento e la cancellazione di nodi codificano l'inserimento e la cancellazione di lati.

La distanza di edit $d(G_1, G_2)$ e il massimo comune sottografo sono legati dalla relazione

$$d(G_1, G_2) = |G_1| + |G_2| - 2|mcs(G_1, G_2)| \quad (3.5)$$

dove le cardinalità sono espresse come numero di nodi.

Un approccio intuitivo come quello appena descritto è efficace se si considerano grafi di dimensione ridotta. In particolare, è possibile allineare due grafi in modo tale da rendere il massimo comune sottografo immediatamente individuabile e il calcolo di una misura di similarità più agevole. Per questo motivo l'allineamento tra grafi costituisce un aspetto importante.

Metodi di allineamento tra grafi

Il problema dell'allineamento tra due grafi consiste nello stabilire le corrispondenze tra i rispettivi nodi. Questo problema è **NP**-completo a causa dell'ordine arbitrario dei nodi, e dalla possibilità di trovare un allineamento molti-a-molti. Tuttavia, molti approcci risolvono il problema di trovare una relazione uno-a-uno tra gli insiemi di nodi N_1 e N_2 di due grafi G_1 e G_2 . L'inclusione del nodo \emptyset negli insiemi di entrambi i grafi ($N_1 \cup \{\emptyset\}$ e $N_2 \cup \{\emptyset\}$) permette ai nodi di un grafo di essere in corrispondenza con nessun nodo o alternativamente (come nel nostro approccio) di essere associati a un nodo fittizio.

I due approcci principali per la risoluzione di questo problema sono i metodi di ricerca e i metodi di ottimizzazione non lineare [73]. I metodi di ricerca costruiscono uno spazio degli stati che rappresenta tutte le possibili

¹Un grafo completo è un grafo in cui ogni nodo è direttamente collegato a tutti gli altri nodi.

relazioni tra i due grafi; tale spazio è poi esplorato per trovare la corrispondenza tra i nodi che minimizza un certo criterio. Tuttavia, siccome lo spazio degli stati cresce esponenzialmente con il numero di nodi dei grafi che devono essere allineati, anche tecniche approssimate di ricerca dell'ottimo quali gli algoritmi genetici sono inadatte a fornire allineamenti per grandi grafi in tempi ragionevoli. Per questo motivo si sono cercate soluzioni alternative nei metodi di ottimizzazione non lineare.

Umeyama [68] utilizza un approccio di tipo spettrale per risolvere il problema dell'allineamento per grafi della stessa dimensione nel caso di archi orientati e in quello di archi non orientati. Il metodo effettua la decomposizione spettrale sulla matrice di adiacenza \mathbf{A} (nel caso non orientato) o su una matrice Hermitiana computata a partire da quella di adiacenza (nel caso orientato). Se due grafi G_1 e G_2 sono isomorfi allora la matrice di permutazione \mathbf{P} che risolve $\mathbf{P}\mathbf{A}_1\mathbf{P}^T = \mathbf{A}_2$ fornisce l'associazione tra i nodi. La risoluzione esatta dell'equazione è un problema combinatorio; in [68] viene quindi proposto un metodo approssimato per trovare una matrice di permutazione \mathbf{P}' che risolve il problema in modo tale che l'errore associato al mapping \mathbf{P}' sia il più vicino a zero possibile.

[68] mostra che la matrice di permutazione \mathbf{P}' che massimizza $tr(\mathbf{P}'^T\bar{\mathbf{V}}_1\bar{\mathbf{V}}_2^T)$ (dove $\bar{\mathbf{V}}_1$ e $\bar{\mathbf{V}}_2$ sono le matrici in valore assoluto degli autovettori dei grafi G_1 e G_2 , rispettivamente) sarà molto vicina alla matrice di permutazione ottima quando G_1 e G_2 sono quasi isomorfi. Trovare \mathbf{P}' che massimizzi la traccia può essere calcolato dal metodo ungherese con complessità temporale $O(n^3)$ [32] (rispetto al numero di nodi n).

La matrice di permutazione prodotta da questo metodo peggiora progressivamente con l'aumentare della distanza tra i due grafi. Applicare ulteriori metodi di ottimizzazione alla matrice di permutazione subottima può migliorare la soluzione, fermo restando che in caso di distanza significativamente alta tra i due grafi è opportuno utilizzare un metodo diverso.

Gold e Rangarajan [23] descrivono un metodo di allineamento che si basa su due idee chiave: iterative projective scaling e graduated non-convexity. La prima è usata per rinforzare i vincoli di assegnamento doppi; in questo caso, gli autori utilizzano l'algoritmo *Softassign* [60] per esprimere il vincolo che un nodo in un grafo può essere allineato al più a un nodo nell'altro grafo, e viceversa. La seconda è una tecnica di continuazione che trasforma una matrice che descrive i mapping discreti in un'altra che contiene valori continui. Questo limita la possibilità di trovare un punto di minimo locale, fornendo un parametro di controllo che può spostare la soluzione da valo-

ri continui (all'inizio dell'esecuzione) a valori quasi discreti (verso la fine dell'esecuzione).

Poiché la tecnica di graduated non-convexity fornisce solo una matrice quasi discreta per valori alti del parametro di controllo, è necessario effettuare computazione addizionale sulla matrice di allineamento non-discreta per trasformarla in una versione discreta. Gli autori usano una semplice euristica che pone a 1 l'elemento massimo in ogni colonna e tutti gli altri a 0. Tuttavia usare questa euristica può risultare in un mapping subottimo dal momento che gli assegnamenti sono determinati in sequenza e non simultaneamente. E' possibile ottenere risultati migliori utilizzando un algoritmo che consideri tutte le corrispondenze tra nodi simultaneamente quale il metodo ungherese [32].

L'algoritmo proposto in [23] ha una complessità temporale di $O(lm)$, dove l e m rappresentano il numero di lati dei due grafi da allineare. Nel caso pessimo, rappresentato da due grafi completi, entrambi di cardinalità n , la complessità equivalente è $O(n^4)$. Se si vuole effettuare un paragone con il metodo di [68], quest'ultimo risulta più rapido, con una complessità di $O(n^3)$. Tuttavia, se l'algoritmo di [68] richiede preliminarmente che i grafi da allineare siano strutturalmente simili, il metodo di [23] non risente di questo problema. In particolare, esso risulta efficace anche su grandi istanze di grafi aleatori.²

Nel nostro caso applicativo, vi è un altro punto a favore dell'algoritmo di [23]. Esso può essere facilmente esteso per modellare ulteriori vincoli sui grafi. Ad esempio, se a ogni nodo è associata una label, affinché l'allineamento sia valido è necessario avere una corrispondenza tra nodi con la stessa etichetta. In altre parole, potrebbero esserci più allineamenti topologicamente equivalenti; in questo modo le label costituiscono un ulteriore vincolo da considerare in fase di allineamento. Data una funzione $f : N \rightarrow \mathcal{L}$ che associa a ogni nodo una label, una coppia di lati (n_1, n_2) e (n'_1, n'_2) può considerarsi allineata se $f(n_1) = f(n'_1)$ e $f(n_2) = f(n'_2)$. Utilizzando il metodo di [68], che opera sulle matrici di adiacenza, dovremmo considerare a posteriori la validità dell'allineamento ottenuto. Questo passo aggiuntivo non si verifica in [23].

²Un grafo aleatorio è un grafo in cui, dati n nodi, questi vengono collegati secondo una certa distribuzione di probabilità.

Segmentare i grafi

Nel caso dell'analisi di un grafo costituito da molti nodi e lati, in alcune applicazioni può essere utile effettuare una divisione in k sottografi. Se i pesi dei lati rappresentano una misura di similarità tra i nodi, questo porterà ad avere una similarità alta tra nodi appartenenti allo stesso sottografo e bassa tra nodi appartenenti a sottografi diversi. Questa divisione avviene effettuando un *taglio* sul grafo di partenza. Se un grafo $G = (N, E)$ è diviso in due sottografi G_1 e G_2 costituiti dagli insiemi di nodi disgiunti N_1 e N_2 , rispettivamente, allora il valore del taglio è la somma di tutti i lati che hanno un nodo in N_1 e l'altro in N_2 .

$$cut(G_1, G_2) = \sum_{n_1 \in N_1, n_2 \in N_2} w(n_1, n_2) \quad (3.6)$$

La bipartizione ottima di un grafo è quella che minimizza questo valore di taglio.

Sono stati proposti vari criteri di taglio, con l'obiettivo di dividere il grafo nella maniera più naturale. Uno di questi è *Minimum Cut* [75]. Questo metodo cerca di minimizzare il taglio massimo attraverso tutti i sottografi. Questo approccio, seppur semplice, tende a tagliare piccoli insiemi di nodi isolati nel grafo, senza tenere conto della struttura globale.

Una soluzione a questo problema è stata fornita in [58], dove gli autori propongono un nuovo criterio che calcoli il costo del taglio come frazione del totale dei lati connessi a tutti i nodi del grafo. Tale criterio è denominato *normalized cut*:

$$Ncut(G_1, G_2) = \frac{cut(G_1, G_2)}{assoc(G_1, G)} + \frac{cut(G_2, G)}{assoc(G_2, G)} \quad (3.7)$$

dove $assoc(G_1, G) = \sum_{n_1 \in N_1, n \in N} w(n_1, n)$ è matematicamente equivalente alla definizione di *cut* e rappresenta la totalità delle connessioni dei nodi in G_1 ; $assoc(G_2, G)$ è definito allo stesso modo.

Usare il valore di associatività come normalizzazione fa sì che il taglio non venga fatto quando il numero di nodi nei due insiemi è molto differente.

L'algoritmo si basa su di una matrice di similarità tra i nodi del grafo, decomposta in autovalori e autovettori. Gli autori mostrano che l'autovettore associato al secondo autovalore più piccolo della matrice Laplaciana, conosciuto anche come *Fiedler value*, può essere usato per partizionare il grafo in accordo alla definizione di *Ncut*. Infine, viene fatta una verifica

per determinare se il grafo è stato segmentato a sufficienza (o se è stato raggiunto un numero fisso di partizioni). In caso contrario il processo si ripete ricorsivamente. Gli autori valutano il loro metodo sia su immagini statiche che su sequenze in movimento, ottenendo in entrambi i casi buone performance.

Verso un modello generativo

Un *modello generativo* è un modello che specifica una distribuzione di probabilità congiunta per i campioni osservati, dati alcuni parametri nascosti.

In letteratura sono presenti numerosi esempi di modelli generativi nel dominio vettoriale, mentre nel dominio di modelli generativi di gradi vi sono ancora pochi contributi. I modelli generativi sono usati nell'ambito dei metodi statistici per il pattern recognition per descrivere la distribuzione di probabilità in uno spazio vettoriale. I pattern individuali sono definiti da vettori, permettendo così la definizione delle feature individuali dei pattern.

L'idea di utilizzare i grafi in modo generativo è invece piuttosto recente e si è sviluppata negli ultimi anni.

In [37] gli autori propongono un modello generativo che utilizza un insieme di grafi come input. Il procedimento principale alla base del metodo è la vettorizzazione della matrice di adiacenza di ogni grafo campione; in questo modo viene costruita una descrizione vettoriale sulla quale si possano utilizzare tecniche statistiche applicabili ai dati vettoriali. Prima di fare questo, tuttavia, i grafi vengono allineati utilizzando un algoritmo descritto dagli stessi autori in [36]. Le differenze nelle dimensioni dei grafi sono gestite estendendo i grafi più piccoli con dei nodi fittizi. Dalla matrice allineata estesa \mathbf{A}_k di ogni grafo campione S_k viene ricavato un vettore monodimensionale per ogni grafo, ottenendo $\mathbf{a}_k = \text{vect}(\mathbf{A}_k)$, dove *vect* è l'operazione di vettorizzazione di una matrice. A questo punto è possibile costruire un modello lineare deformabile³ sui vettori risultanti da utilizzare per esprimere le variazioni strutturali presenti nell'insieme dei grafi campione. Questo viene fatto calcolando la media $\boldsymbol{\mu}$ e la covarianza $\boldsymbol{\Sigma}$ dell'insieme dei vettori (considerando quindi ogni elemento del vettore \mathbf{a}_k una variabile aleatoria) e effettuando la decomposizione spettrale della matrice di covarianza nella matrice degli autovalori ordinati $\boldsymbol{\Lambda}$ e nella matrice degli autovettori ordinati $\boldsymbol{\Phi}$. Quest'ultima contiene le componenti principali della varianza dei dati, mentre l'insieme degli autovalori determina il grado della varianza as-

³Con modello lineare deformabile si intende un modello di oggetti deformabili come i volti umani che sono parametrizzati usando modelli lineari separati per la forma e l'aspetto.

sociata a ogni componente. Tale metodo consente alla descrizione vettoriale di un grafo campione di essere rappresentata come una media sommata a un vettore che determina il modo specifico con cui il grafo varia strutturalmente. Questo vettore è poi proiettato sulle componenti principali della distribuzione, come illustrato dalla seguente operazione:

$$\hat{\mathbf{a}}_k = \boldsymbol{\mu} + \boldsymbol{\Phi}\mathbf{b} \quad (3.8)$$

dove \mathbf{b} è un parametro vettoriale che descrive il grado di variazione di un grafo S_k nella direzione di ogni componente principale. Gli autori testano il loro modello per esplorare la variazione strutturale presente in un insieme di grafi ricavati da volti.

L'approccio appena descritto risulta particolarmente interessante perché permette di utilizzare metodi statistici sui grafi. Vi sono tuttavia due aspetti che possono risultare problematici:

- Gli autori di [37] propongono un modo molto semplice di rappresentare un grafo con un vettore, ossia vettorizzando la matrice di adiacenza. Questo comporta che dato un grafo di n nodi, esso verrà rappresentato da un vettore di n^2 elementi. Una rappresentazione più compatta che contenesse solo alcune informazioni peculiari del grafo permetterebbe di ottenere un beneficio in termini computazionali, conservando nel vettore \mathbf{a}_k le informazioni più rappresentative della linearizzazione della matrice di adiacenza.
- I grafi campione vengono considerati nella loro interezza e non si considera il fatto che si potrebbe individuare all'interno di ciascun grafo dei pattern ricorrenti, dei sottografi che si ripetono in grafi diversi. Utilizzare quindi come unità atomica di un eventuale modello generativo non più i grafi, ma i singoli sottografi che si individuano in ciascun grafo, potrebbe condurre a dei benefici sfruttando questo tipo di informazione aggiuntiva.

Sulla base di queste considerazioni, in [70, 71, 72, 73] è stato implementato un modello generativo sui grafi, che sia capace di utilizzare le tecniche descritte finora sui grafi in modo tale da produrre nuovi grafi coerenti a quelli di partenza.

La prossima sezione si propone di evidenziare le idee chiave dell'approccio di [70, 71, 72, 73], che verrà più avanti usato come riferimento per il nostro lavoro, indirizzato a un ambito specifico quale il mapping semantico.

3.4 Un metodo generativo, spettrale e part-based

I grafi non sono rappresentati naturalmente in uno spazio vettoriale dal momento che non esiste un ordinamento naturale dei nodi dei grafi, e diversi ordinamenti portano a diverse rappresentazioni della struttura del grafo. Ciò diventa problematico, dal momento che semplici quantità statistiche come la media e la varianza diventano difficili da definire per un insieme di grafi.

Il problema chiave nell'utilizzo delle rappresentazioni dei grafi si trova nella misurazione della loro similarità strutturale. Questo è un problema complesso, in quanto non esiste un ordinamento esplicito delle parti, e tipicamente devono essere stabilite delle corrispondenze prima che si possa stabilire la similarità. Gli algoritmi di allineamento oggetto della Sezione 3.3 sono proprio finalizzati all'individuare similarità strutturali all'interno di un insieme di grafi.

Come viene rilevato in [70], la rappresentazione spettrale di un grafo presenta due vantaggi significativi.

Per prima cosa, la decomposizione spettrale già di per sé fornisce un ordinamento parziale della rappresentazione dei grafi attraverso l'ordine degli autovalori (in [68] l'allineamento sfrutta la rappresentazione spettrale). Se ad esempio due grafi sono uguali ma non allineati, le matrici degli autovalori ordinati coincidono. Completando il processo di ordinamento, è possibile identificare quali parti del grafo corrispondono reciprocamente e quindi definire le quantità statistiche come media e varianza che sono necessarie per realizzare un modello generativo.

L'altro punto chiave da risolvere per avere un modello generativo riguarda la capacità di ricostruire pattern da punti nello spazio dei campioni. L'uso di un modello spettrale permette di generare nuovi grafi a partire dalle rappresentazioni spettrali dei grafi di input. La decomposizione spettrale rappresenta difatti le componenti principali dei grafi.

Dato un insieme di grafi di input, vi possono essere differenze consistenti tra i singoli grafi. Ciononostante, è possibile che grafi apparentemente molto diversi siano invece costituiti dalla composizione di stesse strutture o da strutture molto simili. Allineare sottografi di dimensioni più ridotte potrebbe risultare più semplice ed efficace. Inoltre si potrebbe concepire un modello generativo che utilizzi i sottografi come componenti da disassemblare nei grafi campione e riassemblare nei grafi generati. Questa idea viene discussa in [72].

Un approccio *part-based* scompone gli oggetti in parti che possano essere utilizzate separatamente. In un approccio part-base sui grafi, le parti sono i sottografi.

In [72] è descritto un approccio part-based per la generazione di grafi. La variazione di pattern strutturali nell'insieme di grafi di partenza viene modellata suddividendo ogni grafo in sottografi e osservando quali sottografi sono presenti in ogni grafo e come sono connessi tra loro. Tale approccio è efficace su quei tipi di dati che sono naturalmente portati a un processo di partizione, in cui cioè la presenza di sottostrutture è evidente, anche a livello intuitivo.

Un esempio di grafi facili da decomporre sono le strutture chimiche, come discusso in [73]. Un approccio part-based sui grafi individua sottografi tali per cui nodi appartenenti a uno stesso sottografo sono molto più connessi tra loro rispetto a nodi appartenenti a sottografi diversi. Questo significa effettuare la segmentazione di un grafo, come in [58].

Una volta determinati i sottografi, è possibile compiere un'analisi su di essi. Ad esempio, è possibile verificare se vi sono sottografi ricorrenti in grafi diversi, oppure computare la similarità tra sottografi e raggrupparli in cluster.

Un *cluster* è un raggruppamento di elementi che tra loro presentano delle caratteristiche simili. Raggruppare i sottografi in cluster permette di considerare assieme i sottografi tra loro simili.

Il metodo proposto in [72] è quello di generare i nuovi grafi campionandoli da tre diversi passaggi ed utilizzando il loro raggruppamento in cluster:

- Un primo passaggio determina la configurazione dei cluster che sarà presente nel nuovo grafo. La configurazione dei cluster stabilisce se e quanti sottografi appartenenti a un determinato cluster comporranno il grafo generato.
- Successivamente vengono scelti i sottografi che rappresentano ogni cluster.
- Infine, si campionano le connessioni che collegano i sottografi facendo sampling da una approssimazione delle connessioni tra cluster di sottografi a una distribuzione normale multivariata.

3.5 Un modello generativo sui grafi per predire le parti inesplorate di un edificio

Il modello generativo proposto sfrutta la nozione di tipologia edilizia [39, 40]. Una *tipologia edilizia* è un insieme di edifici che condividono la stessa funzione primaria [55]. La funzione primaria rappresenta l'uso a cui un edificio è destinato, ovvero quali sono le attività che vi si svolgeranno all'interno. Ad esempio, un edificio adibito ad attività lavorativa appartiene alla classe di tipologia edilizia UFFICI, mentre una scuola appartiene, alla tipologia edilizia SCUOLE. Questa semplice catalogazione, immediata per un essere umano, contiene al suo interno un gran numero di informazioni sul singolo ambiente.

La funzione di un edificio impone la sua struttura, la sua planimetria e la struttura delle sue stanze. Ogni edificio, avendo una precisa funzione, condivide alcune caratteristiche strutturali con tutti gli altri edifici che svolgono la medesima funzione. Queste caratteristiche vengono utilizzate dalle persone quotidianamente per localizzarsi e orientarsi all'interno di un edificio.

Utilizzare ogni volta edifici della stessa tipologia permette di meglio identificare le loro caratteristiche comuni, e facilita inoltre i processi di segmentazione, clustering e campionamento dei grafi che rappresentano edifici.

Il nostro sistema consta di alcune fasi principali per ottenere il modello generativo e fare predizione. Dato un insieme di grafi \mathcal{G} , ciascuno rappresentante un piano di edifici appartenenti a una tipologia edilizia:

1. ogni grafo viene diviso in sottografi;
2. tutti i sottografi così ottenuti vengono raggruppati in cluster composti da sottografi simili;
3. dall'analisi delle connessioni tra i sottografi nei grafi originali, vengono modellate le connessioni tra i cluster;
4. i sottografi che costituiscono il nuovo grafo sono campionati dai cluster;
5. le connessioni tra i cluster sono campionate al fine di connettere tra loro i sottografi e formare il grafo finale.

Il sistema è basato su un modello generativo spettrale part-based.

- E' **generativo** perché permette di estrarre nuovi campioni dalla distribuzione di grafi creata sui dati dei piani di edifici originali \mathcal{G} .
- E' **spettrale** poiché sfrutta le proprietà dei grafi in relazione agli autovalori e agli autovettori delle loro matrici di adiacenza.
- E' **part-based** perché si fonda sul partizionamento dei grafi in sotto-grafi rilevanti sia per il suo sviluppo sia per il campionamento di nuove istanze di grafi.

Nel prossimo capitolo descriviamo nel dettaglio le fasi del modello realizzato in questo lavoro tesi.

Capitolo 4

Un modello generativo di grafi per il mapping semantico

“Gli umani eccellono in alcuni campi, i robot in altri. Gli umani posseggono emozioni e sentimenti altamente sviluppati. Una consapevolezza estetica. Siamo sensibili ai colori e ai suoni e a quello che ci dice il tatto e alla musica dolce unita al buon vino. Tutte ottime cose. Digne di interesse. Ma questi campi sono del tutto al di là dei robot. I robot sono puramente intellettuali. Anche questo va benissimo. Entrambi i nostri regni sono di per sé buoni. Gli umani emotivi, sensibili all’arte e alla musica e alla drammaticità. I robot che pensano e progettano e creano macchine. Ma questo non significa che siamo uguali.”

Philip K. Dick, *James P. Crow*

In questo capitolo si descrive il modello generativo di grafi proposto in questa tesi, realizzato per essere applicato nell’ambito del mapping semantico.

L’utilizzo di un modello generativo si pone l’obiettivo di effettuare il campionamento da una distribuzione di grafi, che rappresentano mappe semantiche, in modo che i nuovi dati campionati possano essere impiegati per effettuare predizione semantica sulla topologia e il tipo di stanze di una parte inesplorata di edificio appartenente a una certa tipologia edilizia.

Il nostro approccio è basato sull’analisi spettrale dei grafi e segue le linee guida di [73]. Dal momento che il nostro lavoro è indirizzato a un ambito specifico quale il mapping semantico, l’implementazione vera e propria del modello sarà più specifica e funzionale all’ambito trattato.

Per un robot, la possibilità di campionare nuovi dati da un modello generativo permette di inferire nuove informazioni circa le stanze e la struttura di un edificio a partire da una base di conoscenza preesistente limitata rispetto agli ambienti in cui deve o dovrà operare.

Prima di descrivere nel dettaglio il sistema da noi proposto, è opportuno aprire una breve parentesi riguardo a due aspetti fondamentali che concorrono nella realizzazione di un buon modello generativo che debba essere applicato a un ambito specifico.

- Un primo aspetto è la coerenza del modello generativo rispetto alla distribuzione reale dei dati. Con essa intendiamo che la distribuzione di probabilità con cui si modellano i dati di input deve essere quanto più possibile vicina alla distribuzione reale dei dati. Ciò significa che se la distribuzione dalla quale si campionano nuovi grafi è identica alla distribuzione reale, allora i dati campionati e i dati di input condividono la stessa distribuzione.
- Un secondo aspetto è l'efficacia del sampling dal modello generativo ai fini della predizione. Ciò significa che i dati che vengono campionati devono contribuire alla predizione in caso di conoscenza parziale dei dati.

La solidità del sistema basato su un modello generativo si misura in base a entrambi gli aspetti. Il primo garantisce che i dati campionati presentino determinate caratteristiche e proprietà. Il secondo garantisce invece che il modello sia effettivamente utilizzabile in un sistema specifico. Nel caso del mapping semantico questi aspetti determinano la capacità del robot di effettuare predizioni esatte sulla struttura di un edificio senza che questa sia precedentemente nota.

Il sistema che proponiamo è *sequenziale* e suddiviso in passi. Ognuno di questi passi metodologici è incentrato su un particolare tipo di elaborazione dei dati che è da un lato funzionale al passo successivo, e dall'altro è facilmente descrivibile in termini di mapping semantico. Segmentazione, clustering e campionamento sono quindi non solo passi metodologici ma anche logici: ognuno di essi intende ha un'interpretazione diretta riferita alle mappe semantiche.

4.1 Distinzione tra costruzione del modello generativo e campionamento

Un approccio di tipo generativo, a prescindere dal tipo di dati e dal modello teorico utilizzati, è suddiviso in due momenti fondamentali.

- In un primo momento, la costruzione del modello generativo, i dati di input sono elaborati con lo scopo di dedurre la loro distribuzione. Per fare ciò, i dati di input possono essere semplicemente riorganizzati in una diversa struttura, oppure possono esplicitare nuove informazioni estratte dai dati di input da algoritmi specifici. Il risultato della costruzione del modello generativo è la presentazione dei dati in base alla distribuzione a cui sono stati approssimati.
- Una volta costruito il modello generativo sui dati di input, questo permette di effettuare il campionamento che porta alla formazione di nuovi dati che condividono la medesima distribuzione con quelli di input.

La costruzione di un modello generativo permette di evidenziare feature e proprietà implicitamente presenti nei dati iniziali. I singoli approcci si distinguono poi per il tipo di elaborazione effettuata sui dati per esplicitare queste caratteristiche, e di conseguenza per il tipo di variabili sulle quali è ottenuta una distribuzione approssimata dei dati.

Nel nostro approccio, la costruzione del modello generativo implica la ristrutturazione dei dati originali, dove i grafi di input sono segmentati in sottografi, e l'insieme costituito da tutti i sottografi è raggruppato in cluster individuati in base alla similarità tra sottografi. Queste due operazioni si basano su due assunzioni che noi facciamo circa la composizione e la struttura dei grafi.

- La prima è che un grafo sia descrivibile come un insieme di parti, i sottografi, e che ogni grafo sia il risultato della composizione di queste parti.
- La seconda è che esistano delle categorie di sottografi simili tra loro, i cluster, e che ogni grafo sia composto da elementi di tali cluster secondo una certa configurazione.

In questo modo si cerca di sopperire alla difficoltà di utilizzo dei grafi nell'ambito dei modelli generativi.

Un modello generativo di grafi per la predizione semantica

In Figura 4.1 è mostrata la sequenza delle attività fondamentali che permettono la predizione semantica di parti dell'ambiente ignote a priori. Da un insieme noto di edifici vengono estratte le mappe semantiche, che sono rappresentate da grafi. Il lavoro di tesi, evidenziato dal riquadro in figura, consiste nell'utilizzare grafi, che nel nostro caso rappresentano mappe semantiche di edifici, per creare un modello generativo di grafi su cui poter campionare nuovi grafi. Il risultato del campionamento è un insieme di nuovi grafi che, rappresentando a loro volta degli ipotetici edifici o parti di edifici, possono essere utilizzati da un robot per la predizione di parti inesplorate di un edificio.

Dal momento che il modello generativo che proponiamo è specificatamente indirizzato al mapping semantico, assumiamo l'equivalenza tra edificio e mappa semantica: in altre parole, in questo contesto ci riferiamo agli edifici tramite il concetto di mappa semantica. Analogamente, è possibile fare un parallelo tra mappe semantiche e grafi, cioè i grafi su cui è costruito il modello generativo sono interpretabili in termini di mappe semantiche.

4.2 La base di conoscenza del robot

I dati di input nel nostro modello sono i grafi che rappresentano le mappe semantiche di piani di edifici appartenenti a una certa tipologia edilizia. Rispetto al problema del mapping semantico l'insieme di questi dati rappresenta la base di conoscenza del robot, cioè l'insieme degli edifici di cui il robot conosce già la topologia e il tipo di stanze che li compongono.

Nel nostro caso, una mappa semantica è rappresentata come un grafo non orientato $G = (N, E)$, dove ogni nodo $n \in N$ è una stanza e ogni lato $e = (n, n') \in E$ rappresenta un collegamento fisico tra due stanze n and n' (ad esempio tramite una porta)¹. Una label semantica può essere assegnata a ogni nodo attraverso una funzione $f : N \rightarrow \mathcal{L}$.

Nel nostro caso, un nodo n può essere associato a una label $f(n)$ in $\mathcal{L} = \{\text{CORRIDOIO, SALA, INGRESSO, STANZA FUNZIONALE PICCOLA, STANZA FUNZIONALE MEDIA, STANZA FUNZIONALE GRANDE, STANZA DI SERVIZIO PICCOLA, STANZA DI SERVIZIO MEDIA, STANZA DI SERVIZIO GRANDE}\}$.

¹Per semplicità, a volte abusiamo della notazione scrivendo $n \in G$ ma intendendo $n \in N$ e $G = (N, E)$.

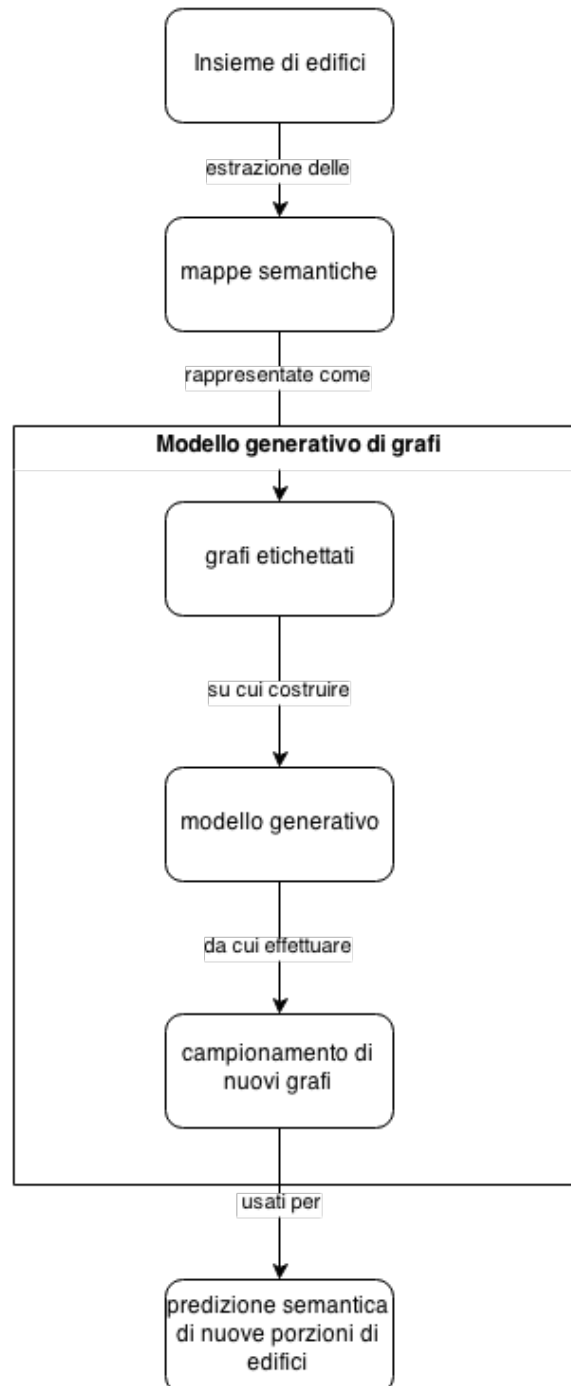


Figura 4.1: Modello della predizione semantica di edifici. Il riquadro evidenzia gli aspetti trattati nel lavoro di tesi.

Tutte le rappresentazioni matriciali per grafi non orientati mostrati nella Sezione 3.2 possono essere utilizzate. La rappresentazione che utilizziamo come riferimento, se non esplicitamente indicato, è una rappresentazione compatta che, oltre a memorizzare per ciascuna coppia di nodi la presenza o meno di un lato che li colleghi (come nella matrice di adiacenza \mathbf{M}), tiene traccia sulla diagonale principale degli identificatori delle label corrispondenti ai nodi. Chiamiamo la matrice \mathbf{G} *matrice del grafo semantico*.

$$G(n_i, n_j) = \begin{cases} g(n), & \text{se } n_i = n_j \\ 1, & \text{se } n \neq n' \wedge (n_i, n_j) \in E \\ 0, & \text{altrimenti} \end{cases} \quad (4.1)$$

laddove $g : \mathcal{L} \rightarrow \mathbb{N}$ associa a ogni label memorizzata in f un identificativo numerico.

È bene sottolineare che questo tipo di rappresentazione non va in alcun modo confuso con la matrice di adiacenza di un grafo con i pesi sui nodi. Gli elementi sulla diagonale principale di \mathbf{G} non sono pesi ma hanno l'esclusivo scopo di identificare la label della stanza rappresentata.

Indichiamo con $\mathbf{M} = \text{adj}(\mathbf{G})$ l'operazione invertibile che permette di passare dalla matrice del grafo semantico alla matrice di adiacenza.

Il tipi di mappe semantiche rappresentate dai grafi

Il nostro metodo assume le mappe semantiche come già date: questo significa che le mappe semantiche devono essere state ottenute precedentemente in qualche modo. Nel nostro caso, ad esempio, sono state ottenute manualmente a partire da planimetrie su carta di edifici. Alternativamente, le mappe potrebbero essere state ottenute direttamente dal robot durante precedenti esplorazioni.

A prescindere dalle modalità attraverso le quali sono state ottenute le mappe, l'elemento importante da considerare è che le mappe siano rappresentative della tipologia edilizia che si sta analizzando.

Per rappresentatività di una tipologia edilizia si intende che, dato un insieme di input di grafi, essi contengono nel complesso una varietà di sottografi sufficiente a comporre un grafo di quella tipologia; in un approccio part-based infatti i sottografi rappresentano le sottostrutture ricorrenti che si trovano in un edificio. Se il processo di segmentazione ha il compito di dividere i grafi in modo da ricavarne sottografi significativi, è comunque richiesto che i grafi di partenza costituiscano un campione adeguato su cui

generare poi nuovi grafi.

In questa tesi vengono utilizzati i risultati di classificazione delle tipologie edilizie ottenuti in [39, 40]. In particolare ogni tipologia si contraddistingue per le label semantiche che identificano le stanze che compongono gli edifici di quella tipologia. Avere un sistema di label che classifichi in maniera chiara, esaustiva ma non ridondante le stanze influisce sulla qualità dell'informazione semantica contenuta in una mappa.

In questo lavoro consideriamo tre tipologie edilizie: le SCUOLE, gli UFFICI e le ABITAZIONI.

4.3 Segmentare i grafi

Il primo passo della fase di costruzione del modello è costituito dalla segmentazione dei grafi di input.

Ogni grafo $G \in \mathcal{G}$ è segmentato in sottografi non sovrapposti

$$S = \{s_1, s_2, \dots\}$$

in modo tale che² $\forall s_i, s_j \in S, s_i \cap s_j = \emptyset$ e $\bigcup_{s \in S} s = G$, utilizzando il metodo di partizionamento spettrale di Ncut [58] riassunto nell'Algoritmo 1.

Dal momento che Ncut utilizza le matrici di adiacenza, nel caso di grafi dotati di label è necessario innanzitutto passare dalla matrice del grafo semantico \mathbf{G}_G alla matrice di adiacenza $\mathbf{M}_G = \text{adj}(\mathbf{G}_G)$. Viceversa, alla matrice di adiacenza \mathbf{M}_s di un sottografo s sarà fatta corrispondere la matrice del sottografo semantico $\mathbf{G}_s = \text{adj}^{-1}(\mathbf{M}_s)$, secondo la notazione introdotta nella Sezione 4.2.

L'algoritmo di Ncut dai noi utilizzato non presenta modifiche alla versione presentata dagli autori. Vi sono tuttavia alcuni aspetti su cui è necessario soffermarsi.

4.3.1 Risolvere il problema spettrale generalizzato

Gli autori di Ncut dimostrano che l'individuazione del taglio ottimo sia un problema **NP**-completo, per cui sviluppano un algoritmo che individua una soluzione subottima al problema riconducendolo a un problema spettrale, computazionalmente non oneroso.

²Le operazioni \cup e \cap sono definite banalmente per i grafi $G = (N, E)$ e $G' = (N', E')$ come $G \cup G' = (N \cup N', E \cup E')$ e $G \cap G' = (N \cap N', E \cap E')$, rispettivamente.

Input: Un grafo $G = (N, E)$ e la sua matrice di adiacenza \mathbf{M} , un parametro di controllo ρ

Output: Due (o più) partizioni s_a, s_b of G tali che $s_a \cup s_b = G$,
 $s_a \cap s_b = \emptyset$

1 computare la matrice diagonale \mathbf{D} di G , contenente il grado di ogni nodo

2 risolvere $(\mathbf{D} - \mathbf{M})\mathbf{x} = \lambda\mathbf{D}\mathbf{x}$ per gli autovettori associati agli autovalori più piccoli

3 bipartire il grafo G in accordo alla grandezza dei componenti dell'autovettore \mathbf{v} associato al secondo autovalore più piccolo

4 valutare la partizione utilizzando il valore di *normalized-cut*
 $Ncut(s_a, s_b)$

$$Ncut(s_a, s_b) = \frac{Cut(s_a, s_b)}{Assoc(s_a, G)} + \frac{Cut(s_a, s_b)}{Assoc(s_b, G)}$$

dove $w(u, v)$ è il peso dei lati che collegano i nodi $u, v \in N$ e

$$Cut(s_a, s_b) = \sum_{u \in s_a, v \in s_b} w(u, v)$$

$$Assoc(s_a, G) = \sum_{u \in s_a, t \in G} w(u, t)$$

5 suddividere ricorsivamente le partizioni se $Ncut(s_a, s_b) < \rho$, altrimenti terminare

Algoritmo 1: Algoritmo normalized cut (Ncut).

La versione standard del problema degli autovalori richiede di risolvere l'equazione lineare $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$, dove \mathbf{A} rappresenta la matrice di cui si vogliono calcolare gli autovalori e gli autovettori, \mathbf{x} rappresenta un autovettore e λ l'autovalore associato.

L'algoritmo di Ncut richiede di risolvere il problema spettrale *generalizzato*, cioè una versione del problema spettrale in cui gli autovalori e gli autovettori legano due matrici \mathbf{A} e \mathbf{B} tramite la relazione $\mathbf{A}\mathbf{x} = \lambda\mathbf{B}\mathbf{x}$. Nel caso specifico, la matrice \mathbf{A} è rappresentata da $(\mathbf{D} - \mathbf{M}) = \mathbf{L}$, cioè dalla matrice Laplaciana e la matrice \mathbf{B} è rappresentata da \mathbf{D} , la matrice diagonale dei gradi dei nodi.

Gli autori di Ncut riconducono il problema spettrale generalizzato al problema standard risolvendo $\mathbf{D}^{-\frac{1}{2}}(\mathbf{D} - \mathbf{M})\mathbf{D}^{-\frac{1}{2}}\mathbf{x}' = \lambda\mathbf{x}'$, avendo posto $\mathbf{x}' = \mathbf{D}^{\frac{1}{2}}\mathbf{x}$. Ciò è possibile poiché la matrice Laplaciana è semidefinita positiva [49].

Risolvere il problema spettrale standard richiede $O(n^3)$ operazioni, dove n è il numero di nodi del grafo. Questo può rappresentare un problema

nel caso, in cui i grafi siano composti da molti nodi, ad esempio in image segmentation, in cui ogni nodo di un grafo rappresenta un pixel e quindi vi possono essere milioni di nodi. Nel nostro caso il numero di nodi di un grafo è dato dal numero di stanze contenute in un edificio. Le istanze tipiche di edifici non sono tali da richiedere particolari ottimizzazioni per poter risolvere il problema spettrale di Ncut in tempi ridotti.

In ogni caso, gli autori di [58] evidenziano come sia possibile ridurre la complessità fino a $O(n^{\frac{3}{2}})$ sfruttando il fatto che i grafi sono spesso connessi solo localmente e portano a sistemi di autovalori sparsi. Inoltre non è necessario calcolare tutti gli autovettori, ma solo quelli relativi agli autovalori più piccoli, perché sono questi a contenere l'informazione utile al partizionamento dei grafi. Questa riduzione di complessità è possibile attraverso un risolutore quale il metodo di Lanczos, descritto in [24].

4.3.2 Utilizzare il Fiedler value per partizionare il grafo

Una volta che sono stati determinati gli autovalori e gli autovettori, si procede al processo di partizionamento del grafo vero e proprio. Questo viene fatto esaminando l'autovettore corrispondente al secondo autovalore più piccolo, denominato *Fiedler value*. Gli autori di Ncut hanno verificato che le componenti del Fiedler value \mathbf{f} determinano una partizione del grafo che è in accordo con la definizione di Ncut. Questo importante risultato teorico porta a una conseguenza importante: non è necessario calcolare tutte le possibili partizioni di un grafo (che da un punto di vista computazionale sarebbe proibitivo), ma è sufficiente ricavare il Fiedler value.

Nella pratica, i sottografi vengono individuati dalle n componenti del Fiedler value. In un caso ideale, le componenti di \mathbf{f} dovrebbero assumere solo due valori discreti di segno opposto, e in questo caso i nodi corrispondenti alle componenti di segno positivo formerebbero una partizione e i nodi corrispondenti alle componenti di segno negativo l'altra. Questo accade in casi di grafi strutturalmente semplici, ad esempio in presenza di un grafo formato da due gruppi di nodi molto connessi che sono collegati tra loro da un lato. Può tuttavia accadere che le componenti di \mathbf{f} assumano valori continui. In questi casi è necessario identificare un *punto di divisione* che determini le due partizioni. Il punto di divisione è un valore p per cui le componenti minori di esso formano la prima partizione e le componenti maggiori formano la seconda.

Il punto di divisione è un valore che deve essere ricavato empiricamente ogni volta che si divide un grafo. Nel caso citato prima in cui \mathbf{f} assume

valori discreti esiste un intervallo molto ampio di valori che possono essere usati come punti di divisione. Nei casi più complessi è necessario utilizzare un'euristica. L'approccio più semplice consiste nel prendere lo zero o il valore mediano come punto di divisione: lo zero rappresenta tipicamente il valore medio delle componenti del Fiedler value, mentre il valore mediano fa sì che il grafo venga diviso in sottografi di dimensioni uguali, il risultato più richiesto nella divisione con Ncut.

L'euristica da noi utilizzata segue l'approccio proposto dagli stessi autori di Ncut. Vengono definiti l punti di divisione equamente spaziatati tra loro, in un intervallo centrato in zero; per ognuno di essi si individuano le partizioni potenziali e se ne calcola il valore di normalized cut come definito nell'Algoritmo 1. Tra tutte queste possibili partizioni si sceglie quella che presenta il valore di Ncut minore. In questo modo non ci si affida a quello che è il comportamento medio del Fiedler value ma si ricerca all'interno di una serie di potenziali processi di partizionamento. Ovviamente il numero di punti che vengono controllati non deve essere elevato in modo da giustificare il risparmio computazionale dato dall'utilizzo dell'algoritmo di Ncut. In generale sono sufficienti pochi controlli per garantire la determinazione della divisione ottima.

Poiché nel caso del mapping semantico gli edifici sono tendenzialmente divisi in macro-aree funzionali per cui stanze appartenenti alla stessa area sono molto connesse tra loro ma poco connesse a stanze appartenenti a un'area diversa (si pensi nelle abitazioni alla divisione tra zona giorno e zona notte), Ncut è stato scelto per segmentare il grafo di una mappa semantica in sottografi in modo agevole.

Un breve esempio può essere utile a mostrare i vantaggi della divisione effettuata da Ncut. In Figura 4.2 viene mostrato un grafo G e due possibili sottografi A e B . Una politica di taglio minimo considererebbe equivalente il taglio di uno qualunque tra i due lati compresi tra le coppie di nodi marcati con le lettere C e H. La nostra versione di Ncut invece privilegia a parità di numero di tagli una divisione in sottografi con un numero di lati simile. La divisione $(A, G \setminus A)$ porta a due sottografi con 6 e 7 lati, rispettivamente. La divisione $(B, G \setminus B)$ porta a due sottografi con 4 e 9 lati, rispettivamente. Per questo motivo la nostra versione di Ncut effettua il taglio $(A, G \setminus A)$.

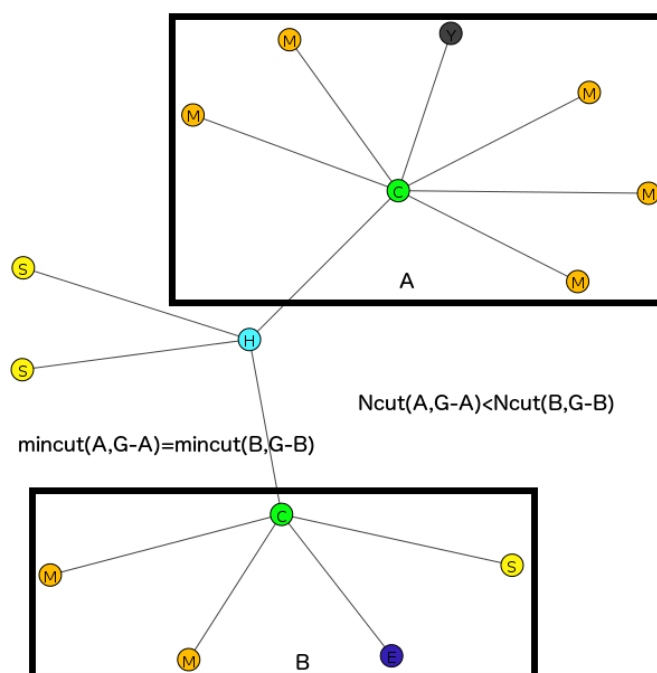


Figura 4.2: Esempio su come $Ncut$ privilegia il taglio di A rispetto al taglio di B .

4.3.3 Gestire la ricorsione

Applicando $Ncut$ ricorsivamente è possibile segmentare i grafi.

Un punto importante riguarda la condizione di arresto, cioè quando interrompere la ricorsione e la seguente divisione in sottografi. Esistono due approcci possibili al problema, in base all'obiettivo desiderato:

- L'obiettivo è ottenere k partizioni. L'algoritmo termina quando il grafo iniziale è stato diviso in k sottografi.
- L'obiettivo è dividere un sottografo intermedio ogni volta in cui il valore del taglio subottimo non eccede rispetto a un parametro di threshold ρ .

Seguiamo il secondo approccio per una ragione legata ai dati. L'obiettivo che ci poniamo in questa fase è ottenere sottografi significativi. Ogni volta che si esegue una divisione di una parte di grafo in due sottografi, a tale divisione è associato il valore di $Ncut$ che esprime il peso del taglio rispetto a entrambi i sottografi. Un valore particolarmente basso di $Ncut$ indica che il taglio effettuato ha un impatto minimo rispetto a due sottografi che presentano molte connessioni tra i loro nodi interni.

Dal momento che i sottografi costituiscono le componenti base del modello generativo, la nostra divisione in sottografi deve rispondere a due esigenze: da un lato da ogni grafo è opportuno massimizzare il numero di componenti (sottografi) estratte. D'altro canto, è anche necessario che le componenti rappresentino parti significative di edifici; un sottografo costituito da due soli nodi e il lato tra essi (cioè due stanze adiacenti), ad esempio, non sarebbe una componente particolarmente utile al processo generativo, non rappresentando una struttura significativa. L'arresto della ricorsione al momento considerato adatto permette di effettuare un compromesso tra queste due esigenze.

Dato un sottografo s , il nostro approccio verifica quale sarebbe il valore N_{cut} relativo alla suddivisione ulteriore di s ; a questo punto un parametro di threshold ρ stabilisce se il partizionamento di s debba effettivamente avvenire. La scelta di ρ è effettuata empiricamente in modo da evitare sottografi troppo piccoli e sottografi troppo grandi: i primi sono poco significativi ai fini del processo generativo, i secondi vincolano eccessivamente la struttura del grafo generato.

La Figura 4.3 illustra il significato del parametro ρ in casi di alcuni grafi con pochi nodi come potrebbe essere una parte di grafo ottenuta dopo alcune ricorsioni di N_{cut} . Per semplicità, abbiamo rappresentato grafi simmetrici rispetto al lato centrale oggetto di un eventuale taglio. Per ogni grafo S_i è stato indicato il parametro $\rho_{min}(S_i)$ minimo al di sopra del quale S_i viene diviso a metà. Questo valore non è altro che il valore di N_{cut} del sottografo.

Per determinare ρ è possibile costruire una serie di esempi di grafi semplici e considerare per ognuno di essi se sia opportuno o meno effettuare una divisione: ad esempio, se si ritiene che S_1 , S_2 e S_3 debbano essere divisi in due, ma S_4 non debba essere diviso, si considererà $\frac{1}{2} \leq \rho < \frac{2}{3}$.

4.3.4 Memorizzare le connessioni

Una volta segmentato il grafo con N_{cut} , il nostro approccio tiene traccia delle connessioni tra i sottografi rispetto al grafo originale. Questo implica che ogni volta che un grafo o un sottografo G viene diviso in due sottografi s_a e s_b , vengano memorizzati quali sono i lati che sono stati tagliati e quindi i nodi di s_a e i nodi di s_b su cui è stato praticato un taglio.

Definiamo *connessione* tra sottografi un lato che collega due sottografi, tagliato a seguito del processo di segmentazione. L'utilizzo di un termine

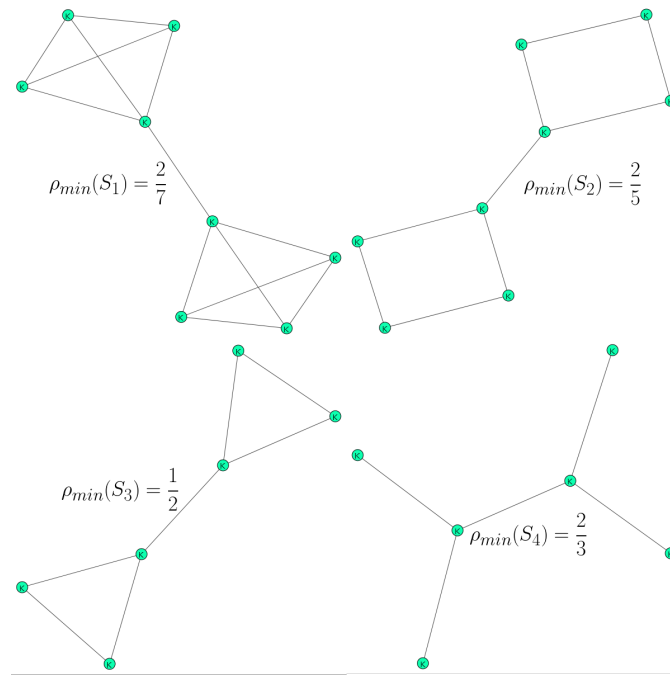


Figura 4.3: Esempi di grafi simmetrici: per ciascuno è indicato il parametro di threshold minimo $\rho_{min}(S_i)$ per cui S_i viene diviso a metà.

differente serve a facilitare la distinzione concettuale di un elemento (un lato che unisce due sottografi) che viene utilizzato in due contesti differenti.

- Un taglio viene applicato nella fase di segmentazione e risponde all'esigenza di dividere un grafo complesso in un due sottografi più semplici.
- Una connessione permette il processo inverso: l'insieme delle connessioni tra tutti i sottografi ottenuti da un grafo G permette di ricostruire G .

La necessità di memorizzare informazioni riguardanti le connessioni è dettata dal fatto che la fase di campionamento deve concludersi ri assemblando i sottografi campionati in un unico grafo, l'operazione simmetrica rispetto a quella appena vista con la segmentazione.

A questo proposito utilizziamo una struttura dati che memorizzi ogni singola connessione. Questo significa che per ogni coppia di sottografi $(s_x, s_y) \in G$ devono essere registrate le connessioni tra essi. Una matrice \mathbf{T}_{s_x, s_y} memorizza le connessioni tra i due sottografi: denotando $n_{s_x, i}$ l' i -esimo nodo di s_x (analogamente per $n_{s_y, j}$),

$$\mathbf{T}_{s_x, s_y}(i, j) = \begin{cases} 1, & \text{se } (n_{s_x, i}, n_{s_y, j}) \in E \\ 0, & \text{altrimenti} \end{cases} \quad (4.2)$$

Dato un grafo $G \in \mathcal{G}$, la connettività tra i suoi sottografi può essere memorizzata in una matrice a blocchi chiamata $\mathbf{I}_{conn}(G)$. Ad esempio, se G ha tre sottografi, s_a , s_b , e s_c ,

$$\mathbf{I}_{conn}(G) = \left[\begin{array}{c|c|c} \mathbf{G}_{s_a} & \mathbf{T}_{s_a, s_b} & \mathbf{T}_{s_a, s_c} \\ \hline 0 & \mathbf{G}_{s_b} & \mathbf{T}_{s_b, s_c} \\ \hline 0 & 0 & \mathbf{G}_{s_c} \end{array} \right]$$

dove \mathbf{G}_{s_a} è la matrice del sottografo s_a , corrispondente alla matrice \mathbf{M}_{s_a} di adiacenza di s_a (analogamente per \mathbf{G}_{s_b} and \mathbf{G}_{s_c}), e \mathbf{T}_{s_a, s_b} è la matrice di connettività che rappresenta i lati di G tra i nodi in s_a e i nodi in s_b (analogamente per \mathbf{T}_{s_a, s_c} e \mathbf{T}_{s_b, s_c}). \mathbf{I}_{conn} è la matrice tri-dimensionale che memorizza $\mathbf{I}_{conn}(G)$ per tutti i grafi $G \in \mathcal{G}$.

E' facile notare che la matrice a blocchi $\mathbf{I}_{conn}(G)$ rappresenta esattamente il grafo di partenza G : se la si considera come un'unica matrice non è altro che una permutazione della matrice \mathbf{G}_G del grafo G .

4.4 Raggruppare i sottografi in cluster

Il secondo passo importante della costruzione del modello generativo è costituito dal *clustering* dei sottografi. Un cluster è un gruppo di elementi che presentano delle caratteristiche simili tra loro e diverse da altri. Tecniche di clustering possono essere applicate su dataset molto grandi, al fine di individuare raggruppamenti all'interno dei quali gli elementi sono omogenei tra loro, per studiarne le proprietà in modo più facile. Il clustering è infatti un approccio al campo dell'unsupervised learning che permette di categorizzare i dati in base a caratteristiche insite in essi.

Una volta che tutti i grafi di input sono stati divisi in sottografi, tutti i sottografi S_ℓ ottenuti dai grafi $G_\ell \in \mathcal{G}$ vengono considerati insieme in $\mathcal{S} = S_1 \cup S_2 \cup \dots \cup S_{|\mathcal{G}|}$. \mathcal{S} rappresenta il dataset di sottografi, con $|\mathcal{S}| = m |\mathcal{G}|$ elementi, dove m indica il numero medio di sottografi contenuti in un grafo.

4.4.1 Identificare i sottografi simili

Requisito necessario per poter effettuare il clustering è identificare le similarità tra i sottografi. L'approccio da noi seguito prevede che per ogni coppia di sottografi s_i e s_j di \mathcal{S} venga calcolato un *valore di affinità*: $a_{i,j} = \phi(s_i, s_j)$.

Per prima cosa, s_i e s_j sono allineati utilizzando il metodo di Gold e Rangarajan [23] presentato nel Capitolo 3 e riportato nell'Algoritmo 2. Un'operazione di allineamento tra due sottografi s_i e s_j computa e applica la migliore matrice di permutazione che proietta ogni nodo del primo grafo s_i su un nodo del secondo grafo s_j . Intuitivamente, il metodo di allineamento utilizzato cerca di massimizzare il numero di lati di s_i e s_j che coincidono dopo aver allineato i nodi di s_i ai nodi di s_j . L'allineamento permette un ordinamento lessicografico dei nodi nei grafi.

Successivamente, l'affinità ϕ tra due sottografi allineati è computata in accordo con questo metodo di allineamento. Nel caso particolare di un grafo privo di label, l'affinità è valutata solo utilizzando le caratteristiche topologiche del grafo. Altrimenti, ϕ dipende sia dalla struttura sia dalle label dei nodi dei grafi. L'affinità è una misura della similarità tra i sottografi.

E' da sottolineare il fatto che in questo frangente l'allineamento sia esclusivamente finalizzato a un corretto calcolo della affinità: questa dipende dall'allineamento, nel senso che due grafi isomorfi s_x e s_y misurano in generale una affinità diversa rispetto a un terzo sottografo s_z . Con l'allineamento si evitano cioè i casi di sottografi identici o molto simili ma disallineati l'uno rispetto all'altro: in caso contrario, il calcolo diretto dell'affinità porterebbe a un valore di ϕ molto inferiore rispetto a quello reale.

4.4.2 Spectral clustering di sottografi con Ncut

La conoscenza dell'affinità ϕ per ogni coppia di sottografi permette la creazione di un *grafo di affinità*, cioè di un grafo completo non orientato \mathcal{C} in cui ogni nodo s rappresenta uno dei sottografi di \mathcal{S} , e il peso del lato tra due nodi di \mathcal{C} , s_i e s_j , rappresenta la loro affinità $\phi(s_i, s_j)$.

La possibilità di rappresentare le similarità tra i vari sottografi con un grafo di affinità è molto utile, perché permette di riunire in cluster i sottografi simili utilizzando nuovamente l'algoritmo di Ncut (Algoritmo 1). Gli aspetti legati a Ncut sono pertanto i medesimi incontrati nella Sezione 4.3: questa volta lo scopo è dividere in componenti separate il grafo \mathcal{C} in modo tale che per ogni coppia di componenti, C_h e C_k , $C_h \cap C_k = \emptyset$ e $\bigcup_{C \subseteq \mathcal{C}} C = \mathcal{C}$. Ogni partizione C di \mathcal{C} è un cluster di sottografi. Dato un sottografo s , chiamiamo $\psi(s)$ il cluster s a cui appartiene, $\psi : \mathcal{S} \rightarrow \{C\}$.

```

Input: Due grafi  $s_i$  e  $s_j$  con nodi  $k \in s_i, l \in s_j$ , una funzione
           obiettivo per il matching di grafi  $E_{wg}$  (vedere i dettagli in
           [23]), un parametro di controllo  $\alpha \in [\alpha_0, \alpha_f]$  con un tasso di
           incremento  $\alpha_r$ , due parametri  $I_0 = 4$  e  $I_1 = 30$ 
Output: una matrice di permutazione  $\mathbf{P}$  che fa corrispondere i nodi
            $k \in s_i \rightarrow l \in s_j$ , affinità  $\phi \leftarrow f(E_{wg})$ 
1  $\alpha \leftarrow \alpha_0$ 
2  $\hat{\mathbf{P}}_{kl} \leftarrow (1 + \epsilon)$  dove  $\hat{\mathbf{P}}$  è  $\mathbf{P}$  estesa con variabili di slack
3 while  $\alpha < \alpha_f$  do
4   while  $\mathbf{P}$  non ha raggiunto la convergenza e # di iterazioni  $< I_0$ 
   do
5      $Q_{kl} \leftarrow \frac{\partial E_{wg}}{\partial P_{kl}}$ 
6      $P_{kl}^0 \leftarrow \exp(\alpha Q_{kl})$ 
7     while  $\hat{\mathbf{P}}$  non ha raggiunto la convergenza # di iterazioni  $< I_1$ 
     do
8       aggiornare  $\hat{\mathbf{P}}$  normalizzando lungo tutte le righe:
9        $\hat{P}_{kl}^1 \leftarrow \frac{\hat{P}_{kl}^0}{\sum_{l=0}^{s_j+1} \hat{P}_{kl}^0}$ 
10      aggiornare  $\hat{\mathbf{P}}$  normalizzando lungo tutte le colonne:
11       $\hat{P}_{kl}^0 \leftarrow \frac{\hat{P}_{kl}^1}{\sum_{k=0}^{s_i+1} \hat{P}_{kl}^1}$ 
     end
   end
12   $\alpha \leftarrow \alpha_r \cdot \alpha$ 
end

```

Algoritmo 2: Metodo di allineamento dei grafi, da [23].

Anche in questo caso la gestione della ricorsione avviene in base a un parametro di threshold ρ_c che, analogamente a quanto accade con ρ nella Sezione 4.3.3, regola le suddivisioni ricorsive del grafo di affinità, e di conseguenza determina il numero finale di cluster. Un buon valore del parametro è ottenuto per via empirica in base al livello di affinità minimo ϕ_{min} che si vuole garantire tra i sottografi appartenenti allo stesso cluster. Analizzando i risultati del clustering su grafi si può notare che, a parità di parametro ρ_c , gli stessi sottografi dotati di label vengono divisi in un numero di cluster maggiore rispetto a non considerare le label. Questa è una conseguenza della differenza nel calcolo delle affinità nei due casi, come spiegato in precedenza.

4.4.3 Eseguire l'allineamento a livello di cluster

L'obiettivo del clustering è identificare quante e quali sono le tipologie principali di sottografi presenti in \mathcal{S} . A questo punto risulta conveniente rendere più omogenei tra loro i sottografi appartenenti a ogni cluster.

Per ogni cluster C_h , un *sottografo di riferimento* $r_h \in C_h$ è selezionato come elemento rappresentativo scegliendo il sottografo del cluster con l'affinità media più alta rispetto agli altri sottografi del cluster:

$$r_h = \arg \max_{s \in C_h} \sum_{s_i \in C_h \setminus \{s\}} \phi(s, s_i)$$

Tutti i sottografi di ogni cluster C_h sono allineati³ a r_h . Poiché l'operazione di allineamento della rappresentazione matriciale di due grafi richiede che le due matrici abbiano le stesse dimensioni, è necessario estendere le dimensioni, aggiungendo dei nodi fittizi a tutti i sottografi, a quelle del sottografo più grande in tutto l'insieme \mathcal{S} , chiamato s_{max} .

Le permutazioni si propagano su \mathbf{I}_{conn} . Pertanto, dopo questo allineamento, abbiamo per ogni grafo $G \in \mathcal{G}$, una matrice $\dot{\mathbf{I}}_{conn}(G)$. Ad esempio, se G ha tre sottografi, s_a , s_b , e s_c ,

$$\dot{\mathbf{I}}_{conn}(G) = \left[\begin{array}{c|c|c} \dot{\mathbf{G}}_{s_a} & \dot{\mathbf{T}}_{s_a, s_b} & \dot{\mathbf{T}}_{s_a, s_c} \\ \hline 0 & \dot{\mathbf{G}}_{s_b} & \dot{\mathbf{T}}_{s_b, s_c} \\ \hline 0 & 0 & \dot{\mathbf{G}}_{s_c} \end{array} \right]$$

dove $\dot{\mathbf{G}}_{s_a}$ è simile a \mathbf{G}_{s_a} ma le sue righe (e colonne) sono ora invertite in accordo alle permutazioni per allineare i sottografi e la sua dimensione è ora uguale al numero di nodi del sottografo più grande di \mathcal{S} .

4.5 Analizzare le connessioni tra sottografi appartenenti a cluster diversi

L'informazione memorizzata in $\dot{\mathbf{I}}_{conn}$ può essere utilizzata per rilevare, data una coppia di cluster (C_h, C_k) se e in che misura per ogni grafo iniziale $G_\ell \in \mathcal{G}$ esistono due sottografi $s_i, s_j \in S_\ell$ per cui $\psi(s_i) = C_h$ e $\psi(s_j) = C_k$.

In particolare, ci riferiamo con l'espressione *connessioni tra cluster* all'insieme di tutte le connessioni tra le coppie di sottografi che soddisfano le

³Si noti che a differenza all'allineamento eseguito tra ogni coppia di sottografi (s_i, s_j) necessario per calcolare $a_{i,j}$ prima del clustering (Sezione 4.4.1), in questo caso si intende che per ogni sottografo di C_h viene effettivamente applicata una permutazione rispetto a r_h .

condizioni sopra riportate.

Mostriamo ora il nostro approccio su come memorizzare le connessioni tra cluster affinché siano utilizzabili in fase di campionamento.

4.5.1 Estrarre il grado di connettività tra le coppie sottografi

La prima operazione da effettuare per modellare la connettività tra cluster è scegliere un'unità elementare di connessione che indica il livello di dettaglio con cui vengono memorizzate le connessioni tra cluster. L'unità elementare di connessione utilizzata nel nostro approccio è il *grado di connettività* tra i sottografi.

Per grado di connettività c_{s_i, s_j} tra due sottografi s_i e s_j appartenenti a un grafo G intendiamo il numero di connessioni tra essi. Il grado di connettività è un valore che quantifica il legame tra due sottografi in termini di connessioni.

L'utilizzo di un modello generativo fa sì che vi sia una relazione tra il grado di connettività tra due sottografi dei dati originali e il grado di connettività tra due sottografi campionati, dal momento che entrambi condividono la stessa distribuzione. Se ad esempio s_i e s_j sono due sottografi ricorrenti in grafi diversi e in molti casi si rileva che sono legati da un grado di connettività alto, è presumibile che un grafo generato che sia al suo interno contenga s_i e s_j abbia a sua volta un grado di connettività elevato tra questi sottografi.

Il grado di connettività c_{s_i, s_j} può essere ricavato da $\mathbf{I}_{conn}(G)$ dalla relazione $c_{s_i, s_j} = \sum_{x=1}^{|s_i|} \sum_{y=1}^{|s_j|} \mathbf{T}_{s_i, s_j}(x, y)$.

È possibile considerare il grado di connettività come una forma di riduzione delle connessioni (lati) tra i nodi di un grafo alle connessioni tra sottografi.

4.5.2 Memorizzare le connessioni tra cluster

Per ogni coppia di cluster (C_h, C_k) , la connessione tra cluster rispetto al grafo $G_\ell \in \mathcal{G}$, $\mathbf{c}_{C_h, C_k}(G_\ell)$, è un vettore i cui elementi sono i gradi di connettività c_{s_i, s_j} relativi a tutte le coppie di sottografi $(s_i, s_j) \in \mathcal{S}$ per cui $\psi(s_i) = C_h$ e $\psi(s_j) = C_k$. Nel caso in cui fosse vuoto, si pone $\mathbf{c}_{C_h, C_k}(G_\ell) = \mathbf{0}$. Le connessioni tra cluster relative a tutti i grafi formano un array a blocchi \mathbf{c}_{C_h, C_k} . Ad esempio, se \mathcal{G} ha tre grafi, G_1 , G_2 , e G_3

$$\mathbf{c}_{C_h, C_k} = \left[\mathbf{c}_{C_h, C_k}(G_1) \mid \mathbf{c}_{C_h, C_k}(G_2) \mid \mathbf{c}_{C_h, C_k}(G_3) \right]$$

Indichiamo con \mathbf{c} l'insieme di tutti i vettori \mathbf{c}_{C_h, C_k} . Si noti che, essendo ciascuno di questi vettori di lunghezza variabile, \mathbf{c} non è una matrice.

Il modo in cui modelliamo le connessioni tra cluster si discosta da quella di [71], in cui gli autori mantengono un approccio a grana fine.

In [71], gli elementi delle matrici $\dot{\mathbf{T}}$ all'interno della matrice a blocchi $\dot{\mathbf{I}}_{conn}$ sono utilizzati per creare un vettore a blocchi \mathbf{v}_ℓ per ogni grafo iniziale G_ℓ . Ogni blocco di \mathbf{v}_ℓ è un vettore che rappresenta le connessioni tra una coppia di cluster (C_h, C_k) . In particolare, il blocco $\mathbf{v}_\ell(C_h, C_k)$ è la media delle vettorizzazioni di tutte le matrici di connessione $\dot{\mathbf{T}}_{s_i, s_j}$ tra due sottografi $s_i, s_j \in S_\ell$ tali che $\psi(s_i) = C_h$ e $\psi(s_j) = C_k$.

I vettori $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$ sono approssimati a una distribuzione normale multivariata $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ che modella la probabilità di avere una connessione tra i nodi dei sottografi.

Rispetto a [71], il nostro metodo modella le connessioni tra cluster in termini di grado di connettività tra sottografi: questo da un lato implica una perdita di dettaglio perché non si considera l'informazione su quali lati connettono due sottografi. D'altra parte, se il metodo di [71] utilizza esclusivamente grafi non pesati, il nostro metodo può essere applicato anche nel caso di grafi etichettati. Se ai nodi sono associate delle label è possibile calcolare la probabilità che due nodi siano collegati in funzione delle label. Questo aspetto, mancante in [71], può essere sfruttato per modellare le connessioni tra i nodi dei sottografi.

La conoscenza delle connessioni tra cluster serve per effettuare la riconnessione dei sottografi campionati durante la fase di campionamento. Questo aspetto è discusso nella Sezione 4.8.

4.6 Generare la nuova configurazione dei cluster

Da questo momento in poi inizia la fase di campionamento del nostro approccio, cioè quella che porta ad ottenere una nuova istanza \hat{G} di un grafo attraverso un processo di campionamento. Se i passi della costruzione del modello generativo iniziano dalla segmentazione dei grafi di input verso un livello più alto di astrazione con la creazione dei cluster, il processo che conduce al campionamento di nuovi grafi è simile ma viene eseguito in ordine

inverso, dai cluster ai grafi generati.

Il nostro approccio mantiene la suddivisione in fasi del campionamento presentata da [73], proponendosi invece di verificare l'efficacia e l'applicabilità dei modelli proposti dagli autori nell'ambito del task specifico, modificando eventualmente i tipi di distribuzione utilizzati nelle specifiche fasi.

Il primo passo è decidere quanti sottografi per ogni cluster C saranno presenti in un grafo campionato \hat{G} . Chiamiamo questa informazione *configurazione dei cluster*.

4.6.1 Significato della configurazione dei cluster

La configurazione dei cluster di un grafo G è una rappresentazione astratta della struttura di G . Al termine del clustering, infatti, ogni grafo $G_\ell \in \mathcal{G}$ può essere descritto sommariamente come costituito da un certo numero di sottografi per ogni cluster.

Se il processo di clustering è stato efficace, ogni cluster identifica una tipologia di sottografo con determinate caratteristiche. Nel caso dei grafi pesati con label la classificazione dei cluster tiene conto non solo di aspetti topologici del grafo ma anche semantici. Mantenendo il parallelismo tra cluster e categorie di sottografi, la configurazione dei cluster di un grafo mostra implicitamente due tipi di informazione:

- quali categorie di sottografi sono presenti nel grafo,
- in che misura ogni categoria di sottografi è ricorrente nel grafo.

Nel caso di applicazione, due sottografi all'interno dello stesso cluster non rappresentano solamente aree di un edificio in cui le porte tra le stanze sono disposte in modo simile, ma anche aree in cui le tipologie di stanze incontrate sono di un certo tipo. Se allora un cluster astrae entrambi i tipi di informazione, è possibile affermare che ogni cluster classifica un certo tipo di area geometrico-funzionale dell'edificio. Come ogni tipo di stanza assolve a un determinato compito nei confronti di chi la utilizza, un insieme di un certo tipo di stanze collegate tra loro in un certo modo può indicare una parte di un edificio che nel complesso viene utilizzata per svolgere un insieme di compiti ben preciso. Il fatto che le stanze di queste aree siano molto connesse tra loro ma poco connesse alle stanze di altre aree dell'edificio (che è il principio alla base di Ncut), suggerisce la possibilità che la struttura topologica e quella semantica si influenzino a vicenda a questo livello di astrazione.

Per questo motivo abbiamo ritenuto legittimo un approccio che utilizzasse il sottografo come componente base della generazione di nuovi grafi ed è per questo che la configurazione di cluster rappresenta l'ossatura di ogni grafo. Per i grafi di input viene dedotta al termine della fase di costruzione del modello, per i grafi generati è la prima informazione in base a cui si costruisce l'intero grafo.

4.6.2 Rappresentazione della configurazione dei cluster e campionamento di Gibbs

La configurazione di cluster è rappresentata come un vettore binario \mathbf{v} . Ogni bit v_h di indice h di \mathbf{v} è o 1 o 0 e mostra se un sottografo appartenente a un cluster è presente o no, rispettivamente.

Dal momento che ogni grafo originale può essere costituito da un numero diverso di sottografi per ogni cluster, è necessario dedicare un numero di bit sufficiente per memorizzare la configurazione per ogni cluster. Per ogni cluster C , sono utilizzati m bit in v per rappresentare la configurazione di cluster per C , dove m è il numero massimo di sottografi appartenenti a C che compaiono in un qualche grafo originale. Ad esempio in Figura 4.4 è mostrato un esempio in cui vi sono tre grafi e quattro cluster. I cluster 1, 2, 3 sono presenti sempre una volta in tutti i grafi, mentre il cluster 4 è presente zero volte nel primo grafo, una volta nel secondo grafo, e due volte nel terzo.

Cluster:	1	2	3	4	4
Zero subgraphs from cluster 4:	1	1	1	0	0
One subgraphs from cluster 4:	1	1	1	1	0
Two subgraphs from cluster 4:	1	1	1	1	1

Figura 4.4: Esempio di configurazioni di cluster con tre grafi e quattro cluster, tratto da [73].

Per generare la configurazione di cluster del grafo \hat{G} , utilizziamo un *campionamento di Gibbs* [22].

Un campionario di Gibbs permette di ottenere un campione di un insieme di variabili aleatorie mediante il campionamento di singole variabili, a turno.

Sia $[x_1^{(0)}, \dots, x_j^{(0)}]$ una inizializzazione arbitraria di variabili aleatorie, inizialmente viene campionato $x_1^{(1)}$ dalla distribuzione condizionata $p(x_1^{(1)} | x_2^{(0)}, \dots, x_j^{(0)})$, successivamente viene campionato $x_2^{(1)}$ dalla distribuzione condizionata $p(x_2^{(1)} | x_1^{(1)}, x_3^{(0)}, \dots, x_j^{(0)})$ e così via finché non si campiona $x_j^{(1)}$ da $p(x_j^{(1)} | x_1^{(1)}, \dots, x_{j-1}^{(1)})$.

Questo processo viene ripetuto per r iterazioni, ottenendo $[x_1^{(r)}, \dots, x_j^{(r)}]$. Se r è abbastanza grande, allora $[x_1^{(r)}, \dots, x_j^{(r)}]$ può essere considerato un campione dalla distribuzione di probabilità congiunta $p(x_1, x_2, \dots, x_j)$, perché la transizione tra stati del sistema è stazionaria.

Nel nostro caso, qualsiasi configurazione del vettore \mathbf{v} è uno stato possibile del campionatore di Gibbs, inizializzato casualmente a \mathbf{v}^0 . Quando il campionatore è in uno stato \mathbf{v}^σ , una transizione verso il nuovo stato $\mathbf{v}^\sigma \rightarrow \mathbf{v}^{\sigma+1}$ è eseguita selezionando, per ogni bit di \mathbf{v}^σ , in sequenza, se deve essere invertito, in accordo a una probabilità calcolata come nell'Algoritmo 3. Dopo un numero di iterazioni fisso $r = I_2$ (comprese tra 150 e 200 nei nostri esperimenti, dando una buona possibilità che lo spazio degli stati sia stato sufficientemente coperto), il campionatore di Gibbs si ferma, con la configurazione di cluster desiderata \hat{v} .

4.7 Selezionare i sottografi che rappresentano la configurazione dei cluster generata

Una volta che la configurazione dei cluster \hat{v} è stata generata attraverso il campionamento di Gibbs, vengono selezionati i sottografi in accordo ad essa. Per ogni valore pari a 1 in \hat{v} relativo al cluster C_h è selezionato un sottografo \hat{s}_i tale che $\psi(\hat{s}_i) = C_h$.

Questo significa dover estrarre un sottografo dal sottoinsieme dei sottografi di partenza

$$\mathcal{S}_{C_h} = \{\forall S \in \mathcal{S} | \psi(S) = C_h\}$$

L'approccio utilizzato, piuttosto intuitivo, consiste nel campionare un sottografo S da \mathcal{S}_{C_h} secondo una distribuzione uniforme. In questo modo la probabilità di selezione è proporzionale alla frequenza dei sottografi nel cluster. Un sottografo s_x che compare tre volte in C_h avrà ad esempio il triplo di probabilità di essere estratto rispetto a un sottografo s_y che compare una volta sola.

Al termine di questa fase si ha l'insieme \hat{S} dei sottografi campionati.

Input: La configurazione di cluster q per ogni grafo $G \in \mathcal{G}$, dove q è un vettore binario con un bit per ogni cluster C (1 o 0 se in G è presente o meno un sottografo del cluster C), un parametro $\vartheta = 4$, un parametro I_2

Output: Una configurazione di cluster campionata rappresentata come un vettore binario \mathbf{v}

- 1 inizializzare $\mathbf{v}^{\sigma=0}$ in modo casuale
- 2 **repeat**
- 3 $\mathbf{v}_a \leftarrow \mathbf{v}^\sigma$
- 4 **foreach** *bit di indice* $h = 1, 2, \dots, |\mathbf{v}|$ **do**
- 5 computare $\mathbf{v}_{!a}$ invertendo il h -esimo bit di \mathbf{v}_a
- 6 usando la distanza di Hamming $H(v, q)$ tra due vettori binari
 computare

$$H_{\mathbf{v}_a} = \sum_{\ell=1}^{|\mathcal{G}|} e^{\vartheta \cdot H(\mathbf{v}_a, q_\ell)}$$

$$H_{\mathbf{v}_{!a}} = \sum_{\ell=1}^{|\mathcal{G}|} e^{\vartheta \cdot H(\mathbf{v}_{!a}, q_\ell)}$$
- 7
$$P_{\mathbf{v}_a} = \frac{H_{\mathbf{v}_a}}{H_{\mathbf{v}_a} + H_{\mathbf{v}_{!a}}}$$

$$P_{\mathbf{v}_{!a}} = \frac{H_{\mathbf{v}_{!a}}}{H_{\mathbf{v}_a} + H_{\mathbf{v}_{!a}}}$$
- 8 selezionare $\mathbf{v}_a \leftarrow \mathbf{v}_a$ o $\mathbf{v}_a \leftarrow \mathbf{v}_{!a}$ con probabilità $P_{\mathbf{v}_a}$ e $P_{\mathbf{v}_{!a}}$,
 rispettivamente
- end**
- 9 $\mathbf{v}^{\sigma+1} \leftarrow \mathbf{v}_a$
- until** # di iterazioni $\sigma > I_2$
- 10 **return** $\hat{\mathbf{v}} \leftarrow \mathbf{v}^{I_2}$

Algoritmo 3: Campionamento di Gibbs, da [73].

4.8 Connettere insieme i sottografi campionati

L'ultima fase importante del processo generativo richiede di connettere insieme i sottografi campionati \hat{S} per formare il grafo finale \hat{G} . La riconnessione tra i sottografi non deve avvenire in modo casuale, ma deve essere effettuata seguendo un modello basato sui dati di input.

Il nostro approccio suddivide la riconnessione tra due sottografi \hat{s}_i e \hat{s}_j in due sottofasi distinte:

- inizialmente si stabilisce il grado di connettività $c_{\hat{s}_i, \hat{s}_j}$ tra i due sottografi, ovvero il numero di lati che connette \hat{s}_i e \hat{s}_j ;

- successivamente, noto $c_{\hat{s}_i, \hat{s}_j}$, si determina quali nodi devono essere connessi.

Come illustrato nella Sezione 4.5, il nostro metodo si distingue da quello di [71] in quanto modelliamo le connessioni tra cluster come gradi di connettività e non come singole connessioni tra nodi. Di conseguenza, se in [71] il processo di riconnessione dei sottografi avviene in un'unica fase di campionamento al termine della quale sono definite le connessioni, nel nostro caso invece è richiesto un passaggio ulteriore.

Questa suddivisione del processo di riconnessione si riflette nell'utilizzo di due distribuzioni distinte sui dati, che illustriamo in seguito.

4.8.1 Determinare i gradi di connettività tra i sottografi

Il grado di connettività \hat{c} tra tutte le coppie di sottografi del grado finale viene assegnato considerando ogni connessione tra una coppia di cluster come una variabile aleatoria $\mathbf{c}_{i,j}$ e approssimando $\mathbf{c} = \{\mathbf{c}_{1,1}, \mathbf{c}_{1,2}, \dots, \mathbf{c}_{n,n}\}$ a una distribuzione normale multivariata $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

Questa approssimazione richiede di ricavare innanzitutto, dalle variabili aleatorie che rappresentano le connessioni tra cluster, il vettore della media $\boldsymbol{\mu}$ e la matrice della covarianza $\boldsymbol{\Sigma}$, sui quali poi campionare la distribuzione normale, similmente a quanto fatto in [71, 73].

Rispetto a [71, 73] il nostro approccio si differenzia per due motivi: il primo è che distinguendo la generazione delle connessioni in due fasi modelliamo con questa distribuzione i gradi di connessione tra sottografi e non le singole connessioni tra i nodi. Il secondo riguarda le modalità attraverso cui si ricavano $\boldsymbol{\mu}$ e $\boldsymbol{\Sigma}$.

Gli autori di [71, 73] creano un vettore \mathbf{v}_ℓ per ogni grafo iniziale G_ℓ , a sua volta formato da sezioni della stessa lunghezza che memorizzano le connessioni tra ogni coppia di cluster C_h, C_k di \mathcal{C} . Così facendo $\boldsymbol{\mu}$ e $\boldsymbol{\Sigma}$ sono calcolate direttamente sulla matrice \mathbf{V} le cui righe sono i vettori \mathbf{v} così ottenuti.

Il limite principale di questo approccio è legato al fatto che \mathbf{V} è costruito in modo tale che, se n è il numero di nodi in un sottografo e c è il numero di cluster, ogni campione \mathbf{v}_ℓ contiene $n^2 \lceil \frac{c(c+1)}{2} \rceil$ variabili aleatorie. Il numero di campioni invece è dato dal numero di grafi. L'approssimazione di \mathbf{V} a una normale multivariata sarebbe giustificata se il numero di campioni fosse almeno pari al numero di variabili. Per ovviare all'impossibilità di avere un

numero così significativo di campioni, l'utilizzo del grado di connettività al posto della singola connessione permette di eliminare il termine quadratico nel numero di nodi.

In secondo luogo, l'approccio di [71, 73] non tiene conto del numero di coppie di sottografi s_i, s_j per le quali $s_i = \psi(C_h)$ e $s_j = \psi(C_k)$, dovuto al fatto che ogni sezione di \mathbf{v}_ℓ ha la stessa lunghezza.

Il nostro approccio invece memorizza le connessioni tra cluster tra ogni coppia di cluster (C_h, C_k) in vettori di lunghezza variabile \mathbf{c}_{C_h, C_k} (vedi Sezione 4.5).

Ogni vettore rappresentante la connessione tra cluster contiene infatti un numero di componenti date dal numero di coppie di sottografi $(s_i, s_j) \in \mathcal{S}$ per cui $\psi(s_i) = C_h$ e $\psi(s_j) = C_k$. Il nostro approccio si distingue da quello di [71, 73] in quanto non riduciamo la connessione tra cluster a un singolo valore calcolando ad esempio la media tra tutte le connessioni tra sottografi appartenenti a quella coppia di cluster. Nel caso di grafi non sufficientemente simili, l'utilizzo della media si è dimostrato un'assunzione troppo semplificante. Pertanto il nostro approccio, una volta filtrate le coppie di cluster che non risultano connesse in alcun grafo, calcola i singoli valori di media di ogni connessione tra cluster \mathbf{c}_{C_h, C_k} e di covarianza tra una coppia di connessioni tra cluster \mathbf{c}_{C_h, C_k} e \mathbf{c}_{C_l, C_m} e successivamente costruisce $\boldsymbol{\mu}$ e $\boldsymbol{\Sigma}$ utilizzando tutti questi valori. Il metodo proposto non presenta perdita di generalità rispetto a quello canonico, ma ha i seguenti vantaggi:

- memorizza le connessioni tra cluster in modo che l'approssimazione a normale multivariata sia più legittima rispetto a [71, 73];
- impedisce che cluster mai connessi tra loro possano risultare connessi in seguito al campionamento;
- tiene traccia di tutte le coppie di sottografi appartenenti a una certa coppia di cluster.

In seguito è descritto in maniera più approfondita questo procedimento.

Per ogni connessione tra cluster si calcola il valor medio μ_{C_h, C_k} . L'insieme di questi valori forma il vettore delle medie delle connessioni $\boldsymbol{\mu}$. Ad esempio, se vi sono tre cluster C_1, C_2 e C_3

$$\boldsymbol{\mu} = \left[\begin{array}{cccccc} \mu_{C_1, C_1} & \mu_{C_1, C_2} & \mu_{C_1, C_3} & \mu_{C_2, C_2} & \mu_{C_2, C_3} & \mu_{C_3, C_3} \end{array} \right]$$

Il calcolo della covarianza empirica viene effettuato attraverso un approccio ad-hoc: date due connessioni tra cluster $\mathbf{c}_1 = \mathbf{c}_{C_h, C_k}$ e $\mathbf{c}_2 = \mathbf{c}_{C_l, C_m}$ la covarianza $\Sigma_{\mathbf{c}_1, \mathbf{c}_2}$ si calcola, come spiegato esaustivamente nell'Algoritmo 4, effettuando l'estensione di \mathbf{c}_1 e \mathbf{c}_2 a due vettori $\dot{\mathbf{c}}_1$ $\dot{\mathbf{c}}_2$ aventi stesse dimensioni.

In particolare, i vettori $\mathbf{c}_1(G)$ e $\mathbf{c}_2(G)$, contenenti i gradi di connettività delle rispettive coppie di cluster relativi al grafo G , sono estesi a $\dot{\mathbf{c}}_1(G)$ e $\dot{\mathbf{c}}_2(G)$ in modo che ogni coppia di gradi di connettività $c_{1,h}(G)$ e $c_{2,h}(G)$ contenga una fra tutte le combinazioni tra un elemento $c_{1,i}(G)$ e un elemento $c_{2,j}(G)$. Questo processo viene iterato su tutti i grafi.

In questo modo è possibile calcolare la covarianza tra i vettori $\dot{\mathbf{c}}_1$ $\dot{\mathbf{c}}_2$.

Se ad esempio vi sono tre grafi e le connessioni tra cluster relative alle coppie \mathbf{c}_1 e \mathbf{c}_2 sono:

$$\mathbf{c}_1 = \left[1, 0, 1 \mid 1, 2 \mid 0, 1, 0 \right]$$

$$\mathbf{c}_2 = \left[1, 1 \mid 1, 1, 0 \mid 2, 0, 1, 0 \right]$$

i vettori tra i quali si calcola la covarianza sono:

$$\dot{\mathbf{c}}_1 = \left[1, 0, 1, 1, 0, 1 \mid 1, 2, 1, 2, 1, 2 \mid 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0 \right]$$

$$\dot{\mathbf{c}}_2 = \left[1, 1, 1, 1, 1, 1 \mid 1, 1, 1, 1, 0, 0 \mid 2, 2, 2, 0, 0, 0, 1, 1, 1, 0, 0, 0 \right]$$

Input: Le connessioni tra cluster $\mathbf{c}_1, \mathbf{c}_2$

Output: Il valore della covarianza $\Sigma_{\mathbf{c}_1, \mathbf{c}_2}$

```

1 for  $G \in \mathcal{G}$  do
2    $l_1 = |\mathbf{c}_1(G)|$ 
3    $l_2 = |\mathbf{c}_2(G)|$ 
4   foreach indice  $h = 1, 2, \dots, l_1 \cdot l_2$  do
5     inserire nelle posizioni di indice  $h$  di due nuovi vettori estesi
        $\dot{\mathbf{c}}_1(G)$  e  $\dot{\mathbf{c}}_2(G)$  di dimensione  $l_1 \cdot l_2$  abbinando in tutti i modi
       possibili una componente di  $\mathbf{c}_1(G)$  e una componente di  $\mathbf{c}_2(G)$ 
6     end
7   ottenere  $\dot{\mathbf{c}}_1(G)$  e  $\dot{\mathbf{c}}_2(G)$  e inserirli in coda a  $\dot{\mathbf{c}}_1$  e  $\dot{\mathbf{c}}_2$ 
8 end
9 calcolare la covarianza  $\Sigma_{\mathbf{c}_1, \mathbf{c}_2}$  tra i vettori  $\dot{\mathbf{c}}_1$  e  $\dot{\mathbf{c}}_2$ 
10 return  $\Sigma_{\mathbf{c}_1, \mathbf{c}_2}$ 

```

Algoritmo 4: Covarianza tra due connessioni tra cluster.

L'insieme delle singole covarianze tra coppie di connessioni tra cluster forma in modo analogo una matrice quadrata Σ di dimensioni $\frac{|C|^2+|C|}{2}$.

Il campionamento di nuovi sample dalla distribuzione normale multivariata viene fatto in questo modo: la matrice di covarianza Σ viene decomposta in matrice diagonale degli autovalori Λ e matrice degli autovettori Φ . A questo punto viene generato un vettore $\hat{\mathbf{c}}$ contenente le connessioni tra tutte le coppie di cluster.

$$\hat{\mathbf{c}} = \boldsymbol{\mu} + \Phi \mathbf{b}$$

dove \mathbf{b} è un parametro vettoriale creato campionando dalla distribuzione normale di media zero e varianza data dagli autovalori di Λ , cioè $\mathbf{b} \sim \mathcal{N}(\mathbf{0}, \Lambda)$.

Si può verificare che in questo modo il campionamento segue la distribuzione normale multivariata dalle seguenti relazioni:

$$E[\hat{\mathbf{c}}] = E[\boldsymbol{\mu} + \Phi \mathbf{b}] = E[\boldsymbol{\mu}] + E[\Phi \mathbf{b}] = \boldsymbol{\mu} + \Phi E[\mathbf{b}] = \boldsymbol{\mu}$$

$$\begin{aligned} Cov[\hat{\mathbf{c}}] &= Cov[\boldsymbol{\mu} + \Phi \mathbf{b}] = Cov[\boldsymbol{\mu}] + Cov[\Phi \mathbf{b}] = \\ &= \Phi Cov[\mathbf{b}] \Phi^T = \Phi \Lambda \Phi^T = \Sigma \end{aligned}$$

Ogni componente di $\hat{\mathbf{c}}$ rappresenta il grado di connessione c_{C_h, C_k} tra due sottografi \hat{s}_i, \hat{s}_j appartenenti rispettivamente ai cluster C_h e C_k . Questo significa che un singolo vettore $\hat{\mathbf{c}}$ può generare solo il grado di connessione di una coppia di sottografi per coppia di cluster. E' quindi necessario campionare $\hat{\mathbf{c}}$ k volte, dove k è il numero di massimo di coppie di sottografi (\hat{s}_i, \hat{s}_j) appartenenti alla stessa coppia di cluster.

Infine il vettore $\hat{\mathbf{c}}$ a valori reali deve essere discretizzato: il risultato del campionamento è infatti un vettore a valori continui, mentre il grado di connessione assume valori interi. In questo caso il mapping da intervalli continui a valori discreti è effettuato in modo che la distribuzione dei gradi di connettività tra i sottografi del grafo generato segua quella tra i sottografi dei grafi di input. In base a questo si ricavano empiricamente dei parametri di thresholding $(\beta_1, \beta_2, \dots, \beta_n)$ tali che $p(c_i = x) = p(\beta_x \leq \hat{c}_i < \beta_{x+1})$.

4.8.2 Determinare i collegamenti tra i nodi dei sottografi

L'ultimo passaggio della fase di riconnessione stabilisce per ogni connessione che deve essere stabilita tra due sottografi \hat{s}_i e \hat{s}_j quali nodi $n_h \in \hat{s}_i$ e $n_k \in \hat{s}_j$ siano collegati.

Il nostro approccio utilizza una rete Bayesiana (mostrata in Figura 4.5) che modella la probabilità di una connessione $p(C_{n_h, n_k})$ sfruttando le informazioni semantiche contenute nelle label, e specifiche del nostro contesto applicativo. Abbiamo considerato alcune relazioni significative tra label e connessioni.

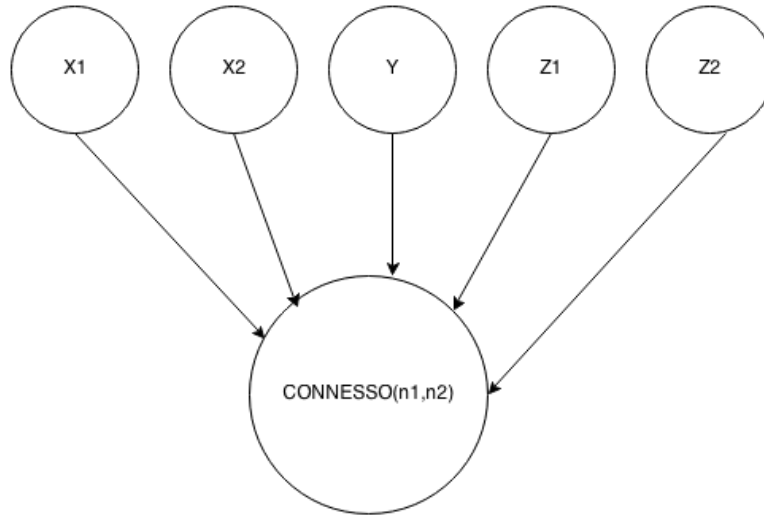


Figura 4.5: Schema della rete Bayesiana per la riconnessione dei nodi di due sottografi. Il significato dei termini è spiegato nel testo.

La rete Bayesiana da noi proposta, abbastanza intuitiva, considera la probabilità $p(C_{n_h, n_k})$ che due nodi n_h n_k siano connessi come prodotto di probabilità condizionate assunte indipendenti:⁴

$$p(C_{n_h, n_k}) = p(C_{n_h, n_k} | X_1) p(C_{n_h, n_k} | X_2) p(C_{n_h, n_k} | Y) p(C_{n_h, n_k} | Z_1) p(C_{n_h, n_k} | Z_2)$$

dove:

- $p(X_1)$ è la probabilità che un nodo di label $l_h = f(n_h)$ abbia un numero di lati incidenti pari al numero di lati incidenti in n_h (considerando anche la connessione con n_k); X_1 rappresenta l'evento "un nodo di label l_h è connesso a $d(n_h)$ nodi"; $p(X_2)$ e X_2 sono definiti in modo analogo,
- $p(Y)$ è la probabilità di trovare una connessione nei sottografi dei dati originali memorizzate in \mathbf{I}_{conn} tra un nodo n_i^0 di label $f(n_i^0) = f(n_h)$

⁴Si sta ancora considerando che $n_h \in \hat{s}_i$ e $n_k \in \hat{s}_j$ anche se per semplicità la notazione lo omette.

e un nodo n_j^0 di label $f(n_j^0) = f(n_k)$; Y rappresenta l'evento "due sottografi sono connessi tra un nodo di label l_h e un nodo di label l_k ",

- $p(Z_1)$ è la probabilità di trovare nei grafi originali in \mathcal{G} un cammino tra un nodo n_g di una certa label $l_g = f(n_g)$, n_i^0 di label $f(n_i^0) = f(n_h)$ e un nodo n_j^0 di label $f(n_j^0) = f(n_k)$, dove n_g è un nodo di \hat{s}_i collegato a n_h ; nel caso in cui n_h sia collegato direttamente a più nodi $n_{g_i} \in \hat{s}_i$, $P(Z_1)$ è calcolata come la probabilità media fra tutti i cammini di tre nodi in cui si mantengono fissi n_h e n_k (e le loro label) e si fa variare n_{g_i} considerando ogni volta la label $l_{g_i} = f_{g_i}$; $P(Z_2)$ è ottenuta in modo analogo mantenendo fissi n_h e n_k e considerando tutti i nodi $n_f \in \hat{s}_j$ collegati direttamente a n_k ; Z_1 rappresenta l'evento "un nodo di label l_h è connesso ad almeno un nodo di label l_k e ad almeno un nodo di label n_g "; Z_2 rappresenta l'evento analogo.

Possiamo interpretare, a un livello intuitivo, questo modello rispetto all'ambito specifico (cioè considerando grafi che rappresentano edifici) nel modo seguente:

- $p(X_1)$ e $p(X_2)$ considerano quanto è frequente che una stanza abbia un certo numero di porte; ad esempio, un corridoio è solitamente collegato a molte stanze, mentre una stanza di servizio ha in genere una sola porta;
- se assumiamo il parallelismo tra un sottografo e un'ala di un edificio, $p(Y)$ considera quali sono le stanze che collegano due ali di un edificio. Ad esempio, è frequente che in un edificio pubblico le ali di un edificio siano collegate da due corridoi o da un atrio e un corridoio;
- $p(Z_1)$ e $p(Z_2)$ forniscono l'informazione sulla frequenza dei pattern di due stanze collegate a una terza; in un'abitazione, ad esempio, è verosimile trovare un corridoio collegato sia a una camera da letto sia a un bagno.

La Figura 4.6 serve a chiarire il significato del modello Bayesiano. Immaginiamo di voler calcolare $p(C_{n_C, n_C})$ tra i due nodi di label C di \hat{s}_1 e \hat{s}_2 . Secondo il nostro modello la probabilità che i due sottografi siano riconnessi attraverso questa coppia di nodi è pari al prodotto di probabilità semplici calcolate come segue:

- $p(X_1)$ è la probabilità che un nodo di label C sia connesso a 6 nodi, cioè che una stanza di tipo 'corridoio' abbia 6 porte. Allo stesso modo $p(X_2)$ è la probabilità che un nodo di label C sia connesso a 4 nodi;

- $p(Y)$ è la probabilità di trovare tra tutte le connessioni tra i sottografi contenute in $\hat{\mathbf{I}}_{conn}$ una connessione tra due nodi di label C ;
- $p(Z_1)$ si calcola considerando tutti i cammini su tre nodi di cui due sono i nodi della connessione $C-C$ e il terzo è un nodo di \hat{s}_1 . Poiché il nodo C di \hat{s}_1 è connesso a 6 nodi M , $p(Z_1)$ è la probabilità media di sei probabilità (in questo caso uguali) di trovare nell'insieme dei grafi di input \mathcal{G} un cammino di tipo $M-C-C$. Allo stesso modo $p(Z_2)$ è la probabilità di trovare in \mathcal{G} un cammino di tipo $S-C-C$.

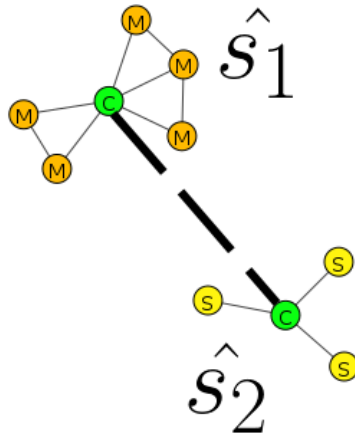


Figura 4.6: Due sottografi campionati \hat{s}_1 e \hat{s}_2 . La linea tratteggiata tra i due nodi di label 'C' indica una connessione su cui valutare $p(C_{n_C, n_C})$.

Le probabilità di connessione vengono calcolate per tutte le possibili coppie di nodi di \hat{s}_i e \hat{s}_j , e sono ordinate per valori decrescenti. I nodi con il valore della probabilità calcolata più alto vengono quindi selezionati per effettuare un numero di connessioni pari al grado di connessione $c_{\hat{s}_i, \hat{s}_j}$.

Rispetto all'approccio di [71, 73], il nostro metodo sfrutta le label per effettuare le connessioni tra nodi dei sottografi, in accordo al campionamento mostrato precedentemente sui gradi di connettività. In [71, 73], il campionamento da una distribuzione normale multivariata, stabilisce in un unico passo come i sottografi saranno riconnessi fra loro. L'aspetto più problematico è legato al fatto che questo tipo di approccio può funzionare adeguatamente solo se tutte le coppie di sottografi appartenenti a una certa coppia di cluster

siano connessi in modo simile, non solo per numero di connessioni, ma anche per i nodi attraverso cui avviene la connessione. Il nostro approccio cerca di disaccoppiare il problema della riconnessione dei sottografi, considerando separatamente due sottoproblemi: quello di determinare il modo in cui sono connessi i sottografi (legato ai cluster) e quello di connettere le coppie di sottografi tra loro (legato alla relazione che intercorre tra topologia di un sottografo e label).

4.9 Ricostruzione del grafo finale

L'insieme dei sottografi generati \hat{S} e delle connessioni tra essi è sufficiente a ricostruire il grafo finale \hat{G} . Il nuovo grafo \hat{G} è pertanto ottenuto connettendo vari sottografi che possono essere identificati nei grafi originali in \mathcal{G} . Sotto questo punto di vista il nostro approccio è simile a quello di [4], mentre se ne distanzia per il fatto che \hat{G} non appartiene a \mathcal{G} in generale. Da un lato l'utilizzo di sottografi provenienti da grafi di input è motivato dall'esigenza di generare grafi che corrispondano effettivamente a edifici di una certa tipologia. Allo stesso tempo, la flessibilità offerta dalle molteplici combinazioni dei sottografi e delle connessioni tra essi permette una certa varietà nei grafi generati.

La fase di campionamento può essere svolta più volte per generare un certo numero di grafi senza dover ricostruire il modello generativo, ma direttamente a partire dalla generazione della configurazione dei cluster descritta nella Sezione 4.6.

Capitolo 5

Il modulo per la generazione di nuovi grafi per il mapping semantico

Mia zia si muove sulla sedia. Io inizio ad avere difficoltà a respirare. I cavi dei fornelli, sfoderati e sfilacciati, sono sparsi sul partitore come capelli grigi. [...] Il lavoro che mi interessa si fa con una matita e un pezzo di carta. Qualche rara volta con la calcolatrice. All'avanguardia dell'ingegneria elettronica, quasi tutte le questioni interessanti si possono risolvere mediante la manipolazione di variabili. Non sono mai stato bocciato a un esame, neanche una volta. Mai. E a quanto pare ho rotto questa povera stronzissima cucina. Non so bene cosa fare.

David Foster Wallace, *Da una parte e dall'altra*

In questo capitolo viene descritto il modulo che permette, a partire da un insieme di grafi che rappresentano planimetrie di piani di interi edifici appartenenti a una tipologia, la generazione di nuovi grafi con cui ampliare la base di conoscenza di mappe semantiche per un robot mobile autonomo. Scopo finale del modulo da noi proposto è l'integrazione in un sistema per il mapping semantico. Offrire questo tipo di conoscenza al robot può permettere la predizione di parti inesplorate di un edificio sulla base dei nuovi dati generati.

La trattazione di questo capitolo tocca tutti gli aspetti sia tecnologici sia implementativi legati al lavoro svolto in questa tesi. In particolare, possiamo considerare la trattazione composta da tre parti principali:

- una prima parte in cui si mostrano le tecnologie scelte per realizzare il modulo, il formato dei dati di input e gli strumenti utilizzati per la visualizzazione dei grafi;
- una seconda parte in cui si mostra l'architettura del sistema, con particolare attenzione ai componenti interni al modulo di generazione dei grafi e al modo in cui interagiscono tra loro;
- una terza parte in cui si descrivono nel dettaglio alcune funzioni significative interne ai componenti.

5.1 Le tecnologie utilizzate

Possiamo individuare tre aspetti principali che riguardano la realizzazione pratica del modulo per la generazione di nuove mappe semantiche.

- Un primo aspetto riguarda il metodo attraverso cui ottenere dataset di grafi con label che rappresentino piani di edifici, l'input del modello generativo.
- Un secondo aspetto riguarda la scelta di un linguaggio con cui implementare il modello descritto nel Capitolo 4.
- Un terzo aspetto riguarda la realizzazione di uno strumento che permetta la visualizzazione dei grafi utilizzati all'interno del modulo.

Il primo e il terzo punto non riguardano direttamente l'implementazione del modello generativo su grafi, ma sono richiesti per il funzionamento complessivo del modulo.

5.1.1 Ottenere i grafi di input

L'unico tipo di dato di input richiesto per l'addestramento del modulo è un insieme di grafi etichettati, nel nostro caso rappresentanti i piani di edifici reali appartenenti a una certa tipologia. In questo lavoro non utilizziamo grafi di mappe semantiche ottenuti da un robot a seguito di esplorazioni, ma otteniamo semi-manualmente i grafi dalle planimetrie di alcuni edifici, che risulta una scelta molto più rapida se serve un insieme consistente di dati. Attualmente, un robot impiega giorni a ottenere il grafo topologico di alcune stanze, mentre un utente umano può ricavare l'intero grafo da un'immagine in pochi minuti.

La traduzione di una planimetria in un grafo semantico avviene attraverso l'ausilio di un tool grafico, realizzato in Processing [50] che permette di ottenere il grafo attraverso un procedimento molto semplice. L'utente sovrappone all'immagine della planimetria di un edificio il grafo selezionando da un menu i nodi etichettati corrispondenti alle tipologie di stanze da noi considerate e li posiziona nelle aree occupate dalle stanze corrispondenti nell'immagine. È poi possibile collegare i nodi seguendo le porte presenti nella planimetria. Un lato è creato selezionando due nodi consecutivamente.

In Figura 5.1 è mostrata una schermata del tool per la tipologia edilizia SCUOLE.

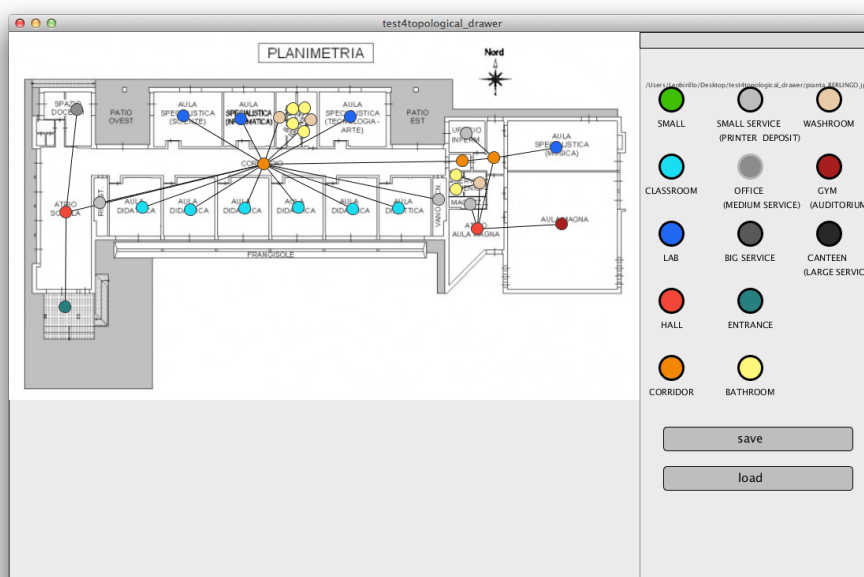


Figura 5.1: Schermata del tool per la creazione dei dataset. L'immagine della planimetria rappresenta la scuole secondaria di primo grado del Comune di Berlingo. Il grafo con le label ottenuto manualmente è sovrapposto all'immagine.

Il risultato è un file in formato XML contenente i nodi con le informazioni sulle label associate e sulle connessioni tra nodi. Il file contiene anche altre informazioni quali i colori delle label e le coordinate dei nodi; in Figura 5.2 ne viene fornito un esempio. Il file XML può essere facilmente convertito in un file testuale contenente la matrice del grafo semantico.

<pre> <node> <id>49</id> <label>OFFICE(MEDIUM SERVICE)</label> <type>R</type> <color b="64" g="64" r="64"/> <position x="222" y="142"/> </node> <node> <id>50</id> <label>OFFICE(MEDIUM SERVICE)</label> <type>R</type> <color b="64" g="64" r="64"/> <position x="263" y="165"/> </node> <node> <id>51</id> <label>SMALL SERVICE</label> <type>R</type> <color b="15" g="244" r="255"/> <position x="257" y="138"/> </node> <node> <id>52</id> <label>SMALL SERVICE</label> <type>R</type> <color b="15" g="244" r="255"/> <position x="242" y="150"/> </node> <node> <id>53</id> <label>SMALL SERVICE</label> <type>R</type> <color b="15" g="244" r="255"/> <position x="318" y="167"/> </node> <node> <id>54</id> <label>SMALL SERVICE</label> <type>R</type> <color b="15" g="244" r="255"/> <position x="318" y="146"/> </node> <node> <id>55</id> <label>HALL</label> <type>C</type> <color b="255" g="245" r="77"/> <position x="297" y="239"/> </node> <node> <id>56</id> <label>CLASSROOM</label> <type>R</type> <color b="0" g="187" r="255"/> <position x="150" y="172"/> </node> </pre>	<pre> <connection> <id>42</id> <id>44</id> </connection> <connection> <id>42</id> <id>50</id> </connection> <connection> <id>50</id> <id>51</id> </connection> <connection> <id>42</id> <id>53</id> </connection> <connection> <id>42</id> <id>54</id> </connection> <connection> <id>41</id> <id>45</id> </connection> <connection> <id>41</id> <id>46</id> </connection> <connection> <id>47</id> <id>41</id> </connection> <connection> <id>41</id> <id>41</id> <id>48</id> </connection> <connection> <id>41</id> <id>49</id> </connection> <connection> <id>49</id> <id>52</id> </connection> </pre>
--	---

(a) Nodi in XML

(b) Connessioni in XML

Figura 5.2: Un esempio di file XML contenente le informazioni di un grafo che rappresenta un edificio. In (a) è mostrato il formato per i nodi, mentre in (b) è mostrato il formato per le connessioni.

5.1.2 Realizzare il modello generativo in MATLAB

Pe l'implementazione del modello generativo sui grafi è stato utilizzato il linguaggio MATLAB. I motivi principali alla base della nostra scelta sono:

- la semplicità nella manipolazione di matrici, le strutture dati con cui rappresentiamo i grafi;
- la presenza di numerose di librerie e funzioni integrate.

In generale, qualunque ambiente di sviluppo dedicato al calcolo matematico-scientifico si presta all'implementazione del modello generativo. Non è necessario, ad esempio, utilizzare un linguaggio di programmazione object-oriented (per quanto MATLAB lo consenta) dal momento che gran parte dei dati utilizzati all'interno del modello sono esprimibili come insiemi di matrici. In MATLAB la scrittura del codice è facilitata grazie a una sintassi semplice e a numerose funzioni integrate per il calcolo matriciale.

5.1.3 Visualizzare i grafi

Avere a disposizione uno strumento per visualizzare i grafi può essere utile non solo per rappresentare graficamente grafi di input e grafi generati, ma anche per ricevere un feedback immediato degli step intermedi del sistema, quali divisione in sottografi, clustering e campionamento. Per fare questo è stato realizzato uno script in Python basato sul package software per l'analisi di reti `iGraph` [17].

Il layout del grafo viene tracciato cercando di minimizzare il numero di lati che si intersecano, utilizzando l'algoritmo per il plot di grafi non orientati discusso in [28].

Lo script sviluppato ricerca ricorsivamente all'interno di un folder specificato dall'utente tutti i file testuali corrispondenti a matrici di grafi semantici, e per ognuno di essi restituisce un file con l'immagine del grafo, evidenziando ogni nodo con un colore e una lettera corrispondente alla label associata. È possibile tracciare più grafi nella stessa immagine (ad esempio se si vogliono mostrare tutti i sottografi appartenenti a un cluster), inserendo in una singola matrice i blocchi corrispondenti alle matrici dei singoli grafi.

5.2 L'architettura del sistema

In questa sezione descriviamo l'architettura generale del sistema che implementa il modulo generativo. L'intero modulo, dai grafi di input costituenti

la base di conoscenza ai grafi generati in output, può essere considerato come una gerarchia di funzioni MATLAB, incluso lo stesso modulo rappresentato una una macro-funzione `GenerateGraphs`.¹ L'implementazione segue pertanto il classico paradigma procedurale.

Il modulo è progettato al fine di essere utilizzato offline, cioè di effettuare la generazione di grafi in modo indipendente dall'esplorazione del robot: il modulo è da intendersi come uno strumento per ampliare la base di conoscenza del robot. Il training del modello è un'attività che avviene offline; la generazione può essere facilmente estendibile online durante l'esplorazione da parte del robot.

Il nostro sistema si può considerare composto da cinque componenti principali, eseguiti sequenzialmente uno rispetto all'altro. Gli ultimi due moduli, responsabili della generazione di un nuovo grafo, vengono ripetuti un numero di volte pari al numero di grafi che si intende generare. La suddivisione dei componenti è molto simile a quella delle fasi concettuali presentata nel Capitolo 4, con alcune differenze dettate da esigenze implementative. Il diagramma di attività mostrato in Figura 5.3 mostra il flusso di esecuzione del modulo generativo a un livello di astrazione molto alto. Per ogni componente principale sono indicati sinteticamente i passi salienti.

- `Segmentation`: effettua la segmentazione dei grafi e calcola il grafo di similarità tra i sottografi ottenuti.
- `Clustering`: effettua il clustering dei sottografi, memorizza la configurazione dei cluster di ogni grafo e le connessioni tra i cluster.
- `ConnectionManager`: determina i parametri della distribuzione delle connessioni tra i cluster e calcola le probabilità del modello Bayesiano per le connessioni tra nodi.
- `SampleSubgraphs`: campiona la configurazione dei cluster di un nuovo grafo tramite il campionamento di Gibbs e i seleziona i sottografi corrispondenti.
- `ReconnectSubgraphs`: Campiona le connessioni tra i cluster del nuovo grafo e riconnette i nodi in base al modello Bayesiano.

¹Nel corso della trattazione si abuserà talvolta del termine “grafo” riferendosi direttamente alla sua rappresentazione tramite matrice di adiacenza con identificativi delle label sulla diagonale principale, o matrice del grafo semantico, come da noi definita nella Sezione 4.2.

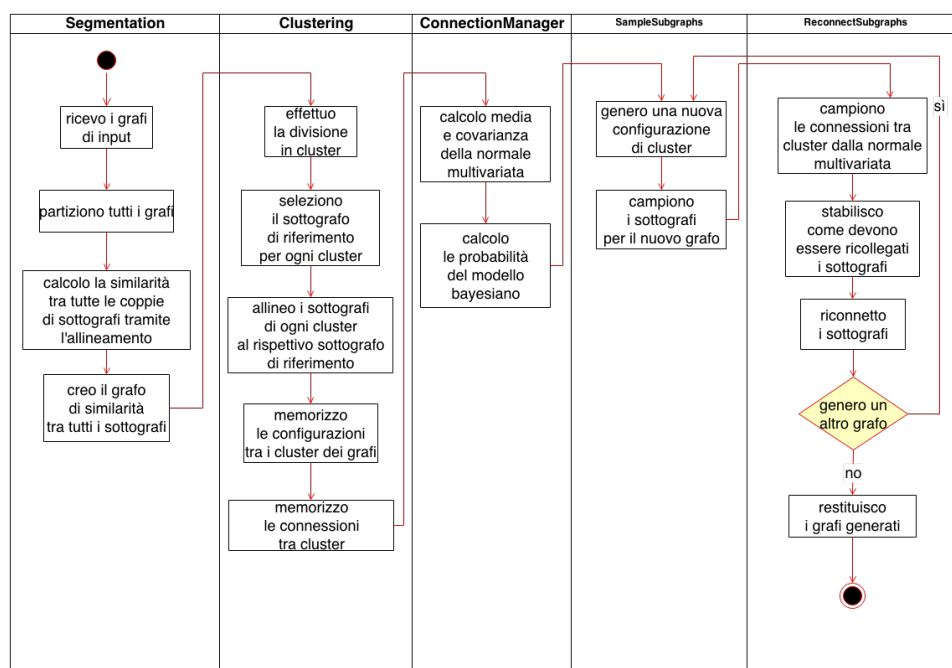


Figura 5.3: Diagramma di attività ad alto livello del modulo *GenerateGraphs*. Ogni colonna corrisponde a uno dei cinque componenti principali.

L'individuazione di componenti svolge una duplice funzione: da un lato porta a un maggior ordine logico durante l'implementazione, dall'altro fa sì che in caso emergano problematiche in un particolare componente si possa modificare con maggiore facilità solo il suddetto componente.

Prima di illustrare i componenti principali, apriamo una breve parentesi su un tipo di struttura dati ricorrente all'interno della trattazione.

L'utilizzo dei cell array

Nel corso dell'implementazione abbiamo utilizzato una struttura dati chiamata *cell array*.

Un cell array è una struttura dati in MATLAB simile a un array che permette di memorizzare elementi di tipo diverso: da un punto di vista concettuale si tratta di una struttura a puntatori in cui ogni elemento indicizzato dell'array punta a una determinata variabile. In questo caso, l'uso di un cell array permette di memorizzare matrici di grafi di dimensioni diverse o, nel caso di un cell array bidimensionale, di memorizzare in ogni riga un numero di elementi diverso (ad esempio le connessioni tra sottografi esisto-

no solo per alcune coppie di sottografi). I cell array sono stati utilizzati per memorizzare diverse matrici in un'unica struttura dati.

Per indicare un elemento appartenente a un cell array utilizzeremo la notazione $\text{var}\{i, j\}$, secondo la sintassi di MATLAB (nel caso di matrici o vettori invece gli indici sono racchiusi tra parentesi tonde, $\text{var}(i, j)$). Inoltre per facilitare la distinzione tra cell array e matrici o vettori, seguiamo la convenzione di attribuire alle variabili di tipo cell array nomi che iniziano con la lettera maiuscola.

5.2.1 Il componente **Segmentation**

Input: grafi di input `Graphs`, parametro di partizionamento `subPar`

Output: insieme dei sottografi estesi `SubExt`, array contenente il numero di sottografi estratti da ogni grafo `numS`, insieme delle connessioni tra sottografi estesi `ConnExt`, matrice di affinità tra i sottografi `sim`

Il componente `Segmentation` è il primo ad essere eseguito e si occupa principalmente della divisione in sottografi dei grafi di input. Ogni grafo viene importato da un file di testo contenente la relativa matrice di adiacenza con label; l'insieme dei grafi di input costituisce un cell array chiamato `Graphs`.

Oltre ai grafi di input, `Segmentation` riceve anche il parametro di partizionamento `subPar`. Questo è il parametro ρ che gestisce la condizione di terminazione dello step ricorsivo di `Ncut`, come discusso nella Sezione 4.3.3. Tipicamente si conosce un range di valori del parametro che determina una divisione in sottografi corretta, mentre il valore esatto al parametro viene attribuito empiricamente a seconda dei grafi di input a seguito di una serie di esecuzioni del componente, in base ai risultati ottenuti. I sottografi sono estesi con nodi fittizi alle dimensioni del sottografo più grande e memorizzati in un cell array chiamato `SubExt`, dove `SubExt\{k, i\}` rappresenta il sottografo i -esimo del grafo k -esimo. Un array `numS` memorizza il numero di sottografi estratti da ogni grafo. Le connessioni sono memorizzate in un cell array `ConExt`, dove `ConExt\{k, i, j\}` rappresenta la matrice di connessione tra i sottografi i -esimo e j -esimo del grafo k -esimo.

`Segmentation` si occupa anche della creazione del grafo di affinità tra tutti i sottografi. L'operazione sarebbe concettualmente legata alla fase di clustering, tuttavia è stata anticipata in sede implementativa. Il motivo è che, a parità di input e del criterio con cui sono stati segmentati i gra-

fi, il grafo di affinità risultante sarà sempre lo stesso. E' quindi possibile calcolare la similarità tra tutte le coppie di sottografi e realizzare il grafo di affinità, memorizzato in una matrice `sim` già all'interno del componente `Segmentation`.

5.2.2 Il componente `Clustering`

Input: insieme dei sottografi estesi `SubExt`, insieme delle connessioni tra sottografi estesi `ConnExt`, matrice di affinità tra i sottografi `sim`, parametro di clustering `clustPar`

Output: matrice dei riferimenti tra cluster e sottografi `clustRef`, insieme dei sottografi estesi allineati `SubAll`, insieme delle connessioni tra sottografi allineate `ConAll`, insieme dei vettori con le connessioni tra i cluster `ConClust`, matrice contenente la configurazione dei cluster nei grafi di input `confInput`, vettore degli slot dei cluster `posClust`

Il componente `Clustering` effettua il raggruppamento dei sottografi `SubExt` in cluster, e riorganizza alcuni dati indicizzandoli rispetto ai cluster.

La divisione del grafo di affinità avviene utilizzando lo stesso metodo di partizionamento basato su `Ncut` utilizzato in `Segmentation`. Anche in questo caso il clustering utilizza un parametro `clustPar` (ρ_c) per determinare la sensibilità del clustering e, di conseguenza, il numero di cluster. Il parametro è ottenuto empiricamente. I cluster di sottografi sono numerati progressivamente a partire da 1, assegnando quindi un indice a ogni cluster², e il risultato del clustering è memorizzato in una matrice `clustRef`, tale per cui `clustRef(k, i)` è il cluster all'interno del quale si trova il sottografo `SubExt{k, i}`. Dal momento che in questo caso è stata utilizzata per semplicità una matrice al posto di un cell array, `clustRef` ha un numero di colonne pari al numero massimo di sottografi all'interno di un grafo; le righe corrispondenti a grafi con un numero di sottografi `numS(k)` inferiore contengono gli elementi in eccedenza posti a zero.

Al termine della fase di clustering si identifica per ciascun cluster il grafo di riferimento che massimizza la somma delle similarità rispetto a tutti gli altri sottografi del cluster. Si effettua l'allineamento di questi ultimi rispetto al primo. Come conseguenza, i sottografi `SubExt` vengono trasformati in `SubAll` e delle connessioni tra i sottografi `ConnExt` in `ConAll`, dove alcune matrici rispetto a prima sono state permutate con l'allineamento.

²Per semplicità, ci si riferirà a un cluster con l'indice `c` ad esso associato.

La configurazione dei cluster per i grafi di input viene memorizzata in una matrice `confInput`. Ogni riga di `confInput` corrisponde a un vettore \mathbf{v} descritto nella Sezione 4.6. Una variabile `posClust` memorizza per ogni cluster quale slot (cioè quali colonne della matrice) è ad esso riservato.

L'ultima operazione effettuata da questo componente è la memorizzazione delle connessioni tra cluster; si tratta di una riduzione del cell array tridimensionale contenente matrici di connessione `ConAll` a un cell array bidimensionale contenente vettori di gradi di connessione `ConClust`.

Il modo in cui otteniamo le connessioni tra cluster è descritto approfonditamente nella Sezione 5.3.

5.2.3 Il componente `ConnectionManager`

Input: matrice dei riferimenti tra cluster e sottografi `clustRef`, insieme dei sottografi estesi allineati `SubAll`, insieme delle connessioni tra sottografi allineati `ConAll`, insieme dei vettori con le connessioni tra i cluster `ConClust`, dati aggregati sulle stanze `DataRooms`

Output: vettore della media delle connessioni tra cluster `meanVect`, matrici degli autovalori `covVal` e degli autovettori `covVett` della covarianza delle connessioni tra cluster, matrici delle probabilità della rete Bayesiana `pDoors`, `pCon` e `pPath3`

`ConnectionManager` prepara il campionamento delle connessioni svolgendo due operazioni. La prima è il calcolo dei parametri $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ della distribuzione normale multivariata con cui approssimiamo i gradi di connettività tra sottografi appartenenti a una coppia di cluster. Il vettore della media delle connessioni tra cluster `meanVect` e la matrice della covarianza delle connessioni tra cluster `covMatr` sono ottenuti dal cell array con le connessioni tra cluster `ConClust`. `covMatr` viene poi decomposta in matrice degli autovalori `covVal` e matrice degli autovettori `covVett`.

La seconda operazione è collegata al campionamento delle connessioni tra i nodi dei sottografi e consiste nel pre-computare i valori delle probabilità del modello Bayesiano descritto nella Sezione 4.8.2. Al fine di ottimizzare l'esecuzione del componente, sia `pDoors` sia `pPath3` sono calcolate a partire da alcuni dati aggregati delle stanze appartenenti ai grafi di input. Questi dati si trovano in un dataset e durante l'esecuzione del modulo vengono diret-

tamente importati e memorizzati in un cell array `DataRooms` che contiene per ogni stanza:

- la label corrispondente,
- il numero di porte,
- il numero di stanze di ogni tipo direttamente collegate ad essa.

In Figura 5.4 è riportato un esempio di dataset che contiene oltre alle informazioni sopracitate anche informazioni metriche quale le aree delle stanze e il rapporto tra le dimensioni principali (larghezza e lunghezza) delle stanze.

Label	Area	Doors	Dimensional Ratio	C	H	G	N	A	S	K	M	P	B	E	U
C	113.0	16	31	3	0	0	0	0	1	0	7	0	2	3	0
M	71	1	1.23	1	0	0	0	0	0	0	0	0	0	0	0
S	3.0	1	1.0	1	0	0	0	0	0	0	0	0	0	0	0
C	199.75	26	1.14	7	2	0	2	0	3	0	10	0	0	2	0
M	55.0	2	1	2	0	0	0	0	0	0	0	0	0	0	0
B	154.0	2	1.3	2	0	0	0	0	0	0	0	0	0	0	0
S	5.0	1	2	1	0	0	0	0	0	0	0	0	0	0	0
C	11.0	5	3.5	1	0	0	0	0	4	0	0	0	0	0	0
S	3.0	1	1	1	0	0	0	0	0	0	0	0	0	0	0

Figura 5.4: Esempio di file di testo contenente dati aggregati sulle stanze, tratto da [38]. Ogni riga contiene informazioni su una stanza.

$p_{\text{Doors}}(i, j)$ rappresenta la probabilità che un nodo di label i -esima (rispetto a un ordinamento convenzionale delle label salvato in un vettore `posLabel`) sia collegato a j nodi; p_{Path3} è una matrice tridimensionale, dove $p_{\text{Path3}}(l, i, j)$ è la probabilità che un nodo di label l -esima sia connesso ad almeno un nodo di label i -esima e ad un nodo di label j -esima.

p_{Con} invece viene calcolata da `SubAll` e `ConAll` e contiene le proprietà $p_{\text{Con}}(i, j)$ che una connessione tra due sottografi colleghi due nodi di label i -esima e j -esima, rispettivamente.

Un aspetto peculiare di `ConnectionManager` è la sua collocazione rispetto alla suddivisione tra fase di costruzione del modello generativo e fase di campionamento discussa nel Capitolo 4. Le operazioni svolte dal componente sono legate alla riconnessione dei sottografi campionati, ma non è necessario effettuarle nuovamente ogni volta che si genera un nuovo grafo

a partire dagli stessi dati iniziali. `ConnectionManager` si colloca a metà tra la fase di costruzione del modello generativo e la fase di campionamento dal modello.

- Può essere considerato un componente della fase di costruzione del modello generativo perché effettua operazioni di elaborazione dei dati e non operazioni di campionamento (che sono ripetute per ogni grafo che si genera).
- Può essere considerato un componente della fase di campionamento perché stabilisce indirettamente il modo in cui sono campionate le connessioni, oggetto del passo successivo.

Se la segmentazione e il clustering sono due operazioni disaccoppiate rispetto alla generazione dei grafi, nel senso che non impongono ulteriori vincoli alla fase di campionamento se non quello di un approccio part-based, la gestione delle connessioni richiede di ragionare in fin dall'inizio sulle distribuzioni dalle quali si campinano le connessioni. Cambiamenti nella fase di campionamento delle connessioni influenzano anche `ConnectionManager`: se ad esempio si cambiasse il modo di riconnettere i nodi, il calcolo delle probabilità del modello Bayesiano sarebbe anch'esso modificato di conseguenza.

5.2.4 Il componente `SampleSubgraphs`

Input: matrice dei riferimenti tra cluster e sottografi `clustRef`, insieme dei sottografi estesi allineati `SubAll`, matrice contenente la configurazione dei cluster nei grafi di input `confInput`, vettore degli slot dei cluster `posClust`

Output: insieme dei sottografi generati `SubGen`, vettore contenente il numero di sottografi appartenenti a ogni cluster `numSClust`

`SampleSubgraphs` rappresenta il componente principale della generazione part-based. Esso effettua il campionamento dei sottografi che compongono il grafo finale.

Il componente è costituito da due fasi principali: la prima effettua la generazione di una nuova configurazione dei cluster attraverso il campionamento di Gibbs, la seconda seleziona per ogni cluster un numero di sottografi pari al numero indicato dalla configurazione per quel cluster.

Il risultato del campionamento di Gibbs sulla matrice delle configurazioni dei grafi di input `confInput` è un vettore `confGen` che rappresenta la configurazione dei cluster del nuovo grafo. Successivamente, attraverso `posClust`, per ogni valore non nullo `confGen` viene estratto con probabilità uniforme un sottografo dal cell array dei sottografi `SubAll`.

L'insieme dei sottografi campionati è memorizzato in un cell array bidimensionale `SubGen`, indicizzato rispetto ai cluster: un elemento `SubGen{c, i}` è la matrice del sottografo *i*-esimo (nel grafo generato) appartenente al cluster *c*-esimo. Allo stesso modo dei sottografi dei grafi di input, un vettore `numSClust` memorizza il numero di sottografi campionati per ogni cluster, e di conseguenza, il numero di elementi in ogni riga di `SubGen`.

5.2.5 Il componente **ReconnectSubgraphs**

Input: vettore della media delle connessioni tra cluster `meanVect`, matrici degli autovalori `covVal` e degli autovettori `covVect` della covarianza delle connessioni tra cluster, vettore con parametri di thresholding `beta`, matrici delle probabilità della rete Bayesiana `pDoors`, `pCon` e `pPath3`, insieme dei sottografi generati `SubGen`, vettore contenente il numero di sottografi appartenenti a ogni cluster `numSClust`

Output: matrice del grafo generato `graphGen`

Il componente `ReconnectSubgraphs` conclude la fase di campionamento, permettendo di assemblare il grafo finale attraverso la riconnessione dei nodi dei sottografi campionati. Il processo di riconnessione stesso avviene tramite campionamento.

La prima parte del componente effettua il campionamento dei gradi di connettività (il numero di lati connessi) tra tutte le coppie di sottografi da una distribuzione normale multivariata il cui vettore delle medie `meanVect` e decomposizione della matrice di covarianza `covMatr` sono già stati calcolati nel componente `ConnectionManager`. Un cell array quadridimensionale `ConGen` contiene i gradi di connettività `ConGen{a, b, i, j}` tra l'*i*-esimo sottografo campionato dal cluster *a* e il *j*-esimo sottografo campionato dal cluster *b*. Poiché i gradi di connettività campionati devono essere valori interi, un vettore di parametri ottenuti empiricamente, `beta`, effettua la discretizzazione dei valori di `ConGen`, mappando ogni intervallo continuo in un valore intero.

La selezione dei nodi di collegamento avviene in una seconda fase per ogni coppia di sottogradi che risulta collegata da almeno una connessione. Utilizzando le matrici `pDoors`, `pCon` e `pPath3` calcolate dal `ConnectionManager`, per ogni coppia di nodi si calcola la probabilità che siano connessi come probabilità congiunta della rete Bayesiana da noi definita nella Sezione 4.8.2, selezionando le connessioni a partire dai valori di probabilità congiunta più alti. Da ogni grado di connettività `ConGen{a,b,i,j}` viene quindi estratta una matrice di connessione `ConSGen{a,b,i,j}`.

L'ultimo passo, la creazione del grafo finale, avviene inserendo in una matrice a blocchi `graphGen` i sottogradi di `SubGen` e le connessioni di `ConSGen` nei blocchi corrispondenti. Una volta formata la matrice del grafo è possibile rimuovere le righe e le colonne di zeri, cioè quelle corrispondenti a un nodo fittizio introdotto da `Segmentation`. Con questo si conclude il processo di generazione di un grafo.

5.2.6 Il modulo generativo `GenerateGraphs`

Input: grafi di input `Graphs`, parametro di partizionamento dei grafi `subPar`, parametro di clustering `clustPar`, numero di grafi da generare `nGen`
output: grafi di output `GenGraphs`

Nel complesso, i componenti formano il modulo `GenerateGraphs` che, a partire dai grafi di input memorizzati nel cell array `Graphs`, genera un numero `nGen` di grafi di output e li memorizza nel cell array `GenGraphs`.

I parametri `subpar` e `clustpar` di partizionamento dei grafi e clustering dei sottogradi sono forniti in ingresso all'intero modulo in modo da poter essere regolati dall'esterno.

E' possibile generare un qualunque numero di grafi dato un certo insieme di input. Per fare ciò è sufficiente ripetere l'esecuzione dei componenti `SampleSubgraphs` e `ReconnectSubgraphs`. Questo permette di minimizzare l'overhead dato dalla generazione di un nuovo grafo.

Dipendenze tra i componenti

In Figura 5.5 sono mostrate le dipendenze tra i componenti. Se si desidera modificare l'implementazione di un componente bisogna verificare che il suo

input sia coerente con i componenti da cui dipende e che il suo output sia coerente con i componenti che da esso dipendono.

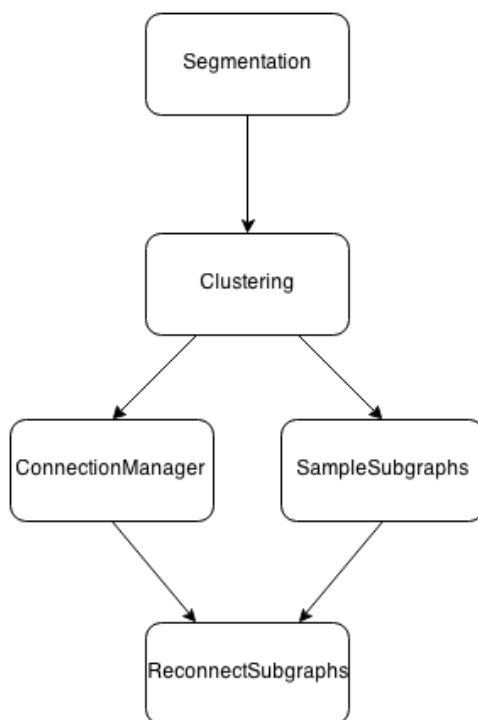


Figura 5.5: Dipendenze tra i componenti di *GenerateGraphs*. Gli archi modellano la relazione "l'output di A è richiesto da B". Si può notare come *ConnectionManager* e *SampleSubgraphs* non dipendano tra loro, per cui la loro esecuzione potrebbe anche essere resa parallela.

5.3 Alcune funzioni del modulo

Dopo aver tracciato le caratteristiche principali dei componenti, illustriamo alcune funzioni particolarmente significative che sono state implementate. Le funzioni che descriviamo con maggior dettaglio sono state scelte o perché sono importanti da un punto di vista concettuale, o perché presentano degli aspetti implementativi che è opportuno spiegare.

5.3.1 Implementare il partizionamento di un grafo

Il partizionamento dei grafi viene gestito da una funzione `partition` che viene utilizzata in `Segmentation` per dividere i grafi di input in sottografi,

e in `Clustering` per dividere il grafo di affinità.

La funzione `partition` è basata sull'algoritmo di `Ncut`, già illustrato in precedenza; mostriamo ora la logica con cui è stata realizzata l'implementazione.

La funzione riceve in input la matrice `G` di un grafo e il parametro per la ricorsione `alfa` (che nel caso specifico della segmentazione rappresenta ρ e in quello del clustering rappresenta ρ_c). L'output è costituito, oltre che dall'insieme delle matrici dei sottografi `S` (implementato come un cell array monodimensionale), dall'insieme delle matrici di connessione tra tutte le coppie di sottografi ottenute `C` (implementato come un cell array bidimensionale), e una matrice `clust` che tiene traccia per ogni sottografo ottenuto di quali nodi del grafo originario contiene: `clust(i, j)` è posto a 1 se il j -esimo nodo del grafo `G` si trova nel i -esimo sottografo. Quali output siano utilizzati e quale sia il loro significato varia a seconda che `partition` sia eseguita nel contesto di `Segmentation` o di `Clustering`: nel primo caso `S` rappresenta i sottografi ottenuti da uno dei grafi di input, `C` le matrici di connessione tra i sottografi e `clust` non viene utilizzato; nel secondo caso `S` rappresenta i cluster di sottografi ottenuti dal grafo di affinità, `C` non viene utilizzato e `clust` serve per associare a ogni sottografo il numero del cluster a cui appartiene.

La ricorsione di `Ncut` è stata resa in modo iterativo. In Figura 5.6 è mostrato il flusso di esecuzione di `partition`: la divisione dei sottografi avviene in ampiezza considerando a ogni iterazione i grafi parziali fino a quel momento ottenuti e controllando se ciascuno di essi può essere ulteriormente suddiviso. Quando un sottografo non viene più suddiviso perché il taglio ha un valore di `Ncut` superiore al parametro `alfa`, questo è un sottografo finale e non deve più essere considerato nelle iterazioni successive. Se durante un'iterazione non vi è stata almeno una suddivisione di un grafo parziale, significa che è stata raggiunta la situazione finale e si restituisce l'insieme dei sottografi `S` così ottenuto.

La divisione di un grafo parziale `G` è effettuata da una funzione `splICE(G)`, che oltre a restituire le matrici delle due partizioni `P1`, `P2`, quella delle connessioni `Con` e il valore `ncut` relativo al taglio, fornisce anche un vettore `a` che tiene traccia dell'appartenenza dei nodi di `G` a `P1` o `P2`. Il vettore `a` è importante per due motivi: il primo è che permette in seguito a una divisione di un grafo parziale, di aggiornare coerentemente le matrici di connessione. Se un grafo `G` è diviso inizialmente in `A` e `B`, viene creata

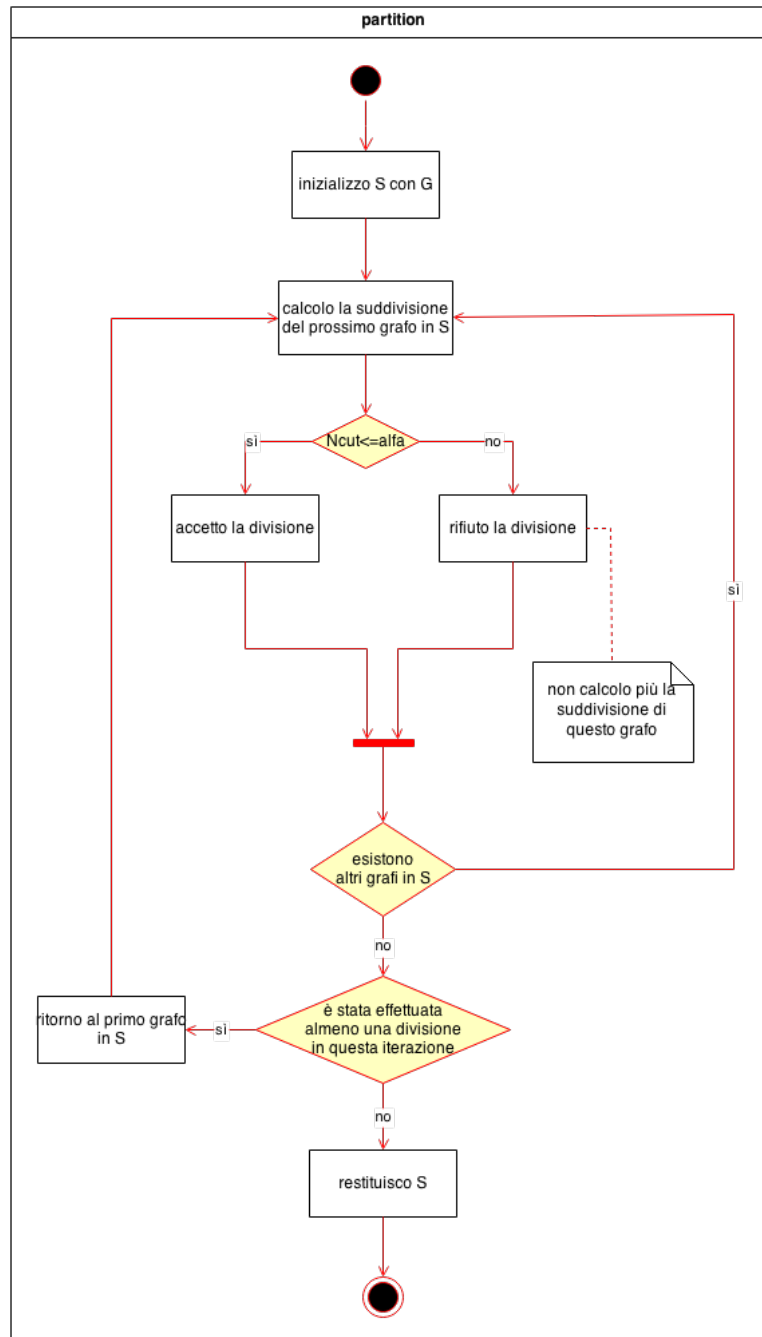


Figura 5.6: Diagramma della divisione in sottografi operata da *partition*. Si può notare come la ricorsione venga gestita in modo iterativo, considerando ad ogni iterazione i grafi parziali temporaneamente ottenuti e memorizzando quali non devono più essere considerati per una suddivisione nelle iterazioni successive.

una matrice di connessione `Cab`. Se nella iterazione successiva `A` è diviso in `A1` e `A2`, oltre alla formazione di `Ca1a2`, si deve aggiornare la matrice `Cab`, suddividendola in due matrici `Ca1b` e `Ca2b`; questo è possibile se si conosce un mapping tra i nodi di `A` e i nodi delle sue partizioni `A1`, `A2`.

Il vettore `a` di `splice` contribuisce inoltre ad aggiornare la matrice `clust` della funzione chiamante `partition` che memorizza il mapping tra i nodi del grafo iniziale e i nodi dei sottografi (cluster).

5.3.2 Memorizzare le connessioni tra cluster

Il calcolo delle connessioni tra cluster `ConClust` a partire dalle matrici di connessione tra i sottografi memorizzate in `ConAll` avviene al termine del componente `Clustering`. Una funzione `conMatr2Clust` effettua le seguenti trasformazioni:

1. ogni matrice di connessione `ConAll{k, i, j}` è ridotta al grado di connettività `ConValue{k, i, j}`, effettuando la somma di tutti gli elementi di `ConAll{k, i, j}`;
2. gli elementi `conValue{k, i, j}` sono aggregati, formando dei vettori `ConClust{c1, c2}`, dove `c1` e `c2` sono gli indici di due cluster.

Il processo di riduzione è piuttosto semplice, mentre il processo di aggregazione è più elaborato. Questo può considerarsi suddiviso a sua volta in due fasi, dove un primo livello di aggregazione inserisce ogni elemento `conValue{k, i, j}` relativo a due sottografi appartenenti, rispettivamente, ai cluster `c1` e `c2`, in coda a un vettore `ConClustGraph{k, c1, c2}`, che contiene tutti i gradi di connettività relativi alla coppia di cluster (`c1`, `c2`) per il grafo `k`. Successivamente, i vettori relativi alla stessa coppia di cluster costituiscono i blocchi di un unico vettore `ConClust{c1, c2}`; una matrice tridimensionale `SlotCon` memorizza l'indice iniziale di ogni blocco `ConClustGraph{k, c1, c2}` all'interno di `ConClust{c1, c2}`.

5.3.3 Calcolare i parametri della distribuzione normale multivariata

La distribuzione dei gradi di connettività tra i cluster viene approssimata a una normale multivariata i cui parametri, vettore delle medie `meanVect` e matrice delle covarianze `covMatr` sono calcolati da una funzione `calcMeanCov` nel componente `ConnectionManager` a partire dalle connessioni tra cluster `ConClust`.

Il primo passo effettuato è l'eliminazione delle coppie di cluster mai connesse: se un vettore $\text{ConClust}\{c1, c2\}$ non ha elementi diversi da zero, la coppia di cluster $(c1, c2)$ non sarà considerata nel calcolo della media e della covarianza. A ogni coppia di cluster (a, b) non eliminata viene associato un indice x che la identifica all'interno delle nuove strutture dati.

Il vettore delle medie meanVect è ottenuto calcolando la media di ogni vettore $\text{ConClust}\{c1, c2\}$ e inserendola nella posizione x corrispondente.

Gli elementi della matrice delle covarianze $\text{covMatr}(x, y)$ tra (a, b) e (c, d) sono ottenuti nel modo seguente:

1. per ogni grafo k si estrae da $\text{ConClust}\{a, b\}$ il relativo blocco $\text{ConClustGraph}\{k, a, b\}$;
2. analogamente si ottiene $\text{ConClustGraph}\{k, c, d\}$ da $\text{ConClust}\{c, d\}$;
3. si costruiscono un vettore $\text{ConClustExtgraph}\{k, a, b\}$ e un vettore $\text{ConClustExtGraph}\{k, c, d\}$ i cui elementi nelle posizioni corrispondenti rappresentano tutti gli abbinamenti tra gli elementi di $\text{ConClustGraph}\{k, a, b\}$ e gli elementi di $\text{ConClustGraph}\{k, c, d\}$;
4. Si ripete il procedimento per ogni altro grafo ottenendo i vettori a blocchi $\text{ConClustExt}\{a, b\}$ e un vettore $\text{ConClustExt}\{c, d\}$;
5. Si calcola la covarianza tra $\text{ConClustExt}\{a, b\}$ e $\text{ConClustExt}\{c, d\}$.

Se ad esempio vi sono tre grafi e le connessioni tra cluster relative alle coppie (a, b) e (c, d) sono:

$$\text{ConClust}\{a, b\} = \left[\begin{array}{c|c|c} 1, 0, 1 & 1, 2 & 0, 1, 0 \end{array} \right]$$

$$\text{ConClust}\{c, d\} = \left[\begin{array}{c|c|c} 1, 1 & 1, 1, 0 & 2, 0, 1, 0 \end{array} \right]$$

i vettori tra i quali si calcola la covarianza sono:

$$\text{ConClustExt}\{a, b\} = \left[\begin{array}{c|c|c} 1, 0, 1, 1, 0, 1 & 1, 2, 1, 2, 1, 2 & 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0 \end{array} \right]$$

$$\text{ConClustExt}\{c, d\} = \left[\begin{array}{c|c|c} 1, 1, 1, 1, 1, 1 & 1, 1, 1, 1, 0, 0 & 2, 2, 2, 0, 0, 0, 1, 1, 1, 0, 0, 0 \end{array} \right]$$

In questo caso, ad esempio, il secondo blocco di $\text{ConClust}\{a, b\}$ indica che nel secondo grafo vi sono due coppie di sottografi associati rispettivamente ai cluster a e b ; una coppia di sottografi è connessa da un lato, mentre l'altra è connessa da due lati. Ciascuno di questi valori è ripetuto in $\text{ConClustExt}\{a, b\}$ un numero di volte necessario ad essere confrontato con tutti valori del secondo blocco di $\text{ConClust}\{c, d\}$, in questo caso tre.

Capitolo 6

Realizzazioni sperimentali e valutazione

*Per quanti blocchi, tovaglioli e margini di giornali riempiano di note ed equazioni, non riescono a giustificare quella frazione. Un fatto incontrovertibile si è messo di traverso: le misure esterne **devono** corrispondere a quelle interne. La fisica dipende da un universo che comunque ha il suo centro nel segno di uguale. Come scrisse lo scienziato e talvolta teologo David Conte: “Dio è in fondo un segno di uguaglianza e, almeno fino a oggi, un dato cui l’umanità è stata capace di credere è che l’universo ha un senso”.*

Mark Danielewski, *Casa di foglie*

In questo capitolo illustriamo le prove sperimentali effettuate sull’implementazione del modello generativo. La trattazione è divisa in tre parti:

- Nella prima parte illustriamo i dataset su cui sono stati effettuati gli esperimenti e l’impostazione generale delle prove sperimentali;
- Nella seconda parte mostriamo i test effettuati su aspetti specifici del sistema implementato;
- Nella terza parte valutiamo l’applicazione complessiva del sistema.

La valutazione dei risultati viene fatta sia da un punto di vista qualitativo (mostrando esempi visuali sui grafi), sia da un punto di vista quantitativo (utilizzando alcune metriche sui grafi).

6.1 I dataset utilizzati

Il sistema realizzato nel lavoro di tesi è indirizzato all'ambito specifico del mapping semantico nella robotica mobile autonoma; le prove sperimentali vertono quindi sull'applicazione del modello generativo su grafi che rappresentano mappe semantiche di edifici.

I dataset che abbiamo utilizzato contengono grafi rappresentanti mappe semantiche ottenute da edifici esistenti. In particolare, le prove sperimentali sono state effettuate su tre dataset distinti, dove ogni dataset rappresenta edifici di una determinata tipologia edilizia. Le tipologie edilizie che abbiamo considerato sono le SCUOLE, gli UFFICI e le ABITAZIONI [40].

Gli edifici rappresentati nei dataset sono stati ottenuti da testi reperiti nelle biblioteche di Architettura del Politecnico di Milano, campus di Milano Leonardo e di Bovisa Durando. L'elenco dei testi utilizzati è riportato qui in seguito, diviso per categoria.

Testi utilizzati per la tipologia edilizia SCUOLE

- Bradford Perkins, *Building type basics for elementary and secondary schools*. New York: Wiley, 2001.
- Bradford Perkins, Rymond Bordwell, *Building type basics for elementary and secondary schools*, 2 Ed.. Hoboke : Wiley, 2010.
- Sibylle Kramer, *Schools : educational spaces*. Berlin: Braun, 2010.
- Comune di Roma, *Ai bambini piacciono le scuole ... : il comune le ha fatte*, 2001-2006. Roma: Gangemi, 2006.

Testi utilizzati per la tipologia edilizia UFFICI

- Silvio San Pietro, Alessandra Vasile, *New offices in Italy*. Milano: L'archivoltò, 2003.
- Stefano Casciani, Orietta Fiorenza, Massimo Roj, *Workspace/Workspace: i nuovi scenari dell'ufficio*. Milano: Skira, 2000.
- Corrado Landi, *Manuale per la progettazione degli uffici*. Roma: Dei, 2002.
- Alexi Marmot, Joanna Eley, *Office space planning: designing for tomorrows workplace*. New York: McGraw-Hill, 2000.

Testi utilizzati per la tipologia edilizia ABITAZIONI

- Marina Pia Arredi, *La casa unifamiliare del Novecento: un secolo di architettura abitativa*. Torino: UTET libreria, 1997.
- Hilary French, *New urban housing*. London: Laurence King, 2006.
- Ivo Ceccarini, *Composizione della casa*. Milano: Hoepli, 1979.

Costruire ogni volta un modello generativo su una specifica tipologia edilizia ha un duplice scopo: da un lato si riduce la variabilità tra i grafi considerando edifici che condividono implicitamente caratteristiche strutturali simili; da un altro, si vuole verificare se, ponendo dei vincoli sul tipo di dati modellati, questi vincoli si mantengono anche nei dati campionati. Il secondo aspetto è particolarmente interessante se messo in relazione all'uso predittivo del modello generativo.

Il modello generativo è testato ponendo particolare attenzione alla tipologia edilizia SCUOLE.

Abbiamo scelto di concentrarci principalmente sugli edifici scolastici perché presentano due importanti caratteristiche:

- sono edifici tipicamente molto estesi e contengono molte stanze al loro interno;
- la loro struttura presenta alcune regolarità e simmetrie in più rispetto ad altre tipologie di edifici.

L'estensione di un complesso scolastico può rendere utile l'uso di un modello generativo per fare predizione, ad esempio in un contesto in cui un robot non conosca l'intera planimetria di un edificio, ma solo alcune sezioni di esso.

La presenza di simmetrie all'interno delle scuole è un aspetto che può risultare interessante nei confronti dell'utilizzo di un modello generativo come il nostro, in cui la segmentazione e il clustering sono indotti dalla struttura stessa dei dati. In altre parole, se i sottografi che compongono un grafo sono facilmente riconoscibili e se tali parti tendono a essere categorizzabili in maniera efficace, allora il processo di creazione del modello rappresenterà più fedelmente i dati. Infine, scuole diverse sono comparabili tra loro perché costruite in modo da rendere evidente la suddivisione tra aree diverse dell'edificio, ad esempio le aree mensa o le aree didattiche. Anche i corridoi che collegano molte stanze sono disposti secondo una certa regolarità, generalmente rispettata in tutti gli edifici scolastici.

6.1.1 Le label delle stanze

Le stanze sono state etichettate con un sistema di labeling specifico per ogni tipologia. Un criterio standard è stato definito come riferimento su cui modellare i singoli sistemi di labeling. Il criterio standard di classificazione delle stanze è mostrato in Figura 6.1 e prescinde dalla tipologia edilizia considerata. Possiamo inoltre classificare due macro-classi di stanze che individuano uno schema elementare di label $\bar{\mathcal{L}}$:

- un CORRIDOIO che svolge la funzione principale di collegare altre stanze tra loro;
- una STANZA che ospita attività di vario tipo.

A partire da $\bar{\mathcal{L}}$ è possibile definire uno schema di label più completo, $\mathcal{L} = \{\text{CORRIDOIO, HALL, INGRESSO, STANZA PICCOLA, STANZA MEDIA, STANZA GRANDE}\}$. Consideriamo \mathcal{L} la classificazione standard di riferimento.

La classificazione standard è la stessa utilizzata per la tipologia edilizia ABITAZIONI.

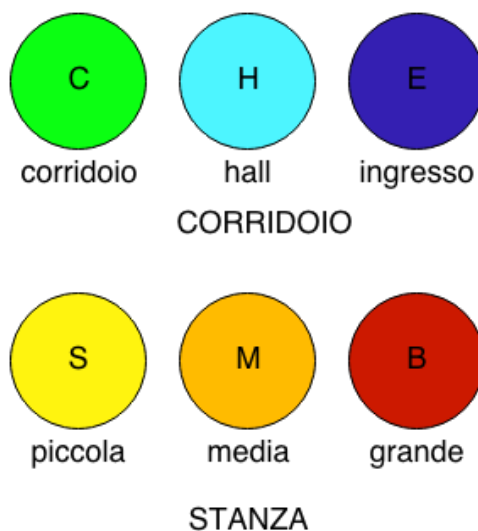


Figura 6.1: La classificazione standard delle stanze, utilizzata come riferimento.

Nel caso delle altre due tipologie, all'intero delle stanze si considera un'ulteriore suddivisione tra stanze funzionali e stanze di servizio:

- le stanze funzionali ospitano attività legate alla funzione primaria di un edificio: in una scuola, ad esempio le classi e le aule destinate all'attività didattica sono stanze funzionali;
- le stanze di servizio sono utilizzate per attività di contorno rispetto alla funzione primaria: una mensa all'interno di un edificio per uffici ne è un esempio.

La distinzione tra stanze funzionali e stanze di servizio è più netta per edifici pubblici quali scuole o uffici, mentre è meno netta nel caso delle abitazioni, essendo tutte le stanze destinate all'essere abitate in una qualche forma. In quest'ultimo caso si è preferito considerare le stanze esclusivamente come funzionali.

Le label nella tipologia edilizia scuole

In Figura 6.2 sono mostrate le label utilizzate per le SCUOLE, di cui forniamo una breve descrizione:

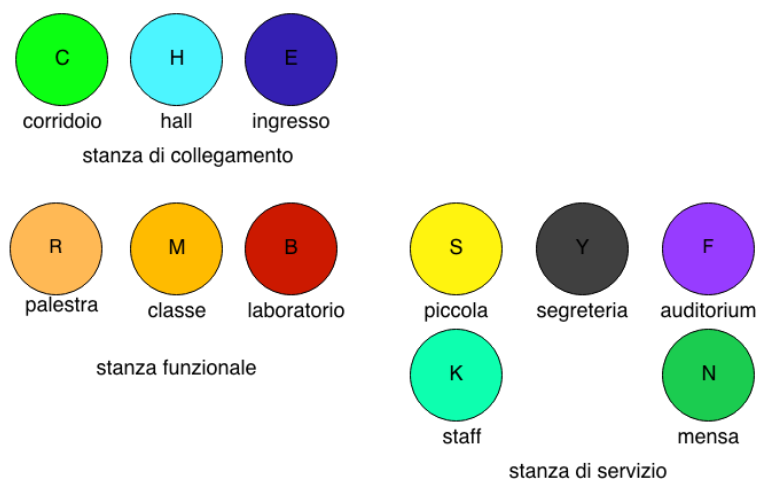


Figura 6.2: La classificazione delle stanze per le SCUOLE.

- **Classe:** una singola classe o aula in cui sono tenute le lezioni. È l'equivalente per le scuole di una stanza funzionale media
- **Laboratorio:** una classe attrezzata a specifiche funzioni, come il laboratorio di disegno, di musica o la sala computer. È l'equivalente per le scuole di una stanza funzionale grande.

- **Palestra:** stanza di grandi dimensioni, contiene in genere un campo da basket. Solitamente si trova in un edificio esterno, staccato dal corpo centrale della scuola. Spesso ha ingressi propri sulla strada.
- **Stanza di servizio piccola:** Stanza di dimensioni molto piccole con funzione di servizio. Può essere un bagno o un ripostiglio.
- **Staff:** stanza utilizzata da un insegnante per preparare le lezioni.
- **Segreteria/Sala professori:** stanza dove vengono svolte attività burocratiche e viene gestita la relazione con l'esterno. Può anche essere una sala professori. È l'equivalente di una stanza di servizio media.
- **Auditorium:** sala comune con platea da una parte e la capacità di accogliere molte persone dall'altra. È l'equivalente di una stanza di servizio grande.
- **Mensa/Biblioteca:** sala di grandi dimensioni con una funzione specifica.

Le label degli uffici

In Figura 6.3 sono mostrate le label per gli uffici.

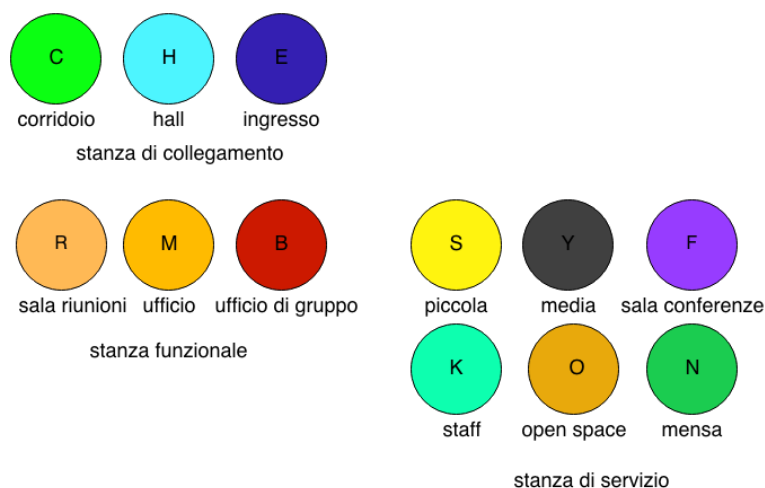


Figura 6.3: La classificazione delle stanze per gli uffici.

Indichiamo anche in questo caso le categorie di stanze specifiche rispetto alla classificazione standard.

- **Ufficio:** ufficio standard, singolo o doppio. È l'equivalente per gli uffici di una stanza funzionale piccola.

- **Ufficio deluxe:** grande ufficio singolo o doppio, riservato a ruoli dirigenziali, spesso ha una propria sala conferenza e un ufficio connesso di supporto.
- **Sala riunioni:** sala riunioni di piano, tipicamente con un tavolo al centro.
- **Stanza di servizio piccola:** stanza di dimensioni molto piccole con funzione di servizio. Può essere un bagno o un ripostiglio.
- **Stanza per lo staff:** sala di appoggio ad un piano, come una cucina, una zona relax, una reception o un asilo dipendenti per le grosse aziende che se ne dotano.
- **Sala conferenze:** grossa stanza per riunioni con buona parte del personale, solitamente con un palco e una platea. È l'equivalente per gli uffici di una stanza di servizio grande.
- **Open space:** ambiente, di grandi dimensioni, in cui si svolgono diversi tipi di attività. In genere è un ufficio di gruppo dove lavorano assieme dalle 5 alle 20 persone, con un compito preciso comune; esempi ne sono un call center aziendale o l'ufficio vendite.
- **Mensa/Biblioteca:** sala di grosse dimensioni che fornisce un supporto ben preciso, come una biblioteca o una mensa.

6.2 Analisi delle fasi principali

In questa sezione mostriamo le prove sperimentali effettuate sulle fasi principali della creazione del modello generativo. In particolare, consideriamo in modo approfondito i seguenti aspetti:

- la segmentazione dei grafi in sottografi, analizzando l'andamento del numero medio di sottografi e del numero medio di nodi in funzione del parametro di segmentazione ρ ;
- il clustering dei sottografi, analizzando l'andamento del numero medio di cluster e del numero medio di sottografi in un cluster in funzione del parametro di clustering ρ_c ;
- il campionamento di Gibbs, determinando il numero di iterazioni necessarie affinché un campionatore raggiunga una distribuzione stazionaria.

- il processo di riconnessione dei sottografi campionati. discutendo le problematiche legate al thresholding dei gradi di connettività tra i sottografi e mostrando una soluzione al problema.

6.2.1 Analisi della segmentazione dei grafi

La segmentazione dei grafi in sottografi (introdotta nella Sezione 4.3 e implementata nella Sezione 5.2.1) usando Ncut è il primo passo fondamentale della costruzione del modello generativo. Da un certo punto di vista, questa si può considerare l'operazione più importante svolta poiché qui vengono stabiliti i componenti costitutivi dei grafi campionati. In Figura 6.4 è mostrato un esempio di un grafo rappresentante una scuola che viene diviso in sottografi attraverso l'utilizzo di Ncut. In questo caso il processo di segmentazione porta a 6 sottografi, di cui 4 presentano una struttura *a stella* (una stanza CORRIDOIO o HALL centrale a cui sono collegate tante stanze), e i restanti hanno una struttura più complessa.

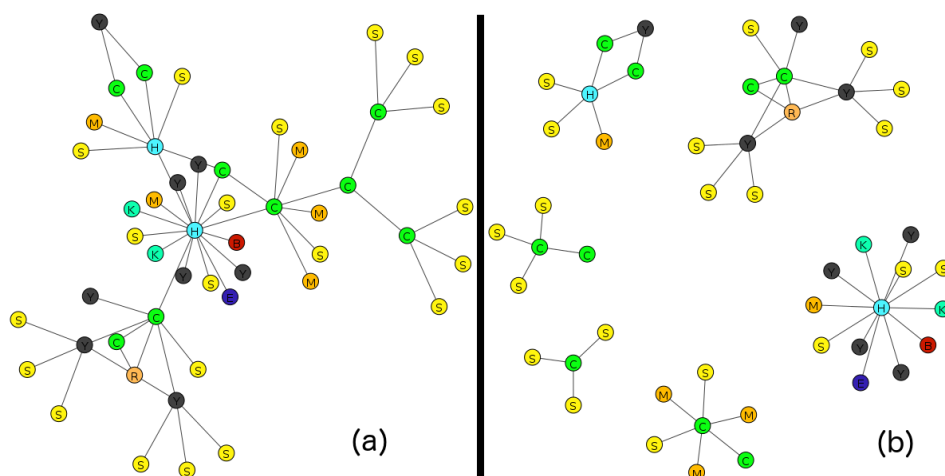


Figura 6.4: Esempio di un grafo che rappresenta una scuola (a) e della sua divisione in sottografi (b).

Da un punto di vista quantitativo, analizziamo due informazioni legate al processo di segmentazione:

- il numero di sottografi in cui un grafo viene diviso,
- le dimensioni di un sottografo (la cardinalità dei nodi).

Nel caso applicativo, oggetto di questo lavoro di tesi, i sottografi sono interpretabili come delle aree all'interno di un edificio in cui le stanze sono

particolarmente connesse tra loro e poco o affatto connesse alle altre stanze. Un esempio potrebbe essere l'area amministrativa in un edificio pubblico.

Il numero di sottografi di un grafo può quindi essere fatto corrispondere al numero di queste aree in un piano di un edificio, e il numero di nodi in un sottografo rappresenta la quantità di stanze in ciascuna di queste aree.

Come abbiamo illustrato nella Sezione 4.3.3, la segmentazione è regolata da un parametro ρ che, gestendo la terminazione della ricorsione di Ncut, determina indirettamente questi due elementi. Abbiamo considerato il valor medio $\mathbb{E}[|S|]$ su un dataset di grafi \mathcal{G} del numero di sottografi $|S|$ ottenuti da un grafo $G \in \mathcal{G}$ e la cardinalità media dei nodi $\mathbb{E}[|s|]$ di un sottografo $s \in \mathcal{S}$, facendo variare il parametro ρ nell'intervallo di valori ragionevoli $[0.2, 1]$. Per $\rho < 0.2$ la condizione di taglio è infatti troppo stringente, e per $\rho > 1$ vengono permessi tagli di un numero di lati maggiore rispetto al numero di quelli che rimangono. L'esperimento è stato effettuato sui tre dataset.

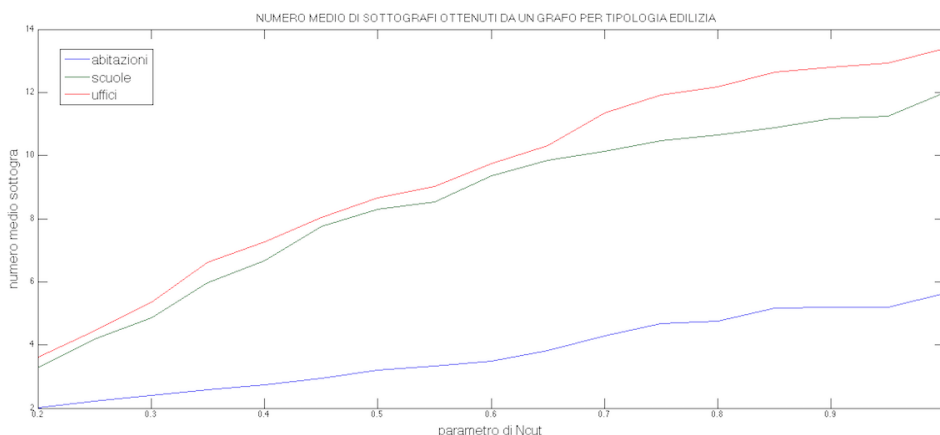


Figura 6.5: Andamento del numero medio di sottografi in cui è diviso un grafo al variare del parametro di Ncut ρ . In blu sono rappresentate le ABITAZIONI, in verde le SCUOLE e in rosso gli UFFICI.

In Figura 6.5 è mostrato l'andamento del numero medio di sottografi ottenuti da un grafo. In accordo a quanto discusso nel Capitolo 4, il valore di $|S|$ cresce con l'incremento di ρ . La crescita è abbastanza regolare in tutti e tre i dataset: nel caso delle ABITAZIONI la crescita è quasi lineare, mentre SCUOLE e UFFICI presentano una leggera concavità verso l'alto. In generale, si può vedere come nel caso della tipologia ABITAZIONI il numero medio di sottografi sia significativamente più basso rispetto a quello delle

altre due tipologie edilizie; questo significa in parte che nelle SCUOLE e negli UFFICI è più evidente la suddivisione in aree distinte, ma è anche coerente al minor numero dei nodi di un'abitazione rispetto alle altre tipologie.

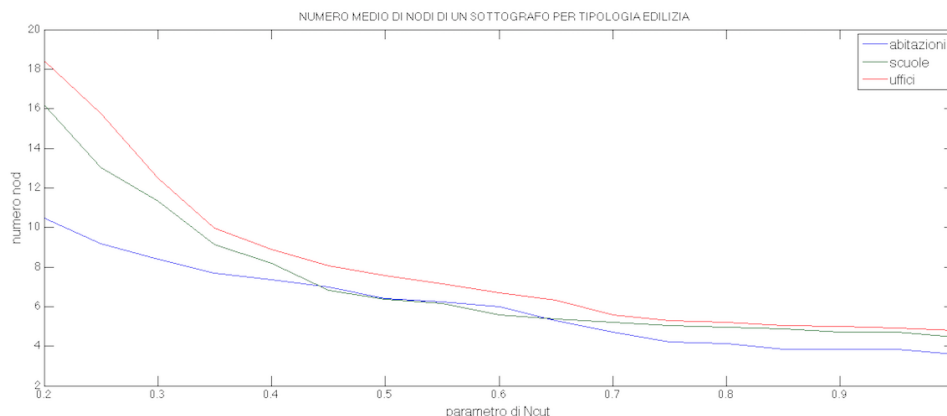


Figura 6.6: Andamento della cardinalità media dei nodi dei sottografi al variare del parametro di Ncut ρ . In blu sono rappresentate le ABITAZIONI, in verde le SCUOLE e in rosso gli UFFICI.

In Figura 6.6 è invece mostrato l'andamento delle dimensioni medie di un sottografo: in questo caso tutti e 3 i dataset presentano un rapido decremento fino a $\rho = 0.35$, a cui segue un andamento più stabile in cui le tre curve sono molto vicine tra loro (nell'intorno di $\rho = 0.5$ le dimensioni dei sottografi di SCUOLE e ABITAZIONI sono pressoché uguali).

La scelta del valore da attribuire a ρ viene fatta in base a un compromesso fra numero e dimensione dei sottografi. A tal proposito, in Figura 6.7, abbiamo effettuato un confronto tra $\mathbb{E}[|S|]$ e $\mathbb{E}[|s|]$ sovrapponendo negli stessi grafici le due informazioni. In particolare, in ognuno dei tre casi vi è un valore di $\hat{\rho}$ (quello in cui si incontrano le curve) per cui:

$$\rho < \hat{\rho} \rightarrow \mathbb{E}[|S|] < \mathbb{E}[|s|]$$

$$\rho > \hat{\rho} \rightarrow \mathbb{E}[|S|] > \mathbb{E}[|s|]$$

$\hat{\rho}$ rappresenta il valore del parametro per cui, mediamente, il numero di sottografi in cui è diviso un grafo è pari al numero di nodi di un sottografo. Nel caso di SCUOLE e UFFICI, $\hat{\rho} \approx 0.45$, mentre nel caso delle ABITAZIONI, $\hat{\rho} \approx 0.7$.

Utilizzare valori di ρ vicini a $\hat{\rho}$ può essere una buona idea se si vuole mantenere un certo bilanciamento tra quantità di gruppi di stanze indivi-

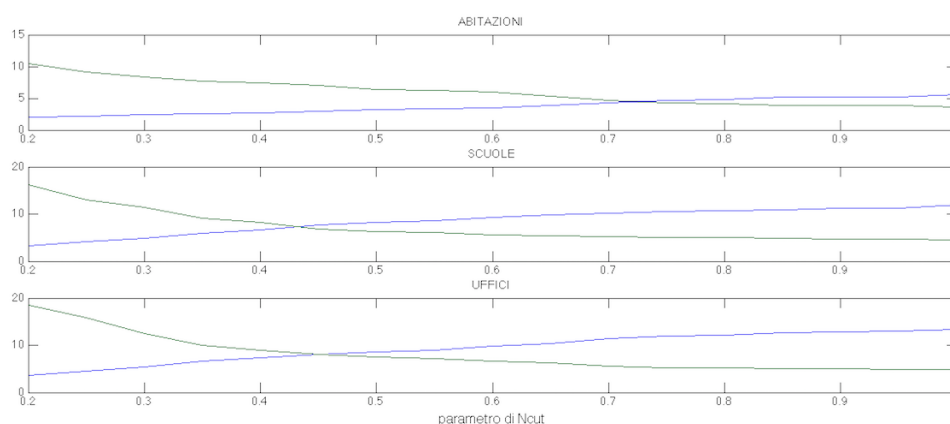


Figura 6.7: Confronto tra il numero medio di nodi e il numero medio di sottografi al variare di ρ . Dall'alto verso il basso si trovano i grafici relativi alle ABITAZIONI, alle SCUOLE e agli UFFICI.

duati in un piano di un edificio e numero di stanze in ogni gruppo. In altri casi può comunque essere più conveniente usare un approccio diverso.

L'intervallo $[0.35, 0.65]$ è quello in cui i tre dataset presentano i risultati più simili e che secondo noi rappresenta l'insieme di valori che possono essere associati al parametro.

Nel seguito delle prove sperimentali, abbiamo preferito utilizzare un valore di riferimento $\rho = 0.5$, comune a tutte e tre le tipologie edilizie. Anche se le abitazioni richiederebbero un valore più alto del parametro per mantenere le stesse proporzioni tra numero e dimensioni dei sottografi, questa scelta del parametro consente di avere dei sottografi di dimensioni simili a quelle di SCUOLE e UFFICI. Nel caso di $\rho = 0.7$, i sottografi delle abitazioni avrebbero mediamente circa 4 nodi, che potrebbe essere un valore basso per mantenere alcune caratteristiche strutturali utili in fase di campionamento.

Nella Tabella 6.1 sono mostrati i valori medi di numero di sottografi e dimensione dei sottografi con $\rho = 0.5$ per le tre tipologie edilizie.

	SCUOLE	UFFICI	ABITAZIONI
$\mathbb{E}[S]$	8.29	8.67	3.20
$\mathbb{E}[s]$	6.36	7.56	6.40

Tabella 6.1: Valori medi del numero di sottografi e delle dimensioni dei sottografi per $\rho = 0.5$.

I sottografi hanno una dimensione tipica compresa tra i 6 e gli 8 nodi. Nel caso delle ABITAZIONI, ogni piano viene diviso mediamente in 3 sot-

tografi, mentre negli UFFICI e nelle SCUOLE ogni piano è diviso in circa 8 sottografi. Oltre al fatto che le abitazioni sono meno strutturate in questo senso rispetto agli uffici pubblici, bisogna tenere conto del fatto che un piano di un'abitazione contiene tipicamente la metà delle stanze rispetto a quello di una scuola o di un ufficio.

6.2.2 Analisi del clustering dei sottografi

Il clustering (sezioni 4.4 e 5.2.1) rappresenta la fase computazionalmente più onerosa, perché la creazione del grafo di affinità richiede di effettuare l'allineamento tra ogni coppia di sottografi $(s_i, s_j) \in \mathcal{S}$, dell'intero dataset. Se n è il numero medio di nodi in un sottografo, la complessità di un allineamento è dell'ordine di $O(n^4)$; dato $m = |\mathcal{S}|$, la creazione del grafo di similarità richiede $O(m^2n^2)$ operazioni.

Nel corso della trattazione illustriamo in modo dettagliato i test sul clustering sulla tipologia edilizia SCUOLE. Per le altre tipologie edilizie l'approccio adottato è il medesimo.

Anche in questo caso abbiamo considerato l'impatto del parametro di ricorsione ρ_c su due misure concettualmente analoghe a quelle analizzate in fase di segmentazione:

- il numero di cluster,
- il numero medio di sottografi raggruppati in un cluster;

In Figura 6.8 è mostrato l'andamento del numero di cluster e delle dimensioni medie (numero di sottografi) di un cluster al variare del parametro di clustering ρ_c , nel caso della tipologia edilizia SCUOLE. I grafi sono stati campionati variando ρ_c con passo 0.025.

In questo caso, un'analisi di tipo quantitativa può essere sicuramente utile per escludere i valori del parametro non utilizzabili. Per stabilire i valori di ρ_c in modo più fine ci si può servire di un approccio di tipo visuale.

La visualizzazione dei cluster permette infatti di esprimere una valutazione qualitativa del processo di categorizzazione dei sottografi. In un caso ideale, ci si aspetterebbe che il numero di cluster sia il minimo necessario per avere una descrizione dei dati abbastanza dettagliata; si può ad esempio, effettuare uno studio di tipo qualitativo, incrementando il parametro ρ_c finché vi sono cluster di sottografi con, visivamente, poche caratteristiche in comune. Questo approccio è utile soprattutto per escludere i valori di

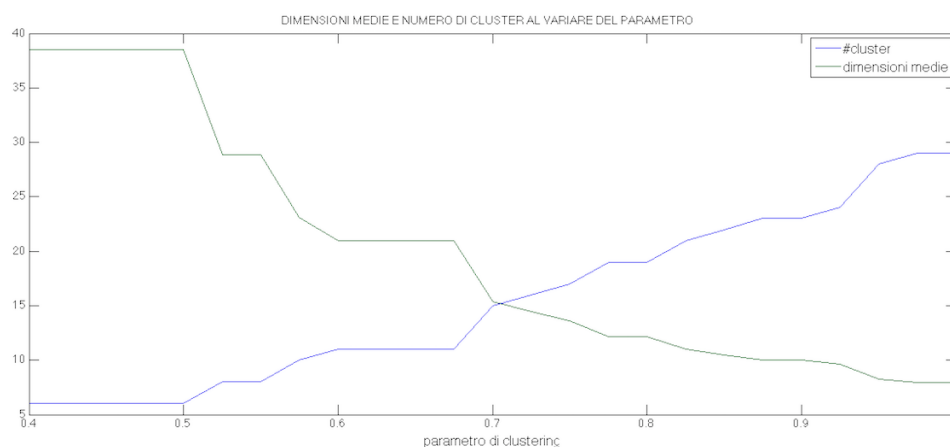


Figura 6.8: Confronto tra il numero medio di sottografi in un cluster (in verde) e il numero cluster (in blu) al variare di ρ_c . Dataset delle SCUOLE.

ρ_c troppo bassi, per i quali il numero di cluster non è sufficientemente descrittivo delle diverse categorie presenti.

Il cluster rappresentato in Figura 6.9 è un esempio di cluster non sufficientemente descrittivo, in quanto all'interno di esso si trovano anche sottografi molto diversi tra loro. Questo significa che il valore $\rho_c = 0.4$ è troppo basso.

In Figura 6.10 è rappresentato invece un cluster ottenuto con $\rho_c = 0.8$. In questo caso l'insieme dei sottografi viene diviso in 20 cluster, contenenti ciascuno in media 12 sottografi.

Concludendo, abbiamo rilevato che in generale l'intervallo $[0.55, 0.85]$ rappresenta l'insieme di valori del parametro ρ che possono essere utilizzati nel clustering per tutte le tre tipologie edilizie considerate.

Clustering senza utilizzare le label

Se si effettua il clustering su sottografi non etichettati, la creazione del grafo di affinità avviene solo considerando similarità topologiche. La conseguenza principale è che il processo clustering, a parità di parametro ρ , determina un numero di cluster decisamente inferiore rispetto al caso dei sottografi etichettati. In generale, abbiamo rilevato che se al parametro ρ vengono attribuiti valori inferiori a 1, non ottengono divisioni dell'insieme dei sottografi. Il motivo principale è rappresentato dal fatto che, da un punto di vista topologico, i sottografi ottenuti dalla segmentazione non presentano differenze significative: in particolare, la maggioranza dei

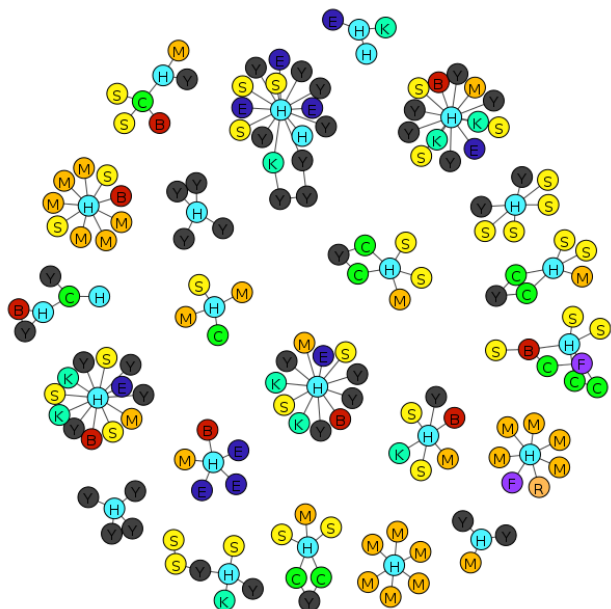


Figura 6.9: Esempio di cluster non rappresentativo, ottenuto con $\rho_c = 0.4$. Ad esempio, sono contenuti sia sottografi che rappresentano un insieme di classi (M) collegate a una hall (H), sia sottografi che rappresentano un insieme di segreterie/uffici (Y) collegati a una hall (H).

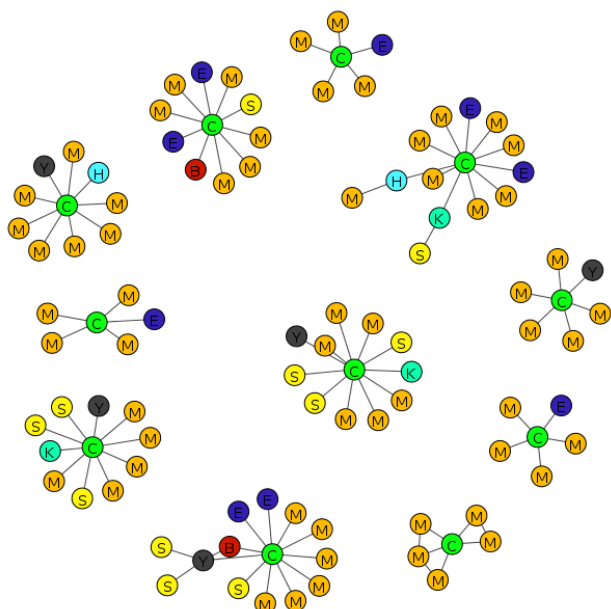


Figura 6.10: Esempio di cluster rappresentativo, ottenuto con $\rho_c = 0.8$. Sono rappresentati principalmente gruppi di classi (M) collegate a un corridoio (C).

sottografi presenta una struttura *a stella*, con una stanza di collegamento centrale a cui sono collegate tante stanze.

I risultati ottenuti in genere penalizzano un tipo di clustering dei sottografi esclusivamente topologico. Ad esempio in Figura 6.11 è mostrato un esempio di cluster ottenuto in questo modo, dove il labeling \mathcal{L} è stato applicato a posteriori. Il valore del parametro di clustering (topologico) con cui si ottengono i risultati migliori è $\rho_c = 1.1$.

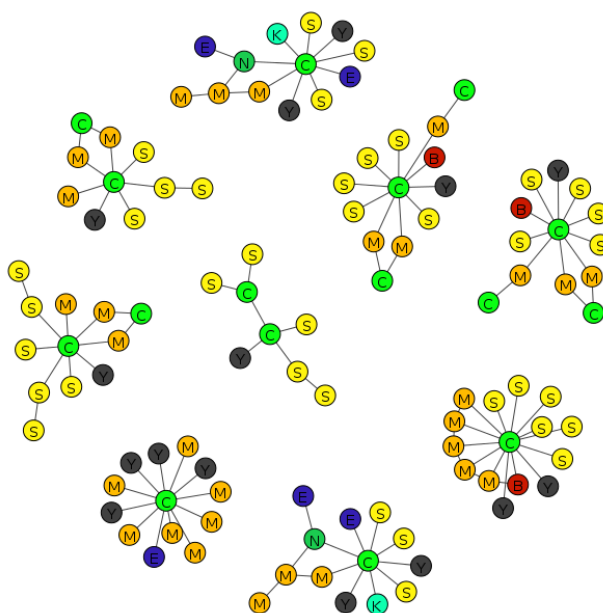


Figura 6.11: Esempio di clustering eseguito solo in base alla topologia, ottenuto con $\rho_c = 1.1$. In questo caso la caratteristica strutturale comune a questi sottografi è la struttura a stella. A livello di label vi sono delle differenze significative tra alcuni sottografi.

Possiamo affermare che l'utilizzo delle label semantiche rappresenta un beneficio significativo a livello di clustering, perché permette di identificare con maggiore facilità i sottografi che rappresentano parti di un edificio simili.

6.2.3 Valutazione del campionamento di Gibbs per le configurazioni dei cluster

La generazione di una nuova configurazione dei cluster su cui modellare un nuovo grafo è effettuata tramite il campionamento di Gibbs (introdotto nella Sezione 4.6 e implementato nella Sezione 5.2.4), un algoritmo derivato da una catena di Markov Monte Carlo.

Un aspetto da considerare quando si utilizza una catena di Markov è la valutazione del tempo richiesto per il mixing della catena, cioè il numero di step dopo i quali la probabilità di transizione tra uno stato A e uno stato B del sistema si avvicina a una distribuzione stazionaria. Dal momento che non esiste un metodo univoco efficace per verificare in maniera definitiva il mixing di un campionamento di Gibbs [29], si può comunque stimare un indicatore che verifichi il numero di passi dopo i quali un campione ha una funzione di transizione stazionaria.

Il problema più comune che impedisce un corretto mixing emerge quando lo spazio degli stati consiste di diverse regioni che sono connesse solo da poche transizioni, con una bassa probabilità. Se la catena inizia in uno stato appartenente a una di queste regioni, è probabile che venga spesa una certa quantità di tempo in quella regione di stati prima di effettuare una transizione in un'altra regione. Intuitivamente, gli stati campionati nella fase iniziale non provengono dalla distribuzione stazionaria, poiché sono fortemente correlati con lo stato iniziale, che è arbitrario. In ogni caso, più avanti nel corso della traiettoria, si potrebbe raggiungere uno stato che, verosimilmente, potrebbe essere stato originato da qualunque stato iniziale. In questo caso, è legittimo considerare che è avvenuto il mixing della catena.

Per verificare il mixing dei campionatori di Gibbs abbiamo utilizzato una tecnica, tratta da [29], basata sull'osservazione che catene multiple che campionano dalla stessa distribuzione dovrebbero, una volta giunte a convergenza, fornire stime simili. Questa tecnica è descritta nell'Algoritmo 5. Le stime basate su un insieme completo di campioni collezionati da tutte le catene dovrebbero avere una varianza comparabile a quella delle catene. Più formalmente, si assuma che K catene separate siano eseguite ciascuna per $T + M$ iterazioni, partendo ognuna da uno stato iniziale diverso. Dopo aver scartato i primi T campioni da ogni catena, sia $\mathbf{X}_k[m]$ un campione della catena k dopo l'iterazione $T + m$. È quindi possibile calcolare le varianze B di campioni appartenenti a catene diverse e W di campioni appartenenti alla stessa catena.

V sovrastima la varianza della stima di f basata sui campioni calcolati. Per $M \rightarrow \infty$, sia W che V convergono alla varianza della stima. \hat{R} rappresenta la misura di disaccordo tra le catene. Se $\hat{R} \simeq 1$, è possibile affermare con una buona approssimazione che le catene hanno raggiunto la convergenza alla distribuzione dei dati.

Abbiamo testato il metodo riportato nell'Algoritmo 5 eseguendo $K = 100$ campionamenti di Gibbs separati su un insieme di 20 configurazio-

<p>Input: Gli step $\mathbf{X}_k[m]$ compresi tra T e T+M di un insieme finito di K catene</p> <p>Output: La misura di disaccordo tra le chains \hat{R}</p> <p>1 for $k \leftarrow 1$ to K do</p> <p style="padding-left: 2em;"> $\bar{f}_k \leftarrow \frac{1}{M} \sum_{m=1}^M f(\mathbf{X}_k[m])$</p> <p style="padding-left: 2em;">end</p> <p>2 $\bar{f} \leftarrow \frac{1}{K} \sum_{k=1}^K \bar{f}_k$</p> <p>3 $B \leftarrow \frac{M}{K^2-1} \sum_{k=1}^K (\bar{f}_k - \bar{f})^2$</p> <p>4 $W \leftarrow \frac{1}{K} \frac{1}{M-1} \sum_{k=1}^K \sum_{m=1}^M (f(\mathbf{X}_k[m]) - \bar{f}_k)^2$</p> <p>5 $V \leftarrow \frac{M-1}{M} W + \frac{1}{M} B$</p> <p>6 $\hat{R} \leftarrow \sqrt{\frac{V}{W}}$</p> <p>return \hat{R}</p>
--

Algoritmo 5: Algoritmo per controllare se il numero di step è sufficiente per il mix delle catene, tratto da [29].

ni dei cluster, ciascuna di lunghezza $|\mathbf{v}| = 20$. Per ciascun campionamento abbiamo eseguito $T + M = 200$ iterazioni, scartando le prime $T = 100$, e considerando le restanti $M = 100$ iterazioni. Come funzione di valutazione $f(\mathbf{X}_k[m])$ abbiamo utilizzato la funzione $H_{\mathbf{v}_a} = \sum_{\ell=1}^{|\mathcal{G}|} e^{\vartheta \cdot H(\mathbf{v}_a, q_\ell)}$, $f(\mathbf{X}_k[m]) = H_{\mathbf{v}_a}$, dove H è la distanza di Hamming tra il campione $\mathbf{X}_k[m] = \mathbf{v}_a$ e la configurazione dei cluster q_ℓ di un grafo G_ℓ . È stato calcolato un valore $\hat{R} = 1.001$.

Possiamo quindi considerare $I_2 = 200$ un numero di iterazioni sufficiente affinché le configurazioni dei cluster campionate seguano la stessa distribuzione dei dati. Se si vuole effettuare un controllo più robusto, si potrebbero usare diverse funzioni di valutazione, o eseguire il test su diversi insiemi di dati.

6.3 La riconnessione dei sottografi

Tra tutte le fasi del metodo, la riconnessione dei sottografi campionati è quella che ha richiesto il maggior numero di adattamenti. L'ostacolo principale è rappresentato dal riuscire a modellare la distribuzione delle

connessioni tra i sottografi in modo computazionalmente efficiente e che permetta di riconnettere i sottografi in modo speculare rispetto ai tagli praticati con Ncut. Se ad esempio il risultato finale del metodo generativo è un grafo dove pochi sottografi sono estremamente connessi tra loro e tutti gli altri non sono connessi a niente, il modello di distribuzione delle connessioni non è aderente a quello dei dati. Nei grafi iniziali \mathcal{G} e nei grafi campionati \hat{G} bisogna considerare le connessioni tenendo traccia sia dei cluster sia dei nodi.

Nel Capitolo 4 sono stati discussi in modo approfondito i limiti dell'approccio di [71, 73] che ci hanno portato a sviluppare un modello diverso. Oltre alle problematiche già discusse legate al modello, ve ne è un'altra che limita in maniera significativa le realizzazioni sperimentali, in particolare nell'ottica di un utilizzo online del sistema: la complessità. In particolare, un aspetto particolarmente inefficiente del metodo proposto da [71, 73] è legato alla decomposizione spettrale della matrice quadrata di covarianza $\Sigma = \Phi \Lambda \Phi^T$ che, dal momento che l'ordine lessicografico tra ogni coppia di nodi imposto con l'allineamento è considerato una variabile aleatoria, ha dimensione $d \approx n^2 c^2$, dove n è il numero di nodi del sottografo più grande, e c è il numero dei cluster. Questo aspetto, all'atto pratico, rende questa operazione estremamente inefficiente, in quanto anche un valore accettabile di n pari a 20 e di c pari a 20, come nel nostro caso, non riesce ad essere trattato efficacemente senza una configurazione specifica e una fase di ottimizzazione computazionale, richiedendo la diagonalizzazione di una matrice quadrata 40000×40000 . In particolare, le dimensioni in nodi del sottografo più grande e il numero di cluster limitano l'efficienza della decomposizione spettrale, rendendo in alcuni casi pratico aumentare ρ in modo tale che il sottografo più grande avesse un numero di nodi inferiore a 15, e diminuendo ρ_c in modo da avere un numero basso di cluster. Ovviamente questa pratica danneggia le fasi di segmentazione e clustering che, come è stato visto precedentemente nel capitolo, devono basarsi su valori dei parametri entro un certo intervallo. Inoltre, per avere una matrice di covarianza ben definita tra $|x|$ variabili aleatorie, servono almeno $|x| + 1$ campioni; nella nostra applicazione abbiamo rilevato $|x| \approx 42000$. Anche effettuando una ottimizzazione computazionale della decomposizione spettrale di una matrice di grandi dimensioni, vi è un problema legato all'indisponibilità attuale di un dataset contenente migliaia di grafi rappresentanti interi piani di edifici, che non permette la costruzione di una matrice di covarianza ben definita. Quest'ultimo aspetto peraltro preclude anche l'utilizzo di altri tipi di decomposizione più efficienti quali la decomposizione di Cholesky ($\Sigma = LL^T$), che può essere effettuata solo

se Σ è definita positiva, mentre, se il numero di campioni è inferiore al numero di variabili, Σ è solo semidefinita positiva.

Per risolvere il problema abbiamo scelto un tipo di rappresentazione che riduce le dimensioni di Σ : nel nostro metodo, ogni elemento della matrice Σ rappresenta la covarianza tra i gradi di connessione tra due coppie di cluster, riducendo la dimensione della matrice a $d' \approx c^2$. L'eliminazione del termine n^2 richiede $|x'| + 1 \approx 100$ campioni e permette di effettuare la decomposizione spettrale in pochi secondi.

La discretizzazione di \hat{c}

Il processo di discretizzazione del vettore \hat{c} contenente i gradi di connessione dei sottografi campionati, è un aspetto molto importante del processo di riconnessione.

Con il parametro di segmentazione ρ è fissato a 0.5, abbiamo rilevato che, dati due sottografi, il massimo numero di lati tagliati da Ncut è 2. Questo significa che il grado di connessione tra due sottografi può essere 0, 1 o 2. Nella Tabella 6.2 sono mostrate le probabilità che, presi due sottografi arbitrari ottenuti dal dataset SCUOLE, questi abbiano un grado di connettività pari a X .

	$X = 0$	$X = 1$	$X = 2$
$p(c_{s_i, s_j} = X)$	0.82	0.16	0.02

Tabella 6.2: Valori delle probabilità che il grado di connessione c_{s_i, s_j} tra due sottografi arbitrari s_i, s_j sia X , avendo effettuato la segmentazione dei grafi con $\rho = 0.5$.

Un aspetto che emerge è la percentuale molto alta di sottografi non collegati tra loro ($p(c_{s_i, s_j} = 0) = 0.82$). Questo fornisce implicitamente un'altra informazione, e cioè che in genere un sottografo è collegato a un numero ridotto di sottografi: un sottografo individuato da Ncut è, il più delle volte, connesso da un solo lato verso uno solo fra tutti gli altri sottografi. In altre parole, astraendo da un grafo G un grafo \hat{G} i cui nodi sono i sottografi di G , il numero di lati tagliati $|E|$ tende al minimo necessario affinché \hat{G} sia un grafo connesso, aspetto che prova l'efficacia della segmentazione di Ncut. In genere, un sottografo campionato dovrebbe avere un numero di connessioni tra i sottografi campionati \hat{S} che permetta di ottenere un unico grafo

connesso, ma allo stesso modo i dati mostrano che il numero di connessioni dovrebbe essere il minimo che soddisfi questo requisito.

Definire i parametri di thresholding β_1, β_2 (definiti nella Sezione 4.8.1) con cui discretizzare \hat{c} in modo da ottenere una distribuzione di p simile a quella indicata in Tabella 6.2 è un processo delicato, perché a seconda del campionamento dei sottografi e delle connessioni a parità di parametro si possono ottenere risultati molto diversi. Un esempio di questo problema è mostrato in Figura 6.12.

La soluzione che abbiamo proposto consiste in un processo di riconnessione incrementale:

- inizialmente connettono i grafi con un thresholding di $\beta_1 = 0.3$, più stringente, in modo da connettere solo le coppie di sottografi con un valore $\hat{c}(\hat{s}_i, \hat{s}_j)$ alto;
- i sottografi \hat{s}_i che non sono stati connessi almeno una volta nel primo step sono collegati al sottografo \hat{s}_j per cui $\hat{c}(\hat{s}_i, \hat{s}_j) = \max(\hat{c}(\hat{s}_i, \hat{s}_x)), \forall \hat{s}_x \in \hat{S}$.

In Figura 6.13 è rappresentato un esempio di applicazione di questa euristica. In questo modo otteniamo un risultato vicino a quello desiderato, cercando di rimanere il più coerenti possibili ai risultati del campionamento dei gradi di connettività \hat{c} .

6.4 Valutazione complessiva del sistema

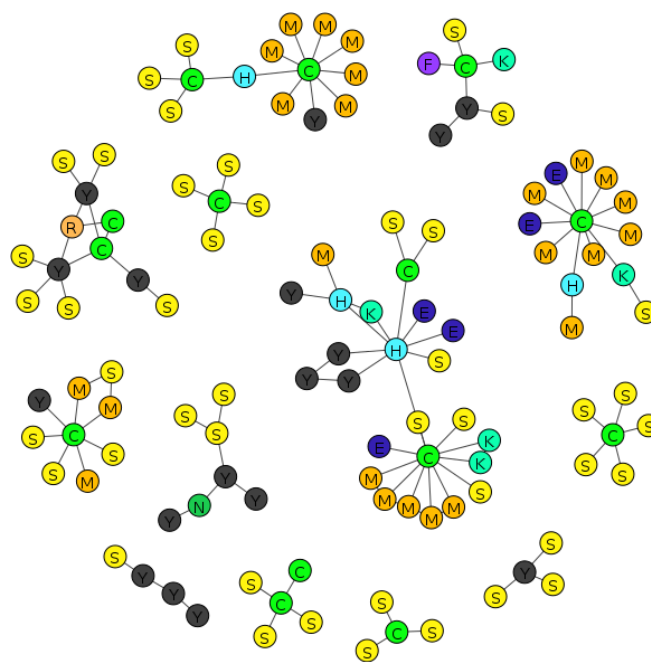
In questa sezione valutiamo il sistema nel suo complesso, ovvero verificando che i grafi generati con il campionamento presentino delle caratteristiche simili a quelli originali. Per fare ciò, abbiamo utilizzato un insieme di metriche che esprimano in modo quantitativo alcune proprietà dei grafi. Il confronto tra tali metriche rispetto a un dataset di grafi \mathcal{G} e ai grafi campionati \hat{G} viene poi analizzato per determinare similitudini e differenze.

6.4.1 Le metriche utilizzate

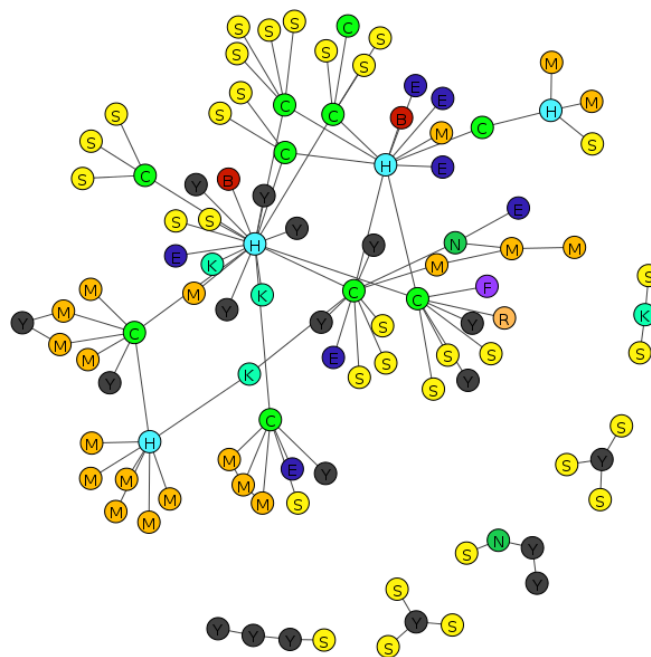
Utilizziamo due tipi di metriche: un primo tipo più elementare riguardante alcune caratteristiche generali dei grafi, e un secondo tipo che studia l'importanza e il ruolo dei nodi attraverso il concetto di *centralità*.

Metriche generali

- **Numero di nodi:** il numero di nodi di un grafo G . Lo indichiamo con $|G|$.



(a)



(b)

Figura 6.12: Un esempio di due grafi ricollegati con $\beta_1 = 0.3, \beta_2 = 0.7$. Si nota che, nonostante gli parametri siano gli stessi, il grafo in (b) sia molto più connesso del grafo (a).

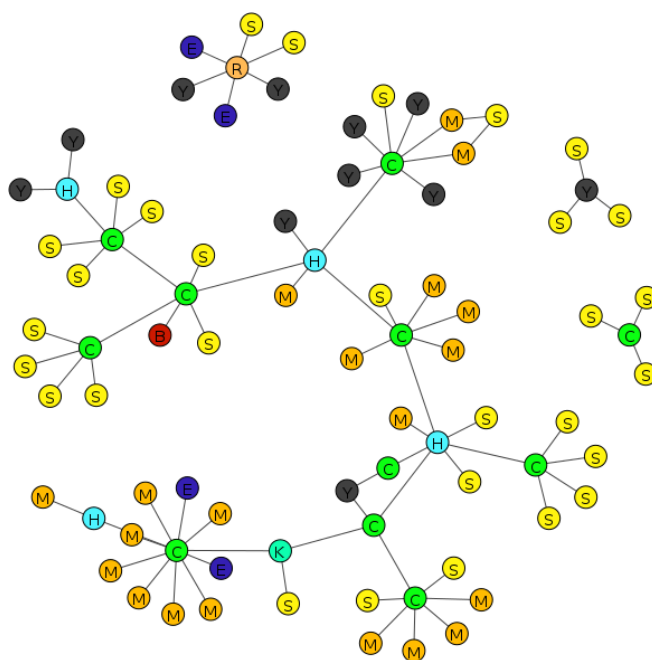
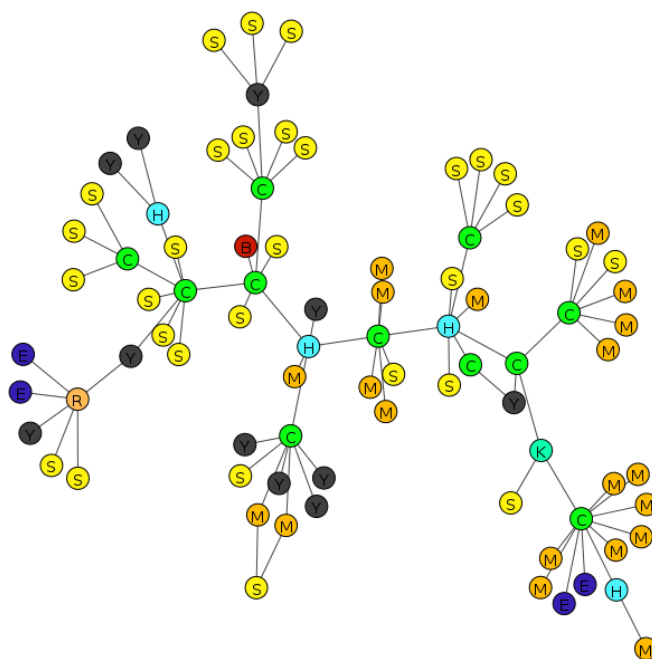
(a) Grafo \hat{G} dopo il primo step.(b) Grafo \hat{G} dopo il secondo step.

Figura 6.13: Un esempio di grafo ottenuto tramite un processo di riconnessione incrementale: inizialmente (a) vengono connessi i sottografi principali con $\beta_1 = 0.3, \beta_2 = 0.7$. In seguito (b) i sottografi \hat{s}_i rimasti non connessi sono collegati al sottografo \hat{s}_j per cui il valore di $\hat{c}(\hat{s}_i, \hat{s}_j)$ è massimo (b).

- **Numero di nodi stanza:** il numero di nodi che rappresentano una STANZA secondo lo schema di label generico $\bar{\mathcal{L}}$ (STANZA PICCOLA, STANZA MEDIA o STANZA GRANDE secondo lo schema di label standard \mathcal{L}). Lo indichiamo con R .
- **Numero di nodi corridoio:** il numero di nodi che rappresentano un CORRIDOIO, secondo lo schema di label generico $\bar{\mathcal{L}}$ (CORRIDOIO, HALL o INGRESSO secondo lo schema di label standard \mathcal{L}). Lo indichiamo con C .
- **Lunghezza di un path:** la distanza media per il percorso più breve tra un nodo n_i e un nodo n_j . La indichiamo con p_{len} .
- **Diametro:** la distanza massima tra due nodi. Lo indichiamo con d .

Metriche sulla centralità dei nodi

Gli indici di centralità sono funzioni che associano un valore reale a ogni nodo di un grafo. Tali valori servono a esprimere quali sono i nodi più importanti [9].

A seconda del concetto di importanza considerato, esistono diverse definizioni di centralità. Abbiamo utilizzato misure di centralità per testare il ruolo di ogni nodo nei grafi originali e in quelli generati. Questo ci permette di verificare se la struttura funzionale di un grafo campionato è consistente con quella di un edificio reale.

Dal momento che la centralità è definita per ogni nodo, valutiamo la media μ e la deviazione standard σ su un insieme di nodi (tutti i nodi, solo i nodi STANZA, solo i nodi CORRIDOIO).

In questo caso abbiamo considerato tre misure di centralità fra le più utilizzate:

- La BETWEENNESS di ogni nodo n è definita come il numero di percorsi più brevi tra due nodi $s \neq t (\neq n)$ che attraversano n :

$$\text{BETWEENNESS}(n) = \sum_{(s,t) \in N, s \neq n \neq t} \frac{\tau_{st}(n)}{\tau_{st}}$$

dove τ_{st} è il numero totale di percorsi più brevi da s a t e $\tau_{st}(n)$ è il numero di questi percorsi che attraversano n . Questa metrica è normalizzata di un fattore $(|N| - 1)(|N| - 2)$.

La BETWEENNESS quantifica il numero di volte in cui un nodo è compreso in uno dei percorsi più brevi tra una coppia di nodi del grafo.

- La CLOSENESS di ogni nodo n considera la distanza più breve tra n e ogni altro nodo del grafo:

$$\text{CLOSENESS}(n) = \sum_{t \in N \setminus \{n\}} 2^{-d_G(n,t)}$$

dove $d_G(n, t)$ è la lunghezza del percorso più breve tra i due nodi n e t .

La CLOSENESS di un nodo n può essere considerata come una misura di quanto rapidamente ci si può muovere da n verso tutti gli altri nodi.

- Infine, la centralità di un autovettore EIG_CEN assegna un punteggio relativo a tutti i nodi nel grafo basato sul concetto che le connessioni verso nodi con un punteggio alto contribuiscono di più al punteggio del nodo rispetto alle connessioni verso nodi con un punteggio basso. È calcolata come:

$$\text{EIG_CEN}(n) = \frac{1}{\lambda} \sum_{t \in N} m_{t,n} \text{EIG_CEN}(k)$$

dove $\lambda \neq 0$ è una costante e $m_{t,n}$ è l'elemento (t, n) della matrice di adiacenza M . Il vettore di centralità EIG_CEN è l'autovettore \mathbf{x} della matrice di adiacenza \mathbf{M} associato all'autovalore λ : $\mathbf{M}\mathbf{x} = \lambda\mathbf{x}$.

6.4.2 Risultati sperimentali

Presentiamo ora i valori delle metriche calcolati nei casi delle tre tipologie edilizie. Per ogni metrica sono presentati la media μ e la deviazione standard σ (nel formato $\mu (\sigma)$), sia per il dataset \mathcal{G} che per l'insieme dei grafi campionati \hat{G} . Le metriche di centralità sono state considerate anche separatamente, considerando solo i nodi che appartengono alla categoria STANZA (CORRIDOIO) per lo schema di label generico \mathcal{L} , indicando i due sottoinsiemi di nodi \mathcal{G}_R e \hat{G}_R (\mathcal{G}_C e \hat{G}_C).

In Tabella 6.3 sono mostrati i risultati relativi alla tipologia edilizia SCUOLE. In questo caso $|\mathcal{G}|$ contiene 28 grafi originali, mentre \hat{G} è costituito da 100 grafi campionati. I parametri di segmentazione e di clustering

	$ G $	$ R $	$ C $	p_{len}	d
\mathcal{G}	52.0 (21.0)	41.2 (17.1)	10.8 (4.7)	3.7 (0.8)	7.0 (2.3)
\hat{G}	70.7 (30.0)	56.8 (24.8)	13.9 (6.0)	4.5 (0.9)	9.6 (2.4)
	CLOSENESS	BETWEENNESS	EIG_CEN		
\mathcal{G}	0.2195 (0.1094)	0.0276 (0.0140)	0.1962 (0.0993)		
\mathcal{G}_R	0.2078 (0.1010)	0.0038 (0.0030)	0.1486 (0.0856)		
\mathcal{G}_C	0.2630 (0.1431)	0.1171 (0.0584)	0.3818 (0.1546)		
\hat{G}	0.2264 (0.0624)	0.0286 (0.0103)	0.1643 (0.0666)		
\hat{G}_R	0.2152 (0.0571)	0.0069 (0.0050)	0.1192 (0.0481)		
\hat{G}_C	0.2723 (0.0831)	0.1191 (0.0467)	0.3468 (0.1391)		

Tabella 6.3: Confronto tra il dataset \mathcal{G} e i grafi campionati \hat{G} per la tipologia edilizia SCUOLE.

utilizzati sono $\rho = 0.5$ e $\rho_c = 0.6$, rispettivamente.

In questo caso si sono ottenuti dei risultati molto simili per quanto riguarda i valori di centralità: un aspetto significativo riguarda la differenza tra il valor medio di BETWEENNESS tra i nodi CORRIDOIO e i nodi STANZA, che si mantiene nei grafi campionati \hat{G} . Questi risultati sono in accordo rispetto alla funzione di collegamento tra altre stanze svolta dai corridoi.

Per quanto riguarda le differenze, si nota che i grafi campionati \hat{G} hanno un numero di nodi medio superiore rispetto ai grafi del dataset \mathcal{G} . In ogni caso il valore piuttosto elevato di σ rivela una certa disomogeneità di cardinalità dei nodi. In \hat{G} p_{len} mantiene comunque un valore relativamente basso di 4.5, che significa che un robot per raggiungere qualsiasi altra stanza ne deve attraversare mediamente altre 3 o 4 oltre a quella di destinazione.

In Tabella 6.4 sono mostrati i risultati relativi alla tipologia edilizia UFFICI. In questo caso $|\mathcal{G}|$ contiene 30 grafi originali, mentre \hat{G} è costituito da 100 grafi campionati. I parametri di segmentazione e di clustering utilizzati sono $\rho = 0.5$ e $\rho_c = 0.8$, rispettivamente.

Per la tipologia edilizia UFFICI valgono molte considerazioni fatte per le SCUOLE: le metriche di centralità sono concordi tra \mathcal{G} e \hat{G} , con una differenza significativa tra un nodo STANZA e un nodo CORRIDOIO. La cardinalità dei nodi è più alta in \hat{G} ma, nonostante questo, viene mantenuta una lunghezza media di un percorso e un diametro poco superiori rispetto al dataset \mathcal{G} .

	$ G $	$ R $	$ C $	$plen$	d
\mathcal{G}	48.6 (27.4)	34.9 (19.9)	13.7 (8.5)	3.5 (0.9)	6.6 (2.2)
\hat{G}	70.2 (32.9)	50.6 (25.1)	19.7 (9.5)	4.2 (0.8)	8.2 (2.0)
	CLOSENESS	BETWEENNESS	EIG_CEN		
\mathcal{G}	0.2203 (0.1484)	0.0329 (0.0259)	0.2299 (0.1194)		
\mathcal{G}_R	0.2061 (0.1355)	0.0056 (0.0131)	0.1666 (0.1125)		
\mathcal{G}_C	0.2576 (0.1841)	0.1067 (0.0759)	0.4035 (0.1685)		
\hat{G}	0.2319 (0.0840)	0.0280 (0.0182)	0.1907 (0.0853)		
\hat{G}_R	0.2173 (0.0734)	0.0035 (0.0039)	0.1263 (0.0746)		
\hat{G}_C	0.2662 (0.0972)	0.0891 (0.0430)	0.3519 (0.1118)		

Tabella 6.4: Confronto tra il dataset \mathcal{G} e i grafi campionati \hat{G} per la tipologia edilizia UFFICI.

In Tabella 6.5 sono mostrati i risultati relativi alla tipologia edilizia ABITAZIONI. In questo caso $|G|$ contiene 73 grafi originali, mentre \hat{G} è costituito da 100 grafi campionati. I parametri di segmentazione e di clustering utilizzati sono $\rho = 0.5$ e $\rho_c = 0.9$, rispettivamente.

	$ G $	$ R $	$ C $	$plen$	d
\mathcal{G}	19.2 (9.7)	11.8 (6.9)	7.4 (4.2)	2.7 (0.9)	5.2 (2.3)
\hat{G}	44.2 (17.5)	28.5 (12.6)	15.8 (5.8)	4.0 (0.7)	8.4 (2.1)
	CLOSENESS	BETWEENNESS	EIG_CEN		
\mathcal{G}	0.2420 (0.1560)	0.0485 (0.0348)	0.3408 (0.1413)		
\mathcal{G}_R	0.2240 (0.1404)	0.0129 (0.0192)	0.2727 (0.1323)		
\mathcal{G}_C	0.2765 (0.1931)	0.1121 (0.0937)	0.4630 (0.1990)		
\hat{G}	0.2198 (0.0883)	0.0380 (0.0174)	0.2157 (0.0817)		
\hat{G}_R	0.2059 (0.0791)	0.0075 (0.0067)	0.1554 (0.0673)		
\hat{G}_C	0.2445 (0.1038)	0.0915 (0.0402)	0.3231 (0.1105)		

Tabella 6.5: Confronto tra il dataset \mathcal{G} e i grafi campionati \hat{G} per la tipologia edilizia ABITAZIONI.

I risultati sono buoni, in particolare per SCUOLE e UFFICI. La tipologia ABITAZIONI invece, pur avendo risultati buoni, risulta essere meno omogenea rispetto alle altre. In particolare, vi sono differenze significative nel numero di nodi, (più che doppio in \hat{G} rispetto a \mathcal{G}), nella lunghezza media di un

percorso p_{len} , nel diametro d e nella metrica di centralità di un autovettore EIG_CEN. Anche questi valori possono però essere considerati corretti dal momento che gli intervalli $[\mu - \sigma, \mu + \sigma]$ non sono disgiunti a quelli di \hat{G} .

La differenza di performance tra le abitazioni e altre tipologie è attribuibile ad alcune caratteristiche intrinseche alla tipologia edilizia ABITAZIONI; considerando che un'abitazione ha un numero di stanze tipicamente ridotto, fare il training di un modello generativo utilizzando, come in questo caso, anche grafi molto elementari potrebbe risultare poco significativo. D'altra parte, se le tipologie UFFICI e, soprattutto, SCUOLE presentano delle maggiori simmetrie e regolarità nelle strutture, nel caso di edifici ad uso abitativo per astrarre questo tipo di informazioni bisognerebbe effettuare un'ulteriore suddivisione della tipologia ABITAZIONI in tipologie più specifiche (ad esempio, distinguendo una villa da un appartamento) o aumentare il numero di label utilizzate.

Valutazione generale dei risultati

Possiamo considerare i risultati ottenuti nel complesso validi, dal momento che i valori calcolati delle metriche rientrano nella media statistica. L'aspetto che emerge maggiormente in tutte le tre tipologie edilizie è la differenza tra la centralità dei nodi CORRIDOIO e quella dei nodi STANZA. Il fatto che i grafi campionati \hat{G} presentino anch'essi questa caratteristica è molto importante perché indica che il modello generativo rappresenta correttamente questo tipo di informazione.

Una discrepanza che abbiamo riscontrato è il maggior numero di nodi nei grafi in \hat{G} rispetto ai dataset \mathcal{G} . Siccome questa tendenza è comune a tutte le tipologie edilizie, abbiamo approfondito questo aspetto analizzando il numero medio di sottografi $|S|$ e il numero medio di nodi per sottografo $|s|$ sia nei dataset \mathcal{G} sia nei grafi campionati \hat{G} . I risultati, mostrati in Tabella 6.6, indicano che la differenza più significativa riguarda il numero di sottografi campionati, la cui media è più alta rispetto ai grafi iniziali.

Le cause di questa differenza potrebbero essere due, eventualmente concomitanti:

- il campionamento di Gibbs delle configurazioni dei cluster dei sottografi permette anche la generazione di configurazioni con un numero elevato di sottografi;
- le variazioni significative del numero di stanze all'interno di una stessa tipologia edilizia (testimoniate dal valore relativamente alto di $\sigma(|S|)$)

	$ S $	$ s $
$\mathcal{G}_{\text{SCUOLE}}$	8.29 (3.48)	6.27 (2.94)
$\mathcal{G}_{\text{UFFICI}}$	7.13 (3.77)	6.80 (2.97)
$\mathcal{G}_{\text{ABITAZIONI}}$	3.20 (1.82)	5.92 (2.29)
$\hat{\mathcal{G}}_{\text{SCUOLE}}$	11.53 (4.91)	6.10 (0.81)
$\hat{\mathcal{G}}_{\text{UFFICI}}$	10.09 (4.42)	6.89 (1.17)
$\hat{\mathcal{G}}_{\text{ABITAZIONI}}$	7.93 (2.81)	3.20 (0.81)

Tabella 6.6: Confronto del numero medio di sottografi $|S|$ e del numero medio di nodi per sottografo $|s|$ tra il dataset \mathcal{G} e i grafi campionati $\hat{\mathcal{G}}$. Di ogni valore è indicata la media μ e la deviazione standard σ .

non consentono di astrarre in modo preciso la distribuzione del numero di stanze di un edificio appartenente a una certa tipologia.

6.4.3 Esempio di grafo campionato

Un aspetto che va considerato ai fini dell'utilizzo predittivo dei grafi campionati $\hat{\mathcal{G}}$ è la rappresentatività dei grafi generati con questo metodo rispetto alla tipologia edilizia considerata. Se infatti un grafo appartenente a un dataset \mathcal{G} è un tipo di rappresentazione di un edificio in cui ogni nodo etichettato è una stanza e ogni lato è un passaggio diretto tra due stanze (tipicamente, una porta), su un grafo campionato $\hat{\mathcal{G}}$ dovrebbe essere possibile interpretare i nodi e i lati come se il grafo rappresentasse effettivamente una scuola.

In Figura 6.14 è mostrato un esempio di un grafo $\hat{\mathcal{G}}$ campionato dal dataset SCUOLE. Si può osservare come alcune caratteristiche strutturali di una scuola reale siano mantenute. Nel riquadro in nero vi è l'entrata principale dell'edificio, con una stanza INGRESSO (indicata con la lettera E) connessa a una HALL (H), che si affaccia su numerose stanze e sui corridoi principali, che nel complesso costituiscono la struttura caratterizzante dell'edificio. Gruppi di classi (riquadri in blu) connesse a un corridoio comune rappresentano le aree destinate alla didattica. Vi sono alcune stanze di tipo SEGRETERIA/SALA PROFESSORI (riquadro in verde, etichettate con Y), principalmente situate vicino alle hall per rendere più facile l'individuazione da parte di persone esterne. In rosso sono invece segnalate le aree con i servizi igienici. Per finire, in fucsia sono evidenziate due aree con una stanza grande di tipo AUDITORIUM, etichettata con F, che può rappresentare una aula magna o una stanza attrezzata per attività didattiche di gruppo.

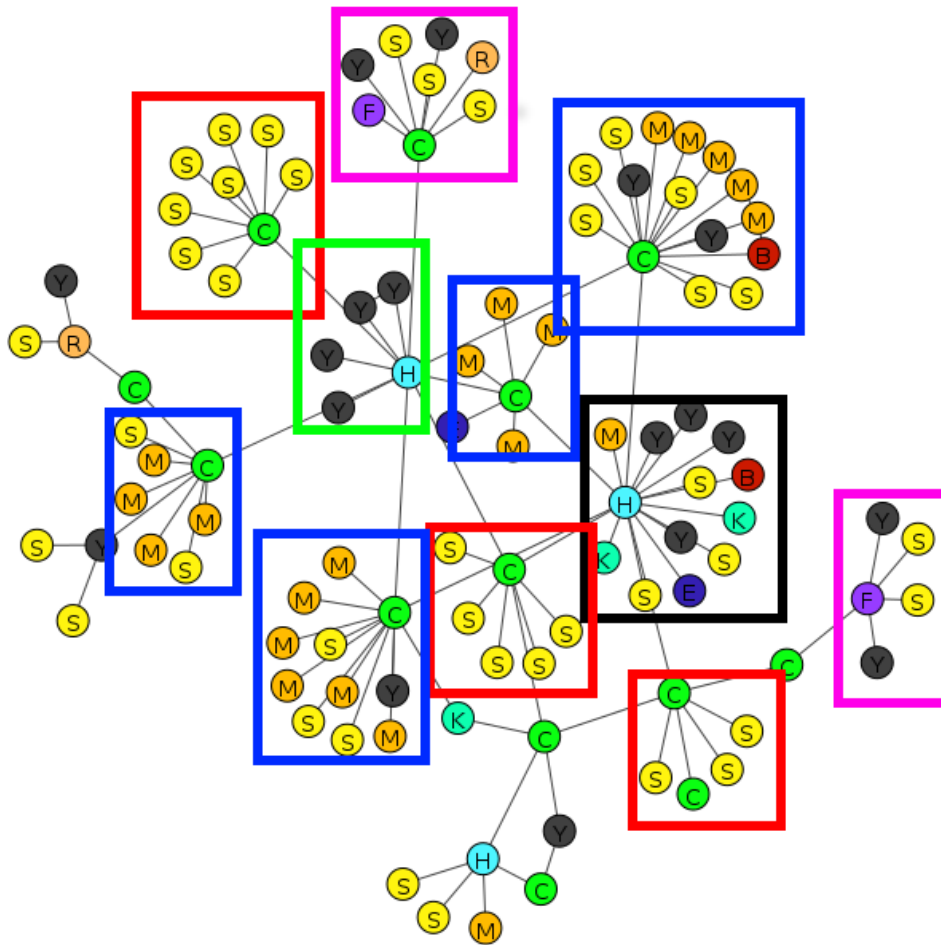


Figura 6.14: Esempio di un grafo \hat{G} campionato dal dataset SCUOLE. In questo caso la struttura è compatibile con quella di una scuola. I colori dei riquadri identificano un tipo di zona interpretabile in termini di una scuola reale: in nero l'ingresso con l'atrio, in blu le zone per la didattica, in verde la zona amministrativa, in rosso i servizi igienici e in fucsia le zone per conferenze e attività di gruppo.

Capitolo 7

Direzioni future di ricerca e conclusioni

Aggiungo: infinita. Non introduco quest'aggettivo per un'abitudine retorica; dico che non è illogico pensare che il mondo sia infinito. Chi lo giudica limitato, suppone che in qualche luogo remoto i corridoi e le scale e gli esagoni possano inconcepibilmente cessare; ciò che è assurdo. Chi lo immagina senza limiti, dimentica che è limitato il numero possibile dei libri. Io m'arrischio a insinuare questa soluzione: La Biblioteca è illimitata e periodica. Se un eterno viaggiatore la traversasse in una direzione qualsiasi, constaterrebbe alla fine dei secoli che gli stessi volumi si ripetono nello stesso disordine (che, ripetuto, sarebbe un ordine: l'Ordine). Questa elegante speranza rallegra la mia solitudine.

Jorge Luis Borges, *La Biblioteca di Babele*

In questo lavoro di tesi è stato sviluppato un modello generativo basato su grafi indirizzato all'ambito specifico del mapping semantico nella robotica mobile autonoma.

Inizialmente è stato considerato il problema, nel campo del mapping semantico, relativo alla predizione di parti di un edificio non ancora esplorate. L'adozione di un modello generativo che rappresenti in maniera analitica le proprietà strutturali di un insieme di dati iniziali, in modo tale da generare nuove istanze di dati aventi le medesime proprietà, può costituire una soluzione al problema.

In particolare, il modello generativo realizzato in questo lavoro rappresenta i piani di edifici come grafi non orientati con label associate ai nodi, indicanti le stanze e i collegamenti tra esse.

Il modello proposto è incentrato su tre fasi fondamentali basate sulla teoria spettrale dei grafi: segmentazione, clustering e campionamento.

La fase di *segmentazione* consiste nella divisione di ogni grafo in un insieme di sottografi, che rappresentano i componenti elementari del modello, non divisibili ulteriormente.

La fase di *clustering* consiste nel raggruppamento in cluster di sottografi considerati simili in base a caratteristiche legate alle label e alla topologia.

La fase di *campionamento* preleva alcuni sottografi dai cluster, secondo una certa distribuzione coerente con i dati iniziali, e li riconnette tra loro formando un nuovo grafo.

Il sistema utilizza diversi algoritmi, alcuni presi dalla letteratura e altri realizzati *ad hoc*, al fine di costruire un modello accurato dei dati astraendo il maggior numero possibile di informazioni. Allo stesso modo, il campionamento dei nuovi grafi avviene attraverso una serie di passaggi utilizzando diversi tipi di distribuzione con cui approssimare quella dei dati reali.

La verifica del sistema è stata effettuata utilizzando tre dataset distinti, ciascuno dei quali contiene grafi di edifici appartenenti a una specifica tipologia edilizia, nel caso in esame SCUOLE, UFFICI e ABITAZIONI.

I risultati mostrano che i grafi campionati dal modello condividono effettivamente molte caratteristiche con gli edifici originali usati per costruirlo. Inoltre, è stata rilevata una maggiore accuratezza dei risultati nel caso delle scuole e degli uffici, confermando l'ipotesi che edifici di grandi dimensioni e con un certo tipo di simmetrie nelle strutture migliorino l'efficacia del sistema.

7.1 Sviluppi futuri

In conclusione, vengono presentati alcuni dei possibili sviluppi futuri del lavoro di tesi proposto.

Ampliamento dei dataset. Avere a disposizione un dataset più vasto permette di inferire informazioni più solide sulla struttura e sulla composizione degli edifici. In questo lavoro di tesi sono stati considerati dei dataset di dimensioni ridotte ($|\mathcal{G}| < 100$), aspetto che può rappresentare un limite

delle tipologie edilizie considerate.

Applicazione ad altre tipologie edilizie. I risultati ottenuti incoraggiano l'approfondimento del concetto di tipologia edilizia per la creazione dei dataset. L'applicazione del modello generativo a dati di altre tipologie edilizie sarebbe interessante, oltre che per validarne l'efficacia, anche per comprendere quali tipologie di edifici sono più strutturate rispetto ad altre. Esempi di tipologie edilizie oggetto di uno studio futuro possono essere gli ospedali o le grandi navi. Questo aspetto è collegato al precedente nel senso che è richiesto avere molti dati per ogni tipologia. Un altro sviluppo possibile rispetto a questo lavoro di tesi è la suddivisione della tipologia ABITAZIONI in tipi più specifici, quali appartamenti o ville.

Rivisitazione dello schema di labeling. L'efficacia delle label nel classificare i diversi tipi di stanze di un edificio e, di conseguenza, i nodi nel grafo corrispondente è molto importante. In questo lavoro di tesi si è cercato di mantenere un insieme di label minimo per caratterizzare le categorie principali di stanze, rifacendosi a uno schema standard di riferimento opportunamente modificato o esteso per ogni tipologia edilizia. Uno schema di labeling diverso potrebbe portare a delle variazioni nei risultati che, indirettamente, fornirebbero un'indicazione su quali schemi siano più descrittivi.

Campionamento dei grafi da altre distribuzioni. Un aspetto particolarmente delicato è la distribuzione a cui si approssimano i dati. Più la distribuzione è vicina a quella reale dei dati, più i risultati del campionamento saranno vicini ai dati iniziali. Posto che i tre tipi di distribuzione utilizzati nei passi del campionamento hanno mostrato buoni risultati, può comunque essere utile considerare altre distribuzioni, evidenziando vantaggi e limiti di ciascuna.

Creare ambienti realistici per le simulazioni. Il sistema sviluppato in questo lavoro di tesi può essere utilizzato per creare nuovi ambienti al fine di valutare l'interazione di robot simulati con ambienti realistici. Dal momento che molte applicazioni di robotica autonoma vengono oggi analizzate in un contesto simulato, l'uso di edifici generati in modo non casuale, ma con caratteristiche simili a quelle di edifici realmente esistenti, può accrescere la validità dei risultati rispetto a un contesto non simulato [3].

Predire le parti ignote di un edificio. Lo sviluppo futuro più importante, che ha motivato questo lavoro di tesi, è l'integrazione del modello

generativo in un sistema per la predizione di parti ignote a priori di un edificio, in modo da facilitare l'esplorazione da parte di un robot mobile autonomo. L'implementazione di un sistema per la predizione degli ambienti può essere ottenuta attraverso due fasi sequenziali.

- Un primo problema da risolvere è l'integrazione del sistema di predizione nell'ambito di un sistema robotico esistente. L'obiettivo è l'inclusione di un modello generativo, come quello proposto in questo lavoro di tesi, all'interno di un framework robotico completo (in aggiunta quindi alla percezione dei dati dall'ambiente esterno tramite i sensori del robot), similmente a quanto discusso in [54]. In questo modo, i nuovi grafi creati a partire dal modello generativo possono essere considerati un ampliamento della base di conoscenza rappresentata dai dataset ricavati dalle esplorazioni precedenti degli ambienti. Questo tipo di integrazione fornisce una capacità basilare di predizione, attraverso il campionamento di molti grafi finché all'interno di un certo numero di essi non si individuano dei sottografi simili al grafo corrispondente alla parte visitata di un edificio.
- Il secondo passaggio riguarda la realizzazione di un sistema di predizione dell'ambiente durante l'esplorazione del robot. In particolare, il sistema deve tenere conto degli aggiornamenti forniti in tempo reale dai dati dei sensori: il modello generativo è perciò adattato in modo da effettuare la fase di campionamento dei grafi online, parallelamente all'esplorazione di un robot. Dal momento che una parte dell'ambiente è già stata esplorata, il campionamento viene vincolato dal grafo parziale della parte nota dell'edificio. Inoltre, potrebbe essere conveniente effettuare dei piccoli campionamenti mirati a predire esclusivamente le zone vicine a quella in cui il robot si trova in quel momento.

Esempio di predizione

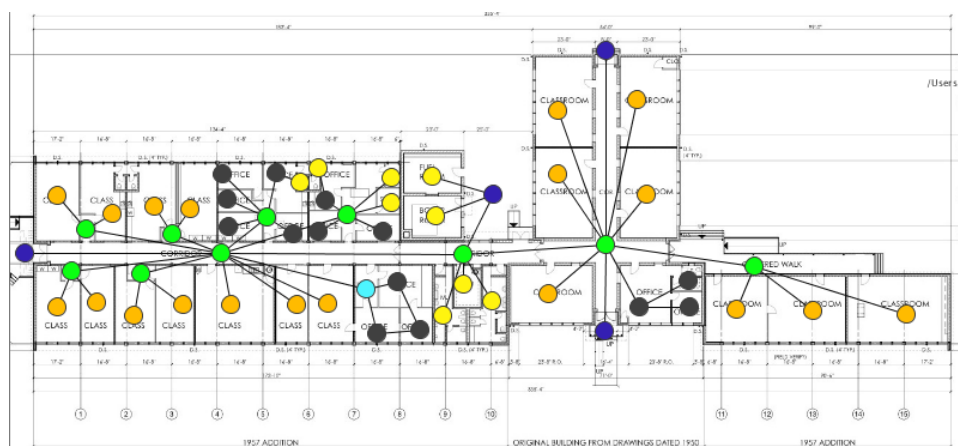
Anche se il sistema proposto nel lavoro di tesi è stato concepito per un utilizzo offline, si vuole mostrare un esempio di come potrebbe avvenire la predizione con l'ausilio del modello generativo. In Figura 7.1 è mostrato un edificio, di cui si immagina che un robot conosca solo parte della struttura, indicata nella parte (b) della figura; in pratica, solo parte del grafo rappresentante l'edificio è nota a priori.

In Figura 7.2 è mostrata la parte di grafo \dot{G} corrispondente alla porzione di ambiente nota al robot e un grafo generato a partire da essa per predire le parti inesplorate dell'edificio. \dot{G} viene processato seguendo i passi seguenti:

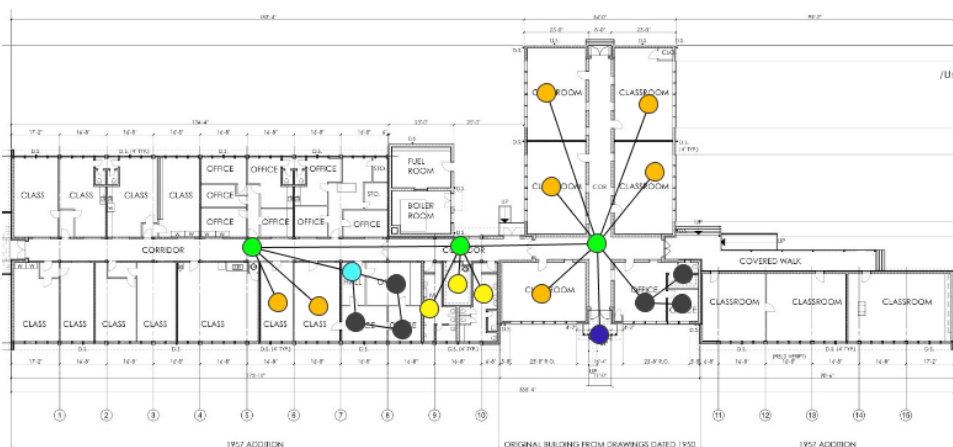
- \dot{G} viene segmentato in sottografi utilizzando Ncut;
- i sottografi così ottenuti vengono assegnati ad uno dei cluster appartenenti al modello generativo utilizzando lo stesso metodo implementato per la creazione del modello;
- il campionamento dei sottografi del grafo generato attraverso il campionamento di Gibbs avviene mantenendo fissi i sottografi di \dot{G} e le connessioni tra essi: \hat{G} conterrà quindi tutti i sottografi presenti in \dot{G} , con relative connessioni, che fungono come base di partenza, fissata a priori, per la fase di campionamento;
- i restanti sottografi sono campionati dal dataset utilizzando il modello implementato in questo lavoro di tesi e integrati, in maniera incrementale, a \hat{G} per ottenere un grafo \hat{G} che estende il grafo parziale \dot{G} ;
- il risultato mostrato in Figura 7.2 presenta una struttura simile al grafo completo dell'edificio mostrato in Figura 7.1, e rappresenta una possibile predizione sulle restanti stanze.

Questo esempio vuole mostrare un possibile metodo per integrare in un modello predittivo il modello generativo implementato in questo lavoro di tesi. Il risultato ottenuto mostra come sia effettivamente possibile predire la struttura di un ambiente indoor a partire da una esplorazione parziale dell'edificio. In particolare si può notare come venga preservata nel grafo generato la struttura lineare della scuola originale, composta da una serie di corridoi principali a cui sono connesse le restanti stanze. Inoltre, il numero di stanze ottenute nell'ambiente generato, intorno alle 75, è simile al numero di stanze individuabili nell'ambiente originale.

Al fine di migliorare ulteriormente il risultato della predizione della struttura dell'edificio si potrebbero integrare degli approcci *ad hoc* nel metodo presentato, adatti a sfruttare le simmetrie e le caratteristiche rilevate nella porzione di ambiente esplorato.

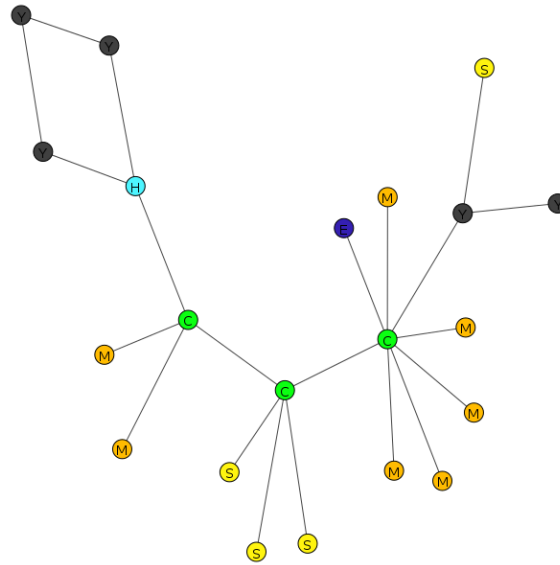


(a) Il grafo completo dell'edificio.

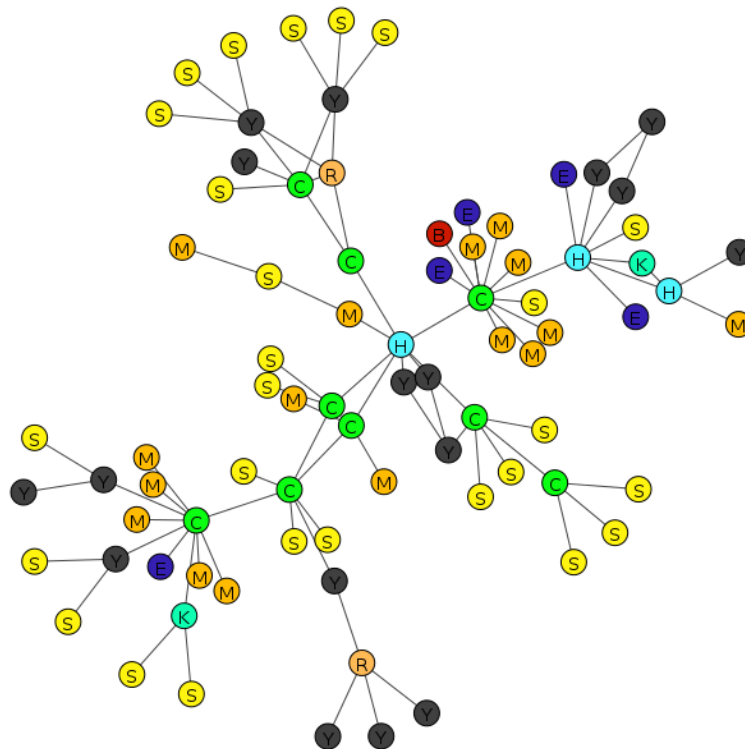


(b) La parte di ambiente nota al robot.

Figura 7.1: In (a) è mostrata la mappa che rappresenta il piano terra della Brunswick Training School, a cui è sovrapposto il grafo corrispondente. In (b) è mostrata la parte di mappa nota a un robot in base alle precedenti esplorazioni. I colori dei nodi nel grafo, indicanti le label delle stanze, fanno riferimento allo schema riportato in Figura 6.2.



(a) Il grafo corrispondente alla parte di ambiente nota.



(b) Un grafo generato a partire dalla parte nota.

Figura 7.2: Un esempio di un passo di predizione. In (a) è mostrato il solo grafo della parte di ambiente nota. In (b) è mostrato un grafo generato a partire dal precedente.

Bibliografia

- [1] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.
- [2] Philipp Althaus and Henrik Iskov Christensen. Behavior coordination in structured environments. *Advanced Robotics*, 17(7):657–674, 2003.
- [3] Francesco Amigoni, Matteo Luperto, and Viola Schiaffonati. Towards generalization of experimental results of autonomous robots. In *Proc. IAS-14*, 2014.
- [4] Alper Aydemir, Patric Jensfelt, and John Folkesson. What can we learn from 38,000 rooms? Reasoning about unexplored space in indoor environments. In *Proc. IROS*, pages 4675–4682, 2012.
- [5] Alper Aydemir, Andrzej Pronobis, Moritz Gobelbecker, and Patric Jensfelt. Active visual object search in unknown environments using uncertain semantics. *IEEE Transactions on Robotics*, 29(4):986–1002, 2013.
- [6] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.
- [7] Patrick Beeson, Nicholas Jong, and Benjamin Kuipers. Towards autonomous topological place detection using the extended voronoi graph. In *Proc. ICRA*, pages 4373–4379. IEEE, 2005.
- [8] Josh Bongard. Biologically inspired computing. *IEEE Computer*, 42(4):95–98, 2009.
- [9] Stephen Borgatti. Centrality and network flow. *Social networks*, 27(1):55–71, 2005.
- [10] Emma Brunskill, Thomas Kollar, and Nicholas Roy. Topological mapping using spectral clustering and classification. In *Proc. IROS*, pages 3491–3496. IEEE, 2007.

-
- [11] Horst Bunke. On a relation between graph edit distance and maximum common subgraph. *Pattern Recognition Letters*, 18(8):689–694, 1997.
- [12] Horst Bunke and Gudrun Allermann. Inexact graph matching for structural pattern recognition. *Pattern Recognition Letters*, 1(4):245–253, 1983.
- [13] Howie Choset and Keiji Nagatani. Topological simultaneous localization and mapping (slam): toward exact localization without explicit localization. *IEEE Transactions on Robotics and Automation*, 17(2):125–137, 2001.
- [14] Fan Rong Chung. *Spectral graph theory*, volume 92. American Mathematical Society, 1997.
- [15] Anthony Cohn and Jochen Renz. Qualitative spatial representation and reasoning. *Handbook of knowledge representation*, 3:551–596, 2008.
- [16] Fabiano Corrêa, Rodrigo Polastro, Fabio Cozman Gagliardi, and Jun Okamoto Jr. Dealing with semantic knowledge in robotics with a probabilistic description logic. In *Argentine Symposium on Artificial Intelligence*, 2010.
- [17] Gabor Csardi and Tamas Nepusz. The igraph software package for complex network research. *InterJournal, Complex Systems*, 1695(5), 2006.
- [18] Elisabetta Fabrizi and Alessandro Saffiotti. Extracting topology-based maps from gridmaps. In *Proc. ICRA*, volume 3, pages 2972–2978. IEEE, 2000.
- [19] John Folkesson, Patric Jensfelt, and Henrik Iskov Christensen. The m-space feature representation for slam. *IEEE Transactions on Robotics*, 23(5):1024–1035, 2007.
- [20] Cipriano Galindo, Juan-Antonio Fernández-Madriral, Javier González, and Alessandro Saffiotti. Robot task planning using semantic maps. *Robotics and Autonomous Systems*, 56(11):955–966, 2008.
- [21] Cipriano Galindo, Alessandro Saffiotti, Silvia Coradeschi, Pär Buschka, Juan-Antonio Fernandez-Madriral, and Javier González. Multi-hierarchical semantic maps for mobile robotics. In *Proc. IROS*, pages 2278–2283. IEEE, 2005.

-
- [22] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.
- [23] Steven Gold and Anand Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(4):377–388, 1996.
- [24] Gene Golub and Charles Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.
- [25] Sachithra Hemachandra, Matthew Walter, Stefanie Tellex, and Seth Teller. Learning spatial-semantic representations from natural language descriptions and scene classifications. In *Proc. ICRA*, pages 2623–2630, 2014.
- [26] Stephen Hirtle and John Jonides. Evidence of hierarchies in cognitive maps. *Memory & cognition*, 13(3):208–217, 1985.
- [27] Jeong-Hun Jang and Ki-Sang Hong. Fast line segment grouping method for finding globally more favorable line segments. *Pattern Recognition*, 35(10):2235–2247, 2002.
- [28] Tomihisa Kamada and Satoru Kawai. An algorithm for drawing general undirected graphs. *Information processing letters*, 31(1):7–15, 1989.
- [29] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.
- [30] Jana Košecká, Fayin Li, and Xialong Yang. Global localization and relative positioning based on scale-invariant keypoints. *Robotics and Autonomous Systems*, 52(1):27–38, 2005.
- [31] Jana Košecká, Liang Zhou, Philip Barber, and Zoran Duric. Qualitative image based localization in indoors environments. In *Proc. CVPR*, volume 2, pages 2–3. IEEE, 2003.
- [32] Harold Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.
- [33] Benjamin Kuipers and Yung-Tai Byun. A robust, qualitative method for robot spatial learning. In *Proc. AAAI*, volume 88, pages 774–779, 1988.

-
- [34] Rolf Lakaemper. Simultaneous multi-line-segment merging for robot mapping using mean shift clustering. In *Proc. IROS*, pages 1654–1660. IEEE, 2009.
- [35] Jean-Claude Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.
- [36] Bin Luo and Edwin Hancock. Structural graph matching using the em algorithm and singular value decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(10):1120–1136, 2001.
- [37] Bin Luo, Richard Wilson, and Edwin Hancock. A linear generative model for graph structure. In *Graph-based Representations in Pattern Recognition*, pages 54–62. Springer, 2005.
- [38] Matteo Luperto. Il concetto di tipologia edilizia per l’etichettatura semantica di stanze nell’ambito della robotica mobile autonoma. Master’s thesis, Politecnico di Milano, 2012.
- [39] Matteo Luperto and Francesco Amigoni. Exploiting structural properties of buildings towards general semantic mapping systems. In *Proc. IAS-13*, 2014.
- [40] Matteo Luperto, Alberto Quattrini Li, and Francesco Amigoni. A system for building semantic maps of indoor environments exploiting the concept of building typology. In *Proc. RoboCup*, 2013.
- [41] Timothy McNamara. Mental representations of spatial relations. *Cognitive psychology*, 18(1):87–121, 1986.
- [42] Joseph Modayil, Patrick Beeson, and Benjamin Kuipers. Using the topological skeleton for scalable global metrical map-building. In *Proc. IROS*, volume 2, pages 1530–1536. IEEE, 2004.
- [43] Michael Montemerlo, Sebastian Thrun, Daphne Koller, Ben Wegbreit, et al. Fastslam: A factored solution to the simultaneous localization and mapping problem. In *AAAI/IAAI*, pages 593–598, 2002.
- [44] Hans Moravec. Sensor fusion in certainty grids for mobile robots. *AI magazine*, 9(2):61, 1988.
- [45] Oscar Martinez Mozos. *Semantic Labeling of Places with Mobile Robots*. Springer-Verlag, 2010.

-
- [46] Oscar Martinez Mozos, Rudolph Triebel, Patric Jensfelt, Axel Rottmann, and Wolfram Burgard. Supervised semantic labeling of places using information extracted from sensor data. *Robotics and Autonomous Systems*, 55(5):391–402, 2007.
- [47] Peter Nacken. A metric for line segments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(12):1312–1318, 1993.
- [48] Paul Newman, John Leonard, Juan Tardos, and José Neira. Explore and return: Experimental validation of real-time concurrent mapping and localization. In *Proc. ICRA*, volume 2, pages 1802–1809. IEEE, 2002.
- [49] Alex Pothén, Horst Simon, and Kang-Pu Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM Journal on Matrix Analysis and Applications*, 11(3):430–452, 1990.
- [50] Processing. <http://processing.org/>, 2014. [Online; accessed 27-November-2014].
- [51] Andrzej Pronobis. *Semantic mapping with mobile robots*. PhD thesis, KTH Royal Institute of Technology, 2011.
- [52] Andrzej Pronobis and Patric Jensfelt. Large-scale semantic mapping and reasoning with heterogeneous modalities. In *Proc. ICRA*, pages 3515–3522, 2012.
- [53] Andrzej Pronobis, Patric Jensfelt, Kristoffer Sjöö, Hendrik Zender, Geert-Jan M Kruijff, Oscar Martinez Mozos, and Wolfram Burgard. Semantic modelling of space. In *Cognitive Systems*, pages 165–221. Springer, 2010.
- [54] Andrzej Pronobis, Oscar Martinez Mozos, Barbara Caputo, and Patric Jensfelt. Multi-modal semantic place classification. *The International Journal of Robotics Research*, 29(2-3):298–320, 2009.
- [55] Aldo Rossi. *The architecture of the city*. MIT Press Cambridge, MA, 1982.
- [56] Axel Rottmann, Oscar Martinez Mozos, Cyrill Stachniss, and Wolfram Burgard. Semantic place classification of indoor environments with mobile robots using boosting. In *Proc. AAAI*, volume 5, pages 1306–1311, 2005.

-
- [57] Alberto Sanfeliu and King-Sun Fu. A distance measure between attributed relational graphs for pattern recognition. *IEEE Transactions on Systems, Man and Cybernetics*, 13(3):353–362, 1983.
- [58] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
- [59] Reid Simmons, David Apfelbaum, Wolfram Burgard, Dieter Fox, Mark Moors, Sebastian Thrun, and Håkan Younes. Coordination for multi-robot exploration and mapping. In *Proc. AAAI*, 2000.
- [60] Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The annals of mathematical statistics*, pages 876–879, 1964.
- [61] Kristoffer Sjøo. Semantic map segmentation using function-based energy maximization. In *Proc. ICRA*, pages 4066–4073, 2012.
- [62] Lauro Snidaro and Gian Luca Foresti. Knowledge representation for ambient security. *Expert Systems*, 24(5):321–333, 2007.
- [63] Cyrill Stachniss, Oscar Martinez Mozos, and Wolfram Burgard. Speeding-up multi-robot exploration by considering semantic place information. In *Proc. ICRA*, pages 1692–1697, 2006.
- [64] Albert Stevens and Patty Coupe. Distortions in judged spatial relations. *Cognitive psychology*, 10(4):422–437, 1978.
- [65] Sebastian Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.
- [66] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. The MIT Press, 2005.
- [67] Koji Tsuda, Michihiko Minoh, and Katsuo Ikeda. Extracting straight lines by sequential fuzzy clustering. *Pattern Recognition Letters*, 17(6):643–649, 1996.
- [68] Shinji Umeyama. An eigendecomposition approach to weighted graph matching problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(5):695–703, 1988.
- [69] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of ‘small-world’ networks. *Nature*, 393(6684):440–442, 1998.

-
- [70] David White and Richard Wilson. Mixing spectral representations of graphs. In *Proc. ICPR*, volume 4, 2006.
- [71] David White and Richard Wilson. Spectral generative models for graphs. In *Proc. ICIAP*, 2007.
- [72] David White and Richard Wilson. Parts based generative models for graphs. In *Proc. ICPR*, 2008.
- [73] David White and Richard Wilson. *Generative Models for Graphs*. PhD thesis, University of York, 2009.
- [74] Denis Wolf and Gaurav Sukhatme. Semantic mapping using mobile robots. *IEEE Transactions on Robotics*, 24(2):245–258, 2008.
- [75] Zhenyu Wu and Richard Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1101–1113, 1993.
- [76] Xifeng Yan and Jiawei Han. gspan: Graph-based substructure pattern mining. In *Proc. ICDM*, pages 721–724. IEEE, 2002.
- [77] Hendrik Zender, Oscar Martinez Mozos, Patric Jensfelt, G-JM Kruijff, and Wolfram Burgard. Conceptual spatial representations for indoor mobile robots. *Robotics and Autonomous Systems*, 56(6):493–502, 2008.