



UNIVERSITA' DEGLI STUDI DI PADOVA

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea in Ingegneria Informatica

Relazione per la prova finale

Node Classification: Formulazione, Metodologie e Applicazioni

Relatrice: Prof.ssa Ornella Irrera

Laureando: Lorenzo Panizzolo

Matricola n.: 2117022

ANNO ACCADEMICO 2025/2026



Perché la Node Classification?

La **Node Classification** è uno dei task fondamentali nell'analisi di reti complesse. L'obiettivo è predire le proprietà mancanti di un nodo analizzando non solo i suoi dati, ma anche le sue relazioni.

Ambiti di Applicazione Reali

- **Cybersecurity e Finanza:** Rilevamento di account malevoli o frodi finanziarie.
- **Bioinformatica:** Predizione delle funzioni delle proteine (protein-protein interaction).
- **Reti Sociali:** Identificazione di *bot* tramite l'analisi delle loro connessioni.



Definizione del problema

- Un grafo $G = (V, E)$ descrive le **entità** e le loro **relazioni**.
- Ogni nodo possiede delle feature x_i ; solo un sottoinsieme ha un'etichetta nota.
- **Obiettivo:** predire le classi dei nodi ignoti usando **feature e topologia**.

Meccanismo predittivo

$$\hat{y}_i = f_{\theta}(v_i; G, X)$$

Differenza dal Machine Learning classico: I modelli standard assumono dati indipendenti. Nei grafi, gli archi rendono le osservazioni dipendenti e veicolano informazione cruciale.



Come affrontiamo il problema?

L'evoluzione dei modelli predittivi su grafi si articola in due paradigmi fondamentali:

1. Approcci Tradizionali

Basati su metriche euristiche o algoritmi di *Random Walk* (es. DeepWalk, Node2Vec).

- Catturano bene la topologia, ma faticano a integrare nativamente le feature originali dei nodi.

2. Graph Neural Networks (GNN)

Modelli di Deep Learning progettati specificamente per operare sui grafi.

- Fondono struttura e feature in una rappresentazione congiunta (*end-to-end*).

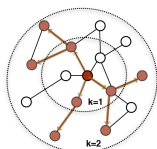
Il meccanismo matematico che consente alle GNN di realizzare tale fusione prende il nome di *Message Passing*.



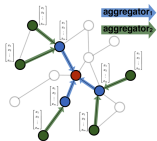
Il framework unificato: Message Passing

Aggiornamento della rappresentazione

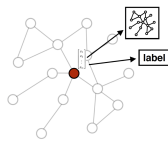
$$\mathbf{h}_i^{(\ell+1)} = \text{UPDATE}^{(\ell)}\left(\mathbf{h}_i^{(\ell)}, \text{AGGREGATE}^{(\ell)}\{\mathbf{h}_j^{(\ell)} : v_j \in \mathcal{N}(i)\}\right)$$



1. Sample neighborhood



2. Aggregate feature information from neighbors



3. Predict graph context and label using aggregated information

L'albero di aggregazione. Fonte: Hamilton et al. (2017)

- **Propagazione locale:** a ogni livello, un nodo aggiorna il proprio stato aggregando i messaggi dei soli vicini diretti.
- **Espansione del contesto:** impilando K layer, il nodo assorbe informazioni fino a K -hop di distanza.
- **Personalizzazione:** le architetture si distinguono per come implementano *AGGREGATE* e *UPDATE*.



Le GNN Classiche: Famiglie Architettureali

Il framework del *Message Passing* è stato declinato in diverse famiglie che costituiscono i *modelli di riferimento* del nostro studio:

- **Convoluzionali (GCN, ChebNet):** Aggregazione uniforme dei vicini basata su approssimazioni spettrali.
 - **Varianti (SGC, APPNP):** Semplificazione lineare (SGC), propagazione a lungo raggio (APPNP).
- **Spaziali (GraphSAGE):** Campionamento dei vicini per garantire scalabilità su reti massive.
- **Attentive (GAT):** *Self-attention* per pesare dinamicamente l'importanza di ciascun vicino.
- **Generative (GAE, VGAE):** *Auto-encoder* per l'apprendimento non supervisionato.



Quando la struttura aiuta? Omofilia ed Eterofilia

L'efficacia del *message passing* dipende fortemente dalla struttura del grafo, misurabile tramite il grado di omofilia $h(G)$.

Grafo Omofilo

- Nodi adiacenti tendono ad avere la **stessa etichetta**.
- Aggregare i vicini ("smoothing") rafforza il segnale corretto.
- Es: Reti di citazione, Reti sociali.

Grafo Eterofilo

- Nodi adiacenti hanno spesso etichette **diverse**.
- Lo smoothing mescola informazioni discordanti, degradando la predizione.
- Es: Reti di transazioni fraudolente.



Oltre il Message Passing: Modelli recenti

Le architetture classiche condividono limiti strutturali:

- **Eterofilia:** Falliscono se i vicini sono discordanti.
- **Over-smoothing:** Troppi layer rendono i nodi indistinguibili.
- **Over-squashing:** Colli di bottiglia nella propagazione.

Soluzioni Architetture Analizzate

- **H2GCN:** Progettato per mitigare l'**eterofilia**, separa le rappresentazioni del nodo stesso, dei vicini di 1° e di 2° grado.
- **Graph Transformer:** Abbandona la connettività locale usando un'attenzione globale, risolvendo l'**over-squashing** a costo di una complessità quadratica ($\mathcal{O}(n^2)$).



Valutazione: Metriche e Tipologie di Dati

Metriche di Valutazione

- **Accuracy:** Misura la percentuale di previsioni corrette.
- **Macro-F1 Score:** Media aritmetica degli F1-score di ogni singola classe. Fondamentale nei grafi reali poiché assegna lo stesso peso alle classi maggioritarie e minoritarie.

Dataset Benchmark

I modelli vengono testati su reti reali:

- **Grafi Omofili:** Reti accademiche (Cora, CiteSeer, PubMed, DBLP, Cora-ML) e reti e-commerce (Amazon CS, Photo).
- **Grafi Eterofili:** Reti di pagine web interconnesse (Texas, Wisconsin, Chameleon) e co-occorrenze (Actor).



Caratteristiche dei dataset

Dataset	Nodi	Archi	Feature	Classi	$h(G)$
Cora	2.708	5.278	1.433	7	0.81
CiteSeer	3.327	4.552	3.703	6	0.74
PubMed	19.717	44.324	500	3	0.80
DBLP	17.716	52.867	1.639	4	0.83
Cora-ML	2.995	8.158	2.879	7	0.79
CS	18.333	81.894	6.805	15	0.81
Photo	7.650	119.081	745	8	0.83
Texas	183	325	1.703	5	0.11
Wisconsin	251	515	1.703	5	0.20
Actor	7.600	30.019	932	5	0.22
Chameleon	2.277	36.101	2.325	5	0.23

Si nota il netto contrasto nell'indice di **omofilia** $h(G)$ tra i due gruppi.



Valutazione Sperimentale: Protocollo

Setup Unificato

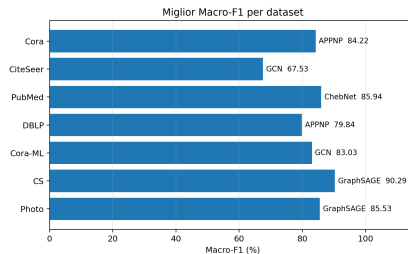
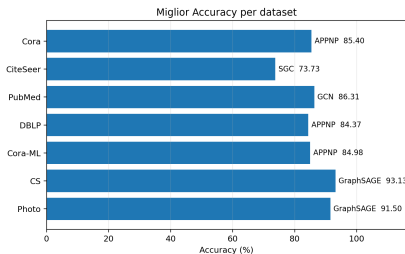
- **7 Dataset Omofili:** Cora, CiteSeer, PubMed, DBLP, Cora-ML, CS, Photo.
- **4 Dataset Eterofili:** Wisconsin, Actor, Texas, Chameleon.
- **Split (semi-supervisionato):** 10% Train (nodi etichettati), 10% Validation, 80% Test.
- **Iperparametri:** Grid Search, ottimizzatore Adam, Early Stopping.

Due scenari di confronto:

1. **Omofilia:** GCN, GAT, GraphSAGE, SGC, ChebNet, APPNP.
2. **Eterofilia:** MLP, GCN, ChebNet, H2GCN, Graph Transformer.



Risultati: Benchmark Omofili

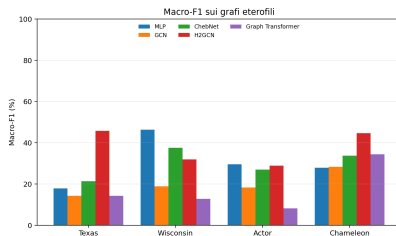
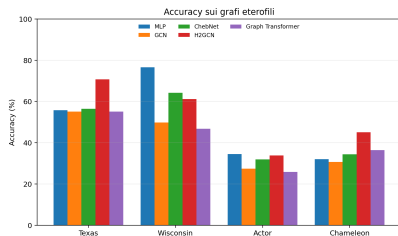


Evidenze Principali

- **Nessun vincitore universale:** APPNP primeggia su Cora/DBLP, GraphSAGE su CS/Photo, GCN e ChebNet su PubMed.
- **Metriche a confronto:** Il Macro-F1 smaschera le difficoltà del classificatore sulle classi minoritarie (es. calo drastico su CS e Photo).



Risultati: Benchmark Eterofili



Evidenze Principali

- **Limiti dell'aggregazione:** Su Wisconsin e Actor l'MLP batte le GNN, dimostrando che un'aggregazione non adatta ai dati eterofili genera rumore.
- **Il valore della topologia:** H2GCN domina su Texas e Chameleon. Separare il vicinato correttamente conferma che la struttura resta un'informazione fondamentale.
- **Mancanza di dati:** Il Graph Transformer ha bisogno di reti molto più grandi per potersi addestrare correttamente.



Conclusioni e Prospettive

Messaggi Chiave dell'Analisi

1. **La struttura non è una garanzia:** Nei grafi eterofili, aggregare i vicini può distruggere il potere predittivo.
2. **Selezione architetturale:** La scelta del modello deve essere guidata dalle proprietà strutturali della rete.
3. **Valutazione rigorosa:** Accuracy e Macro-F1 vanno valutate congiuntamente in contesti reali sbilanciati.

L'analisi preliminare della rete resta il passaggio imprescindibile per applicare il Deep Learning ai grafi con successo in ambiti quali l'identificazione di frodi finanziarie o la bioinformatica.