SGCL: Contrastive Representation Learning for Signed Graphs

==

SIGNED NETWORKS INTRO

1. Signed Networks

A **signed network** is a social network where relationships between users are represented as **positive (friendly)** or **negative (antagonistic)** links. Unlike traditional social network studies, which mostly focus on positive connections (e.g., friendships, followers), this study incorporates both positive and negative interactions.

2. Structural Balance Theory

Structural Balance Theory originates from **social psychology** (Heider, 1946) and was later formalized in graph-theoretic terms by Cartwright and Harary (1956). It focuses on the relationships within **triads** (groups of three users) and posits that:

- **Balanced Triads**:
  - **(+ + +)**: All three individuals are mutual friends.
  - **(+ − −)**: Two friends share a common enemy ("the enemy of my friend is my enemy").
- **Unbalanced Triads**:
  - **(+ + −)**: A common friend links two enemies.
  - **(− − −)**: All three individuals are enemies.

A network is considered structurally balanced if it has **more balanced triads and fewer unbalanced triads**.

3. Weak Structural Balance

A variant proposed by Davis (1960s), which **removes the assumption** that "the enemy of my enemy is my friend." This means that only **(+ + −)** configurations are considered implausible, while **(− − −)** can exist.

4. Status Theory

While **Structural Balance Theory** explains relationships in terms of likes/dislikes, **Status Theory** interprets links as indications of **hierarchical status**. It suggests:

- A **positive** link from A to B implies that **A sees B as having higher status**.
- A **negative** link from A to B means **A considers B lower in status**.

This theory better explains directed networks, where **social hierarchy and status differences influence interactions**.

5. Comparison: Balance Theory vs. Status Theory

- **Balance Theory** predicts that **friends of friends should be friends**, but this is often violated in real-world networks.
- **Status Theory** predicts that **lower-status users should dislike higher-status users** if there is a clear ranking.

For example:

- **A → B (+) and B → C (+)**: Balance theory expects **C → A (+)**, but Status Theory suggests **C → A (-)** (because C is higher in status).

==

<u>Link Prediction in Signed Networks using Contrastive Learning</u>

<u>**1. What is the objective of this project?**</u>

<u>The primary goal is to develop a model that can effectively predict links in signed networks—graphs where relationships can be both positive (friendships) and negative (hostility). Traditional Graph Neural Networks (GNNs) fail to distinguish between these two types of links, leading to poor representations.</u>

<u>Our approach explicitly models friendship and enmity separately while leveraging contrastive learning to ensure:</u>

- <u>Nodes remain close to their positive neighbors (friends).</u>
- <u>Nodes move away from their negative neighbors (enemies).</u>

<u>This technique enhances the ability to predict missing links and their polarity in applications like social networks (e.g., detecting toxicity on Twitter or Reddit), fraud detection, and recommendation systems.</u>

<u>**2. How do we process the input graph?**</u>

<u>We introduce **graph augmentation** to create two modified versions of the original signed graph. These versions preserve the essential structure but introduce small perturbations, such as minor changes in connections, to improve generalization.</u>

<u>Each of these augmented graphs is then split into two separate subgraphs:</u>

- **Positive Graph:** Contains only positive edges (friendships).
- **Negative Graph:** Contains only negative edges (hostility).

Why use multiple views?

Different augmented versions expose the model to diverse structural patterns, helping it generalize better. This ensures that embeddings are not overly dependent on specific structures that may exist only in the training data.

3. What kind of neural network is used?

We employ **two separate Graph Neural Networks (GNNs)** to independently process positive and negative relationships:

- **Positive GNN:** Learns friendship-based embeddings.
- **Negative GNN:** Learns enemy-based embeddings.

These networks share parameters across different graph views to ensure consistency, allowing the model to learn generalizable patterns from different perspectives.

4. How does the aggregation work inside GNNs?

Unlike traditional GNNs that average information from all neighbors, we use **attention mechanisms** to assign different weights to each neighbor.

Why attention?

Not all neighbors contribute equally to a node's representation. Some relationships are more influential, and attention mechanisms help the model focus on the most important ones.

How does attention work?

- Each neighbor's importance is determined based on learned weights.
- The model prioritizes strong connections while giving less importance to weaker or less relevant ones.
- The attention mechanism is inspired by Graph Attention Networks (GATs) but is specifically adapted for signed graphs.

5. Why use multi-head attention?

Instead of computing a single importance score for each neighbor, we use **multiple independent attention mechanisms (multi-head attention).**

Why?

- This improves model stability.
- Different attention heads capture different structural aspects of the network.
- It allows the model to learn a richer representation by focusing on different factors in separate attention heads.

How does it work?

- Each node aggregates information from its neighbors using multiple independent attention heads.
- The outputs of all heads are combined to form the final representation.
- This prevents over-reliance on any single aspect of the network.

6. How is the final node representation obtained?

Since the GNN operates in multiple layers, we combine information from all layers to get a **comprehensive embedding** for each node.

Each node ends up with:

- Two representations from different views of the **positive graph**.
- Two representations from different views of the **negative graph**.

To merge these, we use a **Multi-Layer Perceptron (MLP)** that processes all four embeddings to produce a final, unified representation of the node.

7. What is the role of contrastive learning?

Contrastive learning ensures that the model learns meaningful embeddings by enforcing a simple principle:

- Nodes should be **closer** to their positive neighbors.
- Nodes should be **farther** from their negative neighbors.

How does contrastive learning help?

- It **reinforces structural balance**, ensuring that positive relationships strengthen and negative relationships remain distinct.
- It **improves robustness**, ensuring that the model learns discriminative features that generalize well.
- It **prevents overfitting**, as the model is trained to recognize fundamental patterns rather than memorizing specific graph structures.

8. Why does this approach work well?

- **Signed graphs require explicit handling** → A single GNN struggles to model both positive and negative interactions at once.
- **Contrastive learning enforces meaningful embeddings** → It ensures that the model properly distinguishes between positive and negative relationships.
- **Multi-head attention improves representation** → Different attention heads focus on different aspects of the network.
- **Graph augmentations improve generalization** → By introducing variations in the graph structure, the model learns to focus on core patterns rather than noise.

This comprehensive approach significantly **outperforms traditional methods** in predicting links and their signs in complex signed networks.