Network Analysis of Codeforces Data for Skill Progression and Talent Identification

Group 14

Team Members

Arey Pragna Sri	- 12240230
Matcha Jhansi Lakshmi	- 12241000
Shivangi Gaur	- 12241720
Chelluboina Siri	- 12240460

1 Introduction

This project addresses the need to understand user progress, problem difficulty, and talent identification in competitive programming by implementing network science techniques to analyze user interactions with problems and contests on Codeforces. The need for this project arises due to the vast amount of problem-solving data available on Codeforces, which lacks structured insights into learning patterns, skill development, and rating progression. By conducting this study, we aim to provide insights into key factors influencing skill growth, personalized problem recommendations, and talent identification, benefiting individual learners, competitive programming platforms, and recruiters.

2 Objectives

The main objectives of this project are:

1. **Identifying Must-Solve Problems**
   - Detect critical problems that act as "gatekeepers" for progress in competitive programming.
   - Identify problems frequently solved by users who achieve rapid skill growth.
2. **Community Detection of Users Solving Similar Problems**
   - Cluster users based on their problem-solving patterns to form effective learning communities.
   - Enable peer-to-peer learning by connecting users with similar strengths and weaknesses.
3. **Problem Recommendation System**
   - Develop a network-based recommendation system that suggests problems based on solving patterns and difficulty levels.

- Optimize recommendations by analyzing problems frequently solved together and topic-wise proficiency.

4. **Personalized Learning Path for Problem Solving**
   - Design adaptive learning paths based on a user's past performance and topic-wise proficiency.
   - Ensure structured mastery by guiding users through progressive problem sequences.
5. **Talent Identification for Recruiters**
   - Leverage centrality metrics and problem-solving efficiency to identify high-potential candidates beyond traditional rating systems.
   - Categorize users based on problem-solving expertise to match them with relevant industry roles.

3 Methodology

1. Data Collection

We extract data from the Codeforces API and model it as graphs:

- User-Problem Bipartite Graph: Nodes represent users and problems, edges encode solve attempts (weighted by time/attempts).
- Problem-Problem Co-Solving Graph: Edges connect frequently co-solved problems (weighted by transition frequency).
- Contest Participation Network: Users connected to contests, weighted by performance (rating change).

2. Analysis & Implementation

We apply network analysis techniques to derive insights:

- Personalized Learning Paths
  - Use shortest-path algorithms (Dijkstra's) on the co-solving graph to recommend optimal problem sequences.
  - Apply Markov chains to predict next-best problems based on historical transitions.
- Weaknesses/Strengths & Gatekeeper Problems
  - Compute betweenness centrality to identify critical "gatekeeper" problems.
  - Use community detection (Louvain) to cluster problems by topic, then compare user solve rates per cluster.
- Talent Identification
  - PageRank: Ranks users solving high-impact problems early.
  - Betweenness: Finds versatile solvers bridging multiple topics.

  - Temporal Graphs: Tracks rating growth patterns over time.
  - GNN + Clustering: Groups specialists by skill profiles.
- Rating Boost Strategies
  - Analyze rating change subgraphs to identify contest types/problem patterns that maximize gains.

3. Evaluation

We validate our methods using:

- Quantitative Metrics:
  - Recommendation accuracy (hit rate of suggested problems).
  - Correlation between predicted vs. actual rating changes.
- Qualitative Comparison:
  - Compare detected gatekeepers with Codeforces official problem ratings.
  - Validate talent clusters against known high-performing users.

This network-driven approach transforms raw Codeforces data into actionable insights for learners, coders, problem setters,and recruiters.

4 Expected OutComes

- **Identification of Must-Solve Problems:** A curated list of key problems essential for rating improvement.
- **Formation of Learning Communities:** Clustering users with similar problem-solving patterns for peer-to-peer learning.
- **Problem Recommendation System:** Data-driven recommendations for optimal problem-solving progression.
- **Personalized Learning Paths:** Adaptive learning sequences based on user strengths and weaknesses.
- **Talent Identification for Recruiters:** Identifying top candidates based on problem-solving efficiency.
- **Strategies to Boost Ratings:** Insights into effective contest participation and problem-solving strategies.