Revision List

1. Contribution
   1. The strength of the proposed index is that it enables for more general queries than prior work. Show the significance of the problem by providing some compelling, real-world applications of the index.
2. Introduction
   1. Properly Introduce the problem, explain the main concepts well in an intuitive way. Easy to follow introduction for readers who are not so familiar with the problem.
   2. Define the community model and various types of query clearly.
   3. Clarify the difference between different types of queries.
3. Presentation
   1. Small grammar errors
   2. Capitalize references to pseudo codes, tables or figures
   3. Some parts of the descriptions are hard to follow. Try to explain in simple terms.
      1. Clarify the construction of community graph.
      2. Make the union-intersection algorithm easy to understand. Use simpler commands in pseudocode, and explicitly map the union and intersection operations to code in Algorithm 3.
      3. Preliminaries: Using the running example here to make them much easier to understand. Some more informative descriptions would help.
      4. Explain why sorting the edges in G^o should be useful.
      5. Clarify the complexity analysis parts, give more details.
4. Evaluation
   1. Compare with baseline for multiple types of queries. Even if baseline needs some sort of modifications. If it is not possible, explain why. [Pending]
   2. Why exactly the 2-level index is so much better for some dataset compared to the other datasets. Identify the factors that are beneficial for the 2-level index and explain why these have such an impact on the proposed method. [Pending]
   3. Provide the truss decomposition times for each of the graphs.
5. Small problems
   1. A better description of figure 1 has been added.
   2. A better description of figure 3 has been added.
   3. Include running times for boundary queries? [Pending]
   4. Merge theorem 1 into Definition 7.
   5. Move proves of theorem 1 and theorem 2 to appendix.