Cover Page for SIGMOD 2023 Submission #166

Thanks a lot for your thoughtful and constructive review comments. We have prepared a careful revision of the manuscript to address the comments and advice. A summary of major revisions is given below, followed by the point-to-point response to each reviewer.

- We conduct more comprehensive experiments as suggested by reviewers. We have included experimental results of the Adjusted Rand Index (ARI) (addressing R3/O3); conducted modularity experiments using set containment (addressing R5/O1); included large graphs in the modularity experiments (addressing R5/O3).
- We have done careful proof-reading and improved the presentation (addressing R4/O1, R5/O2, R3/O1);
- We have included more discussion about related work, made the paper more self-contained (addressing R3/O2), and shown some empirical rules on parameter adjustment (addressing R5/O1).

In the revised manuscript, we highlight in blue the changes that are most relevant to the reviewers' comments. Due to limited space, we cannot include all experiments and running examples in the revised manuscript. We provide omitted experiments (using set containment as the similarity measure) in this cover page. For omitted running examples in Sections 3-4, please kindly refer to our full-version technical report[1]. Our responses to the reviewers are as follows.

RESPONSE TO REVIEWER #3

Comment 1.1 **(O1):** The paper is vague in important points of technical development. Bottom-k sketches, which is a very important notion for the paper, are not explained well. The reader is pointed to [9] and [10], so the paper is not self-contained. Furthermore, it is not clear how much of the ideas come from [9] and [10] and how much from the current paper.

Response: As suggested, we have included the formal definition of bottom-k sketches and all-distances bottom-k sketches (ADS) in Section 2.2 for completeness. Computing Jaccard similarity by bottom-k sketches and the concept of ADS are introduced by [8] ([9] in the original manuscript) and [9]([10] in the original manuscript), respectively. Even though bottom-k sketches and ADS are not new ideas, we make a connection to our studied distance-based SCAN problem and tackle the challenging task of Jaccard similarity estimation. In addition, even with bottom-k sketches, we still need to go through k sketch elements to derive the Jaccard similarity between two vertices while the size k might be large. To tackle this efficiency issue, we further present the histogram index to avoid going through bottom-k sketches if the computation of structural similarity can be pruned by the histogram index. We have highlighted the main technical contribution in Section 1.

[1] https://anonymous.4open.science/r/Distance_SCAN_SIGMOD-BCD0

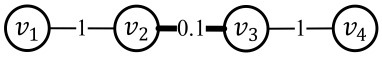


Figure 1: An example graph

Comment 1.2 **(O2):** Important work has only been mentioned in passing without deeper discussion. For example work [16] on WeightedSCAN is used in experiments, but not discussed in Existing Solutions. On the other hand, works described in a lot of detail in Existing Solutions, such as GS*-index [30] are not used in experiments. Also, SHRINK [15] which uses modularity as optimization objective is also mentioned in passing, but no rationale provided why it is not considered as existing solution and included in the experiments.

Response: Following your advice, we have discussed WeightedSCAN in Section 2.2. WeightedSCAN and SCAN are very similar. The main difference is that WeightedSCAN uses the weighted Cosine distance to measure structural similarity. Both WeightedSCAN and SCAN return clustering results with lower modularity than our proposed distance-based SCAN as shown in our experiments.

For GS*-Index, it is not suitable to our problem. As shown in Example 1, the structural similarities between vertices will increase or decrease as the distance threshold increases, so the structural similarities tend to be different under different distance thresholds. In this case, it is difficult for GS*-Index to pre-compute and stores the structural similarity when the distance threshold is unknown, which has been explained in Section 2.2. Therefore, GS*-Index is not suitable for our problem, and we have shortened the discussion about GS*-Index. For pSCAN, as it is an online algorithm, it can be easily modified to our setting (deriving the similarity between two vertices according to the d-neighborhood). Thus, it stands as a baseline of our solution.

For SHRINK, it is orthogonal to our study as they focus on user-friendly design to avoid tuning input parameters yet still find good clustering results. SHRINK still uses structural similarity defined for one-hop neighbors. We can easily extend d-neighborhood structural similarity to SHRINK so as to gain better clustering results.

Example 1. *Fig. 1 shows an example graph of structural graph clustering. This is a weighted undirected graph. The weight on the edge represents the distance between two vertices. A small distance means that the two vertices are closely related. When* $d \in [0.1, 1)$, $N_d(v_2) = \{v_2, v_3\}$, $N_d(v_3) = \{v_2, v_3\}$, *the structural similarity of* v_2, v_3 *is* $\sigma(v_2, v_3) = 1$. *When* $d \in [1, 1.1)$, $N_d(v_2) = \{v_1, v_2, v_3\}$, $N_d(v_3) = \{v_2, v_3, v_4\}$, *the structural similarity of* v_2, v_3 *is* $\sigma(v_2, v_3) = 0.5$. *When* $d \geq 1.1$, $N_d(v_2) = \{v_1, v_2, v_3, v_4\}$, $N_d(v_3) = \{v_1, v_2, v_3, v_4\}$, *the structural similarity of* v_2, v_3 *is* $\sigma(v_2, v_3) = 1$.

Comment 1.3 **(O3):** Some experiments are lacking when it comes to varying parameters. For example, in Fig 4, how does

the running time change when varying distance threshold d. Also, how is Adjusted Rand Index (ARI) [17] calculated? How does it change when default parameters are changed?

Response: Thanks for your insightful comments. We note that Fig. 4 mainly shows the running time of all methods on all datasets with default parameters. We have experiments to examine the impact of different parameters such as d, ϵ, and k in Section 5.3. For example, Fig. 5 shows the running time when we vary the distance threshold d. As the distance threshold d increases, the advantage of DistanceSCAN over pSCAN and EXACT become more significant in terms of clustering efficiency. On the other hand, as we have shown in Fig. 1 and Fig. 3, our solution achieves identical clustering results as EXACT and is more effective than the classic SCAN whose similarity is defined based on one-hop neighbors. This demonstrates that our DistanceSCAN gains a good trade-off between clustering efficiency and clustering quality.

As advised, we have included the definition of ARI and reported the experimental results of ARI of DistanceSCAN and pSCAN over the ground truth clustering result provided by EXACT when we vary d (Figure 3(a)) and vary ϵ (Figure 3(b)) in Section 5.2. The results all show that our DistanceSCAN can gain an ARI score of 1 in all settings, meaning that we derive the same clustering result as EXACT. This demonstrates the high accuracy of our DistanceSCAN.

RESPONSE TO REVIEWER #4

Comment 2.1 (O1): The presentation is rather poor. The text needs proofreading, as it abounds in grammar errors, missing and incorrect words. The paper in general is hard to follow, especially the experimental analysis. See below for more detailed comments.

Response: Thanks for your insightful comments. We have carefully proofread and heavily rewritten the manuscript to improve the presentation based on your suggestions. Moreover, we have thoroughly revised the experimental analysis in Section 5.

Comment 2.2 (O1-M1,M3): *(M1)* The abstract seems too long, while the conclusions are too short, lacking any directions for future work. Please shorten the abstract and extend the conclusions. Add also a couple of sentences about future work. *(M3):* Section 2.2 is too long and detailed. It could be shortened and merged with Section 6.

Response: Following your advice, we have shortened the abstract, condensed the description of GS*-Index in Section 2.2, and added an outlook for future work in the conclusions.

Comment 2.3 (O1-M2): The term "shortest path" appears twice in the paper. I guess it should be replaced with the term "shortest distance".

Response: Thanks for your comments. We note that distance is defined as the sum of the weights on the shortest path. So the distance information needs to be obtained by calculating the shortest path between vertices. Therefore, the two appearances of shortest path in the paper are appropriate. To avoid misunderstanding, we use the term "shortest path" instead of "shortest distance" uniformly in the revision.

Comment 2.4 (O1-M4): In Section 3, no reference is provided for the pruned Dijkstra algorithm.

Response: Sorry about the confusion. The pruned Dijkstra algorithm refers to adding pruning conditions to the Dijkstra algorithm to traverse neighbors whose distance does not exceed a given threshold. We describe the Dijkstra algorithm with pruning conditions in detail in the revision. To avoid misunderstanding, we use the term "the Dijkstra algorithm" instead of "the pruned Dijkstra algorithm".

Comment 2.5 (O1-M5,M8,M11): *(M5)* Please explain Algorithm 1 line by line. Please add a comment in Line 10 that Algo. 2 is called.) is missing from Line 13. *(M8)* In Algorithm 4, Ture should be True. *(M11)* In Algorithm 6, add a comment in Lines 3 and 8 that Algorithm 7 is called. In Lines 4 and 9 change "u is not unclassified" to "u is classified".

Response: Thanks for your comments. We have included a detailed description of Algorithm 1 in Section 3.1. We have fixed the typos, carefully proofread the whole paper, and considerably rewritten the manuscript to improve the presentation.

Comment 2.6 (M6): Are Algorithms 2 and 3 really new?

Response: Thanks for your comments. On one hand, instead of adopting persistent search trees to store ADS as in [9], we propose to use a priority queue to store and read ADS. This greatly reduces the space consumption and construction time of ADS which is verified by the experimental results in Section 5.4. On the other hand, one of our main technical contributions is to connect the bottom-k sketches and ADS with our proposed Distance-based SCAN problem and tackle the efficiency issues of similarity computations in distanced-based SCAN.

Comment 2.7 (M7): It would be nice to add a running example in Sections 3 and 4 to highlight the core notions and operations of these algorithms. It would also be nice to add an approach outline that explains the role and contribution of each algorithm in Sections 3 and 4. At the moment, it is hard to follow their relation.

Response: Thanks for your advice. We have revised Section 3 and Section 4 to make them more organized, and have added the summaries of algorithms and their relationships in both sections. Section 3 has two parts. One is to show how to construct ADS and histograms (Algorithm 1,2) and retrieve the bottom-k sketches from ADS (Algorithm 3). The second one is to use histograms and bottom-k sketches to prune and quickly calculate the structural similarity (Algorithm 4). Section 4 introduces DistanceSCAN (Algorithm 5), which is our main algorithm to derive the clustering result. A main step in DistanceSCAN is to detect core vertices (Algorithm 6). The

classification of vertices as core or non-core vertices is based on the number of structurally similar neighbors. Therefore, Algorithm 7 calls Algorithm 4 and integrates pruning rules to speed up the core vertex detection process (Algorithm 6). We have included the discussion in our revised manuscript.

Due to the limited space, we cannot put running examples in Sections 3-4. We have included running examples in Sections 3-4 in our full-version technical report.

Comment 2.8 (M9): Please add the proofs of Theorem 3.3 and Lemma 3.4 in Section 3.2. You could save the necessary space by shortening Section 2.2. Note also that the text in general is rather repetitive and a lot of space can be saved by avoiding repetitions. In the worst case, the proofs should be provided in an online extended version.

Response: Thanks for your suggestion. We have included the proofs in our full-version technical report. Note that we have renamed Lemma 3.4 as Theorem 3.4 (in the revised manuscript).

Comment 2.9 (M10): In Algorithm 5, add a comment in Line 1 that Algo. 1 is called. Used the same title as Algorithm 1.

Response: As we mentioned in the second paragraph of Section 4.1, InitSketches is not the same as Algorithm 1. If ADS and histograms already exist in the disk or memory, InitSketches directly reads sketches. Otherwise InitSketches calls Algorithm 1 to construct ADS and histograms. In the revised manuscript, we have improved the presentation and added a comment to explain these two cases in the pseudo-code of Algorithm 5.

Comment 2.10 (M12): Please justify the time complexity of Algorithm 7 at the end of Section 4.3.

Response: We note that $O(k + m/n)$ is the amortized complexity of Algorithm 7. Algorithm 7 mainly consists of two parts. One part is to call Algorithm 4 to calculate the structural similarity, and the time complexity of this part is $O(k)$. The other part is to detect the clusters where one-hop neighbors are located and merge the eligible clusters. The time complexity of merging clusters is $O(1)$ due to the use of the disjoint-set data structure. The amortized complexity of traversing one-hop neighbors is $O(m/n)$. So the amortized time complexity is $O(k + m/n)$. We have added an analysis of the time complexity of Algorithm 7 in Section 4.3. In addition, we have further presented the time complexity analysis for Algorithm 6 as it is the main algorithm that invokes Algorithm 7.

Comment 2.11 (M13): At the beginning of Section 5, please add a list of the research questions that are examined in the experimental analysis. Then, explain which experiment answers which question. At the moment, it is hard to follow the meaning of every experiment.

Response: Thanks for your suggestion. Following your advice, we have made three main changes. First, the experimental research questions for all subsections are listed at the beginning of Section 5. Second, the main experimental content is introduced at the beginning of each subsection. Finally, we add boldface headings to the paragraphs of the experimental analysis.

Comment 2.12 (M14): In Tab. 1, please explain the meaning of last two columns in the caption. Please also add a column to indicate which datasets use synthetic weights. Why is the distance of each edge selected from (0.1, 1] instead of (0, 1]?

Response: Following your advice, we have explained the meaning of each column in Table 1 and added a column indicating whether the dataset has real weights. We set the minimum distance to 0.1 instead of 0 for the following two considerations. On one hand, if there are many edges in the graph with distances close to 0, the d-neighborhoods of vertices will grow sharply with the increase of d. If the d-neighborhoods are too large, the d-neighborhoods of vertices will overlap a lot, so that the vertices are easy to be structurally similar. The size of clusters formed in this case will be very large, even close to the size of the graph, making the quality of the clustering results poor. On the other hand, setting a non-zero minimum value for distance helps limit the number of hops of neighbors in the d-neighborhoods. For example, when the weight threshold range is $[0.1, 1)$ and the distance threshold d is less than 0.3, the d-neighborhoods contain at most two-hop neighbors. In the revision, in order to make the distance setting more reasonable, the distances of the datasets other than Topic-Coauthor-T24 are adopted according to the Jaccard similarity of adjacent vertices.

Comment 2.13 (M15): In Section 5.2, EXACT mentioned in the text instead of DistanceSCAN, which appears in the captions of Figures 2 and 3. This is confusing. Why isn't pSCAN included in these experiments? Please also explain why these experiments use only one of the datasets. Please also add a figure to report ARI values discussed at the end of Section 5.2.

Response: Sorry for the confusion. Firstly, we have included the results of SCAN, DistanceSCAN, EXACT, and pSCAN in Figure 2 (evaluating clustering quality with Jaccard similarity). We further revised the description to avoid confusion. Besides, we have added three datasets BrightKite, Youtube, and Livejournal (with different scales) to show their results. We omit the results on UK-2002 as pSCAN and EXACT cannot derive the clustering results and then compute the modularity score (this part can be expensive for a large cluster according to Equation 10) in 48 hours on a fixed set of input parameters. The weight on each edge is calculated based on their Jaccard similarity. Following our previous discussion, we convert it to distance and normalize it to the range of $[0.1, 1)$. In particular, given a Jaccard similarity score x, we transform it to $1 - 0.9x$. As shown in Figure 2 in the revised manuscript, the results show that our distance-based SCAN still returns high-quality clustering results with much higher modularity scores. We note that the range of distance threshold d in Figure 2(a) and Figures 2(c), (e), (f) are different since their weights have different meanings. When the weights have the same meaning, e.g., for Figures 2(c), (e), (f), we can use the same range of distance threshold d.

In Figure 3, it evaluates the effectiveness of our distance-based SCAN problem when we change the distance measure, e.g., using Cosine similarity. For the interest of space, we only show results on two datasets: Topic-Coauthor-T24 (dubbed as TC-T24), and Youtube. We report the results on SCAN, EXACT, pSCAN, and WeightedSCAN. Notice that DistanceSCAN is designed for Jaccard similarity and thus it is not included here. The observation is that EXACT (and hence also pSCAN) can still derive high-quality clustering results with much higher modularity scores than SCAN. These results show that regardless of using Jaccard or Cosine similarity, the clustering results of distance-based SCAN are far better than WeightedSCAN. This is because WeightedSCAN considers the weight information of one-hop neighbors, but the utilization of the weight information is still not as sufficient as our proposed idea of distance-based structural graph clustering. In addition, we observe that for both Jaccard similarity and Cosine similarity, Distance-based SCAN can achieve identical best clustering results in terms of modularity scores. Thus, we focus on Jaccard similarity.

For the ARI values, we have included Figure 3 to show the ARI scores for our DistanceSCAN on UK-2002 dataset. The ground truth clustering results are derived by pSCAN. We add the ARI score of pSCAN (which is always 1) as a reference. The experimental results show that on various parameter settings of the largest dataset UK-2002, the ARI of DistanceSCAN is always 1, which verifies the accuracy of DistanceSCAN.

We have included the above discussions in the revised manuscript.

Comment 2.14 (M16,M17): *(M16)* Please add the unit of measurement in the vertical axis of Figure 4. *(M17)* Please justify the selection of datasets in Figures 5-10 and 12-16.

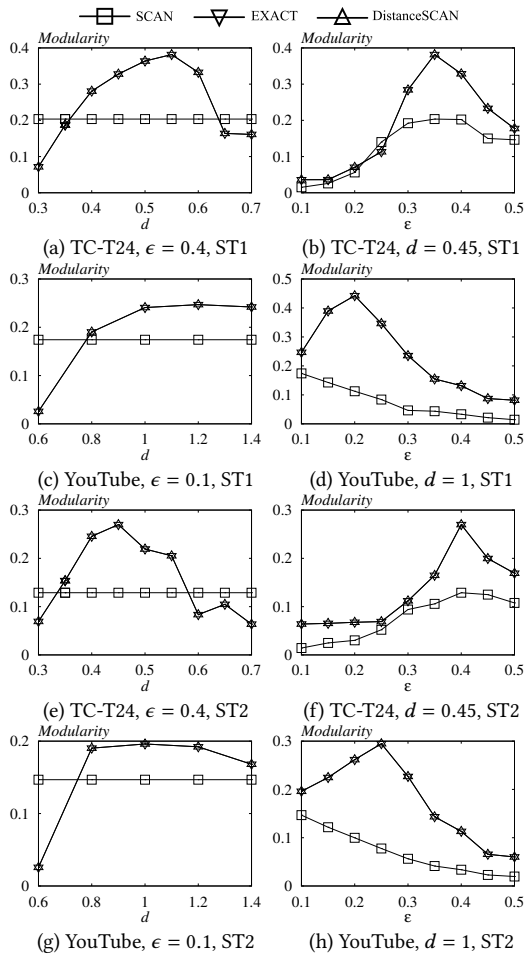
Response: Thanks for your comments. We have corrected Fig. 4 to include the time unit.

Thanks a lot for pinpointing the concerns on the selection of datasets. In the revision, we select datasets according to the size of the graph, more precisely, the number of edges of each graph. We focus on four datasets in these experiments: BrightKite, YouTube, LiveJounal, and UK-2002. To make it consistent, when we have four figures, we will include all these four datasets. When we have only two figures, we show the results for YouTube and UK-2002.

RESPONSE TO REVIEWER #5

Comment 3.1 (O1): Technical Details: *(i)* Jaccard similarity is biased towards small set sizes. Why not consider set containment as in [1]. *(ii)* The proposed method seems to be sensitive to parameter tuning. The authors need to indicate some guidelines on how to set the thresholds d, ϵ and μ?

Response: Thanks for your suggestion. As Jaccard similarity is widely used in the research of SCAN, e.g. [5], [22], we adopt it in this paper. For set containment, it is an asymmetric measure. In particular, given vertices u and v, the similarity can be measured using $|N_d(u) \cup N_d(v)|/N_d(u)$ or $|N_d(u) \cup N_d(v)|/N_d(v)$. After careful consideration, we find that set containment is unsuitable for distance measure. To explain, given vertices u and v on the undirected graph, the distance from u to v is equal to the distance from v to u, i.e., $\delta(u, v) = \delta(v, u)$. However, by using set containment, we cannot satisfy the above properties.



(a) TC-T24, $\epsilon = 0.4$, ST1 (b) TC-T24, $d = 0.45$, ST1
(c) YouTube, $\epsilon = 0.1$, ST1 (d) YouTube, $d = 1$, ST1
(e) TC-T24, $\epsilon = 0.4$, ST2 (f) TC-T24, $d = 0.45$, ST2
(g) YouTube, $\epsilon = 0.1$, ST2 (h) YouTube, $d = 1$, ST2

Figure 2: Evaluation of modularity using set containment

Nevertheless, we still try to provide modularity experiments based on set containment. Notice that results might change if we change the order of computations since the measure is asymmetric. Our DistanceSCAN can be adapted to set containment, and we only need to use bottom-k sketches of the corresponding sets when calculating the structural similarity. Figure 2 reports the modularity of SCAN, EXACT and DistanceSCAN when clustering based on set containment with a fixing order of computation from smaller node ID to larger node IDs. For the scanned vertex u and the neighbor v of vertex u, Figures 2 (a)-(d) show the modularities of the clustering results using $|N_d(u) \cup N_d(v)|/N_d(u)$ as the structural similarity, and Figures 2 (e)-(h) show the modularities of the clustering results using $|N_d(u) \cup N_d(v)|/N_d(v)$ as the structural similarity. The experimental results obtained by measuring the structural similarity in these two ways are similar. Notice that the distance of TC-T24 is the real weight while the distance of YouTube is adopted

according to the Jaccard similarity of adjacent vertices. Due to different physical meanings, the range of the distance threshold d is different (Ref. to our response to Comment 2.13 for more details). Figures 2 (a),(c),(e),(g) find suitable distance thresholds on the similarity threshold ϵ that SCAN performs best. EXACT and DistanceSCAN outperform SCAN for most values of ϵ in Figures 2 (b),(d),(f),(h). However, we emphasize that Distance-based SCAN or SCAN on set containment is not well defined. Hence, we omit the above results in our manuscript and technical report.

In terms of input parameters, we can set them in the same way as SCAN by tuning these parameters while fixing the other two (as we have done in the experiment). We can then choose the one with the highest modularity score. We show the process of finding suitable distance thresholds on multiple datasets in Section 5.2. We also point out that the modularity generally increases first and then decreases as the distance threshold increases, which is very helpful for finding a suitable distance threshold.

Comment 3.2 (O2) Correct Language mistakes. Examples below: "viral marketing[12]", "which are tend to be", "then classifies non-core vertices, hub and outliers" → hub or hubs?, "to an weighted"

Response: Thanks for your comments. We have carefully polished our manuscript.

Comment 3.3 (O3) Experiments: *(i)* Define modularity and explain it intuitively. *(ii)* Add SCAN to Figures 4, 5. 6 and 7. *(iii)* The modularity experiments were run on only the smallest dataset: Topic-coauthor-T24. Please include results on the larger datasets: YouTube, Pokec, LiveJournal, Orkut, UK-2002; *(iv)* Add other baselines such as [2,3].

Response: As suggested, we have added the discussion about the definition and intuition of modularity.

We have added EXACT into Figures 5-7, which is a modified version of SCAN for our problem.

In the modularity experiment, we only included the results on the Topic-coauthor-T24 dataset in the original manuscript as it has real weight and the result will be more meaningful. In the revised manuscript, we have included results on three more datasets: BrightKite, YouTube, LiveJounal, which have different scales of edges. For Orkut and UK-2002, we cannot derive the clustering result and derive the modularity score (this part can be expensive as it involves $O(n^2)$ computation in the worst case according to Equation 10) on Orkut and UK-2002 within 48 hours for a fixed set of input parameters.

As explained in R3/O2, GS*-Index [28] ([2] in O3) can not solve our problem. DynstrClu [22] ([3] in O3) focuses on returning the clustering result of SCAN while supporting dynamic graph updates. The input parameters are fixed and only consider one-hop neighbors. It does not work in our scenario as we need to tune input parameters to get good clustering results.