Visualizing cluster changes. As users adjust clustering parameters, QAGView produces a visualization like that in Figure 5 to help users understand how their adjustment changes an old clustering solution O_1 to a new one O_2. We display the clusters in O_1 and O_2 as two vertical stacks of rectangles, with width proportional to the number of result tuples they contain. For any two clusters $C_1 \in O_1$ and $C_2 \in O_2$ whose contents overlap, we further show a band from C_1 to C_2 with width (in the middle) proportional to the size of the overlap. How to present such relationships between clusters in the old and new solutions in a clean and informative manner is challenging. If we simply stack the clusters in an arbitrary order, lots of bands will cross, creating visual clutter. We approach this challenge in a principled way, formulating it as an optimization problem that looks for the best visual placement of clusters to minimize a measure of "clutter." We reduce this problem to bipartite graph matching, which can be solved efficiently in polynomial time. In practice, generating the optimized visualization takes only a few milliseconds in our experiments [5].

3 DEMONSTRATION

The demonstration of QAGView uses three datasets—besides the *MovieLens* data discussed in Example 1.1, we will also let users explore voting records of legislators in the U.S. Congress as well as player performance statistics in the National Basketball Association.

1. Exploring high-valued result tuples via summary clusters. As Figure 3 illustrates, from the main GUI of QAGView, users can select which database to connect to and what table to view. They can issue a SQL query and specify the clustering parameters k, L, and D (defaults are provided), and then explore the query result tuples as a list of summary clusters, each of which can be further expanded to reveal the list of result tuples therein.

2. Guiding the selection of clustering parameters. Users have the option of asking QAGView to "guide" them through the selection of clustering parameters. Given the coverage parameter L, QAGView plots how the number of clusters (k) affects the clustering quality under different settings of the diversity parameter (D), as illustrated in Figure 4. Each line plot (of different color) represents a different D. Intuitively, given D, the "knees" in the corresponding line plot help users identify good choices of k (e.g., $k = 9$ or 11 for $D = 1$), because additional clusters beyond these points would bring diminishing improvement to clustering quality. In contrast, the ranges of k within which the plot is close to linear or flat (e.g., $k > 6$ for $D = 3$) are less interesting.

3. Visualizing the evolution of clusters as parameters change. When users make some change to the clustering parameters k, L, and D, QAGView provides an option to visually compare the old and new result clusters, such as Figure 5. Recall that if an old cluster (rectangle on the left) and a new cluster (rectangle on the right) overlap in their contents, there will be a band connecting them. When the pointer hovers over an old cluster, for example in Figure 5, QAGView will highlight this cluster as well as the bands (three in this case) that connect to it, letting users easily see where its contents get regrouped under the new clustering. QAGView will also show details about the highlighted cluster, such as the number of aggregate result tuples it covers (size), the number of top-L result tuples it covers (coverage), and the average/maximum/minimum

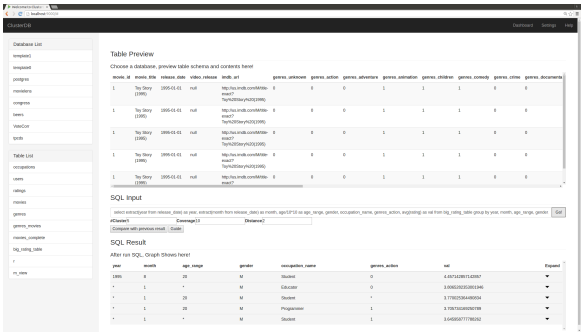

Figure 3: Main GUI of QAGView.

Figure 4: Visualization of the effect of k and D on clustering quality (for a given L), to guide the selection of clustering parameters.

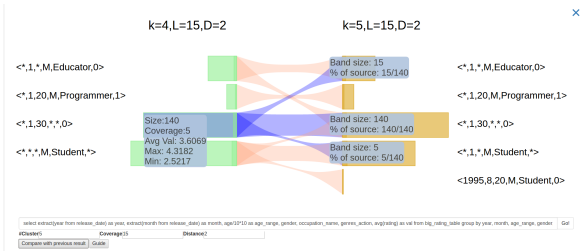

Figure 5: Visualization of how clusters evolve as parameters change.

values of the covered tuples. QAGView shows details about each band connected to the highlighted cluster as well, such as the number of result tuples shared by the source and destination clusters, and the percentage of the result tuples in the source cluster that get regrouped to the destination cluster (note that the sum of percentages over all bands connected to the highlighted cluster may exceed 100%, as destination clusters may overlap).

Acknowledgment. *This work was supported in part by NSF awards IIS-1408846, IIS-1552538, IIS-1703431, IIS-1718398, and NIH award 1R01EB025021-01.*

REFERENCES

[1] Marina Drosou and Evaggelia Pitoura. 2015. Multiple Radii DisC Diversity: Result Diversification Based on Dissimilarity and Coverage. *ACM Trans. Database Syst.* 40, 1, Article 4 (March 2015), 43 pages.
[2] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4, Article 19 (Dec. 2015), 19 pages.
[3] Manas Joglekar, Hector Garcia-Molina, and Aditya G. Parameswaran. 2016. Interactive Data Exploration with Smart Drill-Down. In *Proc. of the 2106 IEEE Intl. Conf. Data Engineering*. Helsinki, Finland, 906–917.
[4] Lu Qin, Jeffrey Xu Yu, and Lijun Chang. 2012. Diversifying Top-K Results. *Proc. of the VLDB Endowment* 5, 11 (2012), 1124–1135.
[5] Yuhao Wen, Xiaodan Zhu, Sudeepa Roy, and Jun Yang. 2018. Interactive Summarization and Exploration of Top Aggregate Query Answers. (2018). Manuscript, https://users.cs.duke.edu/~sudeepa/SummarizingAggregates.pdf.