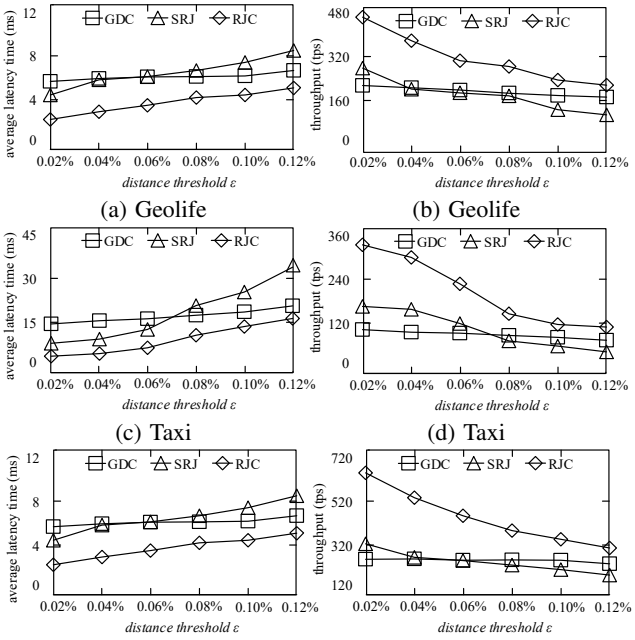

Figure 10: Clustering Performance vs. ϵ

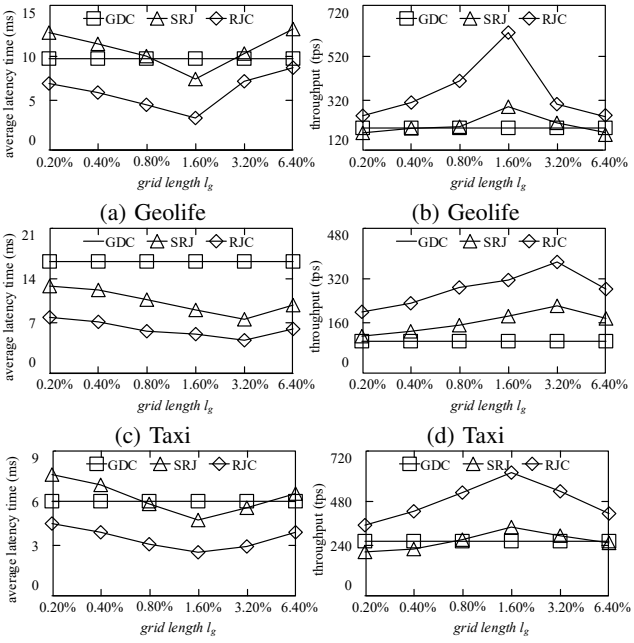

Figure 11: Clustering Performance vs. l_g

the average response time for each snapshot is reported as the latency, while the throughput is defined as the number of snapshots processed per second.

7.1 Clustering Performance

We first explore the effect of the distance threshold ϵ and the grid cell width l_g on the range join based clustering algorithm RJC, compared with the (adapted) existing methods SRJ and GDC.

Effect of ϵ. Fig. 10 depicts latency and throughput when increasing ϵ from 0.02% to 0.12%. As expected, RJC achieves better latency and throughput than SRJ, since Lemmas 1 and 2 make it possible to avoid unnecessary verifications. In addition, RJC performs better than GDC. This is because GDC uses ϵ (i.e., a small value) to divide the data space, resulting in too many partitions. Finally, the latency increases and the throughput decreases with the growth of ϵ due to the resulting larger search space.

Effect of l_g. Fig. 11 plots the latency and throughput when increasing l_g from 0.1% to 6.4%. As observed, the clustering performance (including latency and throughput) of RJC and SRJ first improves and then drops as l_g grows. The reason is that, if l_g is too small, the overhead of managing the many partitions is too high; and if l_g is too large, the punning ability decreases due to too many locations in each partition. However, the clustering performance of GDC stays stable as it does not depends on l_g.

7.2 Scalability

Next, we investigate the scalability of our pattern detection framework, where RJC is used for clustering, and three algorithms BA, FBA, and VBA (covered in Section 6) are used for pattern enumeration. This yields three corresponding methods B, F and V for pattern detection. In this set of and remaining experiments, only Taxi and Brinkhoff are employed due to similar performance on Geolife and the space limitation.

Effect of O_r. Fig. 12 shows the latency and throughput when O_r ranges from 10% to 100%. Here, the bars indicate the average pattern detection latency (including clustering and enumeration), while the curve denotes the average cluster size. The first observation is that, B can only run on small datasets, i.e., when $O_r \leq 60\%$. This is because, the cost of its enumeration method BA is $O(2^n)$, where n is the average cluster size that increases with O_r. Second, V and F can run on large datasets, and F achieves the best latency, while V achieves the best throughput. The reason is that, the costs of their enumeration methods VBA and FBA are linear w.r.t. the average cluster size, and VBA utilizes the variable bit compression and maximal pattern time sequence techniques to trade latency for throughput. As expected, the performance drops as O_r grows, due to the larger search space.

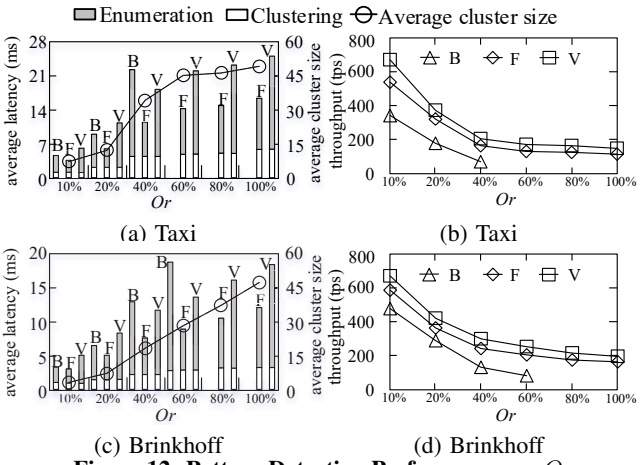

Figure 12: Pattern Detection Performance vs. O_r

Effect of ϵ. Fig. 13 illustrates the latency and throughput when ϵ ranges from 0.02% to 0.12%. As expected, the performance drops when ϵ grows. This is because, as ϵ ascends, the range join cost also increases due to the larger search space, and the enumeration cost grows due to the increasing average cluster size. Note that, the average cluster size is omitted in the remaining experiments, because it is not affected by other parameters.

Effect of N. Fig. 14 plots the latency and throughput when N ranges from 1 to 10. As expected, the average latency drops and the throughput increases as the number of nodes grows.