Christian S. Jensen · Ee-Peng Lim ·
De-Nian Yang · Wang-Chien Lee ·
Vincent S. Tseng · Vana Kalogeraki ·
Jen-Wei Huang · Chih-Ya Shen (Eds.)

LNCS 12683

Database Systems for Advanced Applications

26th International Conference, DASFAA 2021
Taipei, Taiwan, April 11–14, 2021
Proceedings, Part III

3 Part III

Springer

Contents – Part III

Recommendation

Emerging Applications

Industrial Papers

Demo Papers

Ph.D Consortium

Tutorials

FedTopK: Top-K Queries Optimization over Federated RDF Systems

Ningchao Ge[1], Zheng Qin[1(✉)], Peng Peng[1], and Lei Zou[2]

[1] Hunan University, Changsha, China
{ningchaoge,zqin,hnu16pp}@hnu.edu.cn
[2] Peking University, Beijing, China
zoulei@pku.edu.cn

Abstract. Recently, how to evaluate SPARQL queries over federated RDF systems has become a hot research topic. However, most existing studies mainly focus on implementing and optimizing the basic queries over federated SPARQL systems, and few of them discuss top-k queries. To remedy this defect, this demo designs a system named *FedTopK* that can support top-k queries over federated RDF systems. FedTopK employs a cost-based optimal query plan generation algorithm and a query plan execution optimization strategy to minimize the top-k query cost. In addition, FedTopK uses a query decomposition optimization scheme which allow merge triple patterns with the same multi-sources into one subquery to reduce the remote access times. Experimental studies over real federated RDF datasets show that the demo is efficient.

1 Introduction

In recent years, *R*esource *D*escription *F*ramework (*RDF*) has been widely used in many applications. Many data providers publish their datasets using the RDF model at their own sites, and provide the SPARQL interfaces to support users to submit SPARQL queries. In this paper, an autonomous site with a SPARQL interface is called an *RDF source*. To integrate multiple RDF sources, federated RDF systems have been proposed [2–4].

Right now, practitioners are showing a growing interest in top-k queries, which impose an order on the result set and limit the number of results. Top-k queries can be expressed in SPARQL by including the ORDER BY and LIMIT clauses. However, existing federated RDF systems can only support to alter the sequence of solution mappings after the full evaluation of the graph pattern in the WHERE clause. Therefore, this paper implement a federated RDF system, named FedTopK, which optimize evaluation of top-k queries over federated RDF systems. In summary, FedTopK has the following unique features:

- FedTopK have an incremental query execution strategy in accordance with the characteristics of top-k queries, which can greatly improve the query efficiency by terminating the execution as soon as the requested number of final results has been obtained.

C. S. Jensen et al. (Eds.): DASFAA 2021, LNCS 12683, pp. 595–599, 2021.
https://doi.org/10.1007/978-3-030-73200-4_42

- FedTopK can minimize query cost by a cost-based optimal query plan generation algorithm, which can optimize the join order of subqueries.
- FedTopK can reduce the remote access times effectively by a query decomposition scheme, which allows merge triple patterns with the same multi-sources into one subquery.

2 System Architecture and Key Techniques

Figure 1 shows the system architecture of our proposed federated RDF system FedTopK. It consists of a control site and some RDF sources. We assume that queries are submitted to the control site. The control site decomposes the query into several subqueries on relevant sources and generate a query plan. Then, the decomposed subqueries are sent to their relevant sources and executed. Last, matches of subqueries are returned to the control site and joined to form complete matches according to the query plan. In summary, there are three steps during the query processing of FedTopK: *query decomposition and source selection*, *cost-based query plan generation* and *query execution*.

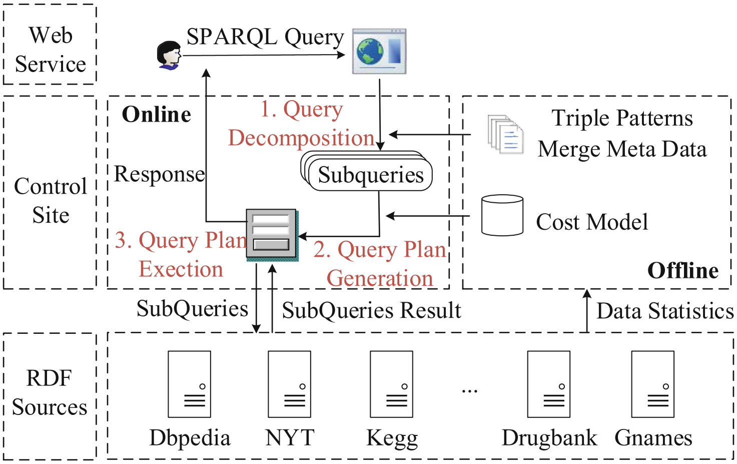
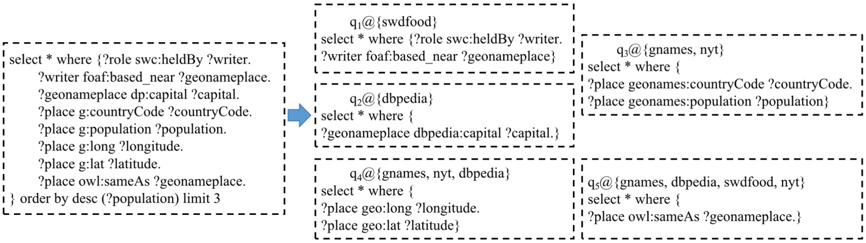


Fig. 1. Scheme for query processing in FedTopK

Query Decomposition and Source Selection. When an user submit a top-k query Q online, the query Q is decomposed into a set of subqueries, $\mathcal{Q} = \{q_1@S_1, q_2@S_2, ..., q_n@S_n\}$, where S_i is the set of relevant sources for q_i. FedTopK can merge triple patterns with the same multi-sources into one subquery by maintaining the triple patterns merge conditions from RDF sources offline. It can reduce the communication overhead effectively by reducing the number of subqueries. For example, Fig. 2 shows an example query decomposition and source selection result.



```
select * where {?role swc:heldBy ?writer.
      ?writer foaf:based_near ?geonameplace.
      ?geonameplace dp:capital ?capital.
      ?place g:countryCode ?countryCode.
      ?place g:population ?population.
      ?place g:long ?longitude.
      ?place g:lat ?latitude.
      ?place owl:sameAs ?geonameplace.
} order by desc (?population) limit 3
```

```
      q1@{swdfood}
select * where {?role swc:heldBy ?writer.
?writer foaf:based_near ?geonameplace}
```

```
      q2@{dbpedia}
select * where {
?geonameplace dbpedia:capital ?capital.}
```

```
      q3@{gnames, nyt}
select * where {
?place geonames:countryCode ?countryCode.
?place geonames:population ?population}
```

```
      q4@{gnames, nyt, dbpedia}
select * where {
?place geo:long ?longitude.
?place geo:lat ?latitude}
```

```
q5@{gnames, dbpedia, swdfood, nyt}
select * where {
?place owl:sameAs ?geonameplace.}
```

Fig. 2. Example query decomposition and source selection result

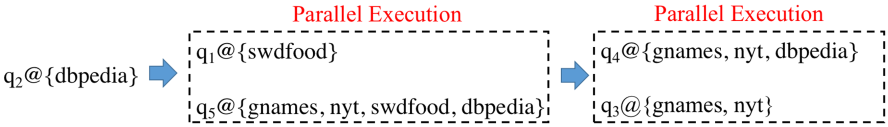


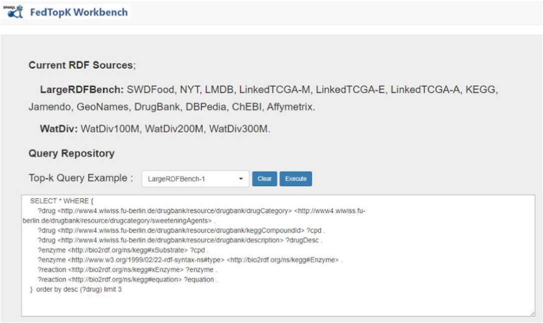
Fig. 3. Example query plan

Cost-Based Optimal Query Plan Generation. A query plan represent a join order of subqueries $\mathcal{Q} = \{q_1@S_1, q_2@S_2, ..., q_n@S_n\}$. Different query plans have different query costs. FedTopK designs a cost model to calculate the query cost and join cost of subqueries in accordance with the statistics data maintained from RDF sources offline. On this basic, the optimal query plan can be obtained by a optimal query plan generation algorithm. For example, Fig. 3 shows an example query plan for the query decomposition and source selection result, and we assume this query plan is the optimal one.

Query Execution. The query plan determines the execution order and execution mode (serial and parallel) of subqueries. For query plan in Fig. 3, subquery $q_2@\{dbpedia\}$ is executed firstly. Then, subqueries $q_1@\{swdfood\}$ and $q_5@\{gnames, dbpedia, swdfood, nyt\}$ can be executed in parallel, and so on. Among that, we propose an optimization in accordance with the characteristics of top-k query. During query execution, when a subquery containing the top-k constraint is executed, its results are sorted and incrementally used to generate the final results in order. The execution can stops as soon as the requested number of final results has been obtained.

3 Demonstration

In this demo, we use two famous comprehensive RDF benchmark suites, LargeRDFBench [5] and WatDiv [1], to show the demonstration of FedTopK. The federated RDF system FedTopK can efficiently support both SPARQL basic queries and top-k queries. More demonstrations can be referred with http://47.111.92.242:8080/FedTopK/Demo/index.html.



Current RDF Sources;

LargeRDFBench: SWDFood, NYT, LMDB, LinkedTCGA-M, LinkedTCGA-E, LinkedTCGA-A, KEGG, Jamendo, GeoNames, DrugBank, DBPedia, ChEBI, Affymetrix.

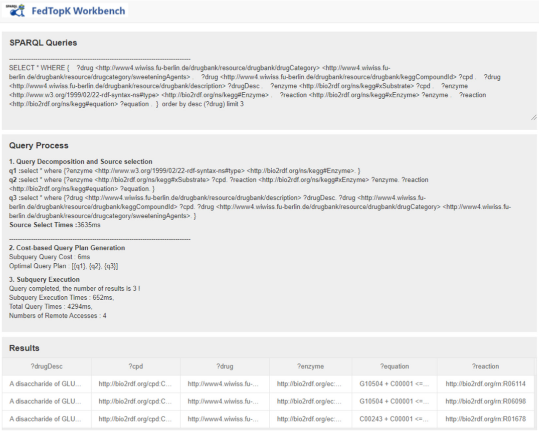
WatDiv: WatDiv100M, WatDiv200M, WatDiv300M.

Query Repository

Top-k Query Example : LargeRDFBench-1

```
SELECT * WHERE {
    ?drug <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/drugCategory> <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugcategory/sweeteningAgents> .
    ?drug <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/keggCompoundId> ?cpd .
    ?drug <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/description> ?drugDesc .
    ?enzyme <http://bio2rdf.org/ns/kegg#xSubstrate> ?cpd .
    ?enzyme <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://bio2rdf.org/ns/kegg#Enzyme> .
    ?reaction <http://bio2rdf.org/ns/kegg#xEnzyme> ?enzyme .
    ?reaction <http://bio2rdf.org/ns/kegg#equation> ?equation .
 }  order by desc (?drug) limit 3
```

Fig. 4. Query page of FedTopK



SPARQL Queries

```
--------------------------------------------------------------------------------
SELECT * WHERE {    ?drug <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/drugCategory> <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugcategory/sweeteningAgents> .    ?drug <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/keggCompoundId> ?cpd .    ?drug <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/description> ?drugDesc .    ?enzyme <http://bio2rdf.org/ns/kegg#xSubstrate> ?cpd .    ?enzyme <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://bio2rdf.org/ns/kegg#Enzyme> .    ?reaction <http://bio2rdf.org/ns/kegg#xEnzyme> ?enzyme .    ?reaction <http://bio2rdf.org/ns/kegg#equation> ?equation .  }  order by desc (?drug) limit 3
```

Query Process

1. Query Decomposition and Source selection
q1 :select * where {?enzyme <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://bio2rdf.org/ns/kegg#Enzyme>. }
q2 :select * where {?enzyme <http://bio2rdf.org/ns/kegg#xSubstrate> ?cpd. ?reaction <http://bio2rdf.org/ns/kegg#xEnzyme> ?enzyme. ?reaction <http://bio2rdf.org/ns/kegg#equation> ?equation. }
q3 :select * where {?drug <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/description> ?drugDesc. ?drug <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/keggCompoundId> ?cpd. ?drug <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/drugCategory> <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugcategory/sweeteningAgents>. }
Source Select Times :3635ms

----

2. Cost-based Query Plan Generation
Subquery Query Cost : 6ms
Optimal Query Plan : [{q1}, {q2}, {q3}]

3. Subquery Execution
Query completed, the number of results is 3 !
Subquery Execution Times : 652ms,
Total Query Times : 4294ms,
Numbers of Remote Accesses : 4

Results

?drugDesc	?cpd	?drug	?enzyme	?equation	?reaction
A disaccharide of GLU...	http://bio2rdf.org/cpd:C...	http://www4.wiwiss.fu-...	http://bio2rdf.org/ec:...	G10504 + C00001 <=...	http://bio2rdf.org/rn:R06114
A disaccharide of GLU...	http://bio2rdf.org/cpd:C...	http://www4.wiwiss.fu-...	http://bio2rdf.org/ec:...	G10504 + C00001 <=...	http://bio2rdf.org/rn:R06098
A disaccharide of GLU...	http://bio2rdf.org/cpd:C...	http://www4.wiwiss.fu-...	http://bio2rdf.org/ec:...	C00243 + C00001 <=...	http://bio2rdf.org/rn:R01678

Fig. 5. Query result page of FedTopK

Figure 4 and Fig. 5 demonstrate the two main pages of FedTopK. Users can enter a SPARQL top-k query or select a query statement from the query sample list in Fig. 4. In the top of Fig. 5, FedTopK shows the detail query process including current SPARQL query statement, the set of subqueries after query decomposition, the optimal query plan and the value of query performance indicators. Finally, the query results can be found in the bottom of Fig. 5.

4 Conclusion

`FedTopK` is a federated RDF system that can support top-k SPARQL queries. It can improve query performance by a cost-based optimal query plan generation algorithm and a query plan execution optimization strategy. It also reduces the remote requests by a query decomposition optimization.

Acknowledgment. This work was supported by the National Natural Science Foundation of China under Grant (No. U20A20174, 61772191), Science and Technology Key Projects of Hunan Province (2019WK2072, 2018TP3001, 2018TP2023,), ChangSha Science and Technology Project (kq2006029), National Key Research and Development Program of China under grant 2019YFB1406401 and Key Research and Development Program of Hubei Province (No. 2020BAB026).

References

1. Aluç, G., Hartig, O., Özsu, M.T., Daudjee, K.: Diversified stress testing of RDF data management systems. In: Mika, P., et al. (eds.) ISWC 2014. LNCS, vol. 8796, pp. 197–212. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11964-9_13
2. Montoya, G., Skaf-Molli, H., Hose, K.: The *Odyssey* approach for optimizing federated SPARQL queries. In: d'Amato, C., et al. (eds.) ISWC 2017. LNCS, vol. 10587, pp. 471–489. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68288-4_28
3. Peng, P., Ge, Q., Zou, L., Özsu, M.T., Xu, Z., Zhao, D.: Optimizing multi-query evaluation in federated RDF systems. TKDE (2019)
4. Quilitz, B., Leser, U.: Querying distributed RDF data sources with SPARQL. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 524–538. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-68234-9_39
5. Saleem, M., Hasnain, A., Ngomo, A.N.: LargeRDFBench: a billion triples benchmark for SPARQL endpoint federation. J. Web Semant. **48**, 85–125 (2018)

Shopping Around: *CoSurvey* Helps You Make a Wise Choice

Qinhui Chen[1], Liping Hua[1], Junjie Wei[1], Hui Zhao[1,2(✉)], and Gang Zhao[3]

[1] Software Engineering Institute, East China Normal University, Shanghai, China
[2] Shanghai Key Laboratory of Trustworthy Computing, Shanghai, China
hzhao@sei.ecnu.edu.cn
[3] Microsoft, Beijing, China
gang.zhao@microsoft.com

Abstract. When shopping online, customers usually compare commodities with each other before making their purchase decision. In addition to the product price, they also concern the word-of-mouth. However, marketing strategies from various e-commerce platforms, along with the diverse online commodities, make it difficult for customers to distinguish the most cost-effective products. Present cross-platform commodity comparison applications merely focus on product prices, without jointly concerning the reviews. In this demonstration, we developed a web-based application, *CoSurvey*, which matches commodities from various e-commerce platforms and analyzes product comment sentiment on the base of the proposed Attention-BiLSTM-CNN Model. The model uses an attention-based Bi-LSTM network to learn sentence sequence information, uses a CNN to learn sentence structure information, and uses a multilayer perceptron (MLP) to learn meta-information. The meta-information in the comment sentiment analysis task includes comment's *like number, reviewer level, additional image, deliver time, and sentence length*. Besides the keyword query, *CoSurvey* provides customers a survey of cross-platform products price changing trends and comment sentiment evolutions. The high concurrency requirements and load balance are also concerned.

Keywords: Sentiment analysis · Entity resolution · E-commerce · Multiple neural network · Attention mechanism

1 Introduction

With the permeation of online shopping, customers usually shop around on different e-commerce platforms. Besides the price and the brand, product reviews play a decisive role in the final purchase decision making as they can reflect on customer's preferences for the product. However, the inconsistency of cross-platform product descriptions, along with massive product reviews, bring about overwhelming information overload. During shopping festivals such as Singles Day (11.11) and Black Friday, this situation aggravates further.

C. S. Jensen et al. (Eds.): DASFAA 2021, LNCS 12683, pp. 600–603, 2021.
https://doi.org/10.1007/978-3-030-73200-4_43

Although there exist some cross-platform commodity comparison applications, such as Kelkoo[1], and Goggle Product Search[2], these applications only focus on the price comparison and overlook the product review analysis jointly. Therefore, we develop *CoSurvey*, which surveys the product information from different e-commerce platforms to help the customer make a wise shopping decision. The application meets two challenges: (1) Product matching, which aims to align the same product of different e-commerce platforms; (2) Product review sentiment analysis, which concerns not only comment text but also its valuable meta information, such as when the review is delivered, how many consumers agree with the review, etc. Since different platforms have different comment meta information, we normalize them by extracting comments' *like number, reviewer level, additional image, deliver time, and sentence length.*

Our implementations can be summarized as follows:

- We propose and train a deep fusion neural network - Attention-BiLSTM-CNN model – which takes both comment and its meta-information to classify the sentiment polarity. The experimental results demonstrate that our model's precision achieves 94.48%, recall is 94.29%, F1 value is 94.38%.
- We also train Attention-BiLSTM-CNN model to calculate the product pairs' match possibility. The blocking strategy[2] is applied to decrease complexity.
- The system concerns high concurrency. Linux Virtual Server and multi-Nginx servers are employed to implement the load balance.

2 *CoSurvey* System Overview and Key Techniques

As shown in Fig. 1, *CoSurvey* system consists of five layers: data layer, NLP layer, business layer, gateway layer, and visual layer. NLP layer provides key techniques for product matching and review sentiment analysis task, which mainly includes:

Data Pre-Process. Data pre-processing steps include filtering out missing values, normalizing comment meta information and product names, etc.

Sentiment Classification. The Attention-BiLSTM-CNN model is applied to predict product review sentiment. The model will be detailed in Sect. 2.1.

Product Matching. An Attention-BiLSTM-CNN model is trained to match products from different platforms. The input of the model is a pair of product descriptions from different platforms. The output is the possibility of whether the descriptions identify the same product. The blocking strategy is used to divide the whole dataset into several subsets by the product brand. The best-matched pairs are stored in the database.

CoSurvey crawls different platforms' commodity information and stores them in MongoDB. Elasticsearch[3] and Redis[4] are used to moderate database pressure. *CoSurvey* meets high concurrency requirements. We employ multiple LVS and Nginx servers in Openresty[5] to implement the load balance. *CoSurvey* also pro-

[1] https://www.kelkoo.co.uk/.

[2] https://shopping.google.com/?nord=1.

[3] https://www.elastic.co/cn/elasticsearch/.

[4] https://redis.io/.

[5] http://openresty.org/cn/.

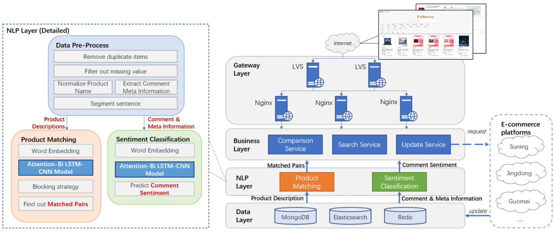


Fig. 1. The framework of *CoSurvey*

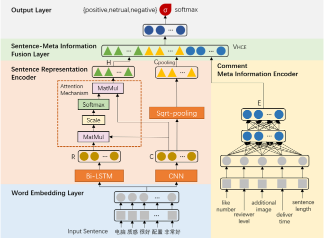


Fig. 2. Attention-BiLSTM-CNN model architecture

vides customers an interactive web-based interface to browse all commodities or search for a specific commodity using keywords or product characteristics. For both query modes, *CoSurvey* presents all the selling links of the commodity and gives out a detail comparison of its review and price. According to these comparisons, customers can obtain the latent relationship between promotion and product feedback.

2.1 Attention-BiLSTM-CNN Model

Model Structure. The model consists word embedding layer, sentence representation encoder (SRE), comment meta information encoder (CMIE), sentence-meta information fusion layer, and output layer, as is shown in Fig. 2.

The word embeddings are initialized by ERNIE [6] model, which has demonstrated outperform BERT in Chinese corpus. The word embedding was fine-tuned during the training process. In SRE, we fuse CNN and BiLSTM via attention mechanism [3] to fully utilize sentence structural and sequential information. Specifically, Bi-LSTM output R and CNN output C are used to calculate the attention result $H = softmax(\frac{K^TQ}{\sqrt{d}}V)$ (where $K, V = C, Q = R$). C is also fed into an sqrt-pooling layer to obtain the pooling result $C_{pooling}$. In CMIE, we obtain high-dimensional meta-information representation E through MLP. Then, the fusion layer concatenates the outputs of SRE and CMIE, forming a fusing representation $V^{HCE} = [H; \;\; C_{pooling}; \;\; E]$. Finally, V^{HCE} is fed into a fully-connected layer with $softmax$ to obtain the sentiment polarity.

Experiments. We compare our model to CNN [1] model, LSTM [4] model, and AT-LSTM [5] model. The experiment results show that our model reaches 94.48% in precision and outperforms other models by 1.65%. We also perform an ablation study, which shows that meta-information makes an improvement by 0.4% in precision in the sentiment classification task.

3 System Demonstration

We provide customers a highly interactive demonstration of our system. Figure 3 shows the main scenarios of the demo. (1) Customers can shop around commodities from different e-commerce platforms and search for products using keywords or product characteristics. (2) Customers can overview the price, the comment number, and the feedback rate distribution of the products. (3) Customers can browse the detailed comparison information for a specific product, like the current lowest price, the word cloud, and the price trends of this product. (4) Customers can view the product sentiment evolution of the product, where the static surveyed emotion, the dynamic emotional tendency, and the supported or inconsistent feedback rates are presented.

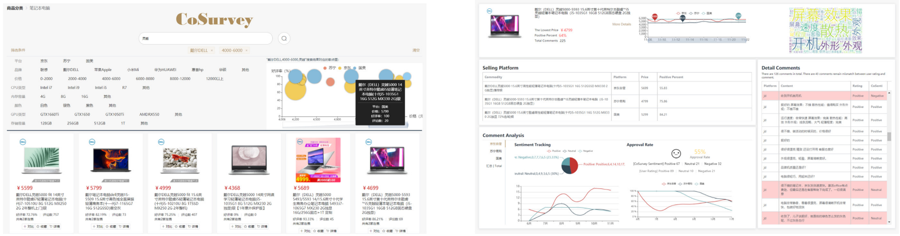


Fig. 3. System Demonstration

4 Conclusion

In this demonstration, we develop a distributed application called *CoSurvey* to survey the product information across different e-commerce platforms. We apply Attention-BiLSTM-CNN Model to implement both the sentiment analysis task and the product matching task. *CoSurvey* provides cross-platform product information survey service to help customers make a wise purchase decision. The application also provides operators insightful feedback to improve the production and the marketing strategy.

Acknowledgements. This work is supported by National Key Research and Development Program (2019YFB2102600).

References

1. Johnson, R., Zhang, T.: Effective use of word order for text categorization with convolutional neural networks. In: NAACL, pp. 103–112 (2015)
2. O'Hare, K., Jurek-Loughrey, A., de Campos, C.: An unsupervised blocking technique for more efficient record linkage. Data Knowl. Eng. **122**, 181–195 (2019)
3. Vaswani, A., et al.: Attention is all you need. In: NIPS, pp. 5998–6008 (2017)
4. Wang, X., Liu, Y., Sun, C.J., Wang, B., Wang, X.: Predicting polarities of tweets by composing word embeddings with long short-term memory. In: IJCNLP, pp. 1343–1353 (2015)
5. Wang, Y., Huang, M., Zhu, X., Zhao, L.: Attention-based LSTM for aspect-level sentiment classification. In: EMNLP, pp. 606–615 (2016)
6. Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., Liu, Q.: ERNIE: enhanced language representation with informative entities. In: ACL, pp. 1441–1451 (2019)

IntRoute: An Integer Programming Based Approach for Best Bus Route Discovery

Chang-Wei Sung[1], Xinghao Yang[2](✉), Chung-Shou Liao[1], and Wei Liu[2]

[1] Department of Industrial Engineering, National Tsing Hua University, Hsinchu, Taiwan
sung103034036@gapp.nthu.edu.tw, csliao@ie.nthu.edu.tw

[2] School of Computer Science, University of Technology Sydney, Ultimo, Australia
xinghao.yang@student.uts.edu.au, wei.liu@uts.edu.au

Abstract. An efficient data-driven public transportation system can improve urban potency. In this research, we propose IntRoute, an Integer Programming (IP) based approach to optimize bus route planning. Specifically, IntRoute first contracts bus stops via clustering and then derives a new bus route via a mixed integer linear program (ILP). This two-phase strategy brings three major merits, i.e., a single bus route without any transfer, the minimal total time consuming, and an efficient optimization algorithm for large-scale problems. Experimental results show that our IntRoute significantly reduces the traditional commuting time in Sydney from 31.53 min down to 18.06 min on average.

1 Introduction

In this study, we consider an important data-driven public transportation problem: finding the best bus route that minimizes passengers' overall commuting time. Given a bus transportation system as well as the requests of specific passengers who commute from different starting locations to a fixed destination, the goal of the problem is designing a new bus route to satisfy the passengers' demand without any transfer.

Previous studies on bus transfer problems were mostly not data-driven [2] due to data skewness problems [3]. In this work, we proposed a new integer-programming based method, which we call IntRoute, to find the route that minimizes the total time cost of the targeted passengers. Specifically, our IntRoute contains two main phases, i.e., the contraction of bus stops via K-means clustering and the derivation of new bus route via a mixed integer linear programming (ILP). The major contributions of this research are listed below.

- We design a single bus route in which all passengers with an identical destination are delivered without any transfers.
- We present a two-phase framework to minimize the total time expense of the specific passengers.
- We develop a genetic algorithm (GA) to solve the integer linear programming (ILP) for large-scale instances.

C. S. Jensen et al. (Eds.): DASFAA 2021, LNCS 12683, pp. 604–607, 2021.
https://doi.org/10.1007/978-3-030-73200-4_44

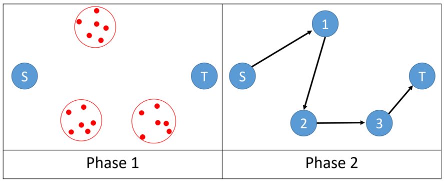


Fig. 1. The framework of the IntRoute.

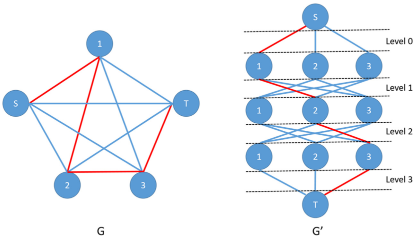


Fig. 2. Graph transformation.

2 Methodology

The main framework of our two-phase IntRoute method are shown in Fig. 1.

Phase 1: Contraction of Bus Stops. In the first phase of IntRoute, we contract multiple requests into a super node by using clustering approaches and determine a pickup bus stop for all the requests inside the super node. In each super node, passengers are asked to walk to the pickup bus stop and wait for buses. The walking time is counted into the total commuting time. We employ K-means clustering, as it consumes the least walking time compared with hierarchical clustering and density-peaks clustering. Then we exploit silhouette method and elbow method to determine the number of pickup stops, i.e., the cluster number n. Both methods indicate the best clustering number should be $n = 20$. Therefore, this phase finds 20 pickup stations that minimizes the passengers total walking time.

Phase 2: Design of One Alternate Route. In the real-world transportation network, an arbitrary node pair (i, j) may be connected (Fig. 2 left). To solve the problem via mathematical IP model, we introduce the *multi-level graph* G' (Fig. 2 right), where each level represents a possible sub-route from one stop to the next one. The red paths in G and G' are equal. The graph G' indicates the order of visiting the pick-up stops directly by the levels.

Modelling via Integer Programming. We denote the node set as $N = \{1, 2, \ldots, n\} \cup \{S, T\}$, where S and T represents the source and destination, respectively. The arc set is $A = \{(i, j) | i \in N,\ j \in N,\ \text{and } i \neq j\}$. The travel time from node i to node j is denoted as c_{ij}, and r_i is the number of passengers who want to go to the destination. $x^l_{ij} \in \{0, 1\}$ represents a binary variable, indicating whether the bus goes from node i to node j at level l on G'. $y^l_{ijk} \in \{0, 1\}$ represents a binary variable which denotes if request k travels through arc (i, j) at level l. $v^l_k \in \{0, 1\}$ represents a binary variable, indicating if request k is served at the level l. Formally, the IP model is formulated as:

$$\min \sum_l \sum_k \sum_{(i,j) \in A} y^l_{ijk} c_{ij} r_k, \quad s.t.$$

$$\textstyle\sum_{(i,j) \in A} x^l_{ij} = 1, for\ l = 0, 1, \ldots, n \tag{1}$$

$$\textstyle\sum_{(j,i) \in A} x^l_{ji} = \sum_{(i,k) \in A} x^{l+1}_{ik}, \forall i; \forall l \tag{2}$$

$$\textstyle\sum_{(S,i)\in A} x^0_{Si} = \sum_{(i,j)\in A} x^1_{ij} \tag{3}$$

$$\textstyle\sum_{(i,j)\in A} x^{n-1}_{ij} = \sum_{(j,T)\in A} x^n_{jT} \tag{4}$$

$$\textstyle\sum_i \sum_l x^l_{ij} \leq 1, for \quad j = 1, 2, \ldots, n, T \tag{5}$$

$$v^l_k \leq v^{(l+1)}_k, \forall k, \forall l \tag{6}$$

$$\textstyle\sum_k v^l_k = l, for\ l = 1, 2, \ldots, n \tag{7}$$

$$x^l_{ij} \leq v^l_i, for\ (i,j) \in A, l = 1, 2, \ldots, n \tag{8}$$

$$y^l_{ijk} \leq (x^l_{ij} + v^l_k)/2,\ \forall (i,j) \in A, \forall k, \forall l \tag{9}$$

$$y^l_{ijk} \geq (x^l_{ij} + v^l_k) - 1,\ \forall (i,j) \in A, \forall k, \forall l \tag{10}$$

$$x^l_{ij} \in \{0,1\};\ v^l_k \in \{0,1\};\ y^l_{ijk} \in \{0,1\} \tag{11}$$

where (1) ensures that each level is exactly passed once. (2)–(4) ensure that flow conservation of the graph. (5) ensures that each node can be entered at most once. (6) ensures that request k must be served at level $l+1$ if it is served at level l. (7) ensures that l requests are on the bus when the bus is serving level l. (8) ensures that if arc (i,j) is picked at level l, request i should be served at level l. (9) and (10) ensure that the request k is served at level l on the arc (i,j) if both x^l_{ij} and v^l_k are equal to one. (11) shows x^l_{ij}, v^l_k, and y^l_{ijk} are all binaries.

Optimization via Genetic Algorithm. We design a genetic algorithm (GA) to solve this IP problem. Specifically, the *chromosome* represents possible sequence of the node set $\{1, 2, \ldots, n\}$, and the *population* is a set of chromosomes.

As shown in Algorithm 1, a new chromosome can be generated by the *crossover* between two parent bus routes with a probability of crossover rate R_c. The *mutation* is defined as the position exchange between two randomly selected near-by bus stoops with a probability of R_m. There are two steps in our GA: (1) the randomly population initialization with a given size M, and (2) the population evolution for T generations by crossover and mutation according to a fitness function. We adopt a 2-OPT technique to avoid the cross sub-paths. Besides, we propose a decomposition technique that clusters the optimal route into three sub-route via k-means. We concatenate the clusters in a reverse order, i.e., from the destination node to the start node. This strategy greatly reduces the travel time.

Algorithm 1: Genetic Algorithm for Solving the IP Problem

```
Input: $T = 1000,\ R_c = 0.1,\ R_m = 0.05$
Output: The chromosome with the best
        fitness function
Initialize the population with size
  $M = 500$;
for $i = 1$ to $T$ do
    Select new population $P_i$ from $P_{i-1}$;
    for individual $p \in P_i$ do
        $offspring \longleftarrow Crossover(p, R_c)$;
        $offspring \longleftarrow$
          $Mutate(offspring, R_m)$;
        $p \longleftarrow offspring$;
    end
end
return The best chromosome.
```

Table 1. Routing time before optimization

Bus route	C'	C
2153142, 2155252, 2148445, 2153226, 2121125, 2074117, 212225, 211220, 211118, 2137134, 212746, 2150145, 2145561, 2150302, 2190145, 2135206, 213447, 203833, 200721, 201635, CBD	33.62 min	31.53 min

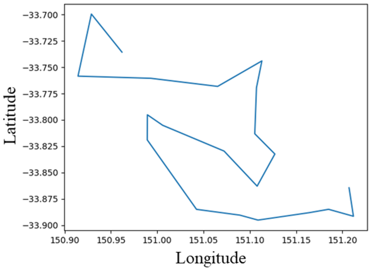


Fig. 3. Bus route without transfer

Table 2. Commute time

Methods	Time
Original route	31.53
Greedy algorithm [1]	53.41
GA	31.27
GA + 2-OPT	33.62
GA + decomposition	18.06

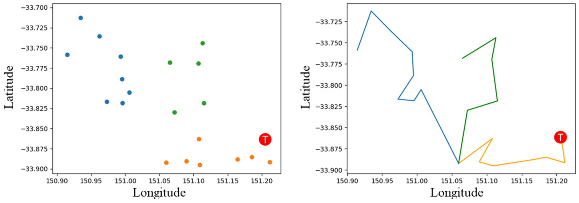


Fig. 4. Bus routes after optimization

3 Experiments and Analysis

Data. The experiment is performed based on publicly available real-world commuting data, retrieved from the card-based transit payment system[1] in Sydney, Australia, including approximately three million trips.

Results. The routing time before optimization are listed in Table 1, where routes are represented by the IDs of the bus stops. Here C denotes the time cost without transfer and without optimization (also demoed in Fig. 3), and C' denotes that of the original commute with transfers. The new route after our optimization is shown in Fig. 4, while the time costs of the bus routes optimized by different methods are listed in Table 2.

Conclusions. Our IntRoute method greatly reduces the time expense for passengers from 31.53 min to 18.06 min on average, saving about 43% of commute time. In future, we plan to investigate more optimization methods for further improving our solutions.

References

1. Li, L., Fu, Z.: The school bus routing problem: a case study. J. Oper. Res. Soc. **53**(5), 552–558 (2002). https://doi.org/10.1057/palgrave.jors.2601341
2. Liu, J., Mao, J., Du, Y.T., Zhao, L., Zhang, Z.: Dynamic bus route adjustment based on hot bus stop pair extraction. In: Li, G., Yang, J., Gama, J., Natwichai, J., Tong, Y. (eds.) DASFAA 2019. LNCS, vol. 11448, pp. 562–566. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-18590-9_87
3. Liu, W., Chawla, S.: A quadratic mean based supervised learning model for managing data skewness. In: Proceedings of SIAM SDM Conference (2011)

[1] https://opendata.transport.nsw.gov.au/dataset/opal-tap-on-and-tap-off.

NRCP-Miner: Towards the Discovery of Non-redundant Co-location Patterns

Xuguang Bao[1], Jinjie Lu[1], Tianlong Gu[1], Liang Chang[1](✉), and Lizhen Wang[2]

[1] Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology,
Guilin 541004, China
changl@guet.edu.cn

[2] Yunnan University, Kunming 650091, China

Abstract. Co-location pattern mining, which refers to discovering neighboring spatial features in geographic space, is an interesting and important task in spatial data mining. However, in practice, the usefulness of prevalent (interesting) co-location patterns generated by traditional frameworks is strongly limited by their huge amount, which may affect the user's following decisions. To address this issue, in this demonstration, we present a novel schema, named NRCP-Miner, aiming at the redundancy reduction for prevalent co-location patterns, i.e., discovering non-redundant co-location patterns by utilizing the spatial distribution information of co-location instances. NRCP-Miner can effectively remove the redundant patterns contained in prevalent co-location patterns, thus furtherly assists the user to make the following decisions. We evaluated the efficiency of NRCP-Miner compared with related state-of-the-art approaches.

Keywords: Spatial data mining · Co-location pattern mining · Prevalent co-location patterns · Redundancy reduction · Decision-making system

1 Introduction

The explosive growth of the spatial data results in significant demand for spatial data mining. Co-location pattern mining, as an important spatial data mining task, has been extensively studied for discovering neighboring relationships of spatial features. A spatial co-location pattern commonly demonstrates neighboring relationships of spatial features. Spatial co-location patterns may yield important insights in many applications, including Earth Science, public health, biology, transportation, etc.

To measure how interesting a co-location pattern is, the PI (Participation Index) value proposed by Huang et al. [1] is commonly used. Given a user-specified minimum prevalence threshold *min_prev*, for a co-location pattern c, if $PI(c) \geq min\_prev$ satisfies, c is called a prevalent co-location pattern (PCP). As a PCP is a set of spatial features, given a spatial dataset containing m spatial features, the number of generated PCPs can reach as much as 2^m. Furthermore, the PI measure satisfies the anti-monotonicity property [1], i.e., if a PCP c is prevalent, all its subsets are also prevalent. However, most of its subsets are redundant by considering their prevalences or PI values, which

C. S. Jensen et al. (Eds.): DASFAA 2021, LNCS 12683, pp. 608–611, 2021.
https://doi.org/10.1007/978-3-030-73200-4_45

may affect the decisions of the user. Thus, it is crucial to reduce the number of PCPs by redundancy reduction.

To reduce the number of PCPs, two classic condensed representations have been proposed—maximal co-location patterns [2] (MCPs) and closed co-location patterns [3] (CCPs), respectively. However, MCPs are considered as a lossy representation because they ignore the PI values of co-location patterns. Although CCPs are lossless representations considering both prevalences and PI values of co-location patterns, they contain redundancies. Thus, Wang et al. [4] proposed an algorithm called RRClosed to select non-redundant co-location patterns from CCPs. Later, they introduced a new lossless and non-redundant representation called SPI-closed (Super Participation Index-closed) co-location patterns (SCPs), and proposed a method called SPI-Miner [5] to efficiently discover SCPs.

In this demonstration, we present a novel and efficient system, named NRCP-Miner, to discover SCPs. Instead of RRClosed or SPI-Miner, we adopt a clique-based approach [6] to discover PCPs, and then furtherly select SCPs. Because the clique-based approach constructs a hash structure that can be stored permanently and is independent of the prevalence threshold, our proposed system performs more efficiently than RRClosed and SPI-Miner, especially when the system needs to be executed multiple times. Besides, as SCPs are subsets of PCPs, our proposed NRCP-Miner can be applied to domains of PCPs. For example, the mobile service provider may be interested in mobile service SCPs frequently requested by geographical neighboring users. Botanists may be interested in SCPs consisting of symbiotic plant species.

2 System Overview

NRCP-Miner undergoes six steps to generate SCPs, as shown in Fig. 1.

Step 1: *Materialization of the inputted spatial data.* This step first gathers all neighboring relationships of each instance by considering a user-given distance threshold *min_dist*, and then groups the neighboring relationships as a neighbor list.
Step 2: *Generation of complete cliques.* This step aims to generate complete cliques using the neighbor list. As the enumeration of maximal cliques is considered as an NP-hard problem, we adopt a linear method [6] to generate complete cliques.
Step 3: *Compression of the complete cliques.* As the calculation of the PI value of a co-location c is only based on the instances participating in c, thus, the complete cliques can be compressed into a hash structure.
Step 4: *Generation of PCPs.* Given the instance hash, the PI value of any co-location pattern can be efficiently calculated by considering the user-specified prevalence threshold *min_prev*.
Step 5: *Selection of CCPs.* As the CCPs are subsets of PCPs, thus, all CCPs can be selected from PCPs by the definition of CCPs, i.e., removing the PCP whose PI value equals the PI value of one of its supersets.
Step 6: *Generation of SCPs.* To generate the SCPs from CCPs, we adopt the latter part of the RRClosed method [4], which generates SCPs from CCPs by designing a NET structure and a lemma for pruning.

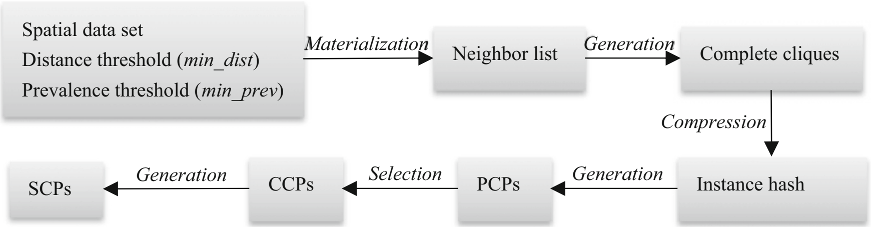
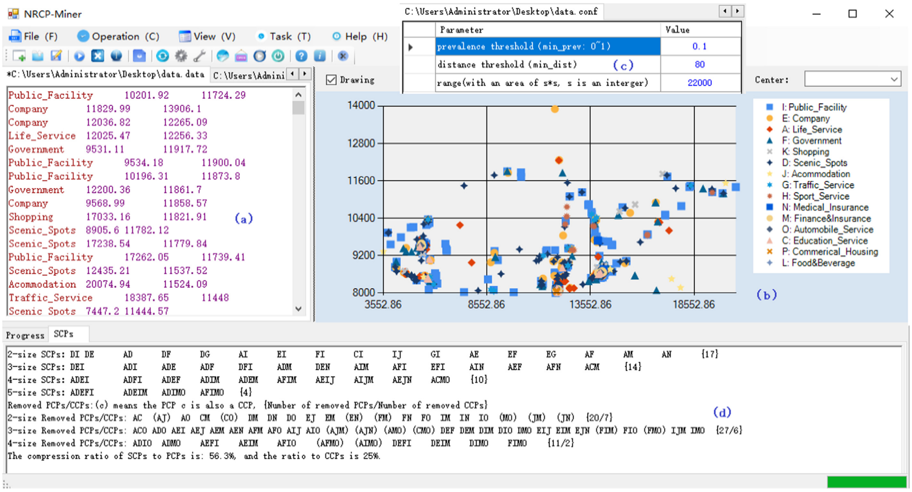


Fig. 1. System description

3 Demonstration Scenarios

NRCP-Miner is well encapsulated with a friendly interface, what the user faces is only a simple user interface. In this demonstration, we use part of the data set from points of interests (POI data) in Beijing to show the demonstration and efficiency of NRCP-Miner. The selected POI data set contains 5,000 POIs (spatial instances).

Demonstration. Figure 2 shows the main interface of NRCP-Miner. Figure 2(a) gives the original spatial instances read from a file or a database, each instance is represented as <feature name, location <x, y>>. The detailed distribution of instances described in Fig. 2(a) is drawn in Fig. 2(b). The parameters with their specified values are listed in Fig. 2(c). Figure 2(d) shows the generated SCPs based on the settings in Fig. 2(c) from the spatial data shown in Fig. 2(a), as well as the number of per-size SCPs and removed CCPs.



```
Public_Facility      10201.92      11724.29
Company      11829.99      13906.1
Company      12036.82      12265.09
Life_Service 12025.47      12256.33
Government   9531.11       11917.72
Public_Facility      9534.18       11900.04
Public_Facility      10196.31      11873.8
Government   12200.36      11861.7
Company      9568.99       11858.57
Shopping     17033.16      11821.91
Scenic_Spots 8905.6 11782.12
Scenic_Spots 17238.54      11779.84
Public_Facility      17262.05      11739.41
Scenic_Spots 12435.21      11537.52
Acommodation 20074.94      11524.09
Traffic_Service      18387.65      11448
Scenic_Spots 7447.2 11444.57
```

C:\Users\Administrator\Desktop\data.conf

Parameter	Value
prevalence threshold (min_prev: 0~1)	0.1
distance threshold (min_dist)	80
range(with an area of s*s, s is an interger)	22000

```
2-size SCPs: DI DE    AD    DF    DG    AI    EI    FI    CI    IJ    GI    AE    EF    EG    AF    AM    AN    {17}
3-size SCPs: DEI      ADI   ADE   ADF   DFI   ADM   DEN   AIM   AFI   EFI   AIN   AEF   AFN   ACM   {14}
4-size SCPs: ADEI     ADFI  ADEF  ADIM  ADEM  AFIM  AEIJ  AIJM  AEJN  ACMO  {10}
5-size SCPs: ADEFI    ADEIM ADIMO AFIMO {4}
Removed PCPs/CCPs:(c) means the PCP c is also a CCP, {Number of removed PCPs/Number of removed CCPs}
2-size Removed PCPs/CCPs: AC  (AJ)  AO  CM  (CO)  DM  DN  DO  EJ  EM  (EN)  (FM)  FN  FO  IM  IN  IO  (MO)  (JM)  (JN)  {20/7}
3-size Removed PCPs/CCPs: ACO ADO AEI AEJ AEM AEN AFM AFO AIJ AIO (AJM) (AJN) (AMO) (CMO) DEF DEM DIM DIO DMO EIJ EIM EJN (FIM) FIO (FMO) IJM IMO  {27/6}
4-size Removed PCPs/CCPs: ADIO  ADMO   AEFI   AEIM   AFIO   (AFMO)  (AIMO)  DEFI   DEIM   DIMO   FIMO   {11/2}
The compression ratio of SCPs to PCPs is: 56.3%, and the ratio to CCPs is 25%.
```

Fig. 2. Demonstration of NRCP-Miner

Efficiency Evaluations. We evaluated the efficiency of NRCP-Miner from two aspects: the compression ratio to CCPs and the running time compared with RRClosed and SPI-Miner. As shown in Fig. 2(d), NRCP-Miner removes 56.3% of PCPs, and 25% of CCPs, and also runs faster than RRClosed and SPI-Miner with the change of the prevalence threshold *min_prev*, as shown in Fig. 3, this is because the hash structure generated by NRCP-Miner is independent of *min_prev*, while the other algorithms have to restart their mining processes with the change of *min_prev*.

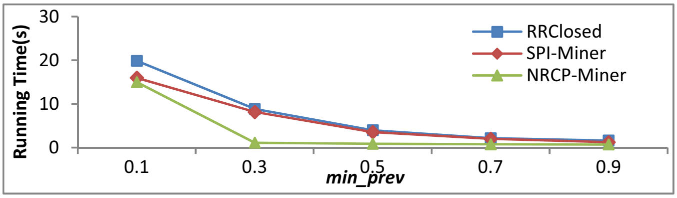


Fig. 3. Efficiency comparison with related literature

4 Conclusion

This demonstration presents a novel and efficient system called NRCP-Miner to discover a newly proposed lossless condensed representation of prevalent co-locations SPI-closed co-locations. Unlike similar approaches mainly focusing on pruning strategies for reducing the number of candidates by using the prevalence threshold, NRCP-Miner gets rid of the constraint of the prevalence threshold. Thus, it can effectively assist the user to find a satisfying prevalence threshold within much less time, and furtherly can well support the decision-making of the user.

Acknowledgements. This work was supported in part by grants (No. 62006057, No. U1811264, No. U1711263, No. 61966009, No. 61762027) from the National Natural Science Foundation of China, in part by National Science Foundation of Guangxi Province (No. 2019GXNSFBA245059), in part by the Key Research and Development Program of Guangxi (No. AD19245011).

References

1. Huang, Y., Shekhar, S., Xiong, H.: Discovering co-location patterns from spatial data sets: a general approach. IEEE Trans. Knowl. Data Eng **16**(12), 1472–1485 (2004)
2. Wang, L., Zhou, L., Lu, J., et al.: An order-clique-based approach for mining maximal co-locations. Inf. Sci. **179**(2009), 3370–3382 (2009)
3. Yoo, J.S., Bow, M.: Mining top-k closed co-location patterns. In: IEEE International Conference on Spatial Data Mining and Geographical Knowledge Services, pp. 100–105 (2011)
4. Wang, L., Bao, X., Zhou, L.: Redundancy reduction for prevalent co-location patterns. IEEE Trans. Knowl. Data Eng. **30**(1), 142–155 (2018)
5. Wang, L., Bao, X., Chen, H., Cao, L.: Effective lossless condensed representation and discovery of spatial co-location patterns. Inf. Sci. **436–437**, 197–213 (2018)
6. Bao, X., Wang, L.: A clique-based approach for co-location pattern mining. Inf. Sci. **490**, 244–264 (2019)