



Time Series Similarity Search Based on Positive and Negative Query

Jimin Wang^(✉), Qi Liu^(✉), and Pengcheng Zhang^(✉)

College of Computer and Information, Hohai University, Nanjing 211100, China
{wangjimin, pchzhang}@hhu.edu.cn, 1034188416@qq.com

Abstract. Traditional time series similarity search, based on relevance feedback, combines initial, positive and negative relevant series directly to create new query sequence for the next search; it can't make full use of the negative relevant sequence, even results in inaccurate query results due to excessive adjustment of the query sequence in some cases. In this paper, time series similarity search based on separate relevance feedback is proposed, each round of query includes positive query and negative query, and combines the results of them to generate the query results of each round. For one data sequence, positive query evaluates its similarity to the initial and positive relevant sequences, and negative query evaluates its similarity to the negative relevant sequences. The final similar sequences should be not only close to positive relevant series but also far away from negative relevant series. The experiments on UCR data sets showed that, compared with the retrieval method without feedback and the commonly used feedback algorithm the proposed method can improve accuracy of similarity search on some data sets.

Keywords: Time series similarity search · Relevance feedback
Positive query · Negative query

1 Introduction

Data mining is the process of extracting knowledge from massive data. In reality, most of the data are time series, so it has important theoretical and practical significance to mine potential useful knowledge from the time series data [1]. Time series data mining mainly includes classification, clustering, sequence pattern matching, similarity search and prediction. In many time, similarity search is the important foundation of the others, and it was proposed by Agrawal in 1993 [2] to find similar pattern for the given pattern in time series. Similarity search can help us to make useful decisions, for example, we can find a similar sales pattern in the sales records of various commodities to make a sales strategy [3], and we can forecast the natural disaster by searching similar precursor of natural disasters [4].

The traditional time series similarity search extracts data features firstly to reduce the data dimension [5–7], and then building index for them [8, 9]. Finally, based on similarity measure function [10–13], the sequences similar to the query sequence are retrieved in the index structure and displayed to the users. However, in the beginning of search, users usually can't describe the query sequence clearly, so once search is unable

to find the suitable similar sequences for users. Feedback-based strategy allows users to express their satisfaction or dissatisfaction on the query results, and make multiple queries to improve the query accuracy and users' satisfaction.

Feedback technology was applied in information retrieval at first, and Keogh introduced it into the time series data mining in the literature [14], user can set different weights indicating the similarity or dissimilarity degree for the query result sequences and return them to search system, then the new query sequence for next query is generated by the feedback sequences by some strategy. Time series similarity search based on relevance feedback diversification was proposed in Literature [15], MMR [16] was used on the feedback sequences to ensure the diversity of the query results, and then new query sequence was generated by the feedback sequences.

The above time series similarity search methods combine initial, positive and negative relevant sequences directly to create new query series for the next query, it's easy to make the query sequence change too much causing the worse query results. For lack of query topics, Wang [17] proposed a negative relevance feedback method in text retrieval, only the negative relevant feedback vectors and query vector were used to do the next query. Peltonen [18] proposed a negative feedback information retrieval system based on machine learning, allowing users to make positive and negative relevance feedback directly by the interactive visual interface. Studies had shown that making full use of negative feedback sequence can improve the retrieval accuracy.

In this paper, we propose a time series similarity search method based on positive and negative relevance feedback sequences separately. The contributions of this paper are summarized as follows:

- A novel similarity search method based on relevance feedback for time series is proposed. By combining positive query and negative query, the characteristics hidden in positive and negative relevant sequences are easily extracted. Considering the multiple categories of negative relevant sequences, two strategies, single positive relevance feedback model and multi-positive relevance feedback model, can be used during negative query. The proposed method can improve the accuracy of the similarity search.
- The proposed method is validated by a set of dedicated experiments. The experimental data are included in UCR archive [22], and experiments show that, compared with non-feedback and traditional feedback methods, the proposed method can improve the accuracy of the query on some data sets of UCR archive.

The rest of the paper is organized as follows. In Sect. 2, we review the relevance feedback strategy based on vector model. In Sect. 3, we describe our proposed time series similarity search based on positive and negative query. Experiments are analyzed and discussed in Sect. 4. We conclude this paper and discuss our future work in Sect. 5.

2 Relevance Feedback Strategy Based on Vector Model

Relevance feedback is one of query extension technologies which have become one of the key technologies to improve the recall and precision in information retrieval. Relevance feedback based on vector model is generally implemented by query modification which is proposed by Van Rijsbergen [19]. For the query results, user can label the similar vectors(called positive relevance feedback vectors) or dissimilar vectors(called negative relevance feedback vectors), query vector will be modified according to the original, positive and negative relevance feedback vectors, and then used to search similar vectors again, until user is satisfied with the results, or give up the search.

2.1 Rocchio Algorithm

The classical feedback algorithm based on vector model was proposed by Rocchio in the SMART system [20], new query vector is generated following Eq. 1

$$\vec{q}_{new} = \alpha \times \vec{q} + \beta \times \frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j - \gamma \times \frac{1}{|D_{nr}|} \sum_{\vec{d}_j \in D_{nr}} \vec{d}_j \quad (1)$$

Where, \vec{q}_{new} is the new query vector, \vec{q} is the original query vector, D_r is the collection of positive relevant document vectors, D_{nr} is the collection of negative relevant document vectors, α , β and γ are the weights to control their impact on new query vector. The optimal weight values can be assessed by knowledge of the data, or experimentally.

2.2 Ide dec-hi Algorithm

Ide [21] improved the Rocchio algorithm by using the most dissimilar negative relevant vector instead of the average of negative relevant vectors, the new query vector is generated following Eq. 2.

$$\vec{q}_{new} = \alpha \times \vec{q} + \beta \times \frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j - \gamma \times \max_{\vec{d}_j \in D_{nr}} \vec{d}_j \quad (2)$$

Where, $\max_{\vec{d}_j \in D_{nr}} \vec{d}_j$ is the most dissimilar negative relevant vector. Both of the above feedback algorithms rely heavily on positive relevance feedback, but if the query topics are few so that the positive relevance vectors are few or no, the above methods will be difficult to work.

2.3 Negative Relevance Feedback Algorithm

Wang [17] proposed a negative relevance feedback algorithm for this extreme case in document retrieval. User labels the negative relevant vectors, and search system establishes a positive query (the query vector is the original query vector) and a

negative query (the query vector is composed of the negative relevant vectors), finally, the similarity is obtained by combining them following Eq. 3.

$$S_{combine}(Q, D) = S(Q, D) - \beta \times S(Q_{neg}, D) \quad (3)$$

Where, Q is the original query vector, D is the document vector to be checked, Q_{neg} is the negative query vector generated from negative vectors by some strategy, $S(Q, D)$ is the similarity measure of the positive query, $S(Q_{neg}, D)$ is the similarity measure of the negative query, $S_{combine}(Q, D)$ is the final similarity measure. β controls the impact of $S(Q_{neg}, D)$ to the final similarity measure.

3 Time Series Similarity Search Based on Positive and Negative Query

In this paper, we introduce the negative feedback strategy in time series similarity search, and propose a time series similarity search method based on positive and negative query. For the query results, user labels the positive and negative relevant sequences, and then search system makes positive query and negative query, finally combines the results of positive query and negative query to get final similar sequences, the above steps will be executed repeatedly until user is satisfied with the query results or ignores the search. This method mainly includes: query sequence modification, positive and negative query, the combination of positive and negative query.

3.1 Query Sequence Modification

Time series similarity search can also be seemed as a kind of information retrieval. In this paper, new query sequence is generated following Eq. 4 based on the Rocchio algorithm. Negative relevant sequences are used to do negative query and do not participate in the modification, so γ is set to be 0.

$$q_{new} = \alpha \times q + \beta \times \frac{1}{|S_{PR}|} \sum_{s_j \in S_{PR}} s_j \quad (4)$$

Where, q_{new} is new query sequence for next query, q is the original query sequence, S_{PR} is the set of positive relevance sequences.

3.2 Positive and Negative Query

During the query, we perform positive query and negative query for every sequence s in data sequences. Positive query computes the similarity between s and q_{new} , negative query computes the similarity between s and the positive relevant sequences.

Positive Query. For every sequence s in data sequences, some similarity measure, such as Euclidean distance, will be used to compute the similarity, identified by $Sim(q_{new}, s)$, between s and q_{new} .

Negative Query. The main problem to be solved in our algorithm is to determine the similarity, identified by $Sim(q_{neg}, s)$, between s and the negative relevance sequences. We present two strategies to solve this problem, one is the single negative relevance feedback model, the other is the multi-negative relevance feedback model.

Single Negative Relevance Feedback Model. All the negative relevance sequences were combined to generate the single average sequence q_{neg} by Eq. 5.

$$q_{neg} = \sum_{s_j \in S_{NR}} s_j * w_j \quad (5)$$

Where, S_{NR} is the set of negative relevance sequences, and $\sum_{j=1}^{|S_{NR}|} w_j = 1$. This strategy treats all the negative relevance sequences as the same category, and use the mean sequence represent the characteristics of most negative relevance sequences. The weight w_j can be set according to the dissimilarity, specified by the user's subjective setting, between the query sequence and each negative relevance sequence. For example, user can ranks all the negative feedback sequences by the dissimilarity subjectively as $s_1, s_2, s_3, \dots, s_n, n = |S_{NR}|$, then we can set $w_j = (n - j + 1) / \sum_{j=1}^{|S_{NR}|} j$. If user can't rank the negative feedback sequences, then we can generate the single average sequence q_{neg} by Eq. 6.

$$q_{neg} = \frac{1}{|S_{NR}|} \sum_{s_j \in S_{NR}} s_j \quad (6)$$

That is, $w_j = \frac{1}{|S_{NR}|}$.

Multi-negative Relevance Feedback Model. The negative relevance sequences may belong to multiple categories. Single negative feedback model can't reflect the difference of the negative relevance sequences between different categories. The multi-negative relevance model clusters the negative relevance sequences into n clusters, and merges the sequences in each cluster to generate the mean sequence $q_{neg_i}, i = 1, 2, 3, \dots, n, n = |S_{PR}|$ to represent the characteristics of each cluster independently. For every sequence s in data sequences, some similarity measure, such as Euclidean distance, will be used to compute the similarity, identified by $Sim(q_{neg_i}, s)$, between s and q_{neg_i} . We get the similarity between s and negative feedback sequences by Eq. 7.

$$Sim(q_{neg}, s) = F(Sim(q_{neg_i}, s)) \quad (7)$$

Where, F may be *MAX*, *MIN* or *AVG* operation. $MAX(Sim(q_{neg_i}, s))$ means the max similarity between s and n representative sequences is chosen to represent the final similarity. $MIN(Sim(q_{neg_i}, s))$ means the min similarity between s and n representative sequences is chosen to represent the final similarity. $AVG(Sim(q_{neg_i}, s))$ means the average similarity between s and n representative sequences is chosen to represent the final similarity.

3.3 The Combination of Positive and Negative Query

In our method, the final query results should be similar to the query sequence, but also are dissimilar to the negative relevant sequences, so the final similarity measure following Eq. 8.

$$S_c(q_{new}, s) = Sim(q_{new}, s) - \lambda \times Sim(q_{neg}, s) \quad (8)$$

$Sim(q_{new}, s)$ measures the similarity between q_{new} and the checked sequence s in positive query. $Sim(q_{neg}, s)$ measures the similarity between the negative relevance sequences and the checked sequence s in negative query, λ controls the magnitude of dissimilarity. $S_c(q_{new}, s)$ represents the final similarity degree between q_{new} and s . It can be seen that to make $S_c(q_{new}, s)$ high, either $Sim(q_{new}, s)$ is high or $Sim(q_{neg}, s)$ is low, that is, either the similarity between q_{new} and s is high or the similarity between q_{neg} and s is low.

4 Experiments and Analysis

4.1 Experimental Data

In this paper, we use 17 sets of the UCR data [22] to make experiment, and all the sequences in each data set have been labeled category. The information of each data set used in our experiments is shown in Table 1.

Table 1. Data set

No.	Data set	Number of categories	Size
1	ChlorineConcentration	3	3840
2	StarLightCurves	3	8326
3	Two_Patterns	4	4000
4	Trace	4	100
5	OliveOil	4	30
6	Haptics	5	308
7	Beef	5	30
8	OSUleaf	6	242
9	Synthetic_control	6	300
10	InlineSkate	7	550
11	Fish	7	175
12	Lighting7	7	73
13	MedicalImages	10	760
14	SwedishLeaf	15	625
15	WordsSynonyms	25	638
16	Adiac	37	391
17	50words	50	455

4.2 Method

Five experiments are performed on each data set, that is similarity searching based on no feedback, Rocchio algorithm, Ide dec-hi algorithm, single negative relevance feedback model (labeled SNRF) and multi-negative relevance feedback model (labeled MNRF). Euclidean distance is used as the distance measure.

Usually the initial query sequence can express most of user's intent, the query sequence modification only do a few fine-tuning. In order to ensure that the modified query sequence dose not deviate from the original sequence largely, the value of α of Eq. 4 is fixed to 1, and the other parameters increment from 0 to 1 by step 0.1 to find the optimal value in our experiments. In our experiments, the negative relevant sequences are clustered according to the category of sequence directly in multi-negative relevance feedback model, and in single negative relevance feedback model, Eq. 6 is used to generate the representative sequence.

In order to validate our proposed single negative and multi-negative relevance feedback similarity search model, k NN and leave-one-out cross validation are performed. P-R value, which is the integration of recall and precision, is used to assess the quality of the query, P represents the accuracy rate and R represents the recall rate. For each query sequence, k NN is performed to find r , which is called recall number, relevant sequences. P-R value of data set for r_j , noted as PR_{r_j} , is calculated by the following Eq. 9.

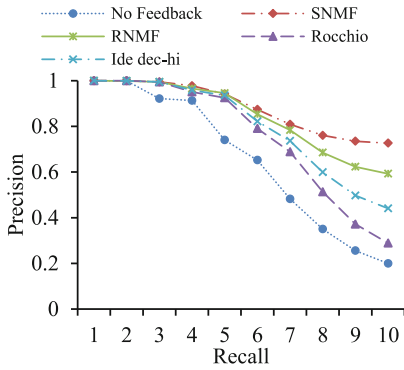
$$PR_{r_j} = \frac{1}{N_q} \sum_{i=1}^{N_q} \frac{r_j}{k_i} \quad (9)$$

where N_q is the number of query sequences, k_i is the nearest neighbor number for the i -th query sequence to find r_j relevant sequences. The P-R average value of data set is the average of all PR_{r_j} , where $r_j= 1,2,\dots,10$. In the similarity search with feedback, for each query sequence, we carried out two times feedback, the optimal P-R average value and the corresponding parameters will be retained.

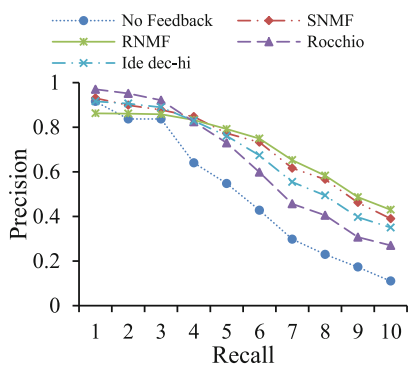
4.3 Discussion and Comparison

Figure 1 shows the precision-recall curve (according to the optimal P-R average value shown in Tables 2, 3, 4, 5 and 6) of some data sets, it can be seen that, precision-recall of the methods based on feedback are obviously better than that of no feedback. When the number of recall is small, the precision of SNRF and MNRF are similar to that of Rocchio and Ide dec-hi. However, when the number of recall increases, the precision of SNRF and MNRF is better than that of Rocchio and Ide dec-hi.

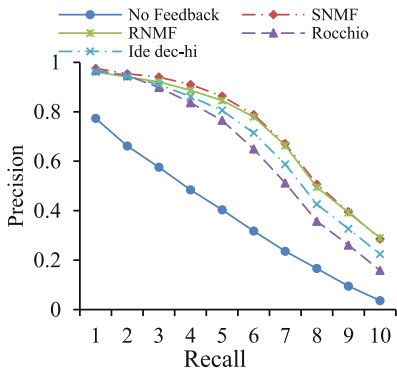
Tables 2, 3, 4, 5 and 6 show the optimal P-R average value of the methods used in our experiments, and the corresponding parameters. Figure 2 shows the P-R average value of 17 data sets. It can be seen that, compared to the no feedback, the similarity search with feedback can obviously improve the accuracy. The results of single-negative feedback model and multi-negative feedback model are all better than that of Rocchio and Ide dec-hi except the results on Two_Patterns and 50words, but the results of five methods on Two_Patterns and 50words are very close. There are some



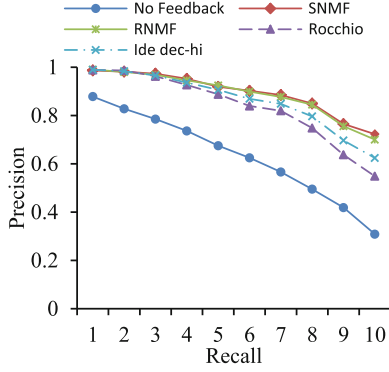
(a) ChlorineConcentration



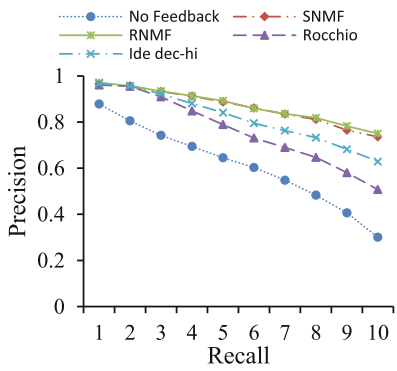
(b) Trace



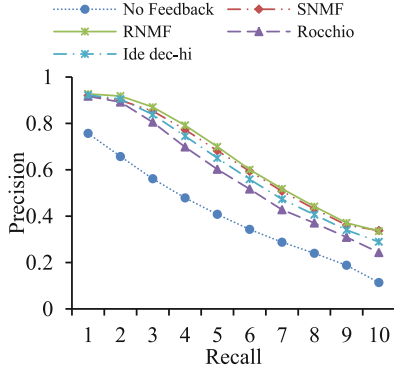
(c) OSUleaf



(d) Fish



(e) SwedishLeaf



(f) Adiac

Fig. 1. Precision-recall

Table 2. P-R of no feedback model

No.	Name of data sets	P-R value
1	ChlorineConcentration	0.651
2	StarLightCurves	0.829
3	Two_Patterns	0.869
4	Trace	0.501
5	OliveOil	0.669
6	Haptics	0.242
7	Beef	0.259
8	OSUleaf	0.375
9	Synthetic_control	0.817
10	InlineSkate	0.205
11	Fish	0.631
12	Lighting7	0.403
13	MedicalImages	0.553
14	SwedishLeaf	0.611
15	WordsSynonyms	0.501
16	Adiac	0.404
17	50words	0.479

Table 3. P-R of single negative relevance feedback model

No.	Name of data sets	P-R value	β	λ
1	ChlorineConcentration	0.881	0.7	1
2	StarLightCurves	0.927	0.8	1
3	Two_Patterns	0.959	0.9	1
4	Trace	0.698	0.9	0.7
5	OliveOil	0.872	0.7	0.5
6	Haptics	0.565	0.8	1
7	Beef	0.470	0.8	0.9
8	OSUleaf	0.728	0.8	0.8
9	Synthetic_control	0.962	0.6	0.3
10	InlineSkate	0.453	0.7	0.6
11	Fish	0.893	0.9	0.8
12	Lighting7	0.685	0.9	0.8
13	MedicalImages	0.805	0.9	0.9
14	SwedishLeaf	0.866	0.8	0.8
15	WordsSynonyms	0.711	0.9	0.7
16	Adiac	0.630	0.7	0.5
17	50words	0.662	0.9	0.7

Table 4. P-R of multi-negative relevance feedback model

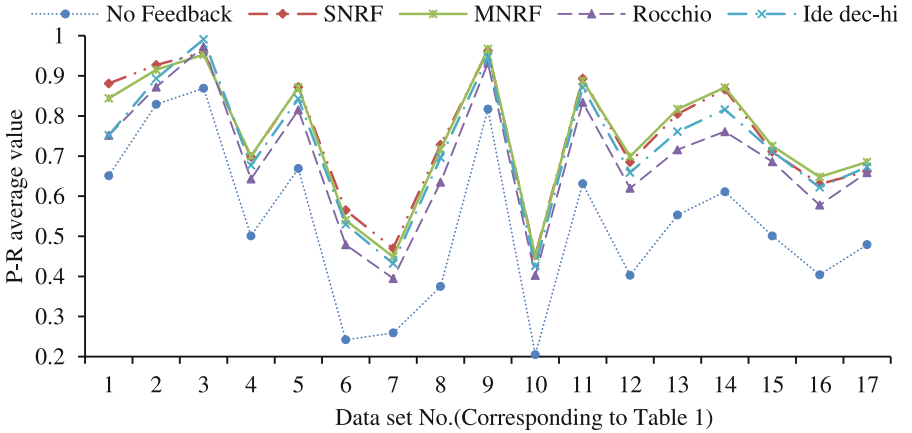
No.	Name of data sets	P-R value	β	λ
1	ChlorineConcentration	0.844	0.7	0.3
2	StarLightCurves	0.915	0.9	1
3	Two_Patterns	0.953	0.7	1
4	Trace	0.700	0.9	0.1
5	OliveOil	0.871	0.8	0.5
6	Haptics	0.540	0.9	0.5
7	Beef	0.449	0.8	0.2
8	OSUleaf	0.717	0.7	0.3
9	Synthetic_control	0.967	0.8	0.5
10	InlineSkate	0.450	0.9	0.1
11	Fish	0.888	0.7	0.5
12	Lighting7	0.699	0.9	0.1
13	MedicalImages	0.817	0.8	0.3
14	SwedishLeaf	0.871	0.9	0.7
15	WordsSynonyms	0.725	0.8	0.1
16	Adiac	0.648	0.9	0.5
17	50words	0.685	0.8	0.1

Table 5. P-R of feedback based on Rocchio

No.	Name of data sets	P-R value	β	γ
1	ChlorineConcentration	0.752	0.7	0.7
2	StarLightCurves	0.872	0.9	0.7
3	Two_Patterns	0.973	0.8	0.6
4	Trace	0.643	0.9	0.5
5	OliveOil	0.815	0.8	0.5
6	Haptics	0.479	0.7	0.3
7	Beef	0.395	0.8	0.6
8	OSUleaf	0.635	0.7	0.3
9	Synthetic_control	0.932	0.9	0.5
10	InlineSkate	0.403	0.7	0.3
11	Fish	0.834	0.9	0.4
12	Lighting7	0.620	0.7	0.7
13	MedicalImages	0.716	1	0.4
14	SwedishLeaf	0.761	0.8	0.6
15	WordsSynonyms	0.686	0.9	0.8
16	Adiac	0.578	0.8	0.5
17	50words	0.659	0.9	0.4

Table 6. P-R average value of feedback based on Ide dec-hi

No.	Name of data sets	P-R value	β	γ
1	ChlorineConcentration	0.752	0.7	0.7
2	StarLightCurves	0.893	0.9	0.6
3	Two_Patterns	0.991	0.9	0.7
4	Trace	0.677	0.6	0.4
5	OliveOil	0.843	0.7	0.4
6	Haptics	0.530	0.6	0.4
7	Beef	0.432	0.8	0.6
8	OSUleaf	0.696	0.8	0.5
9	Synthetic_control	0.947	0.6	0.5
10	InlineSkate	0.426	0.7	0.2
11	Fish	0.871	0.9	0.4
12	Lighting7	0.659	0.8	0.5
13	MedicalImages	0.761	1	0.3
14	SwedishLeaf	0.816	0.9	0.7
15	WordsSynonyms	0.712	0.9	1
16	Adiac	0.622	0.8	0.6
17	50words	0.673	1	0.2

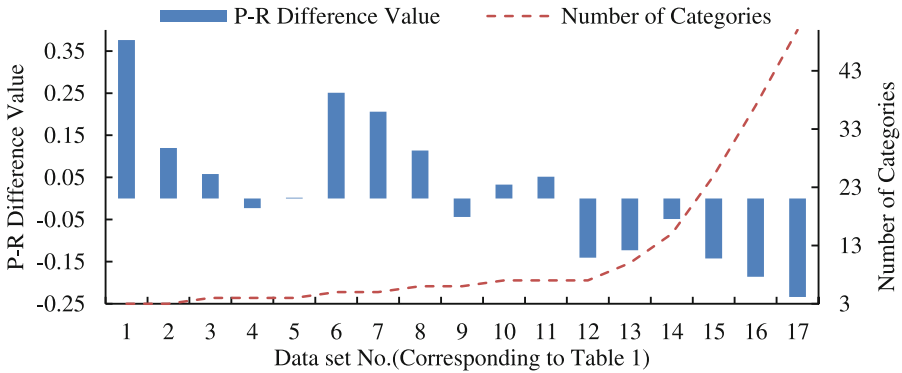
**Fig. 2.** P-R average value of data sets

differences between single negative feedback model and multi-negative feedback model, at the confidence level of 0.05, the Wilcoxon symbol rank sum test shows $p = 0.9758$ (bilateral test), indicating that for the 17 data sets, single negative feedback model and multi-negative feedback model performs basically the same.

However, in Table 7 and Fig. 3, it can be seen that when the number of categories is relatively small, the single negative relevance feedback model is better than the multi-negative relevance feedback model. When the categories is more, the multi-

Table 7. P-R difference value between SNF and MNF

No.	Name of data sets	P-R difference value (P-R of SNRF - P-R of MNRF)	Number of Categories
1	ChlorineConcentration	0.376	3
2	StarLightCurves	0.120	3
3	Two_Patterns	0.058	4
4	Trace	-0.023	4
5	OliveOil	0.002	4
6	Haptics	0.251	5
7	Beef	0.206	5
8	OSUleaf	0.114	6
9	Synthetic_control	-0.044	6
10	InlineSkate	0.033	7
11	Fish	0.052	7
12	Lighting7	-0.140	7
13	MedicalImages	-0.123	10
14	SwedishLeaf	-0.049	15
15	WordsSynonyms	-0.143	25
16	Adiac	-0.186	37
17	50words	-0.234	50

**Fig. 3.** P-R difference value and Number of categories of data set

negative relevance feedback model is better than single negative relevance feedback model, it maybe because the single average sequence q_{neg} can't represent the characteristics of the negative relevance sequences fully when the negative relevance sequences are more dispersed, that is, the number of categories is more.

5 Conclusion

Based on relevance feedback, we proposed a time series similarity search method which fully utilizes the positive and negative relevance sequences. The positive relevant sequences and negative relevant sequences are used to search similar sequences independently, after that, the similarity degree of positive query and negative query are combined to get the final similarity. Experiments show that the proposed method can improve the accuracy of the query on some data sets of UCR archive. Furtherly, we can study a more principled strategy to use the negative feedback sequences to conduct constrained query.

Acknowledgment. This research is supported by the Key Technologies Research and Development Program of China (2015BAB07B01), the National Natural Science Foundation of China (No. 61572171).

References

1. Wang, Y., Xu, C.: Data mining technology. *Electron. Technol. Softw. Eng.* **2015**(8), 204–205 (2015)
2. Agrawal, R., Faloutsos, C., Swami, A.: Efficient similarity search in sequence databases. In: Lomet, D.B. (ed.) *FODO 1993*. LNCS, vol. 730, pp. 69–84. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-57301-1_5
3. Luo, H.: Based on gray-ARIMA financial time series intelligent hybrid forecasting. *Finan. Econ. Theory Pract.* **35**(2), 27–34 (2014)
4. Zhu, Y., Li, S., Fan, Q.: Prediction of hydrological time series based on wavelet neural network. *J. Shandong Univ. (Eng. Sci.)* **41**(4), 119–124 (2011)
5. Li, Z., Guo, J., Hui, X.: Based on the common principal component of the multivariate time series dimensionality reduction method. *Control Decis.* **2013**(4), 531–536 (2013)
6. Li, H.: Research on feature representation and similarity measurement in time series data mining. Dalian University of Technology (2012)
7. Xiao, R.: Study on dimensionality reduction and similarity matching of uncertain time series. Donghua University (2014)
8. Li, Z., Zhang, F., Li, K.: A multi-time series index structure supporting DTW distance. *J. Softw.* **25**(3), 560–575 (2015)
9. Dai, K.: Research on time series query method based on linear hash index. *Softw. Eng.* **19**(8), 1–8 (2016)
10. Zhang, Q., Zhao, Z.: Z tree: an index structure for high-dimensional data. *Comput. Eng.* **33**(15), 49–51 (2007)
11. Xiao, R., Liu, G.: Research on time series similarity measure and clustering based on trend. *Appl. Res. Comput.* **31**(9), 2600–2605 (2014)
12. Zhang, H., Li, Z., Sun, Y.: New time series similarity measure method. *Comput. Eng. Des.* **35**(4), 1279–1284 (2014)
13. Goldin, D.Q., Millstein, T.D., Kutlu, A.: Bounded similarity querying for time-series data. *Inf. Comput.* **194**(2), 203–241 (2004)
14. Keogh, E.J., Pazzani, M.J.: An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In: 4th International Conference of Knowledge Discovery and Data Mining, pp. 27–31. ACM Press, New York (1998)

15. Eravci, B., Ferhatosmanoglu, H.: Diversity based relevance feedback for time series search. *Proc. VLDB Endow.* **7**(2), 109–120 (2013)
16. Carbonell, J., Goldstein, J.: The use of MMR, diversity-based reranking for reordering documents and producing summaries. In: 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Melbourne, pp. 335–336. ACM (1998)
17. Wang, X., Fang, H., Zhai, C.X.: A study of methods for negative relevance feedback. In: International ACM SIGIR Conference on Research and Development in Information Retrieval, Singapore, pp. 219–226 (2008)
18. Peltonen, J., Strahl, J., Floréen, P.: Negative relevance feedback for exploratory search with visual interactive intent modeling. In: 22nd International Conference on Intelligent User Interfaces, Raphael Resort, Limassol, pp. 149–159 (2017)
19. Van Rijsbergen, C.J.: A new theoretical framework for information retrieval. *ACM SIGIR Forum* **21**(1–2), 23–29 (1986)
20. Rocchio, J.J.: Relevance feedback in information retrieval. In: Salton, G. (ed.) *The SMART System: Experiments in Automatic Document Processing*, pp. 313–323. Prentice-Hall, Englewood Cliffs (1972)
21. Ide, E.: New experiments in relevance feedback. In: *The SMART System: Experiments in Automatic Document Processing*, pp. 337–354. Prentice-Hall (2000)
22. The UCR Time Series Classification Archive. http://www.cs.ucr.edu/~eamonn/time_series_data/