Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 31 No. 2s (2024)

Embedding Hybrid Evolutionary Approach for Learning-to-Rank Computation for the Selection of Features Using Machine Learning

Sushilkumar Chavhan[1*], R. C. Dharmik [2]

1,2 Department of Information Technology, Yeshwantrao Chavan College of Engineering ,Nagpur, Maharashtra,

schavhansushil@gmail.com[1],raj_dharmik@yahoo.com[2]

Article History:

Received: *20-03-2024*

Revised: *12-05-2024*

Accepted: *25-05-2024*

Abstract:

Our study proposes a novel model for retrieving objects that utilizes learning-to-rank with L2 regularization. We employed an evolutionary-based simulated annealing technique to select the most informative features for our system and utilized a standardized regulation technique to handle the dropout of active features. Learning to rank is a well-researched area in machine learning and finds application in recommendation systems and search engines. Our study aims to introduce a new approach to feature selection for the learning-to-rank information retrieval model. By dropping inactive features and maintaining active features, we can improve the ranking function's performance. We tested our proposed method on standard datasets and repeatedly improved the feature selection model of the LambdaMart algorithm. Empirical performance results show that our heuristic model provides better feature subset combinations, as measured by the NDCG, P@10, and MAP evolutionary metrics, than do baseline databases. Our proposed method surpasses existing learning-to-rank methods, paving the way for promising future research.

Keywords: Machine Learning, L2 regularization ,learning-to-rank, simulated annealing.

1.Introduction

Ranking refers to the process of ordering a set of items based on a particular criterion or relevance to a given context or query. In the context of information retrieval, ranking involves arranging a set of web pages or documents based on their relevance to a user's query. In other domains, ranking can involve ordering products, services, or individuals based on factors such as popularity, quality, or value. The goal of ranking is to present the most valuable or relevant items at the top of the list, making it easier for users to find what they are looking for.

Various feature selection models and ranking frameworks have been developed and proposed recently. In [4], the authors discussed the significance of features and the role of similarity information in ranking and proposed a two-stage solution using a greedy algorithm. Other models developed for the same purpose but with different hypothesis functions include RankSVM [5] and RankNet [6]. In [7], the authors investigated the boosted tree ranking model with a randomized, greedy technique. FSMRank [10] is another framework for optimized feature selection for ranking. The mRMR [11] model was suggested and used to select subgroups based on relevance and similarity. A lightweight framework for feature selection and UTI, which recommends a more reliable model and model optimization in combination with LamdaMART, was also proposed [2].

The three categories for feature selection approaches are filter, wrapper, and embedding. Filter techniques were used in [4][9–13], where a subset of features was chosen based on their quality and correlation. A similarity measure between the query and document pair values was used to determine whether a feature was stable or active after being dropped. Wrapper methods were used in the development of RankNet [6] and ListNet, but they require more computational time to locate the subset. Embedded algorithms operate with sparse rankers, and attempts have been made in some studies, such as RSRank [15] and FSMRank [8],[15-17], to regularize the number of features. Another study employed a sparse Bayesian solution in [18]. Optimization is one of the more challenging tasks in selecting features based on relevance and similarity while minimizing error.

The current study proposes a novel heuristic method that utilizes hill climbing and random walk with KNN classification to select and optimize features, outperforming existing frameworks. We suggest a hybrid model that selects a feature and optimizes it iteratively. To achieve feature selection, we utilize simulated annealing with the above mentioned techniques and apply L2 regularization for optimization. The selection of active features is based on the activation function of LamdaMart during each iteration until it converges. Evaluation matrices demonstrated significantly improved results. The following questions are the main focus of our research:

- How does the Embedding Hybrid Evolutionary Approach for Learning-to-Rank generalize across different types of ranking task?
- What are the computational costs associated with the Embedding Hybrid Evolutionary Approach for Learning-to-Rank, and how do they compare to alternative feature selection methods, particularly in terms of time complexity and resource utilization?
- What are the implications of using the Embedding Hybrid Evolutionary Approach for Learning-to-Rank in real-world scenarios, such as e-commerce platforms or search engines, and how do these insights inform practical deployment and integration strategies?
- How does the Embedding Hybrid Evolutionary Approach for Learning-to-Rank address challenges related to imbalanced datasets, noisy features, and sparse data, and what techniques can be employed to enhance its robustness and reliability in such scenarios?

In research on Evolutionary learning to rank approaches presented to aims to address the above concerns. Four Different sizes of the LOTOR and Microsoft database which are available in public. Extermination performs on with three traditional algorithms and one evolutionary algorithm. Three evaluation methods are commonly used to evaluate the model, namely accuracy, MAP (average accuracy) and NDCG (value reduction increment). Research results show that using geographic features to drive the model can improve results. The significant contributions of this paper are, in brief, as follows:

- The Embedding Hybrid Evolutionary Approach for Learning-to-Rank typically outperforms traditional feature selection methods due to its ability to adaptively select relevant features and optimize ranking models based on evolutionary processes and embedding techniques. This superiority is demonstrated through comprehensive performance evaluation metrics such as the mean average precision (MAP), normalized discounted cumulative gain (NDCG), and precision
- Effective incorporation of domain-specific knowledge into the hybrid evolutionary approach involves leveraging domain experts' insights to guide the evolutionary process and

embedding strategies. This collaboration enhances feature relevance assessment and ranking model performance, ultimately leading to more interpretable and domain-adaptive solutions.

- While the Embedding Hybrid Evolutionary Approach may entail higher computational costs than simpler methods, its scalability and efficiency are demonstrated through parallelization techniques, adaptive optimization algorithms, and distributed computing frameworks. These optimizations ensure that the approach remains viable for large-scale datasets and complex feature spaces.

The paper is organized as follows: Section II reviews prior related research and provides the motivation for the current study. In Section III, we provide an overview of feature selection for a few fundamental algorithms. Section IV outlines the detailed design and experimentation of the proposed model. The results of the number of features required in less time and how we find suitable features using the proposed framework, which yields better performance than other methods, are presented in Section V along with any limitations or shortcomings.

2.Literature Review

The majority of feature selection models that include learning to rank are employed as an effective means of managing large dimensions with the goal of gathering the ideal subset that offers the right characteristics. The following groups of feature selection techniques, including filters, wrappers, embedding, and hybrid techniques, can be generally categorized. Prior to the commencement of learning, we may identify candidate feature sets with improved performance and accuracy. It is not feasible to apply the approach to every real-time application.We have outlined much research pertaining to distinct feature selection models in this area.

Changsheng Li et. al. [20] proposed a new ranking method with L-T-R in which the feature subset is sorted with respect to ranking and its accuracy. Multiple times, the process was repeated, and the results were merged to obtain highly accurate results.The results are shown where the maximum NDCG is 0.92 with old datasets such as .GOV, and Caltech101. Evaluation on a standard dataset was suggested by the authors.

Parth Gupta et. al.[21] proposed divergence based feature selection on standard datasets. An important feature of this approach is that it is parallelized. Its performance is better in some cases than that of the baseline for all the databases. Finally, the authors applied the greedy large marginal classifier -based ranking method. Han-Jiang Lai et. al. [22] proposed the FSMRank algorithm for feature selection, which is used for ranking. The authors developed a formula with join convex optimization. This approach is useful for reducing the ranking error with feature selection.Finally, they concluded that this is a flexible framework with optimization. Andrea Gigli et. al.[13] proposed three greedy algorithms, namely, NGAS, XGAS, and HCAS. In one of the algorithms, feature selection is based on pairwise similarity with relevance. Furthermore, each algorithm considers a larger number of subset iterations. HCAS works on clustering to perform feature selection. The authors conclude the paper with the future scope of work on various dimensions with other classes of classification.

To address the issue of risk sensitivity, the number of best subsets of features and performance were calculated according to Daniel Xavier et al. [24]. Proposed multii-objective feature selection. Additionally, they reduce the multidimensionality while solving risk in many queries. The results are improved compared with those of previous studies on handling risk. The conclusions of this study suggest that multiple objective methods are useful in many domains. Mehrnoush Shirzad et al. [25] provided an overview of all feature selection methods and applied them to the FSLR framework. The exploration of filter, wrapper and embedded techniques was useful for identifying better subsets of features with respect to the hypothesis of classification. Collaborative features will help improve FS and work on many applications.Fan Cheng et. al.[26] proposed an evolutionary system based on a multiobjective function called MOFSRank. With the integration of MOEA, MOFS, MOEN and MOFS,MOFSRank was developed, and its performance improved. In the future, we will focus on listwise approaches with more evolutionary methods with different datasets. Fahandar, Mohsen et. al.[28] proposed a new framework for feature selection called analogy-based LTR, Author described the problem of FS techniques, which was explored by Fan Cheng et. al.[26]. By correlation and relief base Technique technique, the authors evaluated the real -time data and proved the necessity of FS. The weighted matrix and co-embedding can be used as future functions of the same categorical data.A. Rahangdale et al. [1] suggested a deep learning model for L1 and L2 regularization-based feature selection. The author assessed the outcomes of conventional learning in terms of feature selection methods and rankings. Furthermore, they concluded that a deep neural network design with appropriate regularization techniques can aid in the discovery of greater functionality. A. Purpura et al. [27] suggested a neural network for FSs over large-scale search models named Neural Reranker. It is proposed for the optimization of LETOR methods without changing the architecture.The training time was reduced in the proposed system, and the efficiency was significantly improved. They applied the approach to the standard datasets MSLR-WEB30K and OHSUMED.

The study concluded that this method should be used for other datasets.The above literature review suggested that feature selection is an important aspect of learning to rank, as per past studies, but can be improved by using any hybrid method.

Sr. No.	Citation Number	Approach Type	Method Name	Database Used	Evaluation
1	[20]	Listwise	New ranking method with L-T-R	.GOV, Caltech101	Evaluation on standard dataset suggested by the authors
2	[21]	Pointwise	Divergence-based feature selection, greedy large marginal classifier-based ranking	OHSUMED, HP2004, NP2004, MQ2008	Future focus on listwise approaches with evolutionary methods
3	[22]	Listwise	FSMRank algorithm for feature selection and ranking	OHSUMED, HP2004, NP2004, MQ2008	Future scope on various dimensions
4	[13]	Listwise	NGAS, XGAS, HCAS algorithms for feature selection, pairwise	OHSUMED, Letor 4.0,Y ahoo!,	Future scope of work on various dimensions with

			similarity-based feature selection		other classes of classification
5	[24]	Listwise	Multi-objective feature selection to address risk sensitivity	WEB10K, WEB30K, YAHOO!	Multiple objective methods useful in many domains
6	[25]	Listwise	Overview of all feature selection methods, exploration of filter, wrapper, and embedded techniques	OHSUMED ,TD2003, TD2004, HP2003, HP20044, NP2003, NP2004, MQ2007, MQ2008, Yahoo(SET2) LETOR2.0	Work on many applications
7	[26]	Listwise	Evolutionary system MOFSRank based on multiobjective function	NP2004,HP2004,TD2004,MQ2008, OHSUMED	Focus on listwise approaches with more evolutionary methods with different datasets
8	[28]	Listwise	Analogy-based LTR for feature selection, evaluation using correlation and relief-based techniques	Decathlon, Bundesliga, FIFA , Hotels, Uni. Rankings, Volleyball WL, Netflix	Use weighted matrix and co-embedding as future functions for categorical data
9	[1]	Listwise	Deep learning model for L1 and L2 regularization-based feature selection	TD2003, TD2004, TD2003, TD2004, NP2003, NP2004, HP2003, HP2004, MQ2007, MQ2008, MSLR-WEB30K, MSLR-WEB10K	Apply to other datasets
10	[27]	Listwise	Neural reranker for large-scale search models	MSLR-WEB30K, OHSUMED	Use for other datasets

Table1: Brief summary of feature Selection algorithms

3.Feature Selection of baseline Algorithms:

The general framework for the ranking framework is as follows:

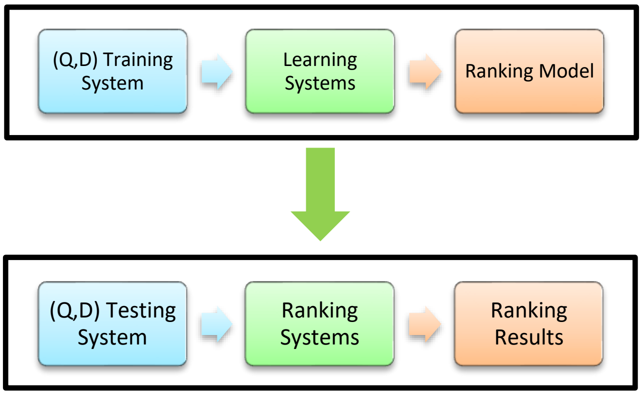


Fig. 1. Basic ranking model

As depicted in the above model diagram, we have a learning system that provides the score for the F (Q, D) pair. After receiving the score, sorting was applied, and then testing was performed.

As mentioned in the above literature survey, we used standard datasets for learning-to-rank. To determine the importance of the FSs, we used the MSLR-WEB10K Microsoft supervised database of 136 features with 10,000 queries and the Q,D pair 1200192 with five relevance labels. It is an open source dataset that provides five fold data.

To identify the best performing algorithm with minimum features, we applied the learning-to-rank algorithm to the above dataset on several algorithms, such as DirectRanker, RankNet, LamdaMart and LamdaRank. We first apply the range for the best 8 features, 16 features and so on, and find the minimum subset requirement.The experimental results of NDCG@10 with respect to the run-time (in (ms) for the abovementioned algorithms are shown below, where each time we increase the feature values.

3.1 DirectRanker Algorithm :

Table 2:NDCG@10 scores with different Feature on DirectRanker

Sr No.	Features value	NDCG@10	Average Runtime
1.	17	0.4378±0.2451	39.6070±0.3022
2.	34	0.4298±0.2459	40.4866±0.2508
3.	51	0.4286±0.2448	41.4559±0.2215
4.	68	0.4272±0.2420	43.7252±0.2889
5.	102	0.4269±0.2419	44.7012±0.2628

The experimental results by using standard procedure with leaning to rank with different number of feature is shown in the table2 represent the NDCG@10 for DirectRanker algorithm.The experiment is executed across Fold 1 of the MSLR-WEB10K dataset.

3.2RankNet Algorithm :

Table 3 - NDCG@10 scores with different Feature on RankNet

Sr No.	Features value	NDCG@10	Average Runtime
1.	17	0.3074±0.244	39.1682±0.3236
2.	34	0.266±0.2110	40.2048±0.2896
3.	51	0.2573±0.2123	40.9475±0.2972
4.	68	0.4272±0.2420	43.7252±0.2889
5.	102	0.4269±0.2419	44.7012±0.2628

The experimental results by using standard procedure with leaning to rank different number of features are shown in the table3 represent the NDCG@10 for Ranknet algorithm. The experiment is executed across Fold 1 of the MSLR-WEB10K dataset.

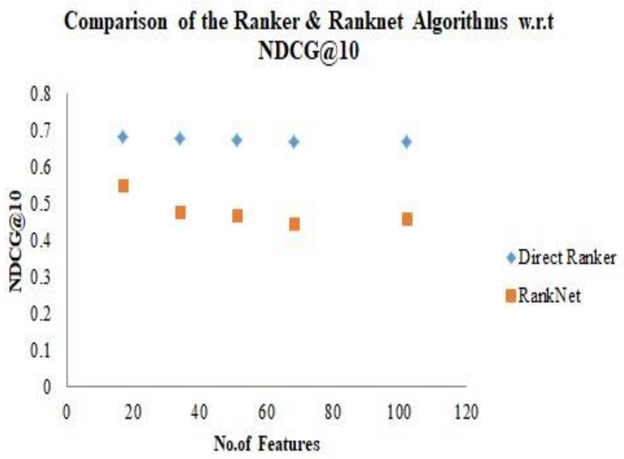


Fig2(a): Pictorial representation of NDCG@10 Ranker & Ranknet

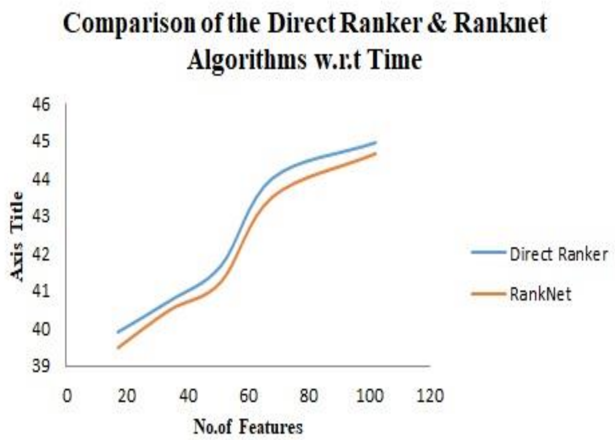


Fig2(b): Pictorial representation of Average Runtime Ranker & Ranknet

The above experimental Fig. 2(a) shows the NDCG@10 results of DirectRanker and RankNet algorithms. Fig. 2(b) shows the change in the average runtime of each execution. It is easily concluded that the direct ranker algorithm performed well compared to the rank net algorithm with the MSLR-Web10k data.

3.3LambdaMART Algorithm :

Table 4 NDCG@10 scores with different Feature on LambdaMART

Sr No.	Features value	NDCG@10	Average Runtime
1.	8	0.6240±0.2729	22.6980±0.1765
2.	16	0.6214±0.0974	24.0207±0.1923
3.	24	0.6176±0.2665	26.2134±0.1308
4.	32	0.6152±0.2632	26.7485±0.1930
5.	40	0.6143±0.27940	28.1716±0.0872
6.	46	0.6050±0.2603	29.1436±0.1652

The experimental results by using standard procedure with leaning to rank different number of features is shown in the table4 represent the NDCG@10 for Ranknet algorithm. The experiment is executed across Fold 1 of the MSLR-WEB10K dataset with across sample dataset of shape (9630, 48).

3.4LambdaRank Algorithm :

Table 5 NDCG@10 scores with different Feature on LambdaMART			
Sr No.	Features value	NDCG@10	Average Runtime
1.	8	0.7504±0.2160	21.7811±0.0925
2.	16	0.7461±0.2244	21.8933±0.6690
3.	24	0.7434±0.22195	22.0993±0.0969
4.	32	0.7427±0.2118	22.3138±0.0674
5.	40	0.7356±0.2184	22.4164±0.0802
6.	46	0.7342±0.2153	22.5089±0.0428

The experimental results by using standard procedure with leaning to rank different number of features are shown in the table4 represent the NDCG@10 for Ranknet algorithm. The experiment is executed across Fold 1 of the MSLR-WEB10K dataset with across sample dataset of shape (9630, 48).

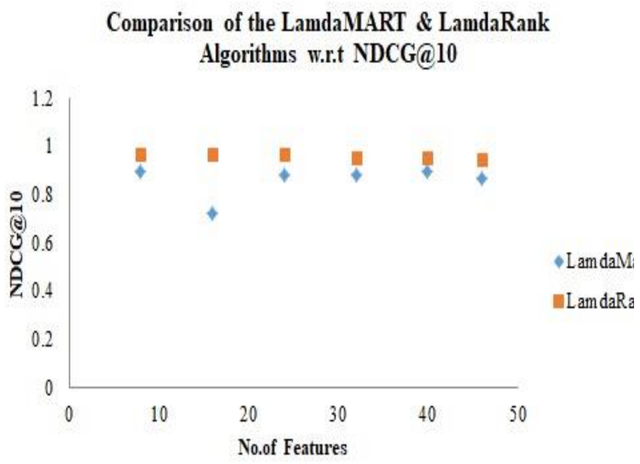


Fig3(a): Pictorial representation of NDCG@10 of LamdaMART & LamdaRank

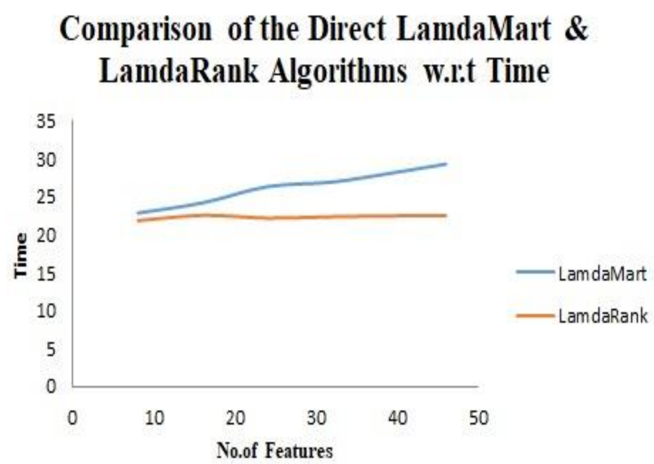


Fig3(b): Pictorial representation of Average Runtime of LamdaMART & LamdaRank

The above experiments clearly reveal that the number of features influences the efficiency of the machine learning model. Based on the results shown in both Fig. 3(a) and Fig. 3(b), both halves of the total feature set will yield better performance in less time. This can also be verified on different datasets.

4.Evolutionary Learning with LamdaMart

4.1 Architecture

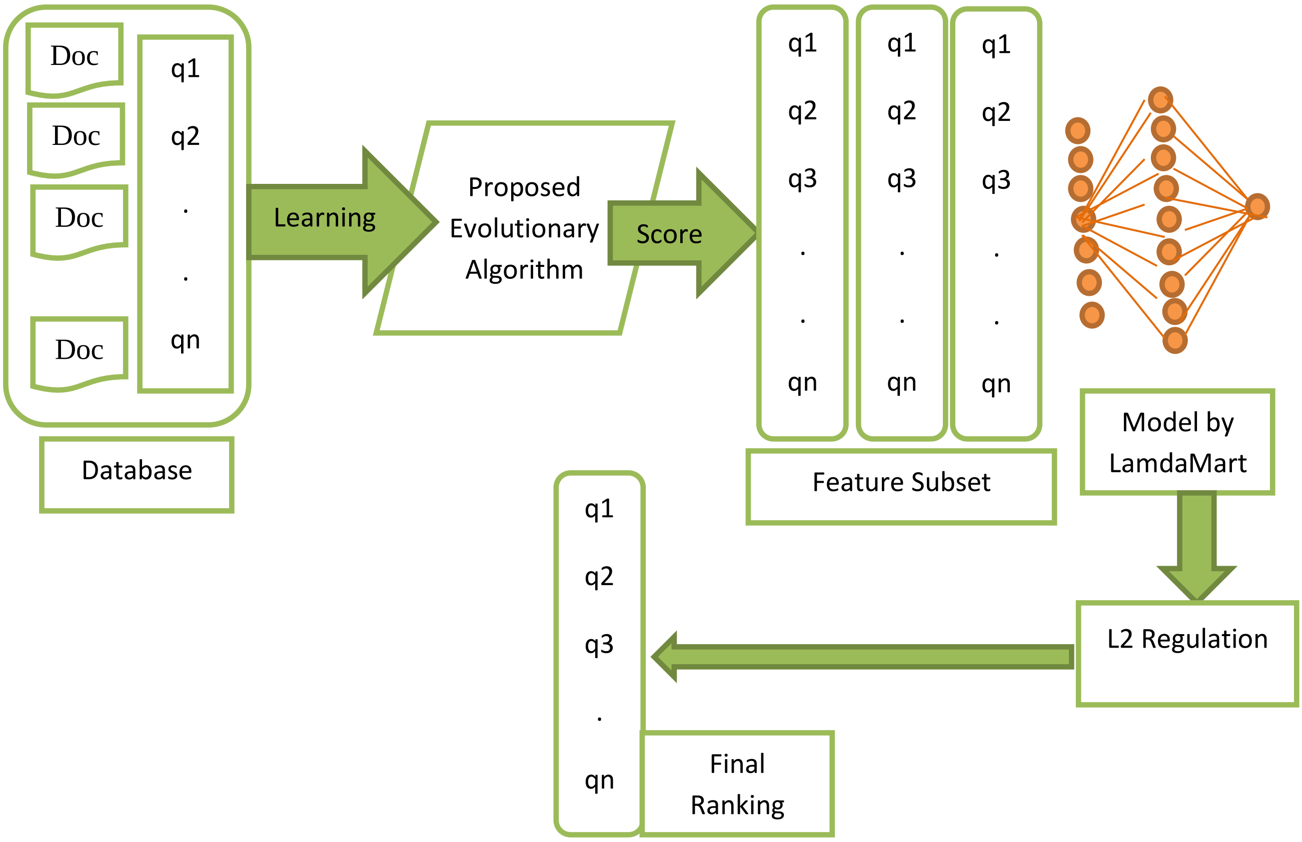


Fig. 4. Proposed architecture with the evolutionary algorithm.

Traditional learning with an evolutionary algorithm multiobjective function or convex optimization is the major focus, as depicted in the literature survey. The main objective of this paper is to use the hybrid evolutionary model for feature selection only. The features of the selected datasets are used as inputs to this model, and the output is the feature subset, which can be trained on the LamdaMart algorithm for ranking. A detailed architecture diagram is shown above in fig6.We evaluate the proposed model with all the conventional metrics, such as precision@k, mean average precision (MAP), and normalized discounted cumulative gain (NDCG).

The evolutionary algorithm used was simulated annealing with a KNN classifier. The modified simulated annealing algorithm can be used to optimize the weights of the ranking function. Initially, by using the elbow method, the optimal k value was found for the classification. We applied standard simulated annealing with some modifications. Given a set of weights w, a query q, and a set of documents D, the relevance scores can be computed using the following method.

Figure 4 Depreciated the flow of process carried out during experimentation. Databases are from above mention datasets. Dataset is given to Proposed Evolutionary Algorithm which mostly used simulated annealing which results into subsets of features. Which then trained on

Activation function of LamdaMart with L2 regularization to avoid the over fitting and generate the sorted list of documents.

4.2. databse

As per Proposed Model Standard datasets are used details of datasets are as follows:

Table 6: Data set Description

Dataset	Benchmark	Queries	Documents	Features
MQ2007	LETOR	1692	70k	46
MQ2008	LETORv2	782	15k	46
MSLR-WEB-10k	Microsoft	31531	3775k	136
MSLR-WEB-30k	Microsoft	10000	1200k	136

We are going to evaluate the proposed model with all the conventional measures such as Precision @k, Mean Average Precision (MAP), Normalized Discounted Cumulative Gain (NDCG).

Evolutionary Algorithm used is simulated annealing with some KNN classifier. The modified Simulated Annealing algorithm can be used to optimize the weights of the ranking function.
Initially by using elbow method find the optimal k value for the classification. We apply standard Simulated Annealing with some modification. Given a set of weights w, a query q, and a set of documents D, the relevance scores can be computed using following method.

Algorithm 1: Process of proposed Evolutionary Model

- Define the search space: Identify the weights that need to be tuned to optimize the ranking function, Create an initial random subset (Let s = s0) with iteration (K).
- Define probability Function of random subset and $S_{newsubset}$
- **while** *kmax != 0* **do**
  - P(accept) ← min (1, $\left|\frac{e(-(obj_{new} - obj_{old})}{T}\right|$)
  - Perturbed the current subset and fit the model;
  - $r(q,d) = \sum(w_i \times f_i(q,d)$
  - Find all the neighbour : Define neighbour function to find random neighbour.
  - $s_{new} = call\ neighbour\ function$;
  - Call probability Function with random subset and $s_{newsubset}$;
    - if performance is better than Perturbed set then
      - Accept new subset
    - else
      - Calculate the acceptance probability;
        - if($P(E(s), E(snew), T)$ random (0,1) then
          - s=s_{new}
        - else

- accept new subset
- end
- end
- Update the K value by 1;
- end

Neighbour function Caution is done as follows:

Let S represent the current feature subset, and let S′ represent a neighbouring solution generated by applying a local change to S.

- Let S′=S∪ {fi} represent the neighbour obtained by adding feature fi to the current subset SS. This operation can be represented as:

  S′=S+fi (1)

- Let S′=S\{fi} represent the neighbor obtained by removing feature fi from the current subset S. This operation can be represented as:

  S′=S−fi (2)

- Let S′ represent the neighbor obtained by swapping two features fi and fj in the current subset S. This operation can be represented as:
  S′=S−fi+fj (3)

Dimention reduction is done via t-Distributed Stochastic Neighbor Embedding by using

X′=t-SNE(X) (4)

where t-SNE is a non-linear dimensionality reduction technique that maps the high-dimensional feature space X to a lower-dimensional space X′X′ while preserving the local structure of the data.

5.Result

The evolutionary result of the modified simulated annealing algorithm with classification for ranking using learning to rank depends on various factors, such as the quality of the training data, the choice of cost function, the selection of the mutation and acceptance functions, and the length of the optimization process. All the data are well organized and provided in different folds; thus, training, validation and testing are easy. In above mentioned algorithm

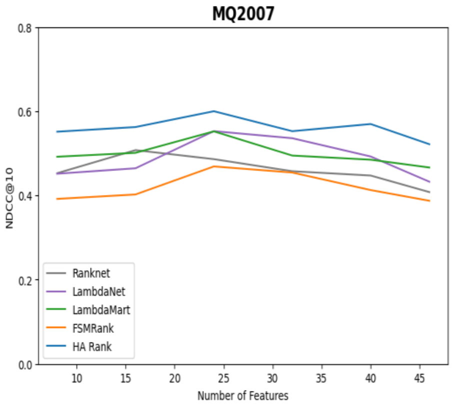
$$r(q,d) = \sum w\_b * f\_b(q,d) \quad (5)$$

where r(q, d) is the relevance score of document d for
query q, w_b is the weight of feature i, and f_b(q, d) is the value of feature i for query q and document d.
where obj_new is the objective function value of the proposed solution, obj_old is the objective function value of the current solution, and T is the current final subset. This equation ensures that the probability of accepting a worse solution decreases as the temperature decreases, allowing the algorithm to converge to a good solution.

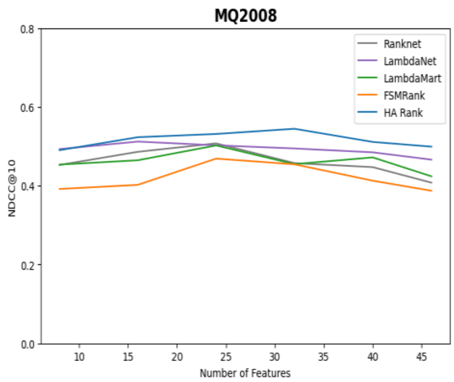
Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 31 No. 2s (2024)

The model's performance was evaluated by comparing it to that of standard benchmark algorithms for learning to rank. Specifically, the RankNet, LambdaNet, and LambdaMart algorithms were chosen because they have demonstrated strong performance in various applications. Additionally, feature selection was evaluated using FSM rank and heuristic rank (HA).

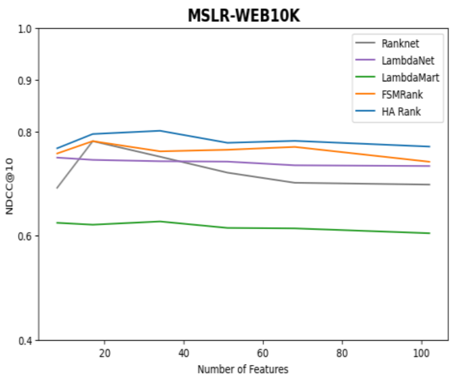
Above Result shows only NDCG@10 Evaluation of MSLR WEB 10K data-sets only. Before Applying to all data-set , the model performance was checked. Above Results shows that subset selection method perform well as compare to existing algorithms mentioned in above also details available in the graph as well.Our proposed algorithm named as HA algorithm.



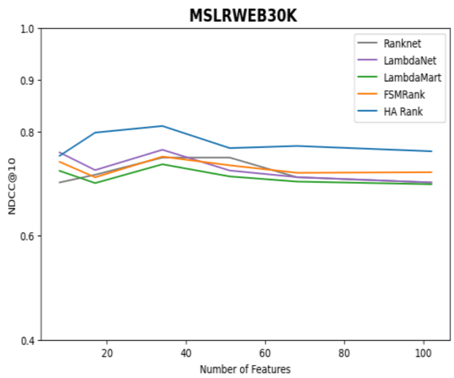
(a)



(b)



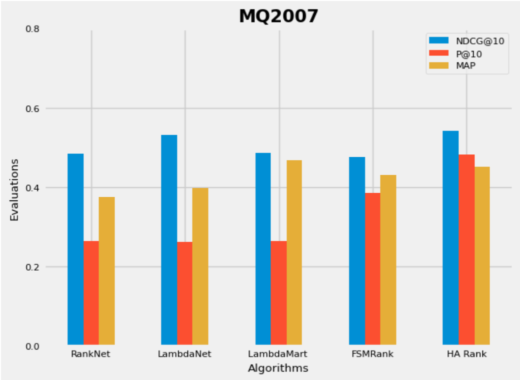
(c)



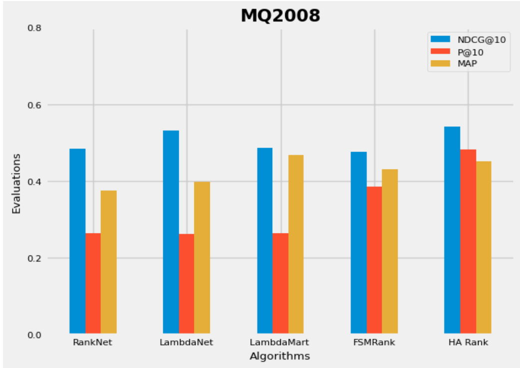
(d)

Fig5: NDCG@10 score comparison on different dataset.

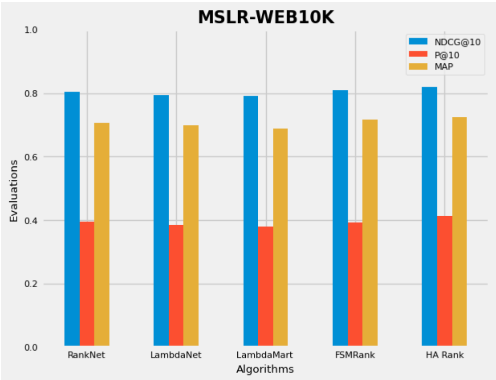
We compare the prediction of our proposed frameworks against those of the other state-of-the-art algorithm RankNet[6], LambdaNet, LambdaMart[3], FSMRank[10]. Figure 5 represent NDCG@10 score of each algorithm with proposed method. We can notice that proposed algorithm perform well in all selected database. Fig 5(a) represents the comparison for MQ2007. MQ2008, MSLR10k and MSLR30k comparison provided in figure number 5(b),5(c),5(d) respectively.



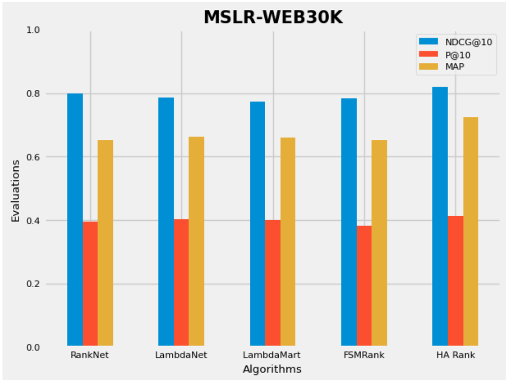
(a)



(b)



(c)



(d)

Fig6: Performance Evaluation with all Matrix

After comparison of with NDCG@10, all algorithms are also comparing with other evaluation parameters such as p@10, MAP. Based on improved performance above model is applied on all detests mention with all fold and also Evacuated on all the parameters.

5.1 Map Analysis:

In Figure6 MAP values of evaluation are better perform where two relevance score are mentioned. By using our architecture we match the relevance appropriately and Find the evaluation of proposed model. In MQ2007 and MQ2008 the Map values are less and MSLR WEB 10k and MSLR WEB 10k the values are more. Amongst all the baseline algorithms shown in diagram indicates that the performance of ranking is improved. Dimension Reduction and Regulation improve the results. Approximately (8%) MAP improvement are visible in the results also show.

5.2 NDCG@n Analysis:

In Figure 6 evaluation shown on NDCG@10 for only MSLR WEB 10K data. In Figure 6 The analysis shown in all the mentioned data where many up-gradation and degradation were visible. In Non Evolutionary category the NDCG@10 values are less in other case it is increased.

In Evolutionary category the performance were increased in all the database. Various of performance is clearly visible in MQ2007 and MQ2008 but for MSLR databases in it slightly improved. Approximately (3 to 4%) improvement are visible in the results. But in case LETOR 3 and 4 the improvement is around (10%).

5.3.P@10 Analysis

In Figure 6 The analysis shown in all the mentioned data where many changes in evaluation measures was found were visible. In Non Evolutionary category the P@10 values are less in other case it is increased.

In Evolutionary category the performance were increased in all the database. Various of performance is clearly visible in MQ2007 and MQ2008 but for MSLR databases in it slightly improved. Approximately (2 to 3%) improvement are visible in the results. But in case LETOR 3 and 4 the improvement is around (15%).

6.Discussion

In the above sections, we have discussed the proposed model and its results on benchmark datasets. The performance feature selection using the heuristic approach method outperforms the other methods, which were compared in the results. The proposed model utilized optimization as a tool for feature selection. In the Complete System, there is more concern about the relevance of the document within the subset so that we can obtain an acceptable subset for learning to rank. By finding the optimized k value, the evolutionary process is completed fewer times with an improvement in the ranking. Figure 7 shows the results of the NDCG measurements only. After improvement,the remaining rankings were calculated with RankNet, Lamdanet, and Lamdamart via one evolutionary algorithm, FSM Rannk. The first three algorithms are non evolutionary algorithms, and the last one is an evolutionary algorithm. Figure 8 shows the comparisons of all the measures. As the first subset is found, LTR is used in most of the cases in which improved results are shown. No changes in the MSLR web10k algorithm were detected for p@k or MAP. The remaining cases were improved on each dataset.

According to the below diagram, the performance of the proposed model increases in the matrix of NDCG and MAP. However, the P@10 values vary with the subset value. By changing the probability function, it will increase. The increasing results are shown in the bellow diagram.

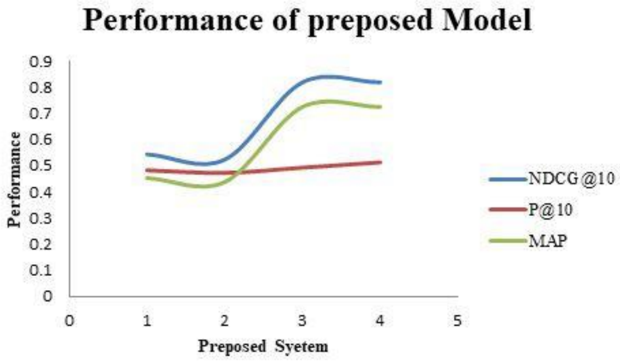


Fig. 9. Performance of the proposed system

7.Conclusion

In this paper, we applied an evolutionary method that involved hill climbing and random walk learning with KNN classification. We propose a hybrid model that selects a feature, optimizes it throughout the process, and outperforms existing frameworks. Simulated annealing using the above mentioned techniques was employed for feature selection, and L2 regularization was applied for optimization. We applied this process to four standard datasets. The selection of active features is based on the activation function of LamdaMart in each iteration until it reaches the convergence state. We compare proposed system with Ranknet,LamdaNet,LamdaMart algorithms which used optimization techniques for proper subset creation and also with FSMRank which is heuristics method for learning to rank. All the case The evaluation matrices show the improved results. Our findings concluded that Optimization Techniques can also be used for feature selection. In the future, we can reduce the dimensionality of features to increase the time complexity of ranking.

References

[1] Rahangdale, Ashwini, and Shital Raut. ”Deep neural network regular- ization for feature selection in learning-to-rank.” IEEE Access 7 (2019): 53988-54006.

[2] Mohd Wazih Ahmad, M.N. Doja, Tanvir Ahmad, Enumerative feature sub- set based ranking system for learning to rank in presence of implicit user feedback, Journal of King Saud University - Computer and Information Sciences, Volume 32, Issue 8,2020, Pages 965-976.

[3] Chavhan, Sushil , Raghuwanshi, M , Dharmik, R. (2021). Information Retrieval using Machine learning for Ranking: A Review. Journal of Physics: Conference Series. 1913. 01215, 2021.

[4] Xiubo Geng, Tie-Yan Liu, Tao Qin, and Hang Li. 2007. ”Feature selection for ranking”. In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR ’07).

[5] T. Joachims, “Training linear SVMs in linear time,” in Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2006, pp. 217–226.

[6] C. J. C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Halmil- ton, and G. Hullender, “Learning to rank using gradient descent,” in Proc. Int. Conf. Mach. Learn., 2005, pp. 89–96.

[7] F. Pan, T. Converse, D. Ahn, F. Salvetti, and G. Donato, “Greedy and randomized feature selection for web search ranking,” in Proc. 11th Int. Conf. Comput. Inf. Technol., 2011, pp. 436–442.

[8] Lai, H. J., Pan, Y., Tang, Y., Yu, R. (2013). FSMRank: Feature selection algorithm for learning to rank. IEEE Transactions on Neural Networks and Learning Systems, 24(6), 940–952. doi:10.1109/TNNLS.2013.2247628 PMID:24808475

[9] Naini, K. D., Altingovde, I. S. (2014). Exploiting Result Diversification Methods for Feature Selection in Learning to Rank. In MECIR 2014, LNCS (Vol. 8416, pp. 455–461)

[10] Gupta, P., Rosso. (2012). Expected Divergence based Feature Selection for Learning to Rank. In Proceedings of the 24th International Conference on Computational Linguistics, COLING-2012 (pp. 10-14).

[11] Shirzad, M. B., Keyvanpour, M. (2015). A feature selection method based on minimum redundancy maximum relevance for learning-to-rank. Qazvin: 5th Conference on Artificial Intelligence and Robotics, IRANOPEN

[12] Gigli, A., Lucchese, C., Nardini, F. M., Perego, R. (2016) Fast Fea- ture Selection for Learning-to-rank. In Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval (ICTIR '16)

[13] Hua, G., Zhang, M., Liu, Y., Ma, S., Ru, L. (2010). Hierarchical fea- ture selection for ranking. In Proceedings of 19th International Conference World Wide Web (pp. 1113–1114).

[14] Sun, Z., Qin, T., Tao, Q., Wang, J. (2009). Robust sparse rank learning for non-smooth ranking measures. In Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 259–266). doi:10.1145/1571941.1571987

[15] Lai, H., Pan, Y., Liu, C., Lin, L., Wu, J. (2012). Sparse learning to rank via an efficient primal-dual algorithm. IEEE Transactions on Computers, 99(6), 1221–1233.

[16] Lai, H., Pan, Y., Tang, Y., Liu, N. (2013). Efficient gradient descent algorithm for sparse models with application in learning-to-rank. Knowledge-Based Systems, 49, 190–198.

[17] Laporte, L., Flamary, R., Canu, S., D´ejean, S., Mothe, J. (2014). Non- convex Regularizations for Feature Selection in Ranking with Sparse SVM. IEEE Transactions on Neural Networks and Learning Systems, 25(6), 1118–1130. doi:10.1109/TNNLS.2013.2286696

[18] Krasotkina, O., Mottl, V. (2015). A Bayesian Approach to Sparse Learn- ing to rank for Search Engine Optimization. In Proceedings of the 11th International Machine Learning and Data Mining in Pattern Recognition Conference MLDM '15, LNCS (Vol. 9166, pp. 382-394).

[19] Changsheng Li, Ling Shao, Changsheng Xu, and Hanqing Lu. 2011. Feature selection under learning to rank model for multimedia retrieve. In Proceedings of the Second International Conference on Internet Multimedia Computing and Service (ICIMCS '10). Association for Computing Machinery, New York, NY, USA, 69–72. https://doi.org/10.1145/1937728.1937745.

[20] Gupta, Parth, and Paolo Rosso. "Expected divergence based feature selection for learning to rank." In Proceedings of COLING 2012: Posters, pp. 431-440. 2012.

[21] Lai, Han-Jiang, Yan Pan, Yong Tang, and Rong Yu. "FSMRank: Feature selection algorithm for learning to rank." IEEE transactions on neural networks and learning systems 24, no. 6 (2013): 940-952.

[22] Gigli, Andrea, Claudio Lucchese, Franco Maria Nardini, and Raffaele Perego. "Fast feature selection for learning to rank." In Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval, pp. 167-170. 2016.

[23] Sousa, Daniel Xavier De, Sérgio Daniel Canuto, Thierson Couto Rosa, Wellington Santos Martins, and Marcos André Gonçalves. "Incorporating risk-sensitiveness into feature selection for learning to rank." In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, pp. 257-266. 2016.

[24] Shirzad, Mehrnoush Barani, and Mohammad Reza Keyvanpour. "A systematic study of feature selection methods for learning to rank algorithms." International Journal of Information Retrieval Research (IJIRR) 8, no. 3 (2018): 46-67.

[25] Cheng, Fan, Wei Guo, and Xingyi Zhang. "Mofsrank: A multiobjective evolutionary algorithm for feature selection in learning to rank." Complexity 2018 (2018): 1-14.

[26] Purpura, Alberto, Karolina Buchner, Gianmaria Silvello, and Gian Antonio Susto. "Neural feature selection for learning to rank." In Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part II 43, pp. 342-349. Springer International Publishing, 2021.

[27] Fahandar, Mohsen & Hüllermeier, Eyke."Analogical Embedding for Analogy-Based Learning to Rank". 10.1007/978-3-030-74251-5_7., 2021.