A Collaborative Filtering Approach for Movies Recommendation Based on User Clustering and Item Clustering

Shristi[1(✉)], Alok Kumar Jagadev[1], and Sachi Nandan Mohanty[2]

[1] KIIT Deemed to be University, Bhubaneswar, India
shristi.shristil9@gmail.com,
alok.jagadevfcs@kiit.ac.in
[2] Gandhi Institute for Technology, Gramadiha, India
dr.sachinandan@gift.edu.in

Abstract. Recommender systems (RS) are software tools that have become increasingly popular in recent years. RS are utilized in a variety of areas including movies, music, news, books, research articles, etc. Typically, there are many items and many users present in these areas making the problem hard and expensive to solve. Collaborative filtering is a widely used approach to design of recommender systems. This method is based on collecting and analyzing a large amount of information on users' behaviors, activities or preferences and predicting what users will like based on their similarity to other users. A key advantage of the collaborative filtering approach is that it does not rely on machine analyzable content and therefore it is capable of accurately recommending complex items like movies without requiring an understanding of the item itself. We present a new approach based on user clustering and item clustering to recommendation for the active user. The K-means clustering algorithm is used to categorize users based on their interests. Our result shows that the proposed algorithm provides improved quality of clusters and also render a better recommendation to the users.

Keywords: Recommender system · Collaborative filtering · K-means Clustering

1 Introduction

Recommender system uses the opinion of a group of users to help individuals in context to identify more effectively the contents of interest from a possibly overwhelming choices set [1]. It has changed the way inanimate websites communicate with their users. The goal of this paper is to provide affordable, personal and high quality recommendations according to users preferences on an item. Types:

1. Content-based Recommendation
2. Collaborative Recommendation
3. Knowledge-based Recommendation
4. Hybrid Recommendation

M. Singh et al. (Eds.): ICACDS 2018, CCIS 906, pp. 187–196, 2018.
https://doi.org/10.1007/978-981-13-1813-9_19

Recommender system that is content-based, recommends items to the users on the basis of correlation in-between the user preferences and the content of items [2]. In this type of system, the user gets recommendation about items that matches the items the user favored in the past [3]. Text documents are widely used as the information source. For making recommendations, content-based structure mostly works by calculating how strongly an item that is not yet seen similar to the active user preferred items in the past.

Collaborative filtering systems are based on gathering and studying a huge volume of info on user's behavior, preferences or activities and then predicting what the user will prefer on the basis of their similarity to another user. Collaborative filtering methods are further categorized as model-based and memory-based collaborative filtering. An eminent case of a memory-based methods is user-based algorithm [4] and that of a model-based methods is kernel-mapping recommender [5].

Regarding the recommendation on the basis of knowledge of user specific tasks can address problem by a knowledge based model. A recommender that is based on knowledge, recommends items on the basis of suggestions about the user's choices and requirements. These knowledges at times hold explicit useful information about how features of a particular product meet users need [6, 7]. A more hybrid method, merging content-based filtering and collaborative filtering. Hybrid methodologies can also be applied in many ways: by making collaborative-based and content-based forecasts individually and then binding them; by adding collaborative-based approach to a content-based capabilities (and vice versa); or by combing the approaches into a single model [8] for an entire review of recommender system. A number of studies empirically match the performance of the content-based and pure collaborative methods with the hybrid and prove that the hybrid approach can offer more precise recommendations than the pure approach. These hybrid methods can be used to eradicate some of the most common issues in recommender systems such as the sparsity problem and cold start.

1.1 Collaborative Filtering

Similarities between users and items and manipulation of relationships between them is computed in the collaborative filtering system. The system manages the interaction of a user with the preferred items. Then the items are recommended in which the targeted user is likely to be interested. Collaborative filtering applies the ratings of the user's community. For this the system forecasts the aimed user's ratings for the items which are not rated till now, so the system has no straight knowledge to define if the user dislikes or likes them [9]. Then the items are ordered according to ranks and the items with top anticipated ratings as per the predicted ranking are recommended to the user.

1.2 Item and User Clustering

User clustering is done on the basis of classifying set of users who seem to have alike ratings. After that the cluster is made and then it becomes easier to make guesses for an aimed user by simply aligning other user's opinion in that cluster.

Item clustering is done on the basis of classifying set of items which seem to have alike ratings. After that the cluster is made and then it becomes easier to make guesses for an aimed user by simply aligning other user's opinion in that cluster.

2 Literature Survey

With the rising concepts in Recommender system approaches and methods, a few existing work discussed as follows:

Phongsavanh Phorasium et al. [10] discusses in this a recommendation method on the basis of user clustering where Euclidean distance is used to calculate two number of users to the clustered dataset. In this paper, recent strategies has been surveyed by grouping them into personalization and hybridization.

Elahi et al. presented the complete outline of interpretation methods that has been applied for testing active learning methods of collaborative filtering [11]. K-means clustering is a method of cluster analysis in which the initial k-centroids are picked up randomly and every item is allocated to the cluster which have the closest centroid. Genetic algorithm works on the candidate solution population; every solution's proximity for best solution of the issue is indicated by its fitness value.

Bhao et al. [12] suggested a new approach where k-means clustering was mapped with genetic algorithm for improving the value of clusters and provided better recommendation to the user. Complexity metrics helps to accelerate error identification and reconstruct task on the most complex parts of a knowledge base.

Felfernig et al. [13] presented knowledge sources to simplify the relationships between different recommendation techniques and outlined open research issues.

Herlocker et al. [14] showed the explanation to the automated collaborative systems has been addressed on the basis of user's conceptual model and experimental evidences has been provided that showed the improvement in the acceptance of automated collaborative systems.

Despite collaborative filtering being mostly used algorithm, it undergoes high running time. In this paper k-means algorithm along with collaborative filtering method enables the users to save time by providing choices to users in less time. Also this paper emphasizes on user and item clustering methods which provides better results than the existing methods on movies recommendations.

3 Problem Definition

We need to examine the application of k-means clustering and collaborative filtering techniques for better recommendations and develop a hybrid algorithm and compare it with the existing algorithm for getting better quality of clusters and less time consumption.

4 Methodology

The proposed framework is shown in Fig. 1. The system shows the work after active user gives a rating to the movie when you watched and requests a recommendation of other movies [13]. User clustering is performed with the k-means algorithm and find out user similarity with Pearson correlation in order to compare profile of active user to others which are under same cluster with active user. Then select closest N neighborhoods that bring profile to all who predict rating movies of current user that has not visited before and then ranked and rated the movies to the current users in order to recommend movies by applying collaborative filtering.

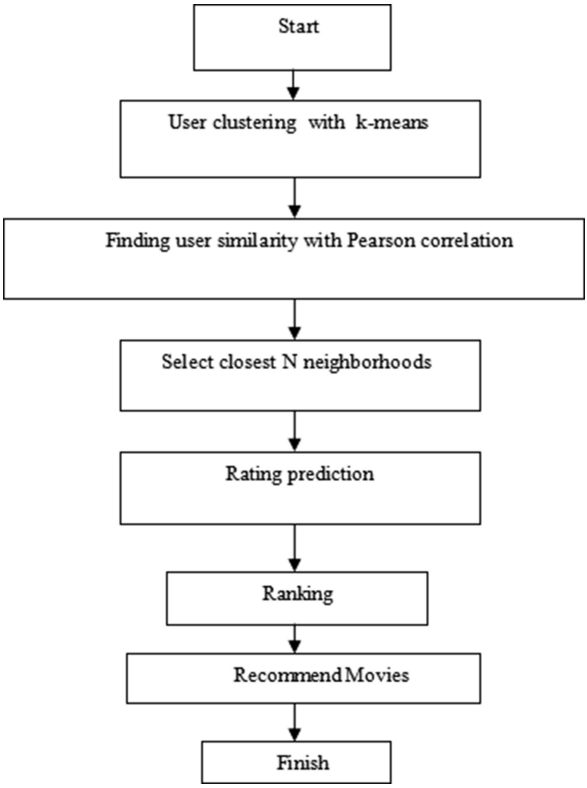

Fig. 1. Proposed model

4.1 Dataset Description

The above model displays the work after active 943 users gives 100000 rating to 1682 movies, when you watched and requests a recommendation system of other movies. The data has been considered from website movie lens project http://grouplens.org/datasets/. Then, by with the help of k means the active user will be clustered by the

system to the recommended group, then the active user will be compared by the system for getting the cluster which have same properties as the active user from the different users who are in the same cluster using Pearson correlation coefficient as the comparison tool. Collaborative filtering will compute the rating between users within the cluster, after that it will compare the similarity values in the user-user similarity matrix. It will select closest user similarity and pull data similarity values for neighbors that come out and compute the predicted rating and put in store to search highest predicted rating for relevant user and recommendation to the user.

4.2 Processing K-means and Collaborative Filtering

The system will search group for users by using k-means to find distance between the users, the group of users and clustering of users. The system performs clustering with the k-means algorithm by calculating the distance of each data point from the center of the 19 groups by using Euclidean distance and calculated information will be collected in the database as shown in Table 1.

Table 1. Initial non-rating values

1,20,	4,50,	7,32,	10,7
1,33,	4,260,	7,163,	10,16
1,61,	4,264,	7,382,	10,100
1,117,	4,288,	7,430,	10,175
1,155,	4,294,	7,455,	10,285
1,160,	4,303,	7,479,	10,461
1,171,	4,354,	7,492,	10,486
1,189,	4,356,	7,497,	10,488
1,202,	4,357,	7,648,	10,504
1,265,	4,361,	7,661,	10,611
2,13,	5,1,8,	22,	11,38
2,50,	5,2,8,	50,	11,110
2,251,	5,17,8,	79,	11,111
2,280,	5,98,8,	89,	11,227
2,281,	5,110,	8,182,	11,425
2,290,	5,225,	8,294,	11,558
2,292,	5,363,	8,338,	11,723
2,297,	5,424,	8,385,	11,725
2,312,	5,439,	8,457,	11,732
2,314,	5,454,	8,550,	11,740
3,245,	6,14,	9,6,	12,82
3,294,	6,23,	9,286,	12,96
3,323,	6,69,	9,298,	12,97
3,328,	6,86,	9,340,	12,132
3,331,	6,98,	9,479,	12,143
3,332,	6,258,	9,487,	12,172
3,334,	6,301,	9,507,	12,204
3,335,	6,463,	9,521,	12,300
3,337,	6,492,	9,527,	12,471
3,343,	6,517,	9,691,	12,735

After that, the system will search for a user similarity established on the definition and will create a matrix of data between users on the movies as shown in Table 2.

Table 2. Values from Pearson correlation

[1:2] = -0.027046	[1:3] = 0.366547	[1:4] = 0.375964	[1:5] = 0.030415
[1:8] = 0.025755	[1:9] = -0.061102	[1:10] = -0.209068	[1:11] = 0.225009
[1:14] = -0.210460	[1:15] = 0.154863	[1:16] = 0.169840	[1:17] = 0.102967
[1:20] = 0.059862	[1:21] = 0.340243	[1:22] = 0.525411	[1:23] = -0.025513
[1:26] = 0.063825	[1:27] = 0.077704	[1:28] = -0.249809	[1:29] = -0.136661
[1:32] = 0.026309	[1:33] = 0.005773	[1:34] = 0.060832	[1:35] = -0.095517
[1:38] = -0.066236	[1:39] = -0.196438	[1:40] = 0.066372	[1:41] = -0.160233
[1:44] = -0.220306	[1:45] = 0.544769	[1:46] = 0.095071	[1:47] = 0.204483
[1:50] = -0.208768	[1:51] = -0.004138	[1:52] = -0.183185	[1:53] = -0.115057
[1:56] = 0.221590	[1:57] = 0.365535	[1:58] = 0.226752	[1:59] = 0.386534
[1:62] = 0.178297	[1:63] = 0.434215	[1:64] = -0.150062	[1:65] = 0.448463
[1:68] = 0.111250	[1:69] = -0.089132	[1:70] = 0.038691	[1:71] = 0.663358
[1:74] = 0.118426	[1:75] = 0.133271	[1:76] = 0.255047	[1:77] = -0.144694
[1:80] = 0.266248	[1:81] = 0.304979	[1:82] = -0.155222	[1:83] = 0.162837
[1:86] = 0.339964	[1:87] = 0.144032	[1:88] = 0.001942	[1:89] = 0.044825
[1:92] = 0.431812	[1:93] = 0.264581	[1:94] = 0.526021	[1:95] = -0.093788
[1:98] = 0.196812	[1:99] = 0.025080	[1:100] = 0.039290	[1:101] = 0.002772
[1:104] = -0.096853	[1:105] = -0.011006	[1:106] = -0.025008	[1:107] = 0.043351
[1:110] = 0.420462	[1:111] = 0.011002	[1:112] = 0.060912	[1:113] = 0.120490
[1:116] = 0.007559	[1:117] = 0.168349	[1:118] = 0.118157	[1:119] = 0.088839
[1:122] = 0.012231	[1:123] = -0.221455	[1:124] = 0.282288	[1:125] = -0.593173
[1:128] = -0.144119	[1:129] = -0.193517	[1:130] = 0.126576	[1:131] = -0.363107
[1:134] = -0.051010	[1:135] = 0.157475	[1:136] = -0.086090	[1:137] = -0.119110
[1:140] = 0.121372	[1:141] = 0.100078	[1:142] = 0.113705	[1:143] = -0.085680
[1:146] = -0.133287	[1:147] = -0.475249	[1:148] = 0.003732	[1:149] = 0.000306
[1:152] = -0.141169	[1:153] = -0.039955	[1:154] = -0.210857	[1:155] = -0.120756
[1:158] = -0.054972	[1:159] = -0.024189	[1:160] = 0.283674	[1:161] = 0.532374
[1:164] = -0.058478	[1:165] = 0.226578	[1:166] = -0.120245	[1:167] = -0.137346
[1:170] = -0.255002	[1:171] = 0.295665	[1:172] = -0.316508	[1:173] = -0.100848
[1:176] = 0.261707	[1:177] = -0.219091	[1:178] = -0.057724	[1:179] = -0.019160
[1:182] = -0.040158	[1:183] = -0.001536	[1:184] = -0.068143	[1:185] = -0.365917
[1:188] = 0.195445	[1:189] = 0.050084	[1:190] = 0.196985	[1:191] = -0.028975
[1:194] = 0.057709	[1:195] = 0.114948	[1:196] = 0.330779	[1:197] = -0.145631
[1:200] = -0.273434	[1:201] = 0.262768	[1:202] = -0.118991	[1:203] = -0.101803
[1:206] = -0.029630	[1:207] = 0.079559	[1:208] = -0.145406	[1:209] = 0.340360
[1:212] = -0.195339	[1:213] = 0.059014	[1:214] = -0.089849	[1:215] = -0.232644

The system will be examined by users who are similar, and compare user 1 to all the others who have pieces of information rating include the user with other user, so at this stage to make a correlation between the User 1 and remaining other users respectively. Pearson correlation displays the similarity to the closeness of making comparisons. After that, the process of leading the user that looks for rating similar to the target the number of k to predict satisfaction as possible by weight sum equation as shown in Table 3.

4.3 Result Discussion

The goal of clustering is to know how many people in the groups and the centroid of the group are present. Then bring centroid to a cluster group for new user to the group by k-means algorithm. In this we use a WEKA software. It is used to cluster a group of users, the data downloaded from the website movie lens project data. It is so big to choose 943 users, 1682 movies records and 100000 ratings. First, we should convert

Table 3. Guessing values from Pearson correlation

1, 20, 3.6923	1, 33, 3.4737	1, 61, 4.2083
1, 155, 3.0000	1, 160, 4.2083	1, 171, 3.3878
1, 202, 3.8667	1, 265, 3.4737	2, 13, 3.6000
2, 251, 3.6000	2, 280, 4.3000	2, 281, 3.7500
2, 292, 3.7647	2, 297, 3.7647	2, 312, 5.0000
3, 245, 3.0000	3, 294, 2.8000	3, 323, 3.0000
3, 331, 2.0000	3, 332, 3.0000	3, 334, 1.0000
3, 337, 4.0000	3, 343, 2.7500	4, 50, 3.0000
4, 264, 5.0000	4, 288, 4.2170	4, 294, 5.0000
4, 354, 4.3418	4, 356, 5.0000	4, 357, 5.0000
5, 1, 3.2500	5, 2, 2.8571	5, 17, 2.8571
5, 110, 3.4167	5, 225, 3.0926	5, 363, 2.8000
5, 439, 2.2381	5, 454, 2.4615	6, 14, 3.6667
6, 69, 3.7333	6, 86, 3.6552	6, 98, 4.5000
6, 301, 3.3611	6, 463, 3.8889	6, 492, 3.6552
7, 32, 4.2500	7, 163, 3.8113	7, 382, 4.4444
7, 455, 3.1429	7, 479, 4.0714	7, 492, 4.1892
7, 648, 3.4167	7, 661, 4.0769	8, 22, 4.6667
8, 79, 4.0000	8, 89, 4.3750	8, 182, 4.1250
8, 338, 2.2857	8, 385, 3.7500	8, 457, 5.0000
9, 6, 4.0000	9, 286, 4.6667	9, 298, 3.9129
9, 479, 4.0552	9, 487, 4.0000	9, 507, 4.0000
9, 527, 4.0000	9, 691, 4.0000	10, 7, 4.2750
10, 100, 4.3333	10, 175, 4.4000	10, 285, 4.2750
10, 486, 3.9000	10, 488, 4.3750	10, 504, 4.2750
11, 38, 3.2747	11, 110, 2.8750	11, 111, 3.4118
11, 425, 3.2750	11, 558, 3.4286	11, 723, 3.7143
11, 732, 3.4118	11, 740, 3.8125	12, 82, 4.0000
12, 97, 4.3000	12, 132, 3.9890	12, 143, 3.0000
12, 204, 4.5000	12, 300, 4.5000	12, 471, 5.0000
13, 56, 3.6091	13, 98, 2.8667	13, 186, 2.8491
13, 215, 3.6091	13, 272, 3.6091	13, 344, 3.6091
13, 526, 2.6522	13, 836, 3.3171	14, 22, 4.4141
14, 111, 3.0000	14, 174, 4.6667	14, 213, 3.8571
14, 357, 4.1500	14, 474, 4.4141	14, 530, 4.3333
15, 25, 2.3077	15, 127, 3.3600	15, 222, 3.3333

data from excel to CSV because this file is supported by WEKA. We can delete something if irrelevant to our data.

In our case we only use user, movies name and rating shown in the Fig. 4. The data is selected and converted into a comma separated values (CSV) and formatted using the CSV converter and then cluster imported to WEKA program. It is the first open WEKA program, explorer, pre-process, and then open your file and click the cluster, after that you will see so many options and then click on the choose button to choose simple k-means. Next left click on the simple k-means N 2-A weka.core. Euclidean Distance R first-last then it will show function. On these pages we can set the number of clusters depending on how many groups do you want. In our paper, we use 10 clusters to get the results as shown in Tables 4 and 5.

Table 5 shows that group 1 includes 6387 people, group 2 includes 4847 people, group 3 includes 4198 people, group 4 includes 6557 people, group 5 includes 3494 people, group 6 includes 4957 people, group 7 includes 4121 people, group 8 includes 5169 people, group 9 includes 5172 people, group 10 includes 4032 people, group 11 includes 5182, group 12 includes 5714 people, group 13 includes 3340 people, group 14 includes 4102 people, group 15 includes 2521 people, group 16 includes 5404

Table 4. Number of group and member of group

```
Time taken to build model (full training data) : 9.52 seconds

=== Model and evaluation on training set ===

Clustered Instances

 0       6387 (  7%)
 1       4847 (  5%)
 2       4198 (  5%)
 3       3865 (  4%)
 4       6557 (  7%)
 5       3494 (  4%)
 6       4957 (  5%)
 7       4121 (  5%)
 8       5169 (  6%)
 9       5172 (  6%)
10       4032 (  4%)
11       5182 (  6%)
12       5714 (  6%)
13       3340 (  4%)
14       4102 (  5%)
15       2521 (  3%)
16       5404 (  6%)
17       4840 (  5%)
18       6668 (  7%)
```

Table 5. Centroid of group

```
=== Run information ===

Scheme:weka.clusterers.SimpleKMeans -N 19 -A "weka.core.EuclideanDistance -R first-last" -I 500 -S 10
Relation:     database-weka.filters.unsupervised.attribute.Remove-R4
Instances:    90570
Attributes:   3
              userid
              movieid
              rating
Test mode:evaluate on training data

=== Model and evaluation on training set ===


kMeans
======

Number of iterations: 78
Within cluster sum of squared errors: 2438.9141903981917
Missing values globally replaced with mean/mode

Cluster centroids:
                      Cluster#
Attribute    Full Data        0        1        2        3        4        5        6        7        8        9
               (90570)   (6387)   (4847)   (4198)   (3865)   (6557)   (3494)   (4957)   (4121)   (5169)   (5172)
=============================================================================================================================
userid          461.494  425.963  562.6872  822.7504  744.8533  556.0554  261.8074  783.3758   78.5404  318.5136   92.2769
movieid         428.1049 268.4555 283.5112 268.6198 869.8116 275.3576 963.2521 249.3609 314.7023 247.5222 303.8213
rating            3.5238        5        3        3   4.2753        4   1.5461        5   2.7607        3        4



Time taken to build model (full training data) : 9.52 seconds
```

people, group 17 includes 4840 people and the last group 18 includes 6868 people, on the same time we will see Fig. 6, that is centroids of all group for movies. Moreover, by considering this set of different groups, using the centroid method we figured out that while the group is increasing, members are getting enhanced in number too.

5 Conclusion and Future Work

This paper suggests an approach to recommendation system where collaborative filtering is used along with k-means algorithm for improving feature of clusters and thereby providing better recommendation to the users. Collaborative filtering provides predictions to users by identifying similar users with k-means. Although this algorithm is preferable it suffers low accuracy. Therefore in future a lot of work still needs to be done to propose a technique for identifying optimum number of clusters for the k-means algorithm. Also, instead of k-means algorithm other techniques like fuzzy c-means can be used to get more effective clusters.

References

1. Ricci, F., Rokach, L., Shapira, B.: Introduction to recommender systems handbook. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 1–35. Springer, Boston, MA (2011). https://doi.org/10.1007/978-0-387-85820-3_1
2. Facebook, Pandora Lead Rise of Recommendation Engines—TIME, 27 May 2010. TIME.com. Accessed 1 June 2015
3. Pu, P., Chen, L.: A user-centric evaluation framework of recommender systems. In: Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, 30 September, pp. 14–21 (2010)
4. Shardanand, U., Maes, P.: Social information filtering: algorithms for automating "word of mouth". In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 1995, pp. 210–217 (1995)
5. Balabanović, M., Shoham, Y.: Fab: content-based, collaborative recommendation. Commun. ACM CACM Homepage Arch. **40**(3), 66–72 (1997)
6. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, UAI 1998, pp. 43–52 (1998)
7. Kernel-mapping recommender system algorithms. Inf. Sci. pp. 81–104 (2015)
8. Zapata, B.C., et al: Revista Ib, Association Ib America de Sistemas e Technologias de Informa, pp. 35–50 (2014)
9. Felfernig, A., Burke, R.: Constraint-based recommender systems: technologies and research issues. In: Proceedings of the 10th International Conference on Electronic Commerce, ICEC 2008 (2008). Article no. 3
10. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. Arch. **17**(6), 734–749 (2005)
11. Elahia, M., Riccib, F., Rubensc, N.: A Survey of Active Learning in Collaborative Filtering Recommender Systems. Elsevier, London (2016)

12. Bhao, K., Kumar, D., Saroj.: An evolutionary k-means clustering approach to recommender systems. In: International Conference on Advanced Computing, Communication and Networks, pp. 851–855 (2011)
13. Phongsavanh, P., Yu, L.: Movies recommendation system using collaborative filtering and k-means. Int. J. Adv. Comput. Res. **7**(29), 52–59 (2018)
14. Herlocker, J.L., Konstan, J.A., Riedl, J.: Explaining collaborative filtering recommendations. In: Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work, pp. 241–250. ACM (2000)