Project Report -I

On

**Product Recommender System (Netflix Challenge 2007)**

***A project report Submitted in partial fulfillment of the requirements of the Degree of***

## Bachelor of Engineering

### In

**(Electronics Engineering) by**

**DEVARSHEE GAURAV DAS (TU1F1617017)**

**SAURABH MANOJ BANSODE(TU1S1718005)**

**NILESH VISHWANATH BORATE (TU1S1718010)**

**DARSHAN RAMESH SAWADEKAR (TU1S1617021)**

**Under the guidance of**

**Dr.B.G.HOGADE**



**Department of Electronics Engineering**

**TERNA ENGINEERING COLLEGE, NAVI MUMBAI**

**University of Mumbai**

**Acadmic Year 2020-21**

Certificate

This is to certify that the requirements for the synopsis entitled ‘**Product Recommender System (Netflix Challenge 2007)**’ have been successfully completed by the following students:

**Name Enrollment No.**

DEVARSHEE GAURAV DAS TU1F1617017

SAURABH MANOJ BANSODE TU1S1718005

NILESH VISHWANATH BORATE TU1S1718010

DARSHAN RAMESH SAWARDEKAR TU1S1617021

in partial fulfilment of Bachelor of Engineering of Mumbai University in the Department of Electronics Engineering, Terna Engineering College Nerul during the Academic Year 2020 – 2021.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Project Guide          Project Coordinator           External Examiner**

**(Name of Project Guide)               Mrs.R.V.Chimankare**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_             \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Head of Department Principal**

**Dr.S.C.Biday                     Dr. L. K. Ragha**

**DECLARATION**

We declare that this written submission represents our ideas in our own words and where other’s ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not taken when needed.

**DEVARSHEE GAURAV DAS SAURABH MANOJ BANSODE**

NILESH VISHWANATH BORATE DARSHAN RAMESH SAWADEKAR

Date: Place:

**ACKNOWLEDGEMENT**

We would like to express our sincere gratitude towards our guide **DR. B. G. Hogade** for the help, guidance and encouragement, she provided during the BE Dissertation. This work would have not been possible without her valuable time, patience and motivation. We thank her for making our stint thoroughly pleasant and enriching. It was great learning and an honor being her student.

We are deeply indebted to DR. **B. G. Hogade** and the entire team in the Electronics Department. They supported us with scientific guidance, advice and encouragement, they were always helpful and enthusiastic and this inspired us in our work.

We take the privilege to express our sincere thanks to **Dr. L.K.Ragha,** our Principal for providing the encouragement and much support throughout our work.

Last but not the least we would like to thank all the helping hands which directly or indirectly helped us in our project.

DEVARSHEE GAURAV DAS

SAURABH MANOJ BANSODE

NILESH VISHWANATH BORATE

D4ARSHAN RAMESH SAWADEKAR

|  |  |
| --- | --- |
| **TABLE OF CONTENT**  **Title**  i.Abstract | **Pg. No.**  3 |
| 1. Chapter 1 | 4-5 |
| Introduction | 4 |
| 1.1.Background | 5 |
| 1.2.Motivation | 5 |
| 2. Chapter 2 | 6-7 |
| Problem Definition | 6 |
| 2.1. Problem statement  2.2. Scope | 7  7 |
| 3. Chapter 3 | 8-11 |
| Literature Survey | 8 |
| 3.1. Literature Survey | 9 |
| 3.2. Existing System | 11 |
| 4. Chapter 4 | 12-16 |
| Proposed System | 12 |
| 4.1. Overview | 13 |
| 4.2. Functional Modules | 16 |
| 5. Chapter 5 | 17-24 |
| Methodology | 17 |
| 5.1. System Simulation & Implementation | 18 |
| 5.2. System Component Selection  5.3. Implementation  6. Chapter 6  Application  6.1. Application  Future Scope  Summary  Reference | 18  19  25-26  25  26  27  28  29 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# ABSTRACT

Recommender systems is a subclass of information filtering system that seeks to predict the "rating" or "preference" a user would give to an item. The system will recommend movies, web series which may users would like to watch. The recommendation system is classified into two types that are Content Based recommendation system and Collaborative based recommendation system. In Content Based Filtering, it will suggest similar items based on a particular item. This system uses item metadata, such as genre, director, description, actors, etc. for movies, to make these recommendations. The general idea behind these recommender systems is that if a person liked a particular item, he or she would also like an item that is similar to it. In Collaborative Filtering, this system matches persons with similar interests and provides recommendations based on this matching. Collaborative filters do not require item metadata like its content-based counterparts

CHAPTER 1

**INTRODUCTION**

**1.1. BACKGROUND**

Watched Netflix and Prime in my free time – the “new” normal for everyone. Netflix and Prime have such engrossing content that keeps us glued to the screen all the time. There is a section on both of these platforms which displays the recommended content on the basis of the previous content that you have watched. These recommendations seem to be quite relevant to your watch history and the kind of content you would want to engage yourselves with. How this works in the background is by designing certain recommendation systems?

**1.2. MOTIVATION**

Recommendation systems help users find and select items (e.g., books, movies, restaurants) from the huge number available on the web or in other electronic information sources. Given a large set of items and a description of the user's needs, they present to the user a small set of the items that are well suited to the description. Recent work in recommendation systems includes intelligent aides for filtering and choosing web sites, news stories, TV listings, and other information. The users of such systems often have diverse, conflicting needs. Differences in personal preferences, social and educational backgrounds, and private or professional interests are pervasive. As a result, it seems desirable to have personalized intelligent systems that process, filter, and display available information in a manner that suits each individual using them. Our goal is to support conversations that become more efficient for individual users over time.

CHAPTER 2

**PROBLEM DEFINITION**

**2.1 Problem Statement-**

Lack of Data: Recommender systems need a large volume of data to make predictions effectively. Huge corporations such as Google, Apple, and Amazon are able to make better recommendations because they continuously collect data on their customers. Also, there are recommender systems based on purchases only, i.e., no ratings are collected. Even Amazon does not know how much a customer liked a book, for example, if they do not rate it.

Cold start: When a new item is added to the catalogue or a new user joins the service, the system has very little historical information with which to make suggestions.

Changing Data and User Preferences: Recommendation is a very dynamic field. Thousands of items are continuously added to catalogues, with millions of features. User preferences are changing, in addition. Therefore, markets and trends are always shifting.

**2.2 Scope-**

As data quantity and quality increases, current algorithms will have to scale efficiently and probably most of the effort will be made in that direction.

Either the improved quality and quantity of data will be used to train deep models and generalize on the “average” human behaviour, or different approaches will have to emerge that are able to filter out less relevant information.

CHAPTER 3

**LITERATURE SURVEY**

**3.1 Literature Survey –**

A Content-Based Recommender works by the data that we take from the user, either explicitly (rating) or implicitly (clicking on a link). By the data we create a user profile, which is then used to suggest to the user, as the user provides more input or take more actions on the recommendation, the engine becomes more accurate.

User Profile:

In the User Profile, we create vectors that describe the user’s preference. In the creation of a user profile, we use the utility matrix which describes the relationship between user and item. With this information, the best estimate we can make regarding which item user likes, is some aggregation of the profiles of those items.

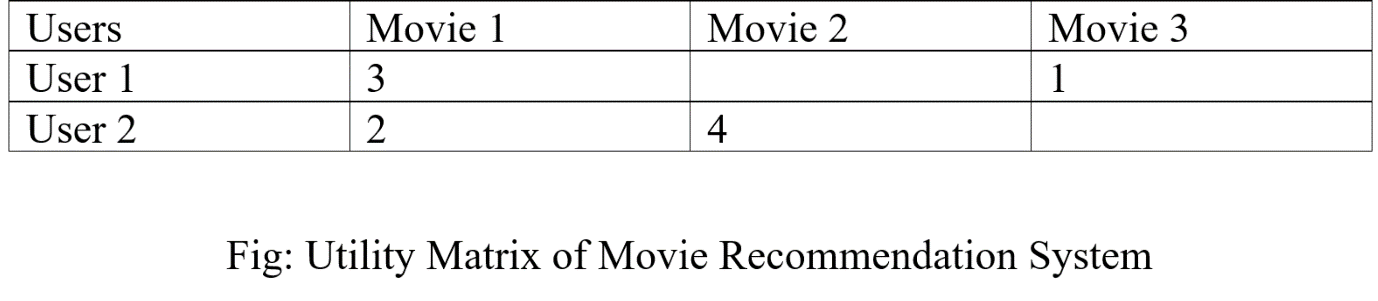
Item Profile:

In Content-Based Recommender, we must build a profile for each item, which will represent the important characteristics of that item.

For example, if we make a movie as an item then its actors, director, release year and genre are the most significant features of the movie. We can also add its rating from the IMDB (Internet Movie Database) in the Item Profile.

Utility Matrix:

Utility Matrix signifies the user’s preference with certain items. In the data gathered from the user, we have to find some relation between the items which are liked by the user and those which are disliked, for this purpose we use the utility matrix. In it we assign a particular value to each user-item pair, this value is known as the degree of preference. Then we draw a matrix of a user with the respective items to identify their preference relationship.



Some of the columns are blank in the matrix that is because we don’t get the whole input from the user every time, and the goal of a recommendation system is not to fill all the columns but to recommend a movie to the user which he/she will prefer. Through this table, our recommender system won’t suggest Movie 3 to User 2, because in Movie 1 they have given approximately the same ratings, and in Movie 3 User 1 has given the low rating, so it is highly possible that User 2 also won’t like it.

Recommending Items to User Based on Content:

Method 1:

We can use the cosine distance between the vectors of the item and the user to determine its preference to the user. For explaining this, let us consider an example:

We observe that the vector for a user will have a positive number for actors that tend to appear in movies the user likes and negative numbers for actor’s user doesn’t like, consider a movie with actors which user likes and only a few actors which user doesn’t like, then the cosine angle between the user’s and movie’s vectors will be a large positive fraction. Thus, the angle will be close to 0, therefore a small cosine distance between the vectors.

It represents that the user tends to like the movie, if the cosine distance is large, then we tend to avoid the item from the recommendation.

Method 2:

We can use a classification approach in the recommendation systems too, like we can use the Decision Tree for finding out whether a user wants to watch a movie or not, like at each level we can apply a certain condition to refine our recommendation.

**3.2 Literature Survey On Existing System-**

Today’s technology includes various online Over–the–top media services (OTT) platforms such as Amazon Prime, Netflix, Hotstar etc. these are some of the popular existing OTT platforms in which we can see the use of movie recommender system. Movie recommender system basically uses these three types of recommender algorithms that are content based, collaborative based and hybrid based. So, what we gone through is content based and collaborative based filtering algorithm in which our first priority will be implementing one of these algorithms, then will move on another algorithm. Among these various OTT platforms, we are implementing this project for Netflix (based on challenge 2007)





CHAPTER 4

**PROPOSED SYSTEM**

**4.1 Overview-**

In today’s digital world where there is an endless variety of movies, web series, documentaries to be consumed, finding the content of one’s liking has become an irksome task. On the other hand, digital content providers want to engage as many users on their service as possible for the maximum time.

This is where movie recommender system comes into picture where the content providers recommend users the content according to the users’ liking. The objective of movie recommender system is to provide accurate movie recommendations to users.

**4.1.1. Advantages of content-based filtering are:**

* They capable of recommending unrated items.
* We can easily explain the working of recommender system by listing the Content features of an item.
* content-based recommender systems use need only the rating of the concerned user, and not any other user of the system.

**4.1.2. Advantages of collaborative filtering based systems:**

* It is dependent on the relation between users which implies that it is content-independent.
* CF recommender systems can suggest serendipitous items by observing similar-minded people’s behavior.
* They can make real quality assessment of items by considering other peoples experience.

**4.1.3. Disadvantages of content-based filtering are:**

* It does not work for a new user who has not rated any item yet as enough ratings are required content-based recommender evaluates the user preferences and provides accurate recommendations.
* No recommendation of serendipitous items.
* Limited Content Analysis- The recommender does not work if the system fails to distinguish the items that a user likes from the items that he does not like.

**4.1.4. Disadvantages of collaborative filtering are:**

* Early rater problem: Collaborative filtering systems cannot provide recommendations for new items since there are no user ratings on which to base a prediction.
* Gray sheep: In order for CF based system to work, group with similar characteristics are needed. Even if such groups exist, it will be very difficult to recommend users who do not consistently agree or disagree to these groups.
* Sparsity problem: In most cases, the amount of items exceeds the number of users by a great margin which makes it difficult to find items that are rated by enough people.
* **4.2 Functional Modules-**

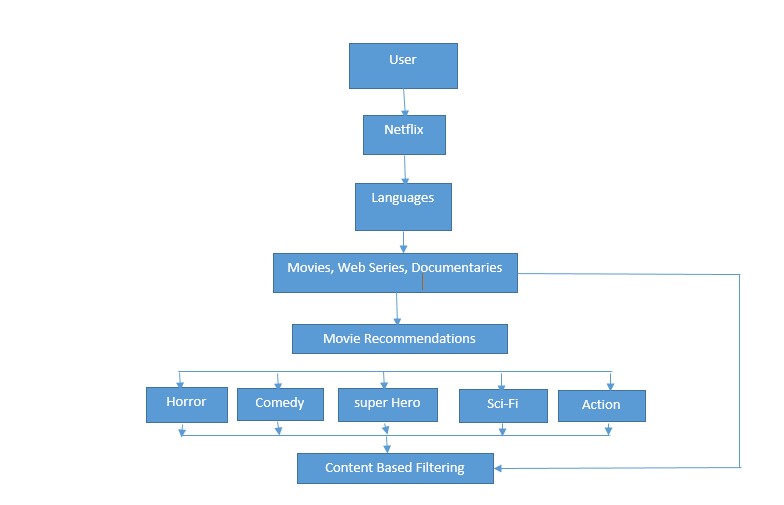
****

Fig. 1.1 Flow Chart

CHAPTER 5

**METHODOLOGY**

**5.1 System Simulation-**

we will present final demo in Pycharm, Jupyter where code will be written in Python language and we will show an output window.

for content based filtering, we will put any movie, web series name as input and so according input, few lists of movies, web series will be recommended as output.

for collaborative filtering, we will put any User ID as input and so according input, few lists of movies, web series will be recommended as output.

**5.2 System Component Selection-**

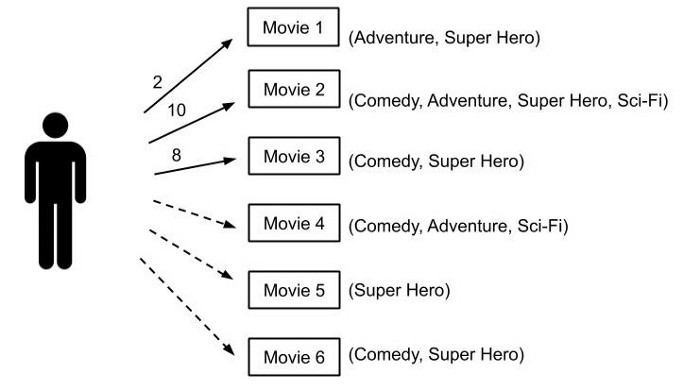
**5.2.1 Software Used-**

**Python-** Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural,

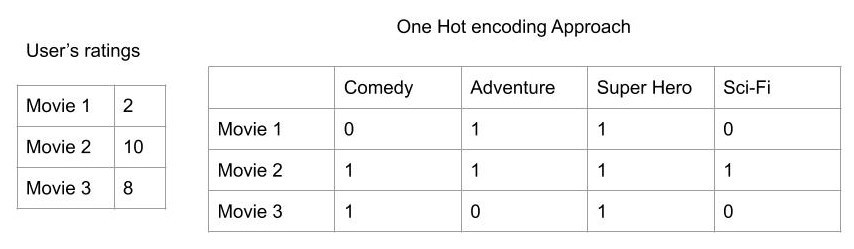
**Jupyter Notebook-** The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modelling, data visualization, machine learning, **5.3 Implementation-**

**5.3.1 Content Based Filtering-**

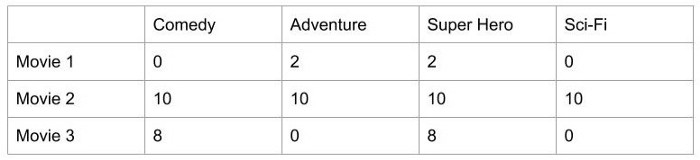
Consider the following example where the user has given the ratings for 3 movies Movie 1, Movie 2, Movie 3 as 2, 10, 8 respectively. Let’s find out what movie from Movie 4, Movie 5, Movie 6 will be recommended to the user.



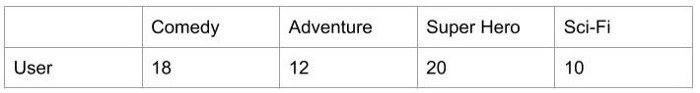
The first step is to make a one-hot encoded matrix based on the genres present in a movie. All the genres of the matrix are assigned a column. If a particular genre is present in the Movie, it is assigned as **1** otherwise **0**



Now, the user rating matrix is multiplied with the one-hot encoded matrix to form **Weighted Genre Matrix.**



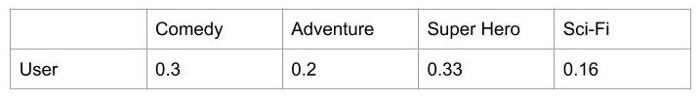
This Weighted Genre Matrix is aggregated to form the user profile which is normalised later to help to make a recommendation matrix.



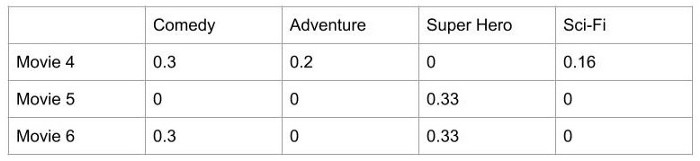
The normalization of the User Profile is done by dividing each element in a row by the sum of the elements in that row.

*Here, 18+12+20+10 = 60*

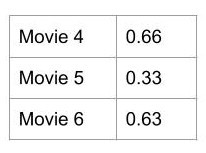
*So each element’s normalized value will become x/60*



Lastly, the Normalized User Profile is multiplied by the one-hot encoded matrix of the remaining available movies which are not rated by that User and then aggregated to give the recommendation matrix.



The Recommendation matrix formed will be used to make recommendations. **The movie with the highest weight will be recommended to the User.**

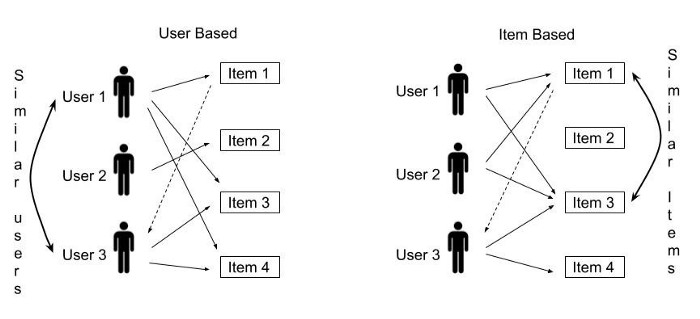


**5.3.1 Collaborative Based Filtering-**

Consider it as the user is saying tell me what’s popular among my neighbours. Finds similar group of users, and provide recommendations based on similar tastes within that group.

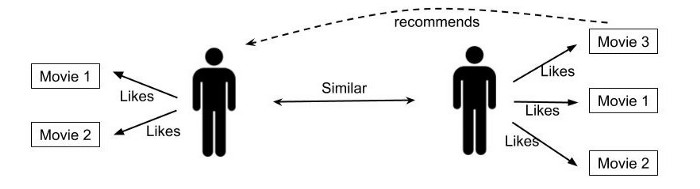
There are two different methods to collaborative filtering —

1. User-Based Collaborative filtering — based on user’s neighbourhood.
2. Item-Based Collaborative filtering — based on the item’s similarity.

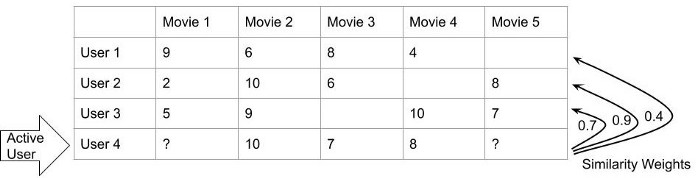


**User-Based Collaborative Filtering-**

The first step is to discover how similar the active user is to the other users.



for example, the similarity, could be 0.7, 0.9, and 0.4 between the active user and other users. These numbers represent similarity weights or proximity of the active user to other users in the dataset.

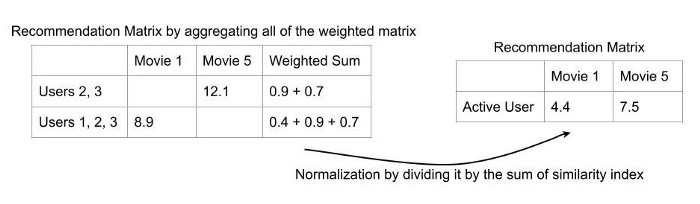


The blank columns in the previous table represent the movie is not rated by the user. Here, User 4 is the Active User for whom we have to recommend a movie out of Movie 1 and Movie 5.

Next, we’ll form the weighted rating matrix using User rating matrix and the similarity weights of each user. The movie columns are selected from which recommendation is to be made from the User rating matrix and each is multiplied by the respective user similarity index.



Now, the weighted rating matrix is aggregated to form the recommendation matrix. The normalized recommendation matrix is made by dividing each movie weighted sum by the sum of the similarity index.



The movie with the highest weight (here **Movie 5**) will be recommended to the Active User I.e. **User 4**. This is how the Collaborative Recommendation System works.

CHAPTER 6

**APPLICATION**

**6.1 Application**

* Increase Revenues
* Create customer satisfaction
* Personalize individual interest
* Reduce Time and effort
* Boost number of items per order

****

**Future Scope**

With more data and consumers, recommender systems become smarter. They not only can predict one user’s preference but a group’s preference, for example, a group of travelers or recommendations for a music playlist.

Detecting unfairness and reducing bias in algorithms are other important considerations. For example, identification of popularity bias is challenging when an item or a movie is very popular or frequently rated, but not relevant for every user.

* Integration with Web Application like Jsp , Servlet
* Integration with Database like
* Hive, Hbase, Mongodb, Couch db
* Cloud based recommendation Service
* Integration of Mahout , Graphlab and Google prediction
* based recommendation services.
* Mobile application integration

**6.1 SUMMARY**

Recommender systems are a powerful new technology for extracting additional value for a business from its user databases. These systems help users find items they want to buy from a business. Recommender systems benefit users by enabling them to find items they like. Conversely, they help the business by generating more sales. Recommender systems are rapidly becoming a crucial tool in E-commerce on the Web. Recommender systems are being stressed by the huge volume of user data in existing corporate databases, and will be stressed even more by the increasing volume of user data available on the Web. New technologies are needed that can dramatically improve the scalability of recommender systems.

**REFERENCE**

* https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada
* https://towardsdatascience.com/recommender-systems-in-practice-cef9033bb23a
* https://en.wikipedia.org/wiki/Recommender\_system
* http://infolab.stanford.edu/~ullman/mmds/ch9.pdf
* https://link.springer.com/article/10.1023/A:1021240730564
* https://arxiv.org/pdf/1707.07435.pdf
* <https://link.springer.com/article/10.1007/s11257-019-09256-1>