

Artificial Immune Systems Assignment: Movie Recommendation System

In this assignment, you will implement a Movie Recommendation system using Artificial Immune Systems (AIS), based on the Continuous Immune Network theory. To assist in the implementation, we provide some basic functions coded in Python, along with databases for users and movies.

Scenario

You have just been assigned to a new project to develop a platform that aims to rival the Internet Movie Database (IMDB). You already have a running platform sporting both movie and user bases, but there's still work to be done, more specifically a system that recommends movies to existing users. The system shall recommend movies to users based on their taste, which is inferred from their movie ratings. Luckily, you have just remembered an incredibly interesting class you had on Artificial Immune Systems, and that one of the algorithms taught in that course had to do with Immune Networks and their applications in recommender systems. Therefore, you decide to use a Continuous Immune Network to encode your solution, which is to find existing users that can provide meaningful movie recommendations to new users.

The existing user database contains around 1500 users, where each one has exactly 150 movie ratings. To test the system, you decide to use one of the users of this database as the antigen and the rest as antibodies. To build an acceptable pool of candidate recommended movies, you feel that **10** antibodies are enough, and that from these 10 antibodies, **5** movies should be selected as the final recommendations.

“So, what should I do”?

If you use our base code you must implement **three** main functions:

- One capable of measuring the affinity between two users. You can start with the ones shown in class. OR Use your creativity, there is no “best way” to implement this function.
- One that finds 10 antibodies to match the antigen using the Continuous Immune

Network. An idea for this was given in class, but if you are still having trouble with it, consult the references.

- One that recommends **5** movies based on the users found by the previous function. Once again, your creativity will shine here.

To aid you, there are useful references in the reference section. You may consult them for a better understanding of the Continuous Immune Network theory and their applicability on recommender systems.

Resources provided

Databases

We provide two databases:

- ***users.txt*** A text file containing a JSON database of users. You can obtain this file [here](#). Each user contains 150 movie ratings.
- ***movies.txt*** A text file containing a JSON database of 248 movies. This file is already on the repository.

Source files

We also provide the following source files:

- [***user.py***](#) - An implementation of the user class.
- [***ais.py***](#) - A sketch of the main class, where you will implement your solution.
- [***movieDB.py***](#) - A class to handle the movie database.

Feel free to use other programming languages as well, but we recommend at least taking a look at the Python files to better understand what you should implement. Keep in mind that if you choose to use another language, you'll have to write your own JSON handlers to read from the database.

Getting Started

Our sources are provided via a Git repository. To start coding your assignment, just execute the

following steps:

1. Clone the repository using the command (you can also download it [here](#)):

```
$ git clone https://bitbucket.org/kirajustice7/ais_assignment.git
```

2. Download the user database [here](#) and put it in the same folder as the [ais.py](#) file.
3. Execute the command:

```
$ python ais.py
```

If everything worked out you should see the following output:

```
Database loaded!
```

```
User database loaded!
```

```
Star Wars: Episode V - The Empire Strikes Back 8
```

... which means you're pretty much set up!

We gave you the parachute, now you just need to jump! Happy coding! :)

References

- [A survey about Artificial Immune Systems](#)
- [A Recommender System based on the Immune Network](#)