**Recommendation system**

An interesting objective of movies characteristics analysis is a recommendation system. That is a system that can suggest a film to user based on its previous preferences.

A common basis for this kind of solution is the *collaborative filtering* that can accomplish this complex task exploiting behaviour similarity with other users. The usage of this strategy could be accomplished in different ways and for different context: Music, Social network, News, and everything else that associate a user to its preference for different but homogeneous items. [CIT. RICHIESTA]

In our requirements we have a high scalability and the ability to work with a vast amount of data, reason why we opted for Alternating Least Squares.

**Collaborative filtering with ALS**

The alternating least squares algorithm substantially works on a matrix of user and item which element are level of appreciation of a user for a certain item. Of course, not every element of this matrix will be available, and a correct completion of this matrix is the solution to our problem.

[DETTAGLI MATEMATICI

Lucky a scalable implementation of ALS algorithm, based on spark [DESCRIVERE DA QUALCHE PARTE SPARK], exists [https://spark.apache.org/docs/2.4.1/ml-collaborative-filtering.html].  
This implementation doesn’t require a explicit matrix building and this is particularly advantageous given the shape of our dataset, in which every rating is represented by a tuple containing a identifier for the user, an identifier for the movie and the rating in a range between 0 and 5.

For comparison reason we built a *null model* in which the prediction of the rating is given by the average rating of the considered movie.

To compare the models, we opted for the Root Mean Square Error and r^2.

The parameters that can be tuned in this model are the numbers of iteration and the regularization one.

After a series of tries the best resulting model was given by 25 iterations and 0.15

|  |  |  |
| --- | --- | --- |
| **Model** | **Test RMSE** | **Test r^2** |
| Null (base) | 0.9335 | 0.1828 |
| ALS | 0.8984 | 0.3756 |

Unfortunately, with a higher number of iterations spark, with a single node configuration, cannot handle the training without fail.

But this should be a trivial problem to solve with a cluster of machines.

Another solution could be use a slightly different version of the algorithm: ALS-NCG[https://spark-packages.org/package/mbhynes/als-ncg] that, as has been shown, requires less iteration to converge[<https://arxiv.org/pdf/1508.03110.pdf>]

**Improve the rating estimations using a hybrid approch [SVILUPPI FUTURI]**

In most of this paper we focus on the analysis of the factors that contribute to the success of a film.

We concluded that plot, genre, and actors give significant representation of a film success, consequently we can deduce that at the same way they could be used to represent their quality.

Then the idea is to use these to define the film itself and compute the similarity between different film and consequently adjust the estimated ratings.

Applying this idea to build a working model could improve the, already significant, results of collaborative filtering.

TODO :D lo mettiamo nei futuri sviluppi?