Louvain School of Statistics, Biostatistics and Actuarial science

Data mining and Decision making: LSINF2275

Collaborative Recommendation Project

Realize by:

Stanley FORPA
Affiliation: DATS
Noma: 94581500

Ernesto Martinez del Pino
Affiliation: SINF
Noma: 99371800

Academic year 2018/2019

Contents

1 Introduction

In this work, we will build a system for finding people who share tastes and for making automatic recommendations bases on items that other people like. We will implement and compare three different algorithms for collaborative recommendation on real dataset: KNN, SGD and RandomForest. We splited the dataset into two parts for train set and test set. Then we splited again the train set into 10 folds. Finally, we build the model using one fold vs rest.

2 Mesures

2.1 Learning Score in Train set (LST)

This mesure reflect how our model can predict the data that it has used in in the training phase. We calcule this by dividing the number of labels well predicted by the number of real labels.

$$LST = \frac{\#((S = B)\&(B \neq 0))}{\#(B \neq 0)}$$

2.2 Mean Absolute Error (MAE)

This measures the average magnitude of the errors in a set of predictions, without considering their direction. It's the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \widehat{y}_i|$$

2.3 Root mean squared error (RMSE)

It is a quadratic scoring rule that also measures the average magnitude of the error. It's the square root of the average of squared differences between prediction and actual observation.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2}$$

2.4 Percentage

It is the percentage of target well predicted.

2.5 Precision

It means the percentage of your results which are relevant.

$$Precision = \frac{Truepositive}{Truepositive + FasePositive}$$

2.6 Recall

It refers to the percentage of total relevant results correctly classified by your algorithm.

$$Recall = \frac{Truepositive}{Truepositive + FaseNegative}$$

2.7 Recall-Aprox

We define out own mesure that it consist of removing 20 links with good score before fitting process. Then, we predict these 20 links and calcule the success rate.

$$RecallAprox = \frac{\sum_{i=1}^{k} |y_i - \widehat{y}_i|}{k}$$

3 Similarity

Consumer similarity matrix $X = (X_{st}), s, t = 1, 2, \ldots, M$. It calculates the similarity score X_{st} on the basis of the row vectors of data using a vector similarity function. A high X_{st} indicates that consumers s and t might have similar preferences because they previously watch many of the same film. We choose a cosine similitude to build the model:

$$SimCosine(\overrightarrow{x}, \overrightarrow{y}) = \frac{\overrightarrow{x} \cdot \overrightarrow{y}}{\|\overrightarrow{x}\| \cdot \|\overrightarrow{y}\|}$$

It usually generates better results the transpose which has more rows. In this case there are more films than users. All guides recommend using cosine distance we have check with other distances (euclidean and cityblock). It is true that they give worse values in general.

4 K-nearest neighbors algorithm for recommendation (KNN)

Our model is a user base collaborative filtering k-nearest neighbors. This algorithm predicts a target consumer's future transactions by aggregating the observed transactions of similar consumers. The more similar the set of consumers who watch the target film is to the target consumer, the more likely the target consumer will be interested in that film.

4.1 Knn Model description

The knn approach for collaborative filtering consist to look in the data, k users that have more similar score as a fixe user. Then predict the score of the fixe user for an item as the mean of the score of the k similar users we found. To ameliorate our model, we set the score of 2.5 for the fixe user if none of his k nearest neighbors gave a score for the item. But we note when k is big, the probability of having at least one nearest neighbor that give a score to the item increase and the error of our model is ameliorated. We build a model on a training dataset of 80.000 users. Then we apply it on a test dataset of 20.000 users.

Parameters:

- Number of neighbors= 15

4.2 Matrix result

We apply the model to predict users score in the test set, then we compare to the original scores and found prediction error:

$$MAE = 0.76825$$

It is a mean absolute error for non-zero sores in the test set given by

$$\frac{1}{m*} \sum_{i=1}^{M} \sum_{J=1}^{N} |score[i,j] - predscore[i,j]| \cdot 1_{\{score[i,j] \neq 0\}}$$

Where M is the number of user, N the number of item and m* the number of non-zero scores in the test set. Using the measures defined above we obtain the following table:

Name	LST	MAE	RMSE	Perc	Precision	Recall	Recall-Aprox
Fold 1	0.50968325	0.788875	1.13525878	0.409	0.00217692	0.35854674	0.6
Fold 2	0.50643234	0.781	1.13032075	0.41675	0.00219497	0.36265806	0.35
Fold 3	0.50662684	0.777375	1.12755266	0.420375	0.00223608	0.363547	0.5
Fold 4	0.50848847	0.7745	1.12976768	0.4255	0.00225477	0.37422013	0.45
Fold 5	0.5080439	0.769875	1.11797809	0.42375	0.00223431	0.36761551	0.5
Fold 6	0.50850236	0.777	1.11800604	0.414125	0.00214207	0.35746596	0.35
Fold 7	0.50719644	0.7738125	1.12142487	0.417375	0.00215146	0.36124762	0.5
Fold 8	0.50709919	0.7796875	1.13425361	0.420625	0.00227245	0.36926162	0.4
Fold 9	0.50719644	0.772	1.11789423	0.41875	0.00223681	0.36765091	0.4
Fold 10	0.50896082	0.798875	1.14373292	0.404375	0.00217946	0.35300191	0.3
Mean	0.50782301	0.7793	1.12761896	0.4170625	0.00220793	0.36352155	0.435

5 MultiDimensional Classification (MDC)

5.1 MDC Model Description

In this context we have 2 sets, one of known variables and another with unknown variables. The model will predict all the unknown variables at the same time using the knowledge extracted from known variables.
Mathematically formulated:

$$\mathrm{f} : \mathbb{R}^n \to \mathbb{R}^m$$

It is equivalent to compute m classifiers. This is what we will do to simulate this situation. To simplify, we assume and use the same classifiers to explain each columns. From this moment we can apply everything we know about machine learning. Mathematically formulated:

Each f_i represents a single column predictor, therefore

$$\forall i \in (1, m)\ f_i : K^n \to K$$

$$f_i\left(\vec{k}\right) = k_i$$

In our problem, the K input is the cosine similarity matrix between users or films. These two possibilities are allowed because we can predict the score column of a user or a film. But in practice, it is usefull to take matrix which have more rows. The reason is very simple, most of the classifiers work better if they have more examples.

5.2 Stochastique Gradient Descent (SGD) Classifier

We are going to start with a simplest classifier, the stochastic gradient descent (SGD). The SGD is an efficient and useful algorithm which produces linear models. It try to separate data using a hyperplane. If data set is completely separable with a hyperplane or in this case, a set of hyperplanes with multilabel classification, then we can find the coefficients with SGD algorithm. Obviously this is not our case. To show this conclusion we calcule MAE.

$$MAE = 0.8719$$

Parameters:

- Number of iterations= 15
- Stopping criterion= 10^{-4}

5.2.1 Matrix result

Using the measures defined above we obtain the following table:

Name	LST	MAE	RMSE	Perc	Precision	Recall	Recall-Aprox
Fold 1	0.89269241	0.876125	1.23273071	0.376125	0.00184829	0.35277293	0.35
Fold 2	0.88559322	0.87575	1.23298418	0.379375	0.00189054	0.37010624	0.45
Fold 3	0.88987219	0.89025	1.2514991	0.3725	0.00182961	0.35389654	0.25
Fold 4	0.88909419	0.873625	1.23323761	0.3835	0.00188706	0.36114791	0.3
Fold 5	0.88255071	0.87875	1.24046362	0.380625	0.00188605	0.35931073	0.35
Fold 6	0.89661017	0.864375	1.21906727	0.3835	0.00189338	0.36136052	0.5
Fold 7	0.88912198	0.879125	1.23526313	0.3755	0.00187604	0.35864281	0.35
Fold 8	0.88644068	0.891625	1.24924977	0.37075	0.00183285	0.36091769	0.6
Fold 9	0.88895527	0.8825	1.2346052	0.372625	0.0018746	0.36212383	0.35
Fold 10	0.87521534	0.872875	1.23110723	0.379875	0.00183936	0.35520996	0.35
Mean	0.88761462	0.8785	1.23602078	0.3774375	0.00186578	0.35954892	0.385

5.3 Random Forest (RF) Classifier

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. It returns a non-linear model. We have used this classifier because

$$MAE = 0.76745$$

Parameters:

- Number of estimators= 15

5.3.1 Matrix result

Using the measures defined above we obtain the following table:

Name	LST	MAE	RMSE	Perc	Precision	Recall	Recall-Aprox
Fold 1	0.999611	0.780625	1.14842283	0.42875	0.00203164	0.38862471	0.4
Fold 2	0.99952765	0.768875	1.12965703	0.427875	0.00205175	0.39514724	0.35
Fold 3	0.99956932	0.78	1.15228035	0.4325	0.00210188	0.39747512	0.5
Fold 4	0.99958322	0.774625	1.12821762	0.42025	0.00201064	0.38505907	0.45
Fold 5	0.99956932	0.765875	1.12277558	0.42975	0.00205006	0.38976421	0.45
Fold 6	0.999611	0.779875	1.14088781	0.426	0.00203043	0.38606469	0.45
Fold 7	0.999611	0.782375	1.14460692	0.42425	0.00201877	0.38180888	0.6
Fold 8	0.99963879	0.804125	1.17510638	0.41825	0.00198747	0.37605974	0.35
Fold 9	0.99965268	0.78125	1.13896883	0.42	0.0020115	0.37895112	0.6
Fold 10	0.99956932	0.774375	1.12976768	0.42175	0.00202887	0.38518589	0.15
Mean	0.99959433	0.7792	1.1410691	0.4249375	0.0020323	0.38641407	0.465

6 Comparison of the three algorithms

After collect all the data. Now, we can make a decision between KNN, SGD and Random-Forest. We are going to describe the advantages and disadvantages of each one:

- KNN: this algorithm give us two good mesures. The first is the MAE = 0.7793 which shows the mean distance between the predicted value and the real value, and the second is RecallAprox = 0.435 which means that we make a good prediction when we talk about high score. But we get but news when we look at LST = 0.50782301 which shows how the model is learning from data train. Does it means that we only need the half of the set to make a good prediction? We are sure that it is not possible. Thus, we are going to see that in the other algorithms this fact does not happen.
- SGD: here we have first algorithm that we must use in machine learning to have an idea of how difficult the problem is. LTS value = 0.88761462 proves that we can not split data with hyperplanes. In conclusion, this model learns good from data train without overfitting but it does not have enough power. We see this in MAE value = 0.8785. It is worse than KNN in test mesures in general.
- RandomForest: this algorithm join the advantages from KNN and SGD. It learns from train data with LST = 0.99959433. Is it possible that we are making overfit? The train error is practically 0. To prevent overfit we can start looking variance in MAE = $9,67*10^{-5}$ which is very low. Cross Validation does not resolve the overfit problem but it helps us to reduce it. We get the best MAE = 0.7792 and Recall-Aprox = 0.465 values.

If we think about own collaborative recommendarion problem like a model that we can implement with hands, I am sure that we will use rules like if you wacht these films you probably will see those films. Random forest algorithm use this kind of rules to build its model. Thus, it is normal that it works and we choose this as the solution of this problem.

7 Conclusion

We have discussed about cross validation, overfitting and some of the most important measures used in machine learning. We have presented 3 algorithm (linear vs non-linear). Finaly, we can say the implementation of these three models allowed us to master the concept of collaborative filtering and know one of the more important algorithm adapted to it in supervise learning.

8 Fail idea

We have been thinking about the number of data and how to increase them. We decide to make a generation machine learning algorithm. In each generation it adds from test set the best target predicted based on LST value to the train set. Thus, as we pass generation we have more data available to fit. We show that MAE increase in each generation because we don't have a good enough error in test. In conclusion the model does not help us to predict difficult values. It works as a chaotic system spreading small mistakes.

9 Somme reference and links

KNN recommendated document.

Stochastic Gradient Descent.

Random Forest.

Idea for transpose similary matrix.