Standard Recognition Computational Work
Academic Year 2022 - 2023
Γ. Tsihrintzis D. Sotiropoulos

Subject: Recommendation Algorithms Using Artificial Neural Networks and Data Clustering Techniques

This work aims to develop algorithms for movie recommendation using artificial neural networks and clustering techniques. The dataset on which you will work can be downloaded from https://ieee-dataport.org/open-access/imdb-movie-reviews-dataset and concerns on a set of users $\boldsymbol{U} = \{U_1, U_2, \cdots, U_N, \}$ who express preferences over a set of movie objects $\boldsymbol{I} = \{I_1, I_2, \cdots, I_N\}$. The user $u \in \boldsymbol{U}$ attributes to the movie $i \in \boldsymbol{I}$ the preference degree $R(u,i) \in \boldsymbol{R}_o = \{1,2,3,4,5,6,7,8,9,10\} \cup \{0\}$, where 0 indicates the absence of an evaluation and not a zero preference score.

The set $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_T\}$ of available data consists of records of the form $\boldsymbol{x}_k = [u_k, i_k, r_k, t_k]$ where $u_\kappa \in \boldsymbol{U}$ and $i_\kappa \in \boldsymbol{I}$ with $r_\kappa = R(u_\kappa, i_\kappa) \in \boldsymbol{R}_o$. The value t_k corresponds to the date of entry of this record.

Data pre-processing:

1. Find the set of unique users $\boldsymbol{U}$ and the set of unique objects .
2. Consider the function Φ: $\mathbf{U} \to \boldsymbol{P}(\boldsymbol{I})$ (where $\boldsymbol{P}(\mathbf{I})$ is the dynamo set of) which $\forall\, u \in \mathbf{U}$ returns the set $\varphi(u) \subset \boldsymbol{I}$ of objects evaluated by user . We can therefore write that $\varphi(u) = \{i \in \boldsymbol{I}: R(u,i) > 0\}$. Restrict the sets of unique users $\mathbf{U}$ and unique objects $\boldsymbol{I}$ to the respective sets $\mathbf{U}^{\wedge}$ and $\mathbf{I}^{\wedge}$ such that $\forall\, u \in \boldsymbol{U}^{\wedge}$, $_{Rmin} \leq |\varphi(u)| \leq {}_{Rmax}$ where $_{Rmin}$ and $_{Rmmax}$ is the minimum required and maximum allowed number of evaluations per user. Consider obviously that $|\boldsymbol{U}^{\wedge}| = \boldsymbol{n} < \boldsymbol{N}$ and $|\boldsymbol{I}^{\wedge}| = \boldsymbol{m} <$.
3. Create and graphically represent frequency histograms for the number and time range of each user's ratings.
4. Create an alternative representation of the dataset as a set of preference vectors $\boldsymbol{R} = \{\boldsymbol{R}_1, \boldsymbol{R}_2, \cdots, \boldsymbol{R}_n\}$ with $\boldsymbol{R}_j = R($ $) \in \boldsymbol{R}_o{}^m$, $\forall\, j \in [n]$. In particular, we can write that:

$$(\,) = \begin{cases} (\,,\,), (\,,\,) > \\ 0, \end{cases} \qquad (\,)$$

$$0, \qquad (\,,\,) > 0.$$

Data Grouping Algorithms

1. Organize the limited set of users $\mathbf{U}^{\wedge}$ into L clusters of the form $\mathbf{U}^{\wedge} = \boldsymbol{U}_{G_1} \cup \boldsymbol{U}_{G_2} \cup \cdots \cup \boldsymbol{U}_{G_L}$ such that $\boldsymbol{U}_{G_a} \cap \boldsymbol{U}_{G_b} \neq \varnothing$, $\forall \boldsymbol{a} \neq \boldsymbol{b}$ based on the vector representation of their preferences via the set . The process of clustering the user preference vectors $\boldsymbol{R}_j = R($ $) \in \boldsymbol{R}_o{}^m$, $\forall\, j \in [n]$ onto the restricted set of objects , can be performed by appropriately parameterizing the **k-means** algorithm by varying the metric that

assesses the distance between two preference vectors for a pair of users u and v as $dist(R_u, R_v)$. In particular, you can use the following metrics (*):

a. $_{euclidean}dist(R_u, R_v) = \sqrt{\sum^{m} |R_u(k) - R_v(k)|^2 \lambda_u(k)\lambda_v(k)}$ (2) such that:

$$\lambda_u(k) = \begin{cases} 1, & r(i_k, u_j) > 0, \\ 0, & r(i_k, u_j) = 0. \end{cases} \quad (3)$$

The function $\lambda_j()$ evaluates whether or not user u_j has evaluated object i_k. Obviously, through this function we can if we calculate the value of $|\varphi(u)|$ which evaluates the number of evaluations provided by user u in total as: $|\varphi(u)| = \sum^{n}_{k=1} \lambda_u(k)$.

b. $_{cosine}dist(R_u, R_v) = 1 - \frac{\sum_{k=1}^{n} r_u(k) r_v(k) \lambda_u(k)\lambda_v(k)}{\sqrt{\sum_{k=1}^{n} R_u(k)^2 \lambda_u(k)\lambda_v(k)} \sqrt{\sum^{m} r_v(k)^2 \lambda_u(k)\lambda_v(k)}}$ (4)

c. Graphically represent the clusters of users identified by the k-means algorithm for each of the above metrics for various values of the parameter k.

d. Comment on the effectiveness of these metrics in assessing the similarity between a pair of user preference vectors R_u and R_v.

*** The metrics represented in equations (2) and (4) are computed for each pair of vectors R_u and R_v only for the subset of objects evaluated by both users u and v via the utility function $\lambda()$.**

Recommendation Generation Algorithms Using Artificial Neural Networks

2. Create an alternative organization of the restricted set of users into L clusters $U = U_{G_1} \cup U_{G_2} \cup \cdots \cup U_{G_L}$ such that $U_{G_a} \cap U_{G_b} \neq \emptyset$, $\forall a, b \in [L]$ with $a \neq b$ using the following metric (**):

$$dist(u, v) = 1 - \frac{|\varphi(u) \cap \varphi(v)|}{|\varphi(u) \cup \varphi(v)|} \quad (5)$$

**** The clustering algorithm for dealing with this query is of your choice. The basic requirement, however, is that it can operate on the square matrix of distances between users described by relation (5).**

a) Explain what this metric expresses and identify its disadvantages compared to the metrics described in relations (2) and (4) in the light of the particular organisation of users that it may entail.

b) The metric described by relation (5) can be used in order to determine the set $N(u) = \{u^{(1)}, u^{(2)}, \cdots, u^{(k)}\}$ of k nearest neighbors of a user $u_a \in \boldsymbol{U}_G\, a$, $\forall \boldsymbol{a} \in [\boldsymbol{L}]$. Therefore, for each user u_a of each cluster $\boldsymbol{U}_{G\,a}$ we can set the vector of personal preferences $R_{u\,a}$ as well as the vectors of k nearest neighbors of $Ru(1)$, $Ru(2)$,..., $Ru(\kappa)$ within cluster $\boldsymbol{U}_G$. Objective of this question is to develop a multilayer neural network for each user cluster $\boldsymbol{U}_{G\,a}$ $\mu\varepsilon\, a \in [L]$ which will approximate the ratings of each user within it by the ratings of its k nearest neighbors via a function of the form:

$$_{u\,Ra} = f_a\,(Ru(1), Ru(2), \cdots Ru(k))),\ \forall u_a \in \boldsymbol{U}_G\, a,\ \forall a \in [L] \mathbf{(6)}$$

c) The set of users in each cluster can be further partitioned into a subset of users for training the neural network $\boldsymbol{U}^{train}$ and a subset of users for testing the performance of the neural network $\boldsymbol{U}^{test}$ such that $\boldsymbol{U}_G = \boldsymbol{U}^{train} \cup \boldsymbol{U}^{test}$ with $\boldsymbol{U}^{train} \cap \boldsymbol{U}^{test} = \emptyset$. The most appropriate configuration of the data for the training of this neural network of the question could be reproduced as follows:

Suppose that the set of users participating in the cluster $\boldsymbol{U}_{G\,a}$ is given by $\boldsymbol{U}_{G\,a} = \{u_{a,1}, u_{a,2}, \cdots, u_{a,n}\ a\}$ with $n_a = |\boldsymbol{U}_G\, a|$. Then the set of feature vectors and the corresponding set of labels could be organized into a table of the form:

$$\begin{bmatrix} Ru(1) & Ru(2) & \cdots & Ru(k) \\ a,1 & \square,1 & & a,1 \\ Ru(1) & Ru(2) & \cdots & Ru(k) \\ a,2 & \square,2 & & \square,2 \\ \vdots & \vdots & \vdots & \vdots \\ Ru(1) & Ru(2) & \cdots & Ruu(k) \\ \square,\square\square & \square,\square\square & & \square,\square\square \end{bmatrix} \text{and} \begin{bmatrix} \square\square\square,1 \\ \square\square\square\square,2 \\ \vdots \\ [\square\square\square,\square \end{bmatrix}$$

d) The approximate accuracy of these neural networks can be measured by the metric of the mean absolute error between the actual and estimated user ratings. Present tables of your results for both training and testing accuracy for each user cluster.

You can work in groups of a maximum of 3 students. The implementation of the project can be done in MATLAB or Python. The code of your project should be accompanied by detailed documentation.

Good luck!