Sample Project: Movie Recommendation

Spring, 2018

Consider the movie recommendation problem studied in class: in this problem the goal is to predict the ratings users assign to movies in a minimum mean-squared error sense. In the provided dataset (`movie-data.zip`), viewer ratings corresponding to 9066 movies and 671 users are provided together with a feature vector for each movie (whose components are indicators of different genres).[1] The data is given in `csv` format with a header row describing the meaning of each column:[2]

- `movie-titles.csv`: title and id of the movies in the dataset (only for information purposes);
- `movie-features.csv`: binary (0-1-valued) genre features for the movies, each row contains variables `movieId,feature1,...,feature18`;
- `ratings-train.csv`: training set $\mathcal{D}$, containing `(userId,movieId,rating)` triplets in each row;
- `ratings-test.csv`: test set $\mathcal{T}$, containing `(userId,movieId,rating)` triplets in each row.

Here `userId` $\in \{1, 2, \ldots, 671\}$, `movieId` $\in \{1, 2, \ldots, 9066\}$, and `rating` $\in \{0.5, 1, 1.5, \ldots, 5\}$.

The task in this problem is to develop and test different predictors for the movie ratings whose performance is measured by the squared error. Training should only be performed on the training set, final test results should be measured on the test set, and no parameter of the training method (including which algorithm to use) should be selected based on the test set. For each subquestion, explain clearly and concisely what you do and why (including formulas with derivations if necessary), and present the answers in the most meaningful way.

(a) Constant base predictors: The simplest predictor for this problem is to provide a rating prediction for any movie (independent of the user) or any user (independent of the movie). Find such predictors and report their performance. Which one would you prefer, constant ratings based on the movies or the users?

(b) Linear regression baseline: Based on the features provided for the movies, estimate the rating given by each user using linear regression, possibly with some added regularization. That is, if $\hat{v}_j \in \mathbb{R}^d$ is the (possibly normalized) feature vector for movie j, find weights $u_i \in \mathbb{R}^d$ for user i with the aim of minimizing the error $(r_{ij} - u_i^\top \hat{v}_j)^2$ over the test set $\mathcal{T}$.

(c) Linear regression with transformed features: Suggest some way of transforming the features in a non-linear manner and repeat the above with the transformed features.

(d) Collaborative filtering: Learn simultaneously the weights for the users and the features for the movies by devising predictions of the form $u_i^\top v_j$ where $u_i, v_j \in \mathbb{R}^K$ for some $K \geq 1$ and v_j is the K-dimensional (learned) feature vector for movie j.

[1] The data is extracted from a dataset provided by GroupLens about ratings collected by MovieLens `https://grouplens.org/datasets/movielens/`. For licensing and other information, see `http://files.grouplens.org/datasets/movielens/ml-latest-small-README.html`.

[2] Recommendation: use built-in csv-reader functions to read the files.

In all the above questions, select the parameters (such as K, coefficients for penalties, step size for gradient descent, etc.) in a disciplined way (e.g., using cross-validation) and do not forget to report how these are done. If you are using gradient descent, do not forget to derive the update rule and give the exact algorithm you are using. Analyze the results and report your findings (including a comparison of the solutions you get at different stages of this problem). You can use any available standard library functions in your code (e.g., for ridge regression).