Institute for Pure & Applied Mathematics
IPAM
University of California, Los Angeles

Doubly Ensemble Movie Prediction With Social Media Data Using TBEEF

Christopher V. Rackauckas, University of California, Irvine
Xu Chenxiao, Hong Kong Baptist University
Colin Jarvis, Macalester College
Cai Weijie, Hong Kong University of Science and Technology
Advisor: Avery Ching, Hong Kong University of Science and Technology

香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

Introduction

Recommender systems have become commonplace in recent years:

- Amazon shows users other compelling purchases.
- Netflix shows users new movies that may be of interest.
- Google shows targeted advertisements.

Many companies have large swaths of user's information from social networks such as Facebook. As the largest internet company in China, Baidu Inc. runs popular social media networks and is looking to use this information to improve their video service.

Goal: Develop algorithms to incorporate social media data into recommendation systems.

The Problem

Baidu Inc. provided our team with a dataset containing:

- Approximately 10,000 users rating around 8,000 movies.
- User's browser history.
- The users that a given user has "followed".
- Movie tags as set by users.

Our project was to train a model which would be predict a user's rating for a movie. Our model was tested on a hidden dataset using the RMSE metric:

$$RMSE = \sum_{i}^{n} (\hat{y}_i - y_i)$$

where $\hat{y}_i$ is the model prediction and y_i is the known user movie rating. As the RMSE measures the distance between prediction and reality, our goal was to find a model which minimizes the RMSE.

Particular difficulties included:

- Sparsity: users do not rate most movies.
- Computational efficiency: the algorithm must run in a reasonable time on large datasets.

Summary of Main Techniques

Solving Sparsity: Factorizations

- Idea: To find the interaction (estimated correlation) between Alice in Wonderland and Star Trek, assume the interaction is the same as some other related and rated movie (i.e. Star Wars)
- This can be done mathematically modeling interactions with an $n \times m$ matrix:
  - Let m be the number of movies.
  - Let $n \ll m$.
  - Define the interaction effect to be the dot product between columns.
- Training will make similar movies have similar columns causing high interaction effects.

Ensembling: "Knowledge of the Crowd"

- In Who Wants to Be a Millionaire, the "ask the audience" option was almost always right whereas "phone a friend" (supposed expert) was usually wrong!
- Idea: Generate many recommendations from diverse models and aggregate the results.
- Work to develop and aggregate many models instead of a "best" model.

Introduction to TBEEF

Our team decided to create a software package for developing prediction algorithms utilizing a doubly ensemble framework. This program is called TBEEF, Triple Bagged Ensemble Ensemble Framework. It has three steps:

1. Model: Runs factorization models on sparse datasets to generate predictions.
2. Hybrid: Uses an ensemble of ensembling techniques to generate predictions.
3. Synthesize: Aggregates the hybrid step predictions using one ensemble technique.

A plugin interface has been implemented to let users easily create factorization models for the Model step and ensemble models for the Hybrid and Synthesize steps. This program is publicly distributed with the MITx license and can be found on the Machine Learning Open Source Software (MLOSS) repository under the name *TBEEF: Triple Bagged Ensemble Ensemble Framework*.

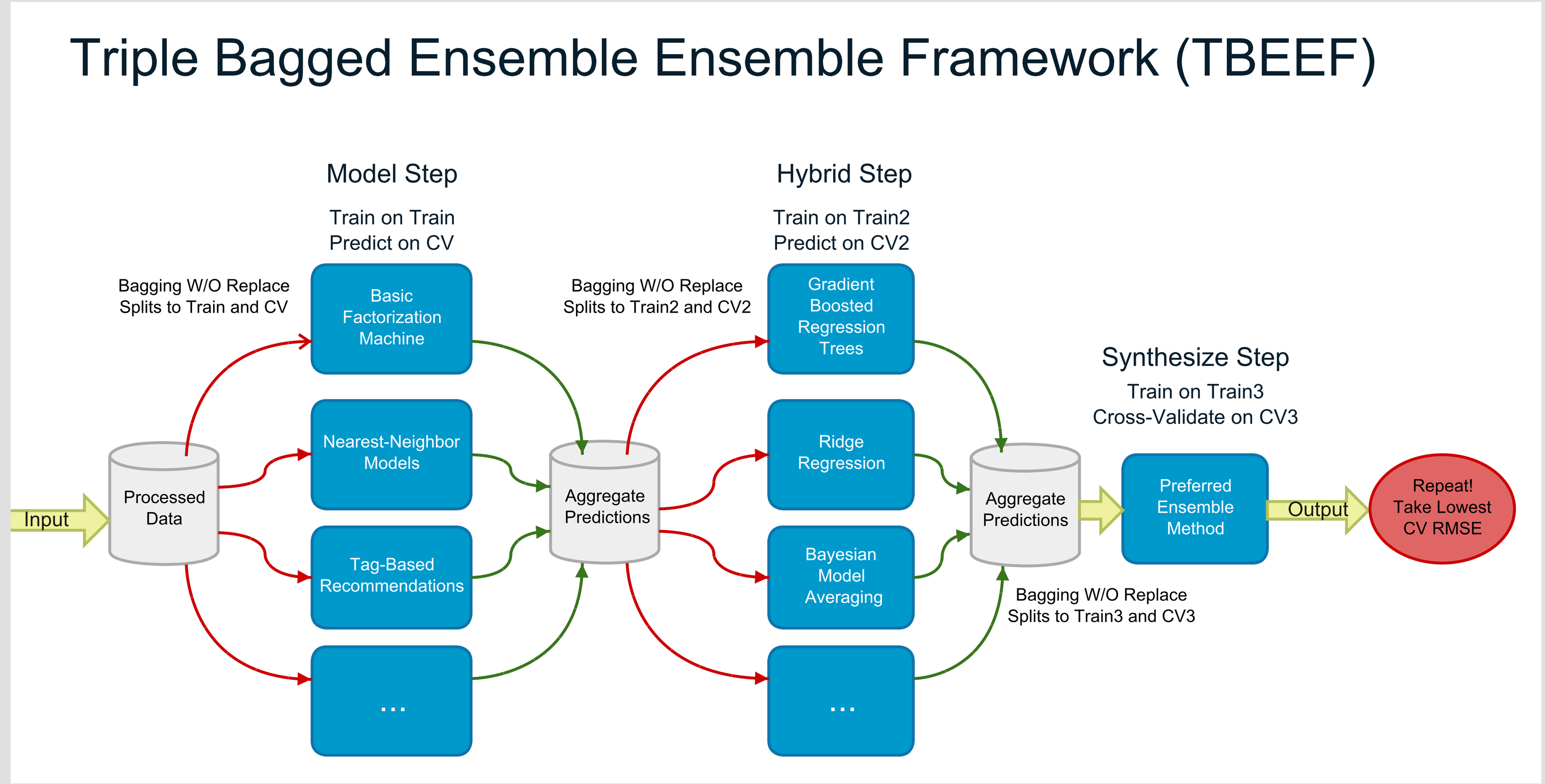

Diagram of the TBEEF program. Shown are the three steps, the Model step, the Hybrid step, and the Synthesize step. Example models are shown in the Model and Hybrid steps.

The Factorization Model

General Factorization Model:

$$\hat{y} = w_0 + \sum_{j=1}^{m} w_j x_j + \sum_{j=1}^{m} \sum_{k=j+1}^{m} \langle v_i, v_j \rangle x_i x_j,$$

- $\hat{y}$ is the predicted movie rating.
- x is the data vector to be specified for the user/movie pair.
- w_0 is the global average rating.
- w_i is the average for the ith parameter.
- $\langle v_i, v_j \rangle$ is defined as the dot product between the i-th and the j-th rows of the factor matrix $\mathbf{V}$.

Standard Factorization Model
The standard model is to use information of only the user and the movie. Index the movies and users numerically and use binary variables to denote the user/movie pair with a data vector:

$$(u, i) \rightarrow \mathbf{x} = (\underbrace{0, \ldots, 0, 1, 0, \ldots, 0}_{|\mathbf{U}|}, \underbrace{0, \ldots, 0, 1, 0, \ldots, 0}_{|\mathbf{I}|})$$

to give the model:

$$\hat{y}(x) = w_0 + w_u + w_i + \sum_{f=1}^{k} v_{u,f} v_{i,f}$$

where w_u is the user's average rating and w_i is the movie's average rating.

Incorporating Social Media Information

Example Network Models

Followed Network Model
Using social data collected about the users, we construct our input vectors using user ID u, movie ID i, and a set of followed users S. Each friend in the vector is given a $1/m$ value where m is the total number of users followed by u. To encode this model into the factorization machine, one would use the input vector:

$$\mathbf{x} = (\underbrace{0, \ldots, 1, 0, \ldots}_{|\mathbf{U}|}, \underbrace{0, \ldots, 0, 1, 0, \ldots, 0}_{|\mathbf{I}|}, \underbrace{0, \ldots, 1/m, 0, \ldots, 1/m, \ldots, 0}_{|\mathbf{U}|}).$$

to define the model:

$$\hat{y}(x) = w_0 + w_u + w_i + \langle v_u, v_i \rangle + \frac{1}{m} \sum_{j=1}^{m} \langle v_i, v_{s_j} \rangle + \frac{1}{m} \sum_{j=1}^{m} w_{s_j}$$

$$+ \frac{1}{m} \sum_{j=1}^{m} \langle v_u, v_{s_j} \rangle + \frac{1}{m^2} \sum_{j=1}^{m} \sum_{j'>j}^{m} \langle v_{s_j}, v_{s_{j'}} \rangle.$$

This correlates the user's predicted ratings with those of the people who he/she follows on the social networks.

Browser History Model
Using browser history for each implicitly defined user, we give each movie viewed in the history a r_j/m value where m is the total number of movies in the history and r_j is either the rating of the movie l_j or the user never rated it, then simply a 1.

$$\mathbf{x} = (\underbrace{0, \ldots, 0, 1, 0, \ldots, 0}_{|\mathbf{I}|}, \underbrace{0, \ldots, r_1/m, 0, \ldots, r_m/m, \ldots, 0}_{|\mathbf{I}|}).$$

This penalizes movies for which the user browsed to but did not watch.

The Ensemble Problem

Ensemble models were used to aggregate predictions from many models under certain conditions:

1. Needed to prevent over-fitting data.
2. No worry about sparsity: factorization models generate a prediction for every movie.
3. Needed to account for biases induced by various models.

Overfit Protection: Bagging without replacement
To prevent over-fitting, bootstrap aggregation without replacement was used. Thus each model saw a random subset of the data with which to train. User/movie pairs used as predictions in the next round were omitted from the training set to reduce the bias imposed by utilizing multiple steps.

Ensemble Models

Example Ensemble Method: Ordinary Least Squares Regression (OLS)
Given j factorization models, one way to ensemble the predictions is to run a regression

$$y_i = \sum_{j=1}^{m} x_{i,j} \beta_j.$$

which calculates the "best" way to create a prediction as a weighted average of other predictions.

Advanced Ensemble Models

1. Random Forest Models (Bagged Random Forest, Conditional Random Forest): Split data into homogeneous regions to run OLS.
2. Gradient Boosted Regression Trees: Repeatedly cycle residuals back into OLS.
3. Bayesian Model Averaging Regression: Make predictions on all 2^j product combinations of predictions.

Acknowledgments

This research was made possible through the Research Industrial Projects for Students, Hong Kong (RIPS-HK), an undergraduate research opportunity ran by UCLA's Institute for Pure and Applied Mathematics (IPAM). This project was funded by the NSF. We thank our academic mentor Dr. Avery Ching, Hong Kong University of Science and Technology, and our industry client Baidu, Inc. for helping guide us through the project. For more information, please see our technical report, *Doubly Ensemble Movie Prediction With Social Media Data Using TBEEF*.