CS 761 Final Project Report

Submitted by: Sneha Rudra and Clarence Cheung

Abstract

Recommendation systems can be thought of as information filtering systems that predict ratings which users would assign to products or services that they have not used or experienced before. We apply the ideas of recommendation systems to Yelp dump dataset available for download from Kaggle. Yelp hosts and markets a web and mobile based application that publishes reviews and ratings for a variety of local businesses. We adopt a model based collaborative filtering approach to predict the popularity of businesses. Using feature extraction we obtain a set of possibly useful features from the dump data and apply regression models - Linear Regression, Ridge Regression and Lasso Regression to learn the weights. To deal with missing entries in the test set, we use the nuclear norm minimization based matrix completion method learnt in class and use the Proximal Gradient algorithm to solve all the underlying optimization problems. Lastly, for tuning our regularization parameter, we use cross validation and evaluate the different models based on metrics such as Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Normalized RMSE and MAE.

1 Introduction

According to Resnick and Varian [1], recommendation systems are systems that utilize the opinions of community users to help individuals more effectively identify content of interest from a potentially overwhelming set of choices. In recent years, recommender systems are widely popular across the internet and most commercial and social websites suggest options such as - movies, music, news, books, research papers, search queries, social tags, restaurants etc to users[2]. One of the websites that utilizes recommendation system extensively is Yelp. Yelp hosts a web and mobile platform by the same name that provides crowd-sourced reviews. Users write their reviews and rate products or services of a specific business on a scale of one to five (denoted by stars). These ratings can then be used by other users to make informed decisions.

2 Literature Comparison and Methodology

Approaches to generating recommendations can be broadly classified into two main categories: (i) Content Based Methods and (ii) Collaborative Filtering. In content-based recommendation methods [3], the system tries to generate user profiles to learn patterns in the user's past preferences and then items which are similar to those preferences are suggested. It is often common to request the user to fill out questionnares inorder to gather more data to create user profile. Content based methods are used in practice to recommend items that are labelled with tags/keywords. Hence if every item is characterized by a content profile -a set of keywords (which may have been obtained through feature extraction), then the user profile (vector indicating the relevance of a given keyword to the user) can be obtained from the content profiles of the items liked by the user in the past through means such as Rocchio algorithm [4]. Other content based approaches include building Bayesian classifier/decision tree to estimate probability of an item being liked. Usually classifiers take a long time to construct and are mostly used for smaller problems [5].

Collaborative Filtering methods take a different approach and try to suggest items to users based on the items previously rated by other like-minded users (those who would rate items similarly as the given user) [3]. Breese et. al [6] organize collaborative filtering algorithms into memory-based and model based. Memory- based methods predict the rating that the user would assign to an item as an aggregate of the ratings given to the same item by other similar users. Common aggregation strategies include simple average, user-similarity weighted sum etc. Several approaches such as Pearson correlation, cosine based and graph theoretic approaches [7] have been used to compute the user-similarity. It is expected that several user-specified ratings would be needed for this method to work well and in the case of missing ratings, experiments show that imputing values can help [6]. Model-based algorithms on the other hand first learn a model from a set of known ratings and use the model to make predictions. Several model-based collaborative methods that have been explored in the literature include statistical model for collaborative filtering [8], bayesian model [9], linear regression [10] etc.

2.1 Data Models

For this project we use model based collaborative filtering to predict business popularities. This choice is driven by the fact that model based approaches offer several advantages such as scalability, speed and avoidance of overfitting if the dataset is fairly representative [7]. In particular, we apply three regression models - Linear Regression, Lasso Regression and Ridge Regression. Each of these regressions can be formulated as an optimization problem with the following general form:

$$\min_{\mathbf{w} \in R^p} \sum_{i=1}^{n} l(y_i, \mathbf{x}_i^T \mathbf{w}) + \lambda \|\mathbf{w}\|_q^q$$

where n is the number of training samples, y_i denotes the ratings/stars for a given business, $\mathbf{x}_i$ denotes a set of features specific to the given business such as - #reviews for this business, whether the business is open, #funny, #useful and #cool votes assigned to the reviews for this business, average funniness, average usefulness and average coolness of the users who reviewed this business etc., $\mathbf{w}$ denotes the set of weights we wish to learn and λ is a regularization parameter that controls the tradeoff between the fit to the data and sparsity of the weight vector. Further,

- For the Linear Regression: $l(y_i, \mathbf{x}_i^T \mathbf{w}) = (y_i, \mathbf{x}_i^T \mathbf{w})^2$ and $\lambda = 0$
- For the Lasso Regression: $l(y_i, \mathbf{x}_i^T \mathbf{w}) = (y_i, \mathbf{x}_i^T \mathbf{w})^2$ and $q = 1$
- For the Ridge Regression: $l(y_i, \mathbf{x}_i^T \mathbf{w}) = (y_i, \mathbf{x}_i^T \mathbf{w})^2$ and $q = 2$

The goal is therefore to learn weights $\mathbf{w}$ so as to be able to predict the popularity y_i for a given business based on features $\mathbf{x}_i$

2.2 Matrix Completion based on Nuclear Norm Minimization

Since the feature matrix corresponding to the testing set was half-sparsed (we observed roughly 50% sparsity and hence $m \in n_1 \times n_2$ where the n_1 denotes the number of rows in the feature matrix and n_2 denotes the number of columns in the same), we employed matrix completion based on nuclear norm minimization used in class [11] to reconstruct the low rank matrix subject to the constraint that the completed matrix must agree with the values and the locations of the known entries. The specific optimization problem (assuming noisy observations) solved by the Singular Value Thresholding algorithm is:

$$\min_{\mathbf{X}} \tfrac{1}{2} \|R_\Omega(\mathbf{X} - \mathbf{M})\|_F^2 + \lambda \|\mathbf{X}\|_*$$

where $\|\cdot\|_*$ denotes the nuclear norm of a matrix defined as the sum of the singular values of the matrix and $\|R_\Omega(\mathbf{X} - \mathbf{M})\|_F^2 = \sum_{i,j} \in \Omega(X_{i,j} - M_{i,j})^2$, $\mathbf{M}$ is the observed matrix for which a low rank, completed reconstruction $\mathbf{X}$ is desired and Ω is set of locations corresponding to each known value of matrix $\mathbf{M}$.

2.3 Proximal Gradient Algorithm

All the optimization problems encountered in this project (Lasso, Ridge, Nuclear Norm Minimization) are of the form $\min_{\mathbf{w}} f(\mathbf{w}) + c(\mathbf{w})$ where f is convex, ∇f is L-Lipschitz and c is convex. As

a result, we used Proximal Gradient algorithms [12] [13] which are a class of iterative, computationally efficient algorithms with state-of-the art performance. The update steps are:

- For the Lasso Regression:

$$w_j^{k+1} = \begin{cases} w_j^k + 2t^k \sum_{i=1}^n (y_i - x_i w_j^k) x_i - t^k \lambda, & \text{if} w_j^k - t^k \nabla f(w_j^k) > t^k \lambda \\ w_j^k + 2t^k \sum_{i=1}^n (y_i - x_i w_j^k) x_i + t^k \lambda, & \text{if} w_j^k - t^k \nabla f(w_j^k) < -t^k \lambda \\ 0 & \text{otherwise} \end{cases}$$

- For the Ridge Regression: $\mathbf{w}^{k+1} = \frac{1}{1+2t^k\lambda}(\mathbf{w}^k - 2t^k \sum_{i=1}^n (\mathbf{y}_i - \mathbf{x}_i^T \mathbf{w})\mathbf{x}_i)$
- Nuclear Norm Minimization: $\mathbf{X^{k+1}} = prox_{t_{k+1}\|\cdot\|}[\mathbf{X}^k - t_k R_\Omega(\mathbf{X^k} - \mathbf{M})]$ where the prox operator for the nuclear norm implies soft thresholding singular values of $\mathbf{X}^k$

2.4 Cross Validation

The regularization parameter λ in above the optimization problems controls the tradeoff between obtaining good data fits versus using fewer features to make the label predictions. A key question then is how to tune λ to obtain the right balance and Cross Validation[14] is the commonly used strategy. We used the training set with labelled examples to learn classifiers for a finite set $\Lambda = [0.001, 0.0025...25, 50, 100]$ of λs. The prediction error of these classifiers on the hold-out/training set was then used to determine the best λ from the set for each data model.

3 Analysis

3.1 Dataset and Feature Extraction

The Yelp dataset is available for download from Kaggle[15]. The entire dump data contains information about 11k busineses, 43k users and 200k reviews. Business data is characterized by several features - Business ID, Type, Review Count, Votes (useful,funny, cool) etc. Reviews and Users are similarly characterized. The test data is similar in format but has many missing entries. After parsing through the JSON files using Python scripts, we obtain a random subset of the data as our training samples and testing instances. The training set has 8070 samples and the ratio of cardinality of training set to hold out set is roughly 7:3.

The goal of our project was to predict business popularity/ average number of stars assigned to a business using features extracted from the business, review and user data. Key steps in the feature extraction process were (i) encoding non-numeric features (eg. categories, business is open/permanently closed) into numeric type and (ii) aggregating features non unique to a specific business for eg. multiple reviews exist for a specific business, we have information about the #'useful' votes assigned to each such review and we would like to estimate of the average 'usefulness' of the reviews for this specific business.

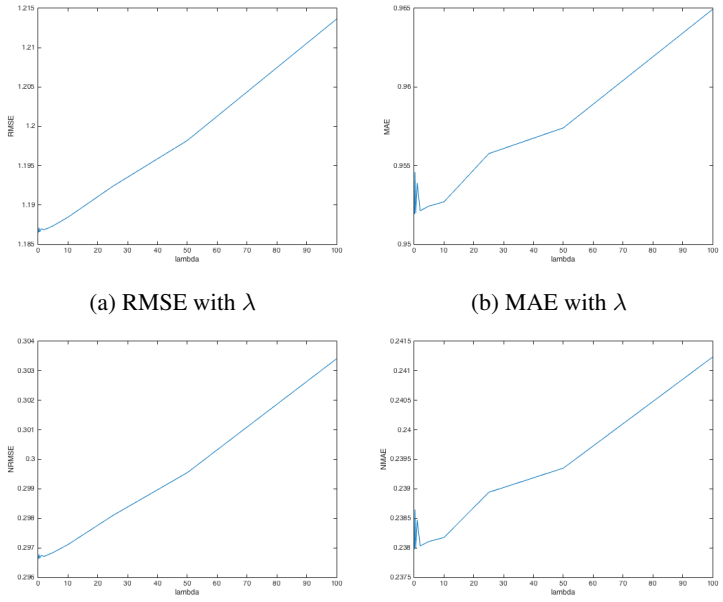
For encoding categories, we parsed the data files for 20 commonly occuring categories - art, media, food etc. and further used a 5 digit binary number to encode each.We performed aggregation by importing the parsed JSON files into PostgreSQL using multiple join and aggregation queries. Business Ids, Review Ids and UserIds were used only for inferential purposes during query evaluation (and were not used as features).

3.2 Evaluation Metrics

The data models (Lasso, Linear and Ridge Regression) were solved using MATLAB and evaluated based on the following metrics: Root Mean Squared Error (RMSE) $= \sqrt{\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}}$, Mean Absolute Error (MAE) $= \sqrt{\frac{\sum_{i=1}^n |\hat{y}_i - y_i|}{n}}$, Normalized Root Mean Squared Error (NRMSE) $= \frac{\text{RMSE}}{y_{max} - y_{min}}$, Normalized Mean Absolute Error (NMAE) $= \frac{\text{MAE}}{y_{\text{max}} - y_{\text{min}}}$, where y_{max} and y_{min} are largest and smallest values of y.

4 Results

- All the results discussed in this section have been obtained after performing matrix completion for the feature matrix of the testing data. Rank r of the test-feature matrix was 10 and we used $\lambda = 1$ in the singular value thresholding algorithm.
- Among the different values of λ tried during Cross Validation, $\lambda = 0.1$ has the smallest RMSE, MAE, NRMSE and NMAE. Figure 1 illustrates that increasing λ further results in increasing errors in all cases.



(a) RMSE with λ

(b) MAE with λ

(c) NRMSE with λ

(d) NMAE with λ

Figure 1: Different evaluating metrics with different λ

- Table 1 shows the prediction errors on the test data (after matrix completion and for $\lambda = 0.1$ after cross validation) using the different metrics for each model. We can see that, while the performances are comparable, linear regression based model performs slightly better under each evaluation metric.

Table 1: Testing error with different evalulation metrics

	Lasso Regression	Linear Regression	Ridge Regression
RMSE	1.6595	1.6549	1.6596
MAE	1.1210	1.1202	1.1210
NRMSE	0.4149	0.4137	0.4149
NMAE	0.2802	0.2800	0.2802

- Upon inspection, we find that the 'average user usefulness' feature (i.e. the average usefulness of the users who reviewed a specific business) gets the largest value of weight among all other features across all experiments. This seems consistent with the intuition that reviews deemed 'useful' should play an important role is predicting a business's polpularity.
- 'Average coolness' of the users who reviewed a business was found to be yet another important predictive feture.

References

[1] Resnick, Paul, and Varian, Hal R. Recommender systems. In *Communications of the ACM 40, no.3(1997)*: 56-58.

[2] Koren, Yehuda, Bell, Robert and Volinsky, Chris. Matrix factorization techniques for recommender systems. In *Computer 8 (2009)*: 30-37.

[3] Adomavicius, Gediminas, and Tuzhilin, Alexander. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. In *Knowledge and Data Engineering, IEEE Transactions on 17, no. 6 (2005):* 734-749.

[4] Rocchio, J.J. Relevance Feedback in Information Retrieval, SMART Retrieval System Experiments in *Automatic Document Processing, G. Salton, ed., chapter 14*, Prentice Hall, 1971.

[5] Pazzani, M. and Billsus, D. Learning and Revising User Profiles: The Identification of Interesting Web Sites. In *Machine Learning, vol. 27:* 313-331, 1997

[6] Breese, J.S., Heckerman, D. and Kadie, C. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In *Proc. 14th Conf. Uncertainty in Artificial Intelligence, July 1998*.

[7] Aggarwal, C.C., Wolf, J.L. Wu, K-L. and Yu, P.S. Horting Hatches an Egg: A New Graph-Theoretic Approach to Collaborative Filtering. In *Proc. Fifth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, Aug. 1999.*

[8] Ungar, L.H. and Foster, D.P. Clustering Methods for Collaborative Filtering. In *Proc. Recommender Systems, Papers from 1998 Workshop, Technical Report WS-98-08 1998*

[9] Y.-H. Chien and E.I. George, A Bayesian Model for Collaborative Filtering. In *Proc. Seventh Int'l Workshop Artificial Intelligence and Statistics, 1999.*

[10] Sarwar, B., Karypis, G., Konstan, J. and Riedl, J. Item-Based Collaborative Filtering Recommendation Algorithms. In *Proc. 10th Int'l WWW Conf., 2001.*

[11] Recht, Benjamin. A simpler approach to matrix completion. In *The Journal of Machine Learning Research 12 (2011):* 3413-3430.

[12] Figueiredo, Mrio AT, and Nowak, Robert D. An EM algorithm for wavelet-based image restoration. In *Image Processing, IEEE Transactions on 12, no. 8 (2003):* 906-916.

[13] Wright, Stephen J., Nowak, Robert D. and Figueiredo, Mrio AT. Sparse reconstruction by separable approximation. In *Signal Processing, IEEE Transactions on 57, no. 7 (2009):* 2479-2493.

[14] Shao, Jun. Linear model selection by cross-validation. In *Journal of the American statistical Association 88, no. 422 (1993):* 486-494.

[15] Recsys challenge 2013: Yelp business rating prediction. `https://www.kaggle.com/c/yelp-recsys-2013`