Practical 1

Team VAJ: Vincent Li, Borui Wang, Albert Young
CS 181: Machine Learning

February 15, 2014

1 Warm-Up

We ran Lloyd's algorithm for k-means clustering on CIFAR-10, which contains 50000 32×32 images. In our experiment, we ran the program using several different values of k, which produced similar mean images, although larger k tended to slow down computational speed and require more iterations before convergence. For example, when $k = 16$, the algorithm takes approximately 120 iterations to arrive at a set of stable mean images. The mean images of the 16 clusters are shown below:

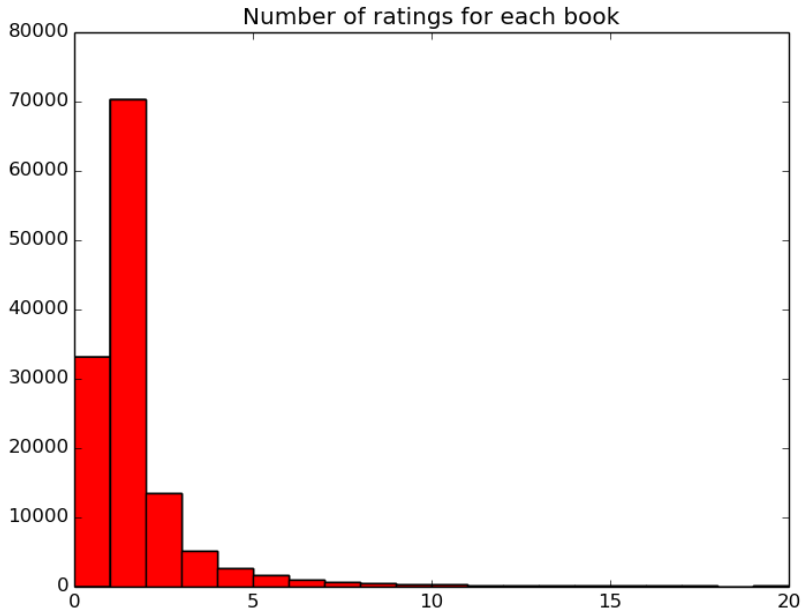
Figure 1: CIFAR-10 Cluster Centers

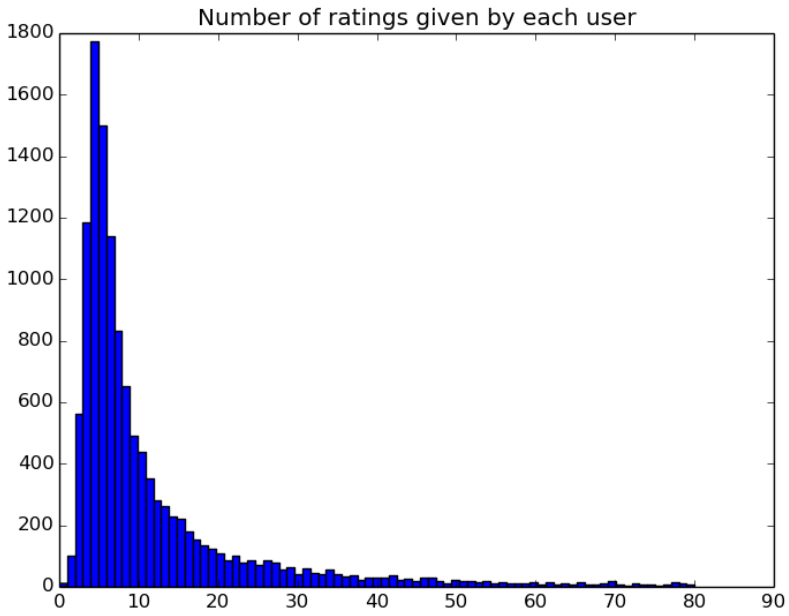
2 Predicting Book Ratings

2.1 Data Inspection

In order to develop an appropriate approach to predicting book ratings, we first sought to understand the structure of the data provided. We explored the density of the data by plotting the number of items (books) rated per user and the number of ratings each item

received:





We found the data to be extremely sparse, with more than 99.988% of all possible ratings missing; both plots follow a heavy-tailed distribution. Each book received a median of only two ratings, and each user rated a median of 7 books.

We then performed a visual inspection for a relationship between average rating and user age (Fig. 2) and between average rating and book publication year (Fig. 3). While no clear relationship existed between a user's average rating and his age, it appeared that books

Figure 2: Average rating versus user age

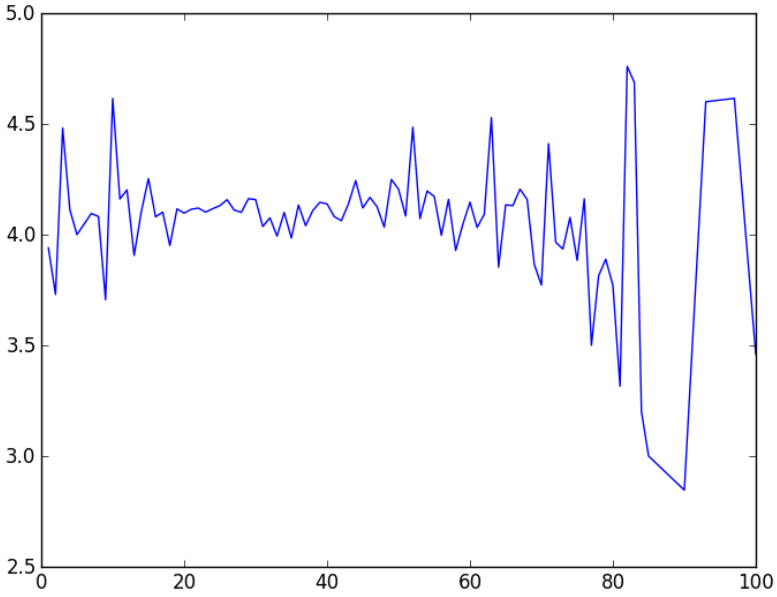
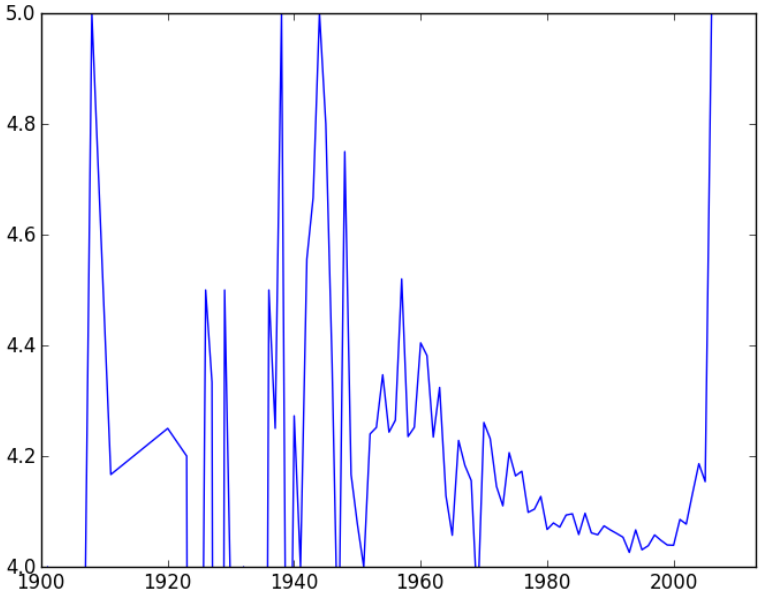


Figure 3: Average rating versus book publication year



published earlier were rated more highly on average than books published later between roughly 1960 and 2000.

2.2 Baseline Estimates

Before capturing variation due to interactions between users and items, we derived a robust baseline estimate for each prediction. We accounted for observed variations in ratings due to biases associated with both users and items. For example, a critical user may systematically rate movies below the global mean or a popular item may consistently receive ratings above the global mean (i.e. the mean rating of all books in the training set). We estimate the baseline rating of each prediction by following Koren's approach [1]:

$$b_{ui} = \mu + b_i + b_u \tag{1}$$

The baseline rating b_{ui} of item i by user u is calculated by adjusting the global mean μ by the item- and user-associated biases, b_i and b_u, respectively.

For each item i we calculate the item bias:

$$b_i = \frac{\sum_{u:(u,i)\in\kappa}(r_{ui} - \mu)}{\lambda_1 + ||u|(u,i) \in \kappa||}. \tag{2}$$

Likewise, for each user u we calculate the user bias:

$$b_u = \frac{\sum_{i:(u,i)\in\kappa}(r_{ui} - \mu - b_i)}{\lambda_2 + ||i|(u,i) \in \kappa||}. \tag{3}$$

To prevent overfitting, we shrank our estimates using the regularization parameters, λ_1 and λ_2, which were optimized to 14.2 and 2.8, respectively, by cross-validation.

Additionally, we tried including a b_y term in our model to account for the bias due to publication year, but this failed to improve our model. We hypothesize that this may be due to the bias due to publication year having already been captured by the item bias or some other confounder (e.g. perhaps users tended to only rate books published during a certain time period).

Finally, we tried to improve our baseline estimate by incorporating data from Amazon reviews (downloaded from http://snap.stanford.edu/data/web-Amazon.html), which included 12,886,488 reviews for 929,264 books. Calculating the item bias with the addition of item bias information from the Amazon data did not significantly improve baseline estimates, and we did not have time to explore the approach fully.

2.3 k-Nearest Neighbors

We first implemented a neighborhood-based approach to calculate a prediction $\hat{r}_{ui}$ due to its simplicity. The algorithm identifies the set of k most similar items to item i, $S^k(i;u)$, also rated by user u, and adjusts the baseline estimate b_{ui} by a weighted sum of the ratings of the k items. The intuition is that a user would be more likely to rate a item comparably with more similar items than to less similar items. To determine the similarity between two

items i and j, the adjusted cosine similarity $\rho(i,j)$ is first calculated, as described by Sarwar et al. [2]:

$$\rho(i,j) = \frac{\sum_{u \in U}(r_{ui} - r_{\bar{u}})(r_{uj} - r_{\bar{u}})}{\sqrt{(\sum_{u \in U}(r_{ui} - r_{\bar{u}})^2)}\sqrt{(\sum_{u \in U}(r_{uj} - r_{\bar{u}})^2)}} \tag{4}$$

where U is the set of all users that co-rated items i and j and $r_{\bar{u}}$ is the user-adjusted baseline of the active user. The similarity measure s_{ij} was then computed:

$$s_{ij} = \frac{n_{ij}}{n_{ij} + \lambda_3}\rho(i,j) \tag{5}$$

where n_{ij} is the number of users that rated both item i and item j and λ_3 is a constant added to reduce the weight of similarities calculated based on fewer ratings.

We then predict $\hat{r}_{ui}$ by taking a weighted average of the ratings of neighboring items, while adjusting for user and item biases:

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{j \in S^k(i;u)} s_{ij}(r_{uj} - b_{uj})}{\sum_{j \in S^k(i;u)} s_{ij}} \tag{6}$$

We also implemented an analogous user-based k-nearest neighbors approach to calculate a prediction $\hat{r}_{ui}$ by identifying the set of k most similar users that also rated item i, and adjusting the baseline estimate b_{ui} by a weighted sum of the ratings of the k users. Adjusted cosine similarity was again used as the similarity measure between two users, with a new constant added to account for age-related differences between users. The intuition is that for a given item, similar users would give more comparable ratings than less similar users.

Both the item-based and user-based neighborhood approaches failed to improve upon our baseline performance, after testing multiple parameter values, thresholds for considering similarity, and neighborhood sizes. We hypothesize that this failure is due to the inaccuracy of similarity assessments between items and between users, as there are few pairs of items that are co-rated by a sufficient number of users. Given more time, we would try to improve our similarity calculations with book metadata such as genre, content, and author.

2.4 Matrix Factorization

Since neighborhood-based methods were limited by the sparsity of the data limitations of looking at users and items in isolation, we aimed to capture the effect of user-item interactions using matrix factorization. The idea is that item features and user characteristics can described with factor vectors, q_i and p_u. The user's estimated interest in the item is captured by $q_i^\intercal p_u$. The prediction $\hat{r}_{ui}$, including bias parameters, is then determined by:

$$\hat{r}_{ui} = b_i + b_u + q_i^\intercal p_u \tag{7}$$

In order to compute a mapping of items and users to factor vectors, we solved the following least squares problem, which includes a regularization constant λ that penalizes complexity to ovoid overfitting.

$$\min_{q_\cdot, p_\cdot} \sum_{(u,i) \in \kappa} (r_{ui} - q_i^\intercal p_u)^2 + \lambda(||q_i||^2 + ||p_u||^2) \tag{8}$$

We implemented the stochastic gradient descent optimization described by Simon Funk (http://sifter.org/ simon/journal/20061211.html), which modifies q_i and p_u relative to prediction error:

$$e_{ui} = r_{ui} - q_i^\intercal p_u \tag{9}$$

$$q_i \leftarrow q_i + \gamma \cdot (e_{ui} \cdot p_u - \lambda \cdot q_i) \tag{10}$$

$$p_u \leftarrow p_u + \gamma \cdot (e_{ui} \cdot q_i - \lambda \cdot p_u) \tag{11}$$

With each loop, the q_i and p_u and modified by magnitude proportion to γ in the opposite direction of the gradient. As per iterative algorithms, the step size and number of epochs were first tuned to convergence. Then the regularization coefficent was chosen with cross-validation.

References

[1] Koren, Y. 2010. Factor in the neighbors: Scalable and accurate collaborative ltering. ACM Trans. Knowl. Discov. Data. 4, 1, Article 1, 2010.

[2] Sarwar, B., G. Karypis, J. Konstan, and J. Riedl. Item-based Collaborative Filtering Recommendation Algorithms. In Proc. of the 10th International WWW Conference, 2001.

[3] C. Volinsky et al.: Matrix Factorization Techniques for Recommender Systems" In: IEEE Computer, Vol. 42 (2009) , pp. 30-37 .