CS 109A/Stat 121A/AC 209A: Final Project
Protopapas, Rader, Dave, Levine

Recommendation Systems

Problem statement

You will create a recommendation system for **restaurants** using regression. We will use Regression, and Matrix Factorization, and something else of your choice, and combine them using an ensemble method. At each point you will see if your mode's performance improves.

The recommenders will not be particularly good or sophisticated, but you should carry out the entire process and on the way, you'll learn a lot about the gotchas and subtleties involved.

Data resources

You will be using the academic dataset (https://www.yelp.com/dataset/challenge) from yelp, and can filter it down to the restaurants, for concreteness.

High Level Goals

Create a Baseline

The first part of any recommender system is to create baseline estimate of the ratings. Indeed, most of any rating can be usually explained by baseline estimates, and all other methods are an attempt (only sometimes successful) to improve these baselines.

We shall create two baselines here: one based on estimating biases with sample averages, and the second one using regularized regression.

We can write the baseline estimate $\hat{Y}_{um}^{baseline}$ for an unknown rating for user u and restaurant m as:

$$\widehat{Y_{um}} = \hat{\mu} + \hat{\theta}_u + \widehat{\gamma_m}$$

where the unknown parameters θ_u and γ_m indicate the deviations, or biases, of user u and item m respectively from some intercept parameter μ.

Notice that θ_u and γ_m are parameters which need to be fit. The simplest thing to start with is to replace them by their "mean" estimates from the data.

Create a Regularized Regression

Now we can do the actual fit as a regression against indicators by recasting

$$\hat{Y}_{um} = \hat{\mu} + \hat{\theta}_u + \hat{\gamma}_m$$

into the following form

$$Y^{baseline}_{um} = \mu + \bar{\theta} \cdot I_u + \bar{\gamma} \cdot I_m$$

where I_u and I_m are indicator variables for the u-th user and m-th item that go into the feature matrix. Here $\bar{\theta}$ is the vector of all the coefficients for users who have made ratings in our training set, and $\bar{\gamma}$ is a vector of coefficients for all the items for which ratings have been made in our training set.

Matrix Factorization

We can think of latent factors as properties of restaurants (e.g., spiciness of food or price) that users have a positive or negative preference for. We do not observe these factors directly, but we assume that they affect how users tend to rate restaurants. One issue that comes up with latent factor models is determining how many latent factors to include. The problem looks like the factorization of a matrix (see Reference 2).
How do we do this? We write the residuals from the baseline thus:

$$r_{um} = Y_{um} - Y^{baseline}_{um} = \bar{q}^T_m \cdot \bar{p}_u.$$

The corresponding regression problem loss function then looks like this:

$$\sum_{u,m} \left(Y_{um} - \mu - \bar{\theta} \cdot I_u - \bar{\gamma} \cdot I_m - \bar{q}^T_m \cdot \bar{p}_u \right)^2 + \alpha \left(\theta^2_u + \gamma^2_m + ||\bar{q}_m||^2 + ||\bar{p}_u||^2 \right)$$

where we regularize the factors as well.

This can be solved by alternating lasso or ridge regressions called **Alternating Least Squares**. You can read more about how to do this in reference 2 below.

References

1. *How the Netflix prize was won*, http://blog.echen.me/2011/10/24/winning-the-netflix-prize-a-summa

2. *Matrix factorization for recommender systems*, https://datajobs.com/data-science-repo/Recommender-Systems-[Netflix].pdf

3. *Ensembling for the Netflix Prize*, http://web.stanford.edu/~lmackey/papers/netflix_story-nas11-slid pdf

4. *Reviews on methods for netflix prize*, http://arxiv.org/abs/1202.1112andhttp://www.grouplens.org/system/files/FnT%20CF%20Recsys%20Survey.pdf

5. *Advances in Collaborative Filtering from the Netflix prize*, https://datajobs.com/data-science-repo/Collaborative-Filtering-%5BKoren-and-Bell%5D.pdf