Yelp++ : 10 Times More Information per View

Sean Choi, Ernest Ryu, Yuekai Sun

December 16, 2011

Abstract

In this project we investigate two machine learning methods, one supervised and one unsupervised, that will allow the information content of Yelp data to be efficiently conveyed to the users. The first is matrix completion via the novel "max-norm" constraint which out results show to be more powerful than the traditional nuclear norm minimization. The second is text summary via sparse PCA which can provide a concise summary of the available immense text reviews. We implement and run these algorithms on actual Yelp data and provide results.

1 Introduction

Modern consumers are inundated with information and choices. Countless services now provide a daunting amount of information to a casual consumer. This plethora of information brings the need for a recommender system provide individually customized lists of relevant products and a concise summary of what the product is.

In this project we investigate two methods to improve Yelp's service: a better recommendation system via matrix completion with the "max-norm" and a text summary system via sparse-PCA.

2 Theory

2.1 Collaborative filtering with uniformly bounded data

Collaborative filtering (CF) has been popularized in the past few years by the Netflix challenge. The mathematical statement of the problem is as follows: can one reconstruct a matrix when only a subset of the entires have been observed?

One popular method is nuclear norm minimization supported by the theory of compressed sensing. [4] In this project, we take a different approach and use what's called the "max-norm."

2.1.1 Data

Our data matrix $X \in \mathbb{R}^{n\times m}$ is the restaurant star ratings matrix, i.e. X_{ij} is the i-th user's raring of the j-th restaurant. It is incomplete so we only know a subset of the entries with indices $(i,j) \in \Omega \subset \{1\cdots n\} \times \{1\cdots m\}$. We normalize the rating by subtracting 3 to every entry so that $X_{ij} \in \{-2,-1,0,1,2\}$. This makes X uniformly bounded in the sense that $|X_{ij}| \leq 2$ for all i,j.

2.1.2 Matrix completion using max-norm regularization

The max-norm of X is defined as

$$\|X\|_{max} = \inf_{X=UV^T} \left(\max_{i=1\ldots m} \|u_i\|_2 \max_{i=1\ldots m} \|v_i\|_2 \right) \quad (1)$$

and u_i, v_i denote the i-th row of U and V respectively.

We consider the max-norm constrained version of matrix completion.

$$\min_{U\in\mathbb{R}^{n\times k}, V\in\mathbb{R}^{m\times k}} \sum_{(i,j)\in\Omega} (X_{ij} - (UV^T)_{ij})^2$$
$$\text{subject to: } \|UV^T\|_{max} \leq \tau$$

where k is the rank of the prediction matrix which can be interpreted as the number of latent variables. Our heuristic to solve this non-convex constrained optimization problem is alternating minimization over U and V. Fixing V, we obtain the following n embarrassingly parallel convex optimization subproblems.

$$\min_{u_i\in\mathbb{R}^n} \|X_{i\cdot}^T - V_j u_i\|_2^2$$
$$\text{subject to: } \|u_i\|_2 \leq \frac{\tau}{\max_{1\leq j\leq m} \|v_j\|_2}$$

which is a $k \times k$ QP and in particular is the well-studied "trust region subproblem" [7] which can be

solved very efficiently for small k by explicitly computing the eigenvalue decomposition.[1] A similar result holds for fixing U and we obtain the following matrix completion algorithm. We again emphasize

Algorithm 1 Max-norm Matrix Completion

```
while not converged do
  for i = 1 ··· n do
    min_{u_i} ‖X_{i·} − u_i V^T‖^2
      subject to: ‖u_i‖_2 ≤ τ / max_{1≤j≤m} ‖v_j‖_2
  end for
  for j = 1 ··· n do
    min_{v_j} ‖X_{·j} − U v_j‖^2
      subject to: ‖v_j‖_2 ≤ τ / max_{1≤i≤n} ‖u_i‖_2
  end for
end while
```

that the for loop operation of algorithm 1 is embarrassingly parallel.

2.1.3 Bias correction

Nothing in algorithm 1 constrains the resulting prediction matrix $Y = UV^Y$ to have the same mean as the data matrix X. In fact the discrepancy of the means of Y and X is not negligible by correcting this "bias" by letting

$$Y = Y - (\mu_Y - \mu_X)$$

where

$$\mu_Y = \frac{1}{nm}\sum_{ij} Y_{ij} \qquad \mu_X = \frac{1}{|\Omega|}\sum_{(i,j)\in\Omega} X_{ij}$$

provides an improved RMSE empirically for us and others. [11]

2.1.4 Interpretation of max-norm regularization

There are two interpretation that justifies the use of max-norm regularization. The first is that the max-norm regularization implies that the entries of Y are uniformly bounded by τ. By the Cauchy-Schwartz inequality,

$$\|Y\|_{\max} \le \tau \Rightarrow |Y_{ij}| = |\tilde{u}_i \tilde{v}_j^T| \le \|\tilde{u}_i\|_2 \|\tilde{v}_j\|_2 \le \tau$$

For the application in consideration, valid predictions are integers between -2 and 2 so a sensible choice of τ is 2. In particular the choice $\tau = 2$ will ensure that all our predictions will lie in the interval $[-2, 2]$.

The second iterpretation is the maximum margin classifier. [10] Consider the max-norm variant of the matrix completion problem

$$\min_Y \|Y\|_{max}$$
$$\text{subject to: } X_{ij}Y_{ij} \ge 1,\ (i,j) \in \Omega$$

Y can be decomposed into $Y = UV^T$ hence every entry of Y can be expressed $Y_{ij} = u_i^T v_j$ where u_i and v_j are the rows of U and V respectively. We can interpret u_i as a feature vector for user i and v_j as a classifier that classifies users into users that like and dislike movie j. The features are the affinity of user i to the k-th latent variable. The constraint $X_{ij}Y_{ij} \ge 1$ ensures the classifier correctly classifies users who have rated movie j.
On the other hand the well-known SVM optimization problem is

$$\min_{w,b} \frac{1}{2}\|w\|_2$$
$$\text{subject to: } y_i(w^T x_i + b) \ge 1,\ i = 1, \cdots, m$$

Since $\|X\|_{max}$ is defined as (1), each subproblem is essentially seeking for a maximum margin classifier

$$\min_w \frac{1}{2}\|v_j\|_2$$
$$\text{subject to: } X_{ij}(u_i^T v_j + 0) \ge 1,\ (i,j) \in \Omega$$

This interpretation suggests that the max-norm regularization is a generalization of the approach to find the maximum margin classifier for a dataset that only contains a boolean affinity of the users for each movie.

2.2 Using SPCA to summarize large text corpora

Yelp has a large database of text data that would make sense to a human reader. However many popular restaurants have more than 500 revies and it is unreasonable to expect users to make a judgement about the restaurant after digesting this large amount of data. In a sense, Yelp is presenting these reviews to users very inefficiently. Our goal is to implement an algorithm that would provide a concise summary that would represent the key features of the restaurant.

[1] The QP matrix is symmetric so for small k the eigenvalue decomposition does not pose a significant numerical challenge. However, the complexity grows at a rate of $\mathcal{O}(k^3)$.

2.2.1 Data

Our data matrix X consists of tf-idf scores. Term frequency inverse document frequency (tf-idf) score, commonly used in natural language processing, is defined as

$$\text{tf-idf}(t,d) = \frac{\#\text{terms} \in d}{\#\text{words} \in d} \log \frac{|D|}{|\{d : t \in d\}|}$$

where t is a specific term and d is a particular documnet. Each column of X corresponds to a document and each row to a term.

2.2.2 Sparse Principal Component Analysis (SPCA)

Principal conponent analysis (PCA) is a widely used tool and is applicable to this application of text summary. However, the principal components generated by PCA are hard to interpret as they are not sparse. El Ghaoui et al. [5] suggested that sparse PCA (SPCA) can be effectively used to extract concise pricipal components which can be presented into a concise list of words.
The mathematical statement of SPCA is [8]

$$\min_{U,V} \frac{1}{2}\|X - UV^T\|_{Fro}^2 + \lambda_u\|U\|_1 + \lambda_v\|V\|_1$$

where $\|\cdot\|_{Fro}$ is the standard Frobenius norm and $\|\cdot\|_1$ is the element-wise ℓ^1 norm, i.e.

$$\|U\|_1 = \sum_{i=1}^{n}\sum_{j=1}^{k}|U_{ij}|$$

λ_u, λ_v are parameters that control the sparsity. This approach is consistent with the standard method of imposing an ℓ^1 penalization term for sparsity.
This optimization problem, however, is non-convex and non-differentiable. The standard heuristic that will find a local optimum is block coordinate descent. To be precise, the i-th principal component and its corresponding loadings are found sequentially with the i-th residual matrix X_i by iteratively minimizing over u and v. The sub-problem becomes

$$\min_{u} \frac{1}{2}\|X - uv^T\|_{Fro}^2 + \lambda_u\|U\|_1$$

and with the use of sub-gradients one can find the analytical solution

$$u = S_t\left(\frac{1}{\|v\|_2^2}X_i v_i, \frac{1}{\|v\|_2^2}\lambda_u\right)$$

where S_t is the soft-thresholding function

$$S_t(x,\eta) = \begin{cases} \text{sgn}(x)(|x| - \eta) & \text{if } |x| \geq \eta \\ 0 & \text{otherwise} \end{cases}$$

This yields the following SPCA algorithm where the application of S_t is embarrassingly parallel.

Algorithm 2 SPCA for Text Summary

```
X_1 = X
for i = 1 ··· k do
  while not converged do
    u_i = S_t(X_i v_i/‖v_i‖_2^2, λ_u/‖v_i‖_2^2)
    v_i = S_t(X_i' u_i/‖u_i‖_2^2, λ_v/‖u_i‖_2^2)
  end while
  X_{i+1} = X_i − u_i v_i^T
end for
```

2.2.3 Varying the sparsity parameter λ_u, λ_v

Although this aspect was concealed in the previous algorithm discussion for the sake of conciseness, it is important to impose a different spasity parameter for each principal components.
When the sparsity parameters λ_u, λ_v are fixed throughout the iterations, eventually the residual matrix $X_{i+1} = X_i - u_i v_i^T$ becomes small[2] and the resulting PC and loadings become identically 0. If λ_u, λ_v are chosen to be small enough to avoid this phenomenon then the initial PC's will be too dense. A solution to this problem is to decrease the value of λ_u, λ_v throughout the iterations. In particular, we let $\lambda_u = \eta_u\lambda_u$ and $\lambda_v = \eta_v\lambda_v$ when the resulting PC became identically zero. Empirically $\eta = 1/3$ worked well.

3 Methods

3.1 Data Processing

To test our algorithms we have collected 6.5 million restaurant reviews for about 400000 restaurants and about 1 million users, ranging across the entire continental USA. We specifically collected four features of each reviews: restaurant name, user name, user ratings and user text reviews. The raw data was first inserted into a database and auto increment primary key of the MySQL database [1] was used to generate

[2] An example of a quantitative measure of this could be the Frobenius norm.

restaurant ids and user ids . The final data was compressed into restaurant id, user id, user rating and user ratings. and exported into a XML format.
To convert the XML format into a format that resembles a sparse matrix representation we utilized the Amazon EC2 cluster and Hadoop.

Extracting the restaurant ids, user ids and user ratings was straightforward. As for processing the text, we decided to use stemming and to remove stop words as it is arguably practice in text data mining [9] and as it empirically gave better results. For stemming we utilized the Apache Lucene [2] library and for stop word removal the Porter Stemmer [3] algorithm. After this pre-processing, the text data was formed into a tf-idf score vector as mentioned before.

3.2 Sparse Linear Algebra

Because of the data size, traditional dense linear algebra becomes infeasible and therefore we utilized Matlab's sparse matrix functionality for our algorithms. However, there were some difficulties, especially in the SPCA algorithm, where we would encounter $X - uv^T$ where X is sparse and u, v are vectors. When this expression is evaluated the resulting matrix loses its sparsity entirely. Our solution was to utilize Stanford iCME's shared memory machine wich offers 128GB of RAM.
For future work, one could consider a sparse linear algebra implementation that can retain the above expression (which is very sparse) without explicitly evaluating the expression. We did not take this path due to time constraints.

4 Results

4.1 Max-norm Matrix Completion

The rank parameter k was set to $k = 32$. As mentioned before there is a significant cost in increasing the value of k but emprically there was diminishing returns in increasing k beyond 32 and in particular $k = 64$ did not yield a significant improvement.
The standard assessment of a matrix completion algorithm is the root-mean-square error (RMSE) of cross-validation. To this end, we held out 25% of the data for cross validation.
The max-norm (MN) and max-norm with bias correction (MNBC) algorithms are benchmarked against the average rating (AR), nuclear norm minimization (Nuc.N), and a result from a candidate in the Netflix competition, [11] which of course is done on an entirely different dataset.

We can see that the max-norm matrix completion

	AR	Nuc.N	MN	MNBC	Netflix
RMSE	1.26	1.11	1.08	1.07	0.91

Table 1: Comparison of matrix completion results

outperforms the other two standard algorithms. We have included the Netflix result as merely a reference to suggest that the performance of MN is not outrageously sub-optimal. The RMSE value of 0.91 is not comparable to our result as the Netflix dataset has quite different statistical properties compared to out Yelp dataset and the difference in performance should not be understood as a defeat.

4.2 SPCA text summary

Many of the sparse principal components from the SPCA algorithm had a clear interpretation. Restaurants with loadings of such PC's usually belonged to the categories consistent with the PC's interpretation.

Table 2 shows some example principal components

3rd PC	5th PC	11th PC	12th PC	13th PC
italian	taco	naan	dimsum	falafel
pepper	burrito	indian	dumplg	gyro
crust	maxica	buffet	chinese	pita
italian	salsa	masala	noodle	sandwi

Table 2: Example principal components with their most significant entries.

from the SPCA algorithm with the 4 most significant (largest value) entries. The meaning of these PC's are unambiguous. Table 3 shows some example loadings

Rest. Name	pc3	pc5	pc11	pc12	pc13
El Gran Amigo Tagueria	.74	-.01	-.04	.17	-.04
La Mediterranee	-.05	-.17	.25	.06	.65
Beijing Restaurant	-.06	-.14	.45	.14	-.12
Tommaso Ristorante Italiano	-.04	.16	-0.06	.95	.08

Table 3: Example loadings of the 5 presented principal components.

from the SPCA algorithm. The loadings provide a clear indication on what type of restaurant each one is.[3]

5 Conclusion

We investigated 2 learning algorithms that are applicable to recommender systems and we tested the feasibility on a specific real-world dataset. Although this report did not detail on the complexity analysis of the optimization algorithms and sparse numerical linear algebra, the algorithms are quite efficient and scalable as they reduce to embarrassingly parallel subproblems.

Matrix completion using max-norm regularization showed promise as it was able to outperform the standard nuclear norm minimization algorithm. Ultimately, it is unlikely that the max-norm matrix completion algorithm alone will fare well against its competitors as the best matrix completion algorithms utilize "blending," or mixing of different such algorithms. [6] However, the max-norm matrix completion algorithm could contribute to this blending as a powerful ingredient.

Text summary via SPCA also showed considerable promise as it was successful in detecting the most important features of restaurants. However, it did lack the ability to extract finer features such as whether the service of the restaurant is good or whether the restaurant is crowded. In that regard, there is room for future work with text summary via SPCA. In a practical standpoint, automatic text summary of Yelp reviews are not very powerful if it cannot extract any information beyond the genre of the restaurant.

References

[1] http://dev.mysql.com/doc/refman/5.0/en/example-auto-increment.html.

[2] http://lucene.apache.org/java/docs/index.html.

[3] http://snowball.tartarus.org/algorithms/english/stemmer.html.

[4] Emmanuel J. Candés and Benjamin Recht. Exact matrix completion via convex optimization. *Found. Comput. Math.*, 9:717–772, December 2009.

[5] L. El Ghaoui, G.-C. Li, V.-A. Duong, V. Pham, A. Srivastava, and K. Bhaduri. Sparse machine learning methods for understanding large text corpora. In *Proc. Conference on Intelligent Data Understanding*, October 2011. Accepted for publication, July 2011.

[6] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 426–434, New York, NY, USA, 2008. ACM.

[7] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, August 2000.

[8] H Shen and J Huang. Sparse principal component analysis via regularized low rank matrix approximation. *Journal of Multivariate Analysis*, 99(6):1015–1034, 2008.

[9] Jeffrey L. Solka. Text data mining: Theory and methods. *Statistics Surveys*, 2:94–112, 2008.

[10] Nathan Srebro. *Learning with matrix factorizations*. PhD thesis, MIT, Cambridge, MA, USA, 2004. AAI0807530.

[11] Yunhong Zhou, Dennis Wilkinson, Robert Schreiber, and Rong Pan. Large-scale parallel collaborative filtering for the netflix prize. *Algorithmic Aspects in Information and Management*, 5034:337?348, 2008.

[3] The restaurant names (which are tremendously descriptive for the above 4 examples) were of course not used in the SPCA algorithm.