Recommender Systems: Notes

Aaron Tuor

March 26, 2015

Contents

1 Aggregating preferences, criteria, or similarities 2

2 Application Domains 2

3 Representing Users and Items 2

4 Similarity Metrics 2
  4.1 Manhattan Distance 2
  4.2 Euclidean Distance 3
  4.3 $L_r$ norm (Minkowski distance) 3
  4.4 Mahalanobis distance 3
  4.5 Jaccard Coefficient 3
  4.6 Tanimoto Coefficient (Extended Jaccard Coefficient) 4
  4.7 Log-likelihood 4
  4.8 Cosine Distance 4
  4.9 Pearson-correlation coefficient (PCC) 4
  4.10 Spearman rank correlation 5

5 Content based recommendations 6
  5.1 Term Frequency-Inverse Document Frequency (TF-IDF) 6

6 Collaborative Filtering: Koren Notation 7

7 Collaborative Filtering: Baseline Estimates 7

8 Collaborative Filtering I: Neighborhood Models 7
  8.1 User-based Neighborhood Models 7
  8.2 Item-based Neighborhood Models 7

9 Collaborative Filtering II: Matrix Factorization 7
9.1 User-Item Utility Matrix .............................................. 8
9.2 SVD ........................................................................... 8
9.3 Latent Factors ............................................................. 9
9.4 Deriving matrix form of gradients ................................. 10
9.5 Gradient Descent ......................................................... 11
9.6 Alternating Least Squares ............................................. 11
9.7 Tikhonov Regularization .............................................. 11
9.8 Initialization of $P$ and $Q$ ........................................... 11
  9.8.1 als ....................................................................... 11
  9.8.2 Funk SVD ............................................................. 11
  9.8.3 SVD++ ................................................................. 11
9.9 Tikhonov Regularization .............................................. 11
9.10 Incorporating Global User and Item Patterns............... 11
9.11 Funk SVD ................................................................. 11
9.12 Assymmetric SVD ...................................................... 11
9.13 Incorporating Additional Factors ................................. 11
9.14 SVD++ ................................................................. 11
9.15 Time Factors ............................................................ 11

10 Hybrid Systems .......................................................... 11
  10.1 Matrix Factorization + Neighborhood ........................ 11

11 Evaluation ................................................................. 11
  11.1 RMSE ....................................................................... 12
  11.2 Absolute mean error .................................................. 12
  11.3 accuracy .................................................................. 12
  11.4 precision .................................................................. 12
  11.5 recall ........................................................................ 12
  11.6 novelty ...................................................................... 12
  11.7 diversity .................................................................... 12

12 References ................................................................. 13

13 Lectures ........................................................................ 15

14 Correspondence Winter 2015 ........................................ 15
  1/8/15 BH-AT ................................................................. 15
  1/15/15 BH-AT ............................................................... 16
  1/17/15 BH-AT ............................................................... 16
  1/20/15 BH-AT ............................................................... 16
  1/22/15 AT-BH ............................................................... 18
  1/23/15 BH-AT ............................................................... 18
  1/23/15 AT-BH ............................................................... 19
1/27/15 AT-AS ................................................................. 21
1/27/15 AS-AT ................................................................. 22
1/27/15 BH-AT ................................................................. 22
1/31/15 AT-BH ................................................................. 23
2/3/15 BH-AT ................................................................. 23
2/13/15 BH-AT ................................................................. 24
2/13/15 BH-HR ................................................................. 24
3/19/15 AT-BH ................................................................. 25
3/26/15 AT-BH ................................................................. 25
3/26/15 BH-HR ................................................................. 25
3/28/15 BH-TS ................................................................. 26
3/31/15 BH-AT ................................................................. 26
4/3/2015 BH-AT ................................................................. 27
4/7/2015 BH-AT ................................................................. 27
4/7/2015 AT-BH ................................................................. 27
4/8/2015 AT-BH ................................................................. 28
4/10/2015 BH-AT ................................................................. 28
4/11/2015 AT-BH ................................................................. 29
4/12/2015 AT-BH ................................................................. 29
4/17/2015 BH-AT ................................................................. 30
4/17/2015 AT-BH ................................................................. 30
4/21/2015 BH-AT,KM .............................................................. 30
4/28/2015 BH-AT,KM .............................................................. 31
5/1/2015 BH-AT ................................................................. 31
5/2/2015 AT-BH ................................................................. 32
5/3/2015 BH-AT ................................................................. 32
5/3/2015 AT-BH ................................................................. 32
5/5/2015 AT-BH ................................................................. 33
5/5/2015 BH-AT,KM .............................................................. 33
5/11/2015 BH-AT ................................................................. 34
5/12/2015 AT-BH ................................................................. 34
5/12/2015 AT-BH ................................................................. 34
5/12/2015 BH-AT ................................................................. 34
5/13/2015 AT-BH ................................................................. 35
5/14/2015 AT-BH ................................................................. 35
5/15/2015 BH-AT,KM .............................................................. 36
5/19/2015 BH-AT,KM .............................................................. 36
5/22/2015 BH-AT,KM .............................................................. 37
5/23/2015 BH-AT ................................................................. 37
5/26/2015 BH-AT,KM .............................................................. 37
5/29/2015 BH-AT,KM .............................................................. 37

15 Correspondence Spring 2015

4/3/2015 BH-AT ................................................................. 27
4/7/2015 BH-AT ................................................................. 27
4/7/2015 AT-BH ................................................................. 27
4/8/2015 AT-BH ................................................................. 28
4/10/2015 BH-AT ................................................................. 28
4/11/2015 AT-BH ................................................................. 29
4/12/2015 AT-BH ................................................................. 29
4/17/2015 BH-AT ................................................................. 30
4/17/2015 AT-BH ................................................................. 30
4/21/2015 BH-AT,KM .............................................................. 30
4/28/2015 BH-AT,KM .............................................................. 31
5/1/2015 BH-AT ................................................................. 31
5/2/2015 AT-BH ................................................................. 32
5/3/2015 BH-AT ................................................................. 32
5/3/2015 AT-BH ................................................................. 32
5/5/2015 AT-BH ................................................................. 33
5/5/2015 BH-AT,KM .............................................................. 33
5/11/2015 BH-AT ................................................................. 34
5/12/2015 AT-BH ................................................................. 34
5/12/2015 AT-BH ................................................................. 34
5/12/2015 BH-AT ................................................................. 34
5/13/2015 AT-BH ................................................................. 35
5/14/2015 AT-BH ................................................................. 35
5/15/2015 BH-AT,KM .............................................................. 36
5/19/2015 BH-AT,KM .............................................................. 36
5/22/2015 BH-AT,KM .............................................................. 37
5/23/2015 BH-AT ................................................................. 37
5/26/2015 BH-AT,KM .............................................................. 37
5/29/2015 BH-AT,KM .............................................................. 37
1 Aggregating preferences, criteria, or similarities

Normally such aggregation is done by using either the arithmetic mean or max/min functions. Alternatives: generalized means, Choquet and Sugeno integrals, ordered weighted averaging, triangular norms, and conorms, bipolar aggregation functions.

2 Application Domains

Entertainment- recommendations for movies, music, and IPTV
Content - personalized newspapers, recommendation for documents, recommendations of Web pages, e-learning applications, and e-mail filters.
E-commerce - recommendations for consumers of products to buy such as books, cameras, PCs etc.
Services - recommendations of travel services, recommendation of experts for consultation, recommendation of houses to rent, or matchmaking services.

3 Representing Users and Items

In a content based recommendation system items are represented as vectors of features. Users can be represented as vectors of preferences for items

4 Similarity Metrics

Many recommendation algorithms employ some form of similarity metric in the generation of ratings predictions. Similarity metrics are often associated with some form of distance measure.

Definition 4.0.1. Let \( \delta \) be a function \( \delta : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R} \). Let \( x, y, z \in \mathbb{R}^n \). Then \( \delta \) is a distance measure if it satisfies the following four properties.

(d1) \( \delta(x, y) \geq 0 \) (no negative distances).
(d2) \( \delta(x, y) = 0 \) if and only if \( x = y \) (different vectors can’t be in the same position).
(d3) \( \delta(x, y) = \delta(y, x) \) (distance is symmetric).
(d4) \( \delta(x, y) \leq \delta(z, x) + \delta(z, y) \) (triangle inequality).

4.1 Manhattan Distance

The Manhattan Distance, \( \delta_M \), between two vectors \( x, y \in \mathbb{R}^n \) is the sum of the magnitudes of the differences in each dimension, i.e.,

\[
\delta_M(x, y) = \sum_{i=1}^{n} |x_i - y_i|.
\]
4.2 Euclidean Distance

The most commonly found distance measure for real valued vectors is the Euclidean distance. For two vectors \( x, y \in \mathbb{R}^n \), the Euclidean distance, \( \delta_E \), between \( x \) and \( y \) is defined as:

\[
\delta_E(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}
\]

4.3 \( L_r \) norm (Minkowski distance)

Euclidean distance, and Manhattan distance are examples of \( L_r \) norms. For two vectors \( x, y \in \mathbb{R}^n \), the \( L_r \) norm, \( \delta_r \), between \( x \) and \( y \) is defined as:

\[
\delta_r(x, y) = \left( \sum_{i=1}^{n} |x_i - y_i|^r \right)^{\frac{1}{r}}.
\]

By the above definition, Manhattan distance is the \( L_1 \) norm and Euclidean distance is the \( L_2 \) norm. Another related distance measure is the \( L_\infty \) norm, \( \delta_\infty \), defined as:

\[
\delta_\infty(x, y) = \lim_{r \to \infty} \left( \sum_{i=1}^{n} |x_i - y_i|^r \right)^{\frac{1}{r}}.
\]

As \( r \) gets larger, only the dimension with the largest difference matters, so the equation above turns out to be equivalent to:

\[
\max(|x_1 - y_1|, |x_2 - y_2|, \ldots, |x_n - y_n|).
\]

4.4 Mahalanobis distance

\[
d(x, y) = \sqrt{(x - y)^{\top} \Sigma^{-1} (x - y)}
\]

4.5 Jaccard Coefficient

The Jaccard Coefficient, \( \delta_J \), between two sets, \( U \) and \( V \), is defined as:

\[
\frac{|U \cap V|}{|U \cup V|}.
\]


In the context of recommendation systems this type of distance measure makes sense when the information about user preferences is binary in nature, such as a rating system of like or dislike, or implicit preference data such as user purchases and items viewed for a significant amount of time.

4.6 Tanimoto Coefficient (Extended Jaccard Coefficient)

Measures the similarity of two sets by comparing the size of the overlap against the size of the two sets. In the case of binary attributes reduces to the Jaccard Coefficient.

\[
T(x, y) = \frac{x \cdot y}{\|x\|^2 + \|y\|^2 - x \cdot y}
\]

4.7 Log-likelihood

4.8 Cosine Distance

The Cosine distance between two vectors, \( x, y \in \mathbb{R}^n \), is defined as the size of the angle between them, i.e.,

\[
\arccos \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i^2}} = \arccos \frac{x \cdot y}{\|x\| \|y\|}
\]

4.9 Pearson-correlation coefficient (PCC)

Unlike the preceding measures the Pearson correlation coefficient is not a distance measure. It is a measure of the linear correlation (dependence) between two random variables \( X \) and \( Y \), giving a value between +1 and -1 inclusive, where 1 is total positive correlation, 0 is no correlation, and -1 is total negative correlation. The Pearson correlation coefficient of variables \( X \) and \( Y \) is defined as:

\[
\frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y}
\]

Let \( x \) and \( y \) be two users and vectors, \( x, y \in \mathbb{R}^n \), consist of ratings of the \( n \) items which both \( x \) and \( y \) have rated. Consider \( x \) and \( y \) as samples from the distributions of ratings of \( x \) and \( y \). Now we can substitute the sample covariance and standard deviations based on \( x \) and \( y \) into the equation above. We end up with the sample Pearson correlation coefficient, \( r_{x,y} \), below:

\[
r_{x,y} = \frac{\frac{1}{n} \sum_{i=1}^{n} (x_i - \mu_X)(y_i - \mu_Y)}{\sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \mu_X)^2} \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \mu_Y)^2}} = \frac{\sum_{i=1}^{n} (x_i - \mu_X)(y_i - \mu_Y)}{\sqrt{\sum_{i=1}^{n} (x_i - \mu_X)^2} \sqrt{\sum_{i=1}^{n} (y_i - \mu_Y)^2}}
\]
Let \( x' = x - \begin{bmatrix}\mu_X \\
\vdots \\
\mu_X \end{bmatrix} \) and \( y' = y - \begin{bmatrix}\mu_Y \\
\vdots \\
\mu_Y \end{bmatrix} \).

Then, \( r_{x,y} = \frac{x' \cdot y'}{\|x'\| \|y'\|} \).

This is just the cosign of the angle between our normalized ratings vectors. Users may have different conventions for employing a given ratings scale. For instance for a 1-5 discrete scale one user may only rate items they like, rating these 5, whereas another may only employ values 1,3, and 5 to signify items they like, are ambivalent about, and dislike respectively, while a third user may employ the full range of the rating scale. The centered ratings obtained by subtracting the mean rating of the user from each rating in the vector counteracts the effects heterogeneous usage conventions.

The correlation coefficient gives a similarity score between -1 and 1, where ratings close to zero signify little correlation, ratings closer to 1 signify that the two users rate items similarly, and items closer to -1 signify that the two users rate items dissimilarly.

The PCC as well as cosine similarity don’t take into account the length of the vectors being compared and so by themselves may provide counterintuitive results as regards similarity of users. For instance consider two users who have rated a lot of items but only a few in common and the ratings of these common items for the two users happen to be very close. This would give a PCC close to 1 when these users have obviously different rating behaviors.

To counteract these types of effects a weight may employed when constructing a metric to discount vectors with few entries. For users with \( n \) items commonly rated, a general formula for the weight may take the form \( \frac{n}{n+\lambda} \) where \( \lambda \) is a parameter which determines how much effect the shortness of ratings vectors detracts from the overall similarity.

If the ratings are normalized by the mean rating of the commonly reviewed items of a user instead of the mean rating of a user in general then the PCC is undefined when users overlap with a single item (as the mean rating equals the value of the rating for the single item) or when either user has the same rating for all overlapping items (as the mean rating equals the value of every item rated).

**4.10 Spearman rank correlation**

The Spearman correlation performs the same calculation as PCC, except that ratings are first mapped to ranks in the following manner. The highest rating gets rank 1, the next highest rank 2 and so on. If items have the same rating then each receives an average rank.

So, for instance, if 3 is the second highest rating and there are 4 items with a rating of 3 their ranking is computed as follows:

\[
\frac{2 + 3 + 4 + 5}{4}
\]
5 Content based recommendations

In a classic content based recommender items are modeled as vectors of features which may be represented on some discrete, real valued or binary scale. The value of the $i$-th component of an item vector represents the level which the item possesses the $i$-th feature. Users are also modeled as vectors of features. The value of the $i$-th component of an user vector represents the level which the user likes, appreciates, or is interested in the $i$-th feature.

Once feature vectors and user vectors are established recommendations may be produced by finding $k$ items similar to users using one of the similarity metrics discussed above.

Item feature vectors may be established by domain experts or by some form of machine learning algorithm.

Pandora Radio uses a content based recommender system to choose songs to play for users. Songs are classified using 400 musical features by a team of trained musicians. The system is not very scalable and in fact cannot even keep up with the proportion of new music being created which the website wishes to represent. [7]

In the case where text descriptions are available item vectors can be developed using TF-IDF weights drawn from the corpus of item descriptions.

5.1 Term Frequency-Inverse Document Frequency (TF-IDF)

- rare terms are not less relevant than frequent terms (IDF assumption);
- multiple occurrences of a term in a document are not less relevant than single occurrences (TF assumption);
- long documents are not preferred to short documents (normalization assumption).

In other words, terms that occur frequently in one document (TF = term-frequency), but rarely in the rest of the corpus (IDF = inverse-document-frequency), are more likely to be relevant to the topic of the document. In addition, normalizing the resulting weight vectors prevent longer documents from having a better chance of retrieval. [20, p. 78]

Let, $t_k$ be a key term in the corpus, $d_j$ a document in the corpus. Then let $f_{k,j}$ be the number of times a key term occurs in a document. Further, let $Q_j$ be the set of key terms in a document. Then,

$$\text{TF}(t_k, d_j) = \frac{f_{k,j}}{\max_{z \in Q_j} f_{z,j}}$$

Let $N$ be the number of documents in the corpus, and $n_k$ be the number of documents in the corpus where $t_k$ occurs at least once. Then,
\[
\text{IDF}(t_k, k_j) = \log \frac{N}{n_k}
\]

So,
\[
\text{TF-IDF}(t_k, d_j) = \text{TF} \times \text{IDF} = \frac{f_{k,j}}{\max_{z \in Q_j} f_{z,j}} \log \frac{N}{n_k}
\]

TF-IDF weights, \(w_{k,j}\), for term \(t_k\), and document \(d_j\) are usually normalized to fall in the [0,1] interval using cosine normalization. Let \(t\) be the total number of key term types (as opposed to tokens) in the corpus.

\[
w_{k,j} = \frac{\text{TF-IDF}(t_k, d_j)}{\sqrt{\sum_{s=1}^{t} \text{TF-IDF}(t_s, d_j)^2}}
\]

[20, p. 78]

6 Collaborative Filtering: Koren Notation

Special indexing letters are reserved for distinguishing users from items: for users \(u, v\), and for items \(i, j\). A rating \(r_{ui}\) indicates the preference by user \(u\) of item \(i\), where high values mean stronger preference. For example, values can be integers ranging from 1 (star) indicating no interest to 5 (stars) indicating a strong interest. Predicted ratings are distinguished from known ones, by using the notation \(\hat{r}_{ui}\) for the predicted value of \(r_{ui}\). The \((u, i)\) pairs for which \(r_{ui}\) is known are stored in the set \(K = \{(u, i)|r_{ui} \text{ is known}\}\). Regularization is controlled by constants which are denoted as: \(\lambda_1, \lambda_2, ...\)

7 Collaborative Filtering: Baseline Estimates

Typical CF data exhibit large user and item effects i.e., systematic tendencies for some users to give higher ratings than others, and for some items to receive higher ratings than others.

8 Collaborative Filtering I: Neighborhood Models

8.1 User-based Neighborhood Models

8.2 Item-based Neighborhood Models

9 Collaborative Filtering II: Matrix Factorization

"Matrix factorization recommenders use dimensionality reduction in order to uncover latent factors between users and items, e.g. by Singular Value De-
composition (SVD), probabilistic Latent Semantic Analysis or Latent Dirichlet Allocation.” [24]

“Matrix factorization models map both users and items to a joint latent factor space of dimensionality \( f \), such that user-item interactions are modeled as inner products in that space. The latent space tries to explain ratings by characterizing both products and users on factors automatically inferred from user feedback. For example, when the products are movies, factors might measure obvious dimensions such as comedy vs. drama, amount of action, or orientation to children; less well defined dimensions such as depth of character development or quirkiness; or completely uninterpretable dimensions.” [11]

9.1 User-Item Utility Matrix

Let \( r_{u,i} \) be user \( u \)'s rating of item \( i \). The total set of user ratings for a set of \( m \) users and \( n \) items can then be modeled by an \( m \times n \) matrix, \( R \), with rows of user ratings vectors, \( R_{i,j} = r_{i,j} \). This matrix, shown below, is referred to in the collaborative filtering literature as the user-item utility matrix.

<table>
<thead>
<tr>
<th></th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( r_{1,1} ) ... ( r_{1,n} )</td>
</tr>
<tr>
<td>m</td>
<td>( r_{m,1} ) ... ( r_{m,n} )</td>
</tr>
</tbody>
</table>

9.2 SVD

The basic idea behind matrix factorization models is that the information encoded for items in the columns of the utility matrix, and for users in the rows of the utility matrix is not exactly independent. Users and items should have relationships that may be obscured by the noisiness of the data. That is, if we did happen to have ratings for all the users and all the items in the dataset we should find either the rank of the utility matrix was significantly lower, perhaps orders of magnitude, lower than either \( m \) or \( n \), or else given a singular value decomposition of the complete utility matrix, \( R = U\Sigma V^T \), by zeroing out the \( k \) last diagonal entries of \( \Sigma \) we are left with a highly interpretable, close approximation of the original ratings matrix.
9.3 Latent Factors

In practice the matrix is factored into two matrices, conventionally denoted as $P$ and $Q^T$, so that $PQ^T \approx R$, and one can imagine the eigenvalues contained in the matrix $\Sigma$ as having been arbitrarily absorbed into either $P$ or $Q^T$ or both. $P_{u,x}$ is interpreted as user $u$’s level of appreciation of the latent or hidden factor $x$. $Q_{i,x}$ is interpreted as item $i$’s level of possession of latent factor $x$. Let $p_u$ denote the $u$-th row of matrix $P$ (associated with user $u$’s preferences for factors), and $q_i$ (associated with item $i$’s possession of factors) be the $i$-th row of the matrix $Q$. Then user-item interaction is modeled by $p_u \cdot q_i$, commonly written as $q_i^T p_u$.

Of course what we in fact have is not a complete matrix but a matrix with sparsely filled in entries. To find the missing entries in $R$ we could first initialize $P$ and $Q$ with some values and then minimize the sum of the squared differences of known ratings and predictions of ratings derived from multiplying the factor matrices. So, we can use either gradient descent or alternating least squares to solve the optimization problem below.
\[
\min_{q_i, p_i} \sum_{(u, i)} (r(u, i) - q_i^T p_i)^2
\]

### 9.4 Deriving matrix form of gradients

\( \mathcal{K} = \{(u, i) | R_{ui} \text{ is known}\} \)

\( \mathcal{K}_u = \{i | R_{ui} \text{ is known}\} \)

\( A \circ B \) is the elementwise product (Hadamard product) of \( A \) and \( B \).

\( \Omega \) is an \( m \times n \) matrix such that:

\[
\Omega_{ui} = \begin{cases} 
1 & \text{if } (u, i) \in \mathcal{K} \\
0 & \text{otherwise}
\end{cases}
\]

Basic MF Loss Equation (Squared error):

\[
\mathcal{L}(U, P) = \sum_{(v, i) \in \mathcal{K}} (R_{vi} - U_v P_i^T)^2 = \text{Tr} \left( ((R - U P^T) \circ \Omega)(R - U P^T)^T \right)
\]

To find the gradients with respect to parameter matrices \( U, P \), we can find the partial derivative of \( \mathcal{L} \) with respect to an arbitrary element for each matrix.

\[
\frac{\partial \mathcal{L}}{\partial U_{wq}} = \frac{\partial}{\partial U_{wq}} \sum_{(v, i) \in \mathcal{K}} (R_{vi} - U_v P_i^T)^2 = \sum_{(v, i) \in \mathcal{K}} \frac{\partial}{\partial U_{wq}} (R_{vi} - U_v P_i^T)^2 = -2 \sum_{i \in \mathcal{K}_w} (R_{wi} - U_{w_i} P_i^T) \frac{\partial}{\partial U_{wq}} U_{w_i} P_i^T
\]

\[
\sum_{i \in \mathcal{K}_w} \frac{\partial}{\partial U_{wq}} (R_{wi} - U_{w_i} P_i^T)^2 = 2 \sum_{i \in \mathcal{K}_w} (R_{wi} - U_{w_i} P_i^T) \frac{\partial}{\partial U_{wq}} (R_{wi} - U_{w_i} P_i^T) = -2 \sum_{i \in \mathcal{K}_w} (R_{wi} - U_{w_i} P_i^T) \frac{\partial}{\partial U_{wq}} U_{w_i} P_i^T
\]

\[
\sum_{i \in \mathcal{K}_w} \frac{\partial}{\partial U_{wq}} U_{w_i} P_i^T = \sum_{m=1}^{k} U_{wm} P_{im} = -2 \sum_{i \in \mathcal{K}_w} (R_{wi} - U_{w_i} P_i^T) \frac{\partial}{\partial U_{wq}} U_{w_i} P_i^T
\]

\[
\Rightarrow \nabla_U \mathcal{L} = -2 \left( \left( \Omega \circ (R - U P^T) \right) P \right)
\]

Notice that, \( \mathcal{L}(U, P) = \sum_{(u, i) \in \mathcal{K}} (R_{ui} - U_{ui} P_{ti})^2 = \sum_{(u, i) \in \mathcal{K}} ((R^T)_{iu} - P_{ti} U_{ui})^2 \).

So, by replacing the relevant symbols in the derivation above we have:

\[
\nabla_P \mathcal{L} = -2 \left( \left( \Omega \circ (R^T - PU^T) \right) U \right)
\]
First we define the following measures:
True Positives (TP): number of instances classified as belonging to class A that truly belong to class A.
True Negatives (TN): number of instances classified as not belonging to class A that in fact do not belong to class A.
False Positives (FP): number of instances classified as class A that do not belong to class A.
False Negatives (FN): instances not classified as belonging to class A but that in fact do
belong to class A. [19]

11.1 RMSE

11.2 Absolute mean error

11.3 accuracy

11.4 precision

11.5 recall

11.6 novelty

Distinction between novelty and serendipity. Serendipity: Helps the user find a surprisingly interesting item that she might not have otherwise discovered. Novelty: Suggests to the user an unknown item that she might have autonomously discovered. [19, p. 97]

11.7 diversity
12 References

References


17


13 Lectures


14 Correspondence Winter 2015

1/8/15 BH-AT

Updates:
- Found a useful book "Graph Mining: Laws, Tools & Case Studies"
- Few chapters on tensor (chpts 14 & 15)
- Free online checkout via WWU

Notes:
- Group meeting presentation on recommender systems?
- Multi-task DNN?
- How many shared layers? How many different layers?
- Tensor DNN?
- Factored tensor?

Next steps:
- BH read scoop paper
- Identify our novelty
- Discuss a first plan
- Discuss features of items
Thanks,
Brian

1/15/15 BH-AT

Updates:
- Found a paper that takes a hybrid approach to recommender systems

Notes:
- Novelty and diversity both desirable features of a recommender system, formalized in a paper cited in the hybrid paper
- Aaron will update doodle poll

Thanks,
Brian

1/17/15 BH-AT

Thanks. Slides for today will go in
/home/hutch_research/doc/20150114
The first person there needs to create the directory; e.g.,
/home/hutch_research/doc/20150114
AND change permission so that other group members can write to it:
chmod g+wx /home/hutch_research/doc/20150114
Or, if someone else has already created the directory for the week, you can just add to it.
That last portion of the path is a YYYYMMDD time stamp, of course, so it will change each week.

Thanks, Brian

1/20/15 BH-AT

Updates:
- Issue checking out repo due to directory permissions
Notes:
- Changed meeting time, 3-3:20 TF
- Recommender system for:
  - Classes
  - Professors
  - Clubs
  - Activities
  - Music
  - Books
  - Movies
  - Restaurants

Next steps:
- Keep reading
- Preparation steps:
  1) Identify the sets of entities (e.g. sets of classes, of restaurants, etc.)
  2) Find some seed recommendations (generic, un-catered)
     - Restaurants from Yelp?
     - Classes and/or profs by publicly available course evals?
     Rate professors?
     - Movies or books from publicly available datasets?
     Heuristics like best-sellers.
  3) [Optional] Content associated with each entity
     - Design our website/app
     - How can we get implicit feedback?
     - What form of explicit feedback? (Up/Down, Stars)
     - How to normalize user feedback?

- BH email Brittany Schade, Martin Granier(?), 49x instructors(?)

Way down the road:
- Corpus paper?

Thanks,
Brian
1/22/15 AT-BH

Below is the website for the large annual recsys conference.

http://recsys.acm.org/

I will comb over this sight to get a feel for current and past trends in recommender systems research. Also, there is access to some data from their yearly recommender system challenges.

Aaron

1/23/15 BH-AT

Updates:
- Found article on evaluating recommender systems

To discuss:
- Toolkits
- Python: scipy, numpy,
- Subversion

Next steps:
- Find a multi-domain corpus and/or single domain corpora in the domains of interest
- Send BH the paper on evaluating recommender systems
- Send BH the links to the ”three main” toolkits
- Go over paper(s) in greater detail
- Dig into recommender system combination (hybrid systems?)
- Preparation steps:
  1) Identify the sets of entities (e.g. sets of classes, of restaurants, etc.)
  2) Find some seed recommendations (generic, un-catered)
     - Restaurants from Yelp?
     - Classes and/or profs by publicly available course evals?
     - Rate professors?
     - Movies or books from publicly available datasets?
     - Heuristics like best-sellers.
  3) [Optional] Content associated with each entity
     - Design our website/app
     - How can we get implicit feedback?
     - What form of explicit feedback? (Up/Down, Stars)
     - How to normalize user feedback?
BH ask registrar or CS support for sanitized registration data, class lists and descriptions

Notes:
- Recommender system for:
  - Classes*
  - Professors(?)
  - Clubs*
  - Activities
  - Hobbies*
  - Music
  - Books
  - Movies
  - Restaurants*

Thanks,
Brian

1/23/15 AT-BH

Brian,

Here are the links to software tools that were mentioned in papers that I was going to check out.

RiVal: recommender evaluation toolkit (open source):
http://rival.recommenders.net/

MyMediaLite: Recommender System Library (open source):
http://www.mymedialite.net/

Lenskit: open-source tools for recommender systems
http://lenskit.org/

Apache mahout: scalable machine learning library (free?):
http://mahout.apache.org/

I’ve found several papers on recommendation evaluation but the one attached is from a recent recsys conference. I’ll put some stuff in the repo when I’m back at campus.
Aaron

Brian,
Below is the short comment about data that the scoop paper makes.

4.1 Amazon Data

Amazon 1 is the most famous e-business website to sell diverse products, such as books, DVDs, shoes, etc. The dataset [10] was crawled from Amazon website and it contains 1,555,170 users and 1-5 scaled ratings over 548,552 different products covering four domains: 393,558 books, 103,144 music CDs, 19,828 DVDs and 26,132 VHS video tapes. Obviously, the users preferences are dependent across these domains, so it is very suitable to test CDCF algorithms over this dataset.

Data Preparation: In this experiment, we selected Book and Music CD as the target domain to evaluate respectively. We filtered out users who have rated at least 50 books or 30 music CDs so that there are enough observations to be split in various proportions of training and testing data for our evaluation. Finally, 2,505 users were selected, and in addition we retrieved all items rated by these users in these four domains and set aside top K rated items for each domain respectively. Table 1 shows the statistics of the data for evaluation. Then, we constructed rating matrices over filtered out data for each domain.

Sparse Data Case: To simulate the sparse data problem, we constructed two sparse training sets, TR 20 and TR 75 , by respectively holding out 80 % and 25% data from the target domain Book, i.e. the remaining data of target domain for training is 20% and 75%. The hold-out data servers as ground truth for testing. Likewise, we also construct two other training sets TR 20 and TR 75 when choosing Music as the target domain. Unacquainted World Case: We randomly select half users and hold out all their data from target domain to simulate the unacquainted world phenomenon. The training set used for this case is denoted as TR uw.

I think we can build a data set that matches the red number in users.
"We filtered out users who have rated at least 50 books or 30 music CDs..."

This quote makes me wonder:
How do recommendation systems that employ a scale, fare when the number of users with meaningful feedback,
in proportion to the number of users who use the system and don’t contribute, varies?
This probably depends on how much implicit feedback the system incorporates.

A couple of things:
If it will take too much effort to gather descriptions for items lets forgoe that avenue for now,
but if it doesn’t take too much effort to get the descriptions with the listings and user stats we should get them looking out for future research projects.
Also, we should solicit and list reviews in the recommender system. I think people like this option and we could gather good data this way. If we don’t have fodder for word bags in the first place, if we collect reviews we’ll have fodder for word bags in the future.

Also attached, a paper, The Dynamics of Viral Marketing, which was cited with a reference to the data. Maybe, the Australasions got ahold of the data that backed this very interesting 40 something page paper. Maybe Leskovec et. al, would let us use this data too.

Aaron

1/27/15 AT-AS

Annika,
I am a graduate student in the Computer Science department at WWU beginning research. The professor I am collaborating with, Brian Hutchinson, expressed interest in developing an algorithm for a cross-domain recommendation system. Recommender systems are the devices that companies like Netflix, Amazon, Google and other corporate and non-profit entities use to filter the wide variety of options available and display preferable content to consumers. I ventured to Brian the idea of a Western recommendation system that could recommend classes, clubs, professors, events, restaurants and possibly books and movies. Professor Hutchinson is on board with this idea, and so I am contacting you to determine the level of student interest in such a system. I would like to talk to someone from the
AS about the reception and interest of students concerning such a system. Basically, there would be a website/mobile app which students anonymously logged in to that would provide recommendations for items in the above categories. On the research end we would track the preferences of users (anonymous of course) in the system and use this information to provide recommendations to students. I think this would be a great resource to students. The system could help students find resources and opportunities relevant to them that otherwise they might not have noticed.

No other university to date has implemented such a recommender system. So, in this respect the idea is cutting edge. Recommendor systems which cater to the interests of a local community are not typically employed. However, at the beginning of developing such a system I think it is important to gage how valuable students feel such a resource could be and to get their input on what would be important to them in a recommender system. I am willing to put together a talk about recommender systems, and the possibility for designing one catered to the needs and preferences of Western students. Afterwords, we could have a forum and some kind of vote about if students would like a recommendor system as part of their Western experience, and what features they would appreciate in such a system.

I think that this is an interesting project and avenue of research that may be beneficial to our community of students at Western. Please let me know what your thoughts are and direct me (or forward my email) to any other people I should talk to concerning the project.

Thanks for the time, Aaron Tuor

1/27/15 AS-AT

Aaron,

Thank you so much for reaching out to us about this, I admittedly dont know much about recommender systems but I would love to talk to you about how you see it helping to support the Western Community and what think our role in helping you could be. I would love to set up a time to meet with you this week if possible, what does your schedule look like?

Cheers,

Josie Ellison Communications Director Associated Students

1/27/15 BH-AT

Updates:
- Found excellent data Amazon reviews dataset
- AS wants to help with the recommender system

Next steps:
- Look over data and meta data
- See what our possibilities are
- Decide what to focus on
- Find the number of domains each user has reviews in
- Compute the domain entropy for each user:
  1) Normalize the counts of reviews in each domain
- Both: read this http://i.stanford.edu/~julian/pdfs/recsys13.pdf
- Decide if we want to do the user interface aspect
- Arrange two meetings:
  - Schade
  - AS

Thanks,
Brian

1/31/15 AT-BH

Updates:

Notes:

Discuss:
- Looking over data, computing things

Next steps:
- Create a document that has four tab-delimited columns:
  userid numberReviews numberDomains domainEntropy
- See your notes for the details
- Both: read this http://i.stanford.edu/~julian/pdfs/recsys13.pdf
- Cancel Schade meeting?
- Cancel AS meeting

Thanks,
Brian

2/3/15 BH-AT

Updates: - Did the analysis of the distribution of items and categories; a few wrinkles - Items can be in multiple categories - Can use his breakdown of items - 20-40 minutes to run - Some items not in categories - Meeting scheduled Wednesday 2/11 1-2pm in Fine Arts 109
Discussion: - Issue: items can be in multiple categories, some ideas: - Create a graph where nodes are categories, and edges are weighted by number of cross-listed - Create a bipartite graph linking items to categories - Create an item-category matrix, do SVD, pick max category.

Next steps: - Fix entropy - Make a graph, nodes are categories, weights on edges are the number of users who have posted reviews in both categories - Keep investigating small vs large file issues.

Thanks, Brian

2/13/15 BH-AT

Updates: - Made even fancier pictures than before

Next steps: - New graphs: - Add top K edges, normalized - Each category has edges to top 5 co-categories, with unnormalized line width - Radial arrangement, show *all* edges, but line width is log of unnormalized counts - Use some of the standard recommender system tools to train single-domain systems - Try training this on top 10 (all?) of our categories - Measure performance - Try using all as domain? Everything is an item. - Create a version of all data with no unknown users.

Down the road: - Develop our tensor or NN models?

Thanks, Brian

2/13/15 BH-HR

A few upcoming items to have on your radar...

- Unless we just started our project, you should plan to present a poster at WWU Scholar’s Week* (May 11-15th). Even if we just started, we still might want to present. Let’s discuss it. - Grad students should plan to present a poster at the WWU Graduate Conference** (May 17th).

These venues are great opportunities to get useful feedback on your work and raise its visibility. You can list your participation in these events on your resumes/CVs.

Also, upcoming paper plans that I have on my radar are: - DP submit latest T-DSN work to IEEE Signal Processing Letters - CN submit TIMIT TELLAR work to Interspeech 2015. Deadline 3/17. - KA+SK submit paper to JQAS or NESSIS once wavelet-style work is done. Deadline 6/15.

If we’ve discussed upcoming papers that I don’t have listed here, please remind me.

Thanks, Brian

http://www.wwu.edu/scholars/index.shtml
http://www.wwu.edu/gradschool/gradconfsubmissions.shtml
3/19/15 AT-BH

Brian, The guy in this video is describing a university recommender system like we were discussing.

http://www.youtube.com/watch?v=kHodCjWP1Us

Aaron

Brian,

We were talking about reading this attached paper earlier. Having finished reading the paper I don’t think we should evaluate our Mahout systems using the Mahout evaluation tools. We need an evaluation tool that can produce evaluations independent of the framework. We could code our own evaluations but I don’t think it’s necessary at this point to start from scratch. The RiVal recommender evaluation tool the attached paper is plugging was the tool used in evaluating the Netflix prize competition. I don’t know what kind of metric Netflix was using but I am sure the metric used is documented somewhere that I could find with a little searching. From my understanding RiVal acts as a wrapper for a recommender system that provides a testing framework which can evaluate under different strategies with full access to the parameters of a given evaluation strategy. Mahout performs worse than the other two frameworks in the attached paper using an out of framework evaluation (RiVal) and a comparison of cross-framework evaluations. This may be due to the movieLens data set being an integral dataset in the development of Lenskit. The occurrence of ”lens” in the two names is not a coincidence.

Fair travel,

Aaron

3/26/15 AT-BH

Brian,

I just finished reading J. McAuley and J. Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. , the paper linked to the dataset website. This paper is really good, and also relevant to both groups working with the Amazon dataset. I would be interested in discussing this paper with the other group if they feel like reading it. I understood most of the paper and am with a little preparation I could explain the basic latent factor model which is based on ratings alone.

Aaron

3/26/15 BH-HR

I’ve come up with a plan for individual meetings and group meetings that should work based on the doodle poll answers. See the schedules here:
https://hutchresearch.cs.wwu.edu/index.php/Individual_meetings

Please double check and make sure that these will work for your schedule. Let me know ASAP if there's a conflict. I’m planning to start meetings next week (first week of the quarter).

Thanks, Brian

3/28/15 BH-TS

If you’re on this email, then I have included you in the rank, sparsity and tensor subgroup. As the group meetings page shows, this subgroup will meet each Wednesday from 4-5pm. Due to scheduling conflicts the first two weeks of the quarter, our first meeting won’t happen until April 15th, when we’ll discuss the following paper:

http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7050385

You should be able to download the pdf if you access the page from any WWU machine. It’s very unlikely we’ll finish it, but let’s try to make a good jump on it.

Please read it carefully ahead of time and come prepared to discuss your understanding of it (including asking questions about parts that aren’t clear to you). You should probably budget a few hours of prep time for reading / googling terms / etc. At that meeting we will set a plan for future meetings.

Thanks, Brian

3/31/15 BH-AT

Updates: - Created expt - 3 diff rec systems - 28x3 combinations - One per subtopics - Didn’t evaluate any of them yet - Printed out top 5 items predicted for first 1k users - Needed to map alphanumeric IDs to numeric ideas

Discuss: - Watches

Next steps: - Dig deeper into the actual algorithms being used - May want to read at some point:

- Look into Rival javadocs (http://rival.recommenders.net/site/0.1/apidocs/) - BH read evaluation paper (recsys-benchmarking) - BH read combined text and matrix paper

Thanks, Brian
15  Correspondence Spring 2015

4/3/2015 BH-AT

Updates: - Found good collab filtering paper
Discuss: - Matrix factorization approaches to collaborative filtering
Notes: - Important characteristics: - Scalable - Add new user without complete retraining
- Idea: - RNN transduction from product to review, per user & item - Sparse + low rank exponential model where users/items are factors
Next steps: - Dig deeper into the actual algorithms being used - May want to read at some point: http://research.microsoft.com/en-us/um/people/gzweig/Pubs/SLT2012.pdf - Look into Rival javadocs (http://rival.recommenders.net/site/0.1/apidocs/) - BH read evaluation paper (recsys-benchmarking) - BH read combined text and matrix paper
Thanks, Brian

4/7/2015 BH-AT

Updates:

Next steps: - Joint train: - Ratings-based recommender - Classifier that maps from product metadata (including description) to rating - Then have a per-user system to map from product metadata to rating - Generalizes very easily to new product (zero ratings needed) - Doesn’t generalize to new users - Some questions: 1) What has been done related to this? - Look into the Related Work cited in McAuley & Leskovec 2) What form should our metadata to rating model take - Idea: log-linear models, where each user’s weights are a column in a low rank matrix - Sigmoid output? - Columns for diff. users are linearly dependent 3) Jointly solve for matrices A, B, C in a combined optimization problem - Search for prior work - (Simultaneous OR joint OR collective) matrix factorization AND recommender system - See if there are any links to/from this paper: http://www.cs.cmu.edu/~ajit/pubs/Singh2008.pdf

Thanks, Brian

4/7/2015 AT-BH

Brian,

The idea you proposed today seems promising. In the literature I have seen CF algorithms give good recommendations for new users, but none that give good recommendations for new items. Content based
systems can give good recommendations for new items, and in a sense if we think of product
descriptions
as clues to the content of an item, what you are proposing is
a blend of content based and CF methods. I think it would be most attractive to design a model that
can accommodate new users and new items. For our MF component we should look into
Assymetric SVD
(section 4, Koren, Factorization Meets the Neighborhood), which gives comparable (better)
results to standard matrix factorization approaches, but has the added benefit of good rec-
ommendations
for new users.
Aaron

4/8/2015 AT-BH

Brian,

Nothing new in ”Solving the cold start problem using product related content”. This is
a student’s final project for some class wherein she uses the HFT model developed in the
McAuley, Leskovec paper with the product descriptions from the Amazon dataset instead of
customer reviews as a text resource. Also she collected some outside resource text related
to items as well for movies. The HFT model does about the same using descriptions instead
of reviews for the amazon dataset.
Aaron

4/10/2015 BH-AT

Updates: - Lit review for simultaneous matrix factorization
Notes: - Famous LDA paper: http://machinelearning.wustl.edu/mlpapers/paper_files/
BleiNJ03.pdf

Next steps: - Make poster - Column 1: - Problem overview - Recommender system overview
- Basic matrix factorization model - Column 2: Cold Start New Users - State the problem -
Discuss existing solutions (should include figures/illustration, can include results) - Column
3: Cold Start New Items - State the problem - Discuss existing solutions (should include
figures/illustration, can include results) - Column 4: Proposed solution to both cold start
problems - State the problem - Our proposed model - Data - Joint train: - Ratings-based
recommender - Classifier that maps from product metadata (including description) to rating
- Asymmetric SVD for generalization to users - Then have a per-user system to map from
product metadata to rating - Generalizes very easily to new product (zero ratings needed) -
Doesn’t generalize to new users - Some questions: 1) What has been done related to this? -
Look into the Related Work cited in McAuley & Leskovec 2) What form should our metadata
to rating model take - Idea: log-linear models, where each user’s weights are a column in a
low rank matrix - Sigmoid output? - Columns for diff. users are linearly dependent 3) Jointly
solve for matrices A, B, C in a combined optimization problem - Search for prior work - Online
training of recommendations - The cold start problem - Enrollment/registration/initiation
for recommender systems (could also be under name of ”active learning”)

Possible contributions of the research: - Rigorous assessment on cold start context

Thanks, Brian

4/11/2015 AT-BH

Brian,

The hottest trend in addressing the cold start user problem is to make decision trees of small
depth from a ratings matrix and use the trees and
a set of queries to estimate a user profile. Then you can use the item profiles from the matrix
factorization and dot product them with the

estimated user profile to estimate ratings. I’m looking into decision tree methods, and I
haven’t seen any based on extracting eigenvalues from

a UV decomposition derived from matrix factorization. Once a complete matrix of ratings
is estimated via latent factor we could SVD this matrix

and find items which participate most heavily in latent factors in order to build an efficient
decision tree. This type of model for building trees seems

simple enough to be scalable. Maybe we could talk about this on Monday or Tuesday.

Aaron

4/12/2015 AT-BH

Brian,

We don’t have this kind of data but we could make a model by randomly generating demo-
graphic information according to some distribution.

This looks nice because we could generate item and user profiles for cold start. Probably it’s
too simple. Picture attached.

Aaron
4/17/2015 BH-AT

Notes: - AT has leads on data

Next steps: - AT send readings and videos to KM - AT port information about data over to KM - Amazon - MovieLens - AT port information about baseline systems over to KM - KM run baseline experiments - Split the data into train, dev and test - Cut unknown user(s) from data - Could be included in item global baseline - Just a single category (probably books) - All data thrown in together - Use mahout three SVD-based approaches - Tune parameters - Port info to wiki - Relevant papers - Background info - Books, videos, etc. - Tools - Tips & Tricks

Poster notes: - can’t -¿ cannot - Use consistent form for the two cold starts $\sum_u \sum_i$

Later steps: - Nail down our new model

Thanks, Brian

4/17/2015 AT-BH

Brian,

We discussed the possibility today of pruning the dataset for a denser observation set. For the baseline project that Katy is getting started on I think it would be interesting to vary the sparseness of data for several baseline models and see how robust they are with respect to sparseness. For the amazon dataset this would entail eliminating portions of the data and seeing how much the accuracy improves. For an artificial dataset generated according to some distribution this would entail eliminating entries in the dataset and seeing how much the accuracy diminishes. The best metrics for recommender systems are currently a debated topic, and I haven’t seen anything in the literature I’ve surveyed that addresses this aspect. If this hasn’t been done before, I think something like this would be an interesting and productive experiment for Katy.

Aaron

4/21/2015 BH-AT,KM

Updates: - AT and KM met to discuss relevant background - Basic methods - Intro to matrix factorization approach - Presented poster on Saturday

Discuss: - Objective

Next steps: - Work on the gradient w.r.t. U, P, and W for our objective - See matrix cookbook - AT port info about data to KM - Amazon - MovieLens - AT port info about baseline systems over to KM - Look into Rival, see if it does what we need - KM run baseline experiments - Split the data into train, dev and test - Cut unknown user(s) from data - Could be included
in item global baseline - Just a single category (probably books) - All data thrown in together
- Use mahout three SVD-based approaches - Tune parameters - KM Port info to wiki
- Relevant papers - Background info - Books, videos, etc. - Tools - Tips & Tricks

Thanks, Brian

4/28/2015 BH-AT,KM

Updates: - Started reading paper (involves boosting)

Next steps: - KM Port info to wiki - Relevant papers - Background info - Books, videos, etc. - Tools - Tips & Tricks - AT port info about data to KM - Amazon - MovieLens - AT port info about baseline systems over to KM - Look into Rival, see if it does what we need - AT and KM arrange weekly meeting time - KM run baseline experiments - Split the data into train, dev and test - Cut unknown user(s) from data - Could be included in item global baseline - Just a single category (probably books) - All data thrown in together - Use mahout three SVD-based approaches - Tune parameters - AT derive gradient - AT send me a list of possible papers for reading group

Thanks, Brian

5/1/2015 BH-AT

Aaron, I spent some time working on the gradients for our joint factorization model. Attached are my notes. The first page defines my variables (sort of) and the last page has the gradients (the circle dot denotes element-wise multiply). The rest of can probably be ignored – I only include it for context. I’m about 90 percent sure on these gradients... sure enough to try to code it up and see if our results make sense. I didn’t include global, user or item biases, and I didn’t include any regularization terms. The latter will be easy to add, the former shouldn’t be too bad, either.

You need to produce the following matrices (where N is the number of users, M is the number of items, and V is the number of word types in our vocabulary):

- Omega (NxM). This is the ratings mask. Contains a 1 in (i,j) if user i rated item j, and 0 otherwise. - R (NxM). This is the ratings matrix. (i,j) contains the rating user i gave item j.
- Q (VxM). This is the word-document matrix. The (i,j) entry should be 1 if word i appears in the document associated with item j, and 0 otherwise.

Once we have these three matrices, I think our algorithm will only take 10 lines of Matlab code to implement, and should be fairly fast since all of the operations are vectorized and Matlab will parallelize them locally.

Thanks, Brian
5/2/2015 AT-BH

Brian,

I like this idea, initializing P and U with an SVD of R, and I have heard it mentioned before without any explanation of the details. I thought about this for a bit and here is my first idea for making this sensible. We can’t just SVD R and use these results for initialization because R is not complete. We have to make a decision about how to fill in the missing entries. I think there are two reasonable approaches. We could find optimal global attributes (user bias, rating mean and item bias) and subtract these from the entries in R to make R prime, in which case we can make the missing entries zero. Then we could do an SVD on R prime, initialize P and U as you have described, and have an objective function on R prime which doesn’t care about the global attributes since we already guessed them. Or else we could find the optimal global attributes and use these to estimate ratings which we place in the missing entries of R to make a different R prime. Then we do an SVD on this R prime and use this to initialize P and U as you have described, and our objective function is concerned with the original R (in this case it doesn’t matter if we use R or R prime since Omega will filter). We should find a sensible heuristic for initializing our P and U for hopefully quicker convergence. Having said this, in the papers I have read on MF approaches to recommending, if anyone ever mentions how they initialize, they claim random initialization. Random may make sense if you are optimizing global predictions at the same time as user-item predictions.

Aaron

5/3/2015 BH-AT

I think it’s worth considering these initialization options, but a good rule is to start with the simplest approach and work outward from there. If people claim to do random initialization, then we should try that. The main advantage of the SVD is that it is simple but more intelligent. If we see a big win there, we should consider some of the variants you propose.

5/3/2015 AT-BH

Brian,

When I was first came to you about research I was entertaining the idea of modelling vagueness in natural language with a sigmoid function.

Logistic regression would be a good tool to learn these functions (associated with a word or concept and a context) from data (if we can find the right sort).

We could make a very simple artificially intelligent machine with vague logic for the semantics of simple sentence.
I will look into this if we have a break from the recommender research.

5/5/2015 AT-BH

Brian,

I finally ran across the paper which was the catalyst for a lawsuit against IMDb, and probably led to the lawsuit against Netflix for the Netflix Prize Competition. IMDb now only allows restricted access to random samplings from their database of user profiles.

Title: Robust De-anonymization of Large Datasets (How to Break Anonymity of the Netflix Prize Dataset)

Probably neither of us will have time to read this but it looks interesting.

Also, the movielens metadata has titles in IMDb format, and IMDb has for each movie, a collection of plot descriptions made by users. So, with some amount of work we can get a text dataset for the movielens data.

There is a pre 2008 (pre-aforementioned paper) movielens 100K dataset that has user profiles (age, zip code, and a few other things) which if we want we could use for a 3-way collective factorization model that I mentioned. We could factor a one-hot sparse matrix of user demographic features (every zip code, every whatever other discrete category, every sex, ... etc for a row). I have seen a recent paper (I think the one you sent me and we discussed briefly because of the cool evaluation technique) where the authors use this dataset to address user cold start.

5/5/2015 BH-AT,KM

Aaron

Updates: - Met with Katy on Friday - Processed data into category csvs, removed top two users (Unknown and some pro reviewer) - Katy will focus on recommender system for Movies amazon category

Discuss: - Sparse matrix

Next steps: - AT to focus on our model, basic MF, collective embeddings - KM to focus on SVD, SVD++, FunkSVD - Each of the dev or test subsets has 5 % of the ratings - Download MovieLens and get it into right format - All sizes - AT add new figure and our gradients to the poster - AT tune R on dev set - AT process text - AT start coding in Matlab - matload/matsave for dense matrices - smatload/smatsave for sparsematrices - KM Port info to wiki - Relevant papers - Background info - Books, videos, etc. - Tools - Tips & Tricks - AT use old text normalization script (from SLP), and then use export PERL5LIB=$PERL5LIB:/home/hutch_research/lib /home/hutch_research/bin/generateVocab.pl Matlab ascii format (each line is one non-zero element):
in matlab... 

\[ X = \text{load('myFile.txt');} \]
\[ X = \text{spconvert(X);} \]

5/11/2015 BH-AT

I’ll take a look at the poster this morning. The script is


Thanks, Brian

5/12/2015 AT-BH

Brian,

For our model, the reason I don’t think the upper factorization from our picture will add anything to the model is because omega is applied to both factorizations. What if we replaced R on the bottom with \( UP^T \). The we have a U on the left of both sides of the bottom factorization so they cancel. The bottom factorization would then be \( P^T = W^T Q \). We could then not apply omega to the bottom factorization and use all the information from Q. This is basically the Local Collective Embeddings model rewritten which is unfortunate but I do think this may still be a superior base factorization to fancy up, as no solution of an equation is necessary online. Attached is a poor drawing that will make the preceding comments clear.

5/12/2015 AT-BH

Brian,

Here’s a rough idea. The reviews of users for products in the Amazon dataset could be put into a user-item-term tensor, and we could use a collective factorization with the item-user ratings matrix for a model. I’m not sure how this relates to your idea elaborating our model using tensors and neural networks but it seems similar from what I gathered at last Friday’s meeting.

Aaron

5/12/2015 BH-AT

I like the direction here. What does the \((i,j,k)\)th element of the user-item-tensor denote/signify? My neural net idea was slightly different, I was thinking of joint factorization of a user-item matrix with one of the following two:

- A user-hidden matrix, where ”hidden” is a deep representation of the document (this would be the last weight layer in a neural net to predict a scalar rating value). - A user-term-hidden matrix, where we map from user and term vector to a hidden representation (which is used to predict a scalar rating value).
But I’m interested to discuss your idea more.

Thanks, Brian

5/13/2015 AT-BH

I was thinking at (i,j,k) index there is a 1 if word k occurs in the review of item j by user i. When thinking about this tensor I was thinking, what do you get when you toss out information? If we sum over the term dimension to collapse this tensor into a matrix (and turn all non-zero entries into 1’s) we have a matrix like the user-document matrix from LCE (1 if user commented on news article zero otherwise). On the other hand if we sum over the user dimension to collapse this tensor into a matrix (and turn all non-zero entries into 1’s) we have a document term matrix where a document is the collection of reviews associated with an item.

Maybe we could do something interesting with a tensor like this.

Aaron

5/14/2015 AT-BH

Brian,

We’ve done a lot of research and thinking on the recommender subject. I think we both have a good understanding on several newish extensions of the MF approach. Right now we are on the cusp of implementing some models which are variations of joint factorization.

While we are fleshing out ideas for a more sophisticated model, the results from these simpler models will be fodder for the idea mill.

My paper idea is this: Exploring Cold Start Recommending Through Matrix Factorization

-with the data sets we’ve gathered and are making we can explore some interesting questions

-How do models fair on sparse item data?
-How do models fair on sparse user data?
-How do models fair on sparser data in general?
-How does the user/item ratio affect the importance you place the the above metrics?
-How to achieve an optimum balance and temper the affect of sparsity on the success of a system?
What is the best model to achieve a minimum slope on a function from sparsity to accuracy?

If you think this might be a good topic I can search around to see if anyone has directly addressed research on sparsity in recommender systems or some analogous branch of study.

5/15/2015 BH-AT,KM

Update: - Script is running, doesn’t finish in 2 hours - Regenerating data since it seems like some reviews were lost - Looked over mahout code

Notes: - Basic idea for using screen is: 1) Type screen (this creates a new screen session) Start something running now... 2) To detach type ”Ctrl+a, d” 3) To reattach, one of two options: screen -x (if there’s only one screen session) screen -r SOMENAME (if there are multiple)

Next steps: - Redownload tar file(s) - AT dig up links - KM focus on making new map - KM run mahout baseline - AT to focus on our model, basic MF, collective embeddings - KM to focus on SVD, SVD++, FunkSVD - Is there a way to save our models? - AT tune R on dev set - AT start coding in Matlab - matload/matsave for dense matrices - smatload/smatsave for sparsematrices - KM Port info to - Relevant papers - Background info - Books, videos, etc. - Tools - Tips & Tricks

Thanks, Brian

5/19/2015 BH-AT,KM

Updates: - Made 5 dictionaries for Amazon dataset - Ran dataset split on Amazon and MovieLens - But will replace this later with Matlab - Discussed Mahout - Global idx to category idx map done

Discuss: - DONE: Vocabs - DONE: New model

Notes: - Most important things in Matlab: save(’filenamegoeshere’,’variablenamegoeshere’); save(’my.mat’,’X’); load(’filenamegoeshere’); X * Y; X .* Y; X’; sum(sum(X)); X(3,:); X(:,5); size(X,1); size(X,2); eye(5); zeros(5,10); 1 - X; x x;

Next steps: - KM: First experiments will use Mahout’s auto dev split, then using our testing scripts for evaluation in the end - Fix Mahout random seed so its comparable with itself - AT finish data prep script (use hash tables) - Discard if multiple matches or no match - AT to focus on our model, basic MF, collective embeddings - KM to focus on SVD, SVD++, FunkSVD - Is there a way to save our models? - AT start coding in Matlab - matload/matsave for dense matrices - smatload/smatsave for sparsematrices - KM Port info to - Relevant papers - Background info - Books, videos, etc. - Tools - Tips & Tricks - AT: Try vocab with cutoff of 5
5/22/2015 BH-AT,KM

Updates: - Training SVD baselines - The few are the best - Tuning K (rank), lambda (L2
regularization coeff), # iterations - Wiki info added

Discuss: - Results

Next steps: - KM: - Try fewer iteration, see how it affects RMSE - Try SVD++, FunkSVD,
others? - Figure out how to save models trained - Set up ssh keypair - Keep up the good
work porting info - Relevant papers - Background info - Books, videos, etc. - Tools - Tips &
Tricks - AT: - Give KM relevant data - Matlab path: /usr/central/bin/matlab - Implement
our models in Matlab

Thanks, Brian

5/23/2015 BH-AT

Brian,

Here is a link to a collection of matlab scripts for methods related to Latent Semantic
Analysis.

http://lear.inrialpes.fr/verbeek/software.php

Aaron

5/26/2015 BH-AT,KM

Updates: - Start coding in Matlab - Loading data, making mask, etc.

Next steps: - AT+KM collaborating on evaluating baseline - KM runs baselines, out-
puts predictions - AT provides matlab evaluation code - Visualize RMSE as a function
of both user and item sparsity - KM: - Try fewer iteration, see how it affects RMSE - Try
SVD++, FunkSVD, others? - Figure out how to save models trained - Set up ssh keypair
https://support.cs.wwu.edu/index.php/Setting_up_SSH_keys_in_Unix_Linux - Keep up the
good work porting info https://hutchresearch.cs.wwu.edu/index.php?title=Recommendation_System
- Relevant papers - Background info - Books, videos, etc. - Tools - Tips & Tricks - AT: -
Give KM relevant data - Matlab path: /usr/central/bin/matlab - Implement our models in
Matlab

5/29/2015 BH-AT,KM

Updates: - Start coding in Matlab - Loading data, making mask, etc.

Next steps: - AT+KM collaborating on evaluating baseline - KM runs baselines, out-
puts predictions - AT provides matlab evaluation code - Visualize RMSE as a function
of both user and item sparsity - KM: - Try fewer iteration, see how it affects RMSE - Try
SVD++, FunkSVD, others? - Figure out how to save models trained - Set up ssh keypair
https://support.cs.wwu.edu/index.php/Setting_up_SSH_keys_in_UNIX_Linux - Keep up the
good work porting info https://hutchresearch.cs.wwu.edu/index.php?title=Recommendation_System
- Relevant papers - Background info - Books, videos, etc. - Tools - Tips & Tricks - AT:
Give KM relevant data - Matlab path: /usr/central/bin/matlab - Implement our models in
Matlab

Thanks, Brian

6/5/2015 BH-AT,KM

Updates: - Swamped, no progress - Getting better at relevant languages

Next steps: - KM fill in AT on everything done - Send him your script - If KM gets script
from AT, can include it in the pipeline - AT - Finish lit review - Port lit to wiki - Eval
script (finished this weekend?) - AT+KM collaborating on evaluating baseline - AT provides
matlab evaluation code - KM runs baselines, outputs predictions?? - Try SVD++, FunkSVD,
others? - Implement our models in Matlab - Visualize RMSE as a function of both user and
item sparsity

Later steps: - Figure out KM 49x project (recommender related?)

Thanks, Brian