Using SVD to Recommend Movies

Michael Percy

University of California, Santa Cruz

Last update: December 12, 2009
1. Introduction

2. Singular Value Decomposition

3. Experiments

4. Conclusion
Collaborative Filtering

- Goal: Predict the preferences of a user based on data taken from “similar” users
Collaborative Filtering (cont.)

- Idea: Distill the essence of the data into features that can be used for prediction
- The Singular Value Decomposition (SVD) can be used to factor the matrix to give an approximation of the missing values
The Netflix Competition

- The Netflix competition pitted teams from across the world to find ways to predict movies better for users.
- The Netflix data set includes:
  - Integral ratings from 1 to 5 for 17,000 movies
  - From 480,000 users
  - Totalling over 100 million ratings
- Essentially a sparse $480000 \times 17000$ matrix
- Total entries: roughly 8.5 billion. Matrix is close to 99% empty!
Singular Value Decomposition (SVD)

- Idea: “Compress” each dimension of the matrix (users × movies) to approximate the known values. Each dimension is then represented by a finite number of features which are combined to estimate the missing values.

\[ \mathbf{M} = \mathbf{U} \Sigma \mathbf{V}^T \]
SVD Definition

The SVD is a factorization of a matrix such that:

\[ M = U \Sigma V^T \]

Where:
- \( M \) is the ratings matrix
- \( \Sigma \) is a diagonal matrix of singular values
- and \( U \) and \( V \) are the eigenvectors of \( MM^T \) and \( M^T M \), respectively.

Intuitively, the singular values in \( \Sigma \) are “weighting factors” for \( U \) and \( V \). The singular values are the square roots of the eigenvalues of \( MM^T \) corresponding to the eigenvectors in \( U \) and \( V \).
Using SVD for Matrix Approximation

If approximation or compression is desired, a lower-rank SVD $\tilde{M}$ can be computed from $M$:

- Rank $r$ is chosen and only the $r$ singular values of greatest magnitude are used.
- Thus the rank of $U$ and $V$ are reduced as well.
- $U$, $\Sigma$, and $V$ are reduced to dimension $m \times r$, $r \times r$, and $r \times n$ respectively.

This approximation minimizes the Frobenius norm of $A = M - \tilde{M}$:

$$\|A\|_F = \sqrt{\sum_{i=1}^{\min\{m, n\}} \sigma_i^2} = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2}$$
An incremental algorithm for SVD was developed by Simon Funk based on a paper by Genevieve Gorrell, in which the SVD is approximated numerically via gradient descent.

An approximation rank $r$ is chosen beforehand such that there are $r$ feature vectors for movies and users.

The $\Sigma$ matrix is left blended into the feature vectors, so only $U$ and $V$ remain.

Initial values for these feature vectors are set arbitrarily, i.e. $0.1$.

Gradient descent is used to minimize the function for each feature vector along users and movies.
Incremental SVD (cont.)

Based on the gradients, the incremental SVD update function for Gradient Descent is:

\[
\begin{align*}
\text{err} &= \eta \left( M_{ij} - \tilde{M}_{ij} \right) \\
\text{userval} &= U_{if} \\
U_{if} &= U_{if} + \text{err}(V_{fj}) \\
V_{fj} &= V_{fj} + \text{err}(\text{userval})
\end{align*}
\]

For learning rate \( \eta \), user \( i \), movie \( j \), and feature \( f \in \{1, r\} \).
Experiments

The data set was reduced so that experimentation and testing was feasible. The set was reduced randomly to:

- 2,000 movies
- 1,000,000 users
- 3,867,371 total ratings

And the data was further split into an 80% training set and a 20% validation set.
Experiments

Learning Rate

Several learning rates were tested, for $\eta = 0.1, 0.01, \text{ and } 0.001$
Number of features with no regularization

The rank $r$, or number of features chosen, was first tested without regularization. The learning rate $\eta = 0.001$ was used based on the above experiments.
Number of features with regularization

Regularization proved to be key in being able to use more features without overfitting.

This is a graph of the validation set RMSE:
Number of features with regularization cont.

And the training set RMSE:

![Graph showing the relationship between features and RMSE. The x-axis represents the number of features, and the y-axis represents the root mean squared error (RMSE). The graph includes lines for different regularization values, showing how RMSE decreases as the number of features increases.]
In conclusion, the incremental SVD seems to do a pretty good job of approximating the missing values in the matrix and predicting expected ratings.

- Learning rate 0.001 seemed to perform the best out of the learning rates tested
- The ideal number of features for this data set is around 70, with regularization parameter 0.015
- Out of the values for regularization tested, the most effective regularization coefficient value was 0.020 on 90 features, getting a validation RMSE of .9101.