URL to the paper: http://www.mimuw.edu.pl/~paterek/ap_kdd.pdf
URL to the poster: http://www.mimuw.edu.pl/~paterek/ap_kdd_poster.pdf

7.04%
improvement over Netflix's recommendation system

Improving regularized singular value decomposition for collaborative filtering

Arkadiusz Paterek <paterek@mimuw.edu.pl>
Institute of Informatics, Warsaw University

The goal

- to predict users' preferences for movies, as accurately as possible

Outline of the approach

- take many well predicting methods
- combine their results with linear regression on the *test* set

The data

The Netflix Prize data consists of the files: training.txt, probe.txt and qualifying.txt.

- *Training* set: training.txt without random 15% of probe.txt – c.a. $100,000,000$ ratings
- *Test* set: 15% of probe.txt – c.a. $200,000$ ratings
- *Validation* set: results reported by the Netflix Prize evaluation system – 50% of qualifying.txt

The most important methods in the ensemble

- regularized SVD - by Simon Funk
- *New:* regularized SVD with biases
- clustering users using K-means
- postprocessing SVD results with item-item K-nearest neighbors
- *New:* postprocessing SVD with kernel ridge regression
- *New:* weighted linear model for each item
- *New:* methods similar to SVD, but with fewer parameters

Together they give 6.34% improvement.
The 7.04% solution is a result of linear regression of 56 predictors and 63 two-way interactions between them, plus partial cross-validation.

Regularized SVD

Predictions for user i and movie j:

$$\hat{y}_{ij} = u_i^{\mathrm{T}} v_j$$

where u_i and v_j are K-dimensional vectors of parameters.

Parameters are estimated by minimizing the sum of squared residuals, one feature at a time, using gradient descent with regularization and early stopping:

$$r_{ij} = y_{ij} - \hat{y}_{ij}$$

$$u_{ik} \mathrel{+}= lrate * (r_{ij} v_{jk} - \lambda u_{ik})$$

$$v_{jk} \mathrel{+}= lrate * (r_{ij} u_{ik} - \lambda v_{jk})$$

- original constant parameters unchanged $lrate = .001$, $\lambda = .02$
- baseline prediction substracted from ratings before training
- $K = 96$ features
- *New:* stopping criterion: when the error rate on the test set increases
- after training each feature, predictions are clipped to $< 1, 5 >$ range

Regularized SVD with biases

We add biases to the regularized SVD model, one parameter c_i for each user and one d_j for each movie:

$$\hat{y}_{ij} = c_i + d_j + u_i^{\mathrm{T}} v_j$$

Weights c_i, d_j are trained simultaneously with u_{ik} and v_{jk}:

$$r_{ij} = y_{ij} - \hat{y}_{ij}$$

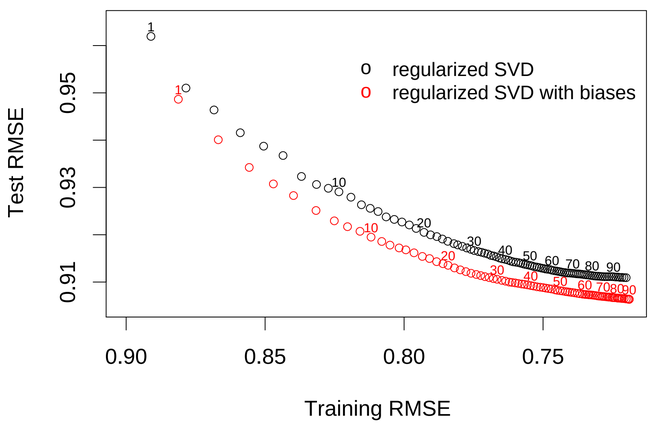
$$u_{ik} \mathrel{+}= lrate * (r_{ij} v_{jk} - \lambda u_{ik})$$

$$v_{jk} \mathrel{+}= lrate * (r_{ij} u_{ik} - \lambda v_{jk})$$

$$c_i \mathrel{+}= lrate * (r_{ij} - \lambda_2 (c_i + d_j - \mu))$$

$$d_j \mathrel{+}= lrate * (r_{ij} - \lambda_2 (c_i + d_j - \mu))$$

- constant parameters: $lrate = .001$, $\lambda_2 = .05$, $\mu = 3.6033$.
- no baseline prediction



Weighted linear model for each item

For a given item (movie) j we are building a weighted linear model, using as predictors, for each user i, a binary vector indicating which movies the user rated:

$$\hat{y}_{ij} = m_j + e_i * \sum_{j_2 \in J_i} w_{j_2}$$

where J_i is the set of movies rated by user i, constant m_j is the mean rating of movie j, and constant weights $e_i = (|J_i| + 1)^{-1/2}$.

Learning: gradient descent with early stopping.

SVD-based methods with fewer parameters

The regularized SVD model has $O(NK + MK)$ parameters (N users, M movies, K features). We propose two models with $O(MK)$ parameters.
The idea is, instead of fitting u_i for each user separately, to model u_i as a function of a binary vector indicating which movies the user rated.
For example $u_{ik} \approx e_i \sum_{j \in J_i} w_{jk}$, where J_i is the set of movies rated by user i (possibly including movies for which we do not know ratings, e.g. qualifying.txt) and constant weights $e_i = (|J_i| + 1)^{-1/2}$. The first proposed model is:

$$\hat{y}_{ij} = c_i + d_j + e_i \sum_{k=1}^{K} v_{jk} \sum_{j_2 \in J_i} w_{j_2 k}$$

The second proposed model is similar, but parameters v_{jk} and w_{jk} are merged and there are no constant weights e_i:

$$\hat{y}_{ij} = c_i + d_j + \sum_{k=1}^{K} v_{jk} \sum_{j_2 \in J_i} v_{j_2 k}$$

Learning: gradient descent with regularization and early stopping.

Postprocessing SVD with kernel ridge regression

One idea to improve SVD is to discard all weights u_{ik} after training and predict y_{ij} for each user i using v_{jk} as predictors. Using ridge regression for that purpose gives similar results to the regularized SVD. It turns out that a better prediction is obtained by using kernel ridge regression with Gaussian kernel.
Let X be a matrix of observations - each row of X is normalized vector of features of one movie j rated by user i: $x_{j_2} = \frac{v_j}{||v_j||}$.
For each user i, we can predict his vector of ratings y using ridge regression:

$$\hat{\beta} = (X^T X + \lambda I)^{-1} X^T y, \quad \hat{y}_j = x_j^{\mathrm{T}} \hat{\beta}$$

Equivalent dual formulation involving Gram matrix XX^T:

$$\hat{\beta} = X^T (X X^T + \lambda I)^{-1} y$$

- in place of the scalar product we use a Gaussian kernel (defining similarity or distance between observations):

$$K(x_j^{\mathrm{T}}, x_k^{\mathrm{T}}) = exp(2(x_j^{\mathrm{T}} x_k - 1))$$

- predictions in kernel ridge regression:

$$\hat{y}_j = K(x_j^{\mathrm{T}}, X)(K(X, X) + \lambda I)^{-1} y$$

- constant parameter $\lambda = .5$
- for efficiency reasons the number of observations (movies) per user was limited to 500

Predictor	Test RMSE with basic predictors	Test RMSE with basic p. and with SVD with biases	Cumulative test RMSE
six basic predictors	.9826	.9039	.9826
regularized SVD	.9094	.9018	.9094
reg. SVD with biases	.9039	.9039	.9018
K-means	.9410	.9029	.9010
postproc. SVD with 1-NN	.9525	.9013	.8988
postproc. SVD with KRR	.9006	.8959	.8933
weighted linear models	.9506	.8995	.8902
small SVD-based 1	.9312	.8986	.8887
small SVD-based 2	.9590	.9032	.8879

Table 1: Linear regression results - RMSE on the test set

Experimental results

- predictor with best results: postprocessing SVD with kernel ridge regression
- SVD with biases combined with six basic predictors gives .9039 on the test set and .9070 (4.67% improvement) on the validation set (qualifying.txt)
- combining all methods from the table plus 2 two-way interactions gives .8877 on the test set and .8911 (6.34% improvement) on the validation set
- the 7.04% solution is a result of combining 56 predictors and 63 two-way interactions
- running times varied from 45min to 20h on a PC with 2GHz processor and 1.2GB RAM

Conclusions

- combining many methods with linear regression on the test set is a very effective approach
- good results of postprocessing SVD with kernel ridge regression suggest possibility of developing better methods than SVD for collaborative filtering