Rating Prediction using Funk-SVD

Analytics Vidhya

2006 "Funk-SVD" and the Netflix prize

- Netflix announced a million dollar prize
  - Goal:
    - Beat their own "Cinematch" system by 10 percent
    - Measured in terms of the Root Mean Squared Error
  - Effect:
    - Stimulated lots of research
- Idea of SVD and matrix factorization picked up again
  - S. Funk (pen name)
    - Use fast gradient descent optimization procedure
    - http://sifter.org/~simon/journal/20061211.html

Algorithm Structure

- Initialize values to train (item/user feature vectors) to arbitrary starting point
  - Must be non-zero
  - Usually must be random
- Try to predict each rating in the dataset
- Use error and update rule to update values for next rating/sample
- Iterate until convergence
  - Stops moving
  - Iterated enough times

Get Rid of Sigma

- Decomposition:

$$\boldsymbol{R = P\Sigma Q^T}$$

$$\boldsymbol{R = PQ^T}$$

- Scoring Rule after dropping Sigma

$$s(i;u) = \hat{r}_{ui} = \sum_f p_{uf} q_{if}$$

Deriving FunkSVD

- Recall our prediction equation

$$s(i;u) = \hat{r}_{ui} = \sum_f p_{uf} q_{if}$$

- We compute Error

$$e_{ui} = r_{ui} - \hat{r}_{ui}$$
$$= r_{ui} - \sum_f p_{uf} q_{if}$$

- We then compute the derivatives $\frac{d}{dp_{uf}} e_{ui}^2 \; and \; \frac{d}{dq_{if}} e_{ui}^2$

$$\theta = < P, Q > \qquad \theta_n = \theta_{n-1} + \Delta g(\theta_{n-1})$$

Analytics Vidhya

Deriving FunkSVD

- Calculating derivative for puf and qif

$$\frac{d}{dp_{uf}} e_{ui}^2 = 2e_{ui} \frac{d}{dp_{uf}} e_{ui}$$

$$= 2e_{ui} \frac{d}{dp_{uf}} (r_{ui} - \sum_f p_{uf} q_{if})$$

$$= -2e_{ui} q_{if}$$

$$p'_{uf} = p_{uf} - \lambda(-2e_{ui} q_{if})$$
$$q'_{if} = q_{if} - \lambda(-2e_{ui} p_{uf})$$

- Final Equations (add regularization to discourage large values)

$$p_{uf} = p_{uf} + \lambda(e_{ui} q_{if} - \gamma p_{uf})$$
$$q_{if} = q_{if} + \lambda(e_{ui} p_{uf} - \gamma q_{if})$$

Analytics Vidhya

Summary

- Matrix factorization
  - Generate low-rank approximation of matrix
  - Detection of latent factors
  - Projecting items and users in a smaller k-dimensional space
- Prediction quality can increase as a consequence of…
  - Small & faster model
  - filtering out some "noise" in the data and
  - detecting nontrivial correlations in the data
- Depends on the right choice of the amount of data reduction
  - number of singular values in the SVD approach
  - Parameters can be determined and fine-tuned only based on experiments