7-3: Training SVDs

Introduction to Recommender Systems

Introduction

- SVDs are slow
- SVDs require care dealing with missing data
- This video: gradient descent
  - Much faster
  - Deals well with missing data

Gradient Descent

- Convex optimization method
- We won't cover gradient descent in general here – that's multivariate calculus
- Basic idea: how far off we are at one point (prediction) tells us how to improve the prediction
- We'll use *stochastic* gradient descent

Algorithm Structure

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

FunkSVD

- Train features one at a time
  - Train first feature until convergence
  - Then train the next
- Ignore missing values (mostly)
- Learn offset from personalized mean
- Fold singular values into left/right matrices

$$R = B + UV^{\mathrm{T}}$$

$$p_{a,i} = b_{a,i} + \sum_f u_{a,f} v_{i,f}$$

Parameters

$$\Delta v_{if} = \lambda((r_{a,i} - p_{a,i})u_{af} - \gamma v_{if})$$
$$\Delta u_{af} = \lambda((r_{a,i} - p_{a,i})v_{if} - \gamma u_{af})$$

- λ – learning rate (how fast to train)
  - Higher – faster training, lower accuracy
- γ – regularization factor
  - prevents 'overfitting' by encouraging small values

Where'd this come from?

- Gradient descent to minimize (global) RMSE of final model
- Update is derivative of squared error with respect to user or item feature value
  - With regularization term
- General idea: use gradient descent (or other method) to find values that minimize the error (cost function)
- Caveat: no longer a true SVD

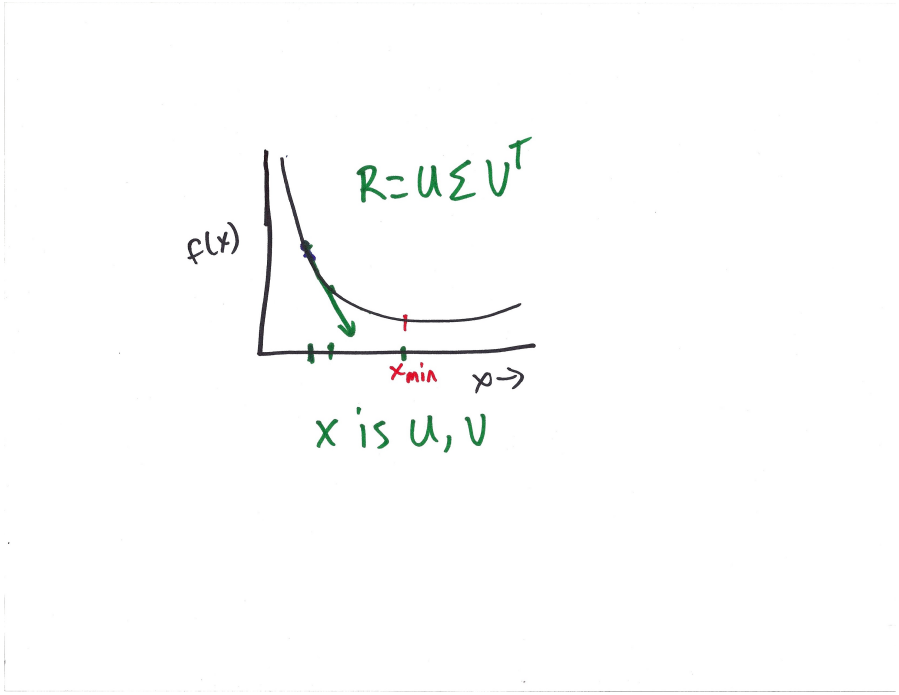
Extending: SVD++

- Minimize-the-error model is very general
  - Gradient descent can solve many of them
- Extension: incorporate has-rated data
- N(a) is items rated (or purchased) by a

$$p_{a,i} = b_{a,i} + (u_a + \sqrt{|N(a)|} \sum_{i \in N(a)} y_i) v_i^{\mathrm{T}}$$

Conclusion

- Gradient descent allows efficient approximation of the SVD from known data
- Extensible to additional data – minimize the error over some predictor
  - Limited by error's ability to capture what's important to recommendation problem
- Many models exist using similar methods



7-3: Training SVDs

$$p_{a,i} = b_{ai} + \sum_f u_{af} v_{if}$$

initialize to 0.1 or nonzero, small random

$$e_{ai} = r_{a,i} - p_{a,i}$$

$$\Delta v_{if} = \lambda (e_{ai} u_{af} - r v_{if})$$

$$\Delta u_{af} = \lambda (e_{ai} v_{if} - r u_{af})$$

small (~0.1) — regularization

learning rate (0.001)