Case Study 4: Collaborative Filtering

Collaborative Filtering
Matrix Completion
Alternating Least Squares

Machine Learning for Big Data
CSE547/STAT548, University of Washington
Emily Fox
February 11th, 2014

©Emily Fox 2014

Collaborative Filtering

- **Goal:** Find movies of interest to a user based on movies watched by the user and others
- **Methods:** matrix factorization, GraphLab
Cold-Start Problem

- **Challenge**: Cold-start problem (new movie or user)
- **Methods**: use features of movie/user
Netflix Prize

- Given 100 million ratings on a scale of 1 to 5, predict 3 million ratings to highest accuracy

- 17770 total movies
- 480189 total users
- Over 8 billion total ratings
- How to fill in the blanks?

Matrix Completion Problem

- \( X_{ij} \) known for black cells
- \( X_{ij} \) unknown for white cells
- Rows index users
- Columns index movies
- Filling missing data?

Figures from Ben Recht
Interpreting Low-Rank Matrix Completion (aka Matrix Factorization)

\[ X = L R' \]

Identifiability of Factors

- If \( r_{uv} \) is described by \( L_u \), \( R_v \) what happens if we redefine the “topics” as
- Then,
Matrix Completion via Rank Minimization

- Given observed values:
  - Find matrix
  - Such that:
  - But…
  - Introduce bias:

- Two issues:

Approximate Matrix Completion

- Minimize squared error:
  - (Other loss functions are possible)

- Choose rank $k$:

- Optimization problem:
Coordinate Descent for Matrix Factorization

\[
\min_{L,R} \sum_{(u,v): r_{uv} \neq ?} (L_u \cdot R_v - r_{uv})^2
\]

- Fix movie factors, optimize for user factors
- First observation:

Minimizing Over User Factors

- For each user \( u \): \[
\min_{L_u} \sum_{v \in V_u} (L_u \cdot R_v - r_{uv})^2
\]

- In matrix form:

- Second observation: Solve by
Coordinate Descent for Matrix Factorization: Alternating Least-Squares

\[
\min_{L,R} \sum_{(u,v) : r_{uv} \neq ?} (L_u \cdot R_v - r_{uv})^2
\]

- Fix movie factors, optimize for user factors
  - Independent least-squares over users
    \[
    \min_{L_u} \sum_{v \in V_u} (L_u \cdot R_v - r_{uv})^2
    \]

- Fix user factors, optimize for movie factors
  - Independent least-squares over movies
    \[
    \min_{R_v} \sum_{u \in U_v} (L_u \cdot R_v - r_{uv})^2
    \]

- System may be underdetermined:

- Converges to

Effect of Regularization

\[
\min_{L,R} \sum_{(u,v) : r_{uv} \neq ?} (L_u \cdot R_v - r_{uv})^2
\]

\[
X = \begin{bmatrix} L \end{bmatrix} \begin{bmatrix} R' \end{bmatrix}
\]
What you need to know…

- Matrix completion problem for collaborative filtering
- Over-determined -> low-rank approximation
- Rank minimization is NP-hard
- Minimize least-squares prediction for known values for given rank of matrix
  - Must use regularization
- Coordinate descent algorithm = “Alternating Least Squares”

Case Study 4: Collaborative Filtering

SGD for Matrix Completion
Matrix-norm Minimization

Machine Learning for Big Data
CSE547/STAT548, University of Washington
Emily Fox
February 11th, 2014
Stochastic Gradient Descent

\[
\min_{L,R} \frac{1}{2} \sum_{u,v} (L_u \cdot R_v - r_{uv})^2 + \frac{\lambda_u}{2} \|L\|_F^2 + \frac{\lambda_v}{2} \|R\|_F^2
\]

- Observe one rating at a time \( r_{uv} \)
- Gradient observing \( r_{uv} \):

Updates:

Local Optima v. Global Optima

- We are solving:
  \[
  \min_{L,R} \sum_{u,v} (L_u \cdot R_v - r_{uv})^2 + \lambda_u \|L\|_F^2 + \lambda_v \|R\|_F^2
  \]

- We (kind of) wanted to solve:

- Which is NP-hard…
  - How do these things relate???
### Eigenvalue Decompositions for PSD Matrices

- Given a (square) symmetric positive semidefinite matrix:
  - Eigenvalues:
  - Thus rank is:

- Approximation:
- Property of trace:

- Thus, approximate rank minimization by:

### Generalizing the Trace Trick

- Non-square matrices ain’t got no trace
- For (square) positive semidefinite matrices, matrix factorization:
- For rectangular matrices, singular value decomposition:
- Nuclear norm:
Nuclear Norm Minimization

- Optimization problem:
  
- Possible to relax equality constraints:

- Both are convex problems! (solved by semidefinite programming)

Analysis of Nuclear Norm

- Nuclear norm minimization = convex relaxation of rank minimization:

\[
\begin{align*}
\min_{\Theta} \ \text{rank}(\Theta) & \quad \min_{\Theta} \ ||\Theta||_* \\
r_{uv} = \Theta_{uv}, \forall r_{uv} \in X, r_{uv} \neq ? & \quad r_{uv} = \Theta_{uv}, \forall r_{uv} \in X, r_{uv} \neq ?
\end{align*}
\]

- Theorem [Candes, Recht '08]:
  - If there is a true matrix of rank $k$,
  - And, we observe at least
  \[
  C' k n^{1.2} \log n
  \]
  random entries of true matrix

- Then true matrix is recovered exactly with high probability via convex nuclear norm minimization!
  - Under certain conditions
Nuclear Norm Minimization versus Direct (Bilinear) Low Rank Solutions

- Nuclear norm minimization: 
  \[ \min_{\Theta} \sum_{uv} (\Theta_{uv} - r_{uv})^2 + \lambda \|\Theta\|_* \]
  - Annoyng because:

- Instead: 
  \[ \min_{L,R} \sum_{uv} (L_u \cdot R_v - r_{uv})^2 + \lambda_u \|L\|_F^2 + \lambda_v \|R\|_F^2 \]
  - Annoyng because:
  - But \( \|\Theta\|_* = \inf \left\{ \min_{L,R} \frac{1}{2} \|L\|_F^2 + \frac{1}{2} \|R\|_F^2 : \Theta = LR^T \right\} \)
    - So
    - And

- Under certain conditions [Burer, Monteiro '04]

What you need to know…

- Stochastic gradient descent for matrix factorization

- Norm minimization as convex relaxation of rank minimization
  - Trace norm for PSD matrices
  - Nuclear norm in general

- Intuitive relationship between nuclear norm minimization and direct (bilinear) minimization