Case Study 4: Collaborative Filtering

Review:
Matrix Completion
Alternating Least Squares

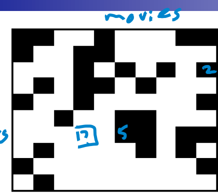
Machine Learning for Big Data
CSE547/STAT548, University of Washington

Emily Fox

February 13th, 2014

Matrix Completion Problem

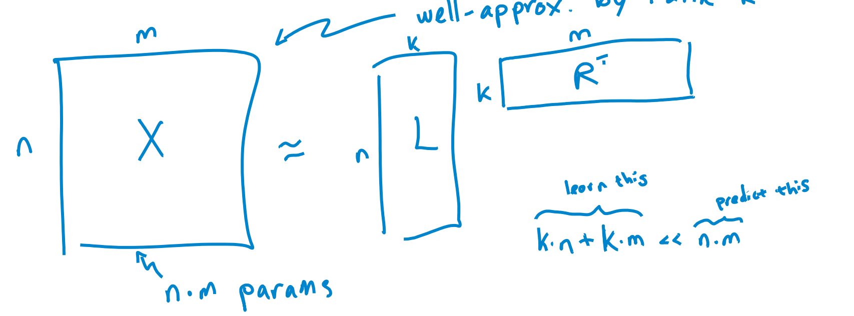
X =



X_{ij} known for black cells
X_{ij} unknown for white cells
Rows index users
Columns index movies

- Filling missing data?

well-approx. by rank k matrix



Matrix Completion via Rank Minimization

- Given observed values: $(u,v,r_{uv}) \in X$ some $r_{uv} = ?$ ← unobs. rating
- Find matrix ${}_n\overset{m}{\Theta}$
- Such that: $\underbrace{\Theta_{uv} = r_{uv}}_{\text{fit } r_{uv}\neq ? \text{ perfectly}} \quad \forall r_{uv} \neq ?$ ← all obs. ratings
- But… want Θ to be low-rank
- Introduce bias:

$$\min_{\Theta} \ \text{rank}(\Theta)$$
$$\text{s.t. } \Theta_{uv} = r_{uv} \quad \forall r_{uv} \neq ?$$

for $k < \min(n,m)$

- Two issues:
  - NP-hard
  - you can't hope to get exact matching

Approximate Matrix Completion

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

$$\min_{\Theta} \sum_{(u,v): r_{uv} \neq ?} (\Theta_{uv} - r_{uv})^2$$

- Choose rank *k:*

$${}_n\overset{m}{\Theta} = {}_n\overset{k}{L}\, {}^{n}R^T$$

- Optimization problem:

$$\min_{L,R} \sum_{r_{uv}} (L_u \cdot R_v - r_{uv})^2$$

non-convex opt. problem … local optima only

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 (R), optimize for user factors (L)
- First observation:

$$\min_{L_1,\dots,L_n} \sum_{(u,v):r_{uv}\neq ?} (L_u \cdot R_v - r_{uv})^2 =$$

V_u = set of movies user u rated

$$\min_{L_1,\dots,L_n} \sum_u \sum_{v\in V_u} (L_u \cdot R_v - r_{uv})^2 =$$

← ind. opt. problem for each user

$$\sum_u \underbrace{\min_{L_u} \sum_{v\in V_u} (L_u \cdot R_v - r_{uv})^2}_{\text{next slide}}$$

← data parallel problem

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:

$$\left\| \underbrace{\begin{bmatrix} R_v \end{bmatrix}}_{|V_u| \times k} \underbrace{\begin{bmatrix} L_u \end{bmatrix}}_{k \times 1} - \underbrace{\begin{bmatrix} r_{uv} \end{bmatrix}}_{|V_u| \times 1} \right\|_2^2$$

Think of as $\|X\beta - y\|_2^2$ normal LS problem

- Second observation: Solve by
  - matrix inversion
  - gradient methods
  - ⋮

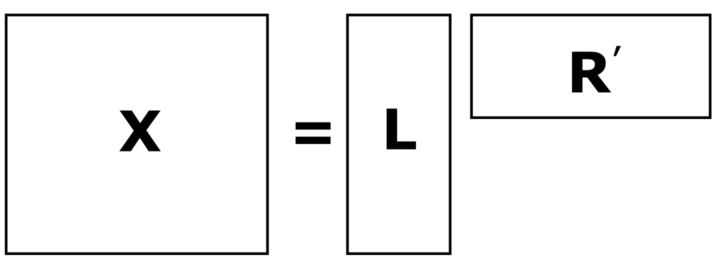
Coordinate Descent for Matrix Factorization: Alternating Least-Squares

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

- 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

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



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 13th, 2014

Stochastic Gradient Descent

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

- Observe one rating at a time r_{uv}
- Gradient observing r_{uv}:
- Updates:

Local Optima v. Global Optima

- We are solving:

$$\min_{L,R} \sum_{r_{uv}} (L_u \cdot R_v - r_{uv})^2 + \lambda_u ||L||_F^2 + \lambda_v ||R||_F^2$$

- 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:

$$\min_{\Theta} rank(\Theta) \qquad \min_{\Theta} ||\Theta||_*$$

$$r_{uv} = \Theta_{uv}, \forall r_{uv} \in X, r_{uv} \neq ? \qquad r_{uv} = \Theta_{uv}, \forall r_{uv} \in X, r_{uv} \neq ?$$

- 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: $\min_{\Theta} \sum_{r_{uv}} (\Theta_{uv} - r_{uv})^2 + \lambda ||\Theta||_*$
  - Annoying because:
- Instead: $\min_{L,R} \sum_{r_{uv}} (L_u \cdot R_v - r_{uv})^2 + \lambda_u ||L||_F^2 + \lambda_v ||R||_F^2$
  - Annoying because:
  - But $||\Theta||_* = \inf \left\{ \min_{L,R} \frac{1}{2}||L||_F^2 + \frac{1}{2}||R||_F^2 : \Theta = LR' \right\}$
    - 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

Case Study 4: Collaborative Filtering

Nonnegative Matrix Factorization
Projected Gradient

Machine Learning for Big Data
CSE547/STAT548, University of Washington
Emily Fox
February 13th, 2014

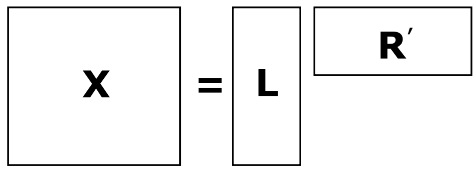
Matrix factorization solutions can be unintuitive…

- Many, many, many applications of matrix factorization
- E.g., in text data, can do topic modeling (alternative to LDA):

$$\mathbf{X} = \mathbf{L}\,\mathbf{R}'$$

- Would like:
- But…

Nonnegative Matrix Factorization



- Just like before, but

$$\min_{L \geq 0, R \geq 0} \sum_{r_{uv}} (L_u \cdot R_v - r_{uv})^2 + \lambda_u ||L||_F^2 + \lambda_v ||R||_F^2$$

- Constrained optimization problem
  - Many, many, many, many solution methods… we'll check out a simple one

Projected Gradient

- Standard optimization:
  - Want to minimize: $\min_{\Theta} f(\Theta)$
  - Use gradient updates:

$$\Theta^{(t+1)} \leftarrow \Theta^{(t)} - \eta_t \nabla f(\Theta^{(t)})$$

- Constrained optimization:
  - Given convex set *C* of feasible solutions
  - Want to find minima within *C*: $\min_{\Theta} f(\Theta)$

$$\Theta \in \mathcal{C}$$

- Projected gradient:
  - Take a gradient step (ignoring constraints):
  - Projection into feasible set:

Projected Stochastic Gradient Descent for Nonnegative Matrix Factorization

$$\min_{L\geq 0, R\geq 0} \frac{1}{2}\sum_{r_{uv}}(L_u \cdot R_v - r_{uv})^2 + \frac{\lambda_u}{2}||L||_F^2 + \frac{\lambda_v}{2}||R||_F^2$$

- Gradient step observing r_{uv} ignoring constraints:

$$\begin{bmatrix} \tilde{L}_u^{(t+1)} \\ \tilde{R}_v^{(t+1)} \end{bmatrix} \leftarrow \begin{bmatrix} (1-\eta_t\lambda_u)L_u^{(t)} - \eta_t\epsilon_t R_v^{(t)} \\ (1-\eta_t\lambda_v)R_v^{(t)} - \eta_t\epsilon_t L_u^{(t)} \end{bmatrix}$$

- Convex set:
- Projection step:

What you need to know…

- In many applications, want factors to be nonnegative
- Corresponds to constrained optimization problem
- Many possible approaches to solve, e.g., projected gradient

Case Study 4: Collaborative Filtering

Cold Start Problem

Machine Learning for Big Data
CSE547/STAT548, University of Washington
Emily Fox
February 13th, 2014

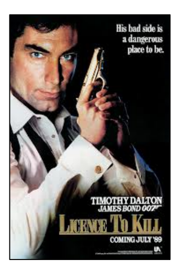
Cold-Start Problem

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

$$\phi(\text{Skyfall}) = \begin{array}{l} \text{action} \\ \text{romance} \\ \vdots \end{array} \begin{pmatrix} 7 \\ 2 \\ 0 \\ \vdots \end{pmatrix}$$

IN THEATERS

$$\phi(\text{FRWL}) = \begin{pmatrix} 8 \\ 1 \\ 6 \\ \vdots \end{pmatrix}$$



Cold-Start Problem More Formally

- Consider a new user *u'* and predicting that user's ratings
  - No previous observations
  - Objective considered so far:

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

  - Optimal user factor:
  - Predicted user ratings:

An Alternative Formulation

- A simpler model for collaborative filtering
  - We would not have this issue if we assumed all users were identical
  - What about for new movies? What if we had side information?
  - What dimension should *w* be?
  - Fit linear model:
  - Minimize:

Personalization

- If we don't have any observations about a user, use wisdom of the crowd
  - **Address cold-start problem**
- Clearly, not all users are the same
- Just as in personalized click prediction, consider model with global and user-specific parameters
- As we gain more information about the user, forget the crowd

User Features…

- In addition to movie features, may have information about the user:
- Combine with features of movie:
- Unified linear model:

Feature-based Approach versus Matrix Factorization

- Feature-based approach:
  - Feature representation of user and movies fixed
  - Can address cold-start problem
- Matrix factorization approach:
  - Suffers from cold-start problem
  - User & movie features are learned from data
- A unified model:

Unified Collaborative Filtering via SGD

$$\min_{L,R,w,\{w_u\}_u} \frac{1}{2}\sum_{r_{uv}}(L_u \cdot R_v + (w + w_u)\cdot\phi(u,v) - r_{uv})^2$$
$$+ \frac{\lambda_u}{2}||L||_F^2 + \frac{\lambda_v}{2}||R||_F^2 + \frac{\lambda_w}{2}||w||_2^2 + \frac{\lambda_{wu}}{2}\sum_u ||w_u||_2^2$$

- Gradient step observing r_{uv}
  - For L,R $\begin{bmatrix} L_u^{(t+1)} \\ R_v^{(t+1)} \end{bmatrix} \leftarrow \begin{bmatrix} (1-\eta_t\lambda_u)L_u^{(t)} - \eta_t\epsilon_t R_v^{(t)} \\ (1-\eta_t\lambda_v)R_v^{(t)} - \eta_t\epsilon_t L_u^{(t)} \end{bmatrix}$
  - For w and w_u:

What you need to know…

- Cold-start problem
- Feature-based methods for collaborative filtering
  - Help address cold-start problem
- Unified approach

Case Study 4: Collaborative Filtering

Connections with Probabilistic Matrix Factorization

Machine Learning for Big Data
CSE547/STAT548, University of Washington
Emily Fox
February 13th, 2014

Probabilistic Matrix Factorization (PMF)

- A generative process:
  - Pick user factors
  - Pick movie factors
  - For each (user,movie) pair observed:
    - Pick rating as $L_u R_v$ + noise
- Joint probability:

PMF Graphical Model

$$P(L, R \mid X) \propto P(L)P(R)P(X \mid L, R)$$

- Graphically:

Maximum A Posteriori for Matrix Completion

$$P(L,R|X) \propto P(L,R,X) = p(L)p(R)p(X \mid L,R)$$
$$\propto e^{\frac{-1}{2\sigma_u^2}\sum_{u=1}^{n}\sum_{i=1}^{k} L_{ui}^2} e^{\frac{-1}{2\sigma_v^2}\sum_{v=1}^{m}\sum_{i=1}^{k} R_{vi}^2} e^{\frac{-1}{2\sigma_r^2}\sum_{r_{uv}}(L_u \cdot R_v - r_{uv})^2}$$

MAP versus Regularized Least-Squares for Matrix Completion

- MAP under Gaussian Model:

$$\max_{L,R} \log P(L,R \mid X) =$$
$$-\frac{1}{2\sigma_u^2}\sum_u\sum_i L_{u_i}^2 - \frac{1}{2\sigma_v^2}\sum_v\sum_i R_{v_i}^2 - \frac{1}{2\sigma_r^2}\sum_{r_{uv}}(L_u \cdot R_v - r_{uv})^2 + \text{const}$$

- Least-squares matrix completion with L_2 regularization:

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

- Understanding as a probabilistic model is very useful! E.g.,
  - Change priors
  - Incorporate other sources of information or dependencies

What you need to know…

- Probabilistic model for collaborative filtering
  - Models, choice of priors
  - MAP equivalent to optimization for matrix completion