Case Study 4: Collaborative Filtering

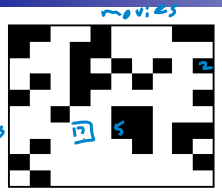
Review: Matrix Completion Alternating Least Squares

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

Emily Fox

February 13th, 2014

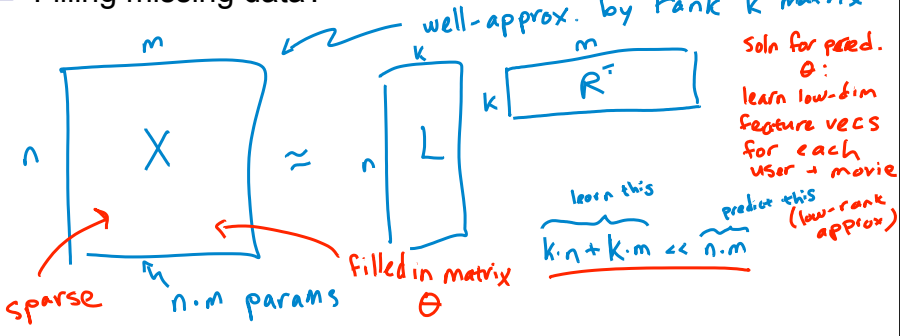
Matrix Completion Problem



X_{ij} known for black cells
X_{ij} unknown for white cells

Rows index users
Columns index movies

- Filling missing data?



Matrix Completion via Rank Minimization

- Given observed values: $(u, v, r_{uv}) \in X$ some $r_{uv} = ?$ ← unobs. rating
- Find matrix Θ ($n \times m$) ← filled in (not sparse)
- Such that: $\theta_{uv} = r_{uv}$ $\forall r_{uv} \neq ?$ ← all obs. ratings
  - fit $r_{uv} \neq ?$ perfectly ← match ratings for all obs. values of X
- But… want θ to be low-rank
- Introduce bias: one possible objective

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

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

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

Approximate Matrix Completion

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

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

instead: ← allow for some error

- Choose rank k:

$$\Theta_{(n \times m)} = L_{(n \times k)} R^T_{(k \times m)}$$

← fix k

- 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, optimize for user factors
- 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

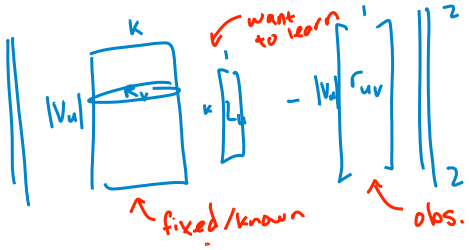
$\min_{L_1,\dots,L_n} \sum_u \sum_{v\in V_u} (L_u \cdot R_v - r_{uv})^2 =$ ← ind. opt. problem for each user

$\sum_u \min_{L_u} \sum_{v\in V_u} (L_u \cdot R_v - r_{uv})^2$ ← data parallel problem

next slide

Minimizing Over User Factors

- For each user u: $\min_{L_u} \sum_{v\in V_u} (L_u \cdot R_v - r_{uv})^2$ just consider one user
- In matrix form: equivalent



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

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

- Fix movie factors, optimize for user factors
  - Independent least-squares over users

$$\min_{L_u} \sum_{v \in V_u} (L_u \cdot R_v - r_{uv})^2 + \lambda_u \|L\|$$

- Fix user factors, optimize for movie factors
  - Independent least-squares over movies

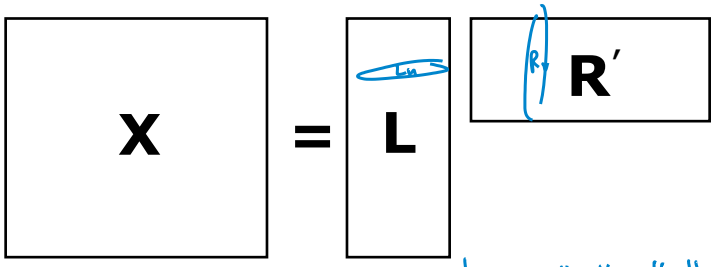
$$\min_{R_v} \sum_{u \in U_v} (L_u \cdot R_v - r_{uv})^2 + \lambda_v \|R\|$$

- System may be underdetermined: use regularization
- Converges to local optima

Effect of Regularization

$$\|A\|_F = \sqrt{\sum_i \sum_j A_{ij}^2}$$

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



If $\|\cdot\| = \|\cdot\|_F^2$ each subproblem uses $\|L_u\|_2^2$ → ridge regression

If $\|\cdot\| = \|\cdot\|_1$ each subproblem $\|L_u\|_1$ → solved by LASSO methods

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 $r_{uv}^{(t+1)}$ $\epsilon_t = L_u^{(t)} \cdot R_v^{(t)} - r_{uv}^{(t)}$
- Gradient observing r_{uv}:

$$\frac{\partial F}{\partial L_u} = \epsilon_t R_v + \lambda_u L_u$$
$$\frac{\partial F}{\partial R_v} = \epsilon_t L_u + \lambda_v R_v$$
$$\nabla F_t = \begin{bmatrix} \epsilon_t R_v + \lambda_u L_u \\ \epsilon_t L_u + \lambda_v R_v \end{bmatrix}$$

- Updates: step size η_t : $\begin{bmatrix} L \\ R \end{bmatrix} \leftarrow \begin{bmatrix} L \\ R \end{bmatrix} - \eta_t \nabla F_t$

$$\begin{bmatrix} L_u^{(t+1)} \\ R_v^{(t+1)} \end{bmatrix} \leftarrow \begin{bmatrix} (1 - \eta_t \lambda_u) L_u^{(t)} - \eta_t \epsilon_t R_v^{(t)} \\ (1 - \eta_t \lambda_v) R_v^{(t)} - \eta_t \epsilon_t L_u^{(t)} \end{bmatrix}$$ ← fast + easy to implement

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:

$$\min_{\theta} \text{rank}(\theta) \quad \theta_{uv} = r_{uv} \quad \forall (u, v, r_{uv}) \in X \text{ s.t. } r_{uv} \neq ?$$

- Which is NP-hard…
  - How do these things relate???

Eigenvalue Decompositions for PSD Matrices

- Given a (square) symmetric positive semidefinite matrix: $\Theta \succeq 0$
  - Eigenvalues: $\lambda_1, \ldots, \lambda_d \geq 0$ $\quad \lambda = (\lambda_1, \ldots, \lambda_d)$
- Thus rank is:

$$|\{\lambda_i : \lambda_i > 0\}| = \text{rank}(\Theta) = \|\lambda\|_0$$

- Approximation:

$$\|\lambda\|_0 \approx \|\lambda\|_1 = \sum_{i=1}^{d} |\lambda_i| \overset{PSD}{=} \sum_{i=1}^{d} \lambda_i$$

(L_1 norm is sum of eigvals)

- Property of trace:

$$\text{trace}(\Theta) = \sum_{i=1}^{d} \lambda_i$$

- Thus, approximate rank minimization by:

$$\min_{\Theta} \text{rank}(\Theta) = \|\lambda\|_0 \quad \text{s.t. } \Theta_{uv} = r_{uv},\ \Theta \succeq 0 \quad \approx \quad \min_{\Theta} \text{trace}(\Theta) \quad \text{s.t. } \Theta_{uv} = r_{uv},\ \Theta \succeq 0$$

Generalizing the Trace Trick

- Non-square matrices ain't got no trace
- For (square) positive semidefinite matrices, matrix factorization: $\lambda_i \geq 0$

$$\Theta = P \Lambda P^{-1} \qquad \Lambda = \text{diag}(\lambda)$$

- For rectangular matrices, singular value decomposition: (SVD)

$$\underset{m \times n}{\Theta} = \underset{m \times r}{U} \; \underset{r \times r}{\Sigma} \; \underset{r \times n}{V'}$$

diagonal matrix w/ entries $\sigma_i(\Theta) \geq 0$

σ_i: i^{th} singular value

- Nuclear norm:

$$\|\Theta\|_* = \sum_{i=1}^{m} \sigma_i(\Theta)$$

$$\min_{\Theta} \|\Theta\|_* \quad \text{s.t. } \Theta_{uv} = r_{uv}$$

convex problem!

Nuclear Norm Minimization

- Optimization problem: (relaxation)

$$\min_{\Theta} \|\Theta\|_* \quad \Theta_{uv} = r_{uv}$$

Θ_{uv} = r_{uv}

can have no feasible soln

- Possible to relax equality constraints: (relaxation of relaxation)

$$\min_{\Theta} \sum_{r_{uv}} (\Theta_{uv} - r_{uv})^2 + \lambda \|\Theta\|_*$$

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

Analysis of Nuclear Norm

- Nuclear norm minimization = convex relaxation of rank minimization:

$$\min_{\Theta} rank(\Theta) \quad \text{NP-hard} \qquad \approx \quad \min_{\Theta} ||\Theta||_* \quad \text{has poly time soln}$$

(convex relaxation)

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

- Theorem [Candes, Recht '08]:
  - If there is a true matrix of rank *k*,
  - And, we observe at least

$$C \; k \; n^{1.2} \log n$$

C: const, k: rank, $n^{1.2}$: assuming $n \geq m$

Original problem has $n \cdot m$ entries

need $\approx k n^{1.2}$ obs

we have $k \cdot n + k \cdot m$ params

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

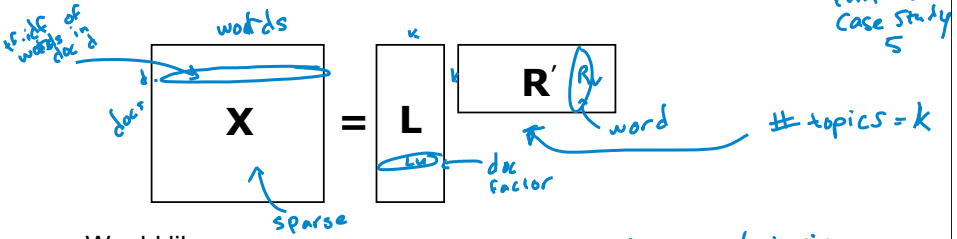
Machine Learning for Big Data
CSE547/STAT548, University of Washington

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):



- Would like:

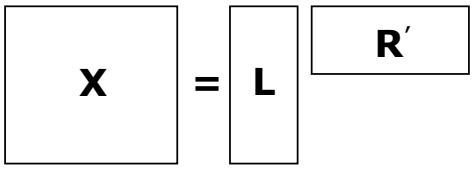
L_u: how much is doc u about each topic

R_v: how much a word v contributes to each topic

- But…

Standard matrix factorization: L_u, R_v can be negative

Nonnegative Matrix Factorization



- Just like before, but

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

non-neg. L, R

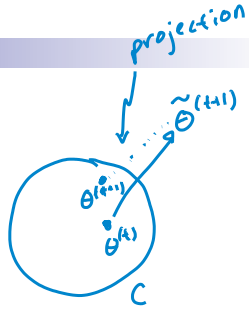
- Constrained optimization problem
  - Many, many, many, many solution methods… we'll check out a simple one

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: $\min_{\Theta} f(\Theta)$ s.t. $\Theta \in \mathcal{C}$



- Projected gradient:
  - Take a gradient step (ignoring constraints):

$$\tilde{\theta}^{(t+1)} \leftarrow \theta^{(t)} - \eta_t \nabla F(\theta^{(t)})$$

  - Projection into feasible set:

$$\Pi_C(\theta) = \underset{\beta \in C}{\operatorname{argmin}} \; \|\theta - \beta\|_2^2$$

often easy to compute (always convex)

$$\theta^{(t+1)} = \Pi_C(\tilde{\theta}^{(t+1)})$$

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: $L_u \geq 0 \quad R_v \geq 0 \quad \forall u, v$
- Projection step: $\Pi_C(\theta) = \underset{\beta \in C}{\text{argmin}} \|\theta - \beta\|_2^2$ ← totally ind. problems per dimension

Single dimension:

$$= \underset{\beta \geq 0}{\text{argmin}} (\theta - \beta)^2$$

$$= \begin{cases} \theta & \text{if } \theta \geq 0 \\ 0 & \text{if } \theta < 0 \end{cases} = (\theta)_+$$

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

set all neg. coords to zero

easy!!!

(phew)

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

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**

For user u', predict $r_{u'v} \approx w \cdot \phi(v)$

- Clearly, not all users are the same ... shared w is strong assumption
- Just as in personalized click prediction, consider model with global and user-specific parameters

Consider user-specific deviations w_u from the crowd w

$r_{uv} \approx (w + w_u) \cdot \phi(v)$ (init. to 0)

- As we gain more information about the user, forget the crowd

w_u more informed

User Features…

- In addition to movie features, may have information about the user:

$\phi(u) = (25, F, MSc, A^+, \ldots)$
age, gender, education, grade in Big Data

- Combine with features of movie:

$\phi(u,v) = (\ldots, \phi(u), \ldots, \ldots, \phi(v), \ldots, \ldots$ cross features...$)$

- Unified linear model:

$r_{uv} \approx (w + w_u) \cdot \phi(u,v)$

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