Case Study 4: Collaborative Filtering

Collaborative Filtering
Matrix Completion
Alternating Least Squares

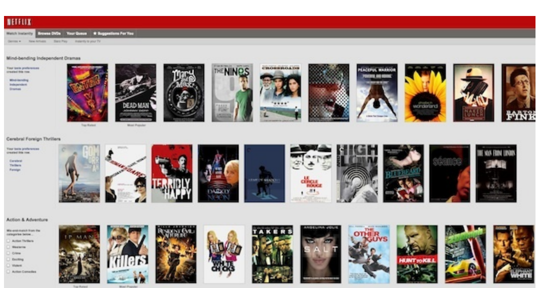
Machine Learning/Statistics for Big Data
CSE599C1/STAT592, University of Washington

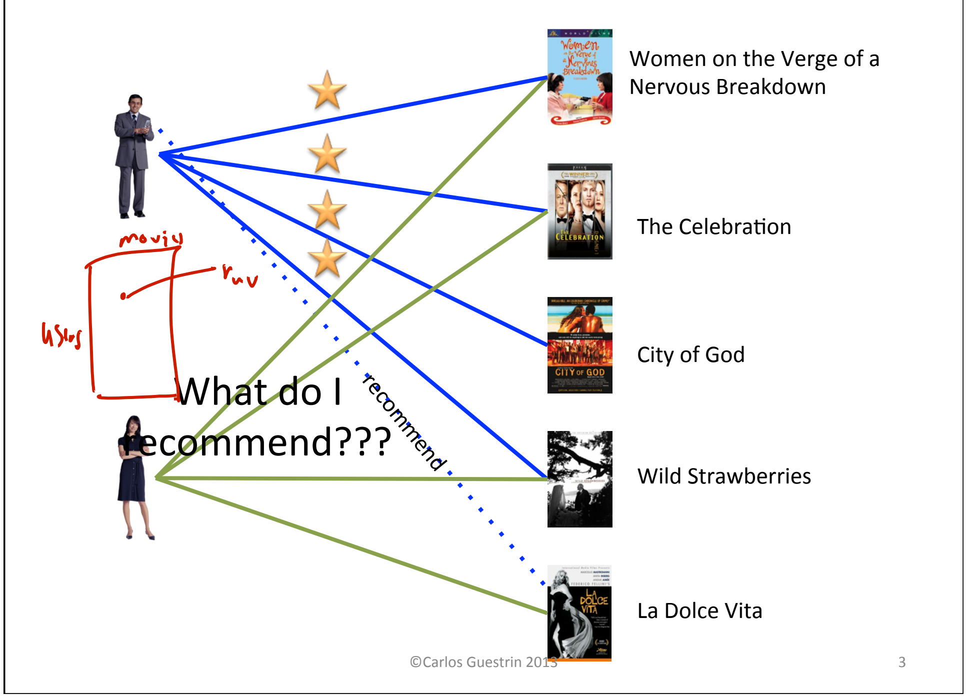
Carlos Guestrin

February 28th, 2013

Collaborative Filtering

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



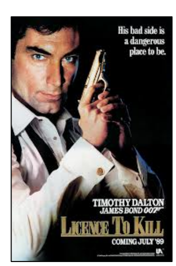


Cold-Start Problem

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

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

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



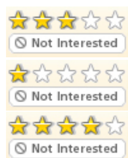
Netflix Prize

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

Skyfall

WGNB

FRWL



PoC

movies

users

- 17770 total movies
- 480189 total users
- Over 8 billion total ratings

8B params

& 100M obs

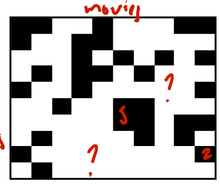
- How to fill in the blanks?

Figures from Ben Recht

Matrix Completion Problem

movies

X =



users

X_{ij} known for black cells

X_{ij} unknown for white cells

Rows index users

Columns index movies

- Filling missing data?

well approx by a rank K matrix

$$\underset{n \times m}{X} \approx \underset{n \times K}{L} \; \underset{K \times m}{R'}$$

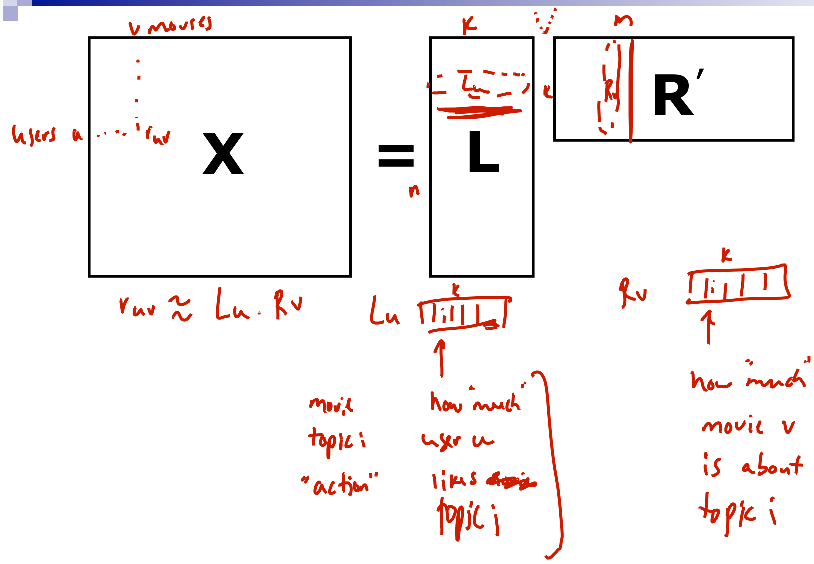
learn this

predict this

n.m params

$K.n + K.m \ll n.m$

Interpreting Low-Rank Matrix Completion (aka Matrix Factorization)



Matrix Completion via Rank Minimization

- Given observed values: $(u, v, r_{uv}) \in X$ Some $r_{uv} = ?$
- Find matrix Θ ($n \times m$)
- Such that: $\Theta_{uv} = r_{uv}$ $\forall r_{uv} \neq ?$

  fit $r_{uv} \neq ?$ perfectly
- But…
- Introduce bias: Low rank –

  $$\min_{\Theta} \ \text{rank}(\Theta)$$
  $$\Theta_{uv} = r_{uv} \quad \forall r_{uv} \neq ?$$
- Two issues: NP-hard; you can't hope to get exact matching

Approximate Matrix Completion

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

$$\min_{\Theta} \sum_{r_{uv} \in X : r_{uv} \neq ?} (\Theta_{uv} - r_{uv})^2$$

- Choose rank *k:*

$${}_{n}\overset{m}{\Theta} = {}_{n}\overset{k}{L}\ {}_{m}\overset{k}{R}'$$

- 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

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

- Fix movie factors R, optimize for user factors L
- First Observation:

$V_u \equiv$ set movies user u rated

$$\min_{L_1,\ldots,L_n} \sum_{(u,v,r_{uv})} (L_u \cdot R_v - r_{uv})^2 \equiv$$

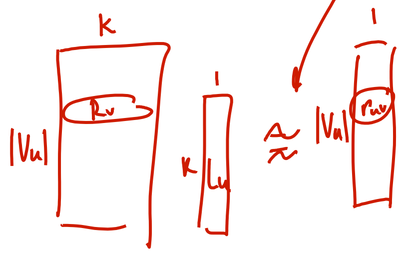
$$\min_{L_1,\ldots,L_n} \sum_{u} \sum_{v \in V_u} (L_u \cdot R_v - r_{uv})^2 \equiv \leftarrow \text{indep opt problem per user}$$

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

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:

least squares sense



- Second observation: Solve by
  - matrix inversion
  - gradient
  - ...

Coordinate Descent for Matrix Factorization: Alternating Least-Squares

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

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

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

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

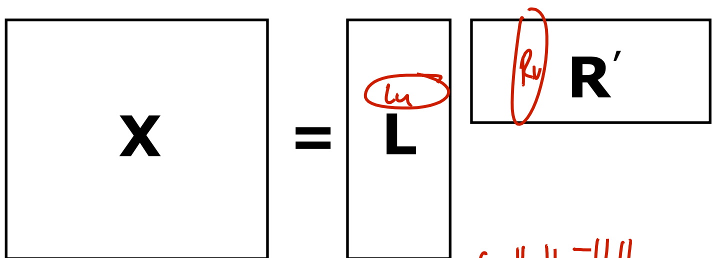
$$\min_{R_v} \sum_{u \in U_v} (L_u \cdot R_v - r_{uv})^2 + \lambda_v \|R_v\|$$

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

Effect of Regularization

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

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



if $\|\cdot\| = \|\cdot\|_F^2$
each sub problem 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/Statistics for Big Data
CSE599C1/STAT592, University of Washington
Carlos Guestrin
March 7th, 2013

Stochastic Gradient Descent

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

$\sum_u L_u \cdot L_u$

- Observe one rating at a time r_{uv}^t $\varepsilon_t = L_u^{(t)} \cdot R_v^{(t)} - r_{uv}$
- Gradient observing r_{uv}:

$$\frac{\partial F}{\partial L_u} = \varepsilon_t R_v + \lambda_u L_u$$

$$\frac{\partial F}{\partial R_v} = \varepsilon_t L_u + \lambda_v R_v$$

$$\nabla F_t = \begin{bmatrix} \varepsilon_t R_v + \lambda_u L_u \\ \varepsilon_t L_u + \lambda_v R_v \end{bmatrix}$$

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

$$\begin{bmatrix} L_u^{(t+1)} \\ R_v^{(t+1)} \end{bmatrix} \leftarrow \begin{bmatrix} (1-\eta_t \lambda_u) L_u^{(t)} - \eta_t \varepsilon_t R_v^{(t)} \\ (1-\eta_t \lambda_v) R_v^{(t)} - \eta_t \varepsilon_t L_u^{(t)} \end{bmatrix}$$

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:

$$\min_{\theta} \ \text{rank}(\theta)$$
$$\theta_{uv} = r_{uv} \qquad \forall r_{uv} \in X, \ r_{uv} \neq ?$$

- 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: $\lambda_1, \ldots, \lambda_d \geq 0 \qquad \lambda = (\lambda_1, \ldots, \lambda_d)$
- 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 eigen values

- Property of trace:

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

- Thus, approximate rank minimization by:

$$\min_{\theta} \ \text{trace}(\theta)$$
$$\theta_{uv} = r_{uv}$$
$$\theta \succeq 0$$

Generalizing the Trace Trick

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

$$d\overset{d}{\Theta} = P \underbrace{\Lambda}_{\mathrm{diag}(\lambda)} P^{-1}$$

- For rectangular matrices, singular value decomposition: e.g. $n \geq m$

$$\begin{array}{c} m \\ n \end{array}\boxed{\Theta} = \begin{array}{c} n \\ n \end{array}\boxed{U} \begin{array}{c} m \\ n \end{array}\boxed{\Sigma} \begin{array}{c} m \\ m \end{array}\boxed{V'}$$

diagonal matrix

entries $\sigma_i(\Theta) \geq 0$ ith singular value

- Nuclear norm:

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

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

convex problem

Nuclear Norm Minimization

- Optimization problem: $\min_{\Theta} \|\Theta\|_*$

$$\Theta_{uv} = r_{uv}$$

- Possible to relax equality constraints:

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

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

Analysis of Nuclear Norm

- Nuclear norm minimization is a convex relaxation of rank minimization problem:

NP-hard

$$\min_{\Theta} rank(\Theta)$$

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

convex relaxation with a poly time solution

$\approx$

$$\min_{\Theta} ||\Theta||_*$$

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

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

original problem $n.m$ entries

rank

need about $kn^{1.2}$

constant

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

we have $kn+km$ params

$n \geq m$

random entries of true matrix

  - Then true matrix is recovered exactly with high probability with convex nuclear norm minimization!
    - Under certain conditions

Nuclear Norm Minimization versus Direct (Bilinear) Low Rank Solutions

- Nuclear norm minimization:

Θ^*: args

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

convex, global opt = close to truth

  - Annoying because: – Θ very large (8B entries in Netflix)
    – SDP solvers are very slow (but poly time)

- Instead:

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

$\approx$10-100M params, solvers very fast

  - Annoying because: – many local optima
  - But $||\Theta||_* = \inf \left\{ \min_{L,R} \frac{1}{2}||L||_F^2 + \frac{1}{2}||R||_F^2 : \Theta = LR' \right\}$
    - So second prob. non convex approx. to first
    - And if pick rank of L&R to be slightly higher rank(Θ^*), local optima of (**) are global optima of (*)
    - 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/Statistics for Big Data
CSE599C1/STAT592, University of Washington
Carlos Guestrin
March 7th, 2013

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):

TFIDF of words in docu

words

k

words

docs

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

R_v $\quad$ k $\quad$ docs $\quad$ L_u

- Would like: L_u: how much a doc is about each topic

  R_v: how much a word contributes to each topic

- But…

  Standard matrix factorization: L_u, R_v can be negative

Nonnegative Matrix Factorization

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

- Just like before, but

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

nonnegative

L, R

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

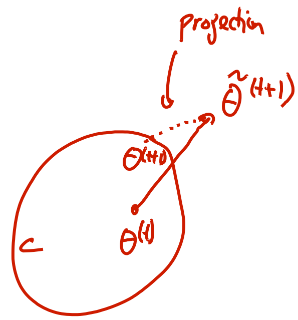
Projected Gradient

- Standard optimization:
  - Want to minimize: $\underline{\min_{\Theta} f(\Theta)}$ e.g.
  - 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):

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

  - Projection into feasible set:

$$\Pi_C(\Theta) \equiv \arg\min_{\beta \in C} \|\Theta - \beta\|_2^2$$ often easy to compute (always convex)

$$\Theta^{(t+1)} = \Pi_C\left(\tilde{\Theta}^{(t+1)}\right)$$

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: $L_u \geq 0$; $R_v \geq 0$ $\forall u, v$
- Projection step:

$$\Pi_C(\theta) = \arg\min_{\beta \in C} \|\theta - \beta\|_2^2$$ ← totally indep prob per dimension

Single dim:

$$= \arg\min_{\beta \geq 0} (\theta - \beta)^2$$

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

$$\begin{bmatrix} L^{(t+1)} \\ R^{(t+1)} \end{bmatrix} \leftarrow \begin{bmatrix} \tilde{L}^{(t+1)} \\ \tilde{R}^{(t+1)} \end{bmatrix}_+$$

set all neg coords to zero, universe on our side!! :)

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