Generalized Low Rank Models

Madeleine Udell

Center for the Mathematics of Information
Caltech

Based on joint work with Stephen Boyd, Corinne Horn, Reza Zadeh, and Nathan Kallus

INFORMS, 11/2/2015

Data table

age	gender	state	income	education	$\cdots$
29	F	CT	\$53,000	college	$\cdots$
57	?	NY	\$19,000	high school	$\cdots$
?	M	CA	\$102,000	masters	$\cdots$
41	F	NV	\$23,000	?	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	

- detect demographic groups?
- find typical responses?
- identify similar states?
- impute missing entries?

Data table

m examples (patients, respondents, households, assets)
n features (tests, questions, sensors, times)

$$\begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix} = \begin{bmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{m1} & \cdots & A_{mn} \end{bmatrix}$$

- ith row of A is feature vector for ith example
- jth column of A gives values for jth feature across all examples

Low rank model

given: $A \in \mathbf{R}^{m\times n}$, $k \ll m, n$
find: $X \in \mathbf{R}^{m\times k}$, $Y \in \mathbf{R}^{k\times n}$ for which

$$\begin{bmatrix} \\ X \\ \\ \end{bmatrix} \begin{bmatrix} & Y & \end{bmatrix} \approx \begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix}$$

i.e., $x_i y_j \approx A_{ij}$, where

$$\begin{bmatrix} \\ X \\ \\ \end{bmatrix} = \begin{bmatrix} \text{—}x_1\text{—} \\ \vdots \\ \text{—}x_m\text{—} \end{bmatrix} \qquad \begin{bmatrix} & Y & \end{bmatrix} = \begin{bmatrix} | & & | \\ y_1 & \cdots & y_n \\ | & & | \end{bmatrix}$$

interpretation:

- X and Y are (compressed) representation of A
- $x_i^T \in \mathbf{R}^k$ is a point associated with example i
- $y_j \in \mathbf{R}^k$ is a point associated with feature j
- inner product $x_i y_j$ approximates A_{ij}

Why use a low rank model?

- reduce storage; speed transmission
- understand (visualize, cluster)
- remove noise
- infer missing data
- simplify data processing

Outline

PCA

Generalized low rank models

Applications

Algorithms

Principal components analysis

PCA:

$$\text{minimize} \quad \|A - XY\|_F^2 = \sum_{i=1}^m \sum_{j=1}^n (A_{ij} - x_i y_j)^2$$

with variables $X \in \mathbf{R}^{m \times k}$, $Y \in \mathbf{R}^{k \times n}$

- old roots [Pearson 1901, Hotelling 1933]
- least squares low rank fitting
- (analytical) solution via SVD of $A = U \Sigma V^T$:

  $$X = U_k \Sigma_k^{1/2} \quad Y = \Sigma_k^{1/2} V_k^T$$

  (Not unique: (XT, TY) also a solution for T invertible.)
- (numerical) solution via alternating minimization

Outline

PCA

Generalized low rank models

Applications

Algorithms

Generalized low rank model

$$\text{minimize} \quad \sum_{(i,j)\in\Omega} L_j(x_i y_j, A_{ij}) + \sum_{i=1}^m r_i(x_i) + \sum_{j=1}^n \tilde{r}_j(y_j)$$

- loss functions L_j for each column
  - *e.g.*, different losses for reals, booleans, categoricals, ordinals, ...
- regularizers $r : \mathbf{R}^{1\times k} \to \mathbf{R}$, $\tilde{r} : \mathbf{R}^k \to \mathbf{R}$
- observe only $(i,j) \in \Omega$ (other entries are missing)

Note: can be NP-hard to optimize exactly...

Related work

- principal components analysis (PCA) [Pearson 1901, Hotelling 1933]
- exponential family PCA [Collins 2001]
- generalized^2 linear^2 models [Gordon 2002]
- convex relaxations of regularization [Srebro 2004]
- matrix factorization as clustering [Tropp 2004]
- matrix factorization models [Singh Gordon 2008]
- penalized matrix decomposition [Witten et al. 2009]
- low rank approximation [Markovsky 2012]

Matrix completion

observe A_{ij} only for $(i,j) \in \Omega \subset \{1,\ldots,m\} \times \{1,\ldots,n\}$

$$\text{minimize} \quad \sum_{(i,j)\in\Omega}(A_{ij} - x_i y_j)^2 + \sum_{i=1}^m \|x_i\|_2^2 + \sum_{j=1}^n \|y_j\|_2^2$$

two regimes:

- **some entries missing:** don't waste data; "borrow strength" from entries that are *not* missing
- **most entries missing:** matrix completion still works!

Theorem ([Keshavan Montanari 2010])

If A has rank $k' \leq k$ and $|\Omega| = O(nk' \log n)$ (and A is incoherent and Ω is chosen UAR), then matrix completion exactly recovers the matrix A with high probability.

Maximum likelihood low rank estimation

noisy data? maximize (log) likelihood of observations by minimizing:

- gaussian noise: $L(u, a) = (u - a)^2$
- laplacian (heavy-tailed) noise: $L(u, a) = |u - a|$
- gaussian + laplacian noise: $L(u, a) = \mathbf{huber}(u - a)$
- poisson (count) noise: $L(u, a) = \exp(u) - au + a \log a - a$
- bernoulli (coin toss) noise: $L(u, a) = \log(1 + \exp(-au))$

Maximum likelihood low rank estimation works

Theorem (Template)

If a number of samples $|\Omega| = O(n\log(n))$ *drawn UAR from matrix entries is observed according to a probabilistic model with parameter* Z*, the solution to (appropriately) regularized maximum likelihood estimation is close to the true* Z *with high probability.*

examples (not exhaustive!):

- additive gaussian noise [Candes Plan 2009]
- additive subgaussian noise [Keshavan Montanari Oh 2009]
- gaussian $+$ laplacian noise [Xu Caramanis Sanghavi 2012]
- 0-1 (Bernoulli) observations [Davenport et al. 2012]
- entrywise exponential family distribution [Gunasekar Ravikumar Ghosh 2014]
- multinomial logit [Kallus U 2015]

Abstract loss

define *abstract feature space* $\mathcal{F}_j$

e.g., $A_{ij} \in \mathcal{F}_j$ can be

- boolean
- ordinal
- categorical
- ranking

just need a loss function $L_j : \mathbf{R} \times \mathcal{F}_j \to \mathbf{R}$.

Regularizers

$$\text{minimize} \quad \sum_{(i,j)\in\Omega} L_j(x_i y_j, A_{ij}) + \sum_{i=1}^m r_i(x_i) + \sum_{j=1}^n \tilde{r}_j(y_j)$$

Choose regularizers r, $\tilde{r}$ to impose structure on representation

- small
  - $r(x) = \|x\|_2^2$
- sparse
  - $r(x) = \|x\|_1$
  - $r(x) = I(\mathbf{card}(x) \leq p)$
- nonnegative
  - $r(x) = \delta(x \geq 0)$
- clustered
  - $r(x) = \delta(\mathbf{card}(x) = 1)$, $\tilde{r}(y) = 0$

Outline

PCA

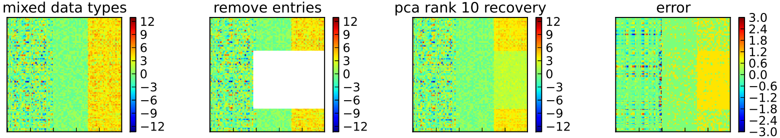
Generalized low rank models

Applications

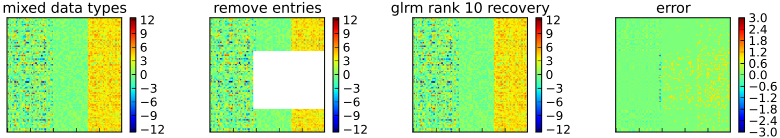
Algorithms

Impute heterogeneous data

PCA:



GLRM:



American community survey

2013 ACS:

- 3M respondents, 87 economic/demographic survey questions
  - income
  - cost of utilities (water, gas, electric)
  - weeks worked per year
  - hours worked per week
  - home ownership
  - looking for work
  - use foodstamps
  - education level
  - state of residence
  - ...
- 1/3 of responses missing

Fitting a GLRM to the ACS

- construct a rank 10 GLRM with loss functions respecting data types
  - huber for real values
  - hinge loss for booleans
  - ordinal hinge loss for ordinals
  - one-vs-all hinge loss for categoricals
- scale losses and regularizers by $1/\sigma_j^2$
- fit the GLRM

in 3 lines of code:

```
A = expand_categoricals(A, categoricals)
glrm, labels = GLRM(A, 10, scale = true)
X,Y = fit!(glrm)
```

American community survey

most similar features (in *demography space*):

- Alaska: Montana, North Dakota
- California: Illinois, cost of water
- Colorado: Oregon, Idaho
- Ohio: Indiana, Michigan
- Pennsylvania: Massachusetts, New Jersey
- Virginia: Maryland, Connecticut
- Hours worked: weeks worked, education

Outline

PCA

Generalized low rank models

Applications

Algorithms

Convergence theory for GLRMs

can we fit GLRMs?

- exactly, always: **no**
  - NP hard to solve weighted PCA [Gillis2011]
- exactly, sometimes: **yes**
  - some GLRMs are equivalent to convex problems
- approximately (heuristically), always: **yes**
  - alternating minimization never increases the objective value

Fitting GLRMs with alternating minimization

$$\text{minimize} \quad \sum_{(i,j)\in\Omega} L_j(x_i y_j, A_{ij}) + \sum_{i=1}^m r_i(x_i) + \sum_{j=1}^n \tilde{r}_j(y_j)$$

repeat:

1. minimize objective over x_i (in parallel)
2. minimize objective over y_j (in parallel)

properties:

- subproblems easy to solve
- objective decreases at every step, so converges if losses and regularizers are bounded below
- (not guaranteed to find global solution, but) usually finds good model in practice
- naturally parallel, so scales to *huge* problems

Alternating updates

given X^0, Y^0
for $t = 1, 2, \ldots$ **do**
  for $i = 1, \ldots, m$ **do**
    $x_i^t = \mathbf{update}_{L,r}(x_i^{t-1}, Y^{t-1}, A)$
  end for
  for $j = 1, \ldots, n$ **do**
    $y_j^t = \mathbf{update}_{L,\tilde{r}}(y_j^{(t-1)T}, X^{(t)T}, A^T)$
  end for
end for

- no need to exactly minimize
- choose fast, simple update rules

A simple, fast update rule

proximal gradient method: let

$$g = \sum_{j:(i,j)\in\Omega} \nabla L_j(x_i y_j, A_{ij}) y_j$$

and update

$$x_i^{t+1} = \mathbf{prox}_{\alpha_t r}(x_i^t - \alpha_t g)$$

(where $\mathbf{prox}_f(z) = \operatorname{argmin}_x (f(x) + \frac{1}{2}\|x - z\|_2^2)$)

- **simple:** only requires ability to evaluate ∇L and $\mathbf{prox}_r$
- **stochastic variant:** use noisy estimate for g
- **time per iteration:** $O(\frac{(n+m+|\Omega|)k}{p})$ on p processors

Exactly, sometimes

Theorem

$(X, Y) \in \mathbf{R}^{m\times k} \times \mathbf{R}^{k\times n}$ is a solution to

$$\text{minimize} \quad F(XY) + \frac{\gamma}{2}\|X\|_F^2 + \frac{\gamma}{2}\|Y\|_F^2 \tag{$\mathcal{F}$}$$

if and only if $Z = XY$ is a solution to

$$\begin{array}{ll} \text{minimize} & F(Z) + \gamma\|Z\|_* \\ \text{subject to} & \mathbf{Rank}(Z) \le k, \end{array} \tag{$\mathcal{R}$}$$

where $\|Z\|_*$ is the sum of the singular values of Z.

- if F is convex, then $\mathcal{R}$ is a rank-constrained semidefinite program
- local minima of $\mathcal{F}$ correspond to local minima of $\mathcal{R}$

Proof of equivalence

suppose $Z = XY = U\Sigma V^T$

- $\mathcal{F} \leq \mathcal{R}$: if Z is feasible for $\mathcal{R}$, then

$$X = U\Sigma^{1/2}, \qquad Y = \Sigma^{1/2}V^T$$

  is feasible for $\mathcal{F}$, with the same objective value

- $\mathcal{R} \leq \mathcal{F}$: for any $XY = Z$,

$$\begin{aligned} \|Z\|_* &= \mathbf{tr}(\Sigma) \\ &= \mathbf{tr}(U^T XYV) \\ &\leq \|U^T X\|_F \|YV\|_F \\ &\leq \|X\|_F \|Y\|_F \\ &\leq \frac{1}{2}(||X||_F^2 + ||Y||_F^2) \end{aligned}$$

Convex equivalence

Theorem

For every $\gamma \geq \gamma^\star(k)$, every solution to

$$\begin{array}{ll} \textit{minimize} & L(Z) + \gamma\|Z\|_* \\ \textit{subject to} & \mathbf{Rank}(Z) \leq k \end{array} \qquad (\mathcal{R})$$

(with variable $Z \in \mathbf{R}^{m\times n}$) is a solution to

$$\textit{minimize} \quad L(Z) + \gamma\|Z\|_*. \qquad (\mathcal{U})$$

proof: find $\gamma^\star(k)$ so large that there is a Z with rank $\leq k$ satisfying optimality conditions for $\mathcal{U}$

- if γ is sufficiently large (compared to k), rank constraint is *not binding*

Certify global optimality, sometimes

two ways to use convex equivalence:

- **convex:**
  1. solve the unconstrained SDP

$$\text{minimize} \quad F(Z) + \gamma \|Z\|_*$$

  2. see if the solution is low rank
- **nonconvex:**
  1. fit the GLRM with any method, producing (X, Y)
  2. check if $XY = U\Sigma V^T$ satisfies the optimality conditions for the (convex) unconstrained SDP

$$\|\partial F(XY) + \gamma UV^T\|_2 \leq 1$$

Why use the factored formulation?

pro

- size of problem variable: $(m+n)k$ vs mn
- smooth regularizer: frobenius vs trace norm
- no eigenvalue computations needed
- (almost) no new local minima if k is large enough
  - solution to rank-constrained SDP is in the relative interior of a face over which the objective is constant [Burer Monteiro]
- linear convergence of gradient descent to local minimum if loss is differentiable and strongly convex on the set of rank-k matrices [Bhojanapalli Kyrillidis Sanghavi 2015]

con

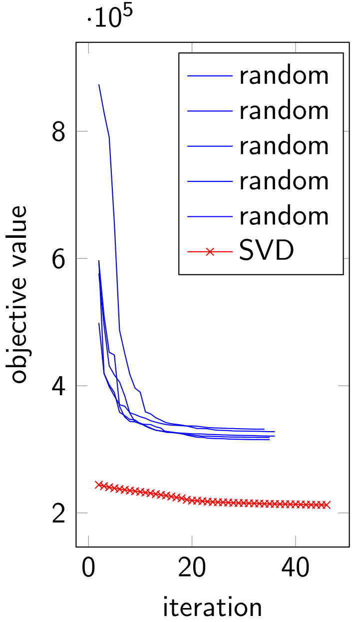
- local minima
- saddle points

Initialization

- fit census data set
- random initialization

$$x_i \sim \mathcal{N}(0, I_k)$$
$$y_j \sim \mathcal{N}(0, I_k)$$

- SVD initialization
  - interpret A as numerical matrix M
  - fill in missing entries in M to preserve column mean and variance
  - center and standardize M
  - initialize XY with SVD of M



Summary

- a general framework for fitting tabular data
  - losses for abstract data types
  - automatic scaling for heterogeneous losses
- a more general algorithm
  - parallel algorithms for (heuristically) fitting *any* GLRM
  - software package(s) implementing framework
  - heuristic initialization rules
- new analytic tools
  - model validation
  - certificates of optimality (sometimes)

paper

`http://arxiv.org/abs/1410.0342`

code

`https://github.com/madeleineudell/LowRankModels.jl`