Generalized Low Rank Models

Madeleine Udell

Cornell ORIE admit visit day 3/18/2016

Data table

age	gender	state	diabetes	education	⋯
22	F	CT	?	college	⋯
57	?	NY	severe	high school	⋯
?	M	CA	moderate	masters	⋯
41	F	NV	none	?	⋯
⋮	⋮	⋮	⋮	⋮	

- detect demographic groups?
- find typical responses?
- identify related features?
- 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, $k \ll m, n$
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

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 $A = U\Sigma V^T$:

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

- (numerical) solution via alternating minimization

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)

Losses

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

choose loss $L(u, a)$ adapted to data type:

data type	**loss**	$L(u, a)$
real	quadratic	$(u-a)^2$
real	absolute value	$\|u-a\|$
real	huber	$\mathbf{huber}(u-a)$
boolean	hinge	$(1-ua)_+$
boolean	logistic	$\log(1+\exp(-au))$
integer	poisson	$\exp(u) - au + a\log a - a$
ordinal	ordinal hinge	$\sum_{a'=1}^{a-1}(1-u+a')_+ + \sum_{a'=a+1}^{d}(1+u-a')_+$
categorical	one-vs-all	$(1-u_a)_+ + \sum_{a'\neq a}(1+u_{a'})_+$
categorical	multinomial logit	$\frac{\exp(u_a)}{(\sum_{a'=1}^{d}\exp(u_{a'}))}$

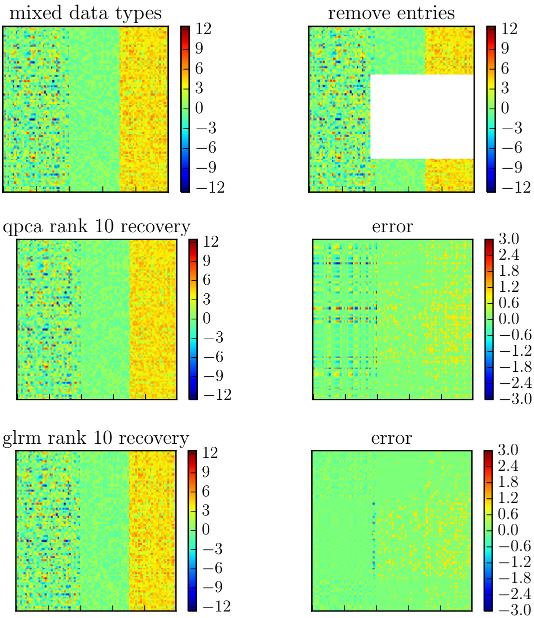
Regularizers

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

choose regularizers r, $\tilde{r}$ to impose structure:

structure	$r(x)$	$\tilde{r}(y)$
small	$\|x\|_2^2$	$\|y\|_2^2$
sparse	$\|x\|_1$	$\|y\|_1$
nonnegative	$\mathbf{1}(x \geq 0)$	$\mathbf{1}(y \geq 0)$
clustered	$\mathbf{1}(\mathbf{card}(x) = 1)$	0

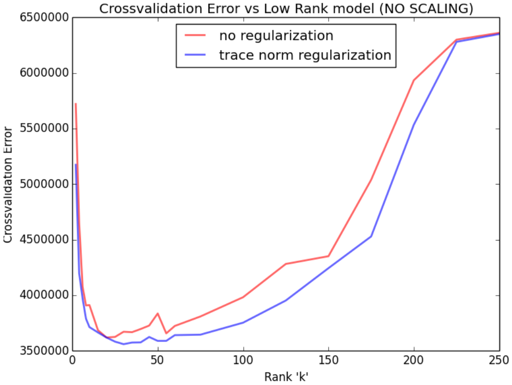
Impute heterogeneous data



US politics are low rank [Sengupta U Evans, in prep]

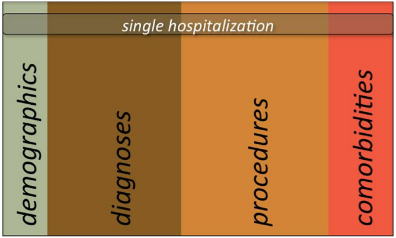
General Social Survey (GSS):

- survey adults in randomly selected US households about attitudes and demographics
- $>$ 33% missing data

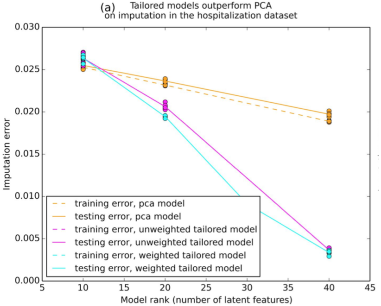


Hospitalizations are low rank [Schuler et al., 2016]

hospitalization data set



GLRM outperforms PCA



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

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

Proximal operator

define the *proximal operator*

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

- **generalized projection:** if $\mathbf{1}_C$ is the indicator function of a set C, then
$$\mathbf{prox}_{\mathbf{1}_C}(z) = \Pi_C(z)$$
- **implicit gradient step:** if $x = \mathbf{prox}_f(z)$, then
$$\begin{aligned} \nabla f(x) + x - z &= 0 \\ x &= z - \nabla f(x) \end{aligned}$$
- **simple to evaluate:** closed form solutions for
  - $f = \|\cdot\|_2^2$
  - $f = \|\cdot\|_1$
  - $f = \mathbf{1}_+$
  - ...

more info: [Parikh Boyd 2013]

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

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

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

Implementations

Implementations in Python (serial), Julia (shared memory parallel), Spark (parallel distributed), and H2O (parallel distributed).

example: (Julia) forms and fits a k-means model with $k = 5$

```
losses = QuadLoss()                # minimize squared error
rx = UnitOneSparseConstraint() # one cluster per row
ry = ZeroReg()                     # free cluster centroids
glrm = GLRM(A,losses,rx,ry,k)  # form model
fit!(glrm)                         # fit model
```

When is a low rank model an SDP?

Theorem

(X, Y) is a solution to

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

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

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

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 $\mathcal{R}$

Dynamic low rank models (with Nathan Kallus)

for $t = 1, \ldots, T$,

- customer $i_t \in \{1, \ldots, m\}$ arrives
- store presents item j_t
- customer takes action (and store observes) $x_{i_t} y_{j_t} + \epsilon_t$ (buys/rates item j)
- store receives utility $r_t = x_{i_t} y_{j_t} + \epsilon_t$

choose j_t to maximize utility $\sum_{t=1}^{T} r_t$?

There's more to do!

theory

- fast algorithms for large-scale low-rank SDPs
- dynamics for fast learning (and profit) (Nathan Kallus)
- asynchronous parallel algorithms for GLRMs (Damek Davis)
- statistical inference and consistency

applications

- medical diagnostics
- social science
- low energy sensing and data processing
- photovoltaic array design

extensions

- using timeseries and graph structure
- learning across data sets