Generalized Low Rank Models

Madeleine Udell

Center for the Mathematics of Information
Caltech

Based on joint work with Stephen Boyd, Anqi Fu, Corinne Horn, and Reza Zadeh

H2O World 11/11/2015

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

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: $(XT, T^{-1}Y)$ also a solution for T invertible.)
- (numerical) solution via alternating minimization

Low rank models for gait analysis [1]

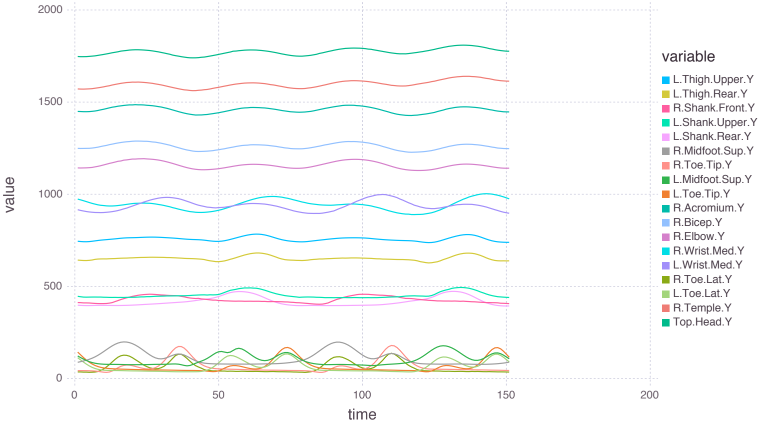
time	forehead (x)	forehead (y)	$\cdots$	right toe (y)	right toe (z)
t_1	1.4	2.7	$\cdots$	-0.5	-0.1
t_2	2.7	3.5	$\cdots$	1.3	0.9
t_3	3.3	-.9	$\cdots$	4.2	1.8
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

- rows of Y are principal stances
- rows of X decompose stance into combination of principal stances

[1] gait analysis demo: `https://github.com/h2oai/h2o-3/blob/master/h2o-r/demos/rdemo.glrm.walking.gait.R`

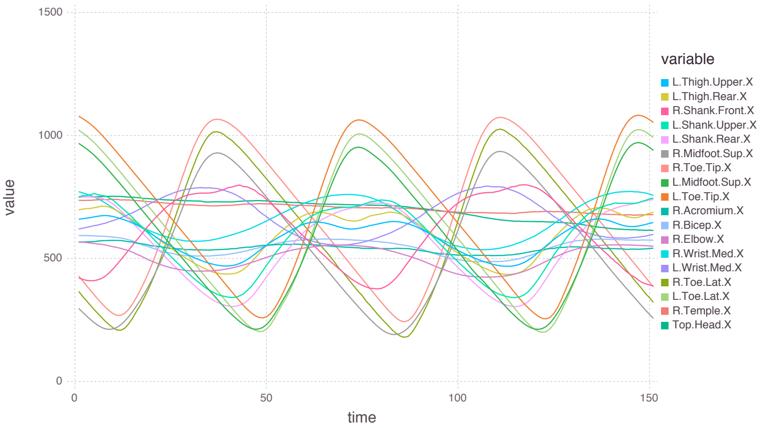
Interpreting principal components

columns of A (features) (y coordinates over time)



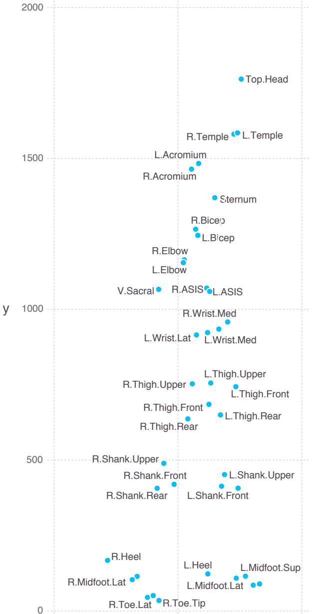
Interpreting principal components

columns of A (features) (z coordinates over time)



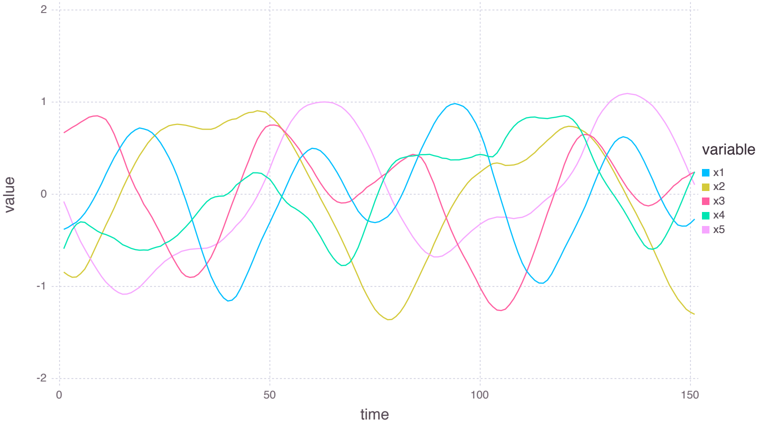
Interpreting principal components

row of Y
(archetypical example)
(principal stance)



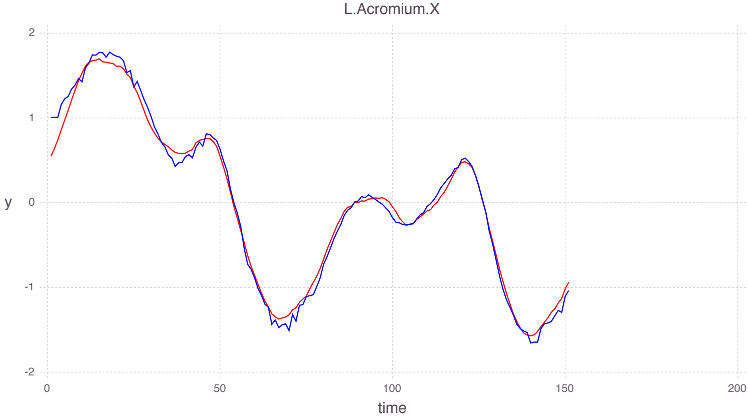
Interpreting principal components

columns of X (archetypical features) (principal timeseries)



Interpreting principal components

column of XY (red) (predicted feature)
column of A (blue) (observed feature)



Generalized low rank model

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

- 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)

Matrix completion

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

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

two regimes:

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

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

Losses

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

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

data type	**loss**	$L(u,a)$
real	quadratic	$(u-a)^2$
real	absolute value	$\lvert u-a \rvert$
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	$(1-u_a)_+ + \sum_{a'\neq a}(1+u_{a'})_+$

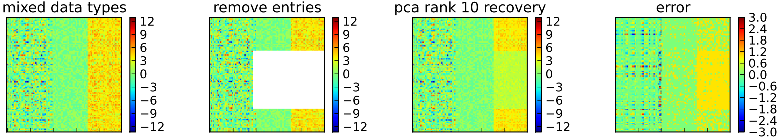
Examples

variations on GLRMs recover many known models:

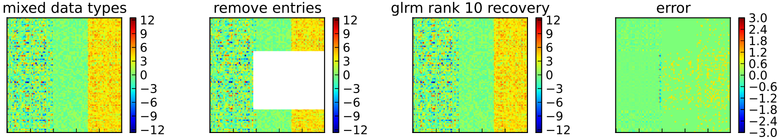
Model	$\mathbf{L_j(u, a)}$	$\mathbf{r(x)}$	$\mathbf{\tilde{r}(y)}$	**reference**
PCA	$(u-a)^2$	0	0	[Pearson 1901]
matrix completion	$(u-a)^2$	$\|x\|_2^2$	$\|y\|_2^2$	[Keshavan 2010]
NNMF	$(u-a)^2$	$\mathbf{1}(x \geq 0)$	$\mathbf{1}(y \geq 0)$	[Lee 1999]
sparse PCA	$(u-a)^2$	$\|x\|_1$	$\|y\|_1$	[D'Aspremont 20
sparse coding	$(u-a)^2$	$\|x\|_1$	$\|y\|_2^2$	[Olshausen 1997]
k-means	$(u-a)^2$	$\mathbf{1}(\mathbf{card}(x) = 1)$	0	[Tropp 2004]
robust PCA	$\|u-a\|$	$\|x\|_2^2$	$\|y\|_2^2$	[Candes 2011]
logistic PCA	$\log(1+\exp(-au))$	$\|x\|_2^2$	$\|y\|_2^2$	[Collins 2001]
boolean PCA	$(1-au)_+$	$\|x\|_2^2$	$\|y\|_2^2$	[Srebro 2004]

Impute heterogeneous data

PCA:



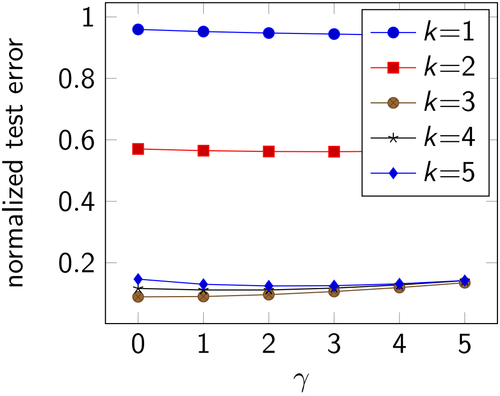
GLRM:



Validate model

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

How to choose model parameters (k, γ)?
Leave out 10% of entries, and use model to predict them



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

Using a GLRM for exploratory data analysis

$$\begin{bmatrix} | & & | \\ y_1 & \cdots & y_n \\ | & & | \end{bmatrix}$$

age	gender	state	$\cdots$
29	F	CT	$\cdots$
57	?	NY	$\cdots$
?	M	CA	$\cdots$
41	F	NV	$\cdots$
$\vdots$	$\vdots$	$\vdots$	

$$\approx \begin{bmatrix} —x_1— \\ \vdots \\ —x_m— \end{bmatrix}$$

- cluster respondents? **cluster rows of X**
- demographic profiles? **rows of Y**
- which features are similar? **cluster columns of Y**
- impute missing entries? $\mathrm{argmin}_a\, L_j(x_i y_j, a)$

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
- fit the 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

Low rank models for dimensionality reduction [2]

U.S. Wage & Hour Division (WHD) compliance actions:

company	# employees	zip	violations	$\cdots$
h2o.ai	58	95050	0	$\cdots$
stanford	8300	94305	0	$\cdots$
caltech	741	91107	0	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	

- 208,806 rows (cases) $\times$ 252 columns (violation info)
- 32,989 zip codes...

[2] labor law violation demo: `https://github.com/h2oai/h2o-3/blob/master/h2o-r/demos/rdemo.census.labor.violations.large.R`

Low rank models for dimensionality reduction

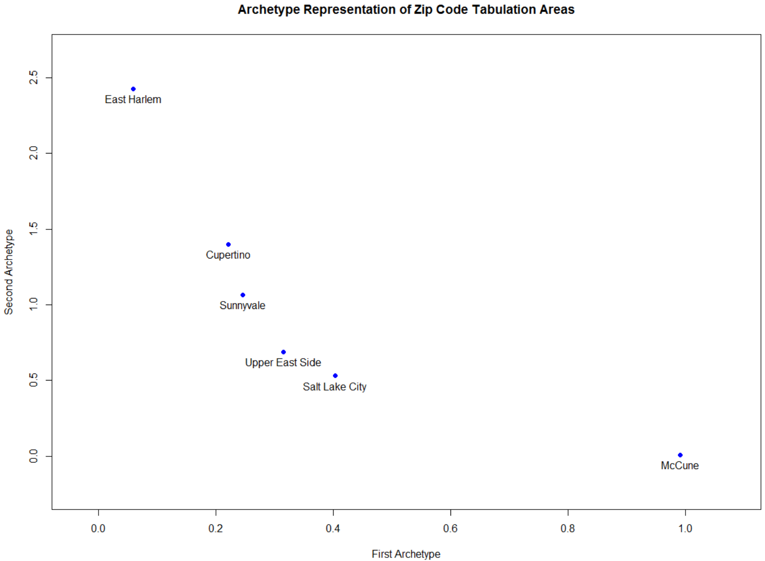
ACS demographic data:

zip	unemployment	mean income	$\cdots$
94305	12%	\$47,000	$\cdots$
06511	19%	\$32,000	$\cdots$
60647	23%	\$23,000	$\cdots$
94121	4%	\$178,000	$\cdots$
$\vdots$	$\vdots$	$\vdots$	

- 32,989 rows (zip codes) $\times$ 150 columns (demographic info)
- GLRM embeds zip codes into (low dimensional) *demography space*

Low rank models for dimensionality reduction

Zip code features:



Low rank models for dimensionality reduction

build 3 sets of features to predict violations:

- categorical: expand zip code to categorical variable
- concatenate: join tables on zip
- GLRM: replace zip code by low dimensional zip code features

fit a supervised (deep learning) model:

method	train error	test error	runtime
categorical	0.2091690	0.2173612	23.7600000
concatenate	0.2258872	0.2515906	4.4700000
GLRM	0.1790884	0.1933637	4.3600000

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

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

- **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

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

Conclusion

generalized low rank models

- find structure in data automatically
- can handle huge, heterogeneous data coherently
- transform big messy data into small clean data

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

H2O:
`https://github.com/h2oai/h2o-world-2015-training/blob/master/tutorials/glrm/glrm-tutorial.md`

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