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

<table>
<thead>
<tr>
<th>age</th>
<th>gender</th>
<th>state</th>
<th>income</th>
<th>education</th>
</tr>
</thead>
<tbody>
<tr>
<td>29</td>
<td>F</td>
<td>CT</td>
<td>$53,000</td>
<td>college</td>
</tr>
<tr>
<td>57</td>
<td>?</td>
<td>NY</td>
<td>$19,000</td>
<td>high school</td>
</tr>
<tr>
<td>?</td>
<td>M</td>
<td>CA</td>
<td>$102,000</td>
<td>masters</td>
</tr>
<tr>
<td>41</td>
<td>F</td>
<td>NV</td>
<td>$23,000</td>
<td>?</td>
</tr>
</tbody>
</table>

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

$m$ examples (patients, respondents, households, assets)  
n$\text{features (tests, questions, sensors, times)}$

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

- $i$th row of $A$ is feature vector for $i$th example  
- $j$th column of $A$ gives values for $j$th feature across all examples
Low rank model

given: \( A \in \mathbb{R}^{m \times n}, k \ll m, n \)

find: \( X \in \mathbb{R}^{m \times k}, Y \in \mathbb{R}^{k \times n} \) for which

\[
\begin{bmatrix}
X \\
Y
\end{bmatrix} \approx 
\begin{bmatrix}
A
\end{bmatrix}
\]

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

\[
\begin{bmatrix}
X
\end{bmatrix} = 
\begin{bmatrix}
x_1 \\
\vdots \\
x_m
\end{bmatrix}
\quad \begin{bmatrix}
Y
\end{bmatrix} = 
\begin{bmatrix}
y_1 \\
\vdots \\
y_n
\end{bmatrix}
\]

interpretation:

- \( X \) and \( Y \) are (compressed) representation of \( A \)
- \( x_i^T \in \mathbb{R}^k \) is a point associated with example \( i \)
- \( y_j \in \mathbb{R}^k \) 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:

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

with variables \( X \in \mathbb{R}^{m \times k}, \ Y \in \mathbb{R}^{k \times n} \)

▶ 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

<table>
<thead>
<tr>
<th>time</th>
<th>forehead (x)</th>
<th>forehead (y)</th>
<th>...</th>
<th>right toe (y)</th>
<th>right toe (z)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>1.4</td>
<td>2.7</td>
<td>...</td>
<td>-0.5</td>
<td>-0.1</td>
</tr>
<tr>
<td>$t_2$</td>
<td>2.7</td>
<td>3.5</td>
<td>...</td>
<td>1.3</td>
<td>0.9</td>
</tr>
<tr>
<td>$t_3$</td>
<td>3.3</td>
<td>-0.9</td>
<td>...</td>
<td>4.2</td>
<td>1.8</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

- 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)
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

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

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

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

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

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

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

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

<table>
<thead>
<tr>
<th>structure</th>
<th>( r(x) )</th>
<th>( \tilde{r}(y) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>small</td>
<td>( |x|_2^2 )</td>
<td>( |y|_2^2 )</td>
</tr>
<tr>
<td>sparse</td>
<td>( |x|_1 )</td>
<td>( |y|_1 )</td>
</tr>
<tr>
<td>nonnegative</td>
<td>( 1(x \geq 0) )</td>
<td>( 1(y \geq 0) )</td>
</tr>
<tr>
<td>clustered</td>
<td>( 1(\text{card}(x) = 1) )</td>
<td>( 0 )</td>
</tr>
</tbody>
</table>
**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:

<table>
<thead>
<tr>
<th>data type</th>
<th>loss</th>
<th>( L(u, a) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>real</td>
<td>quadratic</td>
<td>((u - a)^2)</td>
</tr>
<tr>
<td>real</td>
<td>absolute value</td>
<td>(</td>
</tr>
<tr>
<td>real</td>
<td>huber</td>
<td><strong>huber</strong>((u - a))</td>
</tr>
<tr>
<td>boolean</td>
<td>hinge</td>
<td>((1 - ua)_+)</td>
</tr>
<tr>
<td>boolean</td>
<td>logistic</td>
<td>(\log(1 + \exp(-au)))</td>
</tr>
<tr>
<td>integer</td>
<td>poisson</td>
<td>(\exp(u) - au + a \log a - a)</td>
</tr>
<tr>
<td>ordinal</td>
<td>ordinal hinge</td>
<td>(\sum_{a' = 1}^{a-1} (1 - u + a')<em>+ + \sum</em>{d = a + 1}^{a'} (1 + u - a')_+)</td>
</tr>
<tr>
<td>categorical</td>
<td>one-vs-all</td>
<td>((1 - u_a)<em>+ + \sum</em>{a' \neq a} (1 + u_{a'})_+)</td>
</tr>
</tbody>
</table>
Examples

variations on GLRMs recover many known models:

<table>
<thead>
<tr>
<th>Model</th>
<th>$L_j(u, a)$</th>
<th>$r(x)$</th>
<th>$\tilde{r}(y)$</th>
<th>reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA</td>
<td>$(u - a)^2$</td>
<td>0</td>
<td>0</td>
<td>[Pearson 1901]</td>
</tr>
<tr>
<td>matrix completion</td>
<td>$(u - a)^2$</td>
<td>$|x|_2^2$</td>
<td>$|y|_2^2$</td>
<td>[Keshavan 2010]</td>
</tr>
<tr>
<td>NNMF</td>
<td>$(u - a)^2$</td>
<td>$1(x \geq 0)$</td>
<td>$1(y \geq 0)$</td>
<td>[Lee 1999]</td>
</tr>
<tr>
<td>sparse PCA</td>
<td>$(u - a)^2$</td>
<td>$|x|_1$</td>
<td>$|y|_1$</td>
<td>[D'Aspremont 2004]</td>
</tr>
<tr>
<td>sparse coding</td>
<td>$(u - a)^2$</td>
<td>$|x|_1$</td>
<td>$|y|_2^2$</td>
<td>[Olshausen 1997]</td>
</tr>
<tr>
<td>$k$-means</td>
<td>$(u - a)^2$</td>
<td>$1(\text{card}(x) = 1)$</td>
<td>0</td>
<td>[Tropp 2004]</td>
</tr>
<tr>
<td>robust PCA</td>
<td>$</td>
<td>u - a</td>
<td>$</td>
<td>$|x|_2^2$</td>
</tr>
<tr>
<td>logistic PCA</td>
<td>$\log(1 + \exp(-au))$</td>
<td>$|x|_2^2$</td>
<td>$|y|_2^2$</td>
<td>[Collins 2001]</td>
</tr>
<tr>
<td>boolean PCA</td>
<td>$(1 - au)_+$</td>
<td>$|x|_2^2$</td>
<td>$|y|_2^2$</td>
<td>[Srebro 2004]</td>
</tr>
</tbody>
</table>
Impute heterogeneous data

**PCA:**

- Mixed data types
- Remove entries
- PCA rank 10 recovery
- Error

**GLRM:**

- Mixed data types
- Remove entries
- GLRM rank 10 recovery
- Error
Validate model

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

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

![Graph showing normalized test error vs. $\gamma$ for different $k$ values.]

(normalized test error vs. $\gamma$ for different $k$ values.)
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

\[
\begin{bmatrix}
|     |   \\
|     |   \\
|     |   \\
\end{bmatrix}
\]

<table>
<thead>
<tr>
<th>age</th>
<th>gender</th>
<th>state</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>29</td>
<td>F</td>
<td>CT</td>
<td>...</td>
</tr>
<tr>
<td>57</td>
<td>?</td>
<td>NY</td>
<td>...</td>
</tr>
<tr>
<td>?</td>
<td>M</td>
<td>CA</td>
<td>...</td>
</tr>
<tr>
<td>41</td>
<td>F</td>
<td>NV</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>company</th>
<th># employees</th>
<th>zip</th>
<th>violations</th>
</tr>
</thead>
<tbody>
<tr>
<td>h2o.ai</td>
<td>58</td>
<td>95050</td>
<td>0</td>
</tr>
<tr>
<td>stanford</td>
<td>8300</td>
<td>94305</td>
<td>0</td>
</tr>
<tr>
<td>caltech</td>
<td>741</td>
<td>91107</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

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

---

²labor 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:

<table>
<thead>
<tr>
<th>zip</th>
<th>unemployment</th>
<th>mean income</th>
</tr>
</thead>
<tbody>
<tr>
<td>94305</td>
<td>12%</td>
<td>$47,000</td>
</tr>
<tr>
<td>06511</td>
<td>19%</td>
<td>$32,000</td>
</tr>
<tr>
<td>60647</td>
<td>23%</td>
<td>$23,000</td>
</tr>
<tr>
<td>94121</td>
<td>4%</td>
<td>$178,000</td>
</tr>
</tbody>
</table>

- 32,989 rows (zip codes) × 150 columns (demographic info)
- GLRM embeds zip codes into (low dimensional) demography space
Low rank models for dimensionality reduction

Zip code features:

Archetype Representation of Zip Code Tabulation Areas

- East Harlem
- Cupertino
- Sunnyvale
- Upper East Side
- Salt Lake City
- McCune
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:

<table>
<thead>
<tr>
<th>method</th>
<th>train error</th>
<th>test error</th>
<th>runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>categorical</td>
<td>0.2091690</td>
<td>0.2173612</td>
<td>23.7600000</td>
</tr>
<tr>
<td>concatenate</td>
<td>0.2258872</td>
<td>0.2515906</td>
<td>4.4700000</td>
</tr>
<tr>
<td>GLRM</td>
<td>0.1790884</td>
<td>0.1933637</td>
<td>4.3600000</td>
</tr>
</tbody>
</table>
Fitting GLRM$s$ with alternating minimization

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

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

\[ x_i^{t+1} = \text{prox}_{\alpha_t r}(x_i^t - \alpha_t g) \]

(where \( \text{prox}_{f}(z) = \arg\min_x (f(x) + \frac{1}{2} \|x - z\|^2) \))

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

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