Generalized Low Rank Models

Madeleine Udell

Cornell ORIE admit visit day 3/18/2016
<table>
<thead>
<tr>
<th>age</th>
<th>gender</th>
<th>state</th>
<th>diabetes</th>
<th>education</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>F</td>
<td>CT</td>
<td>?</td>
<td>college</td>
</tr>
<tr>
<td>57</td>
<td>?</td>
<td>NY</td>
<td>severe</td>
<td>high school</td>
</tr>
<tr>
<td>?</td>
<td>M</td>
<td>CA</td>
<td>moderate</td>
<td>masters</td>
</tr>
<tr>
<td>41</td>
<td>F</td>
<td>NV</td>
<td>none</td>
<td>?</td>
</tr>
</tbody>
</table>

- 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)

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

- $i$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: $A$, $k \ll m, n$
find: $X \in \mathbb{R}^{m \times k}$, $Y \in \mathbb{R}^{k \times n}$ 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

$$
\begin{bmatrix}
X \\
\end{bmatrix}
= 
\begin{bmatrix}
\tilde{x}_1 \\
\vdots \\
\tilde{x}_m \\
\end{bmatrix}
\quad 
\begin{bmatrix}
Y \\
\end{bmatrix}
= 
\begin{bmatrix}
y_1 & \cdots & y_n \\
\end{bmatrix}
$$

interpretation:

- $X$ and $Y$ are (compressed) representation of $A$
- $x_i^T \in \mathbb{R}^k$ 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:

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

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 \( A = U\Sigma V^T \):

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

- (numerical) solution via alternating minimizing
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 \( r : \mathbb{R}^{1 \times k} \to \mathbb{R} \), \( \tilde{r} : \mathbb{R}^k \to \mathbb{R} \)
- 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:

<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>(\text{huber}(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>{a'=a+1}^{d}(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>
<tr>
<td>categorical</td>
<td>multinomial logit</td>
<td>(\frac{\exp(u_a)}{(\sum_{a'=1}^{d} \exp(u_{a'}))})</td>
</tr>
</tbody>
</table>
Regularizers

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>$|x|_2$</td>
<td>$|y|_2$</td>
</tr>
<tr>
<td>sparse</td>
<td>$|x|_1$</td>
<td>$|y|_1$</td>
</tr>
<tr>
<td>nonnegative</td>
<td>$\mathbf{1}(x \geq 0)$</td>
<td>$\mathbf{1}(y \geq 0)$</td>
</tr>
<tr>
<td>clustered</td>
<td>$\mathbf{1}(\text{card}(x) = 1)$</td>
<td>0</td>
</tr>
</tbody>
</table>
Impute heterogeneous data

mixed data types

remove entries

qpca rank 10 recovery

error

glrm rank 10 recovery

error
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]

GLRM outperforms PCA
Fitting GLRM with alternating minimization

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
Alternating updates

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

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

define the \textit{proximal operator}

\[
\text{prox}_f(z) = \arg\min_x (f(x) + \frac{1}{2}\|x - z\|^2_2)
\]

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

more info: [Parikh Boyd 2013]
A simple, fast update rule

proximal gradient method: let

\[ g = \sum_{j: (i,j) \in \Omega} \nabla L_j(x_iy_j, A_{ij})y_j \]

and update

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

- **simple:** only requires ability to evaluate \( \nabla L \) and \( \text{prox}_r \)
- **time per iteration:** \( O\left(\frac{(n+m+|\Omega|)k}{p}\right) \) 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 $k = 5$

```python
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

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

if and only if \(Z = XY\) is a solution to

\[
\begin{align*}
\minimize & \quad L(Z) + \gamma \|Z\|_* \\
\text{subject to} & \quad \text{Rank}(Z) \leq k
\end{align*}
\]

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 GLRMds (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