ORIE 4741: Learning with Big Messy Data

Unsupervised Learning

Professor Udell
Operations Research and Information Engineering
Cornell

September 10, 2019
Outline

Missing data

Unsupervised learning
Low rank models
Principal Components Analysis
Generalized Low Rank Models
Imputing missing data
Multidimensional losses
Exotic regularizers
Clustering
Missing data

examples:

- weather data: missing data due to sensor failures
- survey data: missing data due to non-response
- purchase/click/like data: missing data due to lack of purchase/click/like
- drug trial: missing data due to subjects leaving trial
## Data table: survey data

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

strategy 1:

▶ drop rows or columns with missing data
How to cope with missing data?

strategy 1:

- drop rows or columns with missing data

how well would this work for

- weather data
- survey data
- purchase/click/like data
- drug trial
How to cope with missing data?

strategy 2:

▶ fill in missing entries with row or column mean
How to cope with missing data?

strategy 2:
▶ fill in missing entries with row or column mean

how well would this work for
▶ weather data
▶ survey data
▶ purchase/click/like data
▶ drug trial
strategy 3:

▶ use other columns to **predict** missing entries
How to cope with missing data?

strategy 3:

▶ use other columns to **predict** missing entries

how well would this work for

▶ weather data
▶ survey data
▶ purchase/click/like data
▶ drug trial
How to cope with missing data?

strategy 3:

▶ use other columns to **predict** missing entries

how well would this work for

▶ weather data
▶ survey data
▶ purchase/click/like data
▶ drug trial

problem: what if **all columns** have (some) missing data?
Outline

Missing data

Unsupervised learning

Low rank models

Principal Components Analysis

Generalized Low Rank Models

Imputing missing data

Multidimensional losses

Exotic regularizers

Clustering
Unsupervised learning

strategy 4:

- simultaneously learn **regression coefficients** and **covariates** to predict every entry in data well

this is such a weird idea that we will need new terminology:

- we no longer can divide the data into **inputs** and **outputs**, or **features** and **labels**, or **covariates** and **responses**
- all we have are some **features** for each **example**
- this setting is called **unsupervised**
Data table

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

\[
\begin{bmatrix}
Y \\
\end{bmatrix} =
\begin{bmatrix}
Y_{11} & \cdots & Y_{1d} \\
\vdots & \ddots & \vdots \\
Y_{n1} & \cdots & Y_{nd}
\end{bmatrix}
\]

- $i$th row of $Y$ is feature vector for $i$th example
- $j$th column of $Y$ gives values for $j$th feature across all examples
Outline

Missing data
Unsupervised learning

Low rank models
Principal Components Analysis
Generalized Low Rank Models
Imputing missing data
Multidimensional losses
Exotic regularizers
Clustering
Low rank model

given: \( n \times d \) data table \( Y \), \( r \leq n, d \)
find: \( X \in \mathbb{R}^{n \times r} \), \( W \in \mathbb{R}^{r \times d} \) for which

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

i.e., \( x_i^T w_j \approx Y_{ij} \), where

\[
\begin{bmatrix}
X
\end{bmatrix}
=
\begin{bmatrix}
x_1^T \\
\vdots \\
x_n^T
\end{bmatrix}
\begin{bmatrix}
W
\end{bmatrix}
=
\begin{bmatrix}
w_1 \\
\vdots \\
w_d
\end{bmatrix}
\]

interpretation:

- \( r = \text{Rank}(XW) \) is the rank of the model
- \( X \) and \( W \) are (compressed) representation of \( Y \)
- \( x_i \in \mathbb{R}^r \) is a point associated with example \( i \)
- \( w_j \in \mathbb{R}^r \) is a point associated with feature \( j \)
- inner product \( x_i w_j \) approximates \( Y_{ij} \)
 Exact low rank fitting

simplest case: suppose $Y \in \mathbb{R}^{n \times d}$ has no missing entries

**Q:** what is the smallest $r$ so that

$$Y = XW$$

for $X \in \mathbb{R}^{n \times r}$, $W \in \mathbb{R}^{r \times d}$?

($XW$ is called a **factorization** of $Y$)
Exact low rank fitting

**simplest case:** suppose $Y \in \mathbb{R}^{n \times d}$ has no missing entries

**Q:** what is the smallest $r$ so that

$$Y = XW$$

for $X \in \mathbb{R}^{n \times r}$, $W \in \mathbb{R}^{r \times d}$?

($XW$ is called a **factorization** of $Y$)

**A:** $r = \text{Rank}(Y)$!
Exact low rank fitting

**theorem:** for \( Y \in \mathbb{R}^{n \times d} \), \( \text{Rank}(Y) \) is the smallest \( r \) for which we can find

\[
X \in \mathbb{R}^{n \times r}, \quad W \in \mathbb{R}^{r \times d}
\]

so that

\[
Y = XW.
\]

**proof:** \( r \leq \text{Rank}(Y) \):

- suppose \( Y = U\Sigma V^T \) is the skinny SVD of \( Y \)
- then \( \text{Rank}(Y) = \) number of columns of \( U \) and of \( V \)
- let \( X = U \), \( W = \Sigma V^T \)
- then \( Y = XW \)

\( r \geq \text{Rank}(Y) \):

- \( \text{Rank}(Y) = \text{Rank}(XW) \leq \text{Rank}(X)\text{Rank}(W) \leq r \)
Inexact low rank fitting

if we’re willing to represent $Y$ approximately, can we reduce the rank $r$?
Outline

Missing data
Unsupervised learning
Low rank models

**Principal Components Analysis**

Generalized Low Rank Models
Imputing missing data
Multidimensional losses
Exotic regularizers
Clustering
Principal components analysis (PCA)

**Principal components analysis (PCA):** Given $Y \in \mathbb{R}^{n \times d}$, solve

$$\text{minimize} \quad \| Y - XW \|_F^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} (Y_{ij} - x_i^T w_j)^2$$

with $X \in \mathbb{R}^{n \times r}$, $W \in \mathbb{R}^{r \times d}$

- a very old problem (Pearson 1901, Hotelling 1933)
- least squares low rank fitting
Principal components analysis (PCA): Given $Y \in \mathbb{R}^{n \times d}$, solve

$$\minimize \| Y - XW \|_F^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} (Y_{ij} - x_i^T w_j)^2$$

with $X \in \mathbb{R}^{n \times r}$, $W \in \mathbb{R}^{r \times d}$

- a very old problem (Pearson 1901, Hotelling 1933)
- least squares low rank fitting

**notice**: objective depends only on product $XW$, so if $(X, W)$ is a solution, so is $(\tilde{X}, \tilde{W}) = (XT, T^{-1}W)$ for any invertible matrix $T \in \mathbb{R}^{r \times r}$:

$$\tilde{X}\tilde{W} = XTT^{-1}W = XW.$$
Principal components analysis (PCA)

Principal components analysis (PCA): Given $Y \in \mathbb{R}^{n \times d}$, solve

$$\text{minimize } \| Y - XW \|_F^2 = \sum_{i=1}^m \sum_{j=1}^n (Y_{ij} - x_i^T w_j)^2$$

with $X \in \mathbb{R}^{n \times r}$, $W \in \mathbb{R}^{r \times d}$

- a very old problem (Pearson 1901, Hotelling 1933)
- least squares low rank fitting

notice: objective depends only on product $XW$, so if $(X, W)$ is a solution, so is $(\tilde{X}, \tilde{W}) = (XT, T^{-1}W)$ for any invertible matrix $T \in \mathbb{R}^{r \times r}$:

$$\tilde{X}\tilde{W} = XTT^{-1}W = XW.$$ 

make sure interpretation of solution is invariant under $T$
PCA finds best covariates

regression:

\[
\text{minimize } \| Y - XW \|_F^2,
\]

\(d = 2, \ r = 1, \ \text{fix } X = Y_{:,1:r} \) (first \( r \) columns of \( Y \)), variable \( W \)
PCA finds best covariates

PCA:

minimize \( \| Y - XW \|_F^2 \),

d = 2, \( r = 1 \), variables \( X \) and \( W \)
On lines and planes of best fit

[Pearson 1901]
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>-.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 $W$ are principal stances
- rows of $X$ decompose stance into combination of principal stances
Interpreting principal components

columns of $Y$ (features) (height of point over time)
Interpreting principal components

columns of $Y$ (features) (depth of point over time)
Interpreting principal components

row of $W$
(archetypical example)
(principal stance)
Interpreting principal components

columns of $X$ (archetypical features) (principal timeseries)
Interpreting principal components

column of $X$ (red) (predicted feature)
column of $Y$ (blue) (observed feature)
Principal components analysis (PCA): Given $Y \in \mathbb{R}^{n \times d}$, solve

$$\text{minimize} \quad \|Y - XW\|_F^2 = \sum_{i=1}^n \sum_{j=1}^d (Y_{ij} - x_i^T w_j)^2$$

with $X \in \mathbb{R}^{n \times r}$, $W \in \mathbb{R}^{r \times d}$

how should we solve this problem?
Principal components analysis (PCA): Given $Y \in \mathbb{R}^{n \times d}$, solve

$$\text{minimize} \quad \| Y - XW \|_F^2 = \sum_{i=1}^{n} \sum_{j=1}^{d} (Y_{ij} - x_i^T w_j)^2$$

with $X \in \mathbb{R}^{n \times r}$, $W \in \mathbb{R}^{r \times d}$

how should we solve this problem?

- idea 1: use the SVD
- idea 2: alternating minimization over $X$ and $W$
The Frobenius norm

the **Frobenius norm**

\[
\|A\|_F = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{d} A_{ij}^2}
\]

some useful identities:

- \(\|A\|_F = \|\text{vec}(A)\|\)
- \(\|A\|_F = \|A^T\|_F\)
- \(\|A\|_F^2 = \text{tr}(A^T A)\)
- if \(U\) is orthogonal (i.e., \(U^TU = I\)), then \(\|UA\|_F = \|A\|_F\)

proof:

\[
\|UA\|_F^2 = \text{tr}((UA)^T UA) = \text{tr}(A^T U^T UA) = \text{tr}(A^T A) = \|A\|_F^2
\]
PCA: solution via the SVD

**PCA:** with $X \in \mathbb{R}^{n \times r}$, $W \in \mathbb{R}^{r \times d}$, solve

$$\text{minimize} \quad \|Y - XW\|_F^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} (Y_{ij} - x_i^T w_j)^2$$

**Eckart-Young-Mirsky theorem:** if

$$Y = U\Sigma V^T = \sum_{i=1}^{\text{Rank}(Y)} \sigma_i u_i v_i^T$$

is the SVD of $Y$, then

$$X = U_r, \quad W = \Sigma_r V_r^T$$

is a solution to PCA, where

$$\Sigma_r = \text{diag}(\sigma_1, \ldots, \sigma_r), \quad U_r = [u_1 \cdots u_r], \quad V_r = [v_1 \cdots v_r].$$
PCA: solution via the SVD

PCA: with $X \in \mathbb{R}^{n \times r}$, $W \in \mathbb{R}^{r \times d}$, solve

$$\minimize \quad \| Y - XW \|_F^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} (Y_{ij} - x_i^T w_j)^2$$

Eckart-Young-Mirsky theorem: if

$$Y = U \Sigma V^T = \sum_{i=1}^{\text{Rank}(Y)} \sigma_i u_i v_i^T$$

is the SVD of $Y$, then

$$X = U_r, \quad W = \Sigma_r V_r^T$$

is a solution to PCA, where

$$\Sigma_r = \text{diag}(\sigma_1, \ldots, \sigma_r), \quad U_r = [u_1 \cdots u_r], \quad V_r = [v_1 \cdots v_r].$$

with this $X$ and $W$,

$$\| Y - XW \|_F^2 = \| U \Sigma V^T - U_r \Sigma_r V_r^T \|_F^2 = \sum_{i=r+1}^{\text{Rank}(Y)} \sigma_i^2$$
Proof of Eckart-Young-Mirsky theorem I

Proof step 1: reduce to diagonal.
if $Y = UΣV^T$ is the full SVD, then

$$U^TU = UU^T = I \text{ and } V^TV = VV^T = I,$$

so

$$\|Y - XW\|_F^2 = \|UΣV^T - XW\|_F^2 = \|U^TUΣV^TV - U^TXWV\|_F^2 = \|Σ - U^TXWV\|_F^2 = \|Σ - Z\|_F^2$$

where $Z = U^TXWV$ is a rank $r$ matrix.
we want to show

$$\text{Rank}(Y) \sum_{i=r+1}^{\sigma_i} \leq \|Σ - Z\|_F^2$$

for any rank $r$ matrix $Z$. 
Proof of Eckart-Young-Mirsky theorem II

proof step 2: eigenvalue interlacing.

let’s use Weyl’s theorem for eigenvalues:

for any matrices $A, B \in \mathbb{R}^{n \times d}$,

$$\sigma_{i+j-1}(A + B) \leq \sigma_i(A) + \sigma_j(B), \quad 1 \leq i, j \leq n.$$ 

set $A = \Sigma - Z$, $B = Z$, $j = r + 1$ to get

$$\sigma_{i+r}(\Sigma) \leq \sigma_i(\Sigma - Z) + \sigma_{r+1}(Z), \quad 1 \leq i \leq n - r$$

$$\sigma_{i+r} \leq \sigma_i(\Sigma - Z), \quad 1 \leq i \leq n - r,$$

using $\text{Rank}(Z) \leq r$. square and sum from $i = 1$ to $\text{Rank}(Y) - r$:

$$\|\Sigma - \Sigma_r\|_F^2 = \sum_{i=r+1}^{\text{Rank}(Y)} \sigma_i^2 \leq \sum_{i=1}^{\text{Rank}(Y)-r} \sigma_i^2(\Sigma - Z) \leq \|\Sigma - Z\|_F^2.$$
PCA: solution via AM

minimize $\| Y - XW \|^2_F = \sum_{i=1}^m \sum_{j=1}^n (Y_{ij} - x_i^T w_j)^2$

Alternating Minimization (AM): fix $W^0$. for $t = 1, \ldots$,

$\begin{align*}
X^t &= \text{argmin}_X \| Y - XW^{t-1} \|^2_F \\
W^t &= \text{argmin}_W \| Y - X^t W \|^2_F
\end{align*}$

properties:

$\begin{align*}
\&\quad \text{objective decreases at each iteration} \\
\&\quad \text{objective bounded below, so the procedure converges} \\
\&\quad (\text{it is true but we won’t prove that}) \text{ with probability 1 over} \\
\&\quad \text{choices of } W^0, \text{ AM converges to an optimal solution}
\end{align*}$
how would you solve the AM subproblem

$$W^t = \arg\min_W \| Y - X^t W \|_F^2 = \arg\min_W \sum_{j=1}^{d} \| y_j - X^t w_j \|_2^2$$

where $Y = [y_1 \cdots y_d]$, $W = [w_1 \cdots w_d]$?
PCA: AM subproblem is separable

how would you solve the AM subproblem

\[ W^t = \arg\min_W \| Y - X^t W \|_F^2 = \arg\min_W \sum_{j=1}^{d} \| y_j - X^t w_j \|_2^2 \]

where \( Y = [y_1 \cdots y_d] \), \( W = [w_1 \cdots w_d] \)?

- problem separates over columns of \( W \):
  \[ w_j^t = \arg\min_w \| y_j - X^t w \|_2^2 \]

- for each column of \( W \), it’s just a least squares problem!
  \[ w_j = \left( (X^t)^T X^t \right)^{-1} (X^t)^T y_j \]
PCA: solution via AM

\[
\text{minimize } \| Y - XW \|_F^2 = \sum_{i=1}^n \sum_{j=1}^d (Y_{ij} - x_i^T w_j)^2
\]

Alternating Minimization (AM): fix \( W^0 \). for \( t = 1, \ldots, \)

- for \( i = 1, \ldots, n, \)

\[
x_i^t = Y_i : (W^{t-1})^T (W^{t-1} (W^{t-1})^T)^{-1}
\]

- for \( j = 1, \ldots, d, \)

\[
w_j^t = ((X^t)^T X^t)^{-1} (X^t)^T y_j
\]
PCA: solution via AM

\[
\text{minimize } \| Y - XW \|_F^2 = \sum_{i=1}^n \sum_{j=1}^d (Y_{ij} - x_i^T w_j)^2
\]

computational tricks:

- cache gram matrix \( G = (X^t)^T X^t \)
- parallelize over \( j \)

**Alternating Minimization (AM):** fix \( W^0 \) for \( t = 1, \ldots, \)

- cache factorization of \( G = W^{t-1}(W^{t-1})^T \)
- in parallel, for \( i = 1, \ldots, n, \)
  \[
  x_i^t = Y_i : (W^{t-1})^T (W^{t-1}(W^{t-1})^T)^{-1}
  \]
- cache factorization of \( G = (X^t)^T X^t \)
- in parallel, for \( j = 1, \ldots, d, \)
  \[
  w_j^t = ((X^T X^t)^{-1} X^T)^T y_j
  \]
PCA: solution via AM

minimize $\| Y - XW \|_F^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} (Y_{ij} - x_i^T w_j)^2$

complexity?

**Alternating Minimization (AM):** fix $W^0$. for $t = 1, \ldots,$

- cache factorization of $G = W^{t-1}(W^{t-1})^T$
PCA: solution via AM

\[
\text{minimize} \quad \| Y - XW \|_F^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} (Y_{ij} - x_i^T w_j)^2
\]

complexity?

**Alternating Minimization (AM):** fix \( W^0 \). for \( t = 1, \ldots, \)

- cache factorization of \( G = W^{t-1}(W^{t-1})^T \) \((O(dr^2 + r^3))\)
- in parallel, for \( i = 1, \ldots, n, \)

\[
x_i^t = (W^{t-1}(W^{t-1})^T)^{-1} W^{t-1} Y_i^T
\]
PCA: solution via AM

\[
\text{minimize} \quad \| Y - XW \|_F^2 = \sum_{i=1}^m \sum_{j=1}^n (Y_{ij} - x_i^T w_j)^2
\]

complexity?

**Alternating Minimization (AM):** fix $W^0$. for $t = 1, \ldots,$

- cache factorization of $G = W^{t-1}(W^{t-1})^T \left( \mathcal{O}(dr^2 + r^3) \right)$
- in parallel, for $i = 1, \ldots, n,$

\[
x_i^t = (W^{t-1}(W^{t-1})^T)^{-1}W^{t-1} Y_i^T
\]

\left( \mathcal{O}(dr + r^2) \right)

- cache factorization of $G = (X^t)^T X^t$
PCA: solution via AM

minimize \( \| Y - XW \|_F^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} (Y_{ij} - x_i^T w_j)^2 \)

complexity?

Alternating Minimization (AM): fix \( W^0 \). for \( t = 1, \ldots, \)

- cache factorization of \( G = W^{t-1}(W^{t-1})^T \) \( (O(dr^2 + r^3)) \)
- in parallel, for \( i = 1, \ldots, n, \)

\[
x_i^t = (W^{t-1}(W^{t-1})^T)^{-1} W^{t-1} Y_{i:}^T
\]

\( (O(dr + r^2)) \)

- cache factorization of \( G = (X^T)^T X^t \) \( (O(nr^2 + r^3)) \)
- in parallel, for \( j = 1, \ldots, d, \)

\[
w_j^t = ((X^T)^T X^t)^{-1} (X^T)^T y_j
\]
**PCA: solution via AM**

\[
\text{minimize } \| Y - XW \|_F^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} (Y_{ij} - x_i^T w_j)^2
\]

complexity?

**Alternating Minimization (AM):** fix \( W^0 \). for \( t = 1, \ldots, \)

- cache factorization of \( G = W^{t-1}(W^{t-1})^T (O(dr^2 + r^3)) \)
- in parallel, for \( i = 1, \ldots, n, \)
  \[
  x_i^t = (W^{t-1}(W^{t-1})^T)^{-1} W^{t-1} Y_{i:}
  \]
  \((O(dr + r^2))\)
- cache factorization of \( G = (X^t)^T X^t (O(nr^2 + r^3)) \)
- in parallel, for \( j = 1, \ldots, d, \)
  \[
  w_j^t = ((X^t)^T X^t)^{-1} (X^t)^T y_j
  \]
  \((O(nr + r^2))\)
Outline

Missing data
Unsupervised learning
Low rank models
Principal Components Analysis
Generalized Low Rank Models
Imputing missing data
Multidimensional losses
Exotic regularizers
Clustering
Missing data?

now suppose

- $Y \in \mathbb{R}^{n \times d}$
- observe $Y_{ij}$ only for $(i,j) \in \Omega \subset \{1, \ldots, m\} \times \{1, \ldots, n\}$

what should we do?
Missing data?

now suppose

- \( Y \in \mathbb{R}^{n \times d} \)
- observe \( Y_{ij} \) only for \((i, j) \in \Omega \subset \{1, \ldots, m\} \times \{1, \ldots, n\}\)

what should we do?

**Matrix completion:**

\[
\text{minimize} \quad \sum_{(i,j) \in \Omega} (Y_{ij} - x_i^T w_j)^2 + \lambda \sum_{i=1}^n \|x_i\|_2^2 + \lambda \sum_{j=1}^d \|w_j\|_2^2
\]

two regimes:

- **some entries missing:** don’t waste data; “borrow strength” from entries that are **not** missing
- **most entries missing:** matrix completion still works!
Huber PCA

\[
\text{minimize} \quad \sum_{(i,j) \in \Omega} \text{huber}(Y_{ij} - x_i^T w_j) + \sum_{i=1}^n \|x_i\|_2^2 + \sum_{j=1}^d \|w_j\|_2^2
\]
Huber PCA

Huber loss with corrupted data (asymmetric noise)

- glrm
- pca

Fraction of corrupted entries vs. relative MSE.
Generalized low rank models

\[
\text{minimize} \quad \sum_{(i,j) \in \Omega} \ell_j(Y_{ij}, x_i^T w_j) + \sum_{i=1}^{n} r_i(x_i) + \sum_{j=1}^{d} \tilde{r}_j(w_j)
\]

- observe only \((i, j) \in \Omega\) (other entries are missing)
- loss functions \(\ell_j\) for each column
  - assume \(Y_{ij} \in \mathcal{Y}_j\) for every \((i, j) \in \Omega\)
  - \(\ell_j : \mathcal{Y}_j \times \mathbb{R} \to \mathbb{R}\)
  - e.g., different losses for reals, booleans, categoricals, ordinals, …
- regularizers \(r : \mathbb{R}^{1 \times r} \to \mathbb{R}\), \(\tilde{r} : \mathbb{R}^{r} \to \mathbb{R}\)
Demo: generalized low rank models

https://github.com/ORIE4741/demos/blob/master/LowRankModelsDemo.ipynb
Losses

minimize \( \sum_{(i,j) \in \Omega} \ell_j(Y_{ij}, x_i^T w_j) + \sum_{i=1}^n r_i(x_i) + \sum_{j=1}^d \tilde{r}_j(w_j) \)

choose loss \( \ell(y, z) \) adapted to data type:

<table>
<thead>
<tr>
<th>data type</th>
<th>loss</th>
<th>( \ell(y, z) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>real</td>
<td>quadratic</td>
<td>((y - z)^2)</td>
</tr>
<tr>
<td>real</td>
<td>absolute value</td>
<td>(</td>
</tr>
<tr>
<td>real</td>
<td>huber</td>
<td>\textit{huber}(y - z)</td>
</tr>
<tr>
<td>boolean</td>
<td>hinge</td>
<td>((1 - yz)_+)</td>
</tr>
<tr>
<td>boolean</td>
<td>logistic</td>
<td>(\log(1 + \exp(-yz)))</td>
</tr>
<tr>
<td>ordinal</td>
<td>ordinal hinge</td>
<td>(\sum_{y'=1}^{y-1}(1 - z + y')<em>+ + \sum</em>{y'=y+1}^k (1 + z - y')_+)</td>
</tr>
</tbody>
</table>
Impute missing data

Impute most likely true data $\hat{Y}_{ij}$

$$
\hat{Y}_{ij} = \arg\min_{y \in \mathcal{Y}_j} \ell_j(y, x_i^T w_j)
$$

- constraint: $\hat{Y}_{ij} \in y \in \mathcal{Y}_j$
- when $\ell_j$ is quadratic, $\ell_1$, or Huber loss, then $\hat{Y}_{ij} = x_i^T w_j$
- if $\mathcal{Y}_j \neq \mathbb{R}$, $\arg\min_y \ell_j(y, x_i^T w_j) \neq x_i^T w_j$
  - e.g., for hinge loss $\ell_j(y, z) = (1 - yz)_+$, $\hat{Y}_{ij} = \text{sign}(x_i^T w_j)$
Outline

Missing data
Unsupervised learning
Low rank models
Principal Components Analysis
Generalized Low Rank Models
Imputing missing data
Multidimensional losses
Exotic regularizers
Clustering
Impute heterogeneous data

mixed data types

remove entries
Impute heterogeneous data

mixed data types

remove entries

qpca rank 10 recovery

error

glrm rank 10 recovery

error
Impute heterogeneous data

mixed data types

remove entries

qpca rank 10 recovery

error

glrm rank 10 recovery

error
Impute censored data

market segmentation

<table>
<thead>
<tr>
<th>customer</th>
<th>apples</th>
<th>oranges</th>
<th>pears</th>
<th>⋮</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>yes</td>
<td>?</td>
<td>yes</td>
<td>⋮</td>
</tr>
<tr>
<td>2</td>
<td>yes</td>
<td>yes</td>
<td>?</td>
<td>⋮</td>
</tr>
<tr>
<td>3</td>
<td>?</td>
<td>?</td>
<td>yes</td>
<td>⋮</td>
</tr>
<tr>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
</tr>
</tbody>
</table>

- rows of $W$ are purchasing patterns for market segments
- rows of $X$ classify customers into market segment(s)
- imputation: recommend new products, target advertising campaign
**Impute censored data**

**synthetic data:**

- generate rank-5 matrix of probabilities, \( p \in \mathbb{R}^{300 \times 300} \)

<table>
<thead>
<tr>
<th>customer</th>
<th>apples</th>
<th>oranges</th>
<th>pears</th>
<th>⋮</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.28</td>
<td>.22</td>
<td>.76</td>
<td>⋮</td>
</tr>
<tr>
<td>2</td>
<td>.97</td>
<td>.55</td>
<td>.36</td>
<td>⋮</td>
</tr>
<tr>
<td>3</td>
<td>.13</td>
<td>.47</td>
<td>.62</td>
<td>⋮</td>
</tr>
<tr>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
</tr>
</tbody>
</table>
Impute censored data

synthetic data:

▶ entry \((i, j)\) is + with probability \(p_{ij}\)

<table>
<thead>
<tr>
<th>customer</th>
<th>apples</th>
<th>oranges</th>
<th>pears</th>
<th>⋯</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>⋯</td>
</tr>
<tr>
<td>2</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>⋯</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>⋯</td>
</tr>
<tr>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
</tr>
</tbody>
</table>
**Impute censored data**

Synthetic data:

- but we only observe $+$s...

<table>
<thead>
<tr>
<th>customer</th>
<th>apples</th>
<th>oranges</th>
<th>pears</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>+</td>
<td>?</td>
<td>+</td>
<td>...</td>
</tr>
<tr>
<td>2</td>
<td>+</td>
<td>+</td>
<td>?</td>
<td>...</td>
</tr>
<tr>
<td>3</td>
<td>?</td>
<td>+</td>
<td>+</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
**Impute censored data**

synthetic data:

- and we only observe 10% of the +s

<table>
<thead>
<tr>
<th>customer</th>
<th>apples</th>
<th>oranges</th>
<th>pears</th>
<th>⋯</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>+</td>
<td>?</td>
<td>?</td>
<td>⋯</td>
</tr>
<tr>
<td>2</td>
<td>?</td>
<td>+</td>
<td>?</td>
<td>⋯</td>
</tr>
<tr>
<td>3</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>⋯</td>
</tr>
<tr>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
</tr>
</tbody>
</table>
Impute censored data

synthetic data:

> ... and we only observe 10% of the +s

<table>
<thead>
<tr>
<th>customer</th>
<th>apples</th>
<th>oranges</th>
<th>pears</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>+</td>
<td>?</td>
<td>?</td>
<td>...</td>
</tr>
<tr>
<td>2</td>
<td>?</td>
<td>+</td>
<td>?</td>
<td>...</td>
</tr>
<tr>
<td>3</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

can we predict 10 more +s?
Impute censored data

- Precision@10:
  - 47 / 81
Outline

Missing data
Unsupervised learning
Low rank models
Principal Components Analysis
Generalized Low Rank Models
Imputing missing data
Multidimensional losses
Exotic regularizers
Clustering
Multi-dimensional loss

- approximate using vectors $x_i W_j \in \mathbb{R}^{1 \times d_j}$ instead of numbers
- need $\ell_j : \mathbb{R}^{1 \times d_j} \times \mathcal{Y}_j \to \mathbb{R}$

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

- useful for approximating categorical variables
  - columns of $W_j$ represent different labels of categorical variable
- gives more flexible/accurate models for ordinal variables
Multivariate categorical loss

▶ choose any loss function for multiclass classification to penalize \( x_i, Y \)
▶ e.g., one-vs-all (elementwise hinge loss) [Rifkin 2004]

\[
\ell(z, y) = (1 - z_y)_+ + \sum_{y' \neq y} (1 + z_{y'})_+
\]

\[
\begin{bmatrix}
CA & NV & \cdots & PA & NY \\
T & F & \cdots & F & F \\
F & F & \cdots & T & F \\
\vdots & \vdots & \cdots & \vdots & \vdots \\
\end{bmatrix}
\approx
\begin{bmatrix}
\overline{x_1} \\
\vdots \\
\overline{x_m}
\end{bmatrix}
\]
Multivariate ordinal loss

- automatically detect which labels are more similar
- fit positions of data ($X$) and separating hyperplanes ($W$) simultaneously
Scaling losses

Analogue of standardization for GLRMs:

\[
\mu_j = \arg\min_\mu \sum_{i:(i,j)\in\Omega} \ell_j(\mu, Y_{ij})
\]

\[
\sigma_j^2 = \frac{1}{n_j - 1} \sum_{i:(i,j)\in\Omega} \ell_j(\mu_j, Y_{ij})
\]

- \(n_j\) is number of observations in column \(j\)
- \(\mu_j\) generalizes column mean
- \(\sigma_j^2\) generalizes column variance

To fit a standardized GLRM, solve

\[
\text{minimize} \quad \sum_{(i,j)\in\Omega} \ell_j(Y_{ij}, x_i W_j + \mu_j)/\sigma_j^2 + \sum_{i=1}^n r_i(x_i) + \sum_{j=1}^d \tilde{r}_j(W_j)
\]
Scaling losses

Analogue of standardization for GLRMs:

\[ \mu_j = \arg\min_{\mu} \sum_{i:(i,j)\in \Omega} \ell_j(\mu, Y_{ij}) \]

\[ \sigma_j^2 = \frac{1}{n_j - 1} \sum_{i:(i,j)\in \Omega} \ell_j(\mu_j, Y_{ij}) \]

- \( n_j \) is number of observations in column \( j \)
- \( \mu_j \) generalizes column mean
- \( \sigma_j^2 \) generalizes column variance

To fit a standardized GLRM, solve

\[
\text{minimize } \sum_{(i,j)\in \Omega} \ell_j(Y_{ij}, x_i W_j + \mu_j)/\sigma_j^2 + \sum_{i=1}^{n} r_i(x_i) + \sum_{j=1}^{d} \tilde{r}_j(\tilde{W}_j)
\]

can be put in standard form: add an offset by modifying \( r! \)
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
Application: exploratory data analysis

- cluster respondents?

\[
\begin{bmatrix}
  w_1 & \cdots & w_d
\end{bmatrix}
\]

\[
\begin{bmatrix}
  x_1^T \\
  \vdots \\
  x_n^T
\end{bmatrix}
\]
Application: exploratory data analysis

\[
\begin{bmatrix}
\begin{array}{ccc}
w_1 & \cdots & w_d
\end{array}
\end{bmatrix}
\]

<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 \( X \)
Application: exploratory data analysis

<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>

\[
\begin{bmatrix}
  w_1 & \cdots & w_d
\end{bmatrix}
\]

\[
\begin{bmatrix}
  x_1^T \\
  \vdots \\
  x_n^T
\end{bmatrix}
\approx

- cluster respondents? **cluster rows of \( X \)**
- demographic profiles?
Application: exploratory data analysis

<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>

\[
\begin{bmatrix}
\begin{array}{c|c|c|c|}
{w_1} & \cdots & {w_d}
\end{array}
\end{bmatrix}
\]

- cluster respondents? **cluster rows of** \( X \)
- demographic profiles? **rows of** \( W \)
### Application: exploratory data analysis

<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>

$\begin{bmatrix}
  w_1 & \cdots & w_d
\end{bmatrix}$

$\begin{bmatrix}
  x_1^T \\
  \vdots \\
  x_n^T
\end{bmatrix}$

- cluster respondents? **cluster rows of $X$**
- demographic profiles? **rows of $W$**
- which features are similar?
**Application: exploratory data analysis**

$$\begin{bmatrix}
  w_1 & \cdots & w_d \\
\end{bmatrix}$$

<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** $W$
- which features are similar? **cluster columns of** $W$
### Application: exploratory data analysis

The table below shows a sample dataset:

<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>

The dataset is represented as a matrix $X$ and columns are represented as $X^T$.

- **Cluster respondents?** *Cluster rows of $X$*
- **Demographic profiles?** *Rows of $W$*
- **Which features are similar?** *Cluster columns of $W$*
- **Impute missing entries?**

The matrix $W$ is shown as:

$$
W = \begin{bmatrix}
w_1 & \cdots & w_d
\end{bmatrix}
$$
### Application: exploratory data analysis

![Matrix Diagram](attachment://matrix_diagram.png)

<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** $W$
- ▶ which features are similar? **cluster columns of** $W$
- ▶ impute missing entries? $\arg\min_{y \in Y} \ell_j(y, x_i^T w_j)$
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

in 2 lines of code:

```r
glrm, labels = GLRM(Y, 10, scale = true)
X,W = fit!(glrm)
```
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>zip</th>
<th>violations</th>
<th>⋮</th>
</tr>
</thead>
<tbody>
<tr>
<td>Holiday Inn</td>
<td>14850</td>
<td>109</td>
<td>⋮</td>
</tr>
<tr>
<td>Moosewood Restaurant</td>
<td>14850</td>
<td>0</td>
<td>⋮</td>
</tr>
<tr>
<td>Cornell Orchards</td>
<td>14850</td>
<td>0</td>
<td>⋮</td>
</tr>
<tr>
<td>Lakeside Nursing Home</td>
<td>14850</td>
<td>53</td>
<td>⋮</td>
</tr>
</tbody>
</table>

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

\(^1\)labor law violation demo: https://github.com/h2oai/h2o-3/blob/master/h2o-r/demos/rdemo.census.labor.violations.large.R
ACS demographic data:

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

- 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:

Archetype Representation of Zip Code Tabulation Areas

- East Harlem
- Cupertino
- Sunnyvale
- Upper East Side
- Salt Lake City
- McCune
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>
recap: why use GLRM$s$?

use GLRM$s$ to

► fill in missing data
► embed data points into low dimensional space
► reduce dimensionality of large categorical features
► design recommender systems
Outline

Missing data
Unsupervised learning
Low rank models
Principal Components Analysis
Generalized Low Rank Models
Imputing missing data
Multidimensional losses
Exotic regularizers
Clustering
Low rank models for finance

factor model of sector returns

<table>
<thead>
<tr>
<th>ticker</th>
<th>$t_1$</th>
<th>$t_2$</th>
<th>\ldots</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAPL</td>
<td>.05</td>
<td>-.21</td>
<td>\ldots</td>
</tr>
<tr>
<td>KRX</td>
<td>.07</td>
<td>-.18</td>
<td>\ldots</td>
</tr>
<tr>
<td>GOOG</td>
<td>-.11</td>
<td>.24</td>
<td>\ldots</td>
</tr>
<tr>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
</tr>
</tbody>
</table>

- rows of $Y$ are sector return time series
- rows of $X$ are sector exposures
Low rank models for power

**electricity usage profiles**

<table>
<thead>
<tr>
<th>household</th>
<th>$t_1$</th>
<th>$t_2$</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.4</td>
<td>0.5</td>
<td>0.1</td>
</tr>
<tr>
<td>2</td>
<td>2.7</td>
<td>1.3</td>
<td>0.9</td>
</tr>
<tr>
<td>3</td>
<td>3.3</td>
<td>4.2</td>
<td>1.8</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

- rows of $Y$ are electricity usage profiles
- rows of $X$ decompose household power usage into distinct usage profiles
Regularizers

\[
\text{minimize } \sum_{(i,j) \in \Omega} \ell_j(Y_{ij}, x_i^T w_j) + \sum_{i=1}^n r_i(x_i) + \sum_{j=1}^d \tilde{r}_j(w_j)
\]

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>( \mathbf{1}(x \geq 0) )</td>
<td>( \mathbf{1}(y \geq 0) )</td>
</tr>
</tbody>
</table>
Nonnegative matrix factorization

\[
\text{minimize} \quad \sum_{(i,j)\in\Omega} (Y_{ij} - x_i^T w_j)^2 + \sum_{i=1}^{n} 1_+(x_i) + \sum_{j=1}^{d} 1_+(w_j)
\]

- regularizer is indicator of nonnegative orthant

\[
1_+(x) = \begin{cases} 
0 & x \geq 0 \\
\infty & \text{otherwise}
\end{cases}
\]
Nonnegative matrix factorization

minimize \[ \sum_{(i,j) \in \Omega} (Y_{ij} - x_i^T w_j)^2 + \sum_{i=1}^{n} 1_+(x_i) + \sum_{j=1}^{d} 1_+(w_j) \]

- regularizer is indicator of nonnegative orthant

\[ 1_+(x) = \begin{cases} 0 & x \geq 0 \\ \infty & \text{otherwise} \end{cases} \]

subproblems are nonnegative least squares problems:

\[ x_i^{t+1} = \arg\min_{x > 0} \sum_{j: (i,j) \in \Omega} (Y_{ij} - x^T w_j^t)^2 \quad (1) \]

\[ w_j^{t+1} = \arg\min_{w > 0} \sum_{i: (i,j) \in \Omega} (Y_{ij} - (x_i^{t+1})^T w)^2 \quad (2) \]
Outline

Missing data
Unsupervised learning
Low rank models
Principal Components Analysis
Generalized Low Rank Models
Imputing missing data
Multidimensional losses
Exotic regularizers

Clustering
Clustering

A clustering algorithm groups data points into clusters.

Examples:

- **Medical diagnosis.** Cluster patients with similar medical histories.
- **Topic model.** Cluster documents with similar patterns of word usage.
- **Market segmentation.** Cluster customers with similar purchase patterns.
the $k$-means problem:

- given data points $y_i \in \mathbb{R}^d$, $i = 1, \ldots, n$
- find $k$ centers $w_l \in \mathbb{R}^d$, $l = 1, \ldots, k$
- and assignments $c_i \in \{1, \ldots, k\}$, $i = 1, \ldots, n$
- to minimize

$$\sum_{i=1}^{n} \|y_i - w_{c_i}\|^2$$
Lloyd’s algorithm for $k$-means

Lloyd’s algorithm (aka the $k$-means algorithm): to minimize

$$\sum_{i=1}^{n} \| y_i - w_{c_i} \|^2,$$

repeat

1. assign points to centers

$$c_i = \arg\min_{l=1,\ldots,k} \| y_i - w_l \|^2, \quad i = 1, \ldots, n$$

2. update centers: let $C_l = \{ i : c_i = l \}$ be points assigned to cluster $l$, and set

$$w_l = \frac{1}{|C_l|} \sum_{i \in C_l} y_i, \quad l = 1, \ldots, k$$

visualizing the algorithm:

http://stanford.edu/class/ee103/visualizations/kmeans/kmeans.html
Lloyd’s algorithm for $k$-means

Lloyd’s algorithm (aka the $k$-means algorithm): to minimize

$$\sum_{i=1}^{n} \|y_i - w_{c_i}\|^2,$$

repeat

1. assign points to centers

$$c_i = \arg\min_{l=1,\ldots,k} \|y_i - w_l\|^2, \quad i = 1, \ldots, n$$

2. update centers

$$w_l = \frac{1}{|C_l|} \sum_{i \in C_l} y_i = \arg\min_{l=1,\ldots,k} \sum_{i : c_i = l} \|y_i - w\|^2, \quad l = 1, \ldots, k$$
Quadratic clustering

minimize \[ \sum_{(i,j) \in \Omega} (Y_{ij} x_i^T w_j)^2 + \sum_{i=1}^{n} 1_1(x_i) \]

\(1_1\) is the indicator function of a selection, i.e.,

\[ 1_1(x) = \begin{cases} 
0 & x = e_l \text{ for some } l \in \{1, \ldots, k\} \\
\infty & \text{otherwise}
\end{cases} \]

where \(e_l\) is the \(l\)th unit vector
Quadratic clustering

$$\text{minimize} \quad \sum_{(i,j) \in \Omega} (Y_{ij} x_i^T w_j)^2 + \sum_{i=1}^{n} 1(x_i)$$

- $1_{i}$ is the indicator function of a selection, i.e.,

$$1_{i}(x) = \begin{cases} 
0 & x = e_l \text{ for some } l \in \{1, \ldots, k\} \\
\infty & \text{otherwise}
\end{cases}$$

where $e_l$ is the $l$th unit vector

alternating minimization reproduces $k$-means (but allows missing data)
Check AM reproduces $k$-means

let $w^l$ be $l$th row of $W$, $l = 1, \ldots, k$

\[
\sum_{(i,j) \in \{1, \ldots, n\} \times \{1, \ldots, d\}} (Y_{ij} - x_i^T w_j)^2 = \sum_{l=1}^{k} \sum_{i \in C_l} \sum_{j=1}^{d} (Y_{ij} - x_i^T w_j)^2
\]

\[
= \sum_{l=1}^{k} \sum_{i \in C_l} \sum_{j=1}^{d} (Y_{ij} - e_l w_j)^2
\]

\[
= \sum_{l=1}^{k} \sum_{i \in C_l} \sum_{j=1}^{d} (Y_{ij} - w^l_j)^2
\]

\[
= \sum_{l=1}^{k} \sum_{i \in C_l} \| Y_i - w^l \|^2
\]

- to minimize over $W$: set $w^l$ to be the mean of $Y_i$ for $i \in C_l$

$$w^l = \frac{1}{|C_l|} \sum_{i \in C_l} Y_i$$
Check AM reproduces $k$-means

let $w^l$ be $l$th row of $W$, $l = 1, \ldots, k$

\[
\sum_{(i,j)\in\{1,\ldots,n\}\times\{1,\ldots,d\}} (Y_{ij} - x_i^T w_j)^2 = \sum_{l=1}^{k} \sum_{i \in C_l} \sum_{j=1}^{d} (Y_{ij} - x_i^T w_j)^2
\]

\[
= \sum_{l=1}^{k} \sum_{i \in C_l} \sum_{j=1}^{d} (Y_{ij} - e_l w_j)^2
\]

\[
= \sum_{l=1}^{k} \sum_{i \in C_l} \sum_{j=1}^{d} (Y_{ij} - w_j^l)^2
\]

\[
= \sum_{l=1}^{k} \sum_{i \in C_l} \|Y_i - w^l\|^2
\]

▶ to minimize over $X$: set $x_i$ to be the unit vector $e_l$

\[
x_i = e_l \quad \text{where} \quad l = \text{argmin} \ l' \in 1, \ldots, k \|Y_i - w^l\|^2
\]
What’s a cluster?
Modifying *k*-means

different regularizers:

- clusters
- rays
- lines
- planes
- cones
Modifying \( k \)-means

different regularizers:

- clusters
- rays
- lines
- planes
- cones

different losses:

- \( k \)-means: \( \ell(y, z) = (y - z)^2 \)
- \( k \)-medioids: \( \ell(y, z) = |y - z| \)
- \( \ell(y, z) = \text{huber}(y - z) \)
- \ldots \)
Fitting GLRM}s with alternating minimization

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

repeat:

1. minimize objective over \( x_i \) (in parallel)
2. minimize objective over \( w_j \) (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 $X^0, W^0$

for $t = 1, 2, \ldots$ do

  for $i = 1, \ldots, m$ do
    $x_i^t = \text{update}_{L,r}(x_i^{t-1}, W^{t-1}, Y)$

  for $j = 1, \ldots, n$ do
    $w_j^t = \text{update}_{L,\tilde{r}}(w_j^{(t-1)T}, X(t)^T, Y^T)$

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

proximal gradient method: let

\[ g = \sum_{j:(i,j) \in \Omega} \nabla \ell_j(x_i w_j, Y_{ij}) w_j \]

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 \( \text{prox}_r \)
- **stochastic variant:** use noisy estimate for \( g \)
- **time per iteration:** \( O\left(\frac{(n+d+|\Omega|)k}{p}\right) \) on \( p \) processors
Recap: GLRM}s

Generalized Low Rank Models are a framework that encompasses a bunch of unsupervised learning models.

many of these GLRM}s have names:

<table>
<thead>
<tr>
<th>Model</th>
<th>$\ell(y, z)$</th>
<th>$r(x)$</th>
<th>$\tilde{r}(w)$</th>
<th>reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA</td>
<td>$(y - z)^2$</td>
<td>0</td>
<td>0</td>
<td>[Pearson 1901]</td>
</tr>
<tr>
<td>NNMF</td>
<td>$(y - z)^2$</td>
<td>$1_+(x)$</td>
<td>$1_+(w)$</td>
<td>[Lee 1999]</td>
</tr>
<tr>
<td>sparse PCA</td>
<td>$(y - z)^2$</td>
<td>$|x|_1$</td>
<td>$|w|_1$</td>
<td>[D’Aspremont 2004]</td>
</tr>
<tr>
<td>sparse coding</td>
<td>$(y - z)^2$</td>
<td>$|x|_1$</td>
<td>$|w|_2$</td>
<td>[Olshausen 1997]</td>
</tr>
<tr>
<td>k-means</td>
<td>$(y - z)^2$</td>
<td>$1_1(x)$</td>
<td>0</td>
<td>[Tropp 2004]</td>
</tr>
<tr>
<td>matrix completion</td>
<td>$(y - z)^2$</td>
<td>$|x|_2$</td>
<td>$|w|_2$</td>
<td>[Keshavan 2010]</td>
</tr>
<tr>
<td>robust PCA</td>
<td>$</td>
<td>y - z</td>
<td>$</td>
<td>$|x|_2$</td>
</tr>
<tr>
<td>logistic PCA</td>
<td>$\log(1 + \exp(-yz))$</td>
<td>$|x|_2$</td>
<td>$|w|_2$</td>
<td>[Collins 2001]</td>
</tr>
<tr>
<td>boolean PCA</td>
<td>$(1 - yz)_+$</td>
<td>$|x|_2$</td>
<td>$|w|_2$</td>
<td>[Srebro 2004]</td>
</tr>
</tbody>
</table>
Resources

- GLRMds
  https://people.orie.cornell.edu/mru8/doc/udell16_glrms.pdf

- fitting GLRMs
  https://github.com/madeleineudell/LowRankModels.jl