ORIE 4741: Learning with Big Messy Data

Unsupervised Learning

Professor Udell

Operations Research and Information Engineering
Cornell

November 16, 2021

Announcements 11/9/21

- section this week: unsupervised learning
- hw5 out, due Thursday Nov 18 9:30am
- project midterm report peer reviews due Sunday

Announcements 11/16/21

- section this week: autoML in practice
- hw5 due Thursday Nov 18 9:30am

Outline

Missing data

examples:

Missing data

examples:

- weather data: missing data due to

Missing data

examples:

- weather data: missing data due to sensor failures

Missing data

examples:

- weather data: missing data due to sensor failures
- survey data: missing data due to

Missing data

examples:

- weather data: missing data due to sensor failures
- survey data: missing data due to non-response

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

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

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

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

Review: handling missing values

- remove rows/columns with missing entries

Review: handling missing values

- remove rows/columns with missing entries
- (for time series) back-fill with most recent observed value

Review: handling missing values

- remove rows/columns with missing entries
- (for time series) back-fill with most recent observed value
- impute with mean, median, or mode

Review: handling missing values

- remove rows/columns with missing entries
- (for time series) back-fill with most recent observed value
- impute with mean, median, or mode
- fancier imputation methods: matrix completion, copula models, deep learning, ...

Review: handling missing values

- remove rows/columns with missing entries
- (for time series) back-fill with most recent observed value
- impute with mean, median, or mode
- fancier imputation methods: matrix completion, copula models, deep learning, ...
- add new feature: Boolean indicator $\mathbb{1}$(data is missing)
  - can detect if missingness is informative
  - can complement imputation method
  - can use different indicators for different kinds of missingness (refused, missing, illegible response, ...)

Outline

Unsupervised learning for missing value imputation

Matrix completion for missing value imputation:

- 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)

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

- ith row of Y is feature vector for ith example
- jth column of Y gives values for jth feature across all examples

Outline

Low rank model

given: $n \times d$ data table Y, $r \leq n, d$
find: $X \in \mathbf{R}^{n\times r}$, $W \in \mathbf{R}^{r\times d}$ 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

$$\begin{bmatrix} \\ X \\ \\ \end{bmatrix} = \begin{bmatrix} \text{—}x_1^T\text{—} \\ \vdots \\ \text{—}x_n^T\text{—} \end{bmatrix} \qquad \begin{bmatrix} & W & \end{bmatrix} = \begin{bmatrix} | & & | \\ w_1 & \cdots & w_d \\ | & & | \end{bmatrix}$$

interpretation:

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

Exact low rank fitting

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

Q: what is the smallest r so that

$$Y = XW$$

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

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

Exact low rank fitting

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

Q: what is the smallest r so that

$$Y = XW$$

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

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

A: $r = \mathrm{Rank}(Y)$!

Exact low rank fitting

theorem: for $Y \in \mathbf{R}^{n \times d}$,

$$\mathrm{Rank}(Y) = \min\{r : Y = XW, \qquad X \in \mathbf{R}^{n \times r},\ W \in \mathbf{R}^{r \times d}\}$$

Exact low rank fitting

theorem: for $Y \in \mathbf{R}^{n \times d}$,

$$\mathrm{Rank}(Y) = \min\{r : Y = XW, \qquad X \in \mathbf{R}^{n \times r},\ W \in \mathbf{R}^{r \times d}\}$$

proof: 1) we can find X and W with $Y = XW$ and $r = \mathrm{Rank}(Y)$:

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

Exact low rank fitting

theorem: for $Y \in \mathbf{R}^{n \times d}$,

$$\mathrm{Rank}(Y) = \min\{r : Y = XW, \qquad X \in \mathbf{R}^{n \times r},\ W \in \mathbf{R}^{r \times d}\}$$

proof: 1) we can find X and W with $Y = XW$ and $r = \mathrm{Rank}(Y)$:

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

2) for any X and W st $Y = XW$, $\mathrm{Rank}(Y) \leq r$:

- $\mathrm{Rank}(Y) = \mathrm{Rank}(XW) \leq \min(\mathrm{Rank}(X), \mathrm{Rank}(W)) \leq r$

Exact low rank fitting

theorem: for $Y \in \mathbf{R}^{n \times d}$,

$$\mathrm{Rank}(Y) = \min\{r : Y = XW, \qquad X \in \mathbf{R}^{n \times r},\ W \in \mathbf{R}^{r \times d}\}$$

proof: 1) we can find X and W with $Y = XW$ and $r = \mathrm{Rank}(Y)$:

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

2) for any X and W st $Y = XW$, $\mathrm{Rank}(Y) \le r$:

- $\mathrm{Rank}(Y) = \mathrm{Rank}(XW) \le \min(\mathrm{Rank}(X), \mathrm{Rank}(W)) \le r$

so $\mathrm{Rank}(Y)$ is the smallest r st $Y = XW$

Inexact low rank fitting

if we're willing to represent Y approximately,
can we use a smaller rank r?

Outline

Principal components analysis (PCA)

Principal components analysis (PCA): Given $Y \in \mathbf{R}^{n \times d}$, 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 $X \in \mathbf{R}^{n \times r}$, $W \in \mathbf{R}^{r \times d}$

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

Principal components analysis (PCA)

Principal components analysis (PCA): Given $Y \in \mathbf{R}^{n \times d}$, 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 $X \in \mathbf{R}^{n \times r}$, $W \in \mathbf{R}^{r \times d}$

- 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 $T \in \mathbf{R}^{r \times r}$:

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

Principal components analysis (PCA)

Principal components analysis (PCA): Given $Y \in \mathbf{R}^{n \times d}$, 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 $X \in \mathbf{R}^{n \times r}$, $W \in \mathbf{R}^{r \times d}$

- 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 $T \in \mathbf{R}^{r \times r}$:

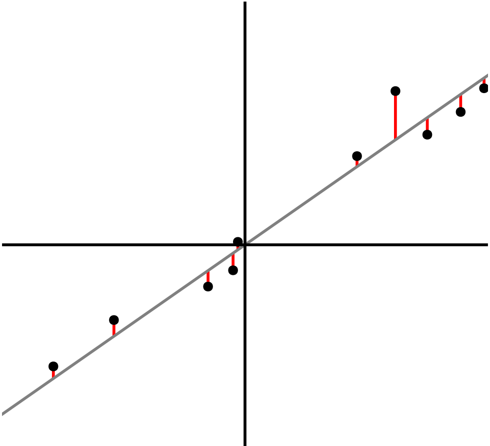
$$\tilde{X}\tilde{W} = XTT^{-1}W = XW.$$

make sure **interpretation** of solution is invariant under T

PCA finds best covariates

example with $d = 2$, $r = 1$
regression: fix $X = Y_{:,1}$ (first column of Y), solve

$$\text{minimize} \quad \|Y - XW\|_F^2 \qquad \text{wrt variable } W \in \mathbf{R}^{1\times 2}$$

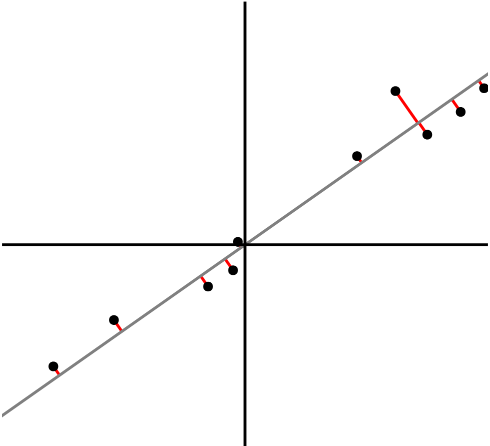


PCA finds best covariates

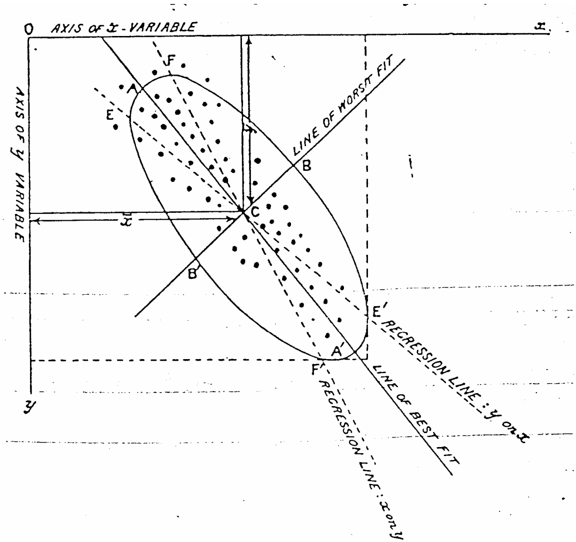
example with $d = 2$, $r = 1$

PCA: solve

$$\text{minimize} \quad \|Y - XW\|_F^2 \qquad \text{wrt variables } X \in \mathbf{R}^{n\times 1}, W \in \mathbf{R}^{1\times 2}$$



On lines and planes of best fit



[Pearson 1901]

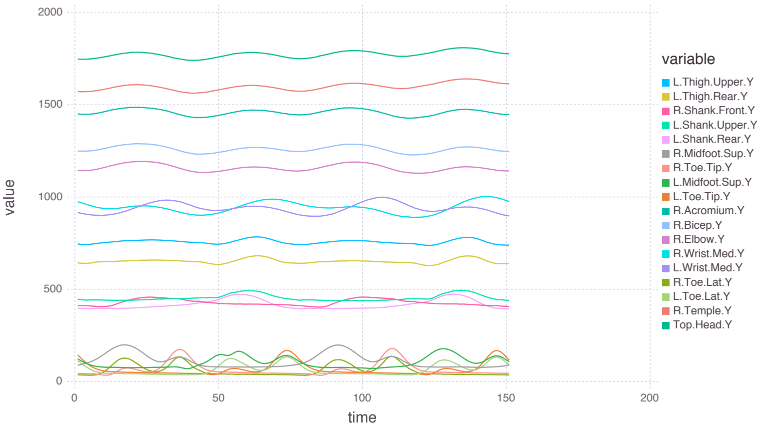
Low rank models for gait analysis

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

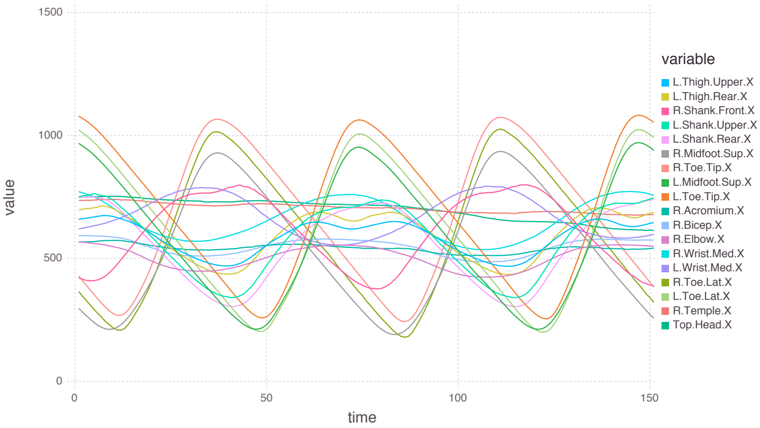
Interpreting principal components

columns of Y (features) (height of point over time)



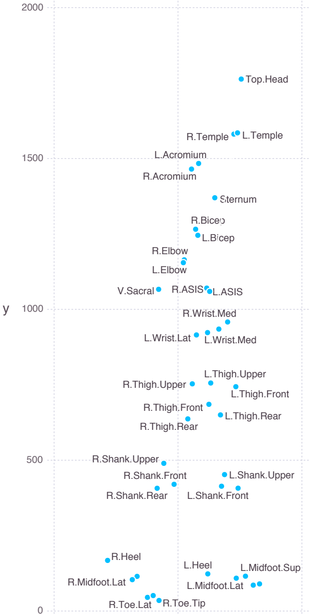
Interpreting principal components

columns of Y (features) (depth of point over time)



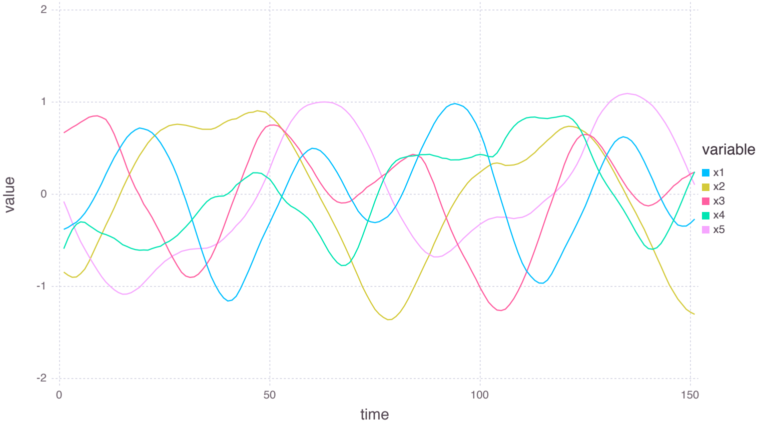
Interpreting principal components

row of W
(archetypical example)
(principal stance)



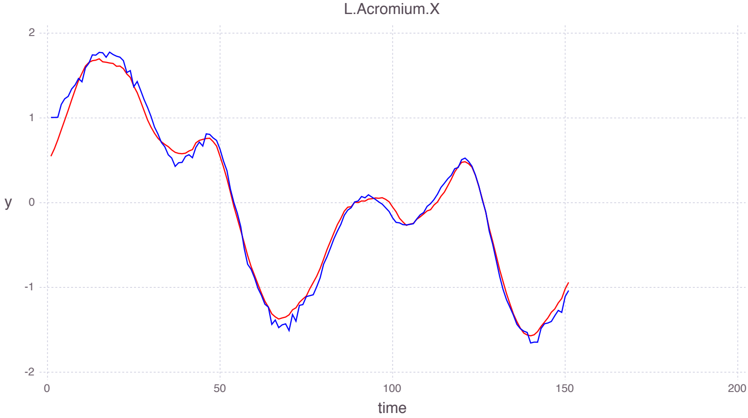
Interpreting principal components

columns of X (archetypical features) (principal timeseries)



Interpreting principal components

column of XW (red) (predicted feature)
column of Y (blue) (observed feature)



Principal components analysis (PCA)

Principal components analysis (PCA): Given $Y \in \mathbf{R}^{n \times d}$, 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 $X \in \mathbf{R}^{n \times r}$, $W \in \mathbf{R}^{r \times d}$

how should we solve this problem?

Principal components analysis (PCA)

Principal components analysis (PCA): Given $Y \in \mathbf{R}^{n \times d}$, 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 $X \in \mathbf{R}^{n \times r}$, $W \in \mathbf{R}^{r \times d}$

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:

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

The Frobenius norm

the **Frobenius norm**

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

some useful identities:

- $\|A\|_F = \|\mathrm{vec}(A)\|$
- $\|A\|_F = \|A^T\|_F$
- $\|A\|_F^2 = \mathbf{tr}(A^T A)$
- if U is orthogonal (*i.e.*, $U^T U = I$), then $\|UA\|_F = \|A\|_F$
  proof:

$$\|UA\|_F^2 = \mathbf{tr}((UA)^T UA) = \mathbf{tr}(A^T U^T UA) = \mathbf{tr}(A^T A) = \|A\|_F^2$$

PCA: solution via the SVD

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

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

Eckart-Young-Mirsky theorem: if

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

is the SVD of Y, then

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

is a solution to PCA, where

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

PCA: solution via the SVD

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

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

is the SVD of Y, then

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

is a solution to PCA, where

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

with this X and W,

$$\|Y - XW\|_F^2 = \|U \Sigma V^T - U_r \Sigma_r V_r^T\|_F^2 = \sum_{i=r+1}^{\mathrm{Rank}(Y)} \sigma_i^2$$

Proof of Eckart-Young-Mirsky theorem I

proof step 1: reduce to diagonal.
if $Y = U\Sigma V^T$ is the full SVD, then

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

so

$$\begin{aligned} \|Y - XW\|_F^2 &= \|U\Sigma V^T - XW\|_F^2 \\ &= \|U^T U\Sigma V^T V - U^T XWV\|_F^2 \\ &= \|\Sigma - U^T XWV\|_F^2 \\ &= \|\Sigma - Z\|_F^2 \end{aligned}$$

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

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

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,$B \in \mathbf{R}^{n \times d}$,

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

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

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

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

$$\|\Sigma - \Sigma_r\|_F^2 = \sum_{i=r+1}^{\mathrm{Rank}(Y)} \sigma_i^2 \leq \sum_{i=1}^{\mathrm{Rank}(Y)-r} \sigma_i^2(\Sigma - Z) \leq \|\Sigma - Z\|_F^2.$$

PCA: solution via AM

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

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

- $X^t = \operatorname{argmin}_X \|Y - XW^{t-1}\|_F^2$
- $W^t = \operatorname{argmin}_W \|Y - X^t W\|_F^2$

properties:

- objective decreases at each iteration
- objective bounded below, so the procedure converges
- (it is true but we won't prove that) with probability 1 over choices of W^0, AM converges to an optimal solution

PCA: AM subproblem is separable

how would you solve the AM subproblem

$$W^t = \operatorname*{argmin}_W \|Y - X^t W\|_F^2 = \operatorname*{argmin}_W \sum_{j=1}^{d} \|y_j - X^t w_j\|^2$$

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

$$W^t = \operatorname*{argmin}_W \|Y - X^t W\|_F^2 = \operatorname*{argmin}_W \sum_{j=1}^{d} \|y_j - X^t w_j\|^2$$

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

- problem separates over columns of W:

$$w_j^t = \operatorname*{argmin}_w \|y_j - X^t w\|^2$$

- for each column of W, it's just a least squares problem!
- $w_j = ((X^t)^T X^t)^{-1} (X^t)^T y_j$

PCA: solution via AM

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

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

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

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 $j = 1, \ldots, d$,

$$w_j^t = ((X^t)^T X^t)^{-1} (X^t)^T y_j$$

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 $t = 1, \ldots,$

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 $G = W^{t-1}(W^{t-1})^T \quad (\mathcal{O}(dr^2 + r^3))$

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

$$x_i^t = (W^{t-1}(W^{t-1})^T)^{-1} W^{t-1} Y_{i:}^T$$

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

$$x_i^t = (W^{t-1}(W^{t-1})^T)^{-1} W^{t-1} Y_{i:}^T$$

- cache factorization of $G = (X^t)^T X^t$ $\quad (\mathcal{O}(nr^2 + r^3))$

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

$$x_i^t = (W^{t-1}(W^{t-1})^T)^{-1} W^{t-1} Y_{i:}^T$$

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

$$w_j^t = ((X^t)^T X^t)^{-1} (X^t)^T y_j$$

Outline

Missing data?

now suppose we observe Y_{ij} only for
$(i, j) \in \Omega \subset \{1, \ldots, n\} \times \{1, \ldots, d\}$

Missing data?

now suppose we observe Y_{ij} only for $(i,j) \in \Omega \subset \{1,\ldots,n\} \times \{1,\ldots,d\}$

Matrix completion:

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

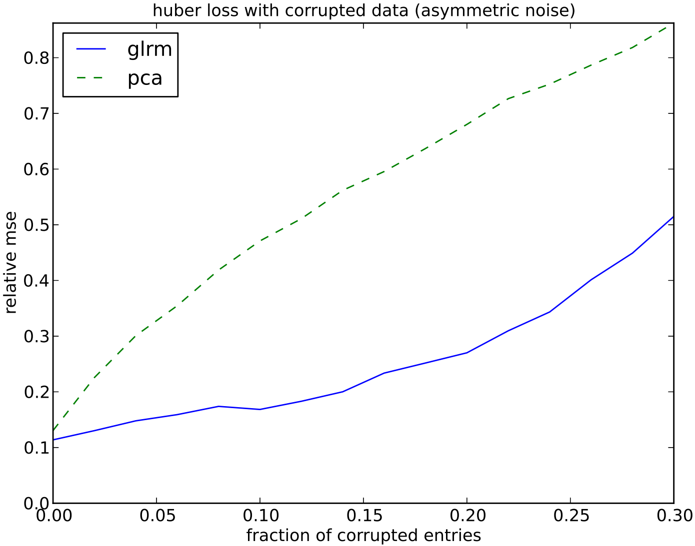
two regimes:

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

Huber PCA

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

Huber PCA



huber loss with corrupted data (asymmetric noise)
glrm
pca
relative mse
fraction of corrupted entries
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.00
0.05
0.10
0.15
0.20
0.25
0.30

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 ℓ_j for each column
  - assume $Y_{ij} \in \mathcal{Y}_j$ for every $(i,j) \in \Omega$
  - $\ell_j : \mathcal{Y}_j \times \mathbf{R} \to \mathbf{R}$
  - *e.g.*, different losses for reals, booleans, categoricals, ordinals, ...
- regularizers $r : \mathbf{R}^{1\times r} \to \mathbf{R}$, $\tilde{r} : \mathbf{R}^r \to \mathbf{R}$

Losses

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

choose loss $L : \mathbf{R} \times \mathcal{F} \to \mathbf{R}$ adapted to data type $\mathcal{F}$:

data type	**loss**	$L(u, a)$
real	`QuadLoss`	$(u-a)^2$
real	`L1Loss`	$\lvert u-a \rvert$
real	`HuberLoss`	$\mathbf{huber}(u-a)$
boolean	`HingeLoss`	$(1-ua)_+$
boolean	`LogisticLoss`	$\log(1+\exp(-au))$
ordinal	`BvSLoss`	$\sum_{a'=1}^{d}(1-u(a \geq a'))_+$
ordinal	`OrdinalHingeLoss`	$\sum_{a'=1}^{a-1}(1-u+a')_+ + \sum_{a'=a+1}^{d}(1+u-a')_+$
categorical	`OvALoss`	$(1-u_a)_+ + \sum_{a'\neq a}(1+u_{a'})_+$
categorical	`MultinomialLoss`	$\frac{\exp(u_a)}{\sum_{a'=1}^{d}\exp(u_{a'})}$

Regularizers

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

choose regularizers r_x, r_y to impose structure:

structure	r_x	r_y
small	`QuadReg`	`QuadReg`
sparse	`OneReg`	`OneReg`
nonnegative	`NonNegConstraint`	`NonNegConstraint`
clustered	`UnitOneSparseConstraint`	`ZeroReg`

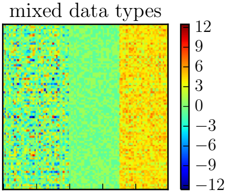
Impute missing data

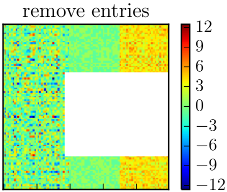
impute most likely true data $\hat{A}_{ij}$

$$\hat{A}_{ij} = \operatorname*{argmin}_a L_j(x_i y_j, a)$$

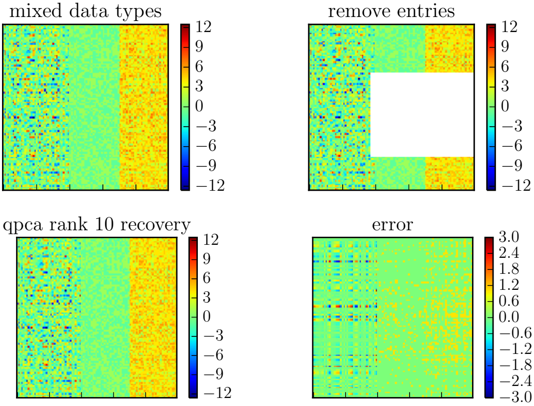
- implicit constraint: $\hat{A}_{ij} \in \mathcal{F}_j$
- when L_j is quadratic, ℓ_1, or Huber loss, then $\hat{A}_{ij} = x_i y_j$
- if $\mathcal{F} \neq \mathbf{R}$, $\operatorname{argmin}_a L_j(x_i y_j, a) \neq x_i y_j$
  - *e.g.*, for hinge loss $L(u, a) = (1 - ua)_+$, $\hat{A}_{ij} = \mathbf{sign}(x_i y_j)$

Impute heterogeneous data

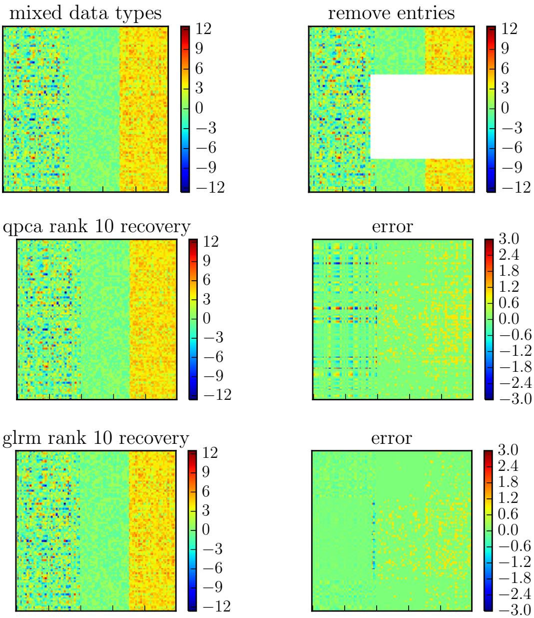




Impute heterogeneous data



Impute heterogeneous data



Julia implementation: demo

example: fit rank 5 GLRM in 2 lines of code:

```
glrm = GLRM(A, 5, datatypes)
X,Y = fit!(glrm)
```

Validate model

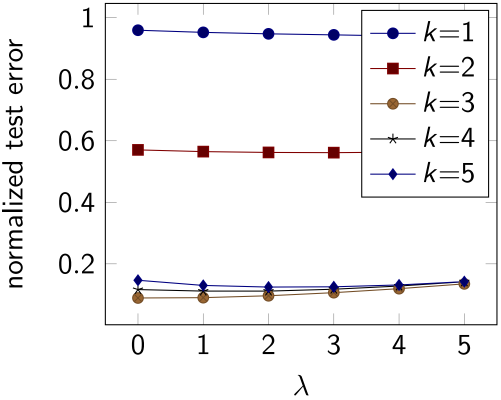
$$\text{minimize} \quad \sum_{(i,j)\in\Omega} L_{ij}(A_{ij}, x_i y_j) + \lambda \left(\sum_{i=1}^{m} r_x(x_i) + \sum_{j=1}^{n} r_y(y_j) \right)$$

How to choose model parameters (k, λ)?

Validate model

$$\text{minimize} \quad \sum_{(i,j)\in\Omega} L_{ij}(A_{ij}, x_i y_j) + \lambda \left(\sum_{i=1}^m r_x(x_i) + \sum_{j=1}^n r_y(y_j)\right)$$

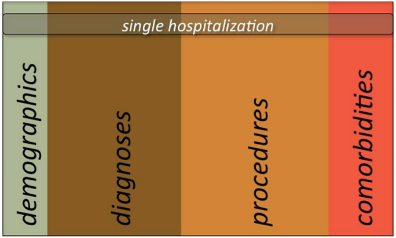
How to choose model parameters (k, λ)?
Leave out 10% of entries, and use model to predict them



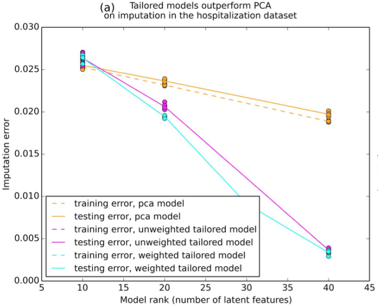
Outline

Hospitalizations are low rank

hospitalization data set



GLRM outperforms PCA



[Schuler et al., 2016]

Impute censored data

market segmentation

customer	apples	oranges	pears	$\cdots$
1	yes	?	yes	$\cdots$
2	yes	yes	?	$\cdots$
3	?	?	yes	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	

- 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, $p \in \mathbf{R}^{300\times 300}$

customer	apples	oranges	pears	$\cdots$
1	.28	.22	.76	$\cdots$
2	.97	.55	.36	$\cdots$
3	.13	.47	.62	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	

Impute censored data

synthetic data:

- entry (i, j) is $+$ with probability p_{ij}

customer	apples	oranges	pears	$\cdots$
1	$+$	-	$+$	$\cdots$
2	$+$	$+$	-	$\cdots$
3	-	$+$	$+$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	

Impute censored data

synthetic data:

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

customer	apples	oranges	pears	$\cdots$
1	$+$	?	$+$	$\cdots$
2	$+$	$+$	?	$\cdots$
3	?	$+$	$+$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	

Impute censored data

synthetic data:

- ...and we only observe 10% of the $+$s

customer	apples	oranges	pears	$\cdots$
1	$+$	?	?	$\cdots$
2	?	$+$	?	$\cdots$
3	?	?	?	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	

Impute censored data

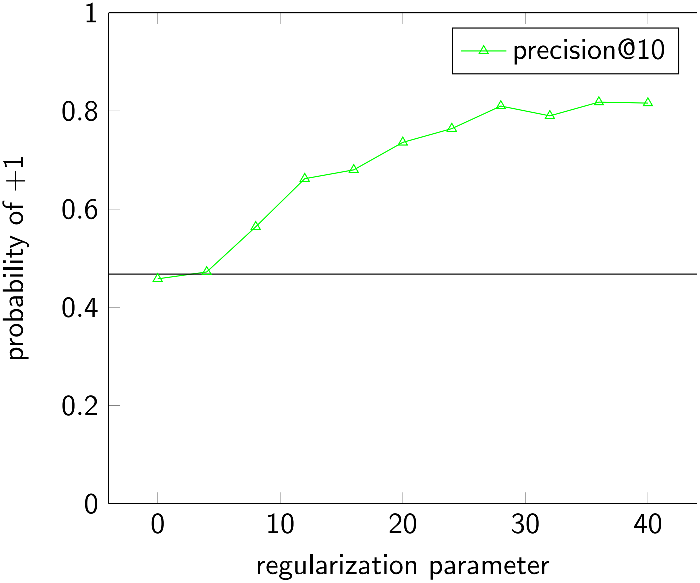
synthetic data:

- ... and we only observe 10% of the $+$s

customer	apples	oranges	pears	$\cdots$
1	$+$	?	?	$\cdots$
2	?	$+$	?	$\cdots$
3	?	?	?	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	

can we predict 10 more $+$s?

Impute censored data



Outline

Multi-dimensional loss

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

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

- 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

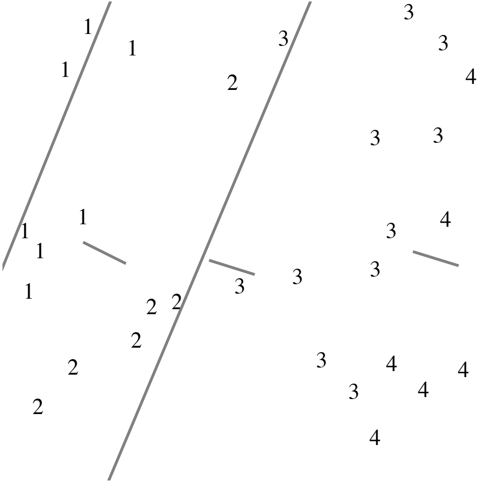
- choose any loss function for multiclass classification to penalize $x_i Y$
  - *e.g.*, one-vs-all (elementwise hinge loss) [Rifkin 2004]

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

$$\begin{array}{ccccc} \text{CA} & \text{NV} & \cdots & \text{PA} & \text{NY} \\ \hline \text{T} & \text{F} & \cdots & \text{F} & \text{F} \\ \text{F} & \text{F} & \cdots & \text{T} & \text{F} \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{array} \approx \begin{bmatrix} \text{---}x_1\text{---} \\ \vdots \\ \text{---}x_m\text{---} \end{bmatrix} \begin{bmatrix} | & | & & | & | \\ \text{CA} & \text{NV} & \cdots & \text{PA} & \text{NY} \\ | & | & & | & | \end{bmatrix}$$

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:

$$\begin{aligned}\mu_j &= \operatorname*{argmin}_{\mu} \sum_{i:(i,j)\in\Omega} \ell_j(\mu, Y_{ij}) \\ \sigma_j^2 &= \frac{1}{n_j - 1} \sum_{i:(i,j)\in\Omega} \ell_j(\mu_j, Y_{ij})\end{aligned}$$

- n_j is number of observations in column j
- μ_j generalizes column mean
- σ_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:

$$\begin{aligned}\mu_j &= \operatorname*{argmin}_{\mu} \sum_{i:(i,j)\in\Omega} \ell_j(\mu, Y_{ij}) \\ \sigma_j^2 &= \frac{1}{n_j - 1} \sum_{i:(i,j)\in\Omega} \ell_j(\mu_j, Y_{ij})\end{aligned}$$

- n_j is number of observations in column j
- μ_j generalizes column mean
- σ_j^2 generalizes column variance

To fit a standardized GLRM, solve

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

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

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

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

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

- cluster respondents?

Application: exploratory data analysis

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

$$\begin{array}{cccc} \text{age} & \text{gender} & \text{state} & \cdots \\ \hline 29 & \text{F} & \text{CT} & \cdots \\ 57 & ? & \text{NY} & \cdots \\ ? & \text{M} & \text{CA} & \cdots \\ 41 & \text{F} & \text{NV} & \cdots \\ \vdots & \vdots & \vdots & \end{array} \approx \begin{bmatrix} —x_1^T— \\ \vdots \\ —x_n^T— \end{bmatrix}$$

- cluster respondents? **cluster rows of** X

Application: exploratory data analysis

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

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

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

- cluster respondents? **cluster rows of X**
- demographic profiles?

Application: exploratory data analysis

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

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

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

- cluster respondents? **cluster rows of** X
- demographic profiles? **rows of** W

Application: exploratory data analysis

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

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

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

- cluster respondents? **cluster rows of** X
- demographic profiles? **rows of** W
- which features are similar?

Application: exploratory data analysis

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

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

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

- cluster respondents? **cluster rows of** X
- demographic profiles? **rows of** W
- which features are similar? **cluster columns of** W

Application: exploratory data analysis

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

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

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

- cluster respondents? **cluster rows of** X
- demographic profiles? **rows of** W
- which features are similar? **cluster columns of** W
- impute missing entries?

Application: exploratory data analysis

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

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

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

- cluster respondents? **cluster rows of** X
- demographic profiles? **rows of** W
- which features are similar? **cluster columns of** W
- impute missing entries? $\mathrm{argmin}_{y \in \mathcal{Y}_j} \ell_j(y, x_i^T w_j)$

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:

```
glrm, labels = GLRM(Y, 10, scale = true)
X,W = fit!(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[1]

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

company	zip	violations	$\cdots$
Holiday Inn	14850	109	$\cdots$
Moosewood Restaurant	14850	0	$\cdots$
Cornell Orchards	14850	0	$\cdots$
Lakeside Nursing Home	14850	53	$\cdots$
$\vdots$	$\vdots$	$\vdots$	

- 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`

Low rank models for dimensionality reduction

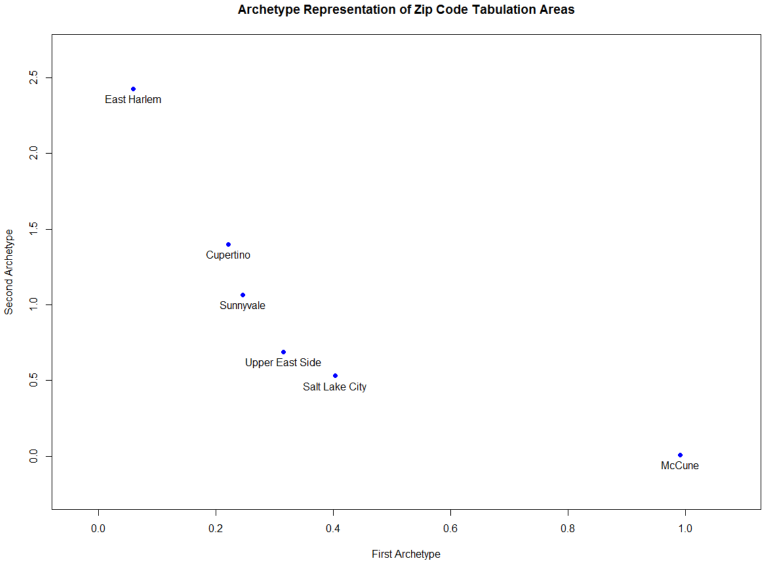
ACS demographic data:

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

recap: why use GLRMs?

use GLRMs to

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

Outline

Low rank models for finance

factor model of sector returns

ticker	t_1	t_2	$\cdots$
AAPL	.05	-.21	$\cdots$
KRX	.07	-.18	$\cdots$
GOOG	-.11	.24	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\ddots$

- rows of Y are sector return time series
- rows of X are sector exposures

Low rank models for power

electricity usage profiles

household	t_1	t_2	$\cdots$	
1	1.4	0.5	0.1	$\cdots$
2	2.7	1.3	0.9	$\cdots$
3	3.3	4.2	1.8	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

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

Regularizers

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

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	$\mathbf{1}(x \geq 0)$	$\mathbf{1}(y \geq 0)$

Nonnegative matrix factorization

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

- regularizer is indicator of nonnegative orthant

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

Nonnegative matrix factorization

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

- regularizer is indicator of nonnegative orthant

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

subproblems are nonnegative least squares problems:

$$x_i^{t+1} = \operatorname*{argmin}_{x>0} \sum_{j:(i,j)\in\Omega} (Y_{ij} - x^T w_j^t)^2 \tag{1}$$

$$w_j^{t+1} = \operatorname*{argmin}_{w>0} \sum_{i:(i,j)\in\Omega} (Y_{ij} - (x_i^{t+1})^T w)^2 \tag{2}$$

Outline

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

k-means

the k-means problem:

- given data points $y_i \in \mathbf{R}^d$, $i = 1, \ldots, n$
- find k centers $w_l \in \mathbf{R}^d$, $l = 1, \ldots, k$
- 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

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

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

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

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

$$c_i = \operatorname*{argmin}_{l=1,\ldots,k} \|y_i - w_l\|^2, \quad i = 1, \ldots, n$$

2. update centers

$$w_l = \frac{1}{|\mathcal{C}_l|} \sum_{i \in \mathcal{C}_l} y_i = \operatorname*{argmin}_{l=1,\ldots,k} \sum_{i : c_i = l} \|y_i - w\|^2, \quad l = 1, \ldots, k$$

Quadratic clustering

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

- $\mathbf{1}_1$ is the indicator function of a selection, *i.e.*,

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

where e_l is the lth unit vector

Quadratic clustering

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

- $\mathbf{1}_1$ is the indicator function of a selection, *i.e.*,

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

where e_l is the lth unit vector

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

Check AM reproduces k-means

let w^l be lth row of W, $l = 1, \dots, k$

$$
\begin{aligned}
\sum_{(i,j)\in\{1,\dots,n\}\times\{1,\dots,d\}} (Y_{ij} - x_i^T w_j)^2 &= \sum_{l=1}^{k} \sum_{i\in\mathcal{C}_l} \sum_{j=1}^{d} (Y_{ij} - x_i^T w_j)^2 \\
&= \sum_{l=1}^{k} \sum_{i\in\mathcal{C}_l} \sum_{j=1}^{d} (Y_{ij} - e_l w_j)^2 \\
&= \sum_{l=1}^{k} \sum_{i\in\mathcal{C}_l} \sum_{j=1}^{d} (Y_{ij} - w_j^l)^2 \\
&= \sum_{l=1}^{k} \sum_{i\in\mathcal{C}_l} \| Y_i - w^l \|^2
\end{aligned}
$$

- to minimize over W: set w^l to be the mean of Y_i for $i \in \mathcal{C}_l$

$$w^l = \frac{1}{\mathcal{C}_l} \sum_{i\in\mathcal{C}_l} Y_i$$

Check AM reproduces k-means

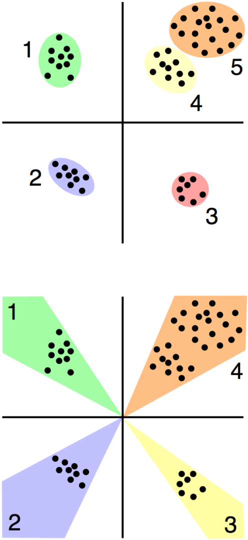
let w^l be lth row of W, $l = 1, \ldots, k$

$$
\begin{aligned}
\sum_{(i,j)\in\{1,\ldots,n\}\times\{1,\ldots,d\}} (Y_{ij} - x_i^T w_j)^2 &= \sum_{l=1}^{k} \sum_{i\in\mathcal{C}_l} \sum_{j=1}^{d} (Y_{ij} - x_i^T w_j)^2 \\
&= \sum_{l=1}^{k} \sum_{i\in\mathcal{C}_l} \sum_{j=1}^{d} (Y_{ij} - e_l w_j)^2 \\
&= \sum_{l=1}^{k} \sum_{i\in\mathcal{C}_l} \sum_{j=1}^{d} (Y_{ij} - w_j^l)^2 \\
&= \sum_{l=1}^{k} \sum_{i\in\mathcal{C}_l} \| Y_i - w^l \|^2
\end{aligned}
$$

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

$$x_i = e_l \quad \text{where} \quad l = \mathrm{argmin}\, l' \in 1, \ldots, k \| Y_i - w^l \|^2$$

What's a cluster?



Tropp 2004

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: $\ell(y, z) = |y - z|$
- $\ell(y, z) = \mathbf{huber}(y - z)$
- $\ldots$

Fitting GLRMs with alternating minimization

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

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

- 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

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

- **simple:** only requires ability to evaluate ∇L and $\mathbf{prox}_r$
- **stochastic variant:** use noisy estimate for g
- **time per iteration:** $O(\frac{(n+d+|\Omega|)k}{p})$ on p processors

Recap: GLRMs

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

many of these GLRMs have names:

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

Resources

- GLRMs
  `https://people.orie.cornell.edu/mru8/doc/udell16_glrm.pdf`
- fitting GLRMS
  `https://github.com/madeleineudell/LowRankModels.jl`