Generalized Low Rank Models

Madeleine Udell

Computational and Mathematical Engineering
Stanford University

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

March 18, 2015

Data table

age	gender	state	income	education	⋯
29	F	CT	$53,000	college	⋯
57	?	NY	$19,000	high school	⋯
?	M	CA	$102,000	masters	⋯
41	F	NV	$23,000	?	⋯
⋮	⋮	⋮	⋮	⋮	

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

Data table

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

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

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

Low rank model

given: $A \in \mathbf{R}^{m \times n}$, $k \ll m, n$
find: $X \in \mathbf{R}^{m \times k}$, $Y \in \mathbf{R}^{k \times n}$ for which

$$\begin{bmatrix} \\ X \\ \\ \end{bmatrix} \begin{bmatrix} \quad Y \quad \end{bmatrix} \approx \begin{bmatrix} \\ A \\ \\ \end{bmatrix}$$

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

$$\begin{bmatrix} \\ X \\ \\ \end{bmatrix} = \begin{bmatrix} —x_1— \\ \vdots \\ —x_m— \end{bmatrix} \qquad \begin{bmatrix} \quad Y \quad \end{bmatrix} = \begin{bmatrix} | & & | \\ y_1 & \cdots & y_n \\ | & & | \end{bmatrix}$$

interpretation:

- X and Y are (compressed) representation of A
- $x_i^T \in \mathbf{R}^k$ is a point associated with example i
- $y_j \in \mathbf{R}^k$ is a point associated with feature j
- inner product $x_i y_j$ approximates A_{ij}

Why use a low rank model?

- reduce storage; speed transmission
- understand (visualize, cluster)
- remove noise
- infer missing data
- simplify data processing

Outline

Principal components analysis

PCA:

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

with variables $X \in \mathbf{R}^{m \times k}$, $Y \in \mathbf{R}^{k \times n}$

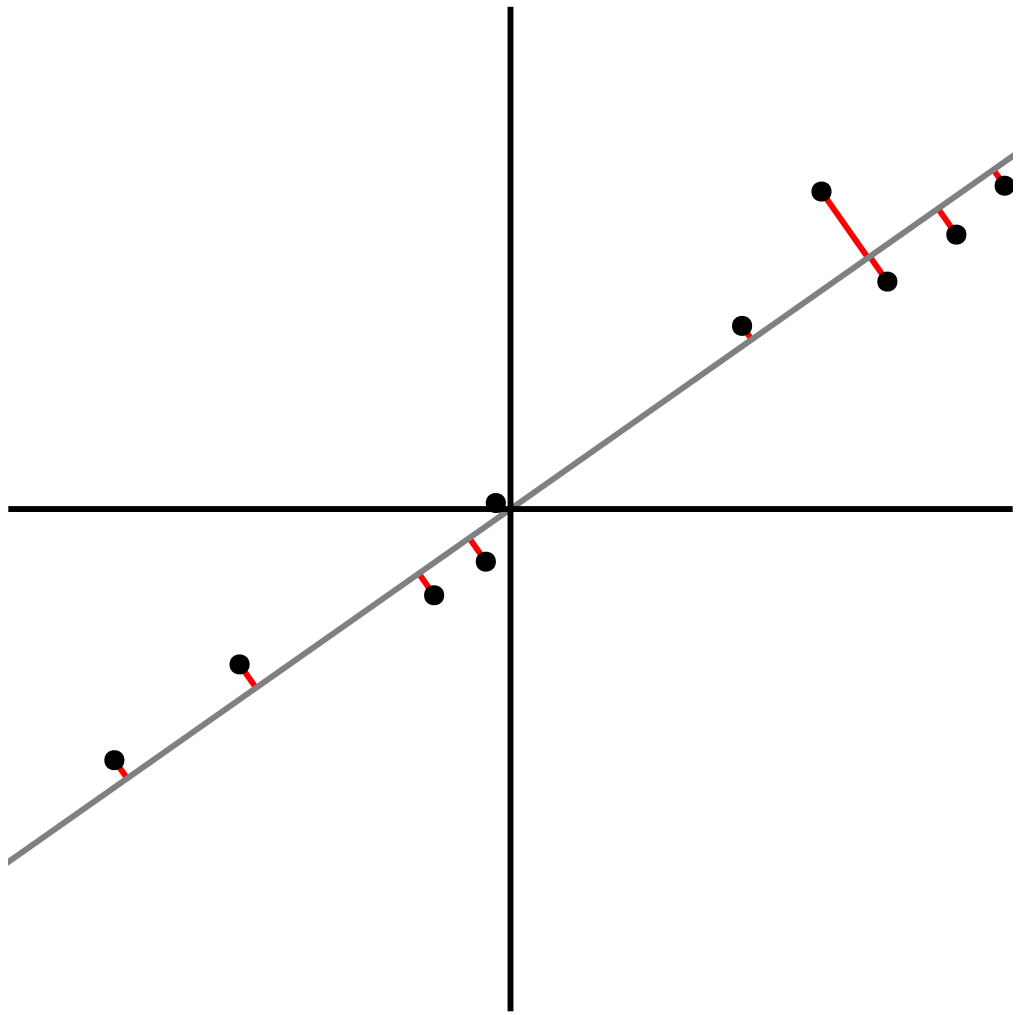
- old roots [Pearson 1901, Hotelling 1933]
- least squares low rank fitting
- (analytical) solution via SVD of $A = U \Sigma V^T$
- (numerical) solution via alternating minimization

PCA finds best covariates

PCA:

$$\text{minimize} \quad \|A - XY\|_F^2,$$

variables X and Y

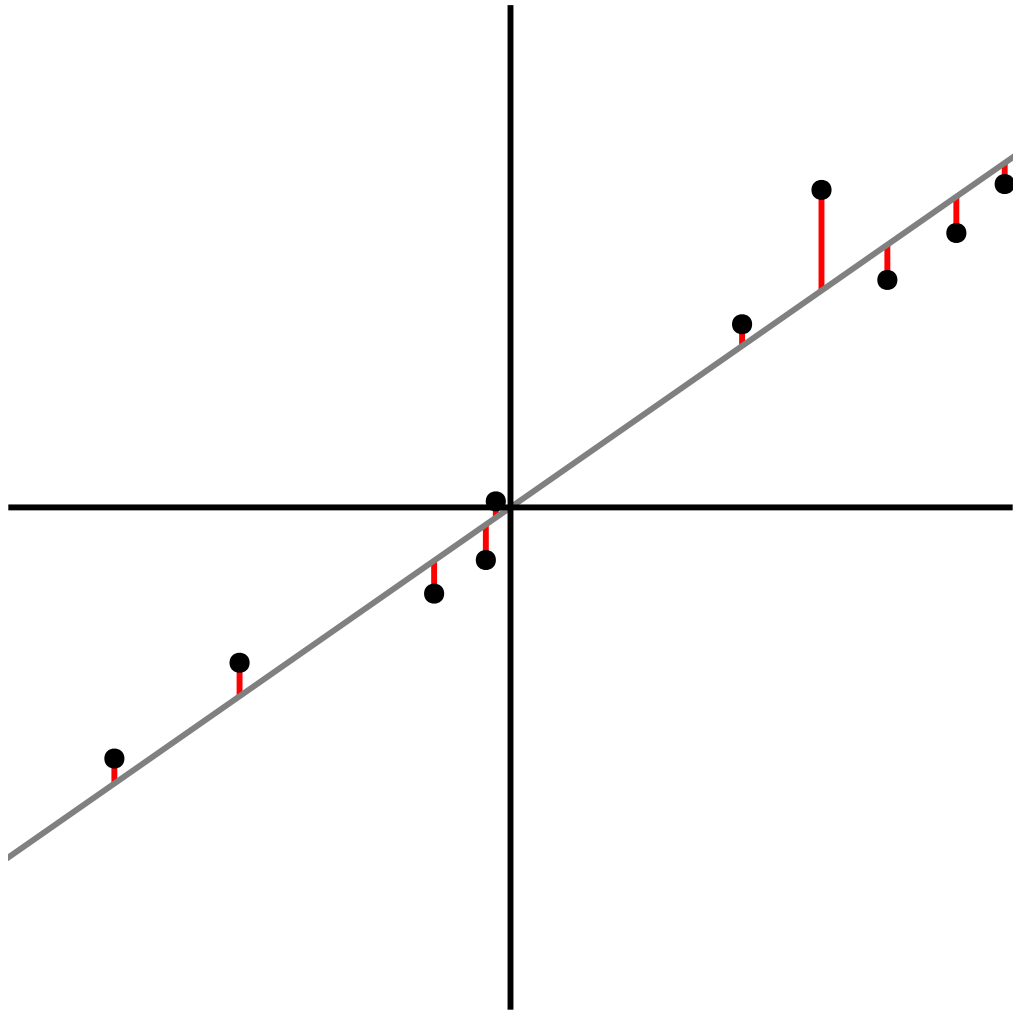


PCA finds best covariates

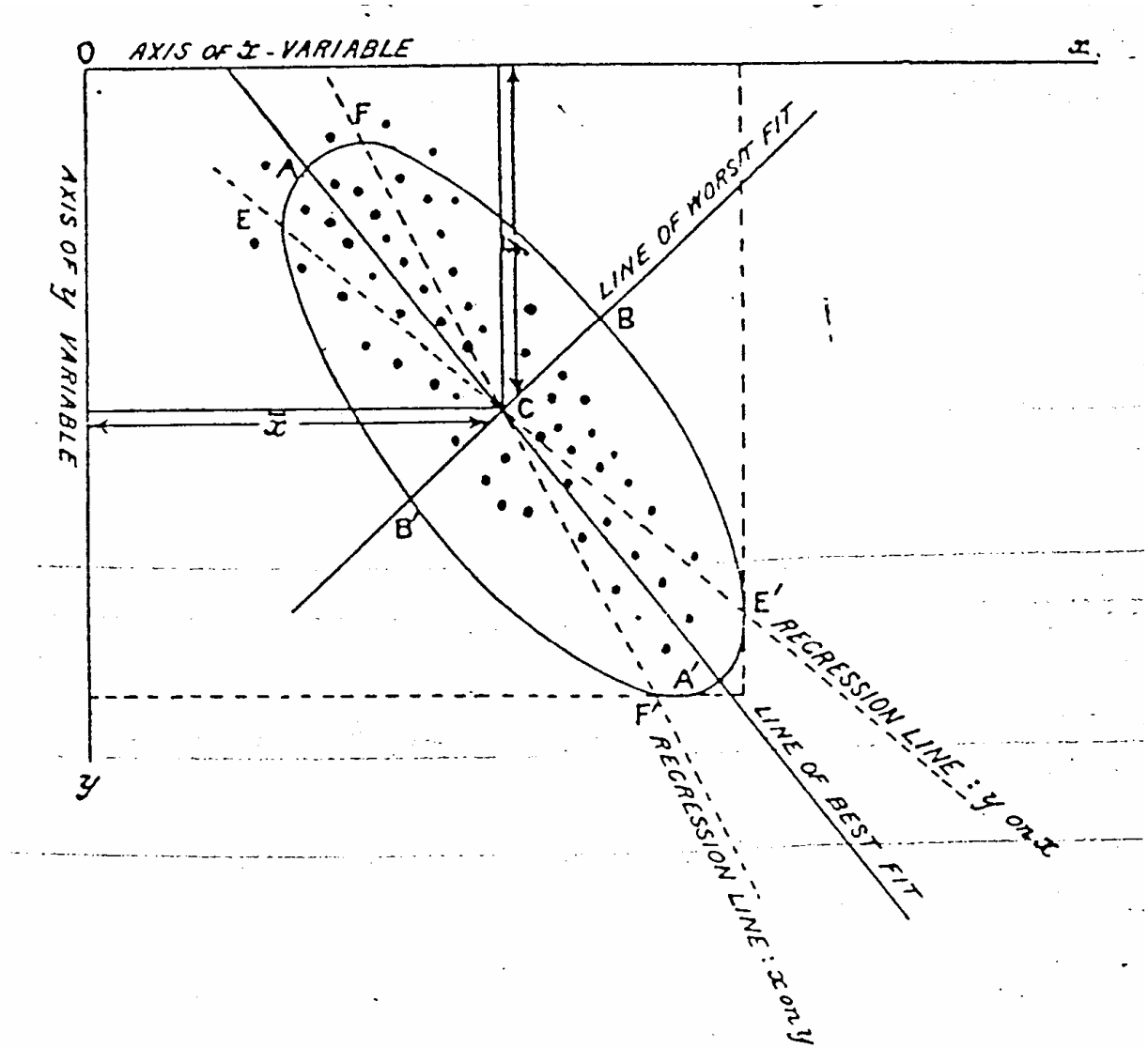
regression:

$$\text{minimize} \quad \|A - XY\|_F^2,$$

fix $X = A_{:,1:k}$ (first k columns of A), variable Y



On lines and planes of best fit



[Pearson 1901]

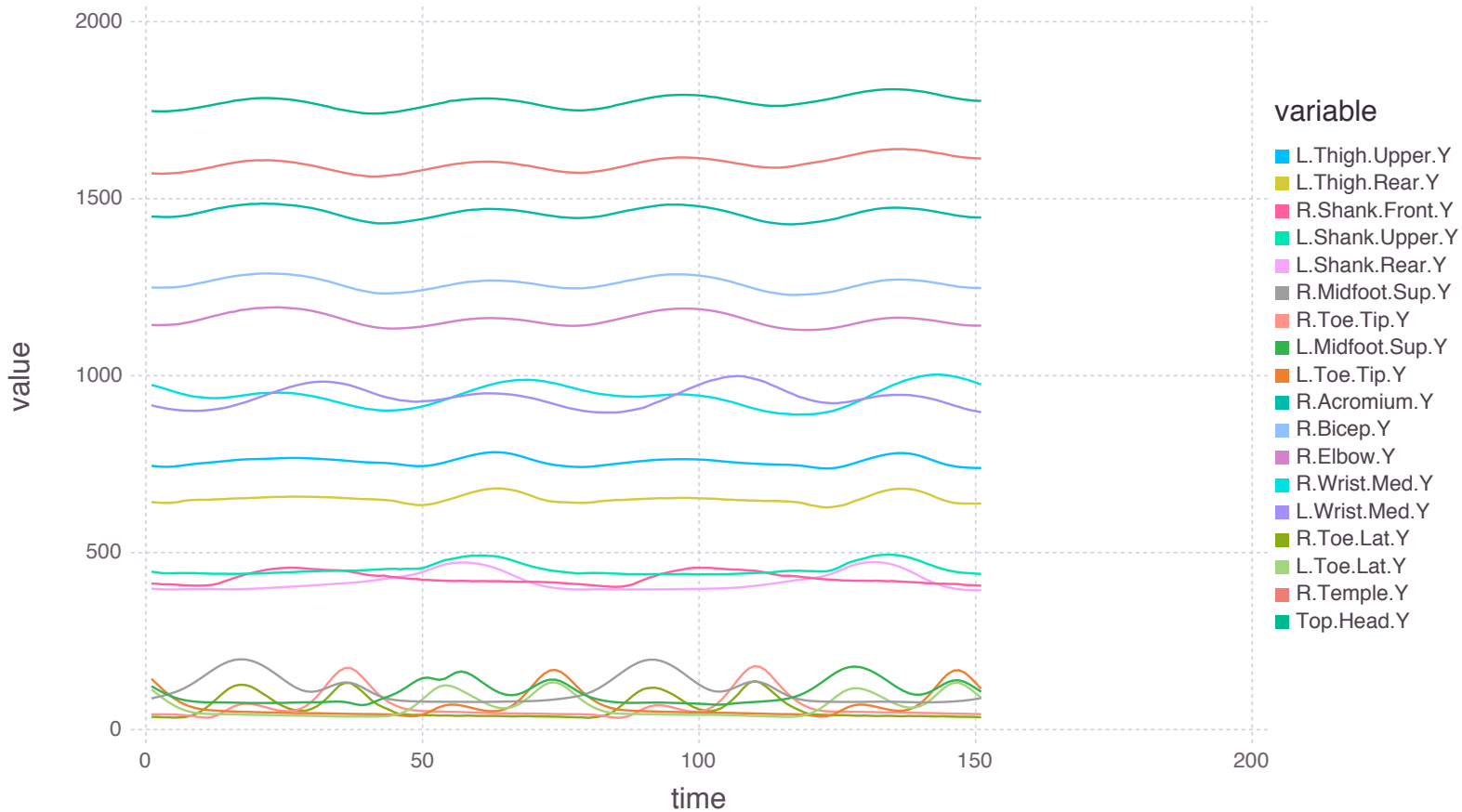
Low rank models for gait analysis

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

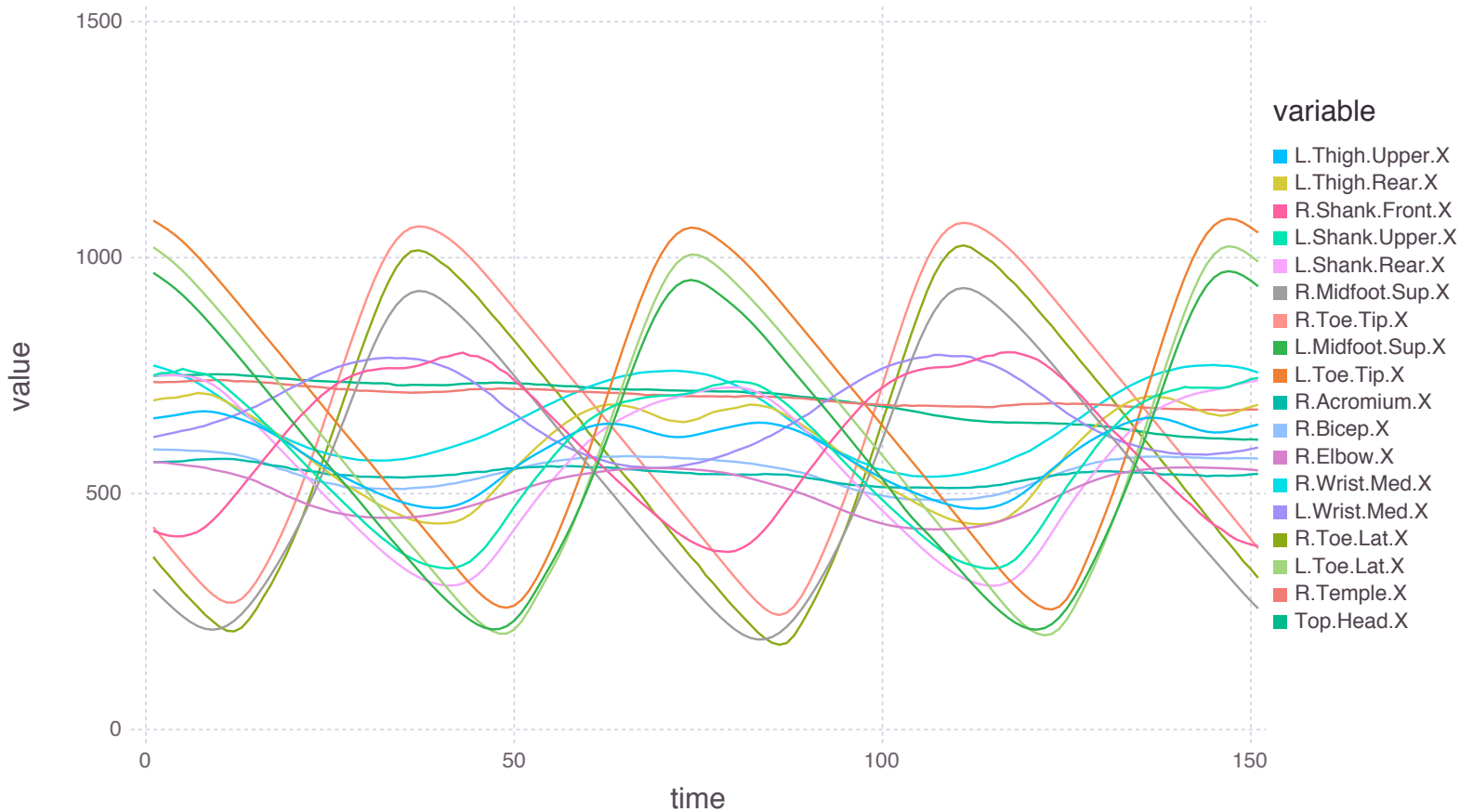
Interpreting principal components

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



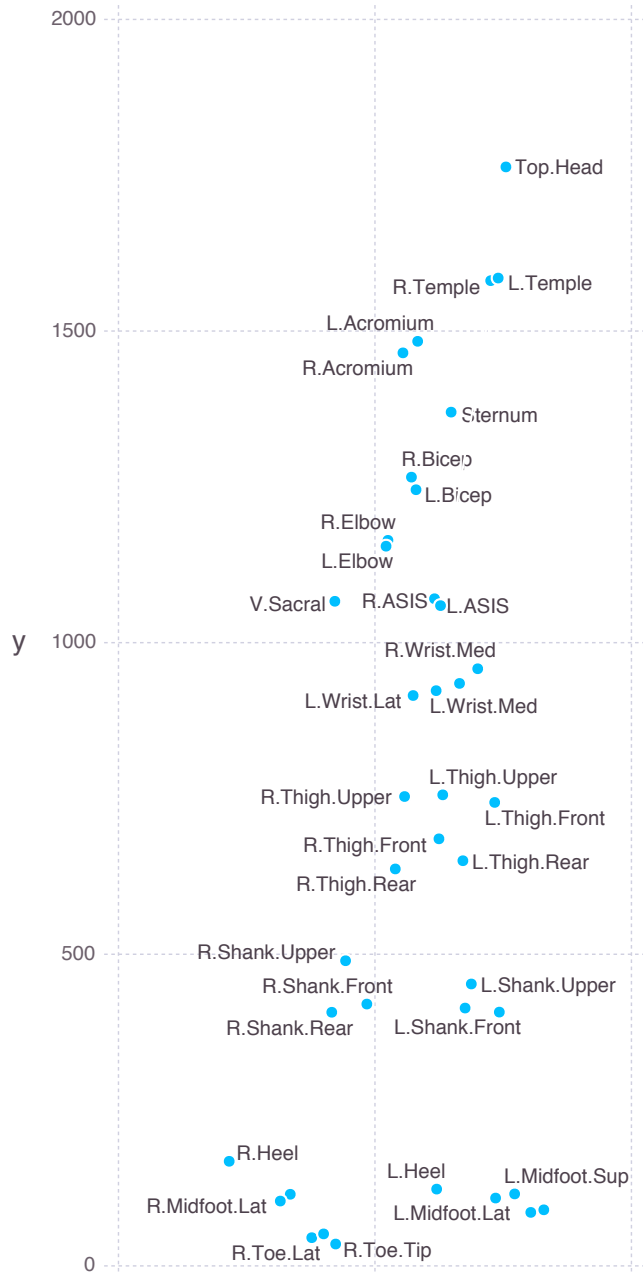
Interpreting principal components

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



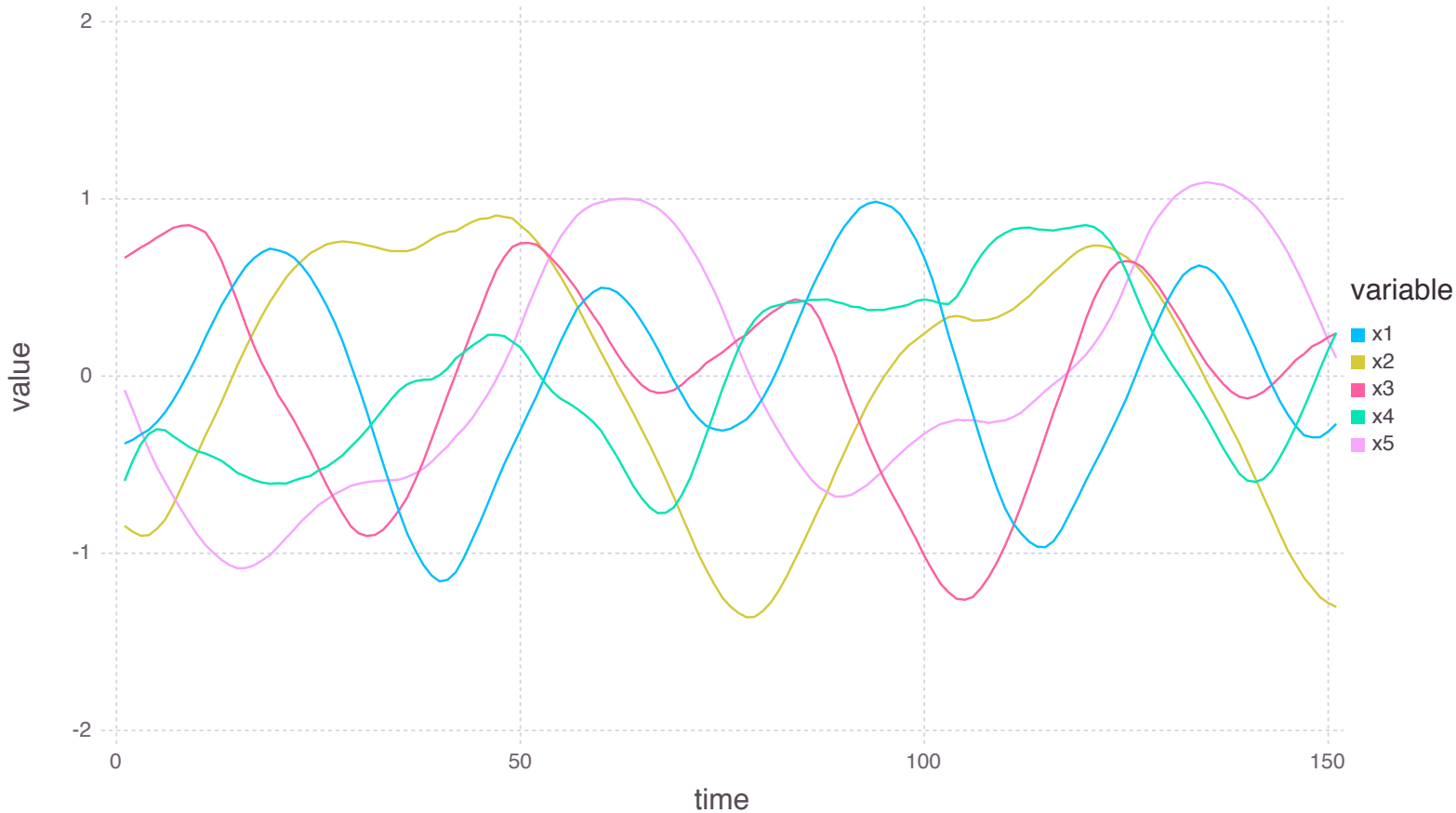
Interpreting principal components

row of Y
(archetypical example)
(principal stance)



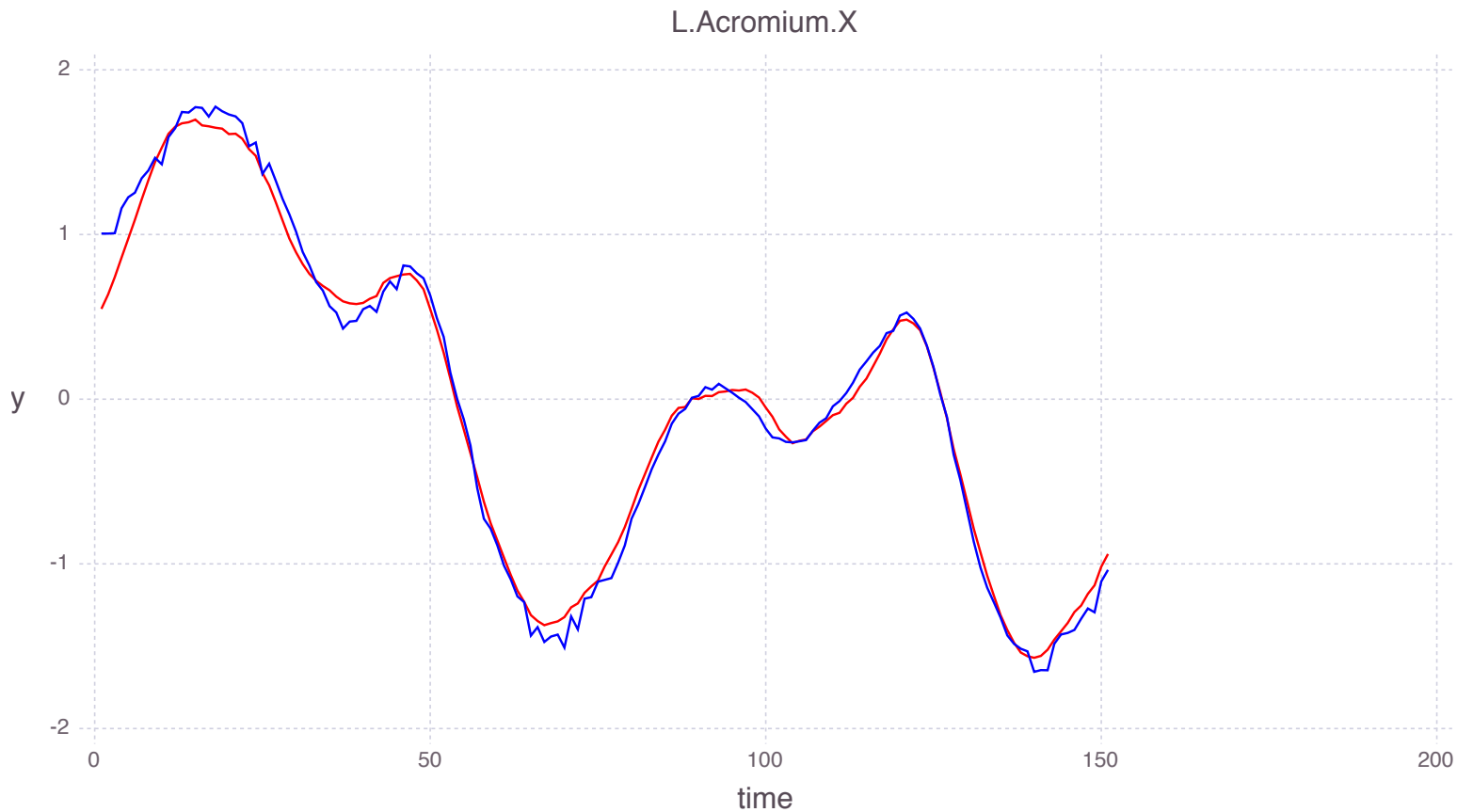
Interpreting principal components

columns of X (archetypical features) (principal timeseries)



Interpreting principal components

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



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

Outline

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 $r : \mathbf{R}^{1\times k} \to \mathbf{R}$, $\tilde{r} : \mathbf{R}^k \to \mathbf{R}$
- observe only $(i,j) \in \Omega$ (other entries are missing)

Note: can be NP-hard to optimize exactly...

Related work

- principal components analysis (PCA)
  [Pearson 1901, Hotelling 1933]
- exponential family PCA [Collins 2001]
- generalized^2 linear^2 models [Gordon 2002]
- convex relaxations of regularization [Srebro 2004]
- matrix factorization as clustering [Tropp 2004]
- matrix factorization models [Singh 2008]
- penalized matrix decomposition [Witten 2009]
- low rank approximation [Markovsky 2012]

Examples

variations on GLRMs recover many known models:

Model	$\mathbf{L_{ij}(u,a)}$	$\mathbf{r(x)}$	$\mathbf{\tilde{r}(y)}$	**reference**
PCA	$(u-a)^2$	0	0	[Pearson 1901]
NNMF	$(u-a)^2$	$I_+(x)$	$I_+(y)$	[Lee 1999]
sparse PCA	$(u-a)^2$	$\|x\|_1$	$\|y\|_1$	[D'Aspremont 2004]
sparse coding	$(u-a)^2$	$\|x\|_1$	$\|y\|_2^2$	[Olshausen 1997]
k-means	$(u-a)^2$	$I_1(x)$	0	[Tropp 2004]
matrix completion	$(u-a)^2$	$\|x\|_2^2$	$\|y\|_2^2$	[Keshavan 2010]
robust PCA	$\|u-a\|$	$\|x\|_2^2$	$\|y\|_2^2$	[Candes 2011]
logistic PCA	$\log(1+\exp(-au))$	$\|x\|_2^2$	$\|y\|_2^2$	[Collins 2001]
boolean PCA	$(1-au)_+$	$\|x\|_2^2$	$\|y\|_2^2$	[Srebro 2004]

Matrix completion

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

$$\text{minimize} \quad \sum_{(i,j)\in\Omega}(A_{ij} - x_i y_j)^2 + \gamma\|X\|_F^2 + \gamma\|Y\|_F^2$$

two regimes:

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

Theorem ([Keshavan 2010])

If A has rank $k' \leq k$ and $|\Omega| = O(nk' \log n)$ (and A is incoherent and Ω is chosen UAR), then matrix completion exactly recovers the matrix A with high probability.

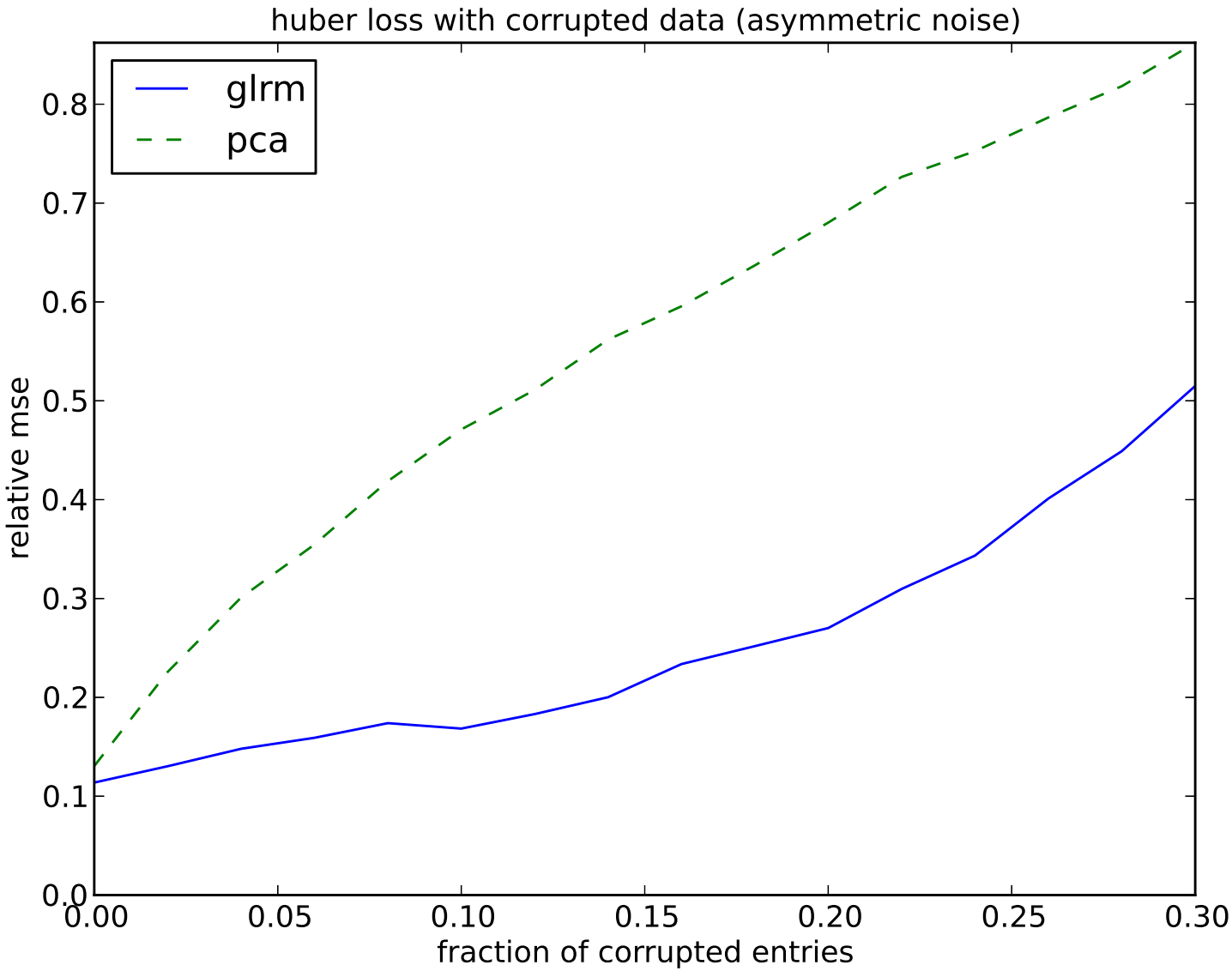
Huber PCA

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

where we define the Huber function

$$\textbf{huber}(z) = \begin{cases} \frac{1}{2}z^2 & |z| \le 1 \\ |z| - \frac{1}{2} & |z| > 1 \end{cases}$$

Huber PCA


huber loss with corrupted data (asymmetric noise)
glrm
pca
relative mse
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
fraction of corrupted entries

Huber PCA

Remark: Huber PCA is equivalent to Robust PCA

$$\begin{array}{ll} \text{minimize} & \sum_{(i,j)\in\Omega}(|S_{ij}| + \frac{1}{2}N_{ij}^2) + 2\|Z\|_* \\ \text{subject to} & Z + N + S = A \\ & \mathrm{Rank}(Z) \leq k, \end{array}$$

[Xu, Caramanis & Sanghavi, 2012]
since $\textbf{huber}(x) = \inf\{|s| + \frac{1}{2}n^2 : x = n + s\}$

Regularizers

Choose regularizer to impose structure on representation

- small
  - $r(x) = \|x\|_2^2$
- sparse
  - $r(x) = \|x\|_1$
  - $r(x) = I(\mathbf{card}(x) \leq p)$
- nonnegative
  - $r(x) = I_+(x)$
- clustered
  - $r(x) = I_1(x)$, $\tilde{r}(y) = 0$

Maximum likelihood low rank estimation

Choose loss function to maximize (log) likelihood of observations:

- gaussian noise: $L(u, a) = (u - a)^2$
- laplacian (heavy-tailed) noise: $L(u, a) = |u - a|$
- gaussian + laplacian noise: $L(u, a) = \textbf{huber}(u - a)$
- poisson (count) noise: $L(u, a) = \exp(u) - au + a \log a - a$
- bernoulli (coin toss) noise: $L(u, a) = \log(1 + \exp(-au))$

Abstract loss

define *abstract feature space* $\mathcal{F}_j$

e.g., $A_{ij} \in \mathcal{F}_j$ can be

- boolean
- ordinal
- categorical
- ranking

just need a loss function $L_j : \mathbf{R} \times \mathcal{F}_j \to \mathbf{R}$

Boolean losses

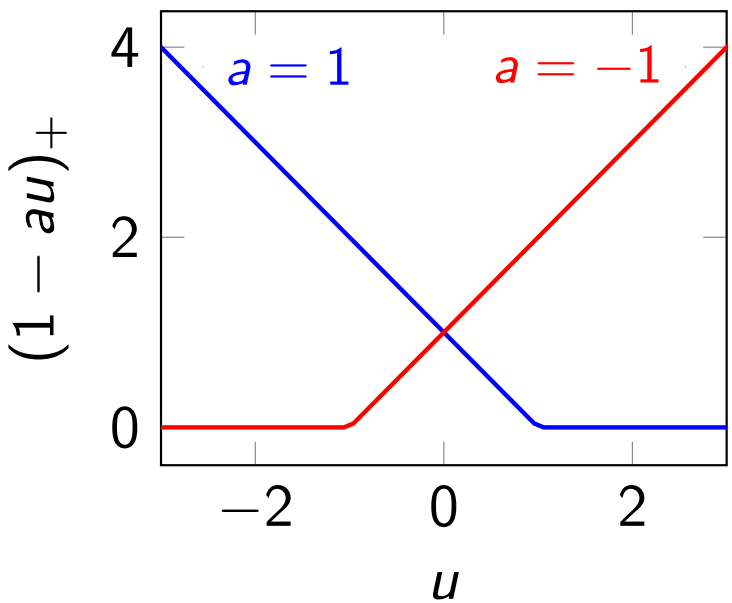
Boolean PCA: $\mathcal{F}_j = \{-1, 1\}$

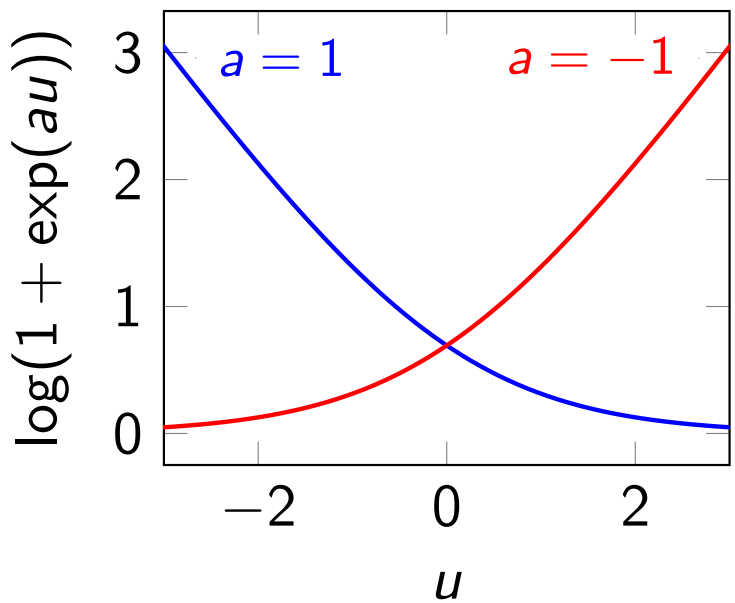
- hinge loss

$$L(u, a) = (1 - au)_+$$

- logistic loss

$$L(u, a) = \log(1 + \exp(-au))$$



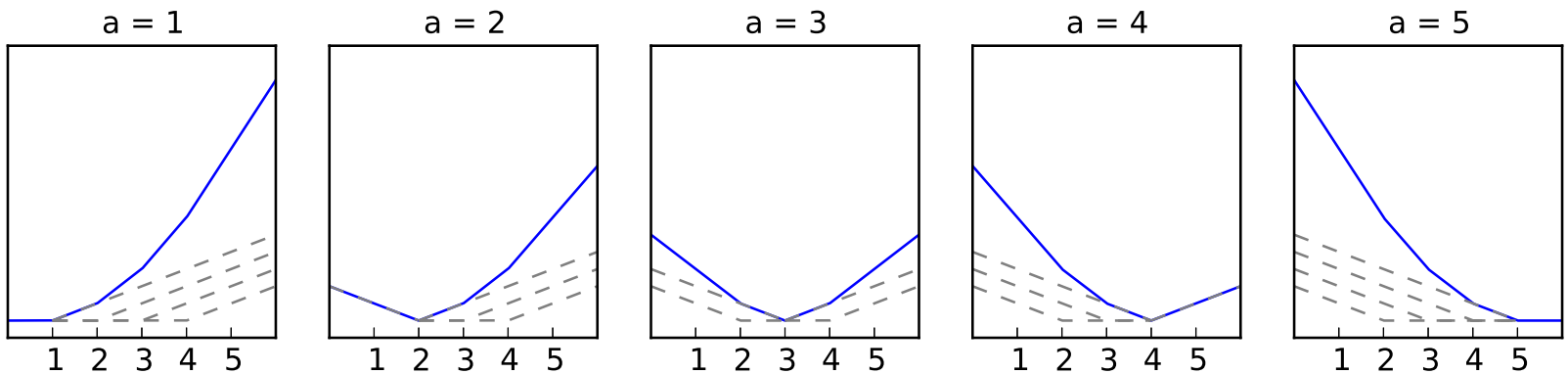


Ordinal loss

Ordinal PCA: $\mathcal{F}_j = \{1, \ldots, d\}$

- ordinal hinge loss

$$L(u, a) = \sum_{a'=1}^{a-1} (1 - u + a')_+ + \sum_{a'=a+1}^{d} (1 + u - a')_+$$



Multi-dimensional loss

- approximate using *vectors* $x_i\, Y_j \in \mathbf{R}^{1\times d_j}$ instead of numbers
- need $L_j : \mathbf{R}^{1\times d_j} \times \mathcal{F}_j \to \mathbf{R}$

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

- useful for approximating *categorical* variables
  - columns of Y_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]

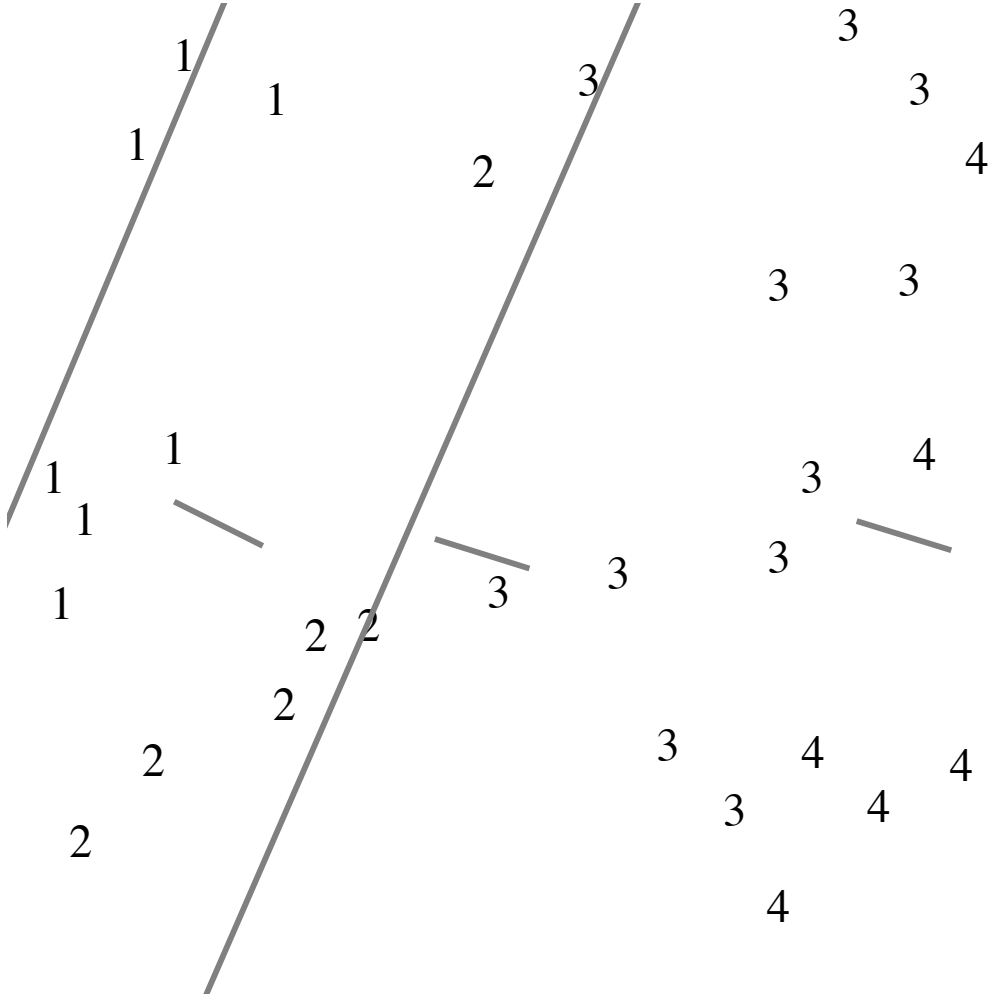
$$L(u, a) = (1 - u_a)_+ + \sum_{a' \neq a} (1 + u_{a'})_+$$

$$\begin{bmatrix} | & | & | & | & | \\ \text{CA} & \text{NV} & \cdots & \text{PA} & \text{NY} \\ | & | & | & | & | \end{bmatrix}$$

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

Multivariate ordinal loss

- automatically detect which labels are more similar
- fit positions of data (X) and separating hyperplanes (Y) simultaneously



Scaling losses

Analogue of standardization for GLRMs:

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

- μ_j generalizes column mean
- σ_j^2 generalizes column variance

To fit a standardized GLRM, solve

$$\text{minimize} \quad \textstyle\sum_{(i,j)\in\Omega} L_j(A_{ij}, x_i y_j + \mu_j)/\sigma_j^2 + \sum_{i=1}^m r_i(x_i) + \sum_{j=1}^n \tilde{r}_j(y_j)$$

Outline

Impute missing data

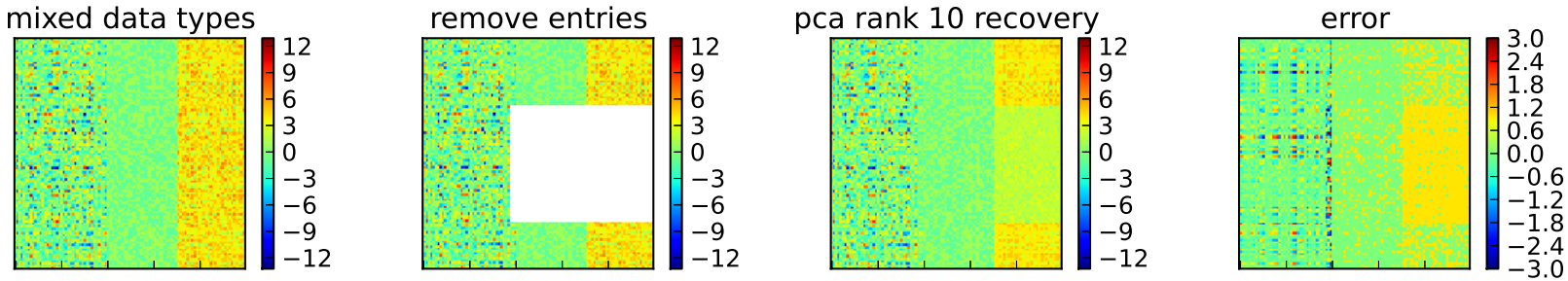
impute most likely true data $\hat{A}_{ij}$

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

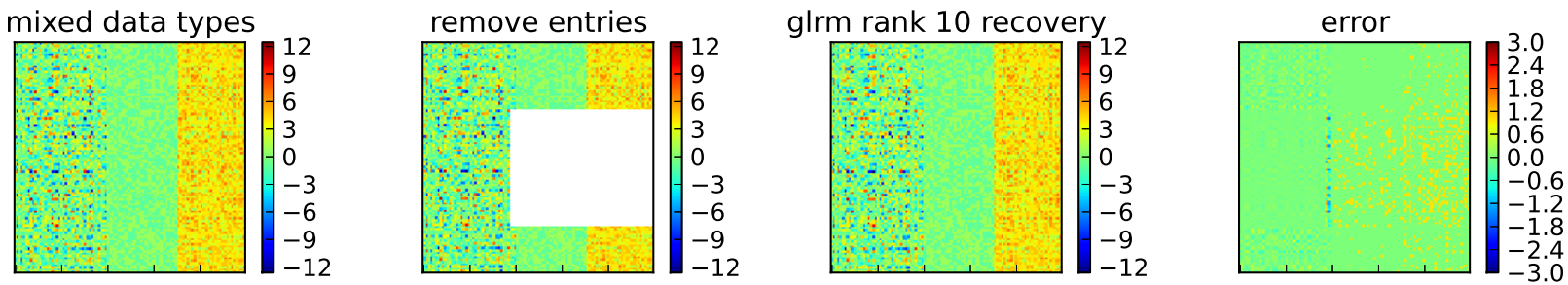
- implicit constraint: $\hat{A}_{ij} \in \mathcal{F}_j$
- when L_j is quadratic, ℓ_1, or Huber loss, then $\hat{A}_{ij} = x_i y_j$
- if $\mathcal{F} \neq \mathbf{R}$, $\operatorname{argmin}_a L_j(x_i y_j, a) \neq x_i y_j$
  - *e.g.*, for hinge loss $L(u, a) = (1 - ua)_+$, $\hat{A}_{ij} = \mathbf{sign}(x_i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

Impute heterogeneous data

PCA:



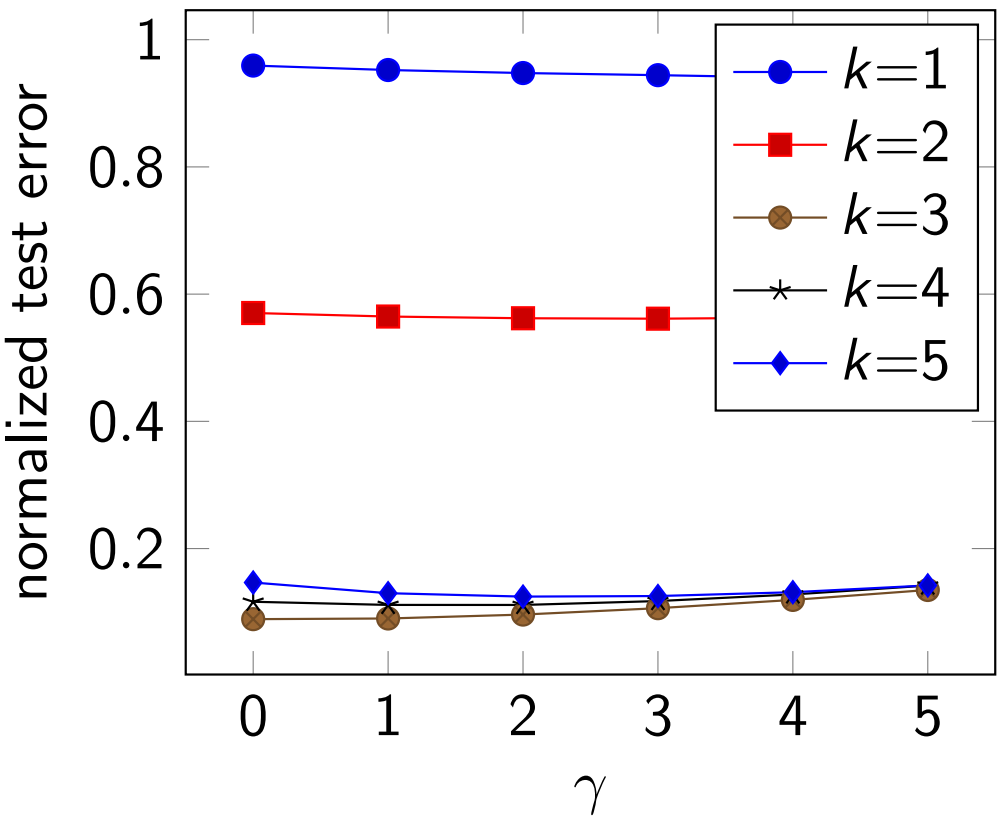
GLRM:



Validate model

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

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



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 Y 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

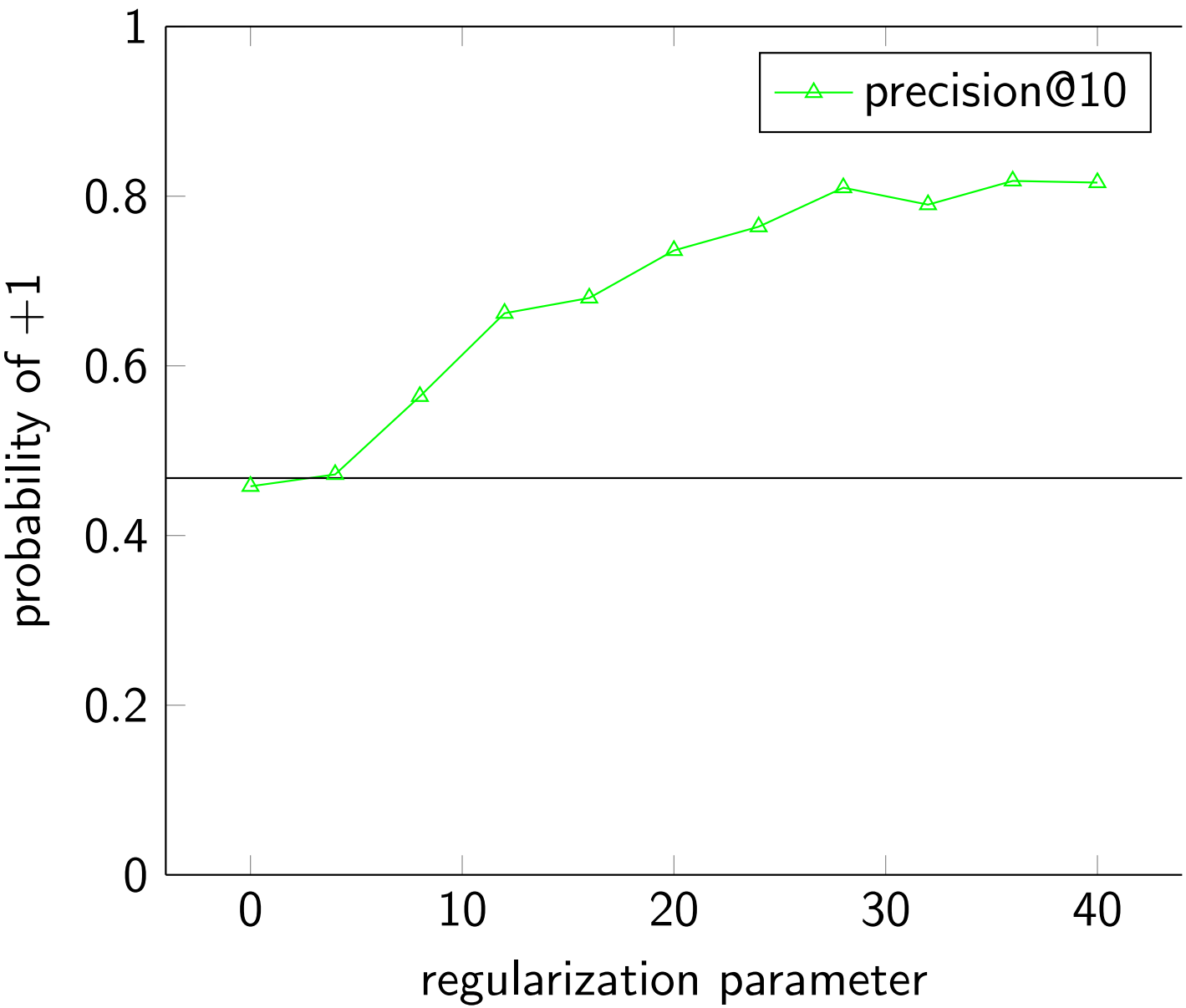
synthetic data:

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

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

can we predict 10 more +s?

Impute censored data



Correct biased sample

two types of people

- type A always fill out all questions
- type B leave question 3 blank half the time

question 1	question 2	question 3	question 4	$\cdots$
2.7	yes	4	yes	$\cdots$
2.7	yes	4	yes	$\cdots$
9.2	no	?	no	$\cdots$
2.7	yes	4	yes	$\cdots$
9.2	no	1	no	$\cdots$
9.2	no	?	no	$\cdots$
2.7	yes	4	yes	$\cdots$
9.2	no	1	no	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	

estimate population mean of question 3?

Correct biased sample

question 1	question 2	question 3	question 4	$\cdots$
2.7	yes	4	yes	$\cdots$
2.7	yes	4	yes	$\cdots$
9.2	no	?	no	$\cdots$
2.7	yes	4	yes	$\cdots$
9.2	no	1	no	$\cdots$
9.2	no	?	no	$\cdots$
2.7	yes	4	yes	$\cdots$
9.2	no	1	no	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	

estimate population mean of question 3:

- excluding missing entries: 3
- imputing missing entries: 2.5

American community survey

2013 ACS:

- 3M respondents, 87 economic/demographic survey questions
  - income
  - cost of utilities (water, gas, electric)
  - weeks worked per year
  - hours worked per week
  - home ownership
  - looking for work
  - use foodstamps
  - education level
  - state of residence
  - ...
- 1/3 of responses missing

Using a GLRM for exploratory data analysis

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

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

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

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

Fitting a GLRM to the ACS

- construct a rank 10 GLRM with loss functions respecting data types
  - huber for real values
  - hinge loss for booleans
  - ordinal hinge loss for ordinals
  - one-vs-all hinge loss for categoricals
- scale losses and regularizers by $1/\sigma_j^2$
- fit the GLRM

in 3 lines of code:

```
A = expand_categoricals(A, categoricals)
glrm, labels = GLRM(A, 10, scale = true)
X,Y = 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

Outline

Convergence theory for GLRMs

can we fit GLRMs?

- exactly, always: **no**
  - NP hard to solve weighted PCA [Gillis2011], k-means [Drineas2004], or NNMF [Vavasis2009]
- exactly, sometimes: **yes**
  - some GLRMs are equivalent to convex problems
- approximately (heuristically), always: **yes**
  - alternating minimization never increases the objective value

Fitting GLRMs with alternating minimization

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

repeat:

1. minimize objective over x_i (in parallel)
2. minimize objective over y_j (in parallel)

properties:

- subproblems easy to solve
- objective decreases at every step, so converges if losses and regularizers are bounded below
- (not guaranteed to find global solution, but) usually finds good model in practice
- naturally parallel, so scales to *huge* problems

Alternating updates

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

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

A simple, fast update rule

proximal gradient method: let

$$g = \sum_{j:(i,j)\in\Omega} \nabla L_j(x_i y_j, A_{ij}) y_j$$

and update

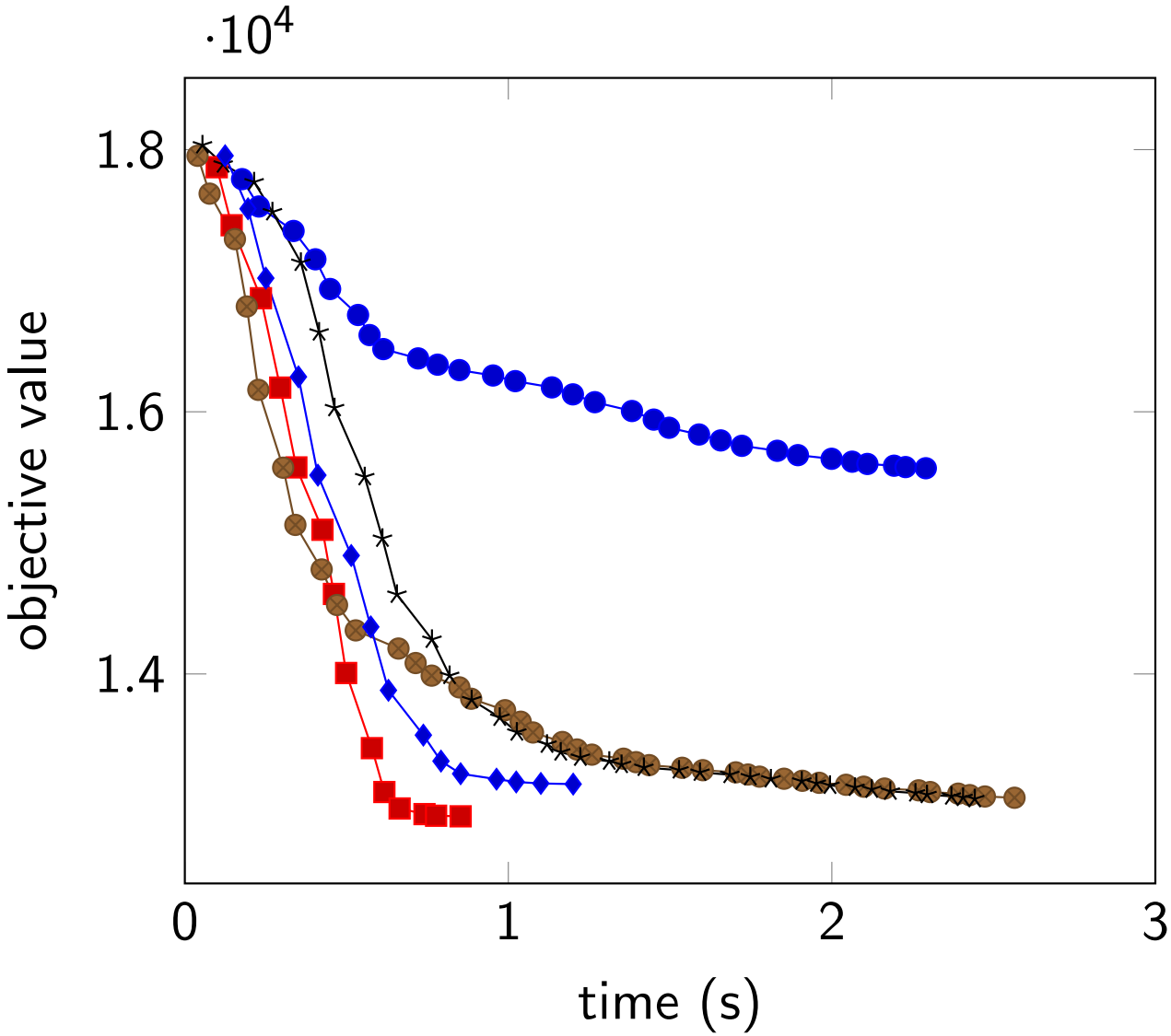
$$x_i^{t+1} = \mathbf{prox}_{\alpha_t r}(x_i^t - \alpha_t g)$$

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

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

Approximately, always

NNMF for $k = 2$: optimal value depends on initialization

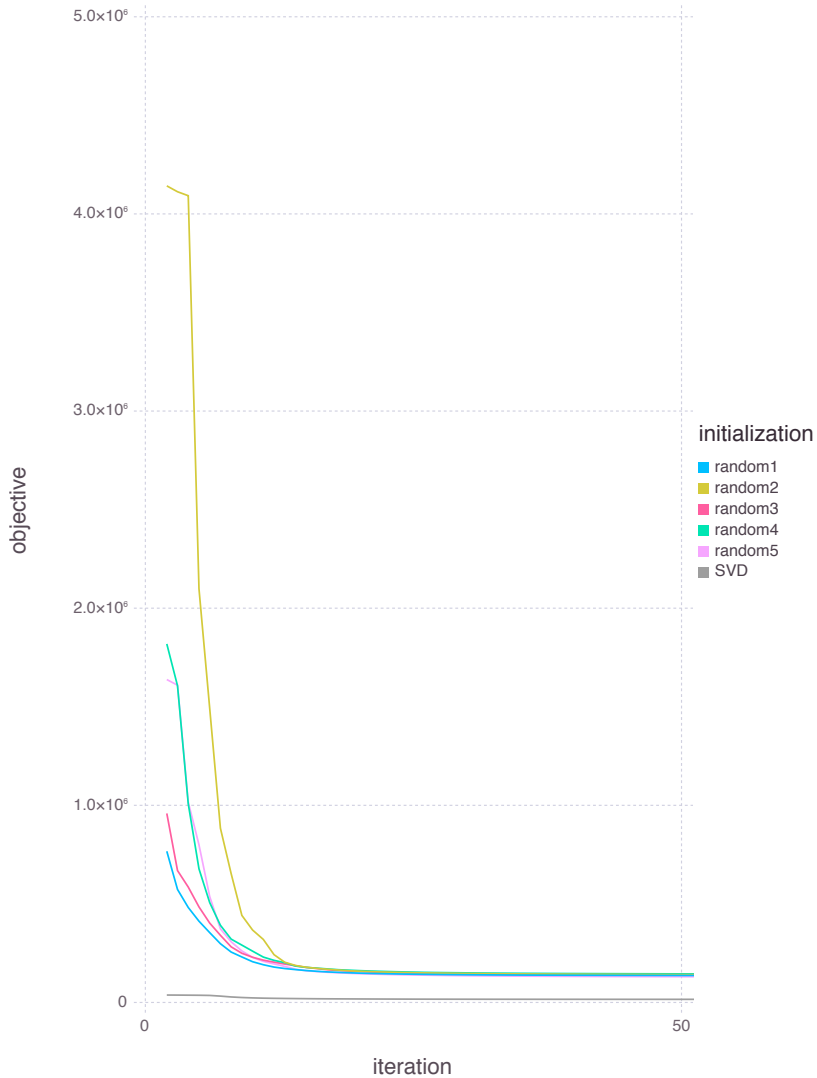


Initializing via SVD

- fit census data set
- random initialization

$$x_i \sim \mathcal{N}(0, I)$$
$$y_j \sim \mathcal{N}(0, I)$$

- SVD initialization
  - fill in missing entries in A to preserve column mean and variance
  - initialize XY with PCA



Exactly, sometimes

Theorem

(X, Y) *is a solution to*

$$\textit{minimize} \quad F(XY) + \tfrac{\gamma}{2}\|X\|_F^2 + \tfrac{\gamma}{2}\|Y\|_F^2 \qquad (\mathcal{F})$$

if and only if $Z = XY$ *is a solution to*

$$\begin{array}{ll} \textit{minimize} & F(Z) + \gamma\|Z\|_* \\ \textit{subject to} & Rank(Z) \leq k \end{array} \qquad (\mathcal{R})$$

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 $\mathcal{R}$

Proof of equivalence

suppose $Z = XY = U\Sigma V^T$

- $\mathcal{F} \leq \mathcal{R}$: if Z is feasible for $\mathcal{R}$, then

$$X = U\Sigma^{1/2}, \qquad Y = \Sigma^{1/2}V^T$$

  is feasible for $\mathcal{F}$, with the same objective value
- $\mathcal{R} \leq \mathcal{F}$: for any $XY = Z$,

$$\begin{aligned} \|Z\|_* &= \mathbf{tr}(\Sigma) \\ &= \mathbf{tr}(U^T XYV) \\ &\leq \|U^T X\|_F \|YV\|_F \\ &\leq \|X\|_F \|Y\|_F \\ &\leq \frac{1}{2}(||X||_F^2 + ||Y||_F^2) \end{aligned}$$

Convex equivalence

Theorem

For every $\gamma \geq \gamma^\star(k)$, every solution to

$$\begin{array}{ll} \textit{minimize} & L(Z) + \gamma\|Z\|_* \\ \textit{subject to} & Rank(Z) \leq k \end{array} \qquad (\mathcal{R})$$

(with variable $Z \in \mathbf{R}^{m\times n}$) is a solution to

$$\textit{minimize} \quad L(Z) + \gamma\|Z\|_*. \qquad (\mathcal{U})$$

proof: find $\gamma^\star(k)$ so large that there is a Z with rank $\leq k$ satisfying optimality conditions for $\mathcal{U}$

- if γ is sufficiently large (compared to k), rank constraint is *not binding*

Certify global optimality, sometimes

two ways to use convex equivalence:

- **convex:**
  1. solve the unconstrained SDP

$$\text{minimize} \quad F(Z) + \gamma \|Z\|_*$$

  2. see if the solution is low rank
- **nonconvex:**
  1. fit the GLRM with any method, producing (X, Y)
  2. check if $XY = U\Sigma V^T$ satisfies the optimality conditions for the (convex) unconstrained SDP

$$\|\partial F(XY) + \gamma UV^T\|_2 \leq 1$$

Implementations

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

example: (Julia) forms and fits a k-means model with $k = 5$

```
losses = quadratic()           # quadratic loss
rx = unitonesparse()           # x is 1-sparse unit vector
ry = zeroreg()                 # y is not regularized
glrm = GLRM(A,losses,rx,ry,k) # form GLRM
X,Y = fit!(glrm)               # fit GLRM
```

Model validation

```
for loss in [quadratic, l1, huber, hinge]
  for reg in [quadreg, onesparse, nonnegative]
    for k in 1:10
      for gamma in logspace(-2, 2, 5)
        glrm = GLRM(A, loss(), reg(), reg(), k)
        train_error, test_error, _ = cross_validate(glrm)
      end
    end
  end
end
```

Timing

fitting quadratically regularized PCA with $k = 10$

- **Spark:** Amazon EC2 cluster with instance types "c3.4xlarge" (16 CPU cores and 30 GB of RAM per machine)
- **Julia:** shared memory machine with 500 GB RAM total

Code (num cores)	Matrix size	# nonzeros	Time/iter (s)
Julia (1)	$10^3 \times 10^3$	10^6	1.5
Julia (30)	$10^4 \times 10^4$	10^8	29
Julia (30)	$10^4 \times 10^5$	10^9	370
Spark (160)	$10^6 \times 10^6$	10^6	9
Spark (160)	$10^6 \times 10^6$	10^9	13
Spark (160)	$10^7 \times 10^7$	10^9	294

Contributions

- a more general framework
  - losses for abstract data types
  - automatic scaling for heterogeneous losses
- a more general algorithm
  - parallel algorithms for (heuristically) fitting *any* GLRM
  - software package(s) implementing framework
  - heuristic initialization rules
- new analytic tools
  - model validation
  - certificates of optimality (sometimes)

Conclusion

generalized low rank models

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

paper

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

code

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