FACULTY OF
COMPUTER SCIENCE
IAȘI

Exploiting a new probabilistic model: S2FA

Simple-Supervised Factor Analysis

Student: Sebastian-Adrian Ciobanu

July, 2019

Scientific Coordinator: Associate Prof. Liviu Ciortuz

Outline

- Introduction
  - Motivation
  - Factor Analysis (FA)
  - Previous work
- The algorithms
  - S2UncFA, ...
  - S3UncFA, ...
  - MS3UncFA, ...
- The R package: s2fa
- Some Experiments
- Conclusion and future work

Introduction

Motivation

GJB

Gaussian Joint Bayes

EM/GMM

Expectation Maximization/Gaussian Mixture Model

$$w_{ij} \stackrel{\text{not.}}{=} p(z_i = j | x_i; \pi', \mu', \Sigma') = \frac{p(x_i | z_i = j; \mu', \Sigma') p(z_i = j; \pi')}{\sum_{l=1}^{K} p(x_i | z_i = l; \mu', \Sigma') p(z_i = l; \pi')}$$

GJB:

$$\pi_j = \frac{1}{n} \sum_{i=1}^{n} 1_{\{z_i = j\}}$$

$$\mu_j = \frac{\sum_{i=1}^{n} 1_{\{z_i=j\}} x_i}{\sum_{i=1}^{n} 1_{\{z_i=j\}}}$$

$$\Sigma_j = \frac{\sum_{i=1}^{n} 1_{\{z_i=j\}} (x_i - \mu_j)(x_i - \mu_j)^\top}{\sum_{i=1}^{n} 1_{\{z_i=j\}}}$$

EM/GMM:

$$\pi_j = \frac{1}{n} \sum_{i=1}^{n} w_{ij}$$

$$\mu_j = \frac{\sum_{i=1}^{n} w_{ij} x_i}{\sum_{i=1}^{n} w_{ij}}$$

$$\Sigma_j = \frac{\sum_{i=1}^{n} w_{ij} (x_i - \mu_j)(x_i - \mu_j)^\top}{\sum_{i=1}^{n} w_{ij}}$$

Factor Analysis

$$z \sim \mathcal{N}(\underline{0, I}),\ z \in \mathbb{R}^{d\times 1}$$

$$x|z \sim \mathcal{N}(\mu + \Lambda z, \Psi),\ x \in \mathbb{R}^{D\times 1}, \mu \in \mathbb{R}^{D\times 1}, \Lambda \in \mathbb{R}^{D\times d}, \Psi \in \mathbb{R}^{D\times D}$$

diagonal

More general:

$$z \sim \mathcal{N}(\underline{\mu_z, \Sigma_z}),\ z \in \mathbb{R}^{d\times 1},\ \mu_z \in \mathbb{R}^{d\times 1},\ \Sigma_z \in \mathbb{R}^{d\times d}$$

$$x|z \sim \mathcal{N}(\mu + \Lambda z, \Psi),\ x \in \mathbb{R}^{D\times 1}, \mu \in \mathbb{R}^{D\times 1}, \Lambda \in \mathbb{R}^{D\times d},\ \Psi \in \mathbb{R}^{D\times D}$$

diagonal

$$\begin{bmatrix} x \\ z \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu + \Lambda\mu_z \\ \mu_z \end{bmatrix}, \begin{bmatrix} \Lambda\Sigma_z\Lambda^\top + \Psi & \Lambda\Sigma_z \\ (\Lambda\Sigma_z)^\top & \Sigma_z \end{bmatrix}\right)$$

$$x \sim \mathcal{N}(\mu + \Lambda\mu_z, \Lambda\Sigma_z\Lambda^\top + \Psi)$$

$$z|x \sim \mathcal{N}(\mu_z + \Sigma_z\Lambda^\top(\Lambda\Sigma_z\Lambda^\top + \Psi)^{-1}(x - \mu - \Lambda\mu_z), \Sigma_z - \Sigma_z\Lambda^\top(\Lambda\Sigma_z\Lambda^\top + \Psi)^{-1}\Lambda\Sigma_z^\top)$$

Previous work

- Supervised PCA/Latent factor regression
- PCA
  - Probabilistic PCA: $\Psi = \eta^2 I, \, \eta \in \mathbb{R}, \eta > 0$
    - Gaussian Process Latent Variable Model (GPLVM)
      - Supervised GPLVM

The algorithms

	UncFA	FA	PPCA	Train Type
S2	S2UncFA	S2FA	S2PPCA	Closed-form
S3	S3UncFA	S3FA	S3PPCA	EM
MS3	MS3UncFA	MS3FA	MS3PPCA	EM/Closed-form

...UncFA: Ψ not constrained

...FA: Ψ diagonal

...PPCA: Ψ scalar

All the algorithms are derived starting from the log-likelihood of (observed) data.

S2 series: Simple-Supervised

Algorithm S2UncFA

function TRAIN$(\{(x^{(i)}, z^{(i)}) | i \in \{1, \dots, n\}\})$

$\hat{\mu}_z = \frac{\sum_{i=1}^{n} z^{(i)}}{n}$

$\hat{\Sigma}_z = \frac{\sum_{i=1}^{n} (z^{(i)} - \hat{\mu}_z)(z^{(i)} - \hat{\mu}_z)^\top}{n}$

$\hat{\Lambda} = \left(n\bar{x}\bar{z}^\top - \sum_{i=1}^{n} x^{(i)} {z^{(i)}}^\top \right) \left(n\bar{z}\bar{z}^\top - \sum_{i=1}^{n} z^{(i)} {z^{(i)}}^\top \right)^{-1}$

$\hat{\mu} = \bar{x} - \hat{\Lambda}\bar{z}$

$\hat{\Psi} = \frac{\sum_{i=1}^{n} (x^{(i)} - \mu - \Lambda z^{(i)})(x^{(i)} - \mu - \Lambda z^{(i)})^\top}{n}$

return $(\hat{\mu}_z, \hat{\Sigma}_z, \hat{\Lambda}, \hat{\mu}, \hat{\Psi})$

function TEST$(x^*, (\hat{\mu}_z, \hat{\Sigma}_z, \hat{\Lambda}, \hat{\mu}, \hat{\Psi}))$

value $= \hat{\mu}_z + \hat{\Sigma}_z \hat{\Lambda}^\top (\hat{\Lambda} \hat{\Sigma}_z \hat{\Lambda}^\top + \hat{\Psi})^{-1} (x^* - \hat{\mu} - \hat{\Lambda}\hat{\mu}_z)$

covarianceMatrix $= \hat{\Sigma}_z - \hat{\Sigma}_z \hat{\Lambda}^\top (\hat{\Lambda} \hat{\Sigma}_z \hat{\Lambda}^\top + \hat{\Psi})^{-1} \hat{\Lambda} \hat{\Sigma}_z^\top$

return (value, covarianceMatrix)

$$\boxed{X_m Z_m^\top}$$

$$X_m = \begin{bmatrix} x^{(1)} & \dots & x^{(n)} \end{bmatrix}$$

$$Z_m = \begin{bmatrix} z^{(1)} & \dots & z^{(n)} \end{bmatrix}$$

Weak equivalence with Linear Regression

- S2UncFA
- S2FA
- S2PPCA

$x \in \mathbb{R}^{1\times 1}$:

$S2UncFA$:

$$z \sim \mathcal{N}(\mu_z, \Sigma_z)$$

$$x|z \sim \mathcal{N}(a^\top z + b, \sigma^2)$$

versus

LR:

$$y|x \sim \mathcal{N}(a^\top x + b, \sigma^2)$$

- The same applies to **multi-output** regression

Strong equivalence with Linear Regression

- S2UncFA

…

For a new instance x^*, we predict:

$$\hat{\mu}_z + \hat{\Sigma}_z \hat{\Lambda}^\top (\hat{\Lambda}\hat{\Sigma}_z\hat{\Lambda}^\top + \hat{\Psi})^{-1}(x^* - \hat{\mu} - \hat{\Lambda}\hat{\mu}_z)$$

…

- ~~S2FA~~
- ~~S2PPCA~~

S3 series: Simple-Semi-Supervised

Algorithm 5 S3UncFA

function $\text{TRAIN}(\{(x^{(1)}, z^{(1)}), \ldots, (x^{(a)}, z^{(a)}), x^{(a+1)}, \ldots, x^{(n)}\}$,nMaxIterations,eps)

$\theta^{(0)} = \text{initializeParameters}(\{(x^{(1)}, z^{(1)}), \ldots, (x^{(a)}, z^{(a)}), x^{(a+1)}, \ldots, x^{(n)}\})$

$l^{(0)}_{\text{RV\_Do}} = l_{\text{RV\_Do}}(\theta^{(0)})$ according to 2.17

for t = 0:nMaxIterations **do**

E step: Compute $E[Z^{(i)}], E[Z^{(i)}Z^{(i)\top}], i \in \{a+1, \ldots, n\}$ according to 2.18 and 2.19

M Step: Compute $\theta^{(t+1)} = (\mu_z^{(t+1)}, \Sigma_z^{(t+1)}, \mu^{(t+1)}, \Lambda^{(t+1)}, \Psi^{(t+1)})$ according to 2.12, 2.13, 2.14, 2.15, 2.16

$l^{(t+1)}_{\text{RV\_Do}} = l_{\text{RV\_Do}}(\theta^{(t+1)})$ according to 2.17

if $\frac{\|\theta^{(t)} - \theta^{(t+1)}\|_2^2}{\|\theta^{(t)}\|_2^2} \leq$eps **or** $\frac{|l^{(t)}_{\text{RV_Do}} - l^{(t+1)}_{\text{RV_Do}}|}{|l^{(t)}_{\text{RV_Do}}|} \leq$eps **then**

break

return $\theta^{(t)}$

function $\text{TEST}(x^*, (\hat{\mu}_z, \hat{\Sigma}_z, \hat{\Lambda}, \hat{\mu}, \hat{\Psi}))$ ▷ The same as in *S2UncFA*

value $= \hat{\mu}_z + \hat{\Sigma}_z\hat{\Lambda}^\top(\hat{\Lambda}\hat{\Sigma}_z\hat{\Lambda}^\top + \hat{\Psi})^{-1}(x^* - \hat{\mu} - \hat{\Lambda}\hat{\mu}_z)$

covarianceMatrix $= \hat{\Sigma}_z - \hat{\Sigma}_z\hat{\Lambda}^\top(\hat{\Lambda}\hat{\Sigma}_z\hat{\Lambda}^\top + \hat{\Psi})^{-1}\hat{\Lambda}\hat{\Sigma}_z^\top$

return (value, covarianceMatrix)

MS3 series: Missing Simple-Semi-Supervised

- …

$$E[X^{(i)}],\ E[Z^{(i)}],\ E[X^{(i)}X^{(i)\top}],\ E[Z^{(i)}Z^{(i)\top}],\ E[X^{(i)}Z^{(i)\top}]$$

- …

The R package: s2fa

- faInit
- faFit
- faPredict
- faPlot

- s2faFit
- s2faPredict
- s2faPlot

- s3faInit
- s3faFit
- s3Predict
- s3faPlot

- mS3faInit
- mS3faFit
- mS3faPredict
- mS3faImpute

Some experiments

Data: House data; Source: 'Long-Kogen Realty, Chicago, USA'; 13 input features, 1 output feature (price); 26 instances

Plots of the learnt hyperplanes

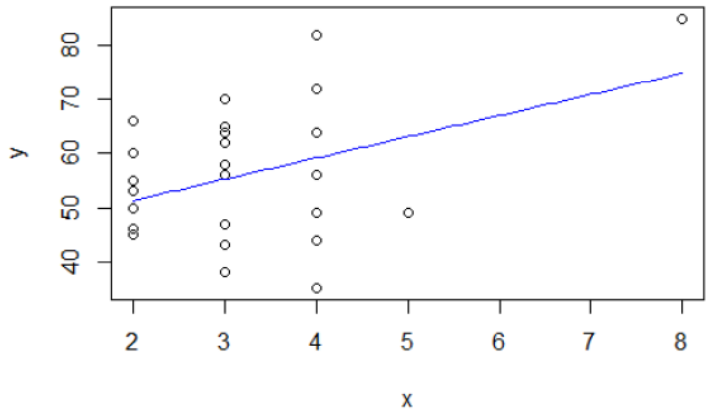


Figure 5.1: Hyperplanes. FA, PPCA - 1D: the second column - input; dimension of latent space = 1

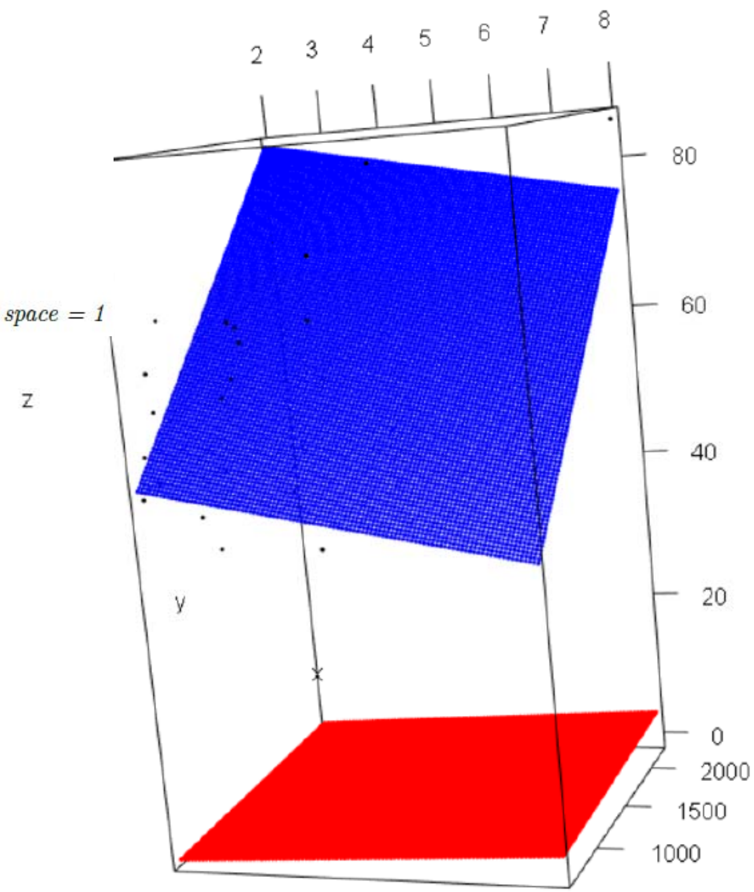


Figure 5.2: Hyperplanes. FA - 2D: the second and third columns - input; dimension of latent space = 1

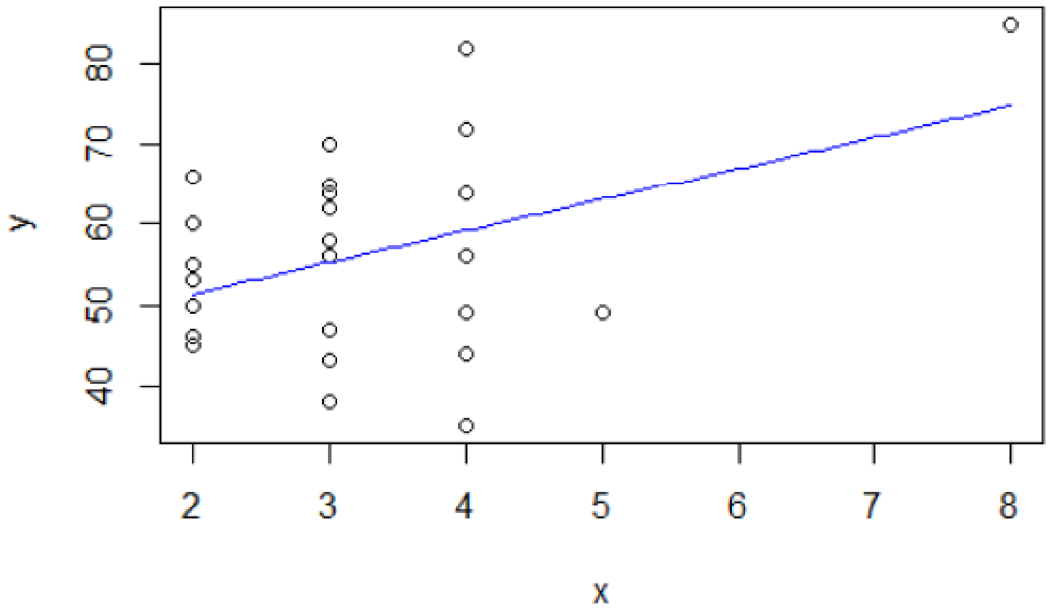
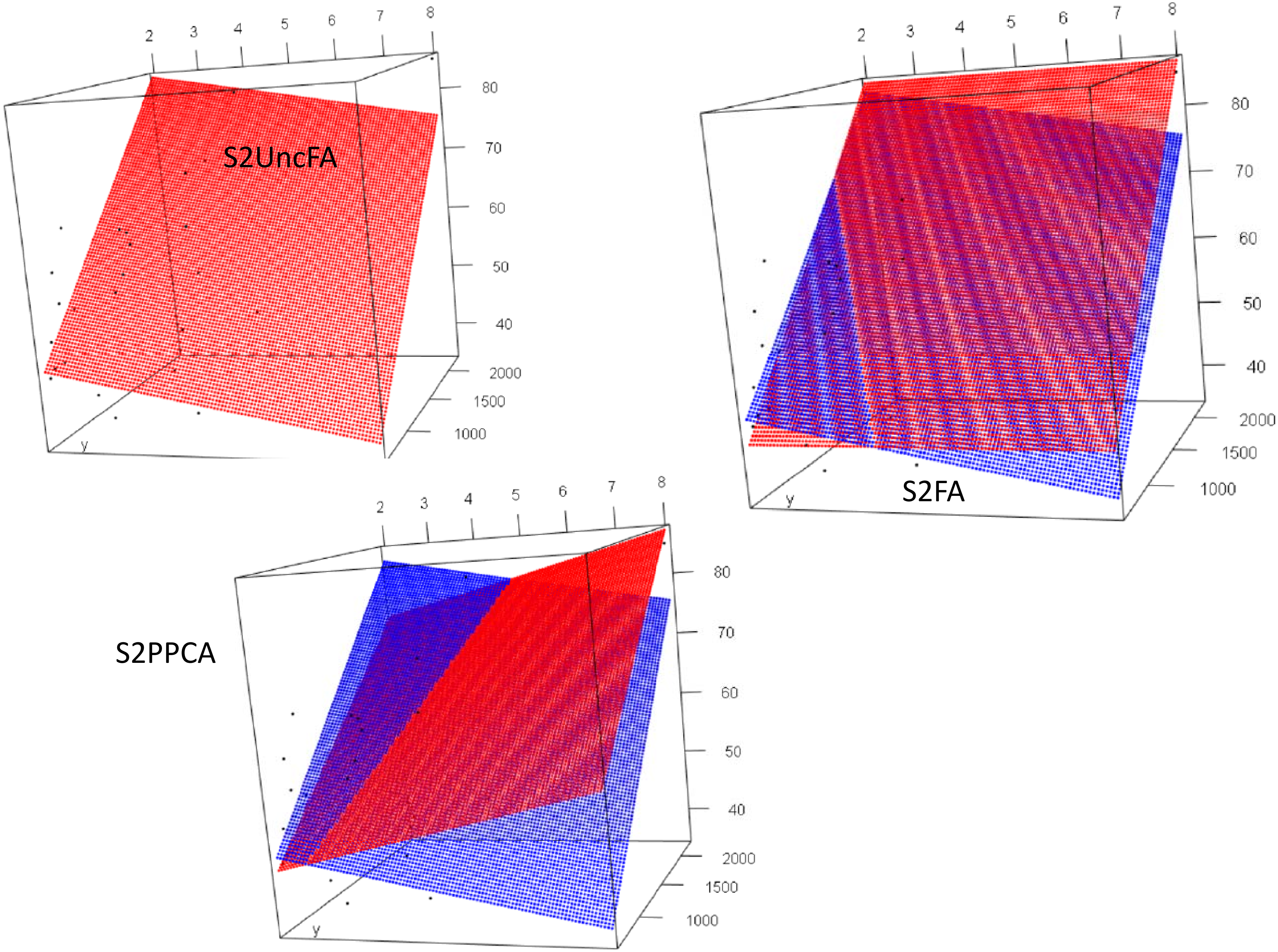


Figure 5.4:
Hyperplanes. S2UncFA, S2FA, S2PPCA - 1D: the second column - input; the first column - output


S2UncFA
S2FA
S2PPCA
2
3
4
5
6
7
8
80
70
60
50
40
2000
1500
1000
y

Data: Synthetic dataset

Synthetic dataset

	Real	UncFA (No sense)	S2UncFA	S3UncFA	MS3UncFA
mu_z_t	0 0	-	-0.01698442 0.09240896	-0.01112323 0.0876206	-0.01755008 0.09365257
sigma_z_t	1 0 0 1	-	0.94201830 0.09317562 0.09317562 0.88439545	0.94707575 0.09747946 0.09747946 0.87222129	0.94392589 0.09366921 0.09366921 0.88623445
mu_t	0.1130546 0.4173151 0.931297	-	0.2905105 0.5091869 0.7305429	0.1665314 0.5363376 0.7030081	0.2531930 0.6981052 0.7073677
lambda_t	61.97006 44.98000 48.59601 45.48474 61.82364 51.47675	-	61.71871 45.28328 48.17748 45.37308 62.15855 51.27872	61.74634 45.33749 48.26275 45.14258 62.22332 51.18338	61.66480 45.01595 48.21913 45.34420 62.18195 51.10132
psi_t	3.42175496 -3.25167111 -0.05563868 -3.25167111 7.02402871 -0.04794944 -0.05563868 -0.04794944 5.97557407	-	3.745374 -3.4351139 1.5632549 -3.435114 7.1646712 -0.9560841 1.563255 -0.9560841 6.0887514	2.9369178 -2.7752778 0.7810976 -2.7752778 6.6938954 -0.5960124 0.7810976 -0.5960124 5.8742916	3.594599 -3.413740 1.468377 -3.413740 6.854124 -1.191240 1.468377 -1.191240 6.615590

Figure 5.20: Synthetic dataset. Unc: 100 training instances, 3-dimensional input, 2-dimensional output

Data: Synthetic dataset

	Real	FA	S2FA	S3FA	MS3FA
mu_z_t	0	-	0.01866815	0.0167579	0.0197953
sigma_z_t	1	-	0.9940638	0.9904724	0.9906906
mu_t	0.5978012 0.6504156	1.567697 1.597662	0.6341849 0.8980653	0.7282556 0.9685072	0.6203953 0.9059031
lambda_t	50.02886 37.33689	42.99078 32.22177	50.00562 37.47543	50.09231 37.54379	50.06370 37.51553
diag(psi_t)	3.014497 2.438143	3.614494 2.03046	3.219408 2.106209	3.605281 2.070043	3.148793 2.027744

Figure 5.21: Synthetic dataset. FA: 100 training instances, 2-dimensional input, 1-dimensional output

	Real	PPCA	S2PPCA	S3PPCA	MS3PPCA
mu_z_t	0.5894521	-	0.5471244	0.5316495	0.5282934
sigma_z_t	4.191633	-	4.307797	4.501333	4.306638
mu_t	0.1844163 0.8506565	3.22599 3.036063	0.1967427 1.337151	0.3565684 1.426059	0.3651258 1.3083244
lambda_t	5.54558 3.140015	11.50368 6.408855	5.536671 3.105167	5.397206 3.028318	5.506781 3.069273
diag(psi_t)	6.467466 6.467466	6.240042 6.240042	6.188948 6.188948	6.782395 6.782395	6.1333 6.1333

Figure 5.22: Synthetic dataset. PPCA: 100 training instances, 2-dimensional input, 1-dimensional output

Data: Abalone dataset; 4177 instances; 8 input variables; 1 output variable (Rings - age of abalone). The first column was converted to numeric as follows: F - 1, I - 2, M - 3

Single-output regression

	S2UncFA	S3UncFA	MS3UncFA	S2FA	S3FA	MS3FA	S2PPCA	S3PPCA	MS3PPCA	Lin. Regr.
MSE	5.559	5.555	5.854	25.816	32.890	32.868	15.194	91.031	134.134	5.559
MAE	1.855	1.854	1.911	4.058	4.556	4.550	3.123	8.413	10.004	1.855
MdAE	1.570	1.567	1.630	3.439	3.790	3.777	2.560	8.764	12.187	1.570
Cor	0.698	0.698	0.696	0.586	0.575	0.574	0.544	0.071	0.020	0.698

Figure 5.23: Single-output regression: 1000 rows

	S2UncFA	S3UncFA	MS3UncFA	S2FA	S3FA	MS3FA	S2PPCA	S3PPCA	MS3PPCA	Lin. Regr.
MSE	4.533	4.533	4.557	11.876	13.603	13.482	7.715	9.718	11.097	4.533
MAE	1.610	1.610	1.615	2.767	2.984	2.969	2.072	2.288	2.440	1.610
MdAE	1.268	1.268	1.275	2.406	2.622	2.608	1.531	1.534	1.650	1.268
Cor	0.704	0.704	0.703	0.578	0.574	0.574	0.535	0.489	0.452	0.704

Figure 5.24: Single-output regression: 3000 rows

	S2UncFA	S3UncFA	MS3UncFA	S2FA	S3FA	MS3FA	S2PPCA	S3PPCA	MS3PPCA	Lin. Regr.
MSE	2.208	2.208	2.223	10.182	10.441	10.399	5.081	5.171	5.187	2.208
MAE	1.172	1.172	1.175	2.524	2.557	2.552	1.707	1.720	1.722	1.172
MdAE	0.993	0.993	1.001	2.124	2.151	2.143	1.324	1.348	1.312	0.993
Cor	0.711	0.711	0.709	0.745	0.745	0.745	0.713	0.712	0.713	0.711

Figure 5.25: Single-output regression: 4000 rows

Data: Boston House Price Dataset; 506 observations; 13 input variables; 1 output variable (MEDV - house price)

Impute missing data

Mean	MS3UncFA 13_1 case	MS3FA 13_1 case	MS3PPCA 13_1 case	MS3UncFA 7_7 case	MS3FA 7_7 case	MS3PPCA 7_7 case
247.83	79.579	233.074	246.726	79.579	83.078	92.016

Figure 5.26: Impute missing data

Data: Boston House Price Dataset; 506 observations; 13 input variables; 1 output variable (MEDV - house price)

Data augmentation

	None	S2UncFA	S2FA	S2PPCA
MSE	54.6032964912411	18.9899257631364 31.8378296477406 21.3312055759326 26.1855496650328 30.8527350903843	23.4357395906758 19.716845445672 19.6978291677237 42.1246407911356 29.8385461398302	59.6056759920788 29.267460697438 54.2180323061209 55.442101380417 46.1089478496146
MAE	5.10758484121894	3.4072616033526 4.3560238568827 3.43847852311047 3.81795446632395 3.97358282751157	3.56575826397014 3.38390650015026 3.40535929394287 5.13308341362212 4.25015201621889	5.60982706798336 4.26750744074158 5.53370310266137 5.51352112908655 5.12140261831075
MdAE	3.72780448717949	2.91929849933377 4.09071963816178 2.66932063622748 3.53323869759357 2.74798439340666	2.60419248641536 2.73645881377479 2.71624283343786 4.24704337050196 3.8221279478492	4.03473804109253 3.27361322555969 3.95293421088071 3.83565838047459 3.84794743362689
Cor	0.570245829704148	0.634686153181339 0.496262781418265 0.584592242957179 0.653144050889019 0.57473159249764	0.59827507396356 0.573740756343142 0.612232987869697 0.612635499921113 0.592321538715421	0.604759443196241 0.573012553785038 0.544968300938752 0.544014211487599 0.571269994950697

Figure 5.27: Data augmentation

Data: We generated 1000 instances with 20-dimensional input and 10-dimensional output with a diagonal Psi.

Time comparisons

- Matrix vs non-matrix form

Unit: seconds	min	mean	median	max	neval
S3FA – fit	4.548421	4.683793	4.641505	4.823499	10
MS3FA – fit	9.851952	10.253672	10.168517	11.123803	10

Figure 5.28: Time: Matrix vs non-matrix form

Data: Boston House Price Dataset; 506 observations; 13 input variables; 1 output variable (MEDV - house price); used only the first 300 instances

Time comparisons

- turboEM

S3_	stop	method	value.objfn	itr	fpeval	objfeval	convergence	elapsed.time
UncFA	objfn	em	18959.10	100	100	101	FALSE	9.57
UncFA	objfn	squarem	18959.81	9	17	11	TRUE	1.07
UncFA	objfn	pem	18953.42	100	210	733	FALSE	69.92
FA	objfn	em	20211.32	100	100	101	FALSE	9.06
FA	objfn	squarem	20211.32	13	25	14	TRUE	1.25
FA	objfn	pem	20211.32	6	22	42	TRUE	3.63
PPCA	objfn	em	33247.17	100	100	101	FALSE	8.53
PPCA	objfn	squarem	33247.14	10	18	12	TRUE	1.02
PPCA	objfn	pem	33247.14	8	26	49	TRUE	4.08
UncFA	param	em	18959.10	100	100	1	FALSE	0.20
UncFA	param	squarem	18953.82	100	199	126	FALSE	12.75
UncFA	param	pem	18953.42	100	210	740	FALSE	74.93
FA	param	em	20211.32	100	100	1	FALSE	0.19
FA	param	squarem	20211.32	15	29	16	TRUE	1.43
FA	param	pem	20211.32	6	22	42	TRUE	3.58
PPCA	param	em	33247.17	100	100	1	FALSE	0.17
PPCA	param	squarem	33247.14	13	24	15	TRUE	1.47
PPCA	param	pem	33247.14	7	24	46	TRUE	3.86

Figure 5.29: Time: turboEM

Conclusion and future work

Conclusion and future work

- Purpose:
  - unsupervised -> supervised
  - exploit: S3, MS3; others
- Future work:
  - Continue with Bernoulli
  - *FixedPoint* package
  - Compare the imputation with normal distribution
  - Image inpainting
  - Mixtures of S2FAs
  - Memory? -> Big data?