2012

Simulation study on using moment functions for sufficient dimension reduction

Lipu Tian
Michigan Technological University

Follow this and additional works at: https://digitalcommons.mtu.edu/etds
Part of the Mathematics Commons

A Simulation Study on Using Moment Functions for Sufficient Dimension Reduction

By

Lipu Tian

A REPORT

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

(Mathematical Sciences)

MICHIGAN TECHNOLOGICAL UNIVERSITY

2012

This report, "A Simulation Study on Using Moment Functions for Sufficient Dimension Reduction," is hereby approved in partial fulfillment of the requirements for the Degree of MASTER OF SCIENCE IN MATHEMATICAL SCIENCES.

Department of Mathematical Sciences

Signatures:

Report Advisor ______________________________
Andreas Artemiou

Committee Member ______________________________
Paul Ward

____
Tom Drummer

Department Chair ______________________________
Mark Gockenbach

Date ______________________________

1. Introduction

With the increase in computing storage power, researchers are able to collect and store large datasets. Thus there is a need to improve analytical techniques for large datasets as most of the times existing techniques are not adequate. Different dimension reduction techniques will be introduced to handle this. In this work, we deal with sufficient dimension reduction in regression and more specifically the use of inverse moments to recover information of the relationship between predictor (X) and response (Y). By projecting the p-dimensional predictor X onto k-dimension subspace (where k≤p), which contains the most information about response Y, and calculating the coefficient of each predictor X, the effective dimension reduction directions can be obtained. The effective dimension reduction directions under mild conditions span a subspace called the central dimension reduction subspace (CDRS). (see Cook, 1998)

Many algorithms were proposed in this regard. A set of algorithms were implemented in an effort to estimate a p x k matrix β that satisfies: Y is independent of X given β^TX. Some of those algorithms are Sliced Inverse Regression (SIR) (see Li, 1991), Sliced Average Variance Estimation (SAVE) (see Cook, Weisberg, 1991), Directional Regression (DR) (see Li, Wang, 2007), later Zhu, Zhu, Feng (2010) used cumulative slicing and proposed Cumulative Mean Estimation (CUME), Cumulative Variance Estimation (CUVE) and Cumulative Directional Regression (CUDR). The goal of this study is to combine these ideas to create a new algorithm and achieve better results. We will also utilize the idea of using all the points to do dimension reduction as in Principal Support Vector Machine (PSVM) (see Li, Artemiou, Li, 2011). Towards this direction we will modify the SIR and SAVE algorithm.

2. Literature Review

2.1 Sliced Inverse Regression

Sliced Inverse Regression (SIR) was introduced by Li (1991). SIR slices the response variable and then it calculates E(X|Y) within each slice. In that sense, since no model is assumed it results into more like a non-parametric method of estimation.

The algorithm to estimate effective dimension reduction directions via SIR is:

1. Standardize x to get z_i=$\sum_{xx}^{-1/2}(x_i - \bar{x})$
2. Divide range of y_i into H non overlapping slices, and count the number of observations n_s fall into each slice, where I_{H_s} is the indicator function of this slice: n_s=$\sum_{i=1}^{n} I_{H_s}\,(y_i)$
3. Calculate the sample mean within each slice: $\bar{z}_s$=$\frac{1}{n_s}\sum_{i=1}^{p} z_i\, I_{H_s}(y_i)$
4. Conduct a principal component analysis on $\bar{z}_s$ to form the candidate matrix: $\hat{\mathrm{V}}$=$\mathrm{n}^{-1}\sum_{\mathrm{s}=1}^{\mathrm{H}} \mathrm{n}_\mathrm{s}\bar{\mathrm{z}}_\mathrm{s}\,\bar{\mathrm{z}}_\mathrm{s}{}^{\mathrm{T}}$
5. Calculate the estimates for the directions (estimated factor coefficients) based on eigenvectors $\hat{v}_i$ of $\hat{V}$: $\hat{\beta}_i$=$\sum_{xx}^{-\frac{1}{2}}\hat{v}_i$

2.2 Sliced Average Variance Estimation

In SIR algorithm, when the response variable is symmetric about some predictor variable around zero, the within-slice means will all be zeros. Thus, the eigenvalues of the covariance matrix formed from the slice mean vectors will have the same values, which will cause SIR to fail to obtain the correct directions. Under such circumstance, although the slice means are zeros for all y, the slice variances are very likely to vary from slice to slice. Therefore, by using second or higher moments the correct directions can be found.

The algorithm to estimate effective dimension reduction directions via SAVE is:

1. Standardize x to get z_i=$\sum_{xx}^{-1/2}(x_i - \bar{x})$
2. Divide range of y_i into H non overlapping slices, and count the number of observations n_s fall into each slice, where I_{H_s} is the indicator function of this slice: n_s=$\sum_{i=1}^{p} I_{H_s}\,(y_i)$

3. Calculate the sample variance within each slice: $\widehat{v}_s$=Var(Z|Y_s)
4. Conduct a principal component analysis on $\widehat{v}_s$ to form the candidate matrix: $\hat{S}=n^{-1}\sum_{s=1}^{H} n_s\left(I_p-\widehat{v}_s\right)\left(I_p-\widehat{v}_s\right)^T$
5. Calculate the estimates for the directions (estimated factor coefficients) based on eigenvectors $\widehat{h}_\iota$ of $\widehat{H}$: $\hat{\beta}_i=\Sigma_{xx}^{-\frac{1}{2}}\widehat{h}_\iota$

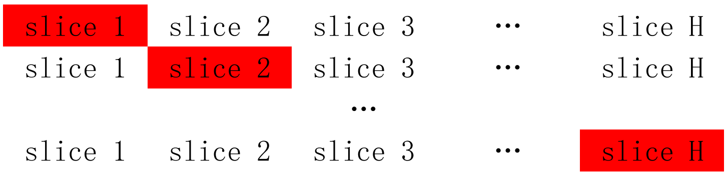
3. Methods

3.1 Existing Methods

Sliced Inverse Regression (SIR)/ Sliced Average Variance Estimation (SAVE)

In SIR, we slice response variable Y into n slices and we calculate the inverse mean within each slice as Figure 1 shown. In SAVE, we calculate the inverse variance within each slice.

Figure 1

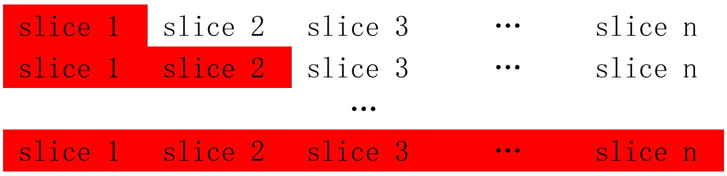


slice 1	slice 2	slice 3	···	slice H
slice 1	**slice 2**	slice 3	···	slice H
		···		
slice 1	slice 2	slice 3	···	**slice H**

Cumulative Mean Estimation (CUME)/Cumulative Variance Estimation (CUVE)

We consider each point a slice and we take n cumulative averages/variances. This way it reduces the necessity to tune the number of slices, a parameter to which SIR and especially SAVE and DR are highly sensitive.

Figure 2



slice 1	slice 2	slice 3	···	slice n
slice 1	**slice 2**	slice 3	···	slice n
		···		
slice 1	**slice 2**	**slice 3**	**···**	**slice n**

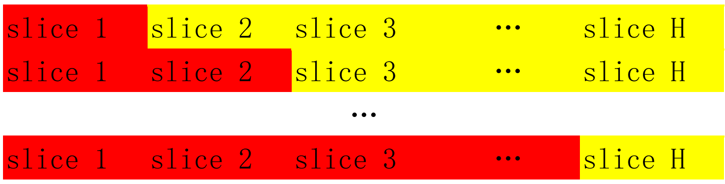
3.2 New Methodology

Li, Artemiou, Li (2011) used machine learning algorithms instead of inverse moments to do dimension reduction. They proposed two ways to implement their ideas: "Left vs. Right" LVR and "One vs. Another" OVA. We will implement these two algorithms with inverse moments to improve the performances of SIR and SAVE.

Left vs. Right (LVR)

In LVR, each dividing point we calculate the inverse mean of the slices on the left: E(X|Y≤y) and the inverse mean on the right: E(X|Y>y), then we take the difference: m_d= E(X|Y>y)- E(X|Y≤y) as illustrated in Figure 3. In SAVE, we take a slightly different approach that we will discuss later since we are dealing with covariance matrices.

Figure 3



🟥 slice 1	🟨 slice 2	🟨 slice 3	🟨 ···	🟨 slice H
🟥 slice 1	🟥 slice 2	🟨 slice 3	🟨 ···	🟨 slice H
		···		
🟥 slice 1	🟥 slice 2	🟥 slice 3	🟥 ···	🟨 slice H

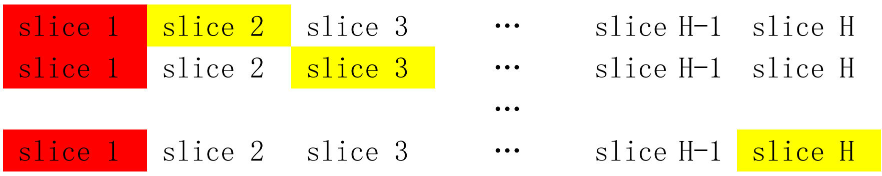
The algorithm to estimate effective dimension reduction directions using LVRSIR is:

1. Standardize x to get z_i=$\Sigma_{xx}^{-1/2}(x_i-\bar{x})$
2. Divide range of y_i into H non overlapping slices, and count the number of observations n_s fall into each slice, where I_{H_s} is the indicator function of this slice: n_s=$\sum_{i=1}^{n} I_{H_s}(y_i)$
3. For each of the H-1 cutoff points between the slices, calculate : $\bar{z}_{LVR}$= E($\bar{z}_s$|Y>y) - E($\bar{z}_s$|Y≤y) (see Figure 3)
4. Conduct a principal component analysis on $\bar{z}_{LVR}$ to form the candidate matrix: $\hat{V}$=$\sum_{s=1}^{H-1}\hat{w}\bar{z}_{LVR}\,\bar{z}_{LVR}^{T}$, where $\hat{w}$ is the weight
5. Calculate the estimates for the directions (estimated factor coefficients) based on eigenvectors $\hat{v}_i$ of $\hat{V}$: $\hat{\beta}_i$=$\Sigma_{xx}^{-\frac{1}{2}}\hat{v}_i$

One vs. Another (OVA)

If there are H slices, there are $\binom{H}{2}$ pairs. We take different pairs each time and we find the difference between E(X|Y=i) and E(X|Y=j), where i≠j.

Figure 4



slice 1	slice 2	slice 3	⋯	slice H−1	slice H
slice 1	slice 2	slice 3	⋯	slice H−1	slice H
			⋯		
slice 1	slice 2	slice 3	⋯	slice H−1	slice H

4. Simulations

4.1 Sample Generation

Five models are used to generate data points for this study, which are:

$$y = x_1 + x_2 + \sigma\epsilon \qquad (1)$$

$$y = \frac{x_1}{0.5+(x_2+1)^2} + \sigma\epsilon \qquad (2)$$

$$y = x_1(x_1 + x_2+1) + \sigma\epsilon \qquad (3)$$

$$y = {x_1}^2 + x_2 + \sigma\epsilon \qquad (4)$$

$$y = ({x_1}^2 + {x_2}^2)^{1/2} \cdot \log[({x_1}^2 + {x_2}^2)^{1/2}] + \sigma\epsilon \qquad (5)$$

* σ is the scaling factor for error ϵ

The default setting for dimension p is 10, and ϵ, x_1, x_2, $\ldots x_p$ are generated from standard normal distribution N(0,1), assuming they are independent to each other.

We will use trace correlation: trace $[(P_\beta{}^T P_{\hat{\beta}})/k]$, (where $P_\beta = \beta(\beta^T\beta)^{-1}\beta^T$, $P_{\hat{\beta}} = \hat{\beta}(\hat{\beta}^T\hat{\beta})^{-1}\hat{\beta}^T$, β and $\hat{\beta}$ are the true beta and estimated directions) as a measurement to compare the performance between the different methods. The value of trace correlation ranges between zero and one, and the closest to 1 the better it is.

We will do simulations with different parameters, which will give us a general idea about how the algorithms and slice calculation methods perform. The different parameters we are going to try are: number of slices: 5, 10 and 20; dimensions of predictor variables: 10, 20 and 30; sigma (scaling factor for random error): 0.2, 0.5, 1 and 2. To have a reasonably accurate estimation, each test will collect the mean from

500 simulations with sample size of 100 and 225 for SIR and SAVE respectively.

4.2 LVRSIR

Our results for LVR are summarized in tables 1 through 4. In Table 1 we see that as sigma increases the performance decreases for all models. In all the cases but one, the LVR algorithm performs better than CUME and classic SIR. In Table 2 we see that as the number of slices increases the performance of LVR increases, which for SIR seems to have fluctuating behavior. CUME is not affected by the number of slices. In any case LVR seems to perform better. In Table 3 we can see that as dimension increases the performance decreases as expected. It is clear that LVR performs better than SIR and CUME.

We have tried an alternative weighting method for LVR in a hope to improve its performance. Instead of calculating the difference between the means of left and right, we obtain the summation of them. In Table 4 LVR with plus weighting method (LVR 2) does not appear to have a better estimation than LVR with original weighting method.

We have also tried the OVA algorithm. The candidate matrix of OVA is $\widehat{\mathrm{V}}$=$\sum_{s=1}^{\binom{H}{2}} \widehat{w}\bar{z}_{\mathrm{OVA}}\,\bar{z}_{\mathrm{OVA}}{}^{\mathrm{T}}$, where $\widehat{w}$ is the weight and we have tried the following six weighting schemes.

OVA 1: 1

OVA 2: $\frac{2}{n}$

OVA 3: $\frac{1}{||mean\ 1 - mean\ 2||}$

OVA 4: $||mean\ 1 - mean\ 2||$

OVA 5: $\frac{1}{||mean\ 1 - mean\ 2||^2}$

OVA 6: $||mean\ 1 - \ mean\ 2||^2$

* n is the number of slices

In Table 5 it seems OVA with weighting method #6 has a slightly more stable performance with higher dimensions. But there is no significant improvement over OVA with original weight and it is no better than classic SIR.

Table 1. Comparison between SIR, LVR and CUME with different sigmas

dimensions=10, slices=10				
model	sigma	SIR	LVR	CUME
1	0.2	.993 (.004)	.993 (.004)	.987 (.008)
	0.5	.980 (.012)	.982 (.010)	.976 (.012)
	1	.937 (.031)	.944 (.027)	.938 (.033)
	2	.742 (.159)	.818 (.088)	.808 (.094)
2	0.2	.824 (.096)	.854 (.065)	.863 (.067)
	0.5	.680 (.130)	.760 (.096)	.742 (.107)
	1	.517 (.119)	.603 (.115)	.577 (.118)
	2	.346 (.122)	.413 (.118)	.399 (.113)
3	0.2	.610 (.170)	.711 (.132)	.689 (.134)
	0.5	.518 (.162)	.636 (.144)	.603 (.141)
	1	.433 (.149)	.537 (.151)	.512 (.149)
	2	.339 (.133)	.413 (.140)	.395 (.137)
4	0.2	.523 (.088)	.560 (.106)	.538 (.089)
	0.5	.512 (.087)	.548 (.108)	.530 (.096)
	1	.476 (.100)	.513 (.107)	.492 (.095)
	2	.373 (.116)	.432 (.110)	.419 (.102)
5	0.2	.119 (.150)	.136 (.162)	.122 (.138)
	0.5	.109 (.137)	.135 (.163)	.113 (.139)
	1	.110 (.146)	.126 (.151)	.111 (.135)
	2	.113 (.134)	.120 (.140)	.114 (.141)

Table 2. Comparison between SIR, LVR and CUME with different slices

sigma=1, dimensions=10				
model	slices	SIR	LVR	CUME
1	5	.930 (.036)	.934 (.035)	.938 (.031)
	10	.937 (.032)	.945 (.029)	.938 (.031)
	20	.928 (.037)	.946 (.028)	.938 (.031)
2	5	.519 (.120)	.553 (.118)	.569 (.121)
	10	.520 (.117)	.600 (.116)	.569 (.121)
	20	.486 (.122)	.607 (.119)	.569 (.121)
3	5	.444 (.153)	.492 (.156)	.510 (.148)
	10	.431 (.140)	.537 (.142)	.510 (.148)
	20	.387 (.155)	.560 (.153)	.510 (.148)
4	5	.472 (.088)	.488 (.096)	.495 (.094)
	10	.464 (.086)	.516 (.101)	.495 (.094)
	20	.454 (.103)	.532 (.117)	.495 (.094)
5	5	.102 (.127)	.114 (.133)	.118 (.141)
	10	.108 (.140)	.131 (.155)	.118 (.141)
	20	.102 (.135)	.149 (.173)	.118 (.141)

Table 3. Comparison between SRI, LVR and CUME with different dimensions

sigma=1, slices=10				
model	dimensions	SIR	LVR	CUME
1	10	.935 (.033)	.944 (.028)	.938 (.031)
	20	.861 (.051)	.882 (.042)	.868 (.045)
	30	.771 (.072)	.808 (.055)	.787 (.060)
2	10	.511 (.117)	.596 (.118)	.570 (.119)
	20	.355 (.096)	.434 (.093)	.410 (.094)
	30	.259 (.086)	.343 (.083)	.318 (.081)
3	10	.429 (.151)	.533 (.155)	.508 (.156)
	20	.266 (.120)	.361 (.121)	.332 (.115)
	30	.182 (.104)	.269 (.111)	.245 (.103)
4	10	.469 (.093)	.508 (.110)	.494 (.101)
	20	.334 (.084)	.382 (.080)	.369 (.071)
	30	.250 (.080)	.306 (.069)	.297 (.066)
5	10	.111 (.133)	.130 (.144)	.116 (.134)
	20	.058 (.082)	.071 (.091)	.061 (.077)
	30	.036 (.052)	.046 (.063)	.042 (.059)

Table 4. Comparison between LVR and LVR 2 with different dimensions

sigma=2, slices=10			
model	dimensions	LVR	LVR 2
1	10	.818 (.087)	.768 (.107)
	20	.646 (.116)	.583 (.130)
	30	.522 (.118)	.453 (.130)
2	10	.413 (.113)	.340 (.118)
	20	.262 (.089)	.248 (.093)
	30	.183 (.074)	.173 (.077)
3	10	.414 (.142)	.409 (.148)
	20	.268 (.109)	.262 (.110)
	30	.181 (.087)	.177 (.089)
4	10	.429 (.101)	.416 (.108)
	20	.278 (.085)	.258 (.089)
	30	.200 (.069)	.180 (.069)
5	10	.110 (.138)	.120 (.147)
	20	.056 (.073)	.059 (.078)
	30	.038 (.051)	.038 (.050)

Table 5. Comparison between SIR and OVAs with different dimensions

sigma=0.2, slices=10								
model	dimensions	SIR	OVA 1	OVA 2	OVA 3	OVA 4	OVA 5	OVA 6
1	10	.993 (.004)	.993 (.004)	.993 (.004)	.991 (.005)	.992 (.005)	.986 (.009)	.990 (.006)
	20	.983 (.008)	.983 (.008)	.983 (.008)	.981 (.009)	.981 (.009)	.974 (.014)	.978 (.011)
	30	.969 (.012)	.969 (.012)	.969 (.012)	.966 (.013)	.966 (.014)	.956 (.019)	.960 (.018)
2	10	.818 (.095)	.818 (.095)	.818 (.095)	.810 (.101)	.820 (.092)	.793 (.112)	.820 (.091)
	20	.668 (.097)	.668 (.097)	.668 (.097)	.662 (.100)	.669 (.096)	.648 (.102)	.666 (.095)
	30	.541 (.094)	.541 (.094)	.541 (.094)	.536 (.094)	.543 (.093)	.524 (.094)	.542 (.092)
3	10	.606 (.164)	.606 (.164)	.606 (.164)	.586 (.164)	.616 (.165)	.538 (.160)	.620 (.165)
	20	.409 (.140)	.409 (.140)	.409 (.140)	.393 (.136)	.418 (.142)	.359 (.133)	.424 (.145)
	30	.291 (.128)	.291 (.128)	.291 (.128)	.277 (.123)	.299 (.130)	.254 (.116)	.303 (.130)
4	10	.528 (.094)	.528 (.094)	.528 (.094)	.524 (.089)	.530 (.097)	.515 (.082)	.530 (.099)
	20	.434 (.058)	.434 (.058)	.434 (.058)	.431 (.057)	.435 (.059)	.424 (.057)	.434 (.060)
	30	.365 (.063)	.365 (.063)	.365 (.063)	.363 (.063)	.365 (.064)	.357 (.063)	.364 (.064)
5	10	.109 (.144)	.109 (.144)	.109 (.144)	.096 (.130)	.115 (.153)	.086 (.111)	.120 (.159)
	20	.060 (.092)	.060 (.092)	.060 (.092)	.056 (.085)	.063 (.096)	.052 (.078)	.064 (.099)
	30	.037 (.052)	.037 (.052)	.037 (.052)	.035 (.049)	.039 (.054)	.034 (.048)	.040 (.055)

* Each of the tables has the mean trace estimate out of 500 simulations and in parenthesis, the standard deviation of the estimate.

4.3 LVRSAVE

To test the performance of these new weighting methods, 6 new models are added to the existing 5 models used for LVRSIR, which are symmetric around zero and are models SAVE will be more appropriate to be used.

y=$(\beta_1^T x/4)^2$ + log(1 + $|\beta_2^T x|^2$) + $\sigma\epsilon$ (6)

y=$(\beta_1^T x/2)^2$ + 4sin(β_2^Tx/4) + $\sigma\epsilon$ (7)

y=0.5$(\beta_1^T x)^3$ + 0.5$(1 + \beta_2^T x)^2$ + $\sigma\epsilon$ (8)

y=0.4$(2 + \beta_1^T x)^3$ + 0.5$(1 + \beta_2^T x/2)^2$ + $\sigma\epsilon$ (9)

y=$|\beta_2^T x/2|^{1/2}$ + $\sigma\{(\beta_1^T x)^2 + 1\}\,\epsilon$ (10)

y=exp[$\sigma(\beta_1^T x + 1)^3$ + σ(1+$(\beta_2^T x/2)^2$) + $\sigma\epsilon$ (11)

* ${\beta_1}^T = [1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad \cdots]$, ${\beta_2}^T = [1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 3 \quad \cdots]$, σ is the scaling factor for error ϵ

The results for SAVE are summarized in tables 6 through 8. The candidate matrix for SAVE is $\hat{\mathrm{S}}=\mathrm{n}^{-1}\sum_{\mathrm{s}=1}^{\mathrm{n}} \mathrm{n_s}\left(I_p - \widehat{v_s}\right)\left(I_p - \widehat{v_s}\right)^{\mathrm{T}}$. Since we are dealing with covariance matrices, we would try different approaches than with SIR. Here we have tried four different approaches. In LVR 1 and LVR 2, $\widehat{v_s}$ is replaced with c_1 and c_2 respectively. In LVR 3 and LVR 4, $I_p - \widehat{v_s}$ is replaced with c_3 and c_4 respectively.

LVR 1: c_1 = variance of right + variance of left

LVR 2: c_2 = variance of right – variance of left

LVR 3: c_3 = (I_p – variance of right) + (I_p – variance of left)

LVR 4: c_4 = (I_p – variance of right) – (I_p – variance of left)

In Table 6 it is clear that as dimension increases the performance decreases, which is expected. LVR 3 and SAVE perform significantly better than other algorithms, where LVR 3 performs slightly better than SAVE in most of the conditions. In Table 7 the algorithms perform worse with larger number of slices, except CUVE, whose performance does not depend on number of slices. LVR 3 performs the best among the other algorithms and it is relatively less sensitive to number of slices. In Table 8 SAVE performs fairly well with low dimensions, but as the dimension goes up its performance drops dramatically. LVR 3 and CUVE are less sensitive to increase in dimension, where LVR 3 performs better in lower dimensions and CUVE performs better in higher dimensions.

Table 6. Comparison between SAVE, CUVE and LVRs with different sigmas

dimensions=10, slices=10							
model	sigma	SAVE	LVR 1	LVR 2	LVR 3	LVR 4	CUVE
1	0.2	.978 (.084)	.006 (.009)	.025 (.040)	.991 (.005)	.018 (.026)	.940 (.034)
	0.5	.859 (.237)	.006 (.009)	.028 (.043)	.984 (.008)	.021 (.030)	.915 (.057)
	1	.151 (.203)	.009 (.012)	.043 (.060)	.961 (.018)	.027 (.038)	.770 (.184)
	2	.046 (.067)	.019 (.027)	.067 (.091)	.576 (.326)	.048 (.067)	.294 (.247)
2	0.2	.476 (.164)	.018 (.012)	.151 (.110)	.821 (.179)	.267 (.147)	.606 (.130)
	0.5	.242 (.130)	.026 (.021)	.228 (.130)	.600 (.154)	.284 (.149)	.592 (.139)
	1	.190 (.104)	.426 (.064)	.106 (.083)	.545 (.122)	.498 (.063)	.538 (.093)
	2	.202 (.112)	.165 (.098)	.271 (.120)	.306 (.142)	.293 (.148)	.354 (.159)
3	0.2	.480 (.093)	.323 (.156)	.211 (.123)	.626 (.157)	.577 (.111)	.708 (.140)
	0.5	.476 (.094)	.353 (.151)	.265 (.110)	.563 (.107)	.615 (.120)	.691 (.138)
	1	.242 (.231)	.231 (.220)	.020 (.029)	.311 (.263)	.356 (.279)	.372 (.279)
	2	.427 (.131)	.383 (.146)	.221 (.117)	.525 (.112)	.558 (.110)	.567 (.109)
4	0.2	.598 (.131)	.470 (.023)	.038 (.031)	.941 (.040)	.523 (.069)	.827 (.118)
	0.5	.529 (.081)	.469 (.026)	.050 (.044)	.918 (.068)	.518 (.062)	.756 (.144)
	1	.490 (.052)	.461 (.032)	.066 (.056)	.778 (.159)	.507 (.058)	.630 (.134)
	2	.453 (.073)	.426 (.064)	.106 (.083)	.545 (.122)	.498 (.063)	.538 (.093)
5	0.2	.456 (.320)	.441 (.303)	.007 (.010)	.458 (.321)	.472 (.332)	.470 (.343)
	0.5	.439 (.313)	.409 (.299)	.009 (.012)	.437 (.317)	.457 (.324)	.484 (.322)
	1	.395 (.289)	.367 (.278)	.011 (.016)	.422 (.304)	.447 (.312)	.461 (.316)
	2	.242 (.231)	.231 (.220)	.020 (.029)	.311 (.263)	.356 (.279)	.372 (.279)
6	0.2	.586 (.111)	.545 (.101)	.048 (.071)	.671 (.144)	.709 (.141)	.711 (.139)
	0.5	.568 (.099)	.525 (.094)	.033 (.033)	.630 (.135)	.685 (.139)	.695 (.137)
	1	.533 (.088)	.485 (.092)	.034 (.026)	.580 (.121)	.633 (.130)	.648 (.135)
	2	.429 (.105)	.348 (.117)	.047 (.040)	.475 (.098)	.524 (.105)	.547 (.113)
7	0.2	.697 (.169)	.344 (.093)	.044 (.038)	.874 (.080)	.438 (.054)	.819 (.100)
	0.5	.656 (.178)	.342 (.094)	.044 (.032)	.870 (.084)	.438 (.057)	.794 (.115)
	1	.520 (.172)	.331 (.102)	.051 (.042)	.847 (.097)	.434 (.056)	.726 (.145)
	2	.343 (.124)	.289 (.111)	.062 (.050)	.776 (.126)	.409 (.077)	.551 (.147)
8	0.2	.555 (.131)	.396 (.073)	.043 (.034)	.876 (.077)	.498 (.053)	.839 (.099)
	0.5	.565 (.132)	.393 (.087)	.044 (.032)	.870 (.083)	.505 (.063)	.837 (.089)
	1	.536 (.117)	.397 (.075)	.045 (.033)	.871 (.074)	.493 (.053)	.832 (.094)
	2	.503 (.106)	.395 (.076)	.048 (.038)	.856 (.084)	.492 (.049)	.824 (.103)

9	0.2	.652 (.172)	.085 (.071)	.048 (.038)	.736 (.160)	.456 (.045)	.877 (.070)
	0.5	.622 (.176)	.084 (.078)	.045 (.035)	.740 (.155)	.456 (.043)	.879 (.059)
	1	.618 (.175)	.084 (.080)	.048 (.039)	.716 (.161)	.455 (.048)	.874 (.072)
	2	.544 (.168)	.094 (.084)	.048 (.035)	.695 (.152)	.442 (.053)	.855 (.081)
10	0.2	.728 (.127)	.629 (.133)	.268 (.096)	.703 (.130)	.811 (.083)	.762 (.108)
	0.5	.558 (.120)	.520 (.082)	.288 (.100)	.529 (.101)	.615 (.140)	.636 (.153)
	1	.503 (.087)	.503 (.064)	.304 (.102)	.499 (.077)	.524 (.098)	.516 (.116)
	2	.490 (.072)	.507 (.066)	.323 (.101)	.494 (.068)	.498 (.073)	.474 (.084)
11	0.2	.662 (.139)	.104 (.084)	.054 (.041)	.620 (.123)	.383 (.093)	.831 (.101)
	0.5	.655 (.140)	.102 (.084)	.054 (.040)	.623 (.126)	.384 (.087)	.835 (.098)
	1	.658 (.138)	.097 (.081)	.050 (.038)	.614 (.128)	.383 (.094)	.828 (.104)
	2	.663 (.140)	.100 (.084)	.056 (.044)	.619 (.125)	.384 (.093)	.833 (.099)

Table 7. Comparison between SAVE, CUVE and LVRs with different slices

dimensions=10, sigma=0.2							
model	slices	SAVE	LVR 1	LVR 2	LVR 3	LVR 4	CUVE
1	5	.992 (.004)	.007 (.010)	.028 (.040)	.989 (.006)	.019 (.029)	.941 (.035)
	10	.975 (.095)	.006 (.008)	.022 (.032)	.990 (.005)	.017 (.023)	.941 (.035)
	20	.007 (.010)	.006 (.008)	.021 (.035)	.991 (.005)	.014 (.019)	.941 (.035)
2	5	.652 (.132)	.020 (.015)	.171 (.119)	.910 (.094)	.271 (.138)	.608 (.132)
	10	.477 (.161)	.018 (.012)	.152 (.110)	.848 (.170)	.288 (.147)	.608 (.132)
	20	.144 (.102)	.016 (.011)	.142 (.109)	.689 (.203)	.240 (.153)	.608 (.132)
3	5	.547 (.089)	.328 (.143)	.238 (.124)	.639 (.140)	.593 (.118)	.706 (.142)
	10	.479 (.082)	.328 (.151)	.214 (.122)	.619 (.160)	.578 (.108)	.706 (.142)
	20	.377 (.142)	.336 (.164)	.220 (.129)	.557 (.134)	.576 (.121)	.706 (.142)
4	5	.866 (.109)	.472 (.023)	.042 (.035)	.948 (.020)	.524 (.065)	.822 (.121)
	10	.606 (.133)	.472 (.022)	.040 (.037)	.945 (.022)	.531 (.076)	.822 (.121)
	20	.481 (.050)	.467 (.027)	.066 (.085)	.901 (.098)	.519 (.068)	.822 (.121)
5	5	.447 (.331)	.417 (.310)	.008 (.010)	.438 (.326)	.455 (.335)	.473 (.339)
	10	.438 (.314)	.422 (.230)	.006 (.009)	.451 (.313)	.464 (.324)	.473 (.339)
	20	.432 (.308)	.436 (.309)	.008 (.016)	.456 (.322)	.474 (.340)	.473 (.339)
6	5	.609 (.112)	.544 (.087)	.029 (.020)	.676 (.138)	.691 (.138)	.712 (.136)
	10	.578 (.102)	.541 (.095)	.046 (.073)	.666 (.139)	.709 (.133)	.712 (.136)
	20	.534 (.084)	.537 (.098)	.107 (.140)	.629 (.140)	.697 (.139)	.712 (.136)
7	5	.826 (.102)	.317 (.099)	.048 (.042)	.873 (.070)	.433 (.055)	.817 (.102)
	10	.679 (.173)	.345 (.096)	.043 (.037)	.877 (.082)	.441 (.049)	.817 (.102)
	20	.344 (.113)	.363 (.088)	.040 (.033)	.861 (.091)	.438 (.049)	.817 (.102)
8	5	.831 (.114)	.404 (.062)	.047 (.033)	.892 (.052)	.501 (.057)	.836 (.098)
	10	.557 (.127)	.393 (.076)	.044 (.032)	.878 (.073)	.499 (.057)	.836 (.098)
	20	.396 (.084)	.393 (.076)	.043 (.031)	.819 (.132)	.494 (.058)	.836 (.098)
9	5	.857 (.087)	.117 (.091)	.043 (.032)	.853 (.100)	.462 (.042)	.880 (.062)
	10	.641 (.169)	.084 (.084)	.045 (.034)	.740 (.165)	.458 (.048)	.880 (.062)
	20	.284 (.118)	.078 (.070)	.043 (.031)	.539 (.150)	.446 (.051)	.880 (.062)
10	5	.765 (.114)	.692 (.126)	.180 (.104)	.743 (.116)	.747 (.119)	.755 (.116)
	10	.725 (.123)	.614 (.129)	.260 (.102)	.703 (.130)	.813 (.086)	.755 (.116)
	20	.583 (.147)	.579 (.124)	.306 (.087)	.627 (.139)	.805 (.090)	.755 (.116)
11	5	.755 (.128)	.118 (.094)	.060 (.047)	.687 (.141)	.390 (.085)	.833 (.100)
	10	.665 (.136)	.098 (.079)	.052 (.041)	.623 (.131)	.381 (.086)	.833 (.100)
	20	.228 (.116)	.095 (.085)	.052 (.040)	.572 (.098)	.372 (.098)	.833 (.100)

Table 8. Comparison between SAVE, CUVE and LVRs with different dimensions

slices=10, sigma=0.2							
model	dimensions	SAVE	LVR 1	LVR 2	LVR 3	LVR 4	CUVE
1	10	.976 (.090)	.006 (.007)	.024 (.035)	.991 (.005)	.019 (.027)	.938 (.037)
	20	.007 (.009)	.006 (.009)	.011 (.015)	.980 (.008)	.008 (.011)	.813 (.103)
	30	.005 (.007)	.006 (.008)	.008 (.010)	.958 (.064)	.007 (.011)	.619 (.175)
2	10	.476 (.156)	.017 (.012)	.145 (.107)	.845 (.167)	.268 (.142)	.605 (.131)
	20	.067 (.052)	.015 (.010)	.040 (.033)	.487 (.019)	.032 (.024)	.384 (.081)
	30	.038 (.030)	.015 (.012)	.024 (.018)	.455 (.051)	.020 (.014)	.258 (.097)
3	10	.487 (.083)	.321 (.147)	.225 (.119)	.638 (.156)	.588 (.116)	.713 (.141)
	20	.290 (.139)	.216 (.137)	.102 (.085)	.381 (.110)	.471 (.074)	.572 (.104)
	30	.160 (.119)	.142 (.109)	.057 (.055)	.229 (.127)	.380 (.087)	.503 (.067)
4	10	.598 (.132)	.468 (.023)	.039 (.032)	.941 (.026)	.526 (.079)	.822 (.124)
	20	.424 (.037)	.411 (.038)	.022 (.016)	.746 (.165)	.460 (.033)	.576 (.110)
	30	.345 (.067)	.344 (.059)	.017 (.013)	.430 (.111)	.413 (.033)	.479 (.074)
5	10	.480 (.322)	.465 (.310)	.007 (.012)	.496 (.328)	.512 (.335)	.505 (.345)
	20	.395 (.281)	.352 (.258)	.006 (.008)	.407 (.291)	.439 (.314)	.462 (.323)
	30	.273 (.222)	.244 (.205)	.005 (.008)	.321 (.243)	.384 (.278)	.409 (.293)
6	10	.585 (.110)	.540 (.094)	.043 (.068)	.665 (.142)	.706 (.139)	.711 (.135)
	20	.467 (.054)	.431 (.058)	.018 (.012)	.483 (.079)	.537 (.090)	.566 (.093)
	30	.384 (.053)	.352 (.054)	.014 (.010)	.399 (.049)	.462 (.055)	.507 (.064)
7	10	.689 (.173)	.347 (.091)	.044 (.036)	.879 (.072)	.439 (.059)	.813 (.103)
	20	.220 (.106)	.214 (.098)	.020 (.015)	.700 (.121)	.322 (.074)	.528 (.135)
	30	.104 (.080)	.126 (.078)	.016 (.012)	.551 (.110)	.221 (.084)	.337 (.108)
8	10	.563 (.140)	.392 (.081)	.042 (.030)	.872 (.078)	.494 (.053)	.842 (.096)
	20	.298 (.104)	.256 (.104)	.023 (.017)	.591 (.172)	.414 (.051)	.612 (.109)
	30	.161 (.101)	.159 (.094)	.019 (.013)	.290 (.132)	.331 (.075)	.489 (.086)
9	10	.648 (.174)	.091 (.084)	.044 (.031)	.746 (.156)	.461 (.045)	.880 (.065)
	20	.159 (.094)	.039 (.038)	.023 (.015)	.506 (.062)	.272 (.117)	.652 (.138)
	30	.079 (.063)	.024 (.021)	.016 (.011)	.449 (.054)	.094 (.083)	.421 (.130)
10	10	.728 (.129)	.622 (.135)	.263 (.105)	.705 (.139)	.812 (.092)	.761 (.113)
	20	.397 (.137)	.442 (.093)	.173 (.080)	.451 (.116)	.573 (.134)	.535 (.119)
	30	.216 (.099)	.319 (.077)	.123 (.065)	.301 (.091)	.367 (.117)	.387 (.109)
11	10	.659 (.143)	.098 (.084)	.058 (.046)	.619 (.126)	.384 (.089)	.832 (.103)
	20	.119 (.085)	.045 (.039)	.026 (.018)	.507 (.047)	.168 (.108)	.583 (.135)
	30	.061 (.051)	.031 (.027)	.020 (.015)	.475 (.028)	.074 (.068)	.393 (.111)

* Each of the tables has the mean trace estimate out of 500 simulations and in parenthesis, the standard deviation of the estimate.

5. Real Data Test

Ecoli data set from University of California, Irvine (see Horton, Nakai, 1996) was used for our real data test. In this data set there were 7 predictor variables and 1 response variable with a total number of 336 data points collected. The predictor variables were score ratings based on different methods, where X_3 and X_4 were discrete and the rest of them were continuous. The response variable denotes the location site of proteins on the cell, which was discrete with following classifications: 1=cytoplasm, 2=inner membrane without signal sequence, 3= inner membrane, cleavable signal sequence, 4= inner membrane lipoprotein, 5=inner membrane, uncleavable signal sequence, 6=outer membrane, 7=outer membrane lipoprotein, 8=perisplasm.

Based on this data set we were trying to see if we can find directions that separate the protein by localization site. In SIR, after the estimated directions (estimated coefficients) were obtained, we used the betas corresponding to the largest two eigenvalues to obtain SIR1 and SIR2, where SIR1=standardized X*β_{sir1} and SIR2= standardized X*β_{sir2}. Similarly, we calculated LVR1, LVR2, CUME1 and CUME2 using β_{lvr1}, β_{lvr2}, β_{cume1} and β_{cume2}. Then we plotted SIR1 vs. SIR2, LVR1 vs. LVR2 and CUME1 vs. CUME2 in order to identify possible group patterns, where different colors mark response from different slices. The same methodology was used to compare LVR with SAVE and CUVE.

Figure 5. SIR1 vs. SIR2

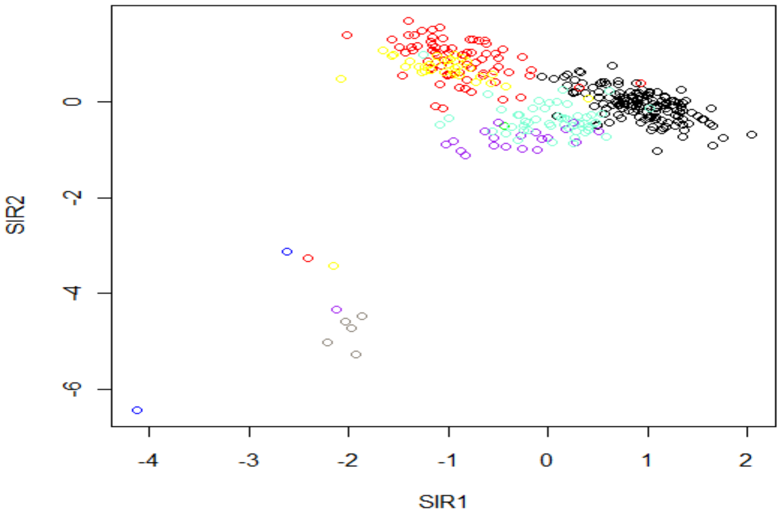


Figure 6. CUME1 vs. CUME2

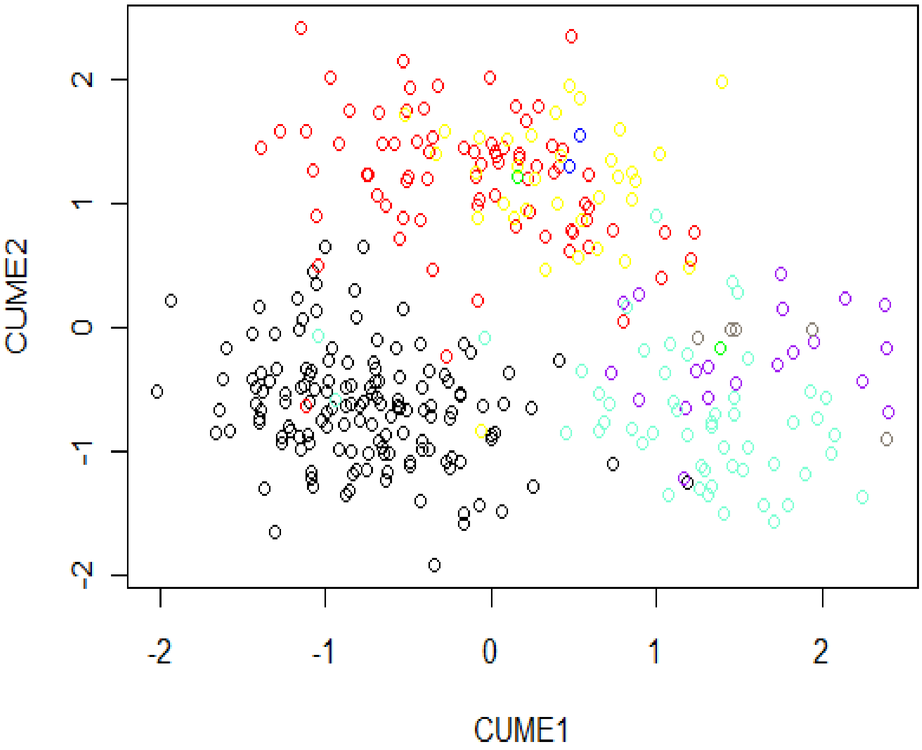
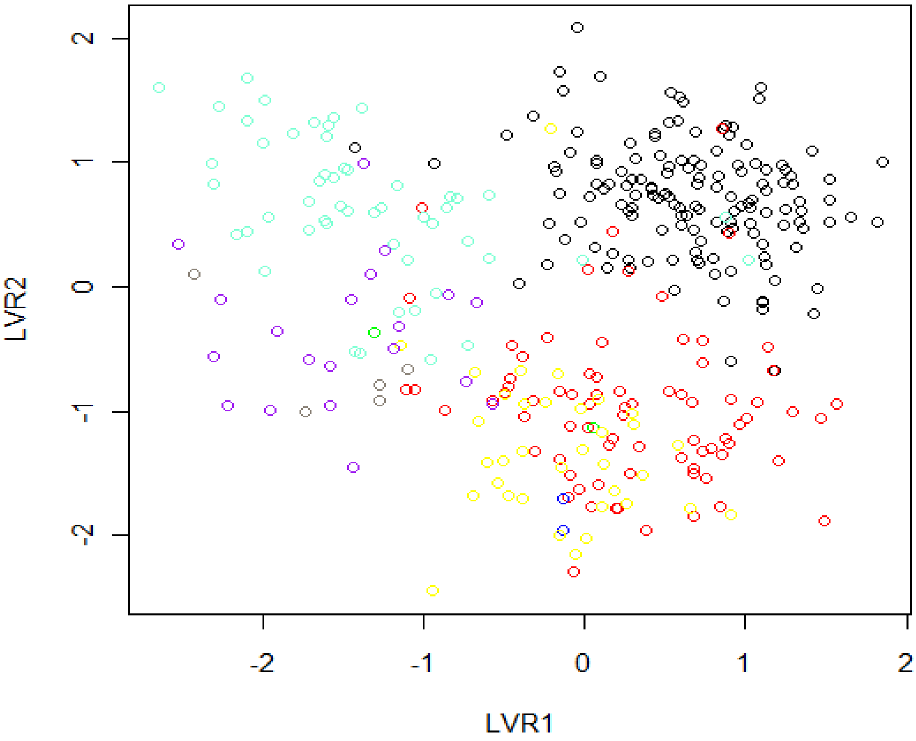


Figure 7. LVR1 vs. LVR2



* Different colors were used to mark different response groups, where black=1, red=2, green=3, blue=4, yellow=5, purple=6, grey=7, light blue=8.

In Figure 5, it showed a fairly clear group pattern, but there were 10 data points located apart from the major clusters. Those 10 points had different values than the rest of the points in two of the predictors. This showed that SIR is influenced by the differences in those two predictors. Compared to Figure 5, all the data points in Figure 6 were relatively close to each other, and a pattern could be seen, which meant the group identification based on the first two directions of CUME was fairly reliable. Figure 7 also provided a relatively clear group pattern without any “outliers”. If we look at the groups, group 1 was the cytoplasm and it formed one cluster, group 2 and 5 were from inner membrane and they formed another cluster, group 6 and 7 were from outer membrane and they formed a third cluster, and finally group 8 was from perisplasm and it formed a fourth cluster. Since group 3 and 4 had only 2 points in the sample respectively, we ignored their roles here.

6. Discussion

In this work we propose the two algorithms for sufficient dimension reduction, one is based on SIR and the second is based on SAVE. Through the simulations we performed, we can see that the methodology proposed called LVRSIR and LVRSAVE perform better than existing methodologies (SIR, CUME, SAVE and CUVE). SIR performs worse when number of slices increases, but LVR’s performance improves (see Table 2). CUME’s performance does not depend on number of slices and it has fairly good averaged performance. When dimension increases each of the methods performs worse, which is as expected (see Table 3). With increasing in sigma their performances drop, but are not as rapidly as they drop with higher dimensions.

In SAVE, several different weighting methods were tested in an effort to maximize the performance of LVR. LVR 3 is found to be the best among all LVRs, and it performs better than SAVE for all models. Unlike in SIR, LVR does not perform better with larger number of slices, although its performance does not drop as quickly as of SAVE (see Table 7). Comparing to CUVE, LVR 3 is not necessarily superior, as it only performs better than CUVE in some cases. CUVE and LVR 3 have their advantage over SAVE when dimension is high, in other words, they perform more consistently with increasing in dimension (see Table 8). More thorough analysis is needed for this; one can extend this job in several directions. An immediate direction is a similar algorithm modification of the DR algorithm to achieve dimension reduction. One can try functions of moments i.e. OVA to do this.

We have tried 6 different weighting methods for OVA in a hope to improve its performance (see Table 5). From the simulation, it seems OVA with new weighting methods do not necessarily perform better than the original OVA. For example, original OVA performs slightly better in the first model; OVA 4 and OVA 6 perform better in model 2, 4 and model 3, 5 respectively.

References

Cook, R. D. (1998), "Regression Graphics: Ideas for Studying Regression Through Graphics", New York, Wiley and Sons

Cook, D. R. and Weisberg, S. (1991), "Sliced Inverse Regression for Dimension Reduction: Comment", Journal of the American Statistical Association, Vol. 86, No. 414, pp. 328-332

Horton, P. and Nakai, K. (1996), "A Probabilistic Classification System for Predicting the Cellular Localization Sites of Proteins", Intelligent Systems in Molecular Biology, pp. 109-115

Li, B., Artemiou, A. and Li, L. (2011), "Principal Support Vector Modeling for Linear and Nonlinear Dimension Reduction", Annals of Statistics, Vol. 39, No. 6, pp. 3182-3210

Li, B. and Wang, S. (2007) "On Directional Regression for Dimension Reduction", Journal of the American Statistical Association, Vol. 102, No. 479, pp. 997-1008

Li, K. C. (1991), "Sliced Inverse Regression", Journal of the American Statistical Association, Vol. 86, No. 414, pp. 316-327

Zhu, L. P., Zhu, L. X. and Feng, Z. H. (2010), "Dimension Reduction in Regressions Through Cumulative Slicing Estimation", Journal of the American Statistical Association December 2010, Vol. 105, No. 492, pp. 1455-1466