EMPIRICAL ANALYSIS OF SPARE PRINCIPAL COMPONENT ANALYSIS

By

Damian Z. Mastylo

A Thesis Submitted to the Graduate

Faculty of Rensselaer Polytechnic Institute

in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

Major Subject: COMPUTER SCIENCE

Approved by the
Examining Committee:

_____
Malik Magdon-Ismail, Thesis Adviser

_____
Elliot Anshelevich, Member

_____
Stacy Patterson, Member

Rensselaer Polytechnic Institute
Troy, New York

April 2016
(For Graduation May 2016)

CONTENTS

LIST OF TABLES iv

LIST OF FIGURES v

ACKNOWLEDGMENT vii

ABSTRACT viii

1. INTRODUCTION 1

1.1 Principal Component Analysis 1

1.2 Sparse Principal Component Analysis 1

1.3 Problem Formulation 2

1.4 Contributions 3

2. Methods 4

2.1 Truncated Power Method 4

2.2 Generalized Power Method 5

2.3 SpaSM 6

2.4 Optimal Sparse Linear Auto-Encoder 7

2.4.1 Column Subset Selection Problem 9

2.4.1.1 Norm Selection 9

2.4.1.2 Right Singular Vector Selection 10

2.4.1.3 Near-Optimal Column-Based Matrix Reconstruction 10

3. Data Sets 12

3.1 Overview 12

3.2 ARCENE 12

3.3 Chronic Kidney Disease 13

3.4 Gisette 13

3.5 Madelon 13

3.6 Ones and Fives 14

3.7 Ozone 14

3.8 Ozone One Hour 14

3.9 Pima Indians Diabetes 14

3.10 SECOM . . . 15
3.11 Spam . . . 15
4. Results . . . 16
4.1 Measuring Performance . . . 16
4.1.1 Information Loss . . . 16
4.1.2 Variance . . . 16
4.1.3 Classification Accuracy . . . 17
4.2 ARCENE . . . 17
4.3 Chronic Kidney Disease . . . 20
4.4 Gisette . . . 23
4.5 Madelon . . . 26
4.6 Ones and Fives . . . 29
4.7 Ozone . . . 32
4.8 Ozone One Hour . . . 34
4.9 Pima Indians Diabetes . . . 36
4.10 SECOM . . . 38
4.11 Spam . . . 41
5. Discussion and Conclusion . . . 44
5.1 CONCLUSION . . . 44
References . . . 48
APPENDICES
A. All Results . . . 49
A.1 ARCENE Results . . . 49
A.2 Chronic Kidney Disease Results . . . 59
A.3 Gisette Results . . . 68
A.4 Madelon Results . . . 77
A.5 Ones and Fives Results . . . 86
A.6 Ozone Results . . . 95
A.7 Ozone One Hour Results . . . 104
A.8 Pima Indians Diabetes Results . . . 113
A.9 SECOM Results . . . 122
A.10 Spam Results . . . 131

LIST OF TABLES

3.1 Data Sets Overview 12

5.1 Classification Accuracy Summary 46

5.2 Information Loss Summary 46

5.3 Symmetric Variance Summary 46

LIST OF FIGURES

4.1 ARCENE Classification Accuracy at k = 3 18

4.2 ARCENE Information Loss at k = 3 19

4.3 ARCENE Symmetric Variance at k = 3 19

4.4 Chronic Kidney Disease Classification Accuracy at k = 3 21

4.5 Chronic Kidney Disease Information Loss at k = 3 22

4.6 Chronic Kidney Disease Symmetric Variance at k = 3 22

4.7 Gisette Classification Accuracy at k = 3 24

4.8 Gisette Information Loss at k = 3 25

4.9 Gisette Symmetric Variance at k = 3 25

4.10 Madelon Classification Accuracy at k = 3 27

4.11 Madelon Information Loss at k = 3 28

4.12 Madelon Symmetric Variance at k = 3 28

4.13 Ones and Fives Classification Accuracy at k = 3 30

4.14 Ones and Fives Information Loss at k = 3 31

4.15 Ones and Fives Symmetric Variance at k = 3 31

4.16 Ozone Classification Accuracy at k = 3 32

4.17 Ozone Information Loss at k = 3 33

4.18 Ozone Symmetric Variance at k = 3 33

4.19 Ozone One Hour Classification Accuracy at k = 3 34

4.20 Ozone One Hour Information Loss at k = 3 35

4.21 Ozone One Hour Symmetric Variance at k = 3 35

4.22 Pima Indians Diabetes Classification Accuracy at k = 3 36

4.23 Pima Indians Diabetes Information Loss at k = 3 37

4.24 Pima Indians Diabetes Symmetric Variance at k = 3 37

4.25 SECOM Classification Accuracy at k = 3 39
4.26 SECOM Information Loss at k = 3 40
4.27 SECOM Symmetric Variance at k = 3 40
4.28 Spam Classification Accuracy at k = 3 42
4.29 Spam Information Loss at k = 3 43
4.30 Spam Symmetric Variance at k = 3 43

ACKNOWLEDGMENT

I would like to thank my adviser Malik Magdon-Ismail for providing me with an interesting project and working with me for a year to complete this thesis. I would also like to thank my parents for their support my whole life.

ABSTRACT

Sparse Principal Component Analysis (SPCA) builds upon regular Principal Component Analysis (PCA) by including a sparsity factor to further reduce the number of dimensions. The goal of this thesis is to demonstrate the benefits of using a SPCA method that focuses on minimizing information loss as opposed to maximizing variance. Current state-of-the-art SPCA methods include TPower, GPower, and Zou's SPCA as implemented in the SpaSM toolkit. These current methods focus on maximizing variance. We hypothesize that the other approach, minimizing information loss, may yield better results in machine learning. We employ a practical approach to examine this problem.

Many optimizations exist via tweaking the sparsity factor, the number of left singular vectors used, or the column subset selection method. Many combinations of these approaches are examined, and their efficacy are reported by comparing information loss, symmetric variance, and the classification accuracy of a Support Vector Machine (SVM) using the transformed data set.

1. INTRODUCTION

Dimensionality reduction is a useful tool in machine learning and statistics. Data sets with a large number of features may be unwieldy and difficult to work with, some features may in fact be useless to the learner, and the curse of dimensionality comes into play. Primary advantages of dimension reduction include the reduction of space and time needed to learn from the data set, as well as potentially improving the performance of the model by removing unnecessary features. This thesis will explore Sparse Principal Component Analysis (SPCA), a special subset of Principal Component Analysis (PCA).

1.1 Principal Component Analysis

PCA is a widely used statistical method for identifying patterns in data sets with high dimensionality. This is done by transforming the original features into principal components, "which are uncorrelated, and which are ordered so that the first few retain most of the variation present in all of the original variables" [1]. With this in mind, dimensionality reduction is possible by simply curtailing the number of components chosen; components with the largest variance are chosen first. PCA is optimal in information loss, and stands as the most popular linear auto-encoder.

1.2 Sparse Principal Component Analysis

PCA is disadvantaged by its maintainence of all the information: some of it may be difficult to interpret, or be irrelevant to a machine learning algorithm [2]. There may be specific features that have greater significance to the data set, and other features may be of little use. Essentially, it is difficult to interpret the principal components since they are comprised of all the features of the original data set. Due to this, SPCA methods have been implemented over the years to perform the analysis using a smaller subset of original features. When introducing sparsity, principal components become much easier to interpret. It is important to understand that by introducing sparsity, we are create a sparse *encoder*, not necessarily a sparse data

set. This is a common confusion; the encoder may be sparse and have only a few nonzero elements, but this does not necessary translate to a sparse data set after the data set has been encoded.

Current state of the art methods may not have the best performance when the encoder transforms the data set and is fed into a machine learning algorithm. "The traditional formulation of sparse PCA is as a cardinality constrained variance maximization problem..." [2]. We aim to challenge the notion of variance maximization being the best solution to the SPCA problem. Instead of being constrained by symmetric variance maximization, the Optimal Sparse Linear Auto-Encoder seeks to minimize information loss while reducing dimensionality. We argue that variance has little intrinsic value, and in fact, removes more data than necessary and creates a worse-performing encoder [2].

We are seeking to test our information loss minimizing method against multiple variance maximizing SPCA methods to challenge our hypothesis that information loss minimization is much more useful in machine learning scenarios. There are two information loss maximizing methods tested: one is a batch algorithm, the other is an iterative algorithm utilizing the original batch algorithm. In our testing the iterative algorithm almost exclusively performs better than the batch algorithm, therefore the batch algorithm results are not included in our analysis.

1.3 Problem Formulation

When using regular PCA, the problem of maximizing symmetric explained variance and minimizing information is equivalent. For a data matrix $X \in \mathbb{R}^{n \times d}$ and encoder matrix $H \in \mathbb{R}^{d \times k}$

$$\max \operatorname{Tr}(H^{\dagger} X^{T} X H) \equiv \min ||X - XHH^{\dagger}||_{F}^{2} \text{ [2]}$$

When sparsity is introduced to PCA, either maximizing variance or minimizing information loss becomes the goal of an SPCA method. Traditional methods have opted to maximize variance, but this can cause too much loss of information; which directly affects the performance of unsupervised machine learning methods.

1.4 Contributions

This research provides empirical data for the performance and efficiency of multiple SPCA algorithms via analysis of information loss, symmetric variance, and the classification accuracy of the model created by an SVM. Data sets of different compositions are tested in an attempt to establish trends in the performance of the algorithms. Specifically, we attempted to find trends or differences in performances in data sets with high dimensionality, low dimensionality, sparse data sets, dense data sets, and data sets with a variable number of instances. The type of data set was a potential consideration as well; a data set that contained vision data may have performed differently than a data set containing data on diseases. Most importantly, we are interested in examining the relationship between the minimization of information loss and the performance of a model created by an SVM to determine if it has more value than an SPCA method that attempts to maximize variance.

2. Methods

The methods outlined below create encoders that can be used to modify the original data set. The encoders are controlled by k and r variables which control the number of components extracted by the encoder, and the number of nonzero elements per component extracted, respectively. For example, an encoder created with $k = 2$ and $r = 4$ will create an encoder with 2 columns, with each column containing at most 4 nonzero elements. Most methods do not guarantee exactly r nonzero elements, and in some cases, there may be less than r nonzero elements.

2.1 Truncated Power Method

The Truncated Power Method (TPower) computes an approximate solution to the sparse eigenvalue problem, "which is to extract dominant (largest) sparse eigenvectors with at most k non-zero components" [3]. It is similar to the classical power method for calculating eigenvalues, but adds an extra step that provides for the sparsity. Recall that an eigenvector is a vector C that is non-zero and has a number λ that fulfills $XC = \lambda C$. If λ exists, it is an eigenvalue of X.

Algorithm 1: TPower Algorithm [3]

Input : matrix $A \in \mathbb{S}^p$, initial vector $x_0 \in \mathbb{R}^p$

Output : x_t

Parameters: cardinality $k \in \{1, ..., p\}$

Let $t = 1$.

repeat

- Compute $x'_t = Ax_{t-1}/\|Ax_{t-1}\|$
- Let $F_t = \text{supp}(x'_t, k)$ be the indices of x'_t with the largest k absolute values.
- Compute $\hat{x}_t = \text{Truncate}(x'_t, F_t)$.
- Normalize $x_t = \hat{x}_t/\|\hat{x}_t\|$.

until *Convergence*;

The MATLAB implementation of this algorithm takes a cardinality vector as input, e.g. $[100, 100, 100]$, with the number of elements representing k and the element values themselves representing r. This is similar to how the iterative Optimal Sparse Linear Auto-Encoder works, however in our tests we kept the r consistent across all elements of the vector to make the results easily comparable.

2.2 Generalized Power Method

The Generalized Power Method (GPower) employs an iterative gradient optimization method to calculate sparse components. Four versions are provided by the authors with differing penalties, and execution methods. Single-unit and block methods are provided, but only the single-unit methods are employed in our testing.

Algorithm 2: GPower Algorithm using the l_1-penalty [4]

Input : matrix $A \in R^{p \times n}$

Sparsity-controlling parameter $\gamma \geq 0$

Initial iterate $x \in S^p$

Output : A locally optimal sparsity pattern P

repeat

$x \longleftarrow \sum_{i=1}^{n} [|a_i^T x| - \gamma] + \text{sign}(a_i^T x) a_i$

$x \longleftarrow \frac{x}{\|x\|}$

until *a stopping criterion is satisfied*;

Construct a vector $P \in \{0, 1\}^n$ such that $\begin{cases} p_i = 1 \text{ if } |a_i^T x| > \gamma \\ p_i = 0 \text{ otherwise} \end{cases}$

Algorithm 3: GPower Algorithm using the l_0-penalty [4]

Input : matrix $A \in R^{p \times n}$

Sparsity-controlling parameter $\gamma \geq 0$

Initial iterate $x \in S^p$

Output : A locally optimal sparsity pattern P

repeat

$x \longleftarrow \sum_{i=1}^{n} [\text{sign}((a_i^T x)^2 - \gamma)] + a_i^T x a_i$

$x \longleftarrow \frac{x}{\|x\|}$

until *a stopping criterion is satisfied*;

Construct a vector $P \in \{0,1\}^n$ such that $\begin{cases} p_i = 1 \text{ if } (a_i^T x)^2 > \gamma \\ p_i = 0 \text{ otherwise} \end{cases}$

Unfortunately, GPower does not allow for an explicit sparsity input; it uses a sparsity factor γ which can be a number in the range $(0, 1)$. "... the resulting number of nonzero elements can not be precisely predicted" [4]. We attempted to match the factor to r to give the desired number of components, however we were unable to find a pattern that worked for automated testing as the factor did not seem to respond to a basic linear model. Therefore, we do not guarantee r was exactly reached in our results.

2.3 SpaSM

SpaSM revolves around solving a "lasso" or "elastic net" which are modified least squares methods, with constraints on regression coefficients [5]. The soft-thresholding on the elastic net, in particular, "... leads to a computational advantage, which is why this option is popular for very high-dimensional data arising from e.g., image or gene expression data. It is our experience that this option also provides better solutions (in terms of explained variance for a fixed level of sparsity) in such cases" [6]. The important take-away is SpaSM is essentially a supervised learner as it is performing regularized linear regression on the data set, so we expect SpaSM to perform very well.

Algorithm 4: SpaSM General SPCA Algorithm [6]

Input : matrix $A \in R^{p \times n}$

Sparsity-controlling parameter $\gamma \geq 0$

Initial iterate $x \in S^p$

Output : A locally optimal sparsity pattern P

Let A start at $V[, 1:k]$, the loadings of the first k ordinary principal components.

Given a fixed $A = [\alpha_1, ..., \alpha_k]$, solve the following elastic net problem for $j = 1, 2, ..., k$

$$\beta_j = \arg_\beta \min(\alpha_j - \beta)^T X^T X(\alpha_j - \beta) + \lambda||\beta||^2 + \lambda_{1,j}||\beta||_1$$

For a fixed $B = [\beta_1, ..., \beta_k]$, compute the SVD of $X^T XB = UDV^T$, then update $A = UV^T$.

Repeat Steps 2-3, until convergence.

Normalization: $\hat{V}_j = \frac{\beta_j}{||\beta_j||}$, $j = 1, ..., k$.

2.4 Optimal Sparse Linear Auto-Encoder

The Optimal Sparse Linear Auto-Encoder differs from the current state of the art methods due to its focus on minimization of information loss, rather than the maximization of variance. There is a batch and iterative version of this algorithm, but in our testing we solely used the iterative version due to its better performance across the board. The iterative encoder is comprised of three main steps: a blackbox algorithm to compute column subset selection, the computation of the encoder, and the iterative step which computes an encoder per component.

Algorithm 5: Blackbox algorithm to compute encoder from CSSP (Column Subset Selection Problem) [2]

Input : $X \in \mathbb{R}^{n \times d}; C \in \mathbb{R}^{n \times r}$ with $C = X\Omega$ and $\Omega = [e_{i1}, ..., e_{ir}]; k \leq r$.

Output: r-sparse linear encoder $H \in \mathbb{R}^{d \times k}$.

Compute a QR-factorization of C as $C = QR$, with $Q \in \mathbb{R}^{n \times r}$, $R \in \mathbb{R}^{r \times r}$.

Compute the SVD of $R^{-1}(Q^T R)_k, R^{-1}(Q^T X)_k = U_R \Sigma_R V_R^T$, where

$$U_R \in \mathbb{R}^{r \times k}, \Sigma_R \in \mathbb{R}^{k \times k} \text{ and } V_R \in \mathbb{R}^{d \times k}.$$

Return $H = \Omega U_R \in \mathbb{R}^{d \times k}$.

This blackbox algorithm requires a column subset selection algorithm to create the input variable C, which is covered later.

Algorithm 6: Sparse Linear Encoder Algorithm [2]

Input : $X \in \mathbb{R}^{n \times d}$; target rank $k \leq rank(X)$; sparsity $r > k$

Output: Near-optimal r-sparse linear encoder $H \in \mathbb{R}^{d \times k}$.

Use a column selection algorithm to compute $C = X\Omega \in \mathbb{R}^{n \times r}$, with inputs X, k, r.

Return the encoder H computed by using X, C, k as input to the CSSP-blackbox encoder algorithm.

The iterative algorithm computes each component separately rather than all at once.

Algorithm 7: Iterative Sparse Linear Encoder Algorithm [2]

Input : $X \in \mathbb{R}^{n \times d}$; target rank $k \leq rank(X)$; sparsity parameters $r_1, ..., r_k$.

Output: $(r_1, ..., r_k)$-sparse linear encoder

Set the residual $\Delta = X$ and $H = []$.

for $i = 1$ *to* k **do**

  Compute an encoder h for Δ using the batch algorithm with $k = 1$ and sparsity parameter $r = r_i$.

  Update the encoder by adding h to it: $H \leftarrow [H, h]$.

  Update the residual $\Delta : \Delta \leftarrow X - XH(XH)^{\dagger}X$.

Return the $(r_1, ..., r_k)$-sparse encoder $H \in \mathbb{R}^{n \times k}$.

2.4.1 Column Subset Selection Problem

The blackbox algorithm for CSSP requires an input of columns "selected" from the original data set. A few methods were implemented to choose the columns from the original data set, but only Near-Optimal Column-Based Matrix Reconstruction was included in the final results due to its performance benefits over the others.

2.4.1.1 Norm Selection

This algorithm involves calculating the norm of each column of the data set and normalizing it by the Frobenius norm of the data set to create a set of probabilities for selecting columns.

Algorithm 8: Norm Selection

Input : $X \in \mathbb{R}^{n \times d}$; sparsity r

Output: $C \in \mathbb{R}^{1 \times r}$

probabilities = [ ]

for $i = 1$ *to* d **do**

  $$\text{probabilities}[i] = \frac{||X_i||^2}{||X||_F^2}$$

Select columns based on the probabilities calculated, with replacement.

2.4.1.2 Right Singular Vector Selection

This algorithm involves calculating the norm of the right singular vectors of the original data set and normalizing it by k to create a set of probabilities for selecting columns.

Algorithm 9: Right Singular Vector Selection

Input : $X \in \mathbb{R}^{n \times d}$; target rank $k \leq rank(X)$;sparsity r

Output: $C \in \mathbb{R}^{1 \times r}$

Compute the SVD of $X, X = U_X \Sigma_X V_X^T$, where

$$U_X \in \mathbb{R}^{n \times k},\ \Sigma_X \in \mathbb{R}^{k \times k} \text{ and } V_X \in \mathbb{R}^{d \times k}.$$

probabilities = [ ]

for $i = 1$ *to* d **do**

  probabilities$[i] = \dfrac{||V_i||^2}{k}$

Select columns based on the probabilities calculated, with replacement.

2.4.1.3 Near-Optimal Column-Based Matrix Reconstruction

This is a far more advanced algorithm for column subset selection which is used in our final experiments. It employs "fast approximate SVD-like decompositions for column-based matrix reconstruction" [7] to create a polynomial time algorithm to select a set of columns which is almost as good as X_k at reconstructing X. [7]. Recall that X_k represents the rank-k approximation of X.

Algorithm 10: Deterministic Dual Set Spectral Sparsification

Input : $\nu = v_1, ..., v_n, \text{with} \sum_{i=1}^{n} v_i v_i^T = I_k (k < n)$

$\mu = u_1, ..., u_n, \text{with} \sum_{i=1}^{n} u_i u_i^T = I_l (l \leq n)$ integer r, with

$k < r < n$

Output: A vector of weights $s = [s_1, ..., s_n]$, with $s_i \geq 0$ and at most r non-zero s_i's.

Initialize $s_0 = 0_{n\times 1}, A_0 = 0_{k\times k}, B_0 = 0_{l\times l}$.

for $r = 0, ..., r-1$ **do**

Compute L_r and U_r from Eqn.(7.4).

Find an index j in $1, ..., n$ such that

$$U(u_j, \delta_U, B_r, U_r) \leq L(v_j, \delta_L, A_r, L_r).$$

Let

$$t^{-1} = \tfrac{1}{2}(U(u_j, \delta_U, B_r, u_r) + L(v_j, \delta_L, A_r, L_r)).$$

Update the jth component of s, A_r and B_r:

$$s_{r+1}[j] = s_r[j] + t, A_{r+1} = A_r + t v_j v_j^T, \text{ and } B_{r+1} = B_r + t u_j u_j^T.$$

Return $s = r^{-1}(1 - \sqrt{k/r}) \cdot s_r$.

3. Data Sets

Our experiment setup required two-class-classification data sets with strictly numerical data, as MATLAB's Support Vector Machine (SVM) implementation only supported this type of learning. Data sets were chosen such that they fit the criteria of the experiment setup, but also provided variety in the number of data points, features, and density of the data so we could examine how the different encoders performed on varied data sets. Most data sets came from UCI's Machine Learning Repository [8].

3.1 Overview

A table below outlines some statistics of the data sets. Rank is defined as the maximum number of linearly independent rows or columns of a matrix. Numerical Rank is defined as $\frac{||X||_F^2}{||X||^2}$.

Table 3.1: Data Sets Overview

Data Sets	Data Points	Dimensions	Rank	Numerical Rank	SVM Perf
ARCENE	100	10000	100	1.4249	$\sim 56\%$
CKD	400	24	24	1.0005	$\sim 95\%$
Gisette	500	5000	500	3.3208	$\sim 56\%$
Madelon	2000	500	500	1.0038	$\sim 50\%$
OF	1561	256	256	1.4128	$\sim 64\%$
Ozone	2534	72	72	1.0014	$\sim 94\%$
OOH	500	72	72	1.0041	$\sim 96\%$
PID	768	8	8	1.2622	$\sim 75\%$
SECOM	1567	589	471	1.3663	$\sim 93\%$
Spam	4601	56	56	1.0646	$\sim 83\%$

3.2 ARCENE

ARCENE attempts to distinguish cancer from normal patterns in mass-spectrometric data. It is a data set that was used in the NIPS 2003 feature selection challenge [9] and contains samples of patients with prostate or ovarian cancer, or control patients

with no cancer, giving a high-dimension 100×10000 data set. This data set includes 3000 "probes" which are features that do not match the underlying model and are supposed to act as distractors to the learner. The SVM is unable to create an accurate model with the untouched data set, giving 56% accuracy on classification.

3.3 Chronic Kidney Disease

The Chronic Kidney Disease (CKD) data set has a variety of medical measurements that attempt to model the possibility of an individual having CKD, giving a low-dimension data set: 400×24. CKD is a gradual and slow degradation of the function of the kidney to process and filter wastes. The data was collected by doctors and researchers working in a hospital in India. Running the SVM on this data set produced a very accurate model that correctly predicted the presence of CKD $\sim 95\%$ of the time.

3.4 Gisette

Gisette is comprised of handwritten fours and nines, which are confused often. Similarly to the ARCENE data set, Gisette was used in the NIPS 2003 feature selection challenge [9] and the data set includes 2500 "probes", giving a high-dimension 500×5000 data set. The SVM is unable to create an accurate model without encoding, giving $\sim 56\%$ accuracy on classification.

3.5 Madelon

MADELON is an artificial data set which was made to be multivariate and non-linear, which makes for a very difficult learning problem. Data points are grouped in 32 clusters on vertices of a five dimensional hypercube, with random labeling. There are 20 real features, and 480 "probe" features, giving a 2000×500 data set. As with ARCENE and Gisette, this data set is from the NIPS 2003 feature selection challenge [9]. As expected, the SVM does poorly, creating a model that is no better than guessing randomly, with a $\sim 50\%$ accuracy on classification.

3.6 Ones and Fives

This data set is adapted from the classic handwritten zip codes digit data. It only contains handwritten black-and-white ones and fives, to make it a classification problem with two classes. Each image is 256 pixels, and the value of the intensity of each black pixel is a "feature" for each data point, giving a 1561×256 data set. This data set is ripe for dimension reduction; many pixels are empty, and the nature of the data causes the SVM to fail at making an accurate model for the data without post-processing the data via feature extraction, or other methods, such as the encoders presented.

3.7 Ozone

The Ozone data set is a 2534×72 data set comprised of various measurements to identify if a particular day is an Ozone peak day or a normal day. The data is based on an eight hour peak and was collected from 1998 to 2004 in Houston, Galveston, and Brazoria. The base SVM performs very well, giving $\sim 94\%$ accuracy on classification.

3.8 Ozone One Hour

Ozone One Hour is very similar to the original Ozone data set, but its data pertains to an one hour peak, rather than an eight hour peak. The base SVM performs even better than the original data set, giving $\sim 96\%$ accuracy on classification.

3.9 Pima Indians Diabetes

This data set attempts to model the presence of diabetes in Pima Indian females at least 21 years old. Numerous medical measurements are provided, giving a low-dimensionality 768×8 data set. Running the SVM on this data set produced a model that correctly predicted the presence of diabetes $\sim 75\%$ of the time.

3.10 SECOM

SECOM stands for Semi-Conductor Manufacturing. During manufacturing, many variables are recorded from sensors to monitor the manufacturing process. There are useful features, and some features that do not add to the predictive power of the model, giving a 1567×591 data set. Missing values represented as a question mark (?) in the data set were replaced with zeros as MATLAB's SVM is unable to handle missing values. The base SVM performs well, giving $\sim 93\%$ accuracy on classification.

3.11 Spam

This data set consists of features from analysis of spam emails to extract features that were commonly found, such as frequency of specific words, sequences of capital letters, and others. The emails marked as spam came from a postmaster, or manual marking of spam by individuals. This 4601×57 data set achieves $\sim 83\%$ accuracy on classification when fed into the SVM.

4. Results

Experiments were run for each data matrix $X \in \mathbb{R}^{n \times d}$. Let us recall that k represents the number of features extracted, and r represents the number of non-zero elements in each feature. We keep k fixed at a specific value and increment r from 1 to d or 25, whichever was smaller. The expensive nature of these tests made it prohibitive to experiment with incrementing r to very large values on high-dimensional data sets. The encoder was used to encode X, and fed into an SVM; its prediction rate was recorded, along with the information loss and symmetric variance of the encoder, and repeated 3 times to get an average result per encoder. k was then incremented and the process restarted. For brevity, we only include the results from the $k = 3$ tests, the full results can be found in Appendix A.

4.1 Measuring Performance

4.1.1 Information Loss

Information Loss is defined as "the minimum possible reconstruction error for X over all linear decoders G" [2]. The aim is to create an encoder that can get as close as possible to optimal information loss, which is achieved by regular PCA. Creating an optimal r-sparse encoder is an NP-hard problem however, so the competing algorithms are approximation algorithms. The information loss is controlled by a sparsity variable r and a variable k controlling the number of features to include in the encoder. r controls how many original features are used to explain the encoder's features.

$$\text{Information Loss} = \frac{\|X - XH(XH)^{\dagger}X\|_F^2}{\|X - X_k\|_F^2} \geq 1$$

4.1.2 Variance

As [2] states, the current state-of-the-art methods outlined above, TPower, GPower, and SpaSM, create an encoder that has orthonormal columns and minimize the reconstruction error $\hat{x} = \|XHH^{\dagger}\|_F^2$. This is equivalent to maximizing the

explained variance, however we use the normalized explained variance of the symmetric variance to compare results. This is for historical reasons, "existing sparse PCA methods have constructed auto-encoders to optimize the symmetric explained variance rather than the true explained variance" [2].

$$\text{Symmetric Explained Variance} = \frac{\|XHH^{\dagger}\|_F^2}{\|X_k\|_F^2} \leq 1$$

4.1.3 Classification Accuracy

We fed the encoded data sets into a ten-fold cross-validation SVM to create a predictive model for the underlying data set. This "encoded SVM" was compared to results from the "base SVM", as well as the other encoded SVMs.

4.2 ARCENE

This high-dimension data set had some success with the encoders. At $k = 1$ the GPower and TPower encoders produced a model with $\sim 65\%$ accuracy, while SpaSM and the Optimal Sparse Linear Auto-Encoder did very poorly. However, as k was increased, the performance increased for SpaSM and the Optimal Sparse Linear Auto-Encoder. GPower's performance did not increase very much with higher k values. SpaSM and the Optimal Sparse Linear Auto-Encoder do better than GPower occasionally in the testing, but it's not until $k = 6$ that SpaSM's overall performance surpasses GPower's performance. Interestingly, increasing the number of non-zero elements actually decreased the average predictive power of the model for SpaSM and the Optimal Sparse Linear Auto-Encoder in most cases. Overall, the models were not incredibly accurate, but the encoders did allow for creation of models stronger than the base SVM.

TPower ran incredibly slowly on this data set and became prohibitively expensive to run, so it was not included in the $4 \leq k \leq 6$ results.

SpaSM had the best information loss ratio across all tests, with the Optimal Sparse Linear Auto-Encoder following closely in second place. The other encoders had significantly worse information loss ratios.

TPower had the highest symmetric variance when it was included in the tests from $k = 1$ to $k = 3$. SpaSM was in second place for $k = 1$ through $k = 3$ and

had the best symmetric variance at higher k values, since TPower was not tested anymore. The other encoders did rather poorly, and increasing non-zero elements did not improve the variance an appreciable amount. This most likely has to do with the very large dimensionality of this data set.

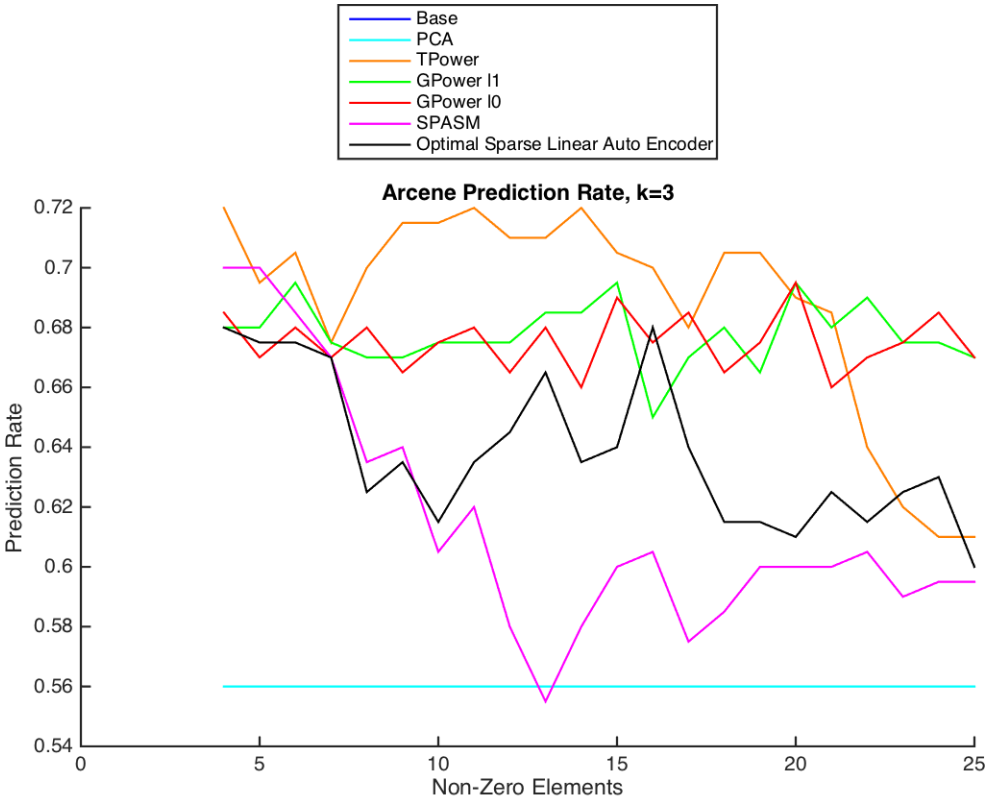


Figure 4.1: ARCENE Classification Accuracy at k = 3

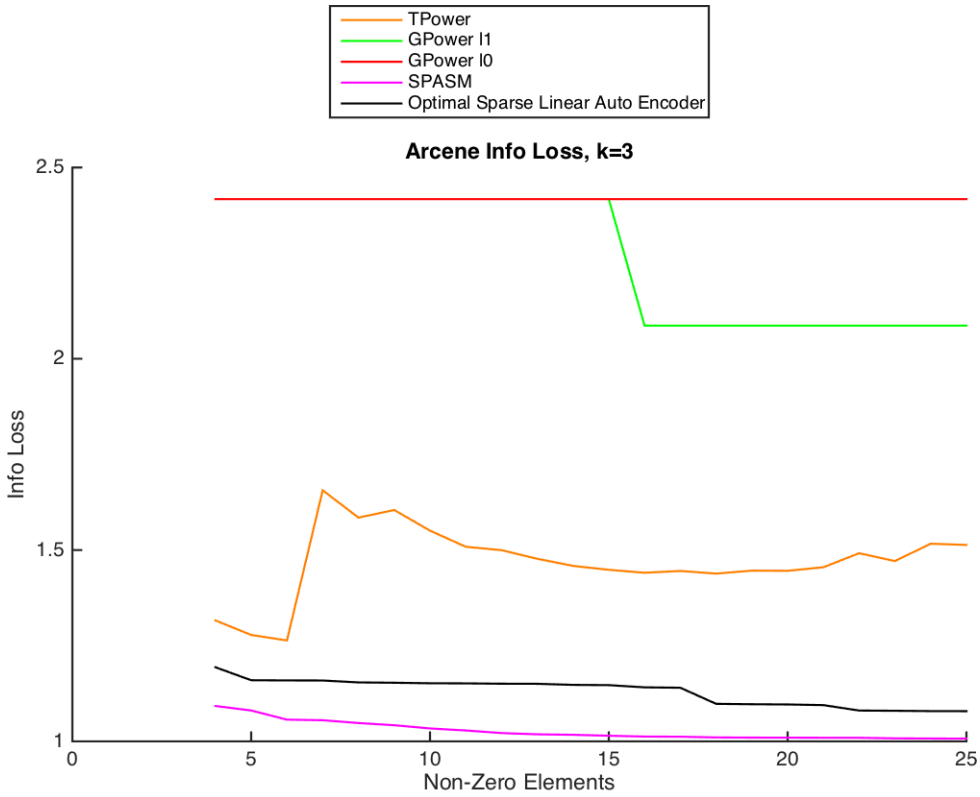


Figure 4.2: ARCENE Information Loss at k = 3

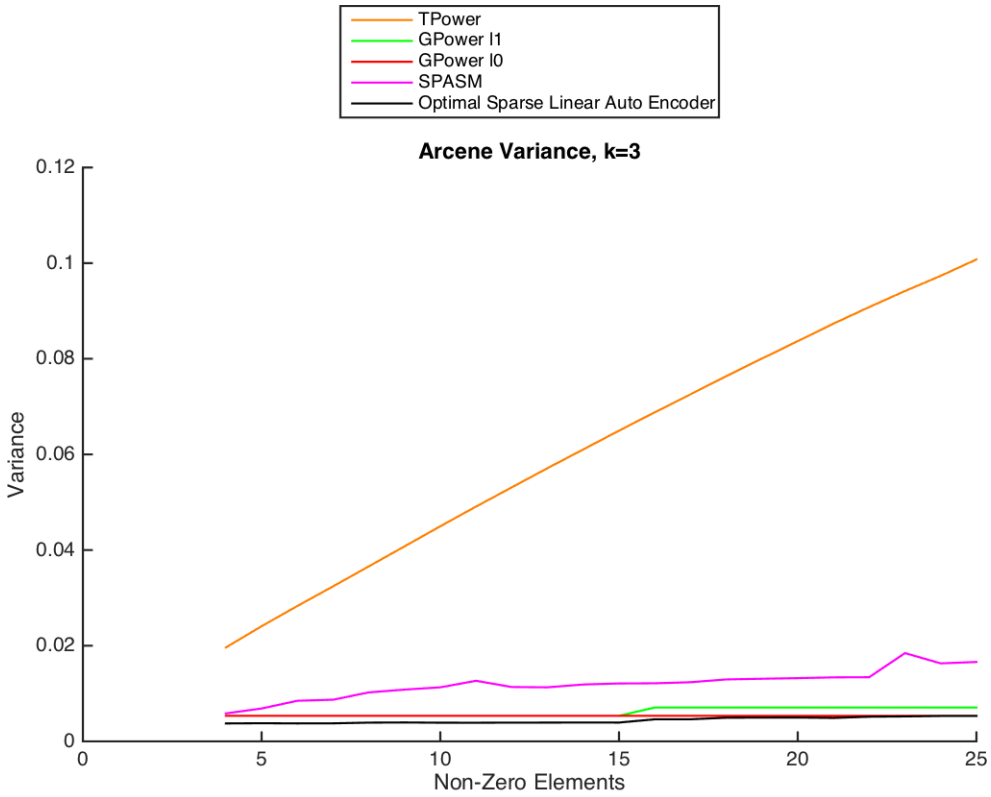


Figure 4.3: ARCENE Symmetric Variance at k = 3

4.3 Chronic Kidney Disease

This data set had great performance in the SVM when unmodified, and worse performance when encoded. Performance was highly correlated with higher k values: at $k = 5$ and $k = 6$ the GPower encoder's performance got close to the base SVM. The Optimal Sparse Linear Auto-Encoder and SpaSM were typically the worst performing encoders tested; there is an exception at $k = 2$ where the Optimal Sparse Linear Auto-Encoder performed better than the rest of the encoders, but by a very small margin. Increasing the number of non-zero elements had little positive impact on the average performance of the encoder.

The Optimal Sparse Linear Auto-Encoder had the best information loss ratio in almost all tests. SpaSM and TPower were typically very close if not equal to the reported information loss ratio after the number of non-zero elements was increased to 7 and higher. GPower overall did poorly compared to the other encoders.

The symmetric variance ratio across all encoders was within ten-thousandths away from 1.0, so all the encoders performed well. TPower overall did the best, but by an extremely small margin. SpaSM and the Optimal Sparse Linear Auto-Encoder were practically equal among all the tests. GPower using the l_1 penalty did better on average than GPower using the l_0 penalty.

Interestingly, GPower did not excel at either the information loss ratio, or the symmetric variance, yet its SVM performance was the best across the encoders.

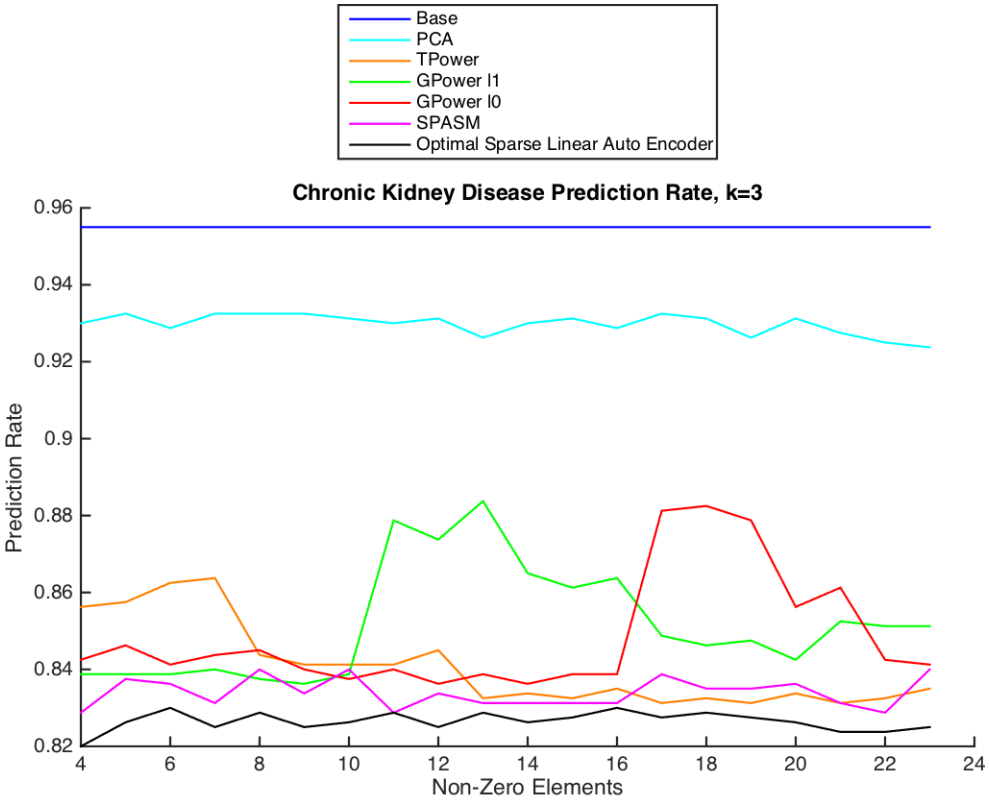


Figure 4.4: Chronic Kidney Disease Classification Accuracy at k = 3

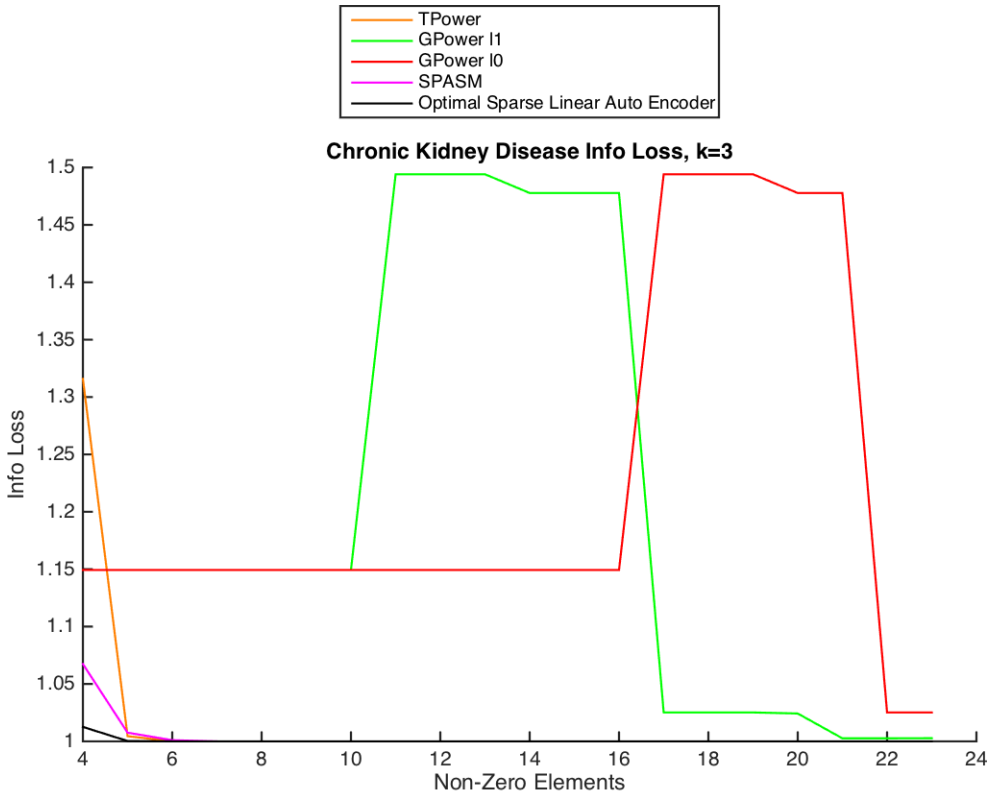


Figure 4.5: Chronic Kidney Disease Information Loss at k = 3

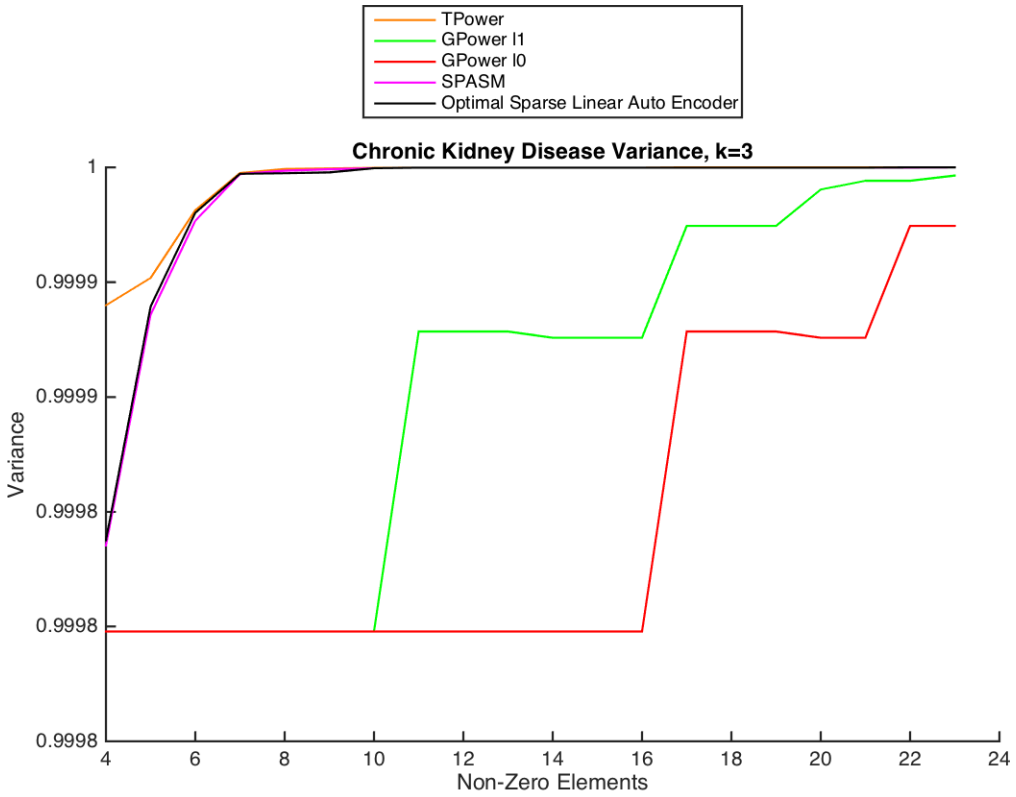


Figure 4.6: Chronic Kidney Disease Symmetric Variance at k = 3

4.4 Gisette

Gisette is a high-dimension data set that benefited from using encoders. The SVM was unable to create an accurate model from the unmodified data set. At $k = 1$ there was very little improvement from using encoders, in fact, most of the encoders created models that did worse than a strategy of guessing randomly. However, as k increased, the performance of SpaSM, Optimal Sparse Linear Auto-Encoder, and to a lesser extent, TPower, improved. SpaSM had a huge performance boost at $k >= 4$, and saw modest gains in performance when increasing the number of non-zero elements. The Optimal Sparse Linear Auto-Encoder, however, had fairly bipolar performance with different r values; at some points its performance surpassed SpaSM, only for its performance to drop over 15% after incrementing r by one. SpaSM did not exhibit this behavior; local minimas were typically very close to local maximas. GPower remained a poor encoder throughout the entire experiment. Overall, SpaSM with $k \geq 4$ was the best performing encoder, and it was able to create a fairly accurate model when fed into an SVM.

Overall, SpaSM had the best information loss ratio across most of the tests. The Optimal Sparse Linear Auto-Encoder had a better ratio at $k = 1$ but was surpassed by SpaSM in the following tests. TPower did slightly worse than the Optimal Sparse Linear Auto-Encoder almost universally, except at $k = 6$ and $r \geq 23$.

TPower had the best symmetric variance across the board, with SpaSM, Optimal Sparse Linear Auto-Encoder, and GPower ranked worse, in that order.

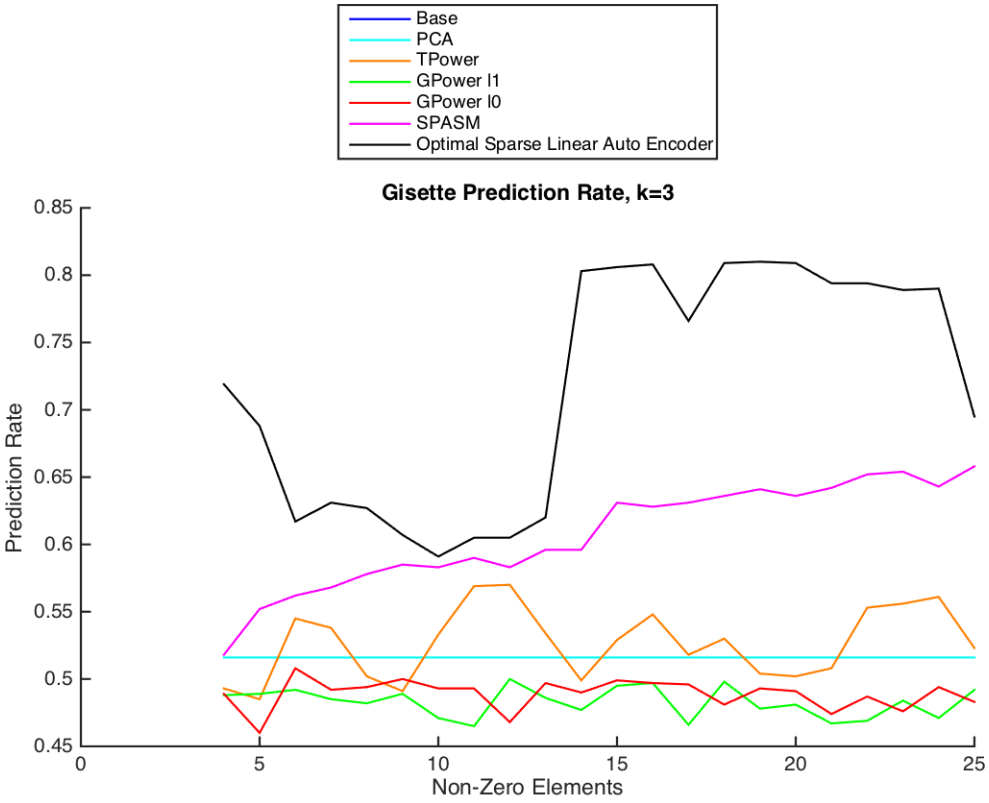


Figure 4.7: Gisette Classification Accuracy at k = 3

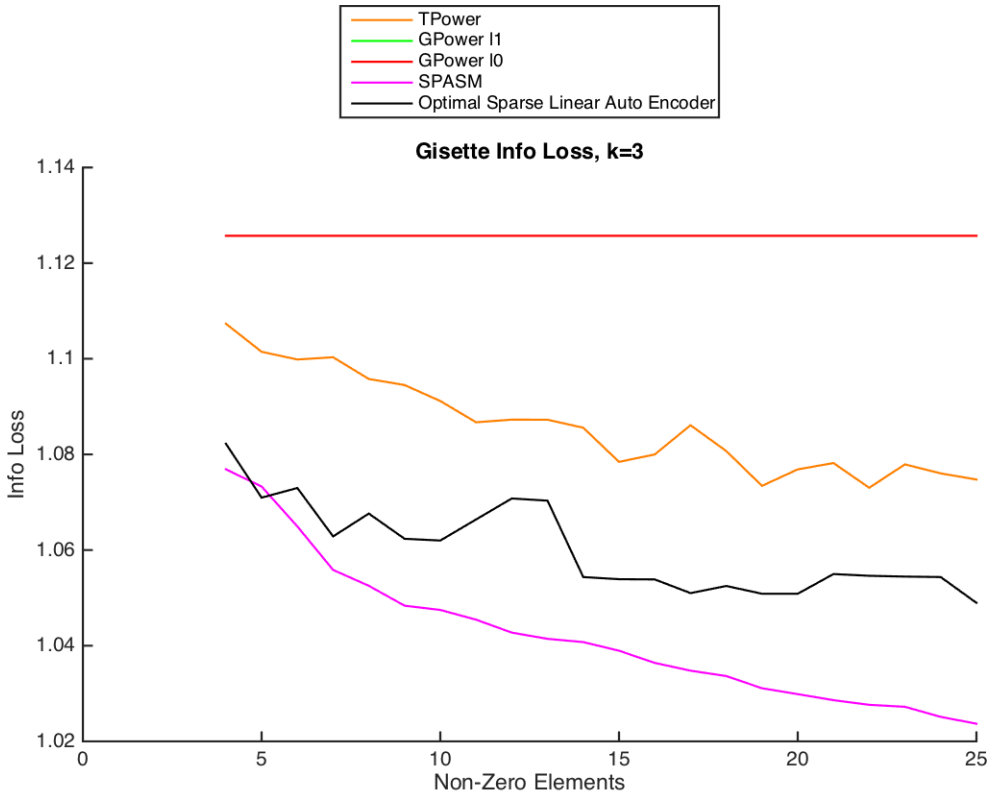


Figure 4.8: Gisette Information Loss at k = 3

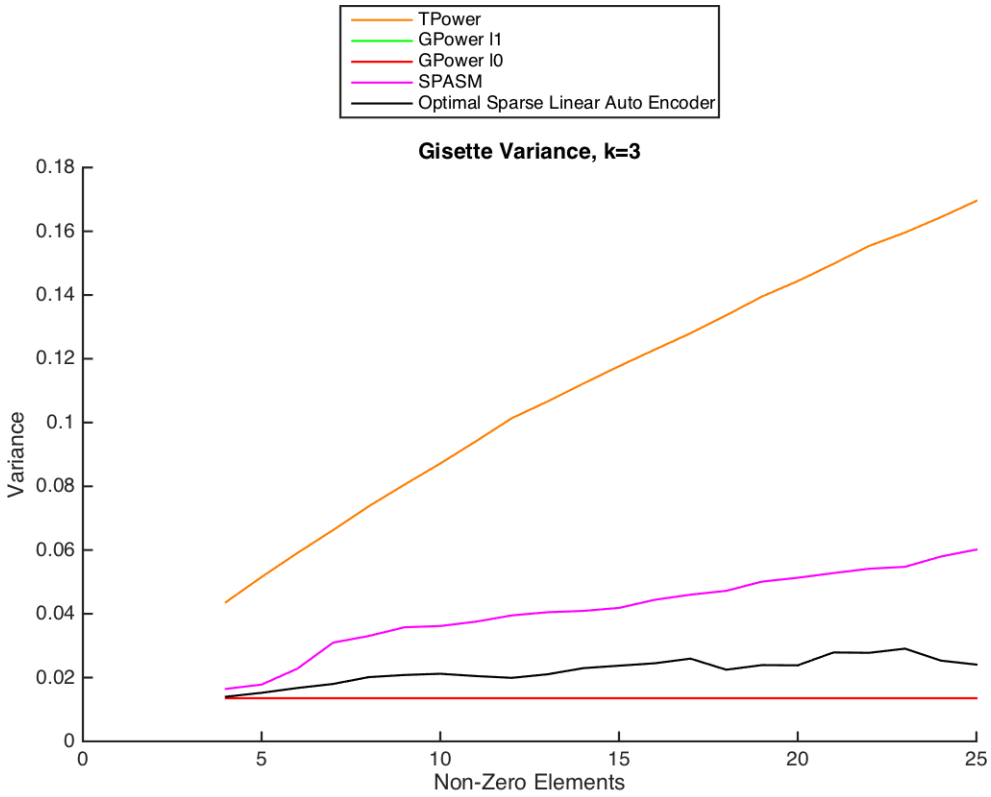


Figure 4.9: Gisette Symmetric Variance at k = 3

4.5 Madelon

Madelon's artificial, multivariate data set makes it very hard for machine learners to create a successful model for classification. As such, the base SVM performs poorly, and the encoders do a good job of creating a better model. At $k \geq 5$ all encoders perform as well if not better than the base SVM, with SpaSM and the Optimal Sparse Linear Auto-Encoder surpassing the base SVM at $k \geq 2$. SpaSM achieves the best accuracy throughout the experiment at $k = 6$ with $> 85\%$ correct classification. The Optimal Sparse Linear Auto-Encoder performs very well compared to SpaSM, until $k = 6$ when it is surpassed. GPower and TPower do fairly poorly in comparison, and do not improve much after $k = 3$.

The Optimal Sparse Linear Auto-Encoder has the best information loss ratio for most of the tests, except when $k = 6$; GPower with penalties l_0 and l_1 surpass its ratio at some points, but their ratios fluctuate a lot throughout the experiment. SpaSM does not have a great ratio in comparison, yet it still performs the best in classification prediction.

GPower with penalty l_1 excels in maximizing symmetric variance for this data set, nearly reaching a 1.0 ratio at $k \geq 2$, while the rest of the encoders lag far behind. In fact, the two best performing encoders for classification have the worst variances.

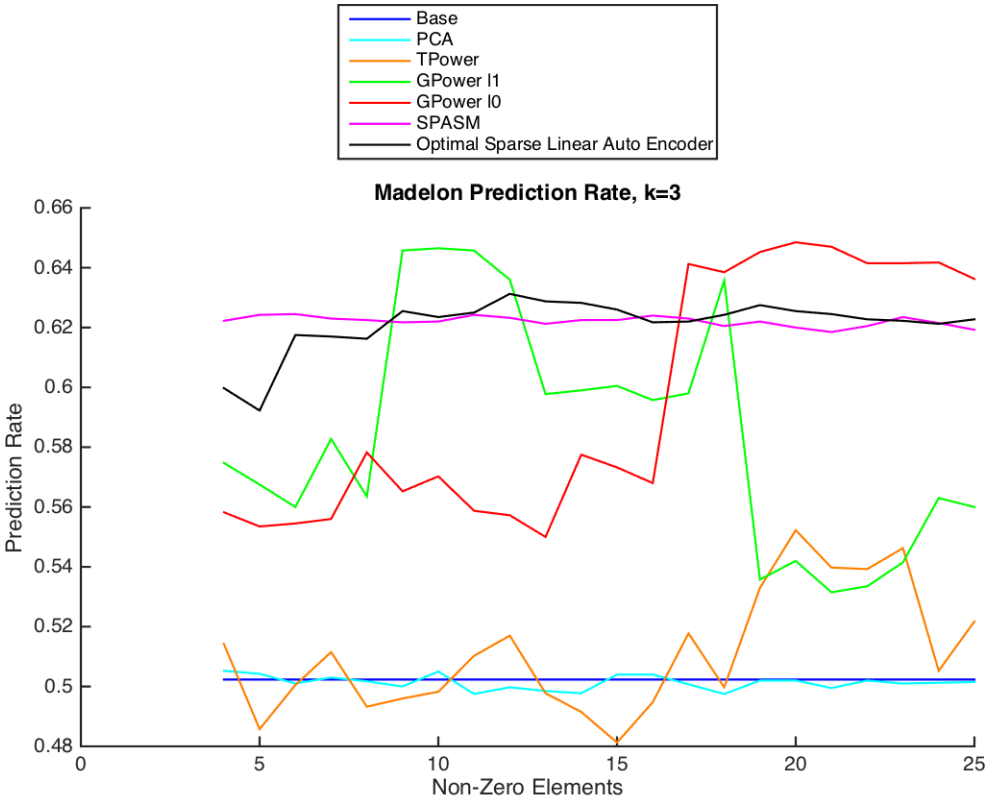


Figure 4.10: Madelon Classification Accuracy at k = 3

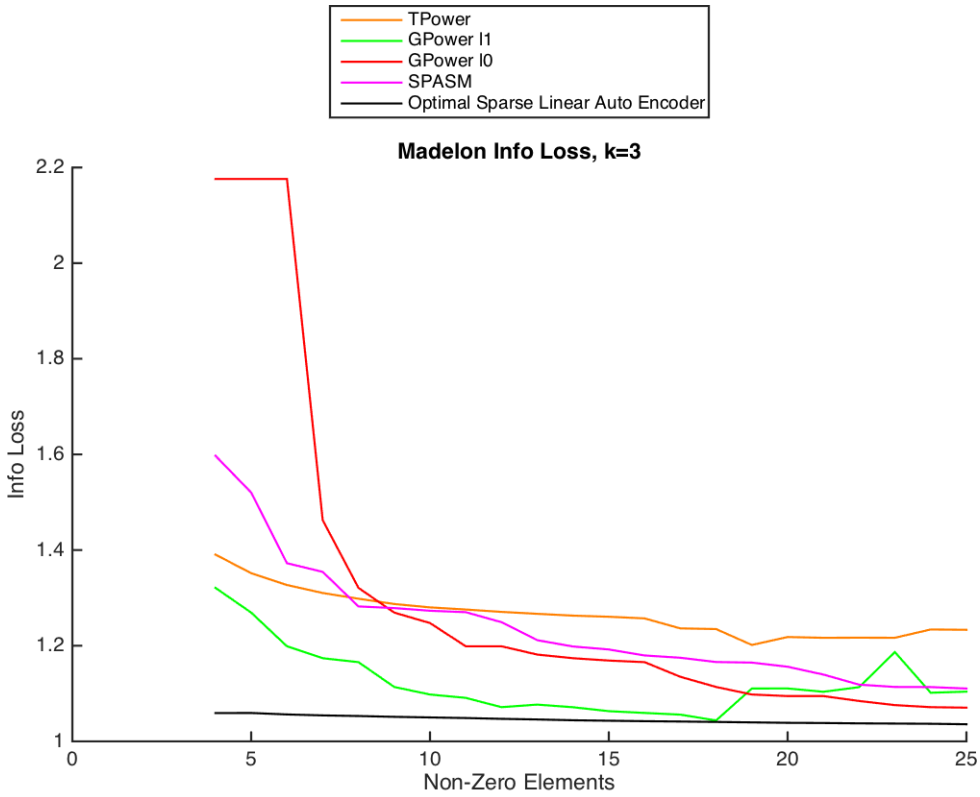


Figure 4.11: Madelon Information Loss at k = 3

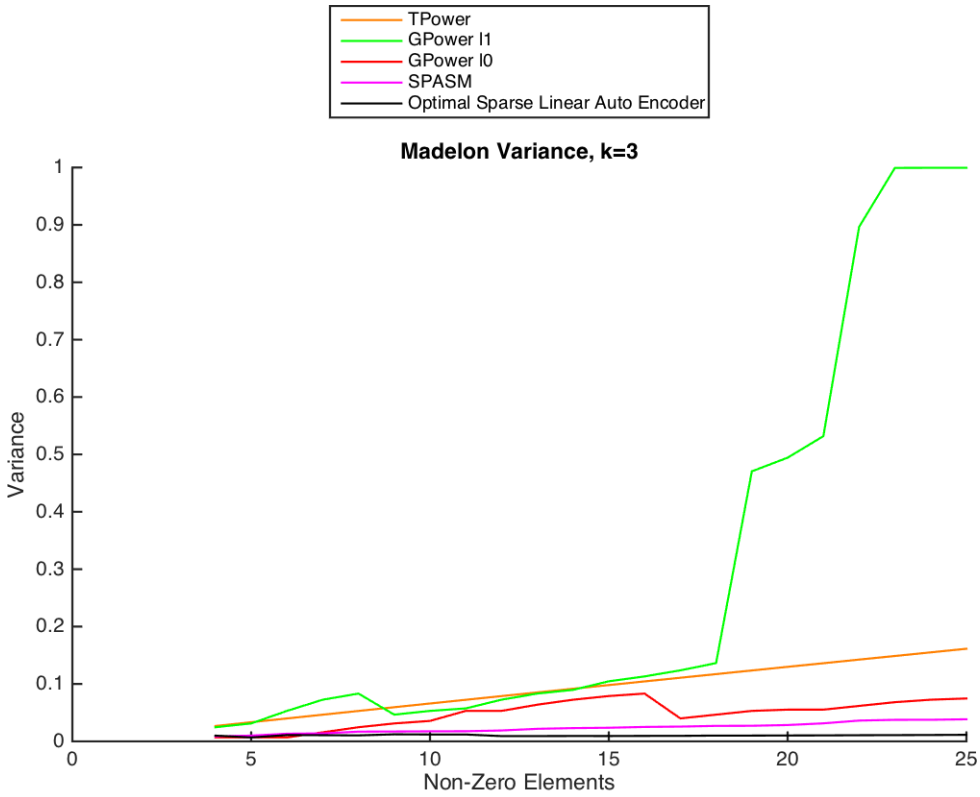


Figure 4.12: Madelon Symmetric Variance at k = 3

4.6 Ones and Fives

Ones and Fives showcased very good performance when using the encoders, especially the Optimal Sparse Linear Auto-Encoder and SpaSM. The SVM with the data set alone performed poorly: about an average $\sim 65\%$ prediction rate. With a modest $k = 1$ and sparsity $r = 2$, the Optimal Sparse Linear Auto-Encoder was able to create a model with over $\sim 95\%$ accuracy, which is incredible. After increasing k to 2, SpaSM was able to match the performance of the Optimal Sparse Linear Auto-Encoder, but the other methods lagged behind and were not able to match the performance, even with higher k values. Increasing k generally increased the performance of all the encoders.

The information loss ratio of all the encoders overall get closer to 1.0 when increasing the number of non-zero elements. The Optimal Sparse Linear Auto-Encoder does the best at $k = 1$ but is beat by SpaSM at $k \geq 4$, and is fairly equal to SpaSM and $k = 2$ and $k = 3$. An interesting observation is these results match the classification accuracy results exactly.

Conversely, the Optimal Sparse Linear Auto-Encoder and SpaSM have the worst symmetric variance comparatively to the other encoders. GPower with penalty l_1 does best until $k \geq 5$ when TPower surpasses it at some points.

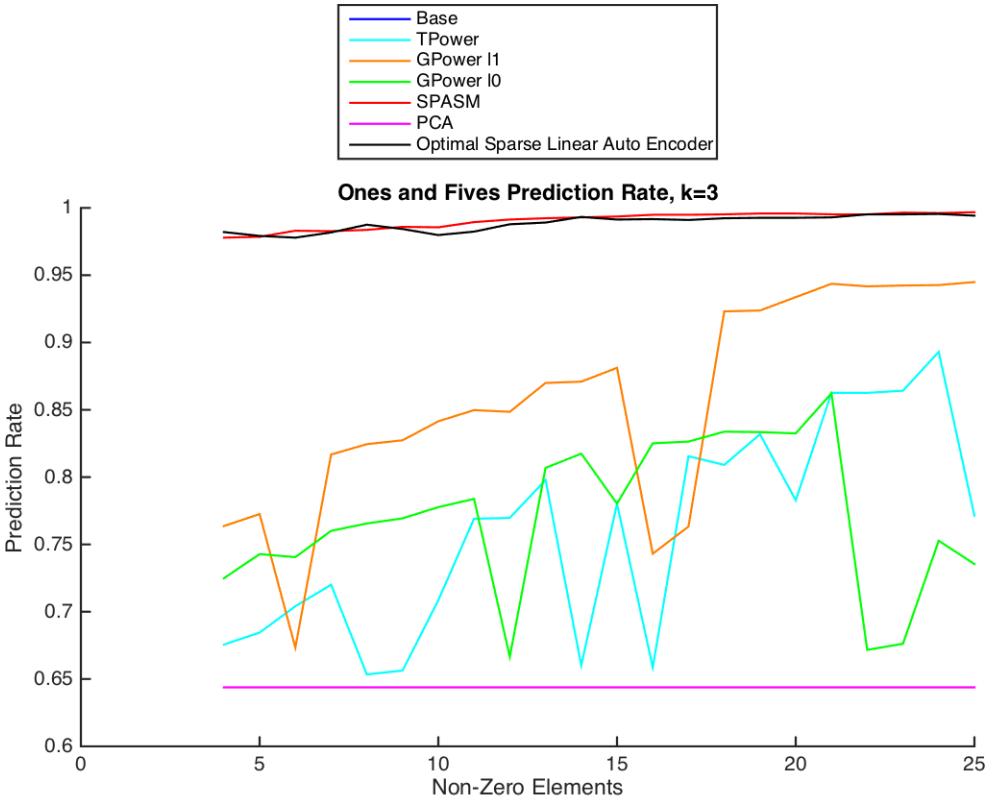


Figure 4.13: Ones and Fives Classification Accuracy at k = 3

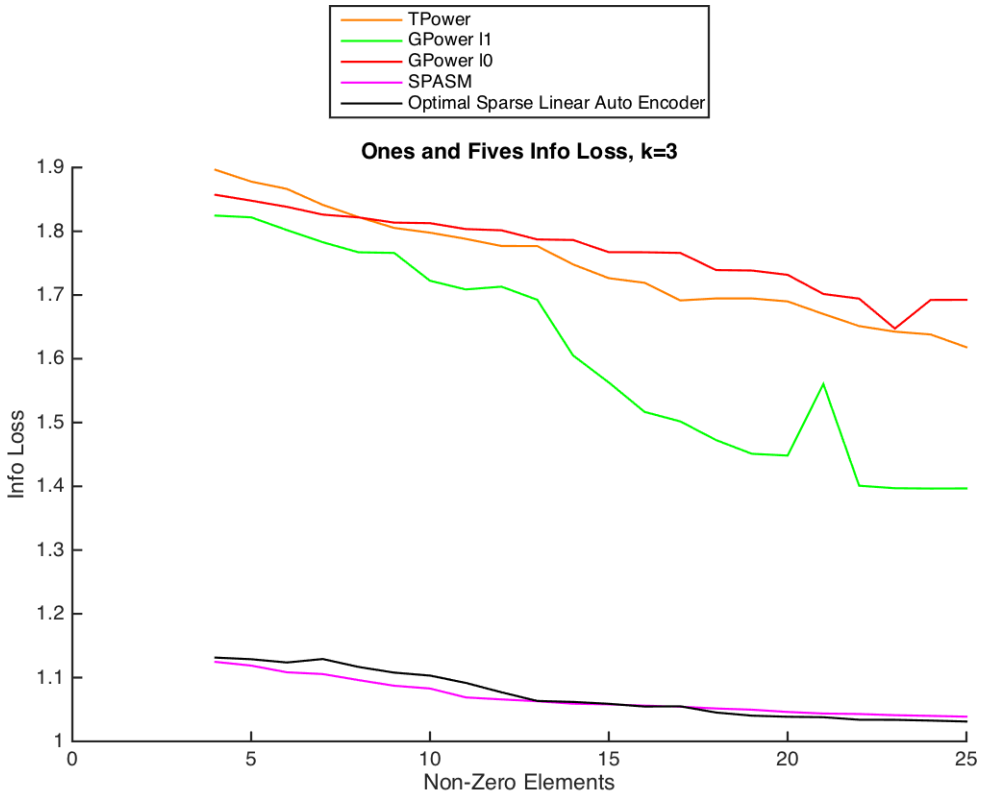


Figure 4.14: Ones and Fives Information Loss at k = 3

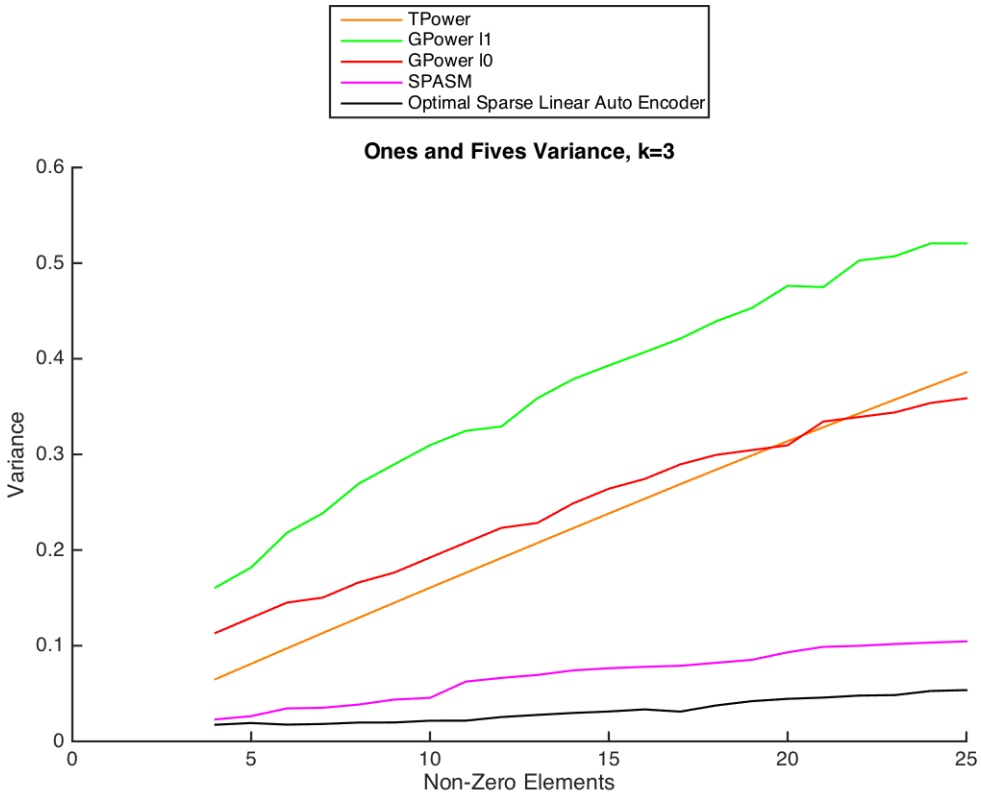


Figure 4.15: Ones and Fives Symmetric Variance at k = 3

4.7 Ozone

This data set had great performance in the SVM when unmodified, and worse performance when encoded. Performance at $k = 1$ was very unpredictable: increasing the number of non-zero elements caused the performance of the encoders to jump from a pitiful 10% accuracy to above 50% in some cases. Overall, the average performance does not become acceptable until $k \geq 5$ where the TPower encoder exceeds 70% classification accuracy. GPower and the Optimal Sparse Linear Auto-Encoder did poorly throughout most of the testing, with SpaSM only performing marginally better. Overall, this data set did not respond well to encoding.

The Optimal Sparse Linear Auto-Encoder and SpaSM had the best information loss ratios, with TPower almost matching at some k values. GPower's ratio did not change dramatically as its γ parameter for controlling sparsity did not respond well to most modifications.

The encoders captured most of the variance at $k \geq 3$, with only GPower with penalty l_0 lacking in the $k = 2$ and $k = 3$ tests. Overall, TPower was the best, but the difference between it and the other encoders is very marginal.

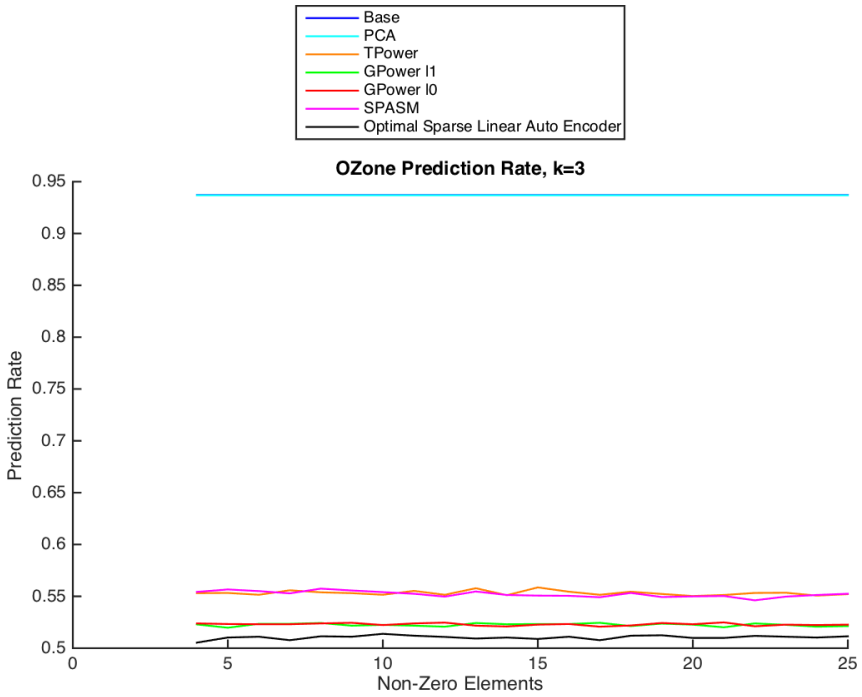


Figure 4.16: Ozone Classification Accuracy at k = 3

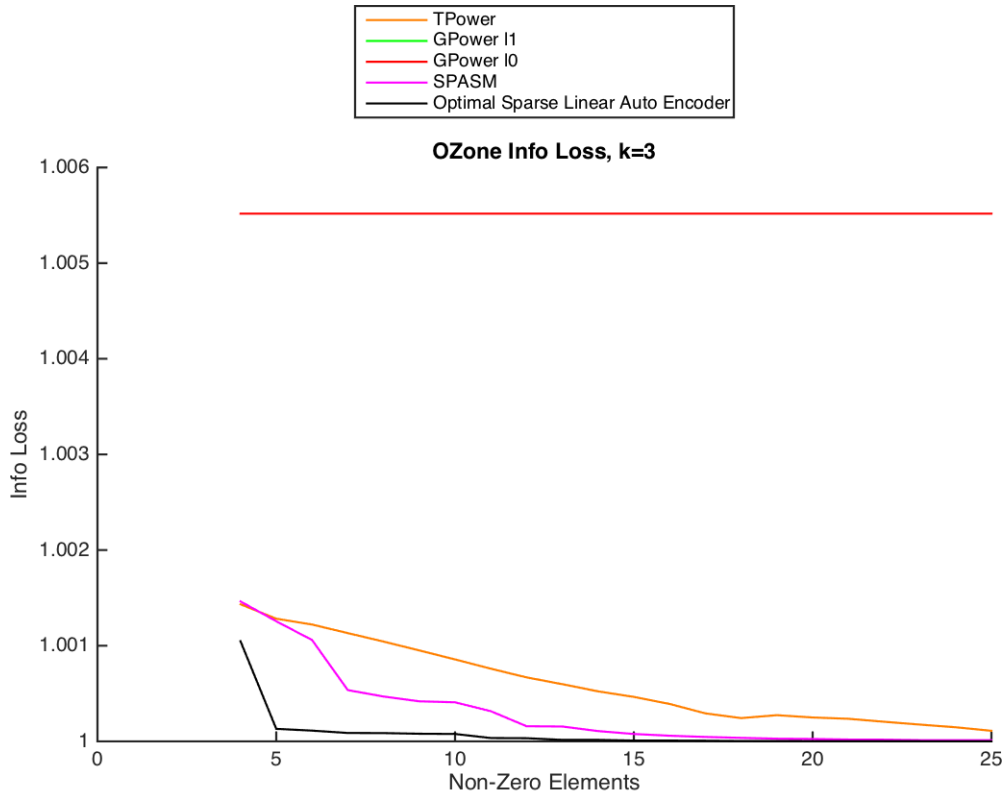


Figure 4.17: Ozone Information Loss at k = 3

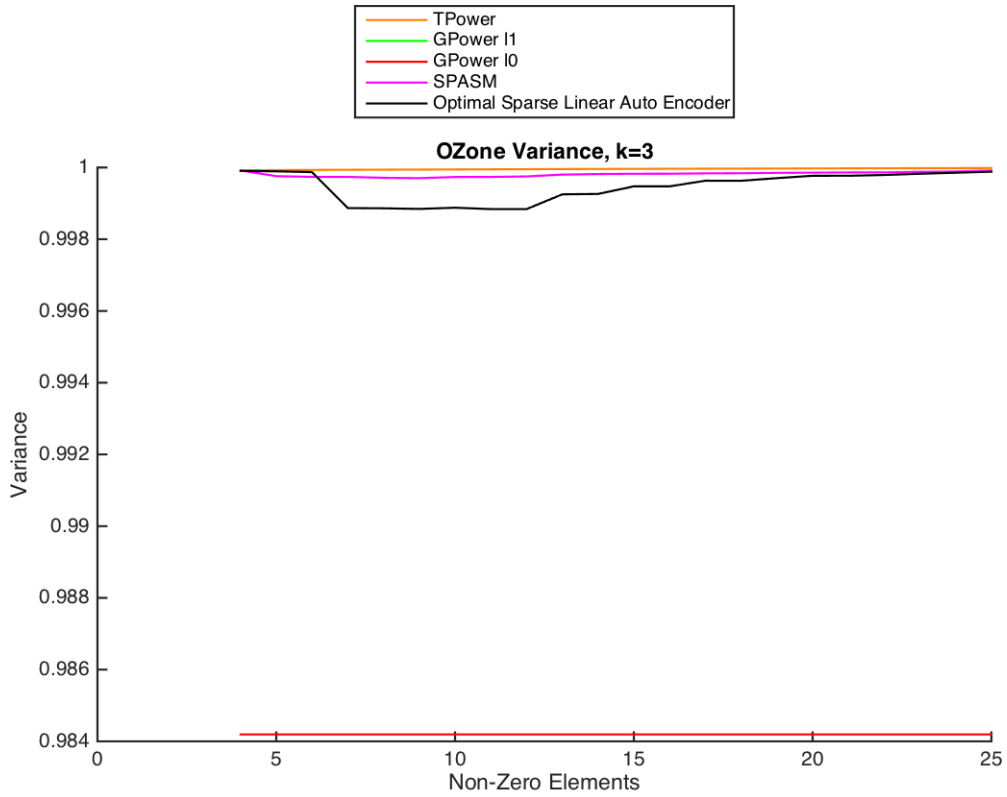


Figure 4.18: Ozone Symmetric Variance at k = 3

4.8 Ozone One Hour

This data set, like the original Ozone data set, has strong base SVM performance. However, there is an anomaly with the encoders: they perform best at $k = 1$, getting very close to matching the base SVM performance, but the performance decreases a significant amount when incrementing k. From $k = 1$ to $k = 3$ the Optimal Sparse Linear Auto-Encoder and GPower methods are the best encoders, but become the worst encoders at $k = 5$ and $k = 6$. TPower and SpaSM become the best encoders at $k = 5$ and $k = 6$.

The results for information loss on this data set are nearly identical to the other Ozone data set. The Optimal Sparse Linear Auto-Encoder and SpaSM had the best information loss ratios, and TPower was fairly close to matching them at some k values.

The encoders captured most of the variance at $k \geq 2$. TPower performs the best in capturing variance, but the difference between it and the other encoders is fairly marginal.

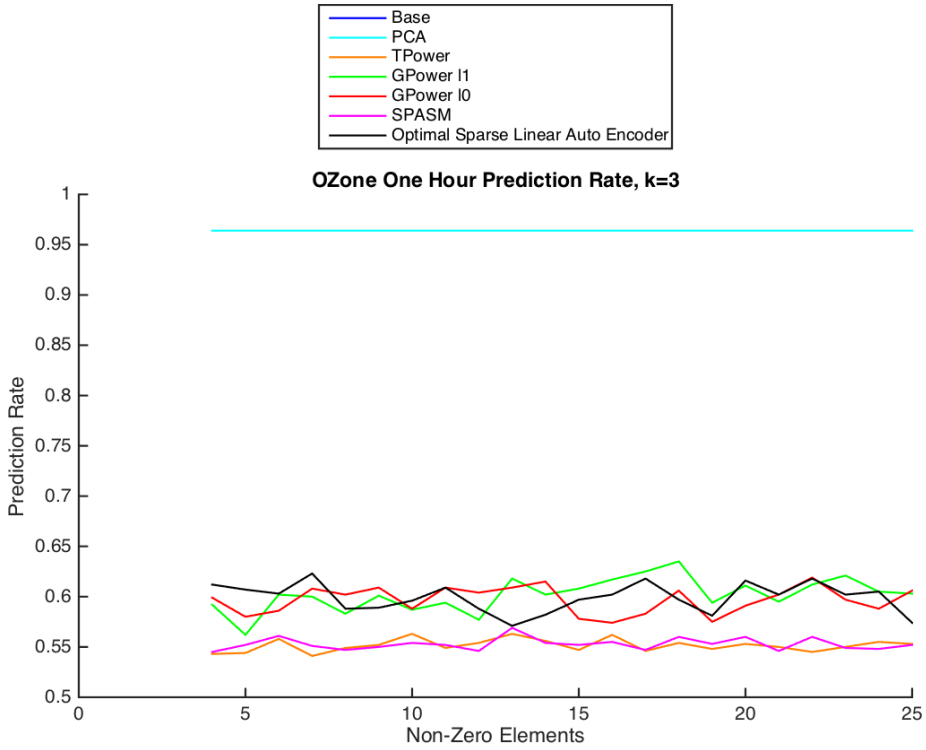


Figure 4.19: Ozone One Hour Classification Accuracy at k = 3

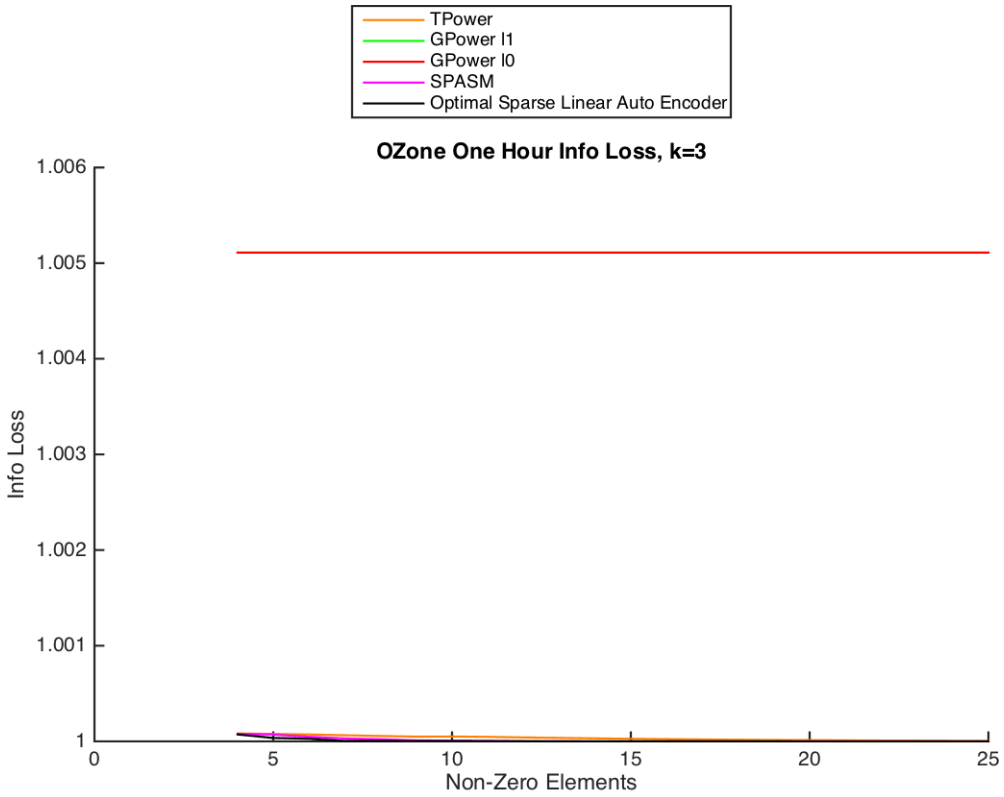


Figure 4.20: Ozone One Hour Information Loss at k = 3

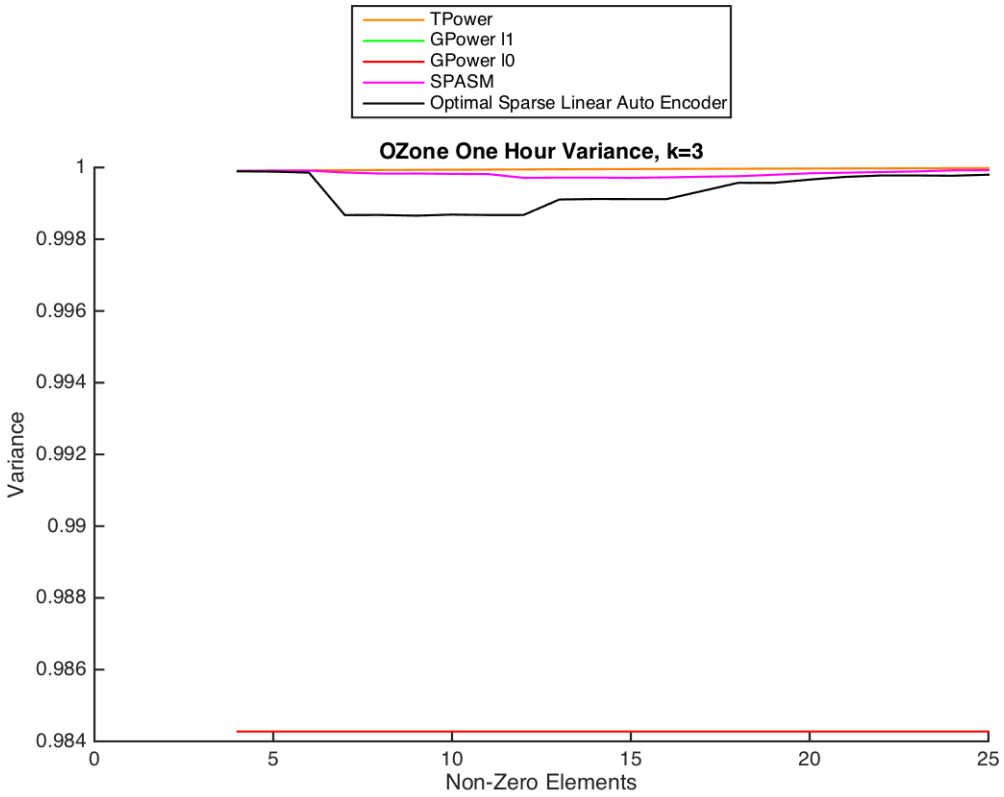


Figure 4.21: Ozone One Hour Symmetric Variance at k = 3

4.9 Pima Indians Diabetes

The Pima Indians Diabetes data sets did not glean much benefit from using the encoders. Without encoding the data set, the SVM created a model with an average $\sim 75\%$ accuracy. Across most of the tests, the SPCA methods were not able to create an encoded data set that surpassed the base SVM's accuracy, however their performances did get close. GPower with penalty l_0 does better than the base SVM at $k = 6$ and $r = 7$ but its performance drops when r is incremented. This data set exhibited a strong correlation between a larger k and increased performance.

The Optimal Sparse Linear Auto-Encoder has the best information loss ratio throughout the tests. SpaSM and TPower typically come close, with GPower being the worst of the set.

TPower excelled in the maximization of symmetric variance for this data set. It beat out the other encoders in all but one test, while SpaSM and the Optimal Sparse Linear Auto-Encoder followed closely to its performance. The Optimal Sparse Linear Auto-Encoder had the best variance in the $k = 5$ test, but becomes the worst encoder in the $k = 6$ test.

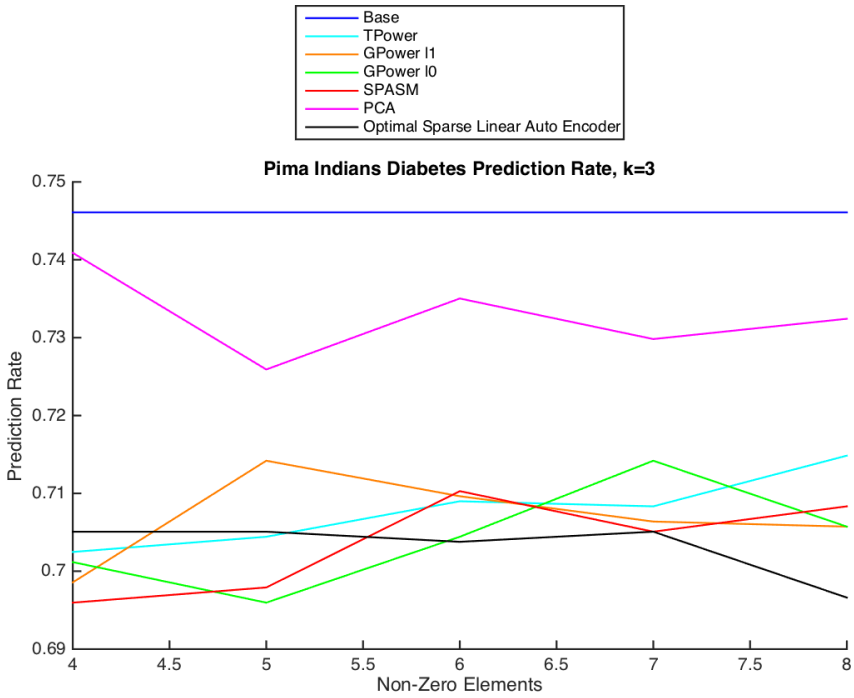


Figure 4.22: Pima Indians Diabetes Classification Accuracy at k = 3

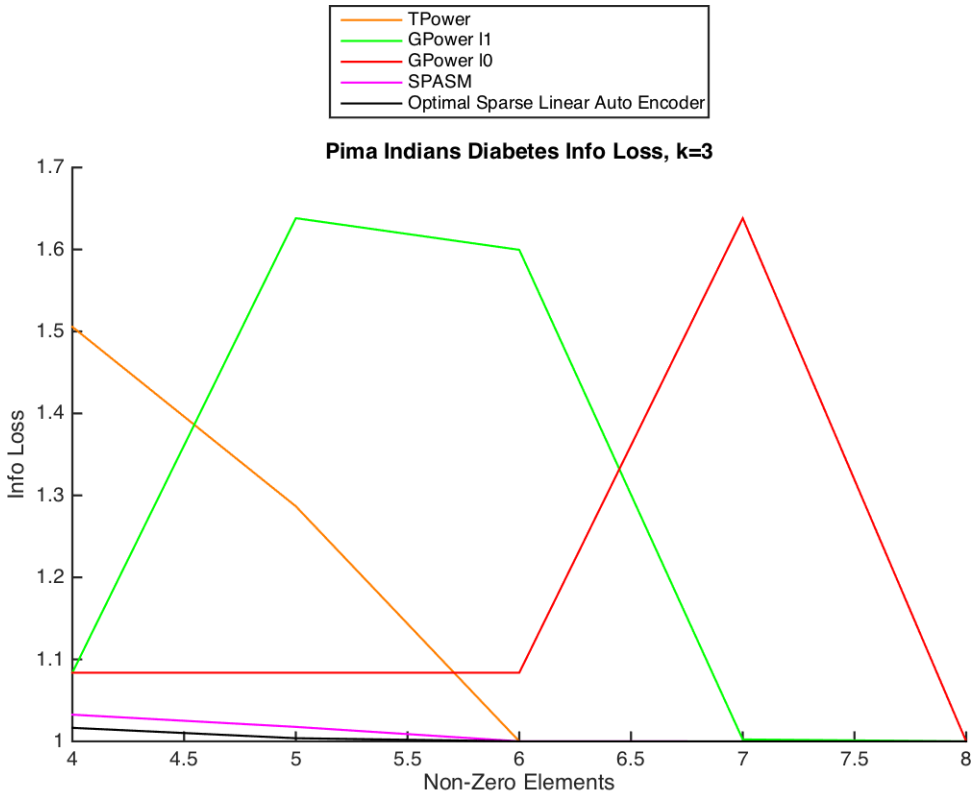


Figure 4.23: Pima Indians Diabetes Information Loss at k = 3

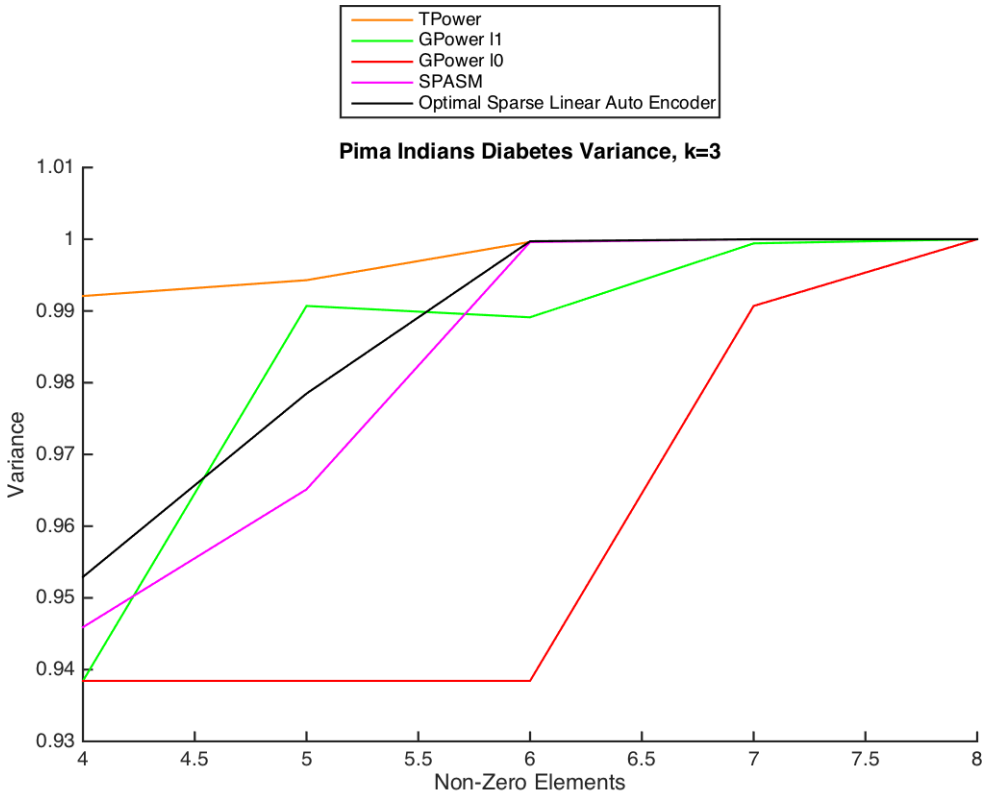


Figure 4.24: Pima Indians Diabetes Symmetric Variance at k = 3

4.10 SECOM

SECOM has strong base SVM performance, and encoding the data set leads to poorer performance. The highest classification accuracy is achieved at $k = 1$ and $r = 3$ by TPower, but its performance decreases as the number of non-zero elements is increased. The Optimal Sparse Linear Auto-Encoder and SpaSM are the best performing encoders at $k = 3$, $k = 5$, and $k = 6$, but are the worst at $k = 4$. TPower performs better than the two at times at $k = 2$, and better than the Optimal Sparse Linear Auto-Encoder for a few iterations at $k = 6$. Overall, however, TPower does worse over time, and its performance decreases with more non-zero elements. Overall, this data set did not respond very well to encoding.

SpaSM has the best information loss ratio, with the Optimal Sparse Linear Auto-Encoder closely following in second place. GPower does poorly; in fact its ratio gets worse when incrementing k.

TPower has the best symmetric variance across all tests. SpaSM has the second best, the Optimal Sparse Linear Auto-Encoder follows in third, and GPower had the worst.

Due to the difficulty of matching the sparsity parameter γ in GPower with the numerical sparsity r in the other encoders, there is little change in the results of GPower in the provided graphs.

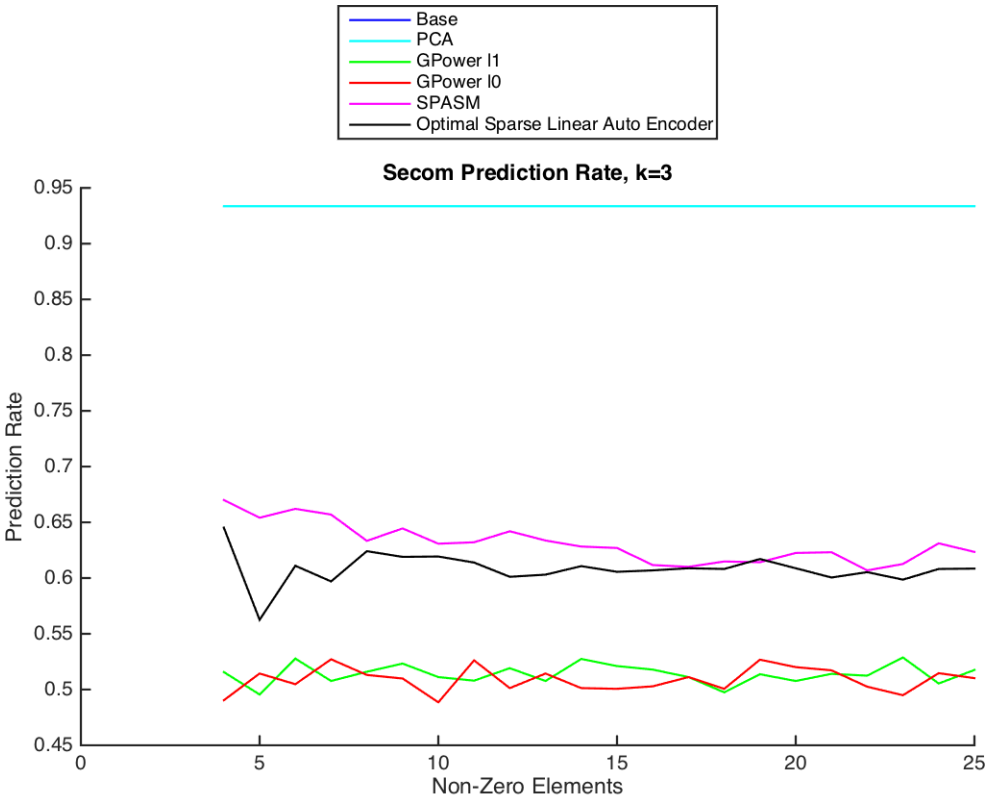


Figure 4.25: SECOM Classification Accuracy at k = 3

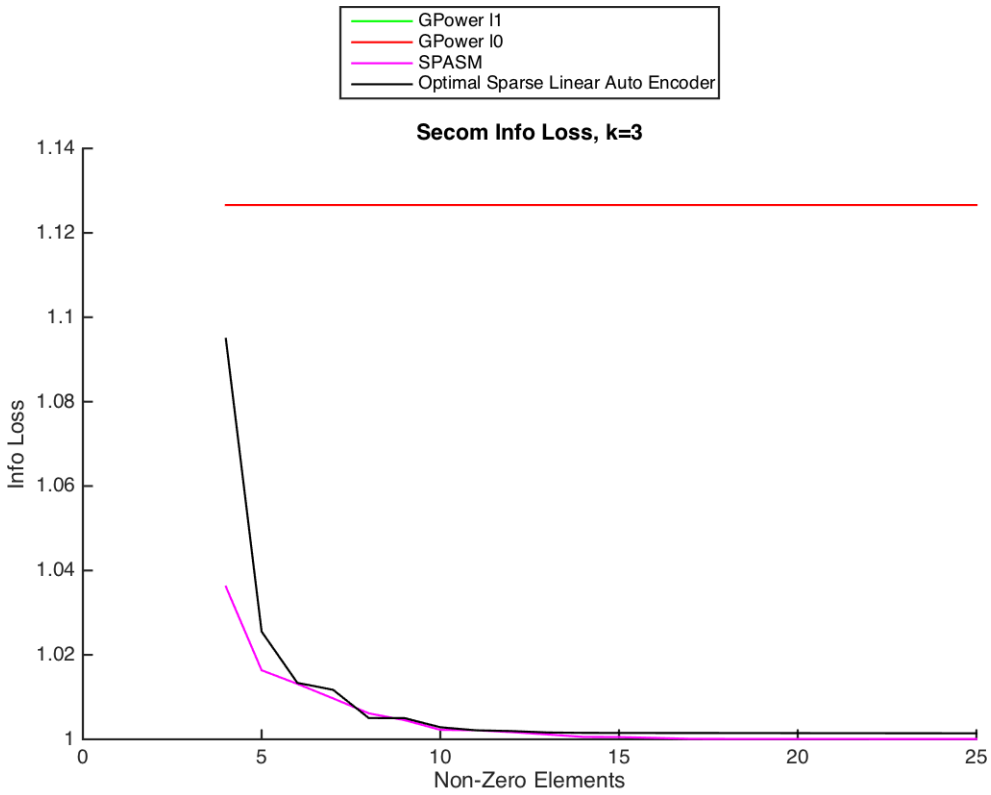


Figure 4.26: SECOM Information Loss at k = 3

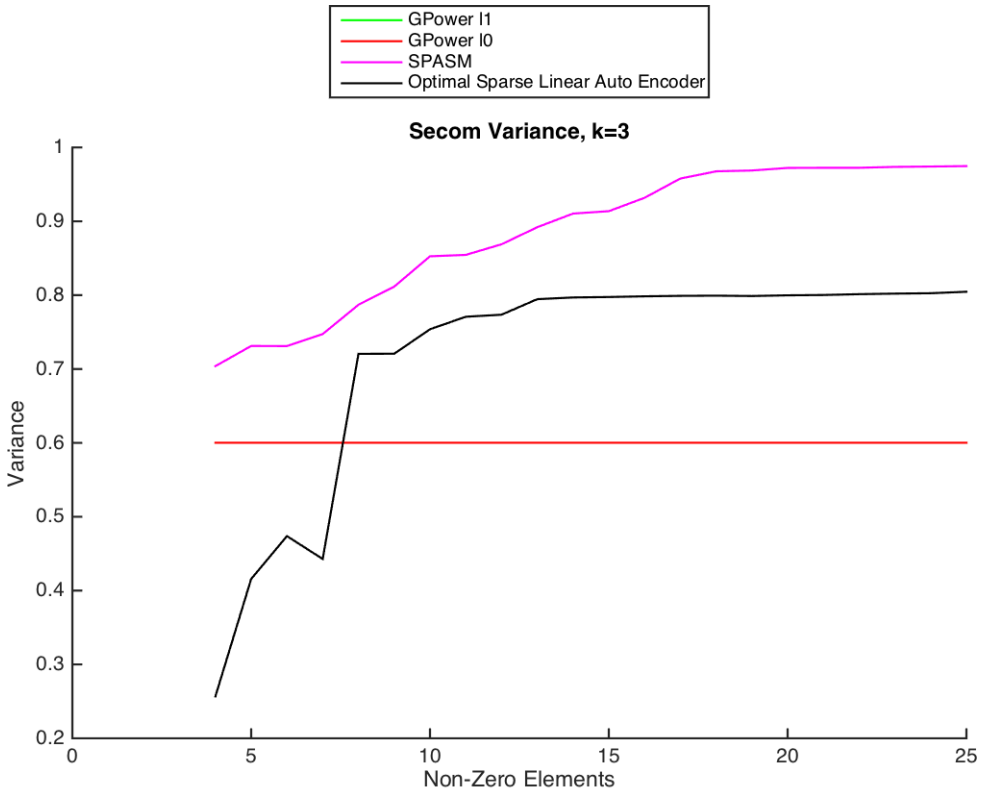


Figure 4.27: SECOM Symmetric Variance at k = 3

4.11 Spam

The Spam data set has strong base SVM performance, and the encoders get close to matching it with higher k values. GPower is the best encoder from $k = 1$ to $k = 3$, but becomes the worst thereafter. TPower, SpaSM, and the Optimal Sparse Linear Auto-Encoder were all the best performing encoders at some point during the testing when $k > 3$. There is a lot of variance in the classification accuracy throughout the tests: incrementing the number of non-zero elements caused accuracy to spike up and down creating many local minima and maxima, which makes results a little unpredictable.

The information loss ratio is so good across the encoders that the results for $k = 1$ and $k = 2$ are essentially hidden due to GPower's "lacking" performance. The Optimal Sparse Linear Auto-Encoder performs best at $k = 3$, but is superceded by TPower for the remaining tests. SpaSM is in second place for $k = 3$ and $k = 6$.

A similar outcome to information loss occurs; GPower does much worse than the rest of the encoders, which are very near to a 1.0 ratio. Overall, TPower does the best in all tests, with SpaSM in second place for all by $k = 5$, where the Optimal Sparse Linear Auto-Encoder beats SpaSM for some iterations of r.

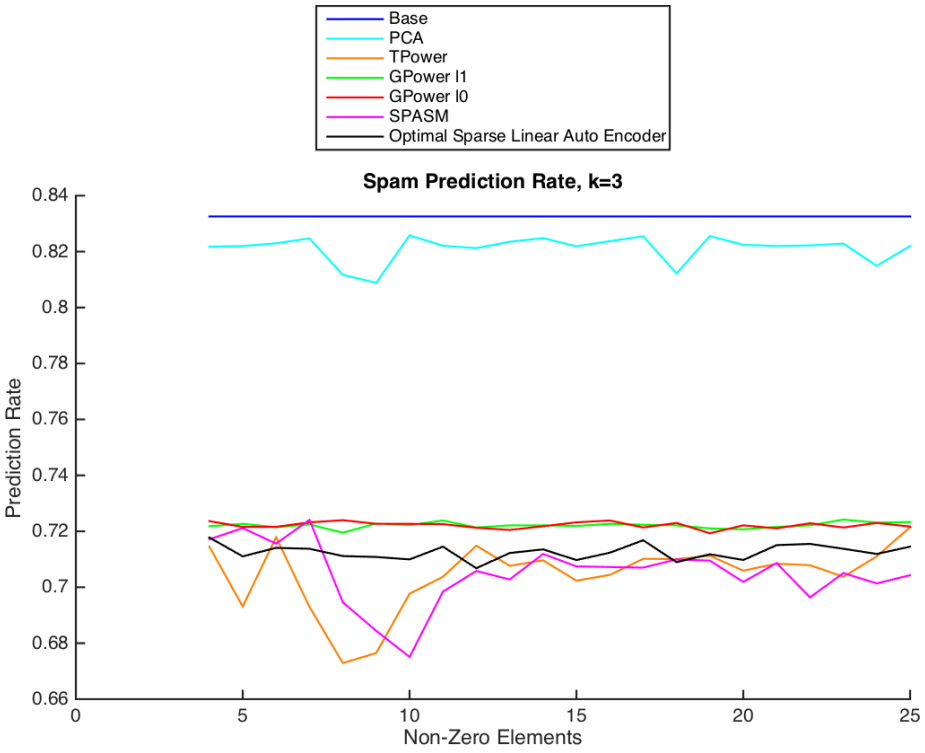


Figure 4.28: Spam Classification Accuracy at k = 3

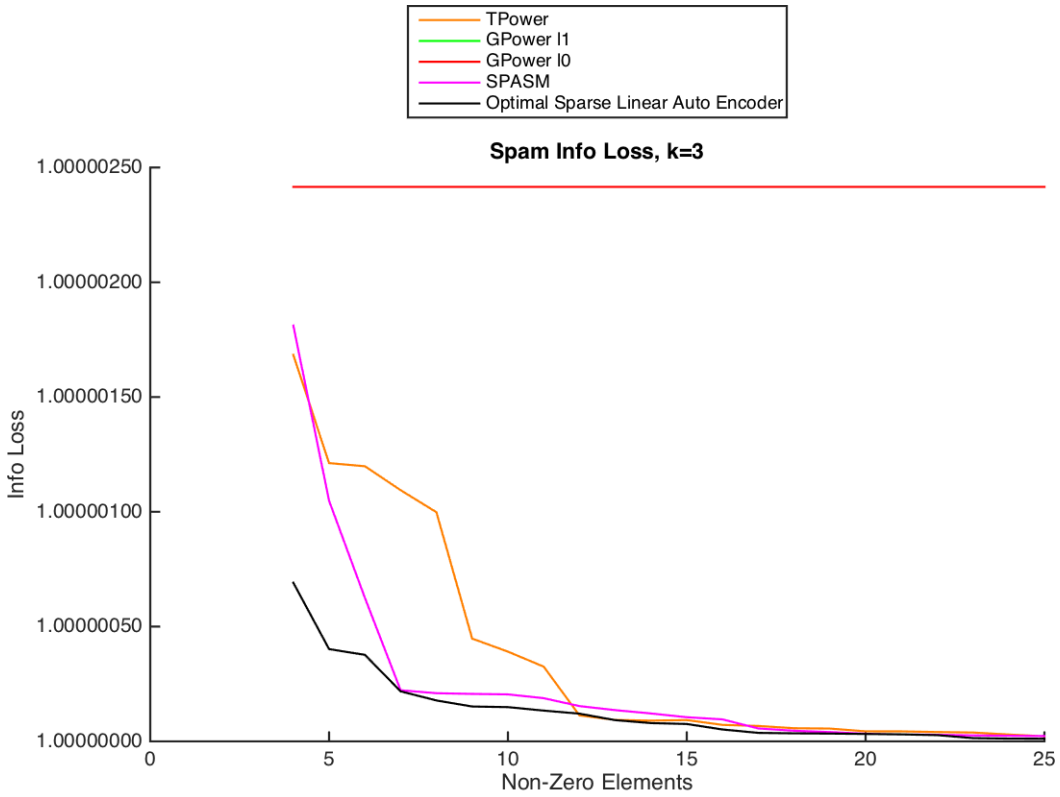


Figure 4.29: Spam Information Loss at k = 3

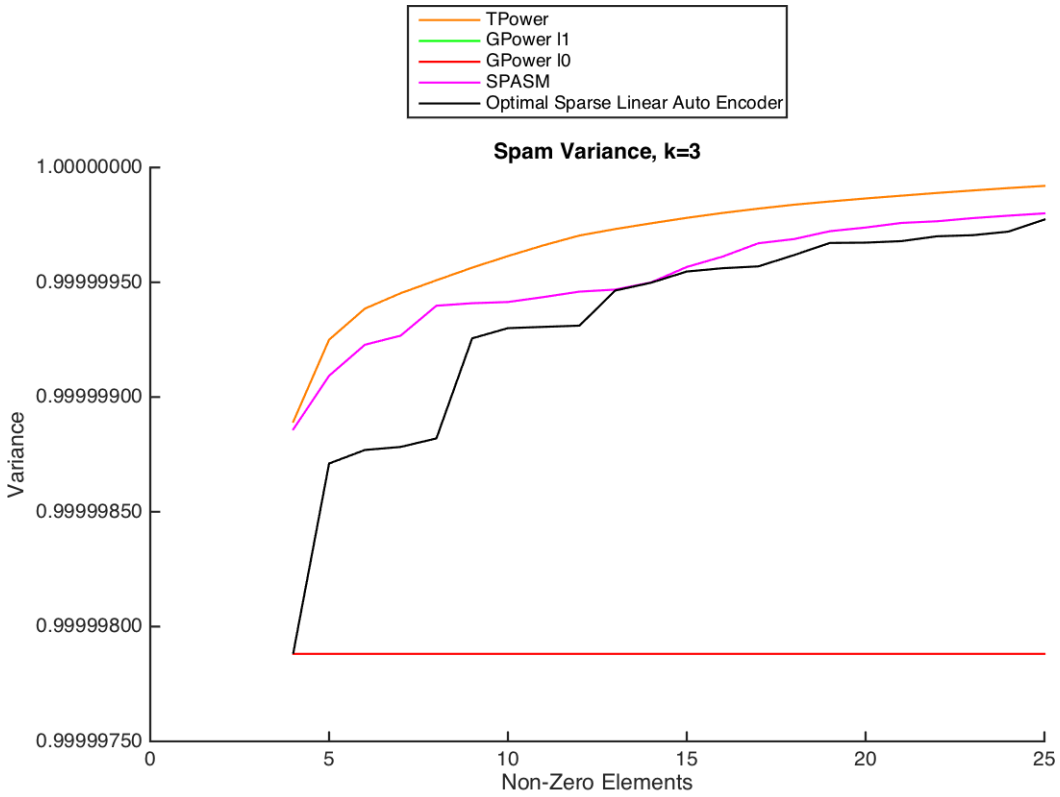


Figure 4.30: Spam Symmetric Variance at k = 3

5. Discussion and Conclusion

5.1 CONCLUSION

Using an encoder for data sets that are learned efficiently by an SVM alone typically does not net good results. In particular, data sets where the SVM was able to create a model with a correct classification rate of 75% or higher exhibited this behavior, such as the Pima Indians Diabetes data set, the two Ozone data sets, the Spam data set, the SECOM data set, and the Chronic Kidney Disease data set. In fact, using an encoder typically will produce a worse-performing model, and it is the opinion of the author that the user is better off ignoring using SPCA if their machine learning method already creates a sufficient model with the unchanged data set. However, in data sets where the base SVM creates a poorly performing model, we saw a marked increase in the performance of the created model when the data set was encoded with the methods tested. In general, it appeared that data sets with high dimensionality exhibited this behavior; the SVM was unable to create a sufficient model, and encoding the data set produced far better performance. Vision data sets such as Gisette and Ones and Fives, in particular, had great performance.

We hypothesize that encoding led to poor results on Chronic Kidney Disease, and Pima Indians Diabetes data sets due to the structure of the data: the researchers who put those data sets together already knew about the relation of the features within that set, and only included features that positively influenced the classification accuracy of the model. In other words, the data set was designed to be easily learned, and all the data given is important to the learning of the model.

Across most of the data sets, increasing the number of non-zero elements while keeping k fixed, had little to negligible impact on the performance of the SVM, even though the symmetric variance and information loss ratios improved. Increasing the value of k typically did increase performance, but with some encoders in specific data sets, it was a fairly negligible increase.

Increasing the number of non-zero elements generally improved the information loss ratio for all the encoders. Due to the nature of feature selection, occasionally

the information loss ratio would get worse for a specific encoder when increasing the number of non-zero elements. However, given enough iterations, the trend was improvement. The Optimal Sparse Linear Auto-Encoder almost universally improved its information loss ratio when increasing the non-zero elements, and this is to be expected by design.

The symmetric variance of the encoded data set had little influence on the performance of the predictive model. In several tests, a poorly performing encoder would have the best symmetric variance out of the encoders tested. It is the opinion of the author, based on the data collected in these experiments, that the goal of maximizing the variance of an encoder may not be essential to creating an SPCA method that can perform well in a machine learning context.

Methods which had a strong information loss ratio typically performed well. Conversely, there is only a strong relationship between a good variance ratio and good SVM performance in a handful of tests, and in some of these tests, the method with the best variance ratio also had the best information loss ratio. Therefore, we saw a stronger relationship with good information loss ratios and SVM performance. In particular, we saw strong relationships between the information loss ratio and classification accuracy in Gisette, Madelon, Ones and Fives, Ozone One Hour, SECOM, and the Spam data sets. The Arcene data set exhibited this for $k = 5$ and $k = 6$.

In the tables on the next page, OSLAE stands for Optimal Sparse Linear Auto-Encoder, CKD stands for Chronic Kidney Disease, OF stands for Ones and Fives, OOH stands for Ozone One Hour, and PID stands for Pima Indians Diabetes. Encoders marked with an asterisk may not have the exact same number of non-zero elements in the encoder, due to limitations of the algorithm in choosing those elements. k is fixed at 6, and r is fixed at 7 for these results. Boldface numbers are the best performers for that data set, while italics denote the second best performer for the data set. Recall that TPower ran too slowly on the ARCENE data set so it was not included in the final results.

Table 5.1: Classification Accuracy Summary

Data Sets	TPower	GPower l_1*	GPower l_0*	SpaSM	OSLAE
ARCENE	0	0.675	0.650	**0.720**	*0.710*
CKD	*0.911*	**0.935**	0.900	0.896	0.880
Gisette	0.547	0.505	0.498	**0.814**	*0.685*
Madelon	0.538	0.665	0.543	**0.868**	*0.764*
OF	0.698	0.663	0.673	**0.993**	*0.992*
Ozone	**0.778**	0.534	0.517	*0.669*	0.560
OOH	**0.777**	0.571	0.543	*0.751*	0.590
PID	0.742	*0.745*	**0.749**	0.742	0.738
SECOM	0.657	0.654	0.674	**0.712**	*0.685*
Spam	**0.801**	0.782	*0.796*	0.781	0.787

Table 5.2: Information Loss Summary

Data Sets	TPower	GPower l_1*	GPower l_0*	SpaSM	OSLAE
ARCENE	0	2.499	2.499	**1.148**	*1.408*
CKD	*1.0004*	3.0712	1.0515	1.0007	**1.00005**
Gisette	1.167	1.190	1.190	**1.072**	*1.107*
Madelon	1.295	*1.077*	1.331	1.398	**1.069**
OF	2.354	2.219	2.291	**1.168**	*1.248*
Ozone	3.185	2.897	2.897	**1.094**	*1.188*
OOH	3.614	2.846	2.846	**1.088**	*1.153*
PID	*1.000002*	1.006	1.023	**1.000001**	6.734
SECOM	1.277	2.855	2.855	**1.027**	*1.044*
Spam	**1.003**	1.028	1.028	*1.011*	1.015

Table 5.3: Symmetric Variance Summary

Data Sets	TPower	GPower l_1*	GPower l_0*	SpaSM	OSLAE
ARCENE	0	**0.010057**	**0.010057**	*0.00877*	0.005496
CKD	**0.9999974**	0.9999749	0.9999779	0.9999968	*0.9999972*
Gisette	**0.099**	0.022	0.022	*0.034*	0.023
Madelon	**0.091**	*0.079*	0.050	0.014	0.029
OF	*0.210*	**0.270**	0.192	0.045	0.032
Ozone	**0.99997**	*0.99991*	*0.99991*	0.99987	0.99894
OOH	**0.999966**	*0.999901*	*0.999901*	0.999890	0.998657
PID	**0.9999945**	0.9996324	0.9995668	*0.9999944*	0.9987460
SECOM	**0.961**	*0.787*	*0.787*	0.757	0.459
Spam	**0.9999985**	0.9999945	0.9999945	*0.9999969*	0.9999958

SpaSM's classification accuracy is the best in five of the ten data sets, and the second best in two, which makes it the best encoder overall for classification. An interesting observation is that for every data set that SpaSM performs well in, the Optimal Sparse Linear Auto-Encoder performs second best, and in some cases, is very close to matching SpaSM's performance. Conversely, when SpaSM does not perform well, the three encoders, TPower, and the two GPower encoders, perform the best, and again, the Optimal Sparse Linear Auto-Encoder matches SpaSM's performance fairly closely. The Optimal Sparse Linear Auto-Encoder strictly depends on the column subset selection algorithm and the choice of this algorithm varies the performance of the encoder. In our research we tested three such algorithms, but one consistently performed the best, regardless of k or r. It is likely that a different column subset selection algorithm may help the Optimal Sparse Linear Auto-Encoder to perform even better, and surpass SpaSM's performance.

In particular, the fact that the Optimal Sparse Linear Auto-Encoder performed very nearly as well as SpaSM is incredible. Recall that SpaSM is essentially a supervised learner by performing regularized linear regression, while the Optimal Sparse Linear Auto-Encoder, and the other encoders, are unsupervised. This bodes very well for the Optimal Sparse Linear Auto-Encoder's usability and performance, and bodes very well for minimizing information loss vs maximizing variance.

References

[1] I. Jolliffe, *Principal Component Analysis*, 2nd ed. New York, NY, USA: Springer-Verlag, 1986, pp. 1 - 2.

[2] M. Magdon-Ismail and C. Boutsidis, "Optimal sparse linear auto-encoders and sparse PCA," *CoRR*, vol. abs/1502.06626, 2015, http://arxiv.org/abs/1502.06626 (Date Last Accessed, March 10, 2016).

[3] X. Yuan and T. Zhang, "Truncated power method for sparse eigenvalue problems," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 899–925, Apr. 2013.

[4] M. Journée, Y. Nesterov, P. Richtárik, and R. Sepulchre, "Generalized power method for sparse principal component analysis," *J. Mach. Learn. Res.*, vol. 11, pp. 517–553, Mar. 2010.

[5] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *J. Computational Graph. Stat.*, vol. 15, no. 2, pp. 265–286, 2006.

[6] K. Sjöstrand, L. H. Clemmensen, R. Larsen, and B. Ersbøll, "Spasm: A matlab toolbox for sparse statistical modeling," 2012, submitted for publication.

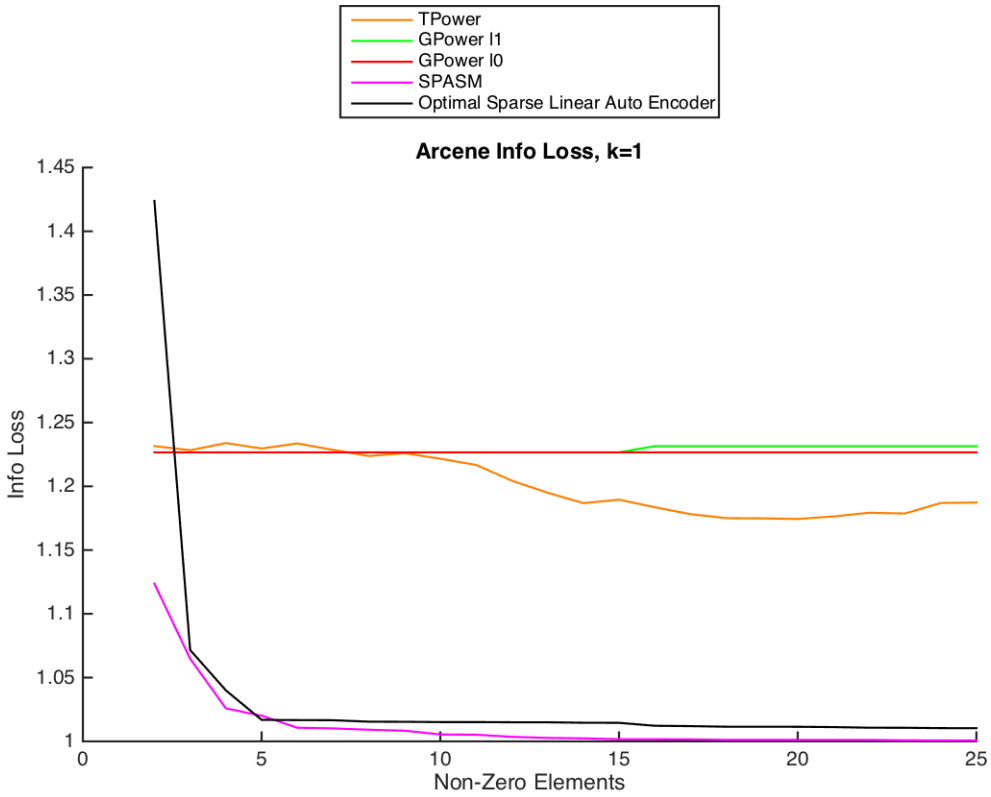
[7] C. Boutsidis, P. Drineas, and M. Magdon-Ismail, "Near-optimal column-based matrix reconstruction," *SIAM J. Computing*, vol. 43, no. 2, pp. 687–717, 2014.

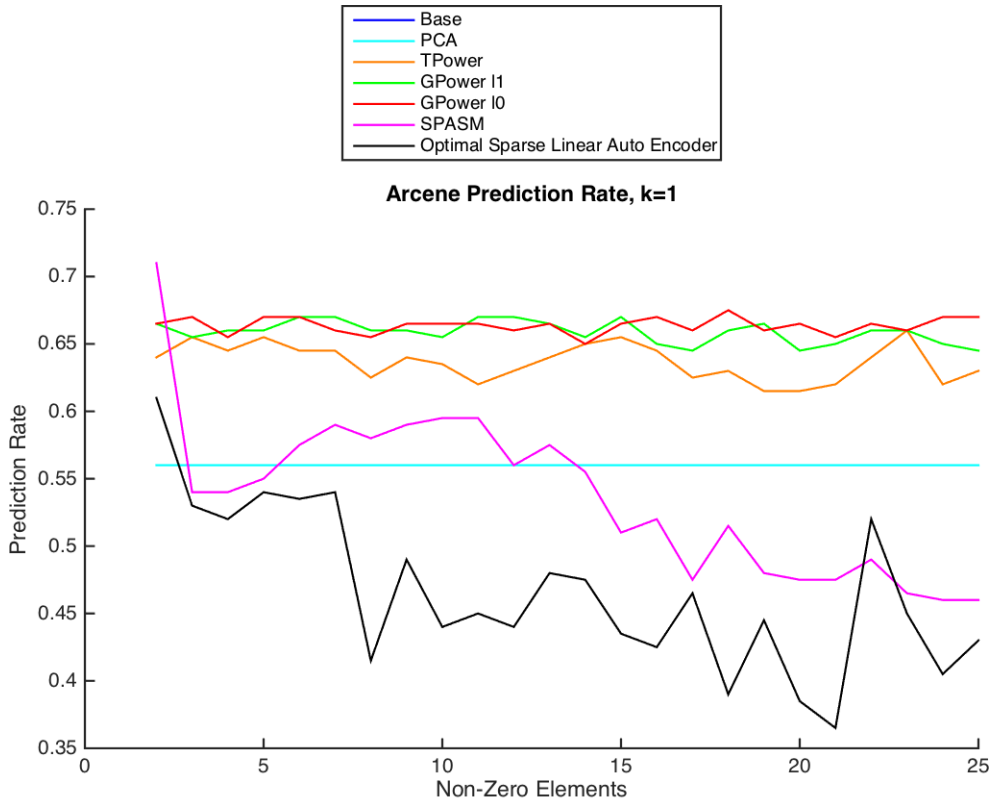
[8] M. Lichman, "UCI machine learning repository," 2013, http://archive.ics.uci.edu/ml (Date Last Accessed, February 18, 2016).

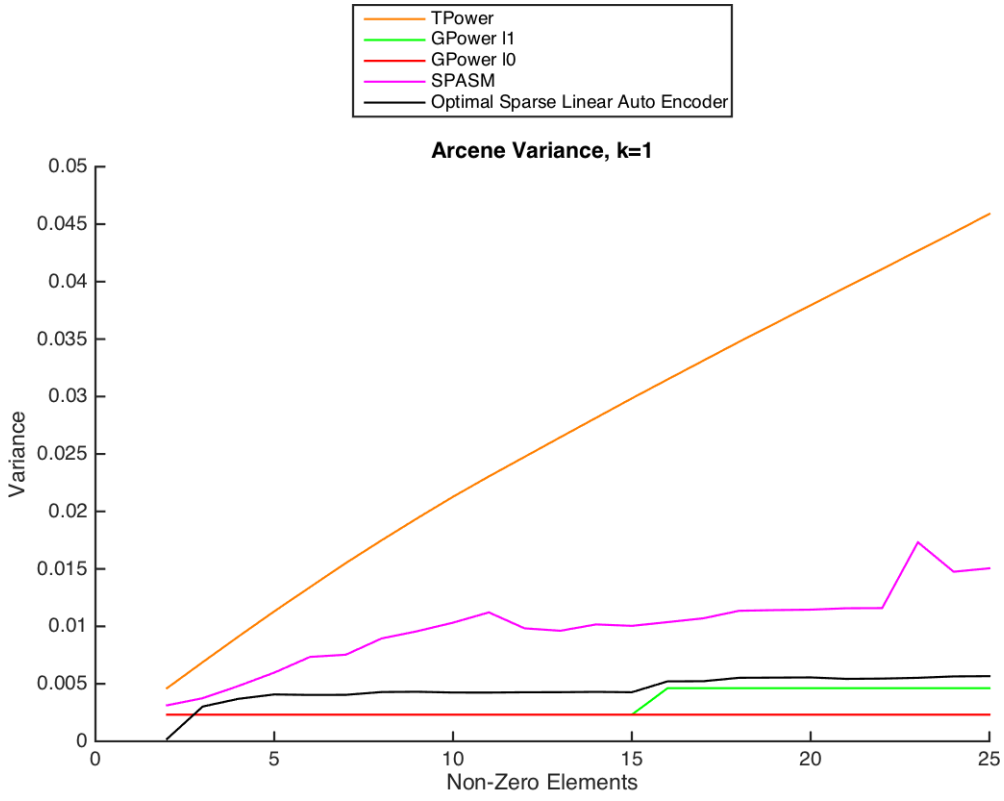
[9] I. Guyon, A. B. Hur, S. Gunn, and G. Dror, "Result analysis of the nips 2003 feature selection challenge," in *Advances in Neural Information Processing Systems 17*. MIT Press, 2004, pp. 545–552.

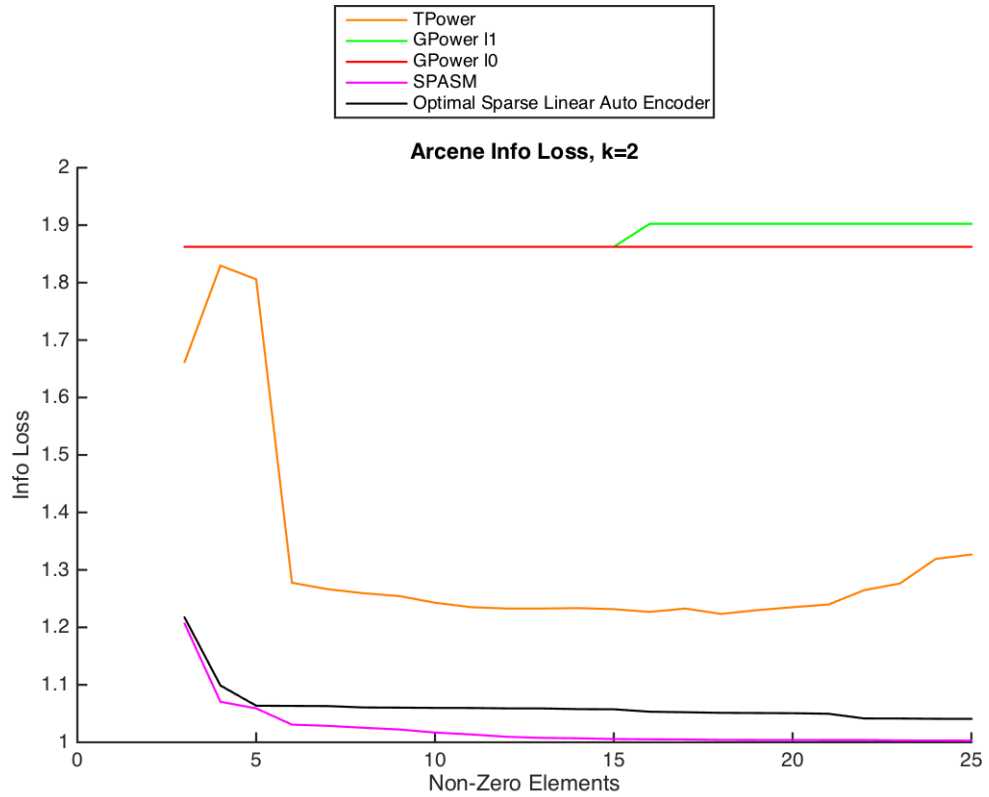
APPENDIX A
All Results

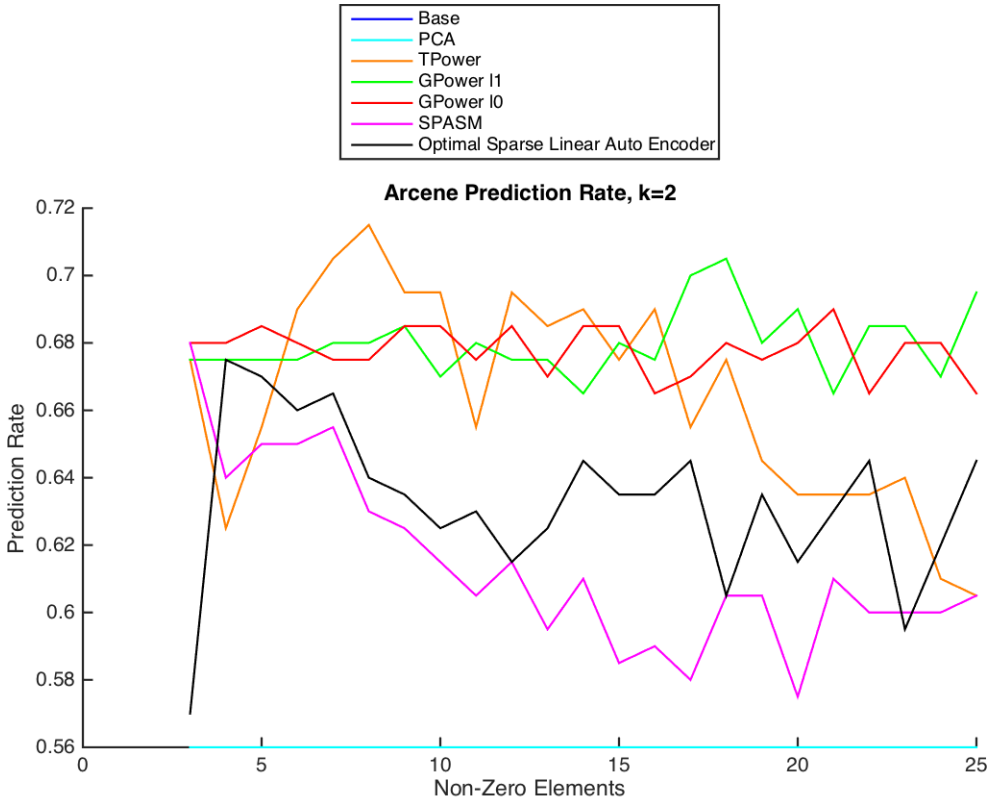
A.1 ARCENE Results

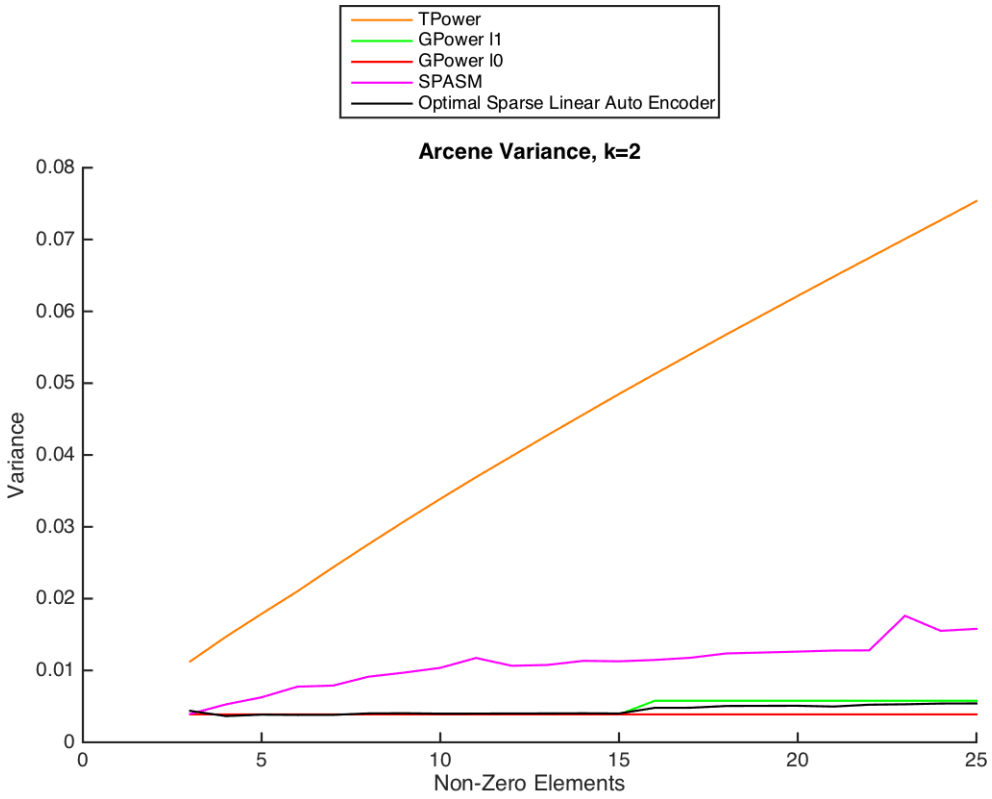


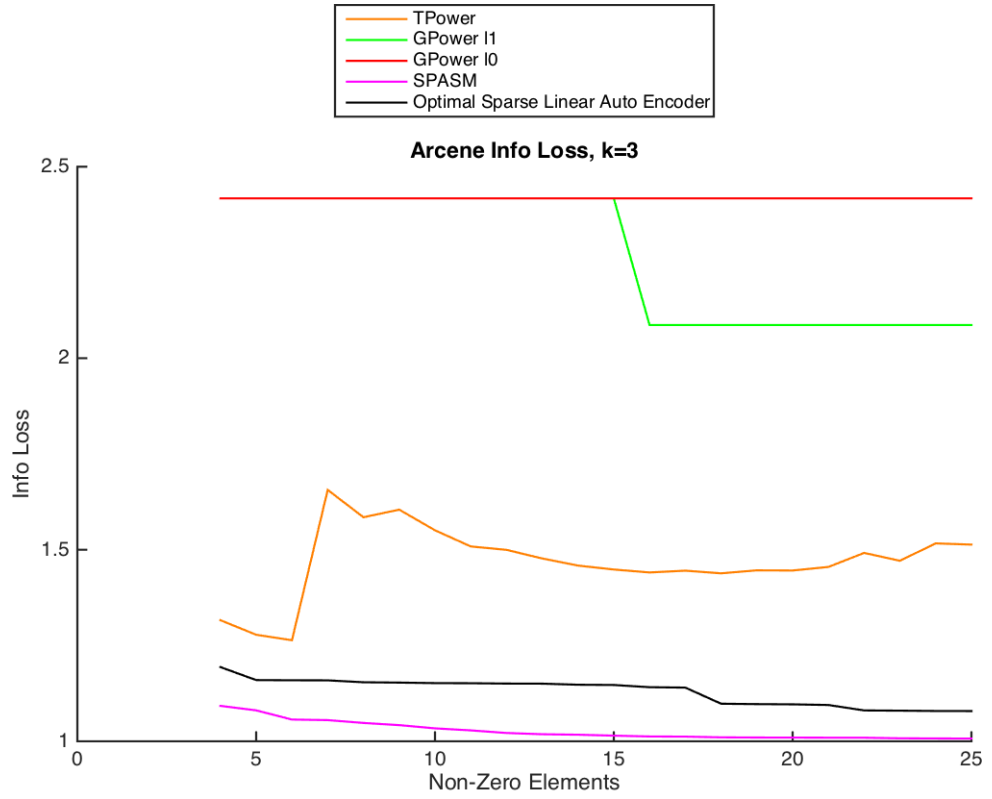

Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Arcene Prediction Rate, k=1
Prediction Rate
Non-Zero Elements
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.4
0.35
0
5
10
15
20
25

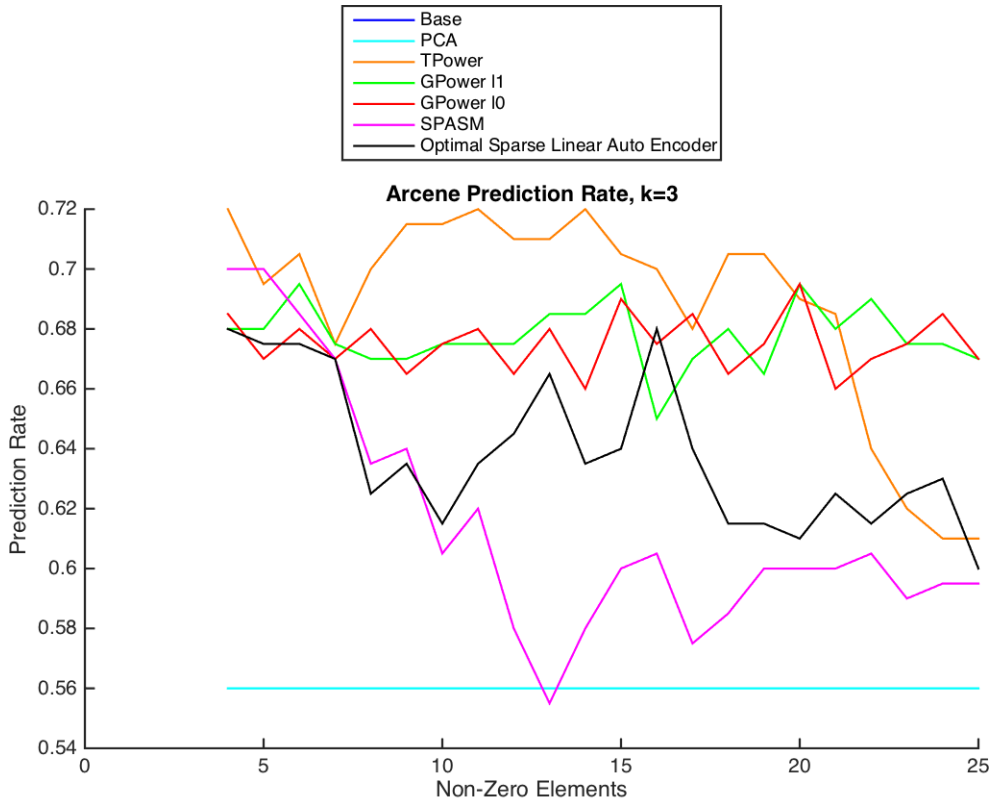

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Arcene Variance, k=1
Variance
Non-Zero Elements
0.05
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0
5
10
15
20
25

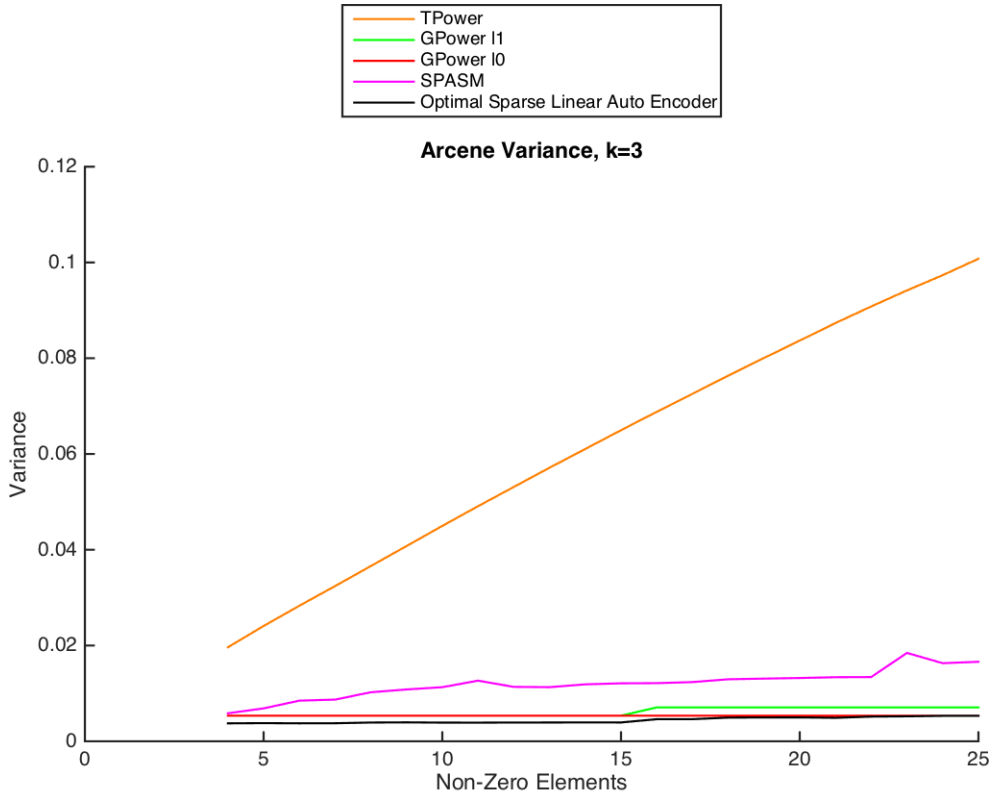

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Arcene Info Loss, k=2
Info Loss
Non-Zero Elements
2
1.9
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1
0
5
10
15
20
25

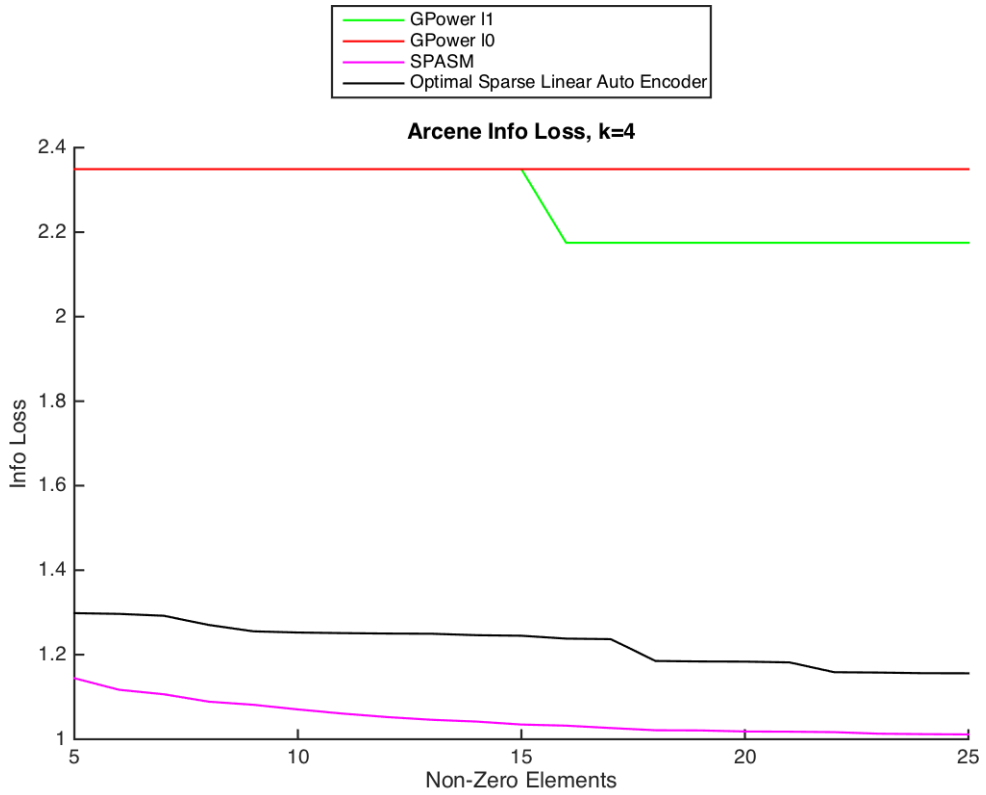

Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Arcene Prediction Rate, k=2
Prediction Rate
Non-Zero Elements
0.72
0.7
0.68
0.66
0.64
0.62
0.6
0.58
0.56
0
5
10
15
20
25

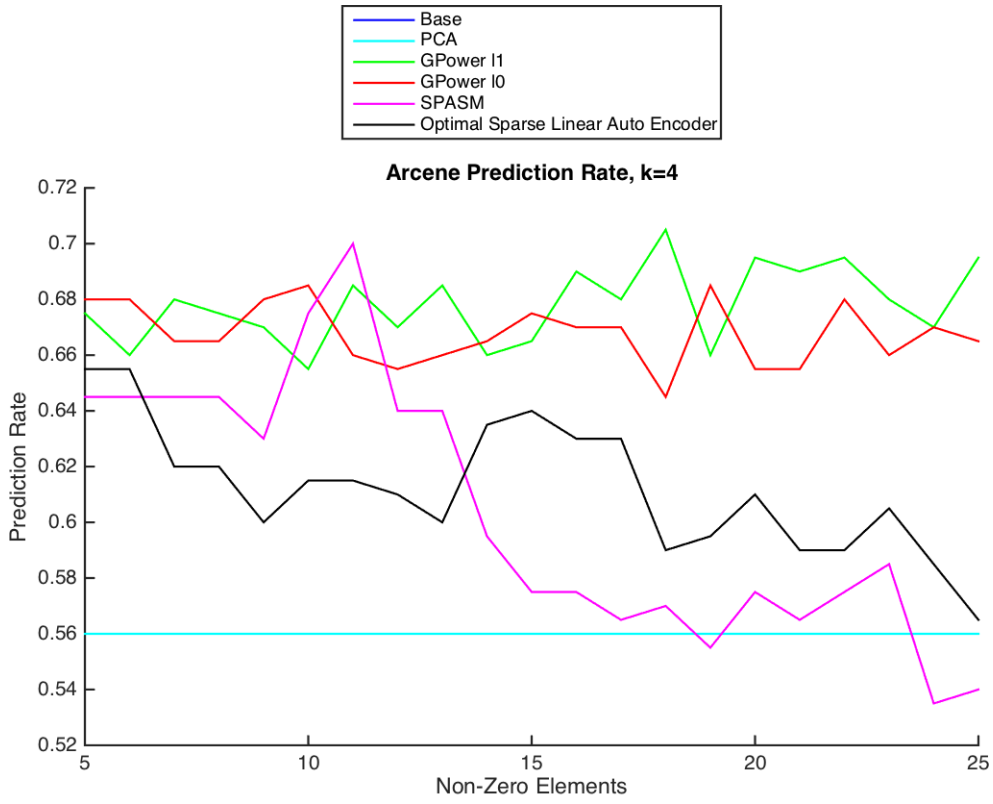

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Arcene Variance, k=2
Variance
Non-Zero Elements
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0
5
10
15
20
25

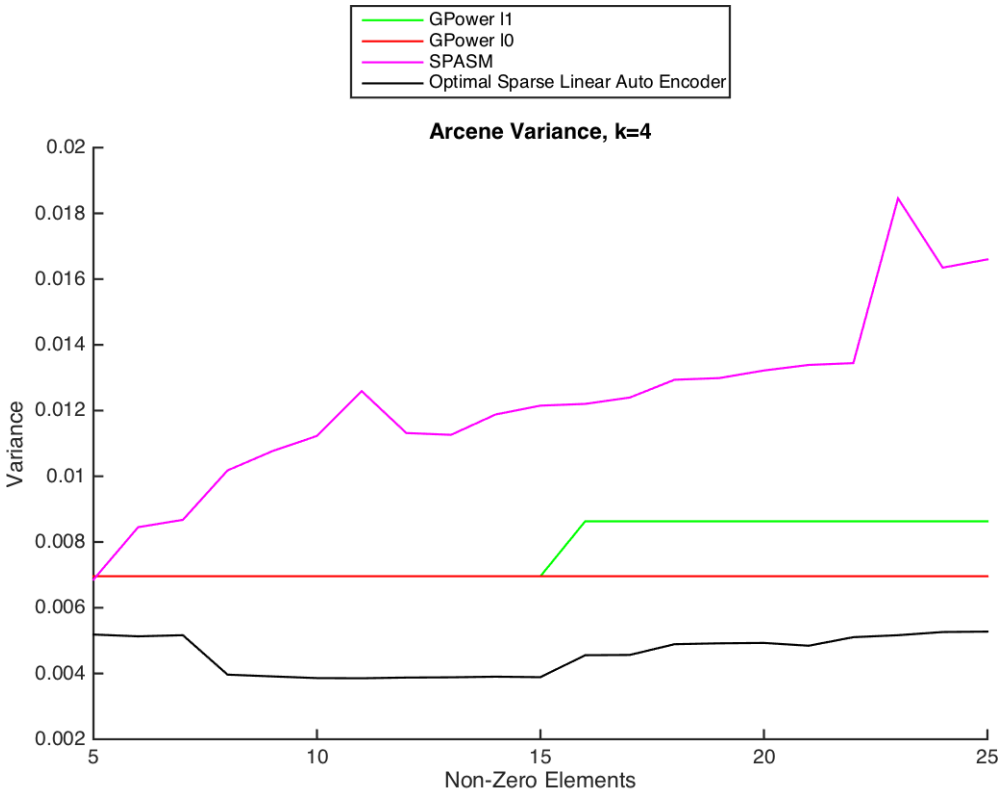

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Arcene Info Loss, k=3
Info Loss
Non-Zero Elements
1
1.5
2
2.5
0
5
10
15
20
25

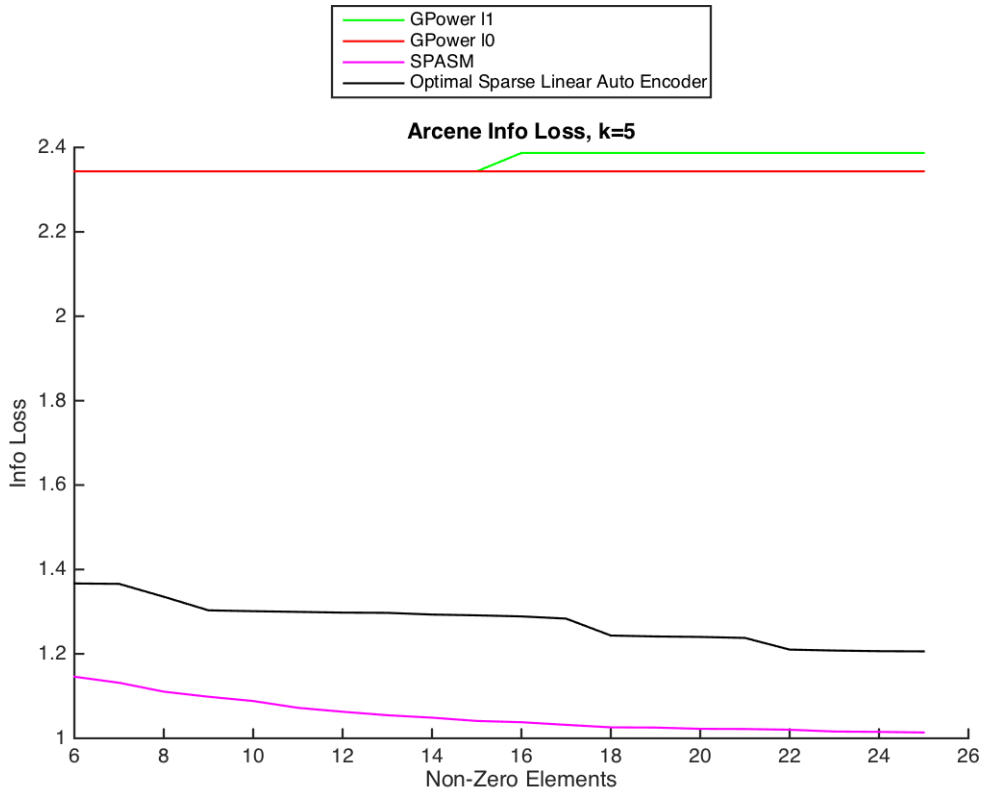

Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Arcene Prediction Rate, k=3
Prediction Rate
Non-Zero Elements
0.72
0.7
0.68
0.66
0.64
0.62
0.6
0.58
0.56
0.54
0
5
10
15
20
25

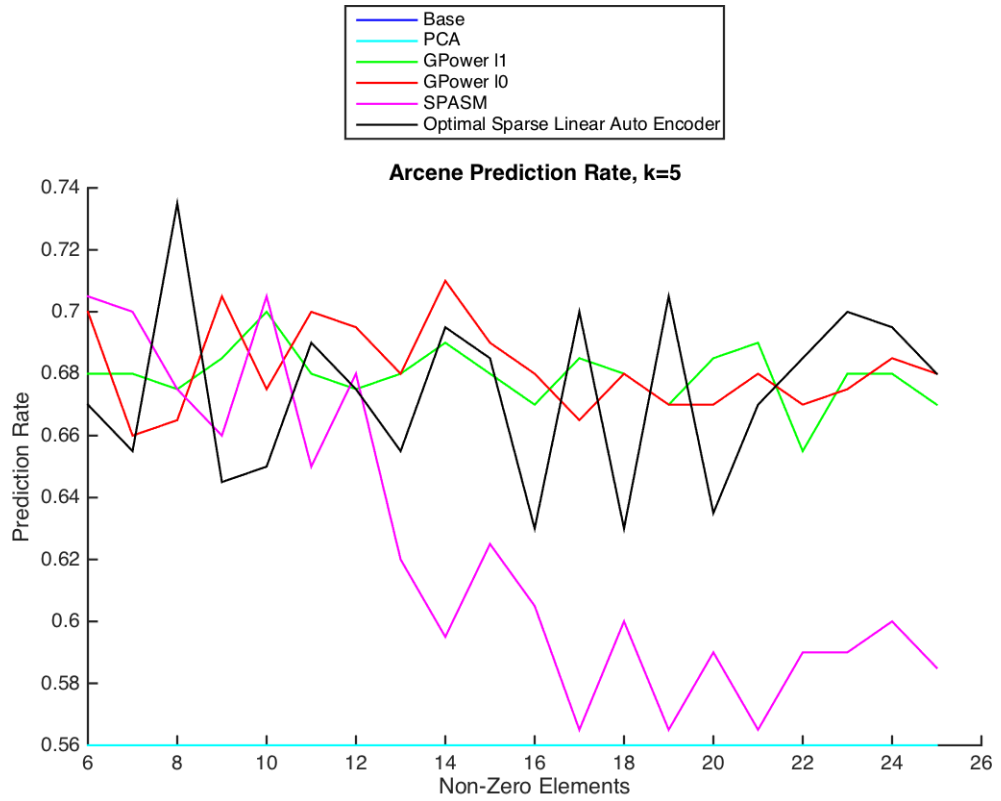

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Arcene Variance, k=3
Variance
Non-Zero Elements
0.12
0.1
0.08
0.06
0.04
0.02
0
0
5
10
15
20
25

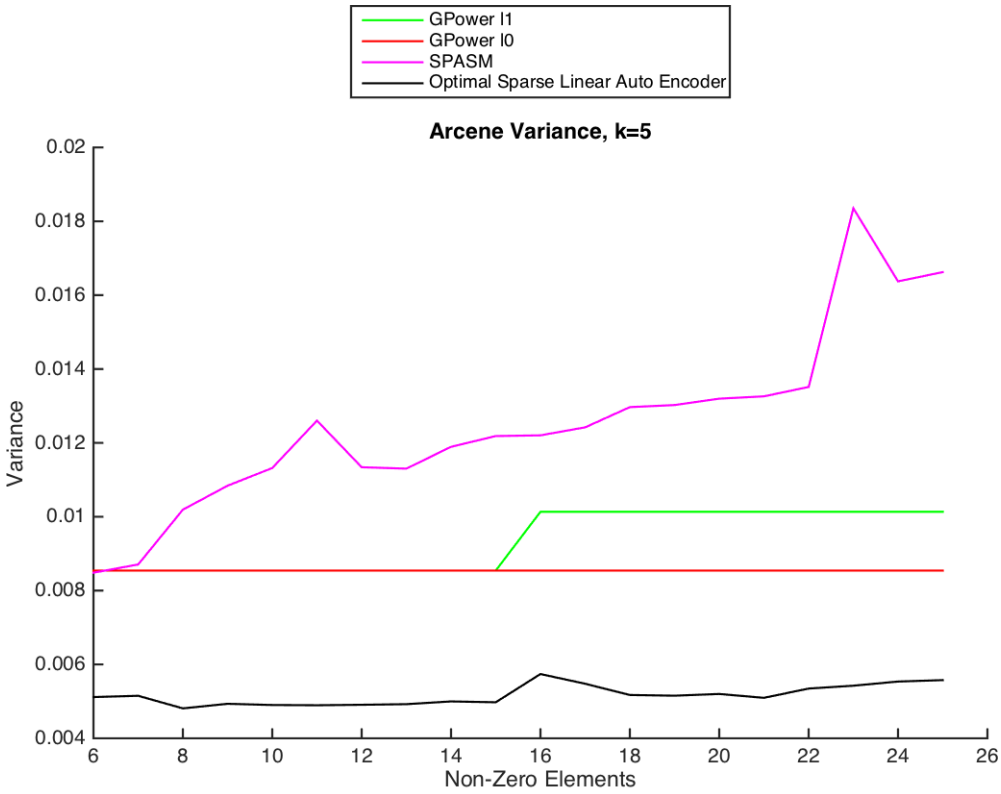

GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Arcene Info Loss, k=4
Info Loss
Non-Zero Elements
2.4
2.2
2
1.8
1.6
1.4
1.2
1
5
10
15
20
25

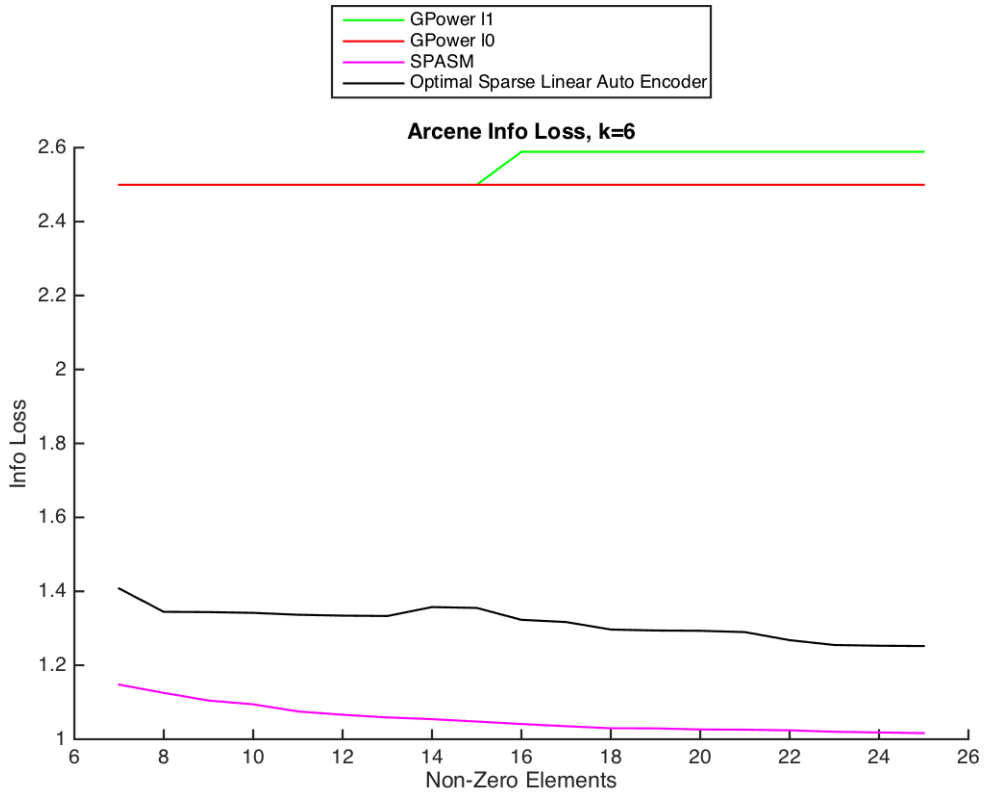

Base
PCA
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Arcene Prediction Rate, k=4
Prediction Rate
Non-Zero Elements
0.72
0.7
0.68
0.66
0.64
0.62
0.6
0.58
0.56
0.54
0.52
5
10
15
20
25

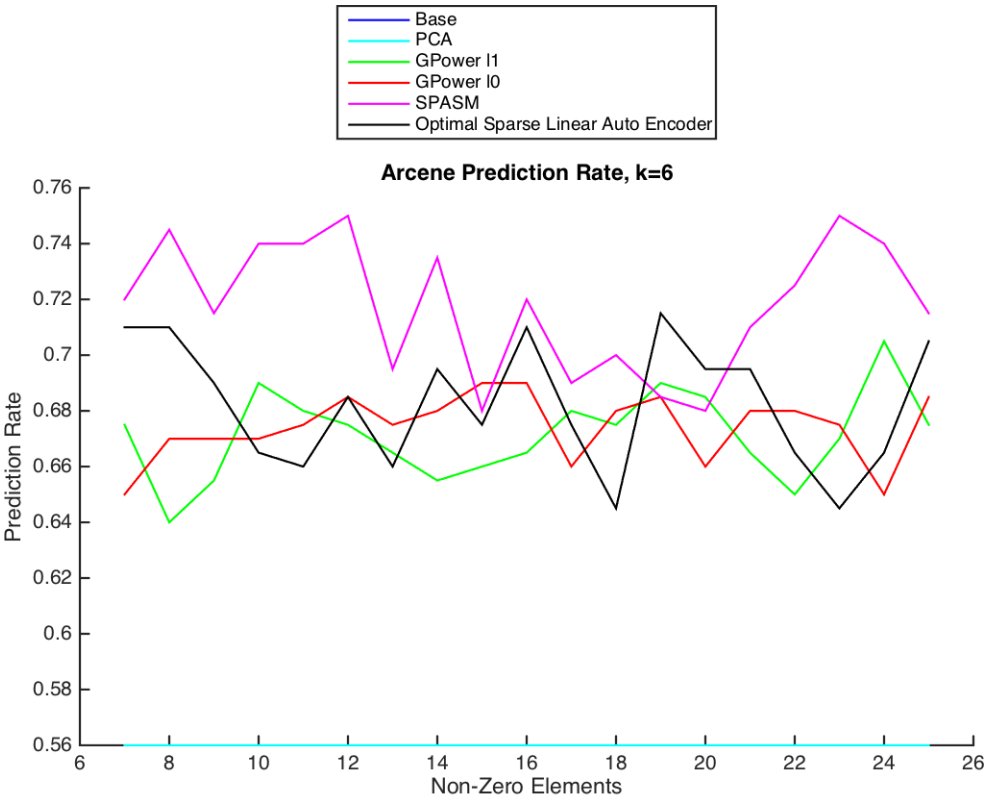

GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Arcene Variance, k=4
Variance
Non-Zero Elements
0.02
0.018
0.016
0.014
0.012
0.01
0.008
0.006
0.004
0.002
5
10
15
20
25

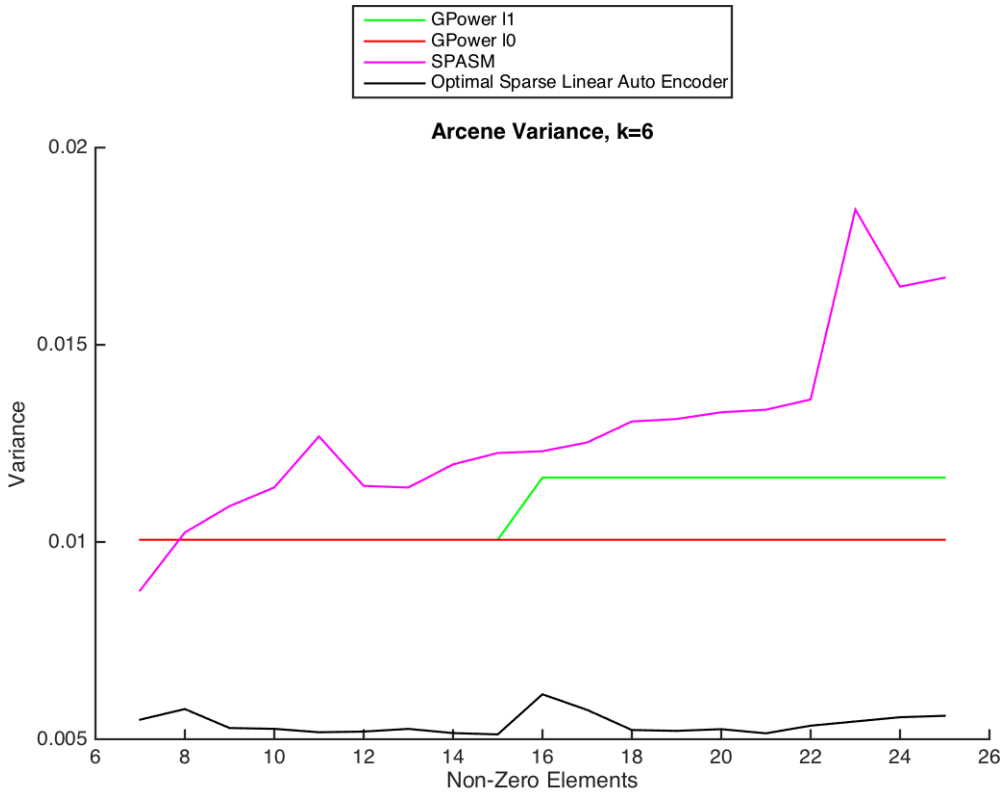

GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Arcene Info Loss, k=5
Info Loss
Non-Zero Elements
2.4
2.2
2
1.8
1.6
1.4
1.2
1
6
8
10
12
14
16
18
20
22
24
26


Base
PCA
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Arcene Prediction Rate, k=5
Prediction Rate
Non-Zero Elements
0.74
0.72
0.7
0.68
0.66
0.64
0.62
0.6
0.58
0.56
6
8
10
12
14
16
18
20
22
24
26

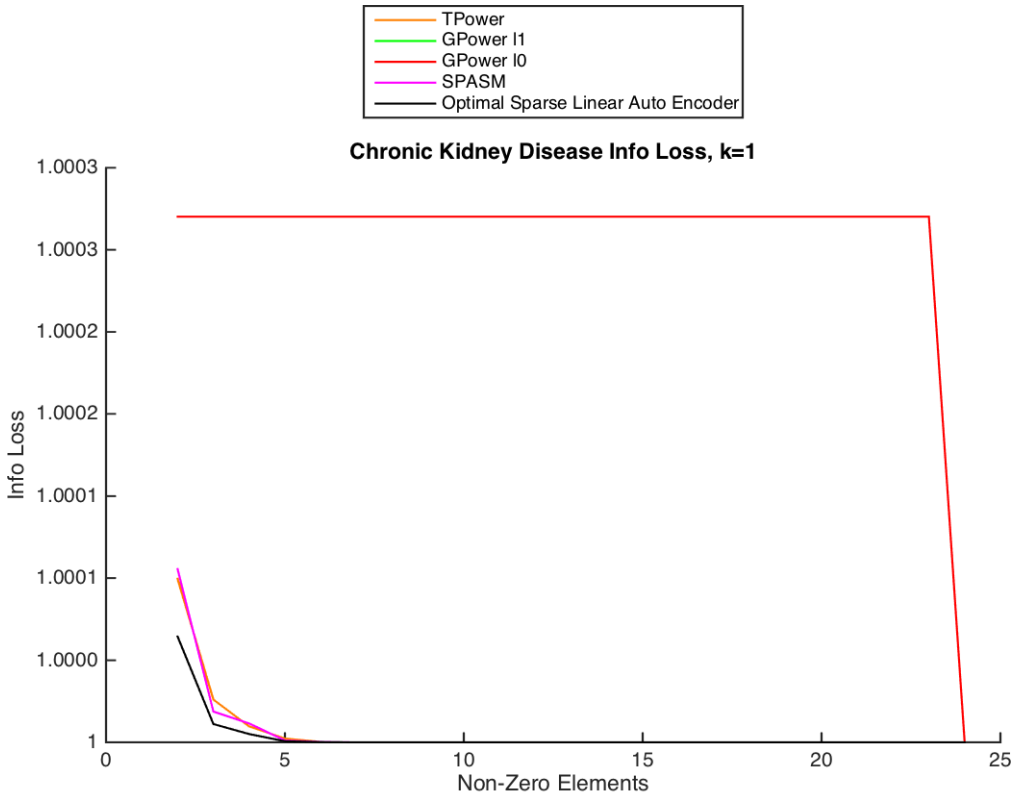

GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Arcene Variance, k=5
Variance
Non-Zero Elements
0.02
0.018
0.016
0.014
0.012
0.01
0.008
0.006
0.004
6
8
10
12
14
16
18
20
22
24
26

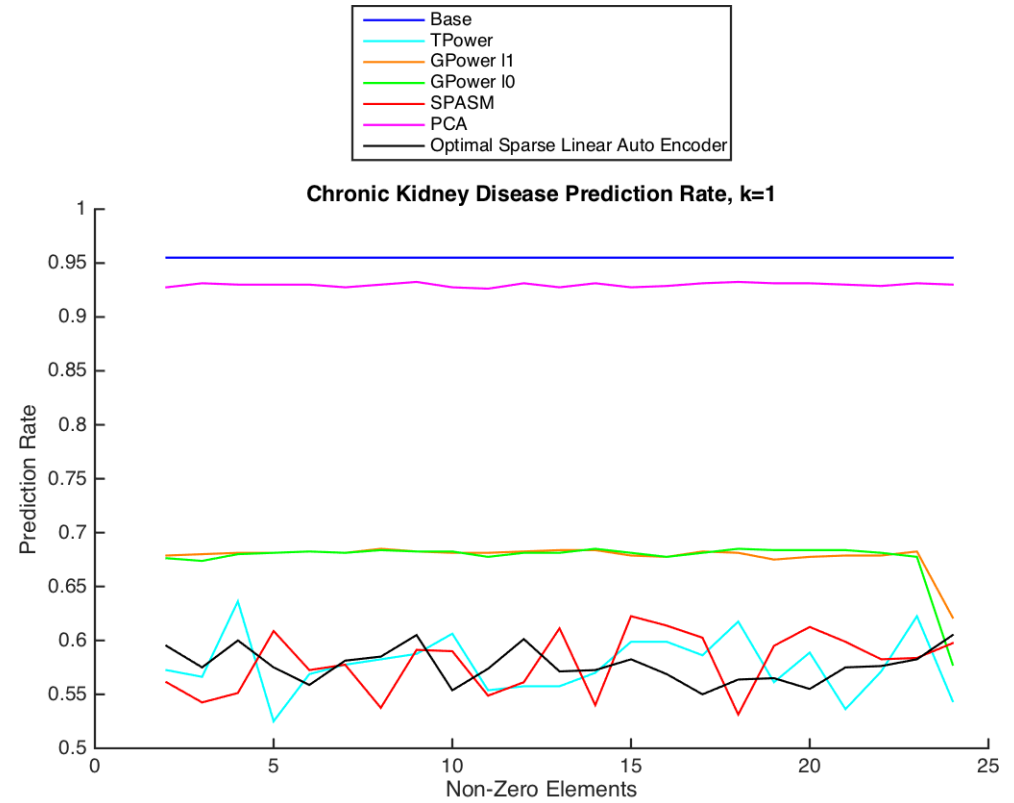

GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Arcene Info Loss, k=6
Info Loss
Non-Zero Elements
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
1
6
8
10
12
14
16
18
20
22
24
26

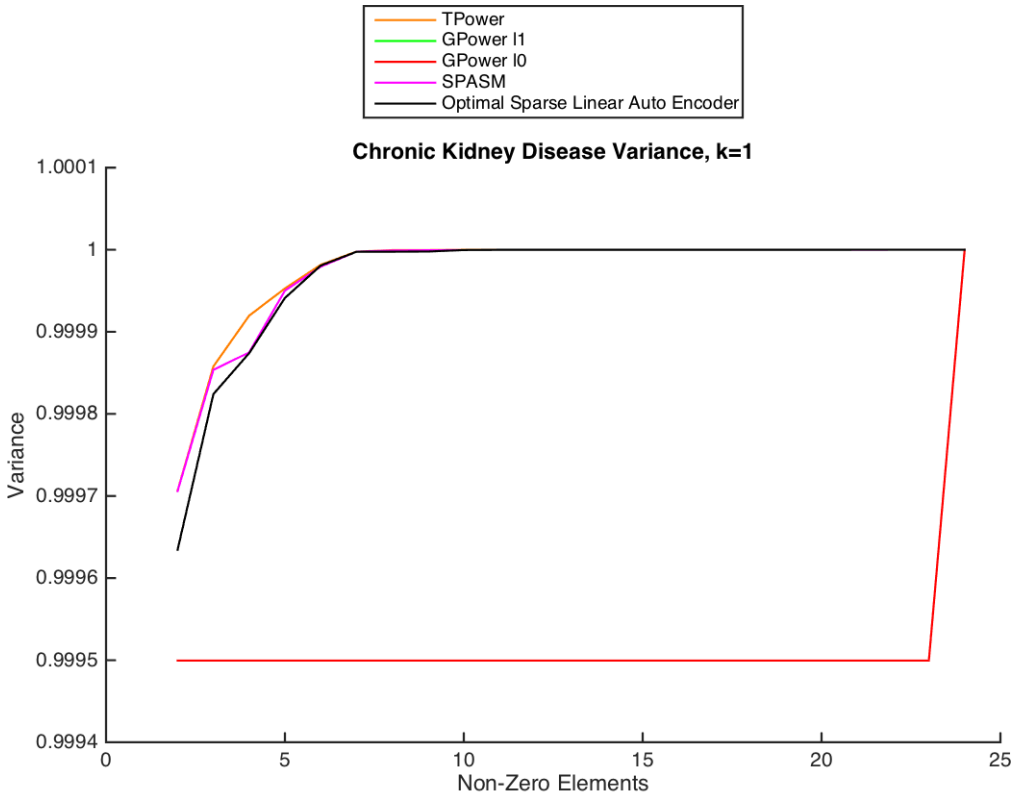

Base
PCA
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Arcene Prediction Rate, k=6
Prediction Rate
Non-Zero Elements
0.76
0.74
0.72
0.7
0.68
0.66
0.64
0.62
0.6
0.58
0.56
6
8
10
12
14
16
18
20
22
24
26

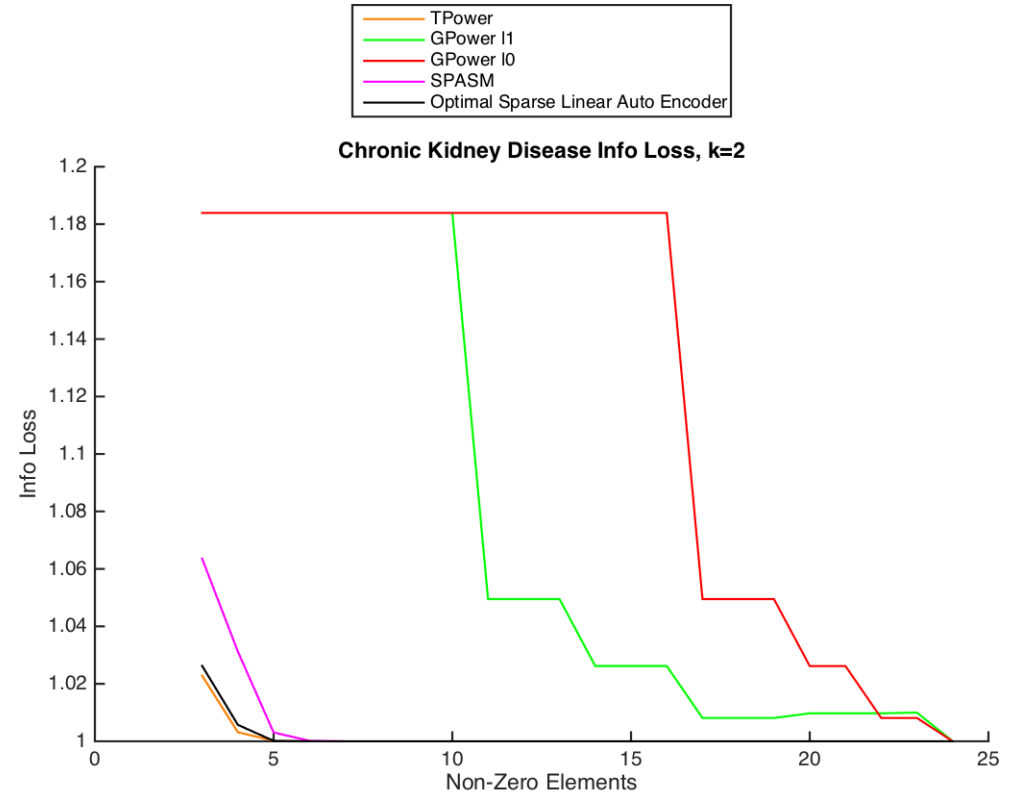

GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Arcene Variance, k=6
0.02
0.015
0.01
0.005
Variance
6
8
10
12
14
16
18
20
22
24
26
Non-Zero Elements

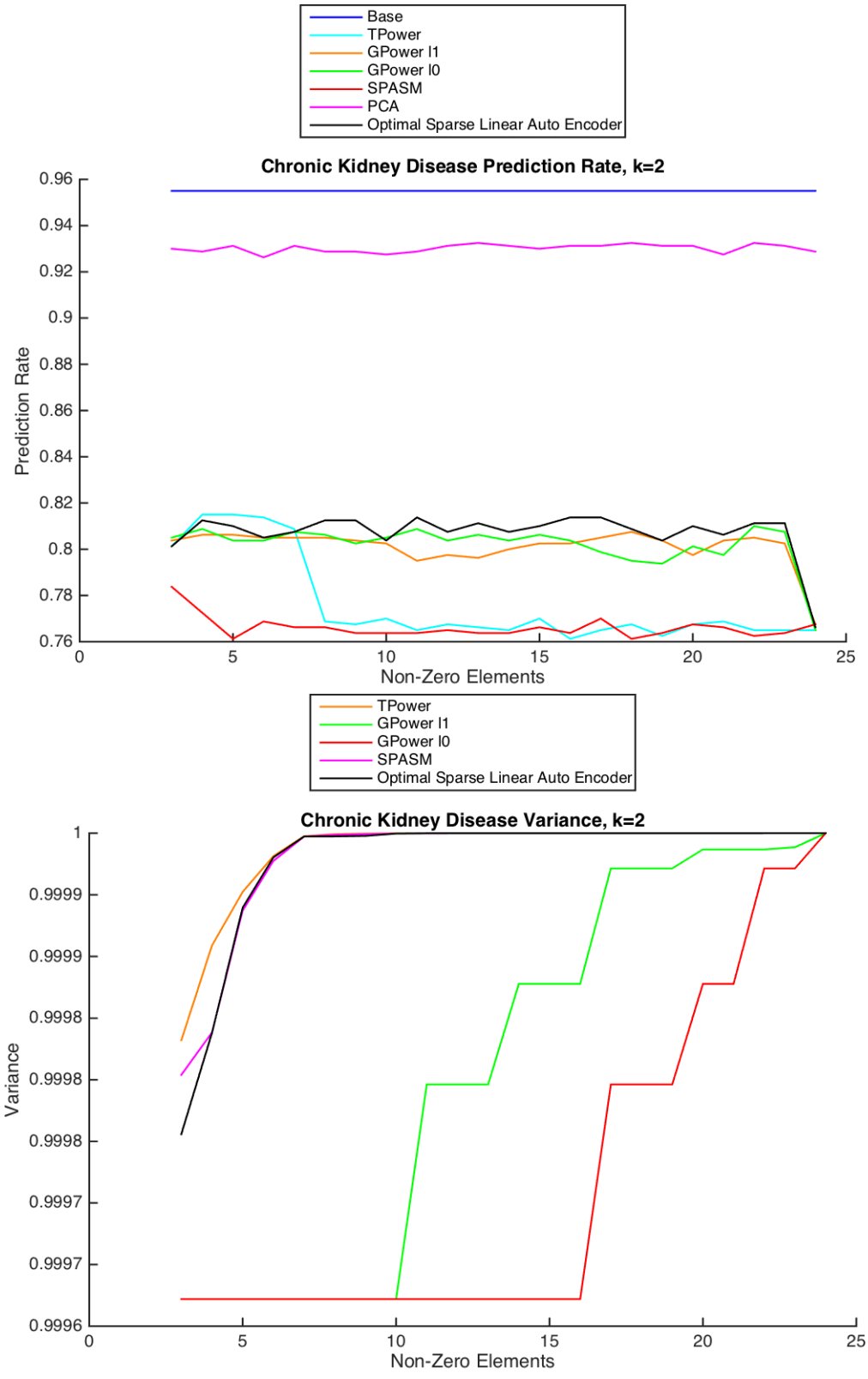
A.2 Chronic Kidney Disease Results

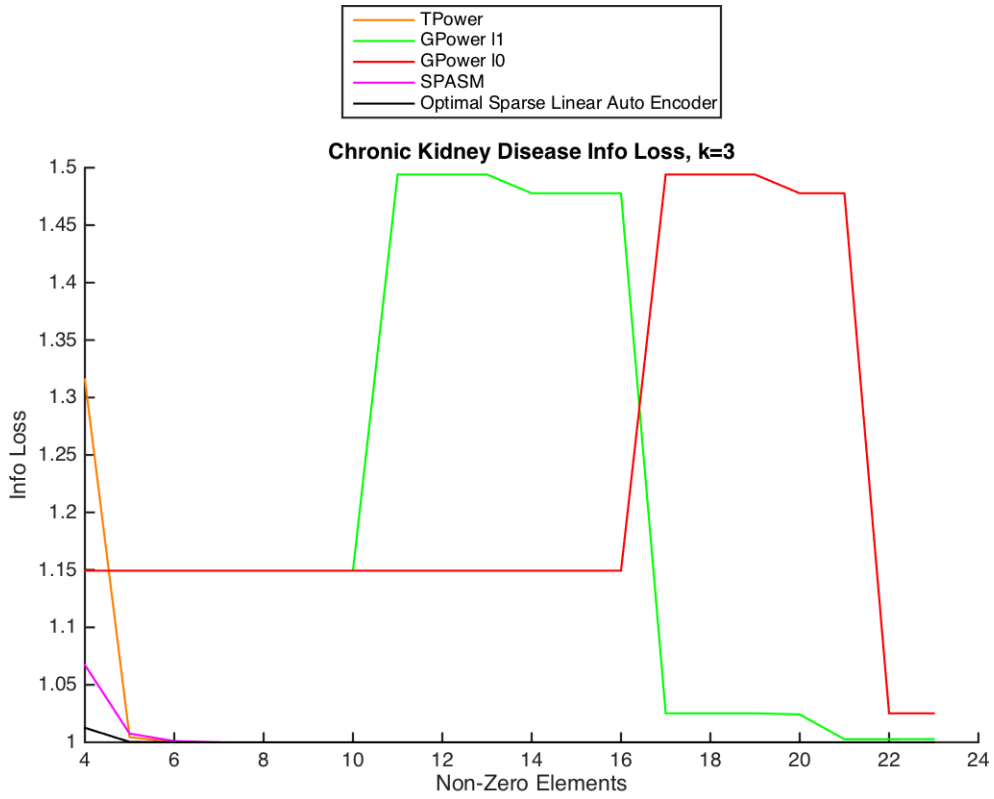


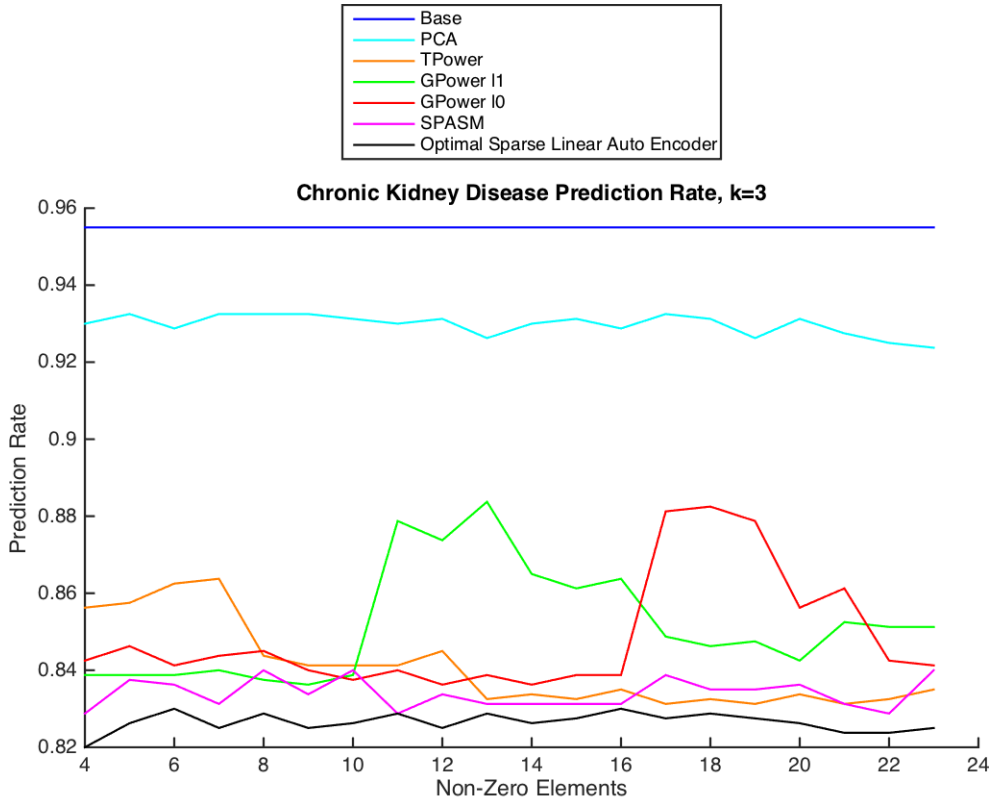


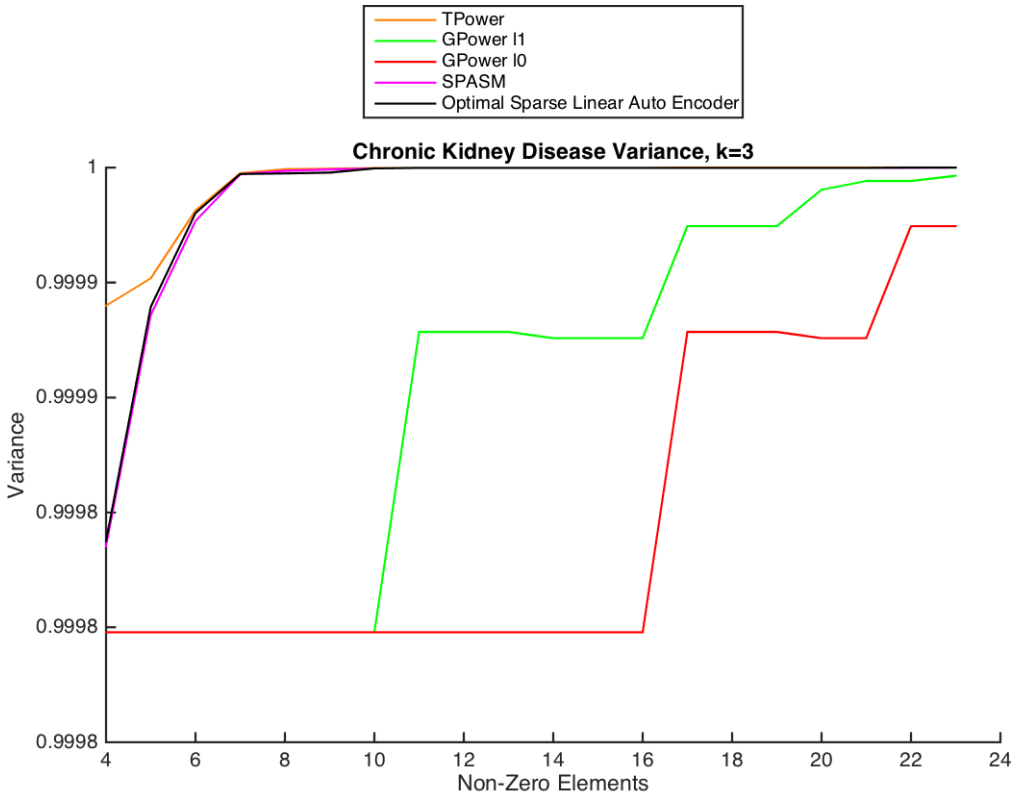

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Chronic Kidney Disease Variance, k=1
Variance
Non-Zero Elements
1.0001
1
0.9999
0.9998
0.9997
0.9996
0.9995
0.9994
0
5
10
15
20
25

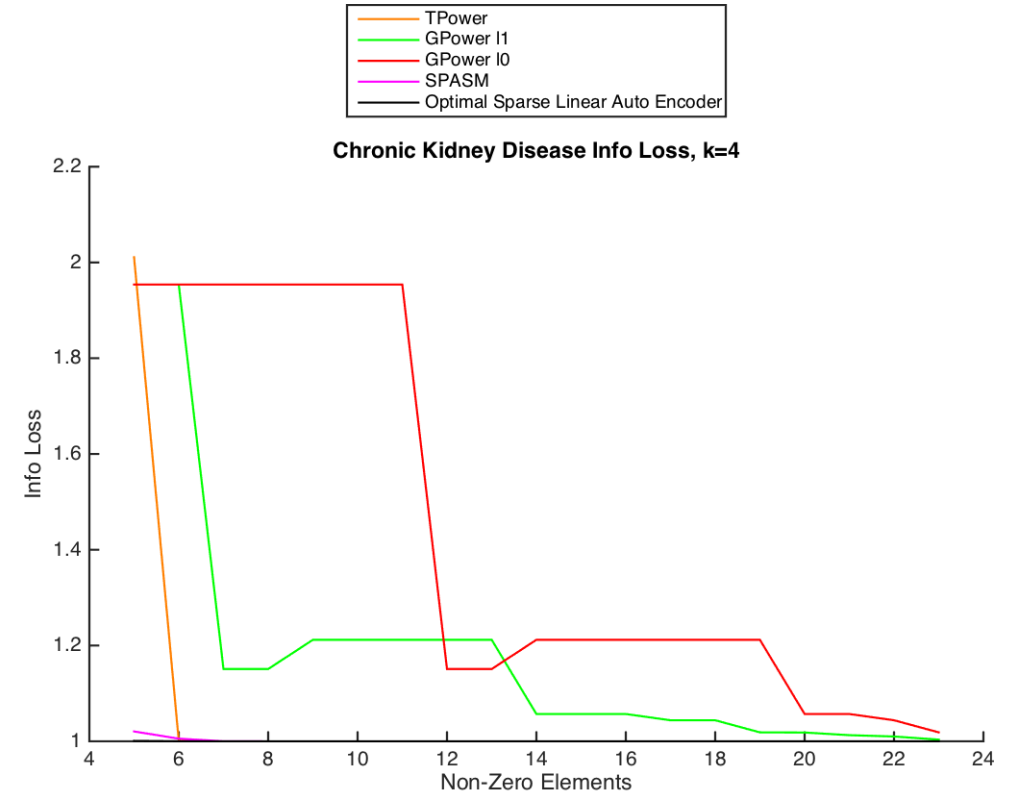

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Chronic Kidney Disease Info Loss, k=2
Info Loss
Non-Zero Elements
1.2
1.18
1.16
1.14
1.12
1.1
1.08
1.06
1.04
1.02
1
0
5
10
15
20
25

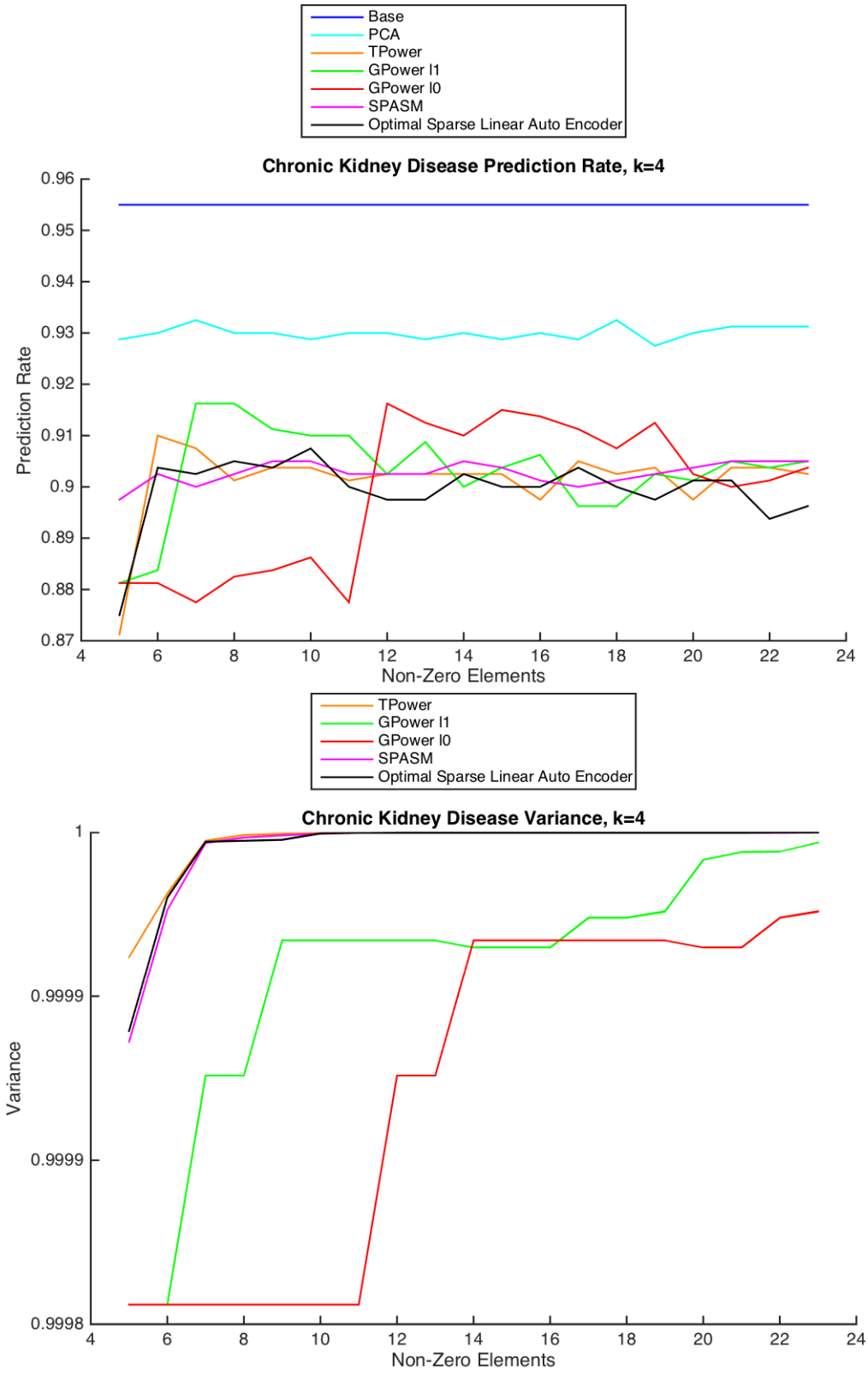

Base
TPower
GPower l1
GPower l0
SPASM
PCA
Optimal Sparse Linear Auto Encoder
Chronic Kidney Disease Prediction Rate, k=2
Prediction Rate
Non-Zero Elements
0.96
0.94
0.92
0.9
0.88
0.86
0.84
0.82
0.8
0.78
0.76
0
5
10
15
20
25
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Chronic Kidney Disease Variance, k=2
Variance
Non-Zero Elements
1
0.9999
0.9999
0.9998
0.9998
0.9998
0.9997
0.9997
0.9996
0
5
10
15
20
25

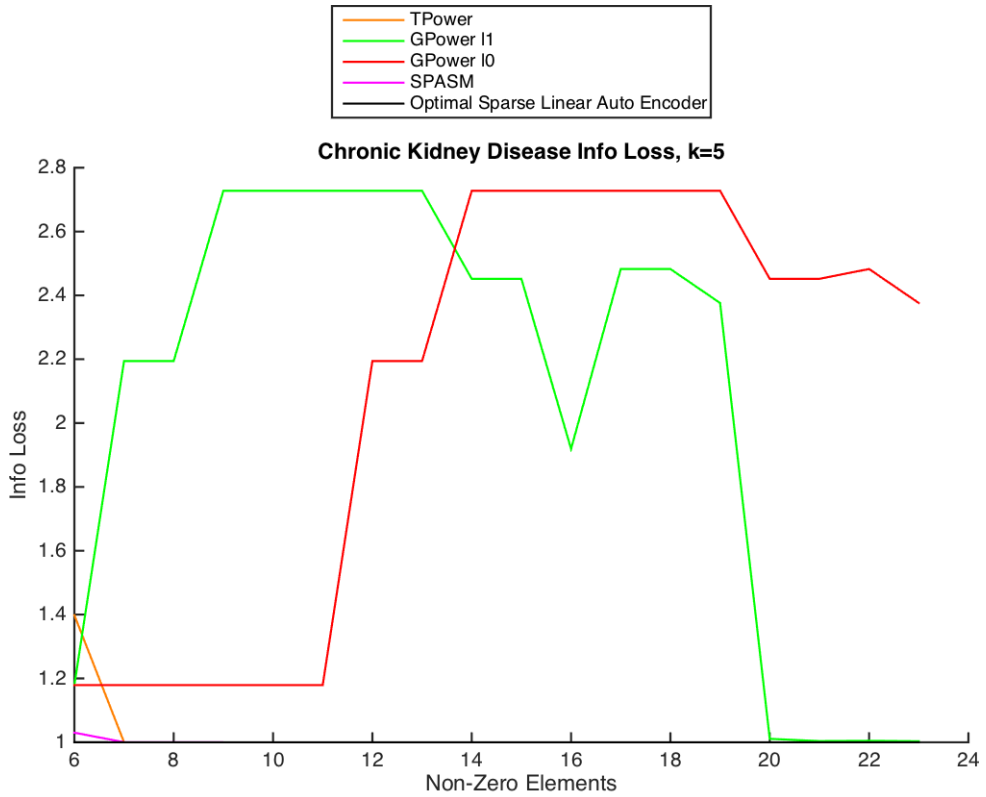

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Chronic Kidney Disease Info Loss, k=3
Info Loss
Non-Zero Elements
1.5
1.45
1.4
1.35
1.3
1.25
1.2
1.15
1.1
1.05
1
4
6
8
10
12
14
16
18
20
22
24

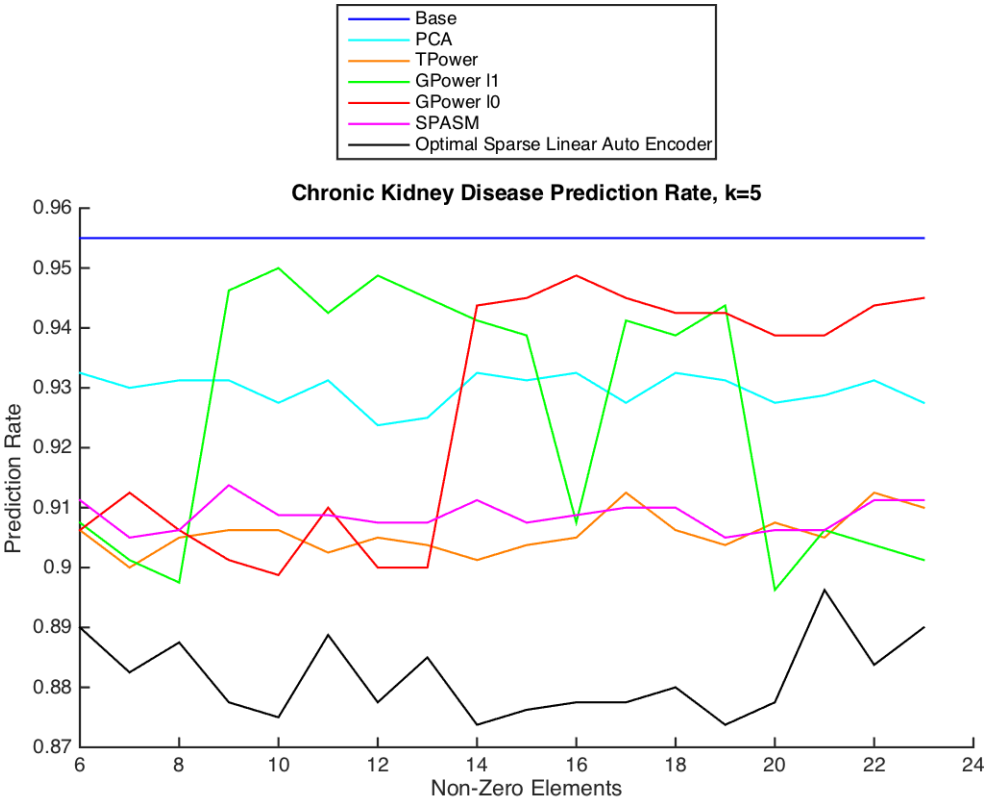

Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Chronic Kidney Disease Prediction Rate, k=3
Prediction Rate
Non-Zero Elements
0.96
0.94
0.92
0.9
0.88
0.86
0.84
0.82
4
6
8
10
12
14
16
18
20
22
24

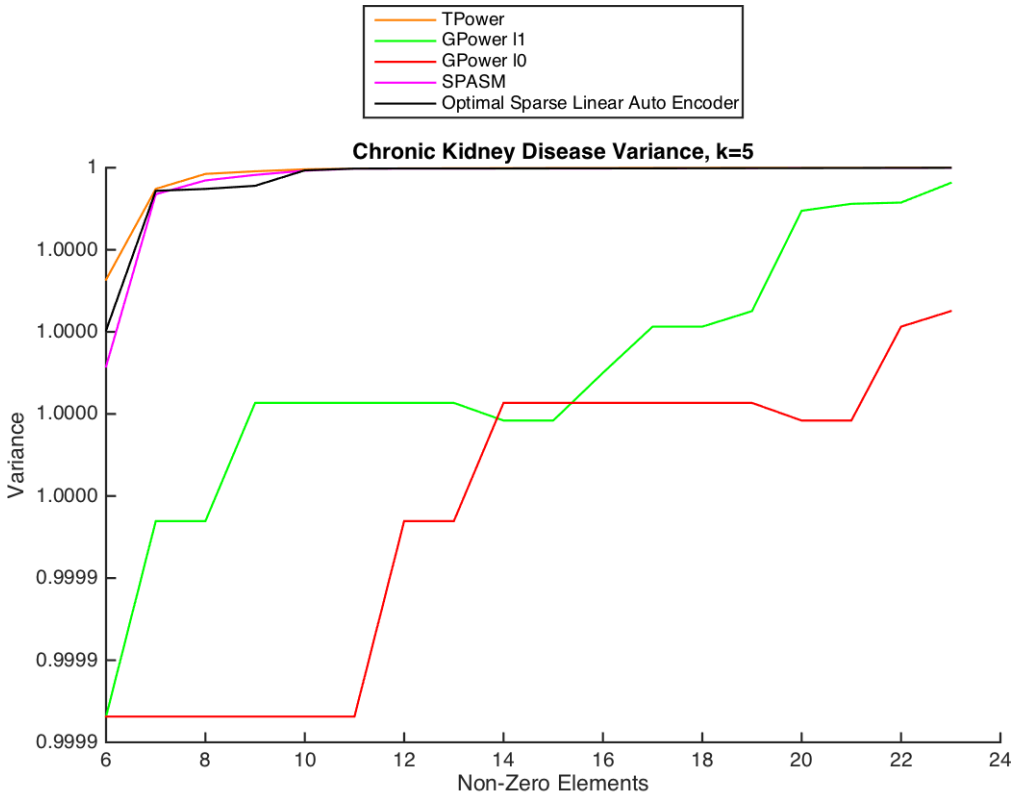

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Chronic Kidney Disease Variance, k=3
Variance
Non-Zero Elements
1
0.9999
0.9999
0.9998
0.9998
0.9998
4
6
8
10
12
14
16
18
20
22
24

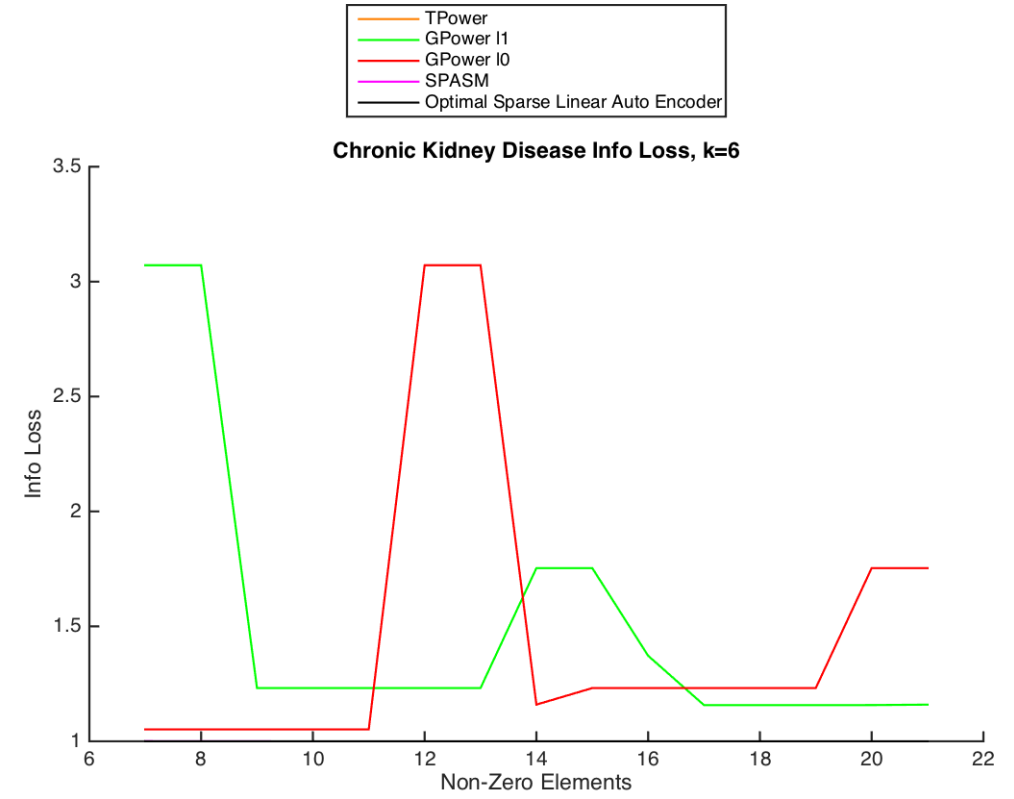

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Chronic Kidney Disease Info Loss, k=4
Info Loss
Non-Zero Elements
2.2
2
1.8
1.6
1.4
1.2
1
4
6
8
10
12
14
16
18
20
22
24

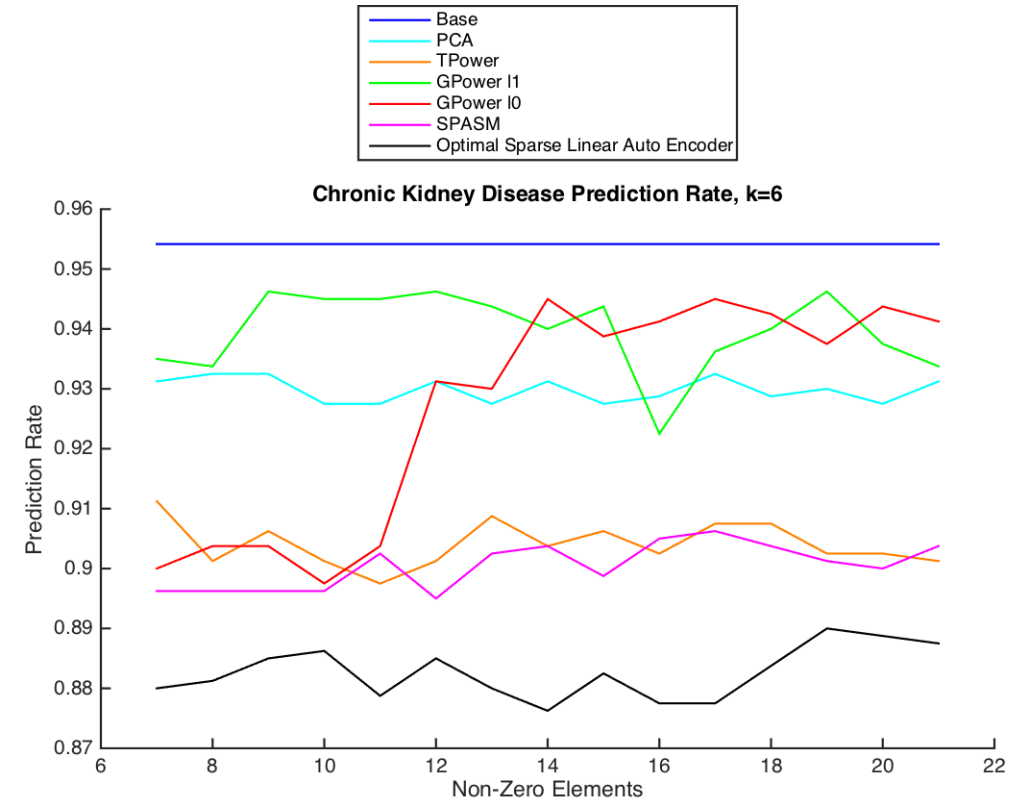

Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Chronic Kidney Disease Prediction Rate, k=4
Prediction Rate
Non-Zero Elements
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Chronic Kidney Disease Variance, k=4
Variance
Non-Zero Elements

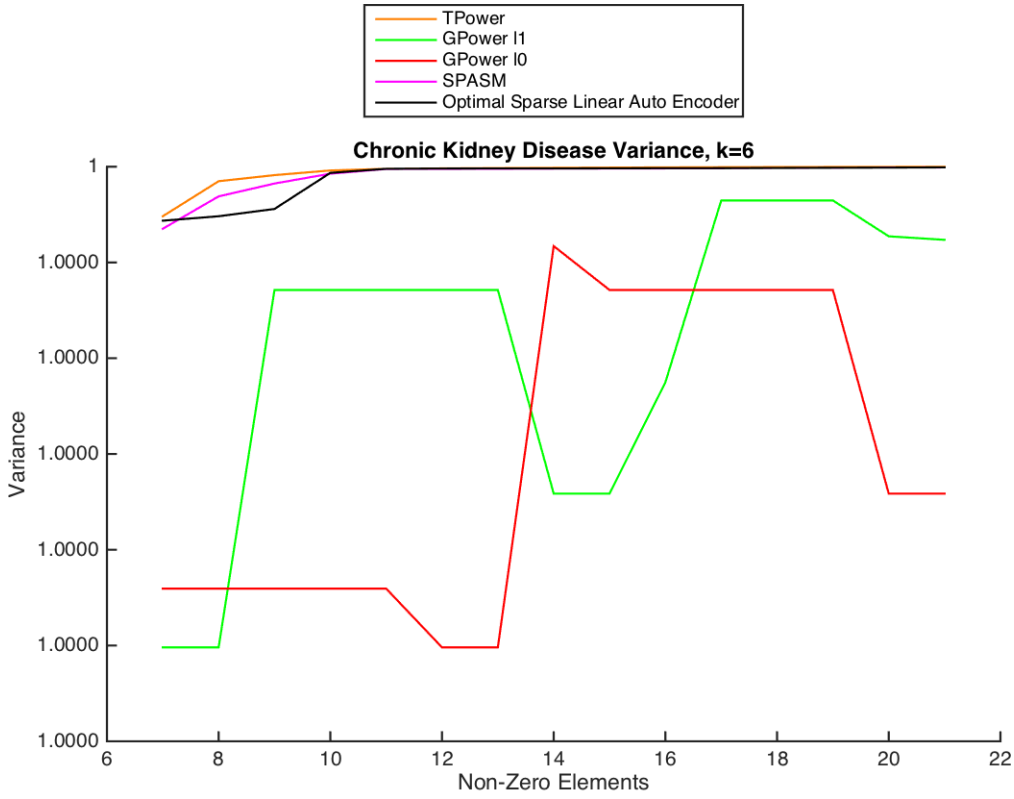

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Chronic Kidney Disease Info Loss, k=5
Info Loss
Non-Zero Elements
2.8
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
1
6
8
10
12
14
16
18
20
22
24


Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Chronic Kidney Disease Prediction Rate, k=5
Prediction Rate
Non-Zero Elements
0.96
0.95
0.94
0.93
0.92
0.91
0.9
0.89
0.88
0.87
6
8
10
12
14
16
18
20
22
24

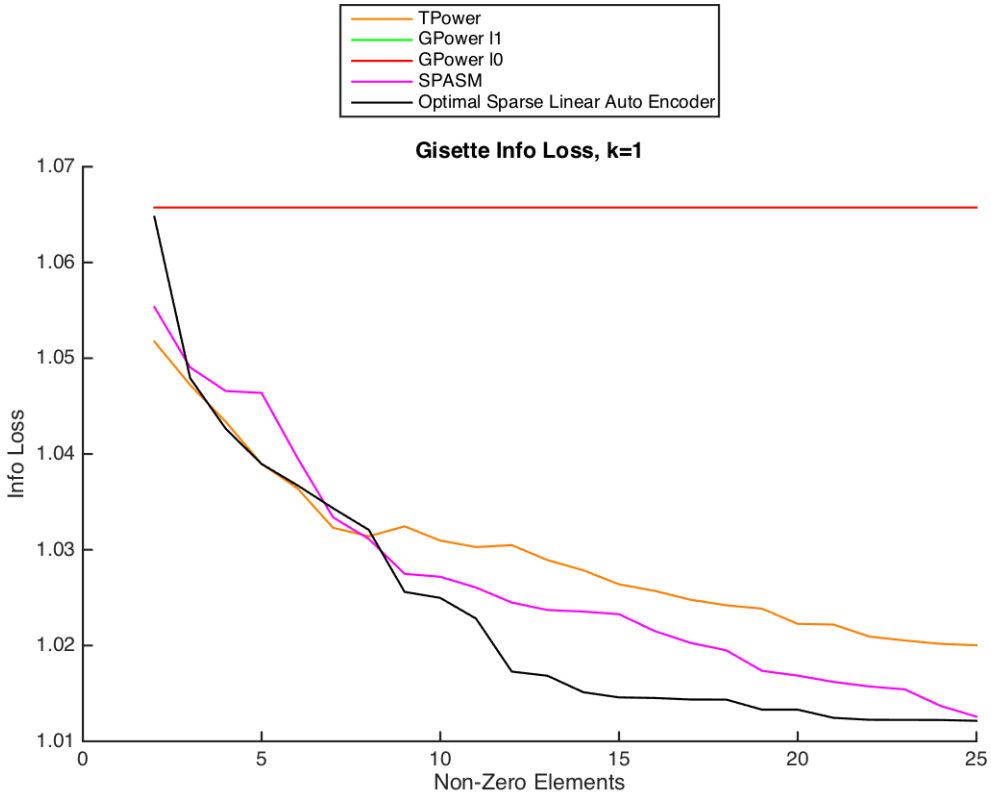

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Chronic Kidney Disease Variance, k=5
Variance
Non-Zero Elements
1
1.0000
1.0000
1.0000
1.0000
0.9999
0.9999
0.9999
6
8
10
12
14
16
18
20
22
24

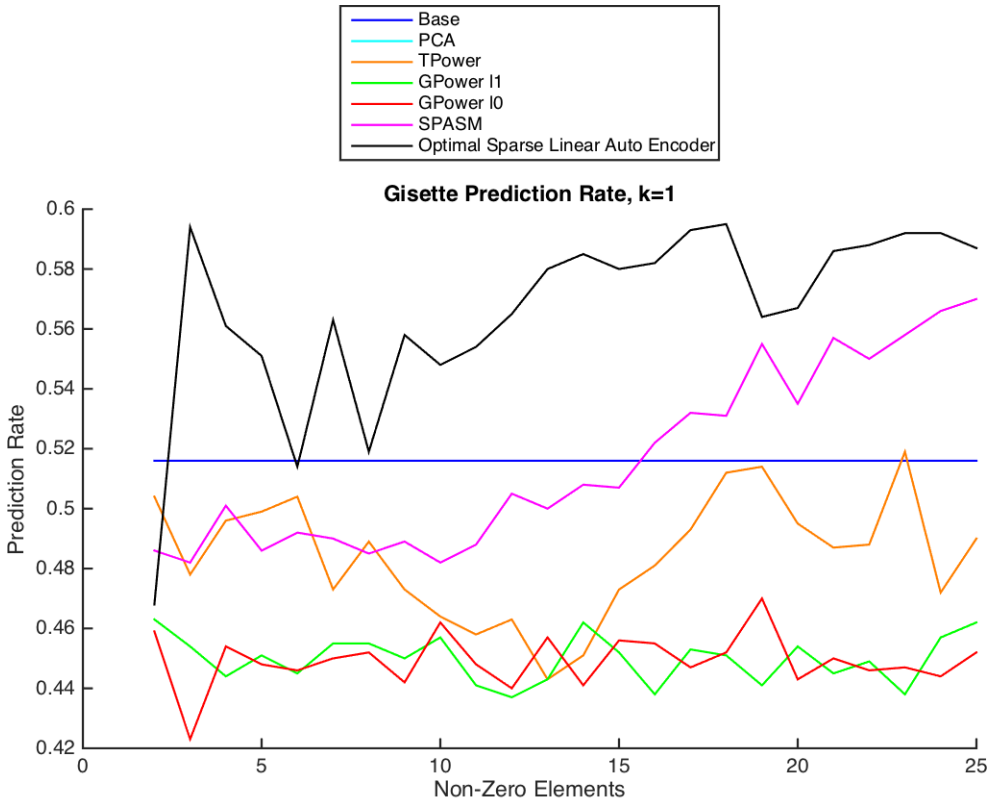

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Chronic Kidney Disease Info Loss, k=6
Info Loss
Non-Zero Elements
3.5
3
2.5
2
1.5
1
6
8
10
12
14
16
18
20
22

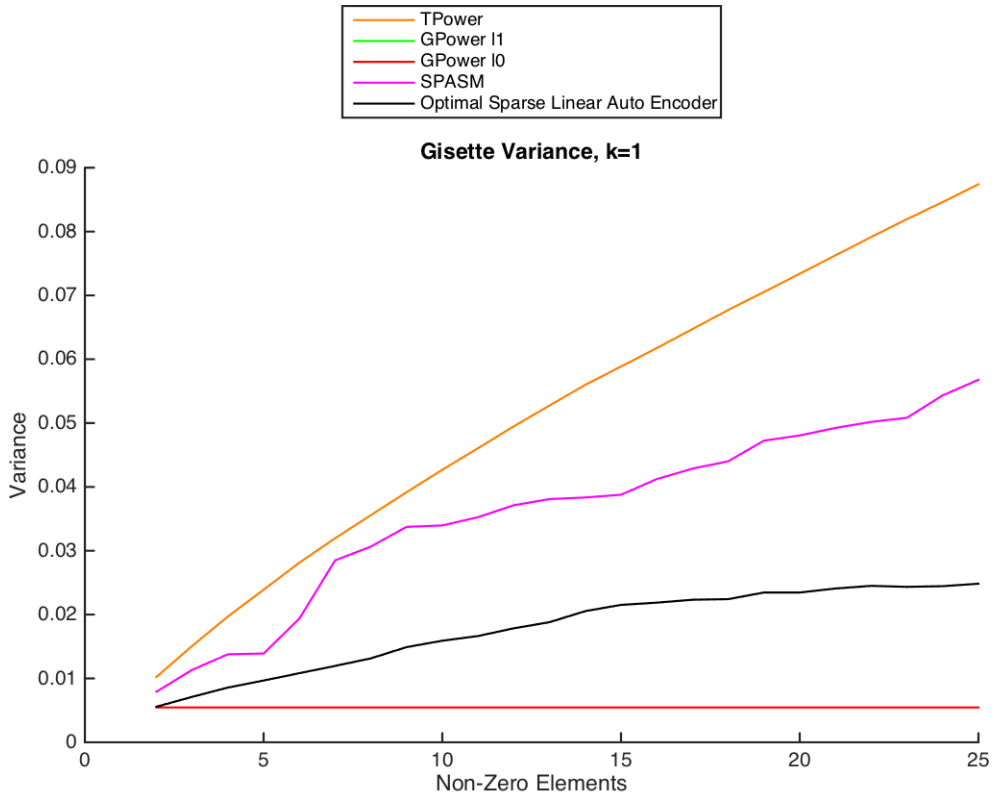

Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Chronic Kidney Disease Prediction Rate, k=6
Prediction Rate
Non-Zero Elements
0.96
0.95
0.94
0.93
0.92
0.91
0.9
0.89
0.88
0.87
6
8
10
12
14
16
18
20
22

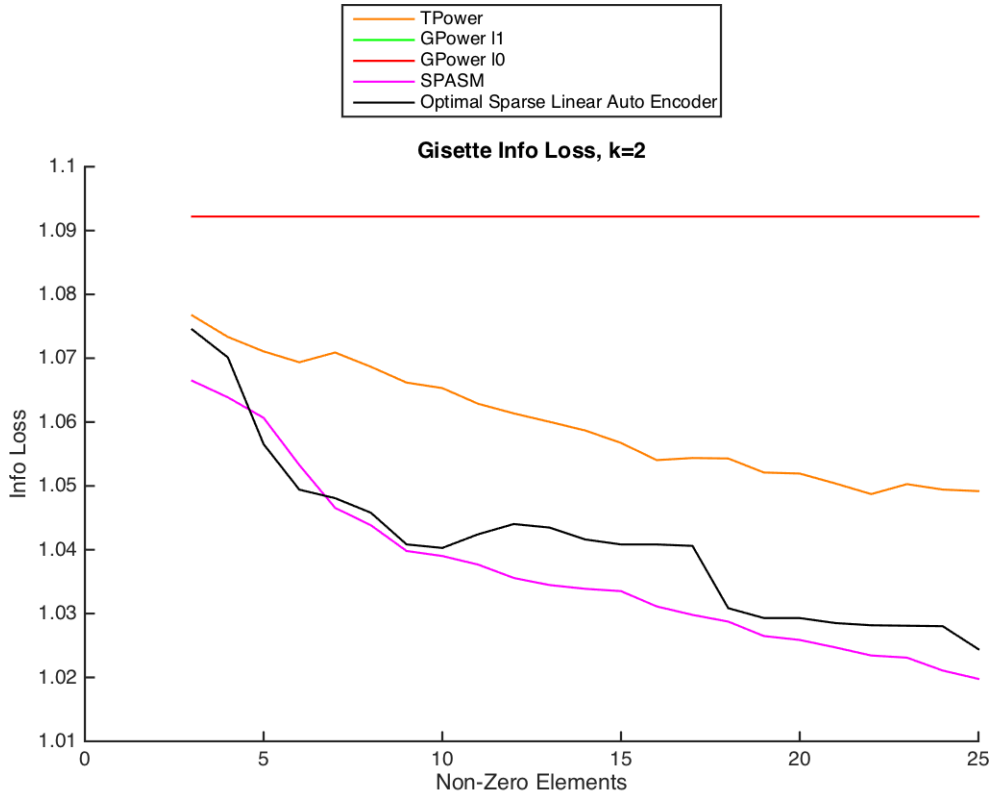

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Chronic Kidney Disease Variance, k=6
Variance
Non-Zero Elements
1
1.0000
1.0000
1.0000
1.0000
1.0000
6
8
10
12
14
16
18
20
22

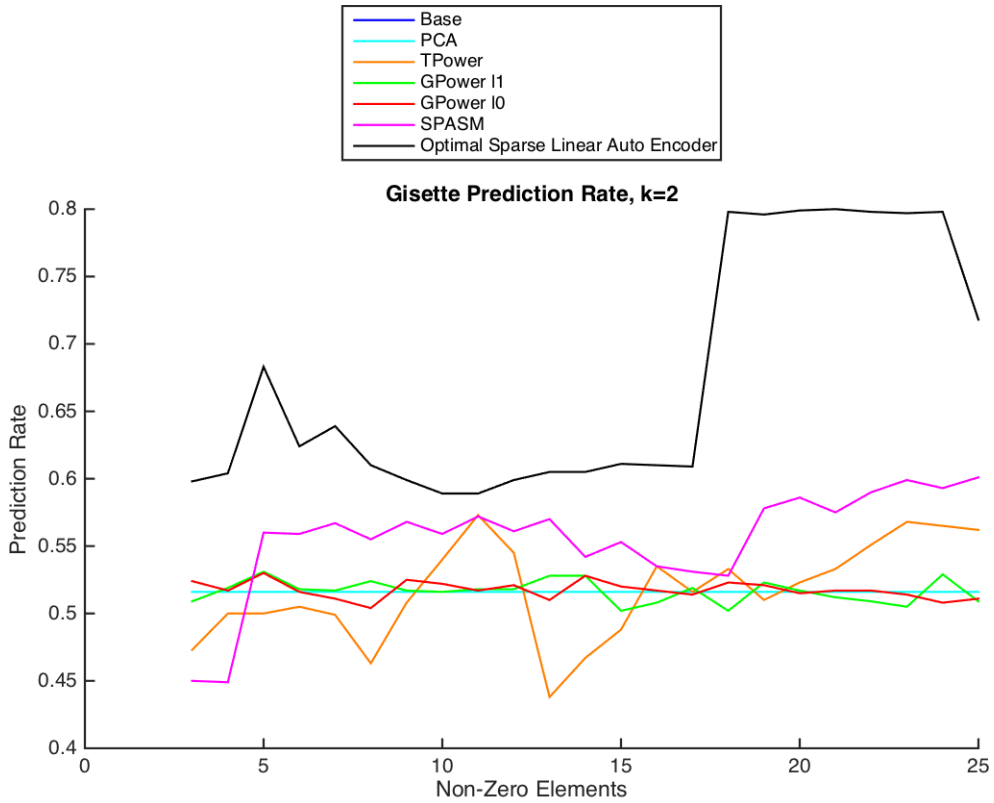
A.3 Gisette Results

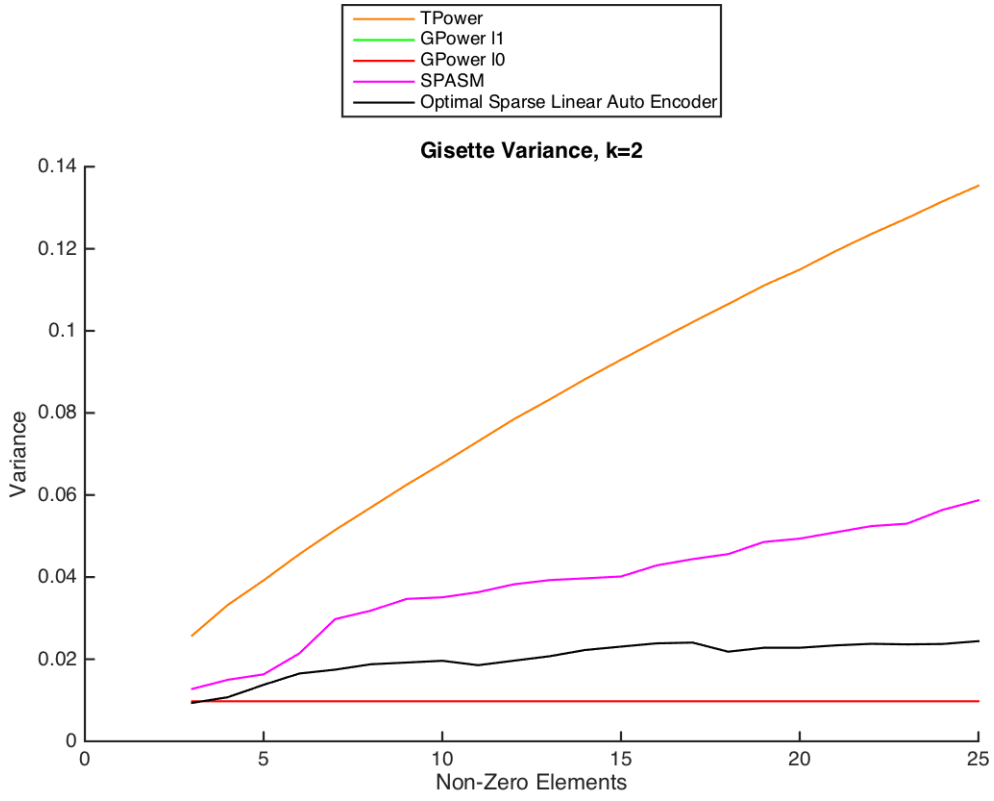


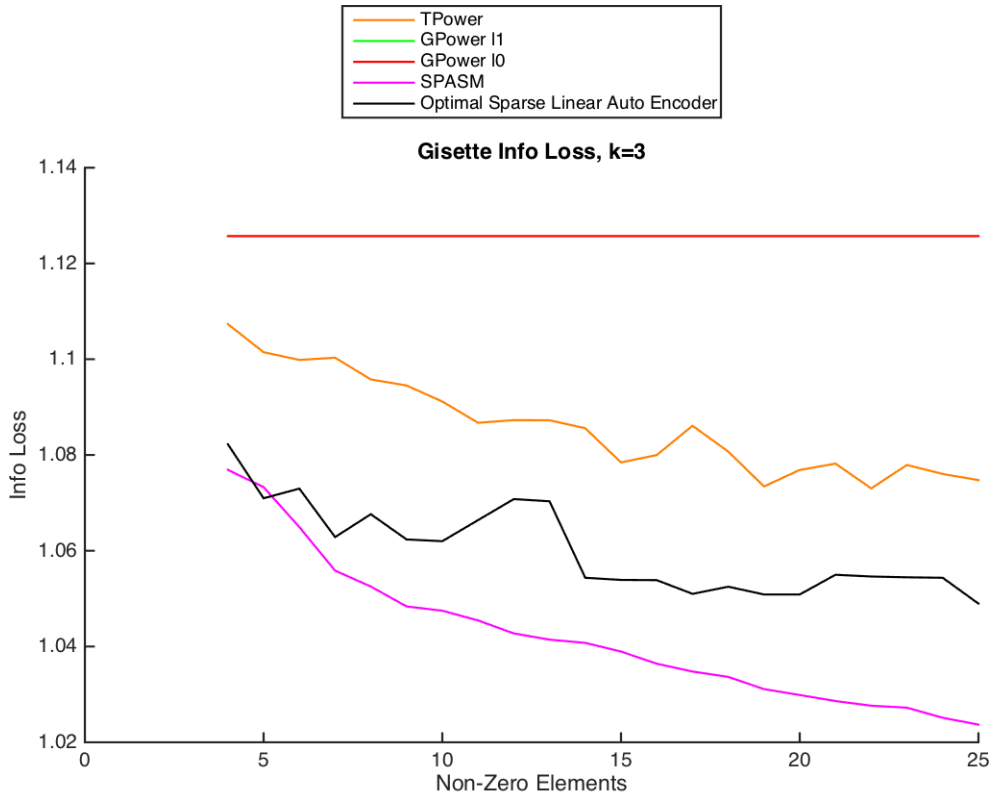


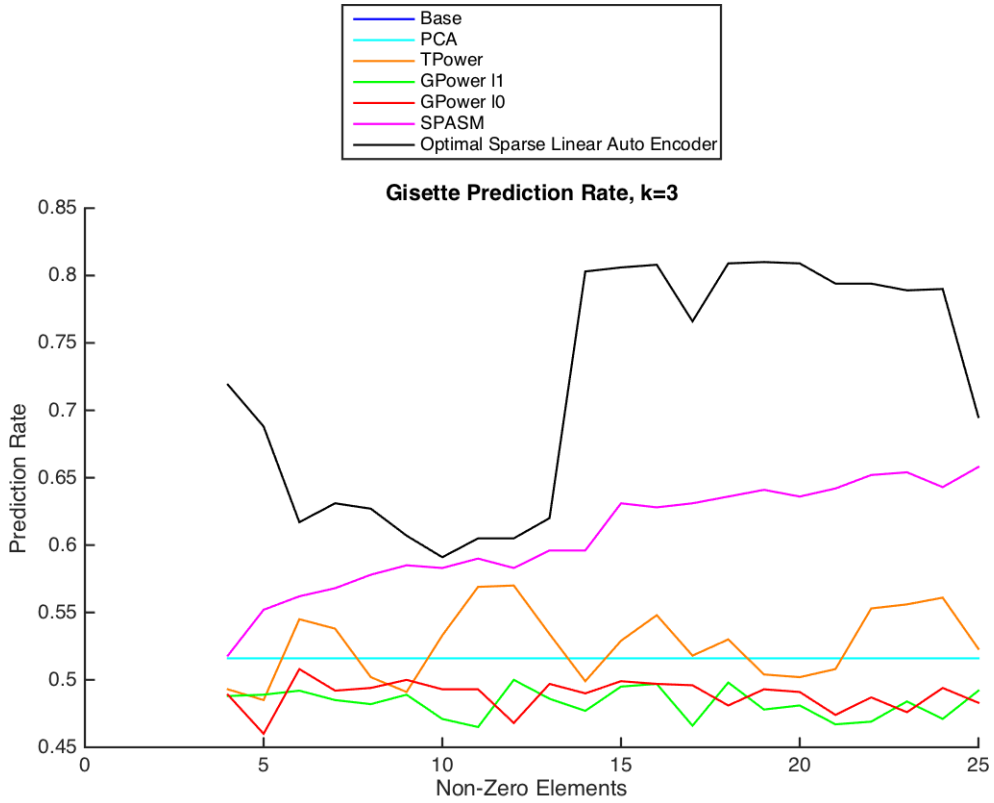

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Gisette Variance, k=1
Variance
Non-Zero Elements
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0
5
10
15
20
25

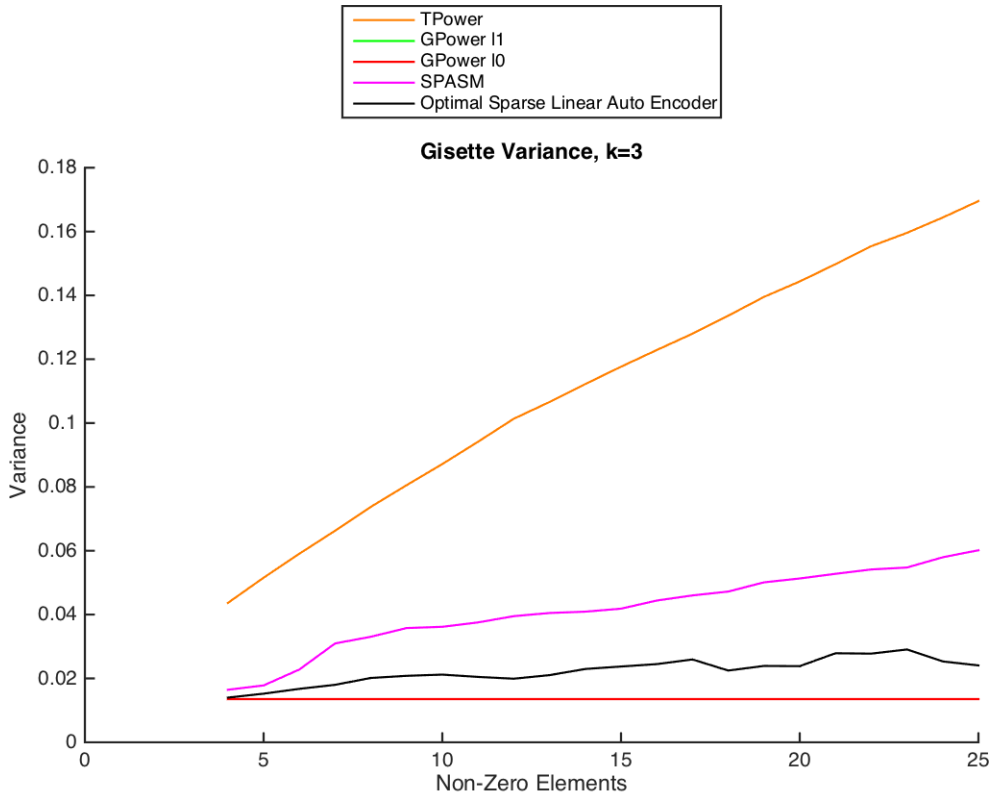

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Gisette Info Loss, k=2
Info Loss
Non-Zero Elements
1.01
1.02
1.03
1.04
1.05
1.06
1.07
1.08
1.09
1.1
0
5
10
15
20
25

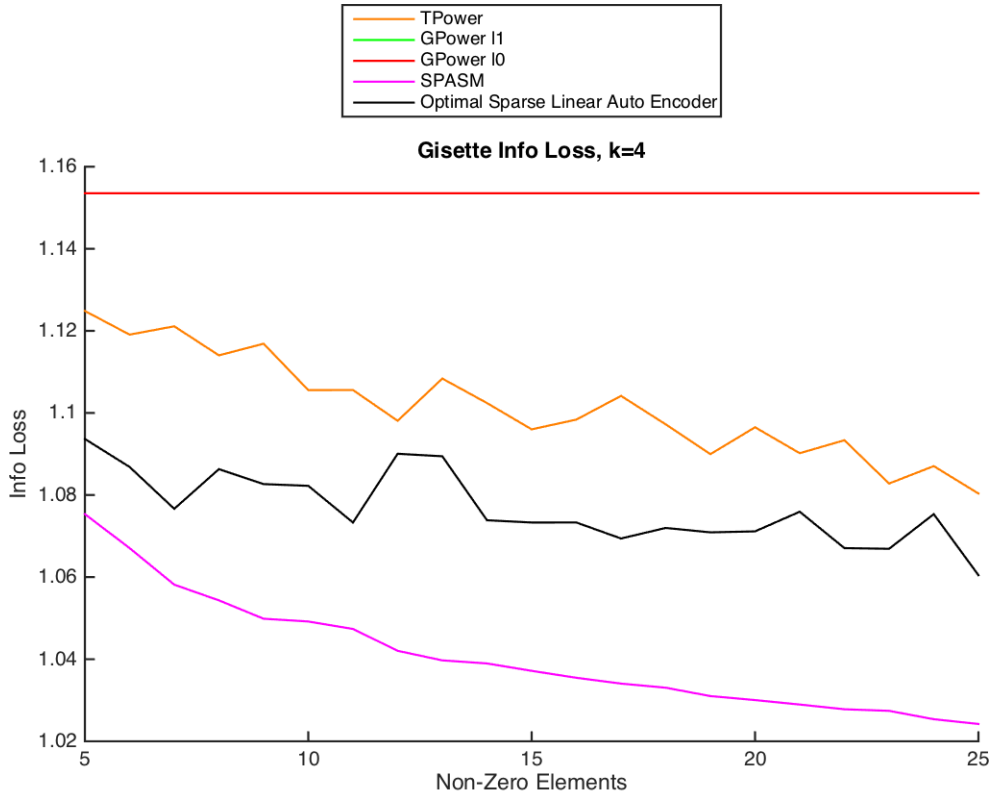

Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Gisette Prediction Rate, k=2
Prediction Rate
Non-Zero Elements
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.4
0
5
10
15
20
25

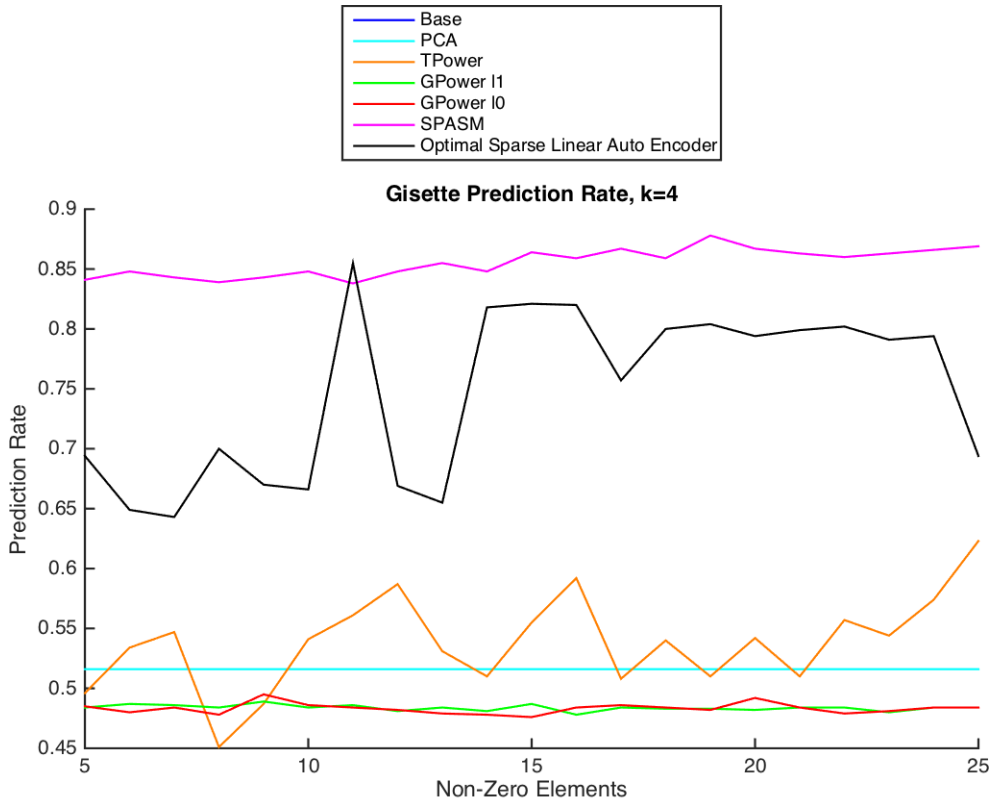

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Gisette Variance, k=2
Variance
Non-Zero Elements
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0
5
10
15
20
25

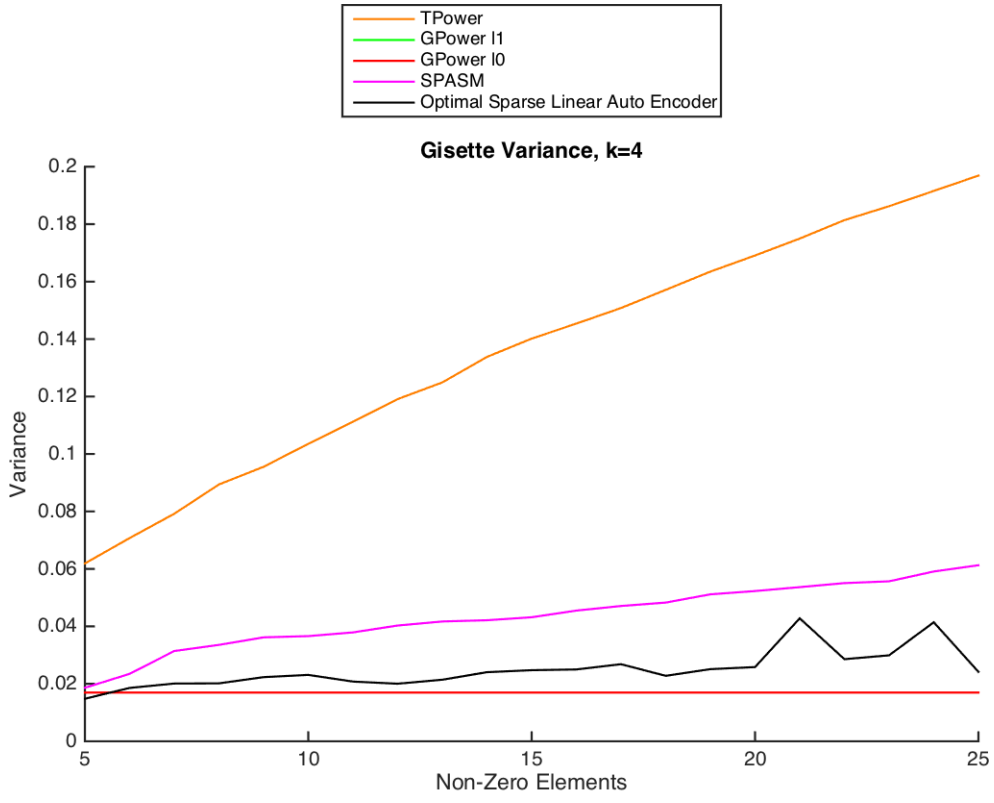

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Gisette Info Loss, k=3
Info Loss
Non-Zero Elements
1.14
1.12
1.1
1.08
1.06
1.04
1.02
0
5
10
15
20
25

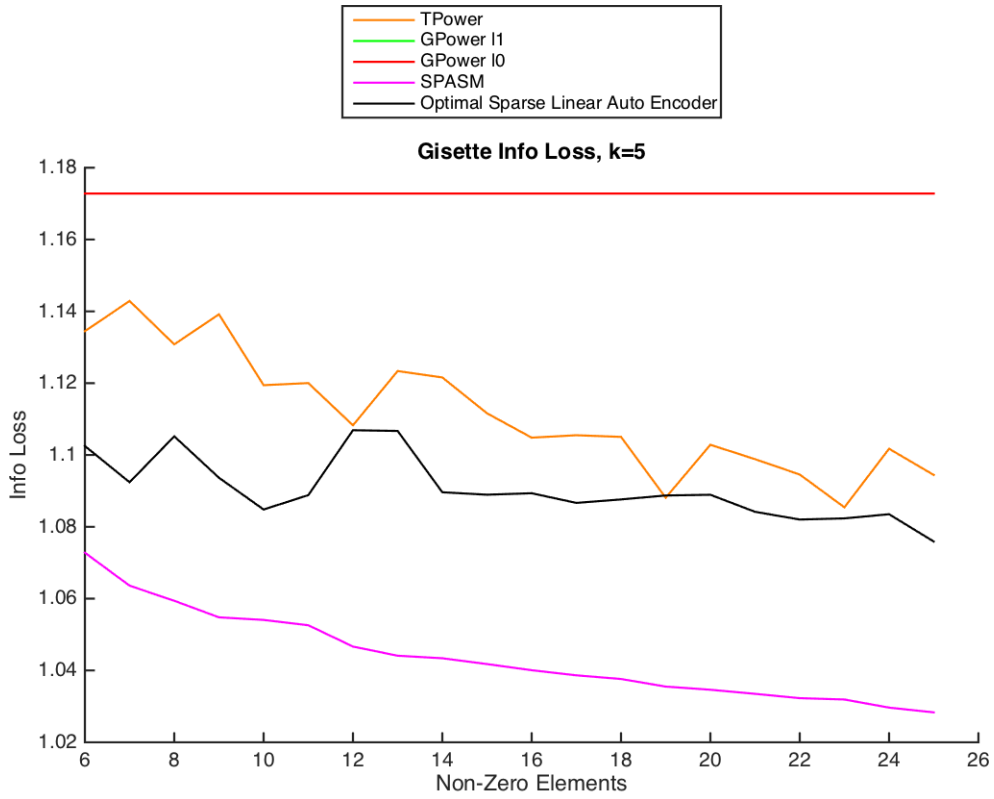

Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Gisette Prediction Rate, k=3
Prediction Rate
Non-Zero Elements
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0
5
10
15
20
25

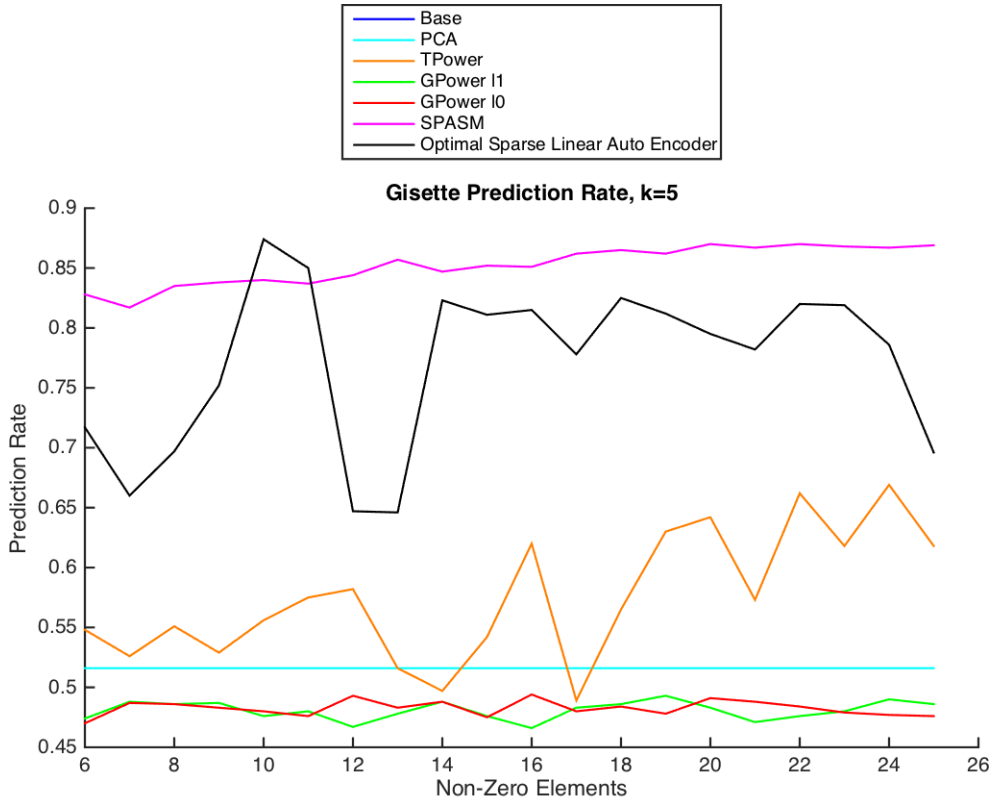

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Gisette Variance, k=3
Variance
Non-Zero Elements
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0
5
10
15
20
25

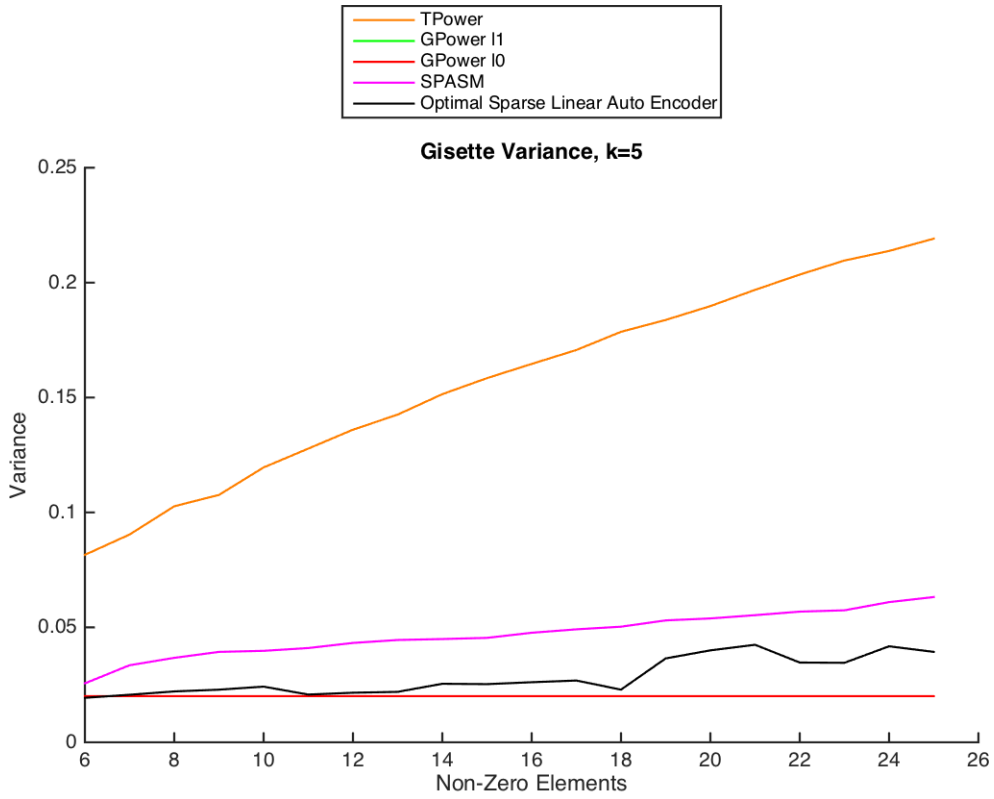

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Gisette Info Loss, k=4
Info Loss
Non-Zero Elements
1.16
1.14
1.12
1.1
1.08
1.06
1.04
1.02
5
10
15
20
25

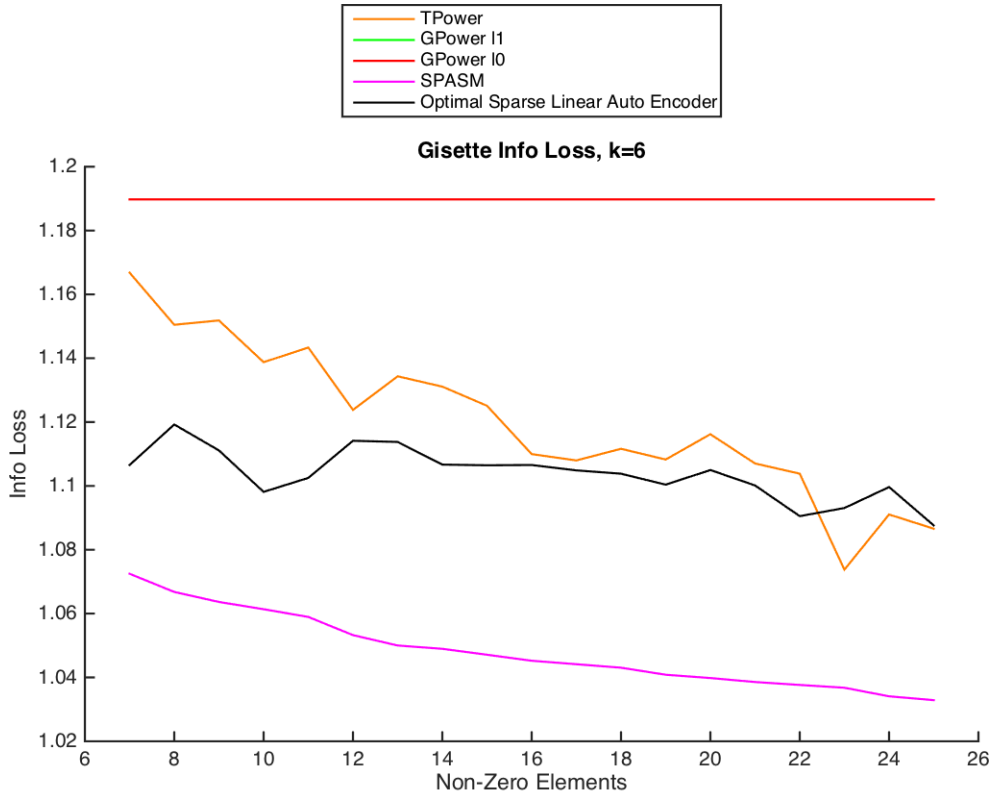

Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Gisette Prediction Rate, k=4
Prediction Rate
Non-Zero Elements
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0.45
5
10
15
20
25

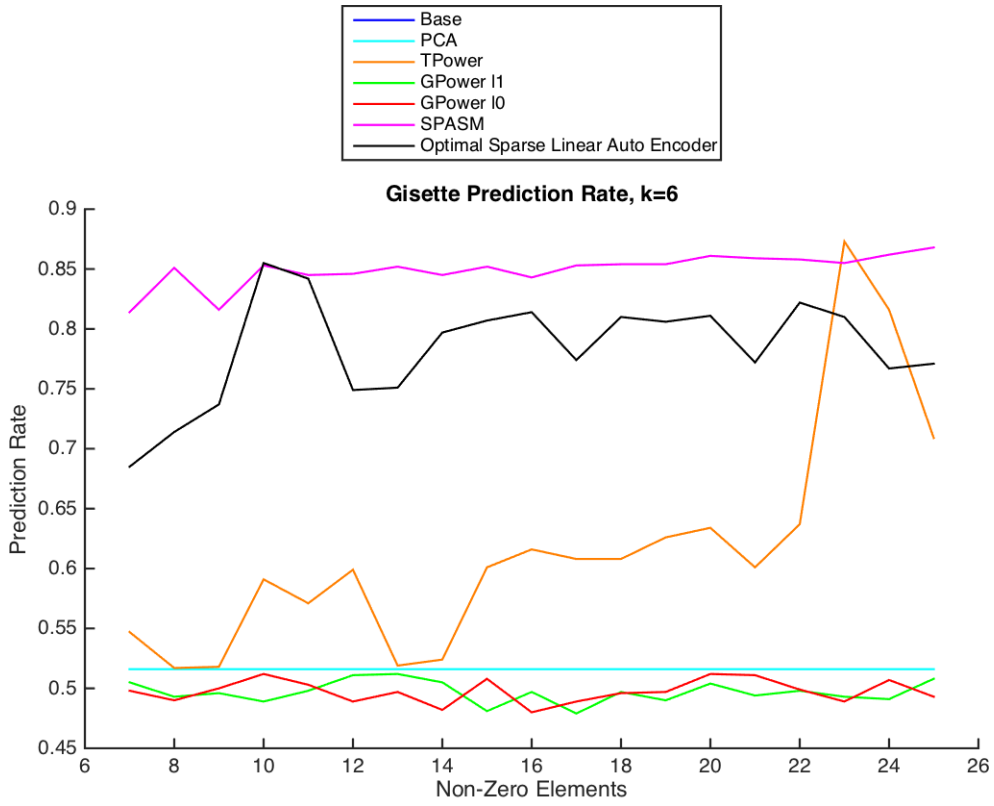

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Gisette Variance, k=4
Variance
Non-Zero Elements
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
5
10
15
20
25

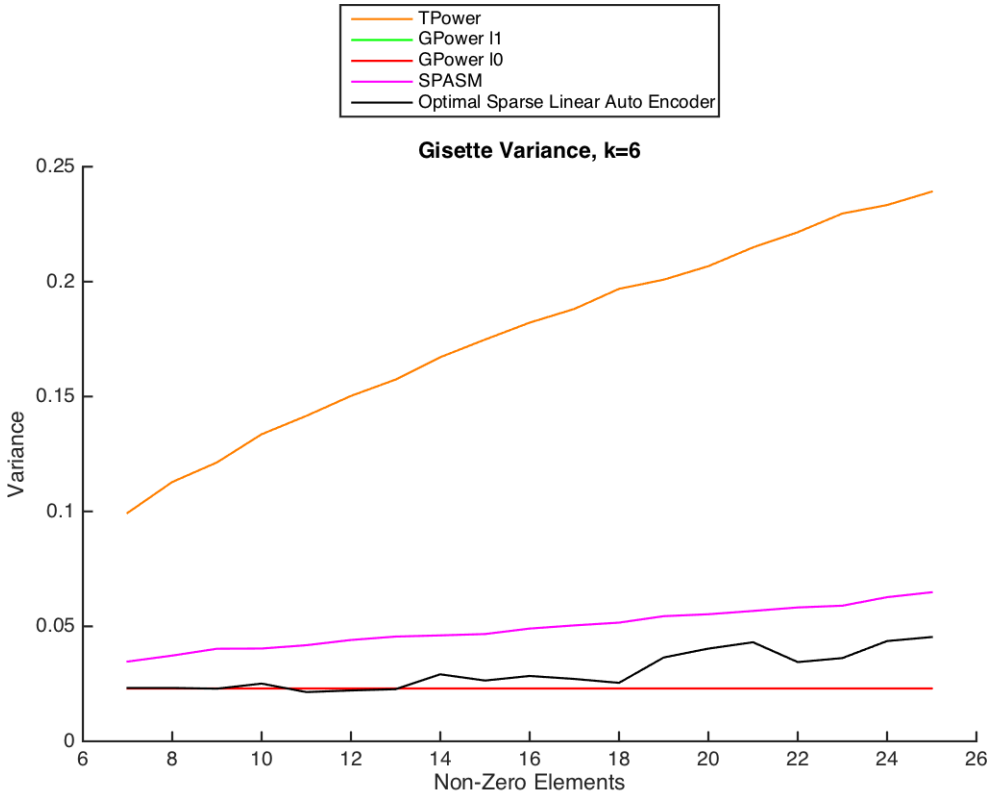

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Gisette Info Loss, k=5
Info Loss
Non-Zero Elements
1.18
1.16
1.14
1.12
1.1
1.08
1.06
1.04
1.02
6
8
10
12
14
16
18
20
22
24
26


Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Gisette Prediction Rate, k=5
Prediction Rate
Non-Zero Elements
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0.45
6
8
10
12
14
16
18
20
22
24
26

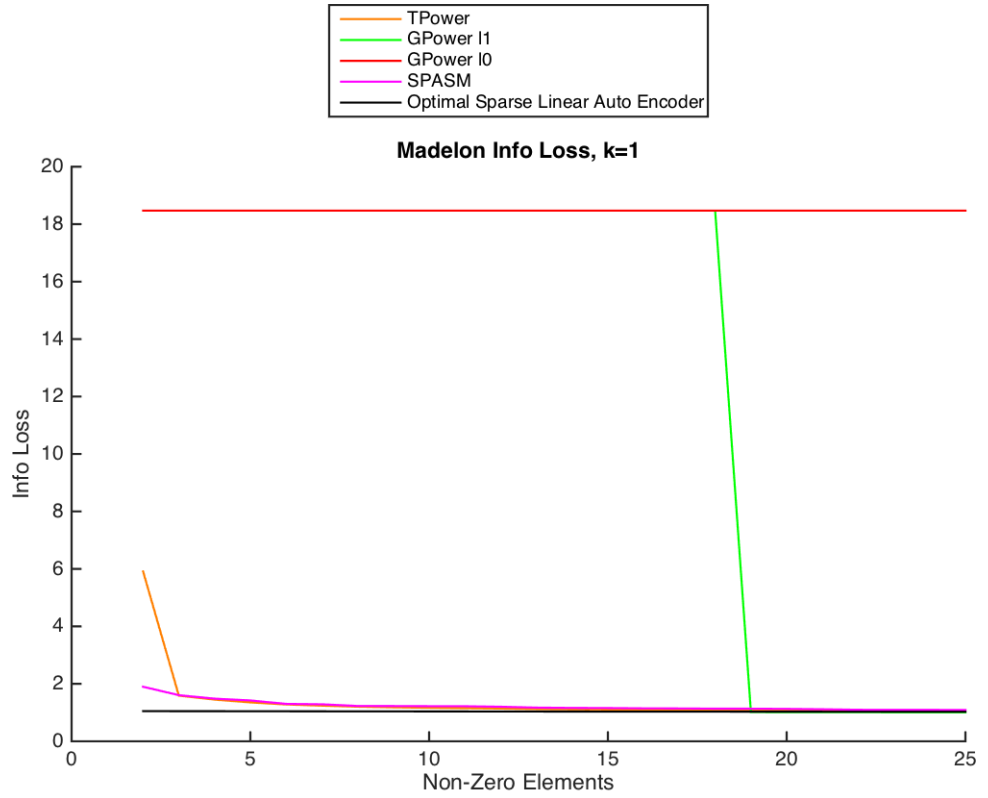

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Gisette Variance, k=5
Variance
Non-Zero Elements
0.25
0.2
0.15
0.1
0.05
0
6
8
10
12
14
16
18
20
22
24
26

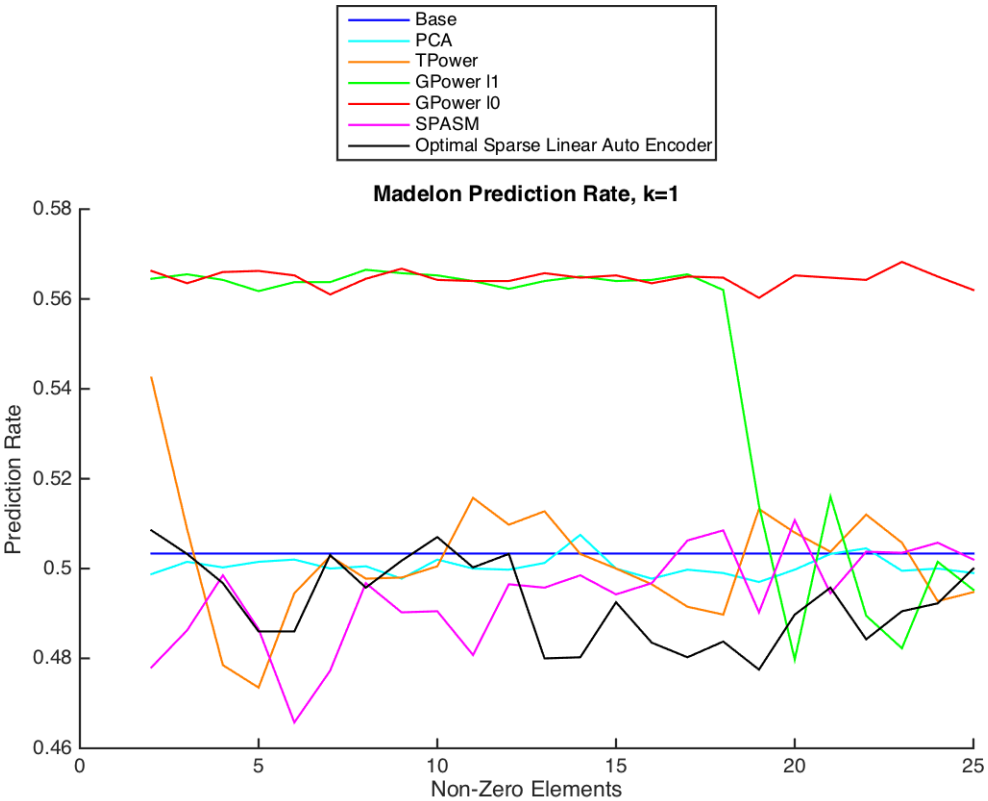

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Gisette Info Loss, k=6
Info Loss
Non-Zero Elements
1.2
1.18
1.16
1.14
1.12
1.1
1.08
1.06
1.04
1.02
6
8
10
12
14
16
18
20
22
24
26

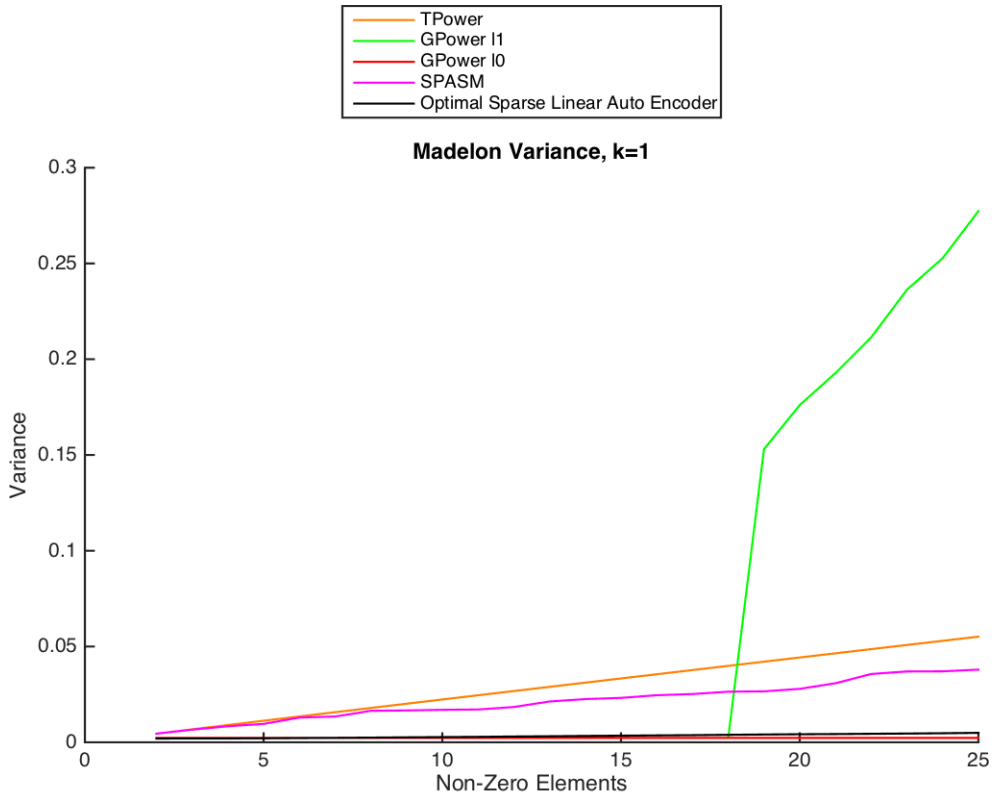

Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Gisette Prediction Rate, k=6
Prediction Rate
Non-Zero Elements
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0.45
6
8
10
12
14
16
18
20
22
24
26

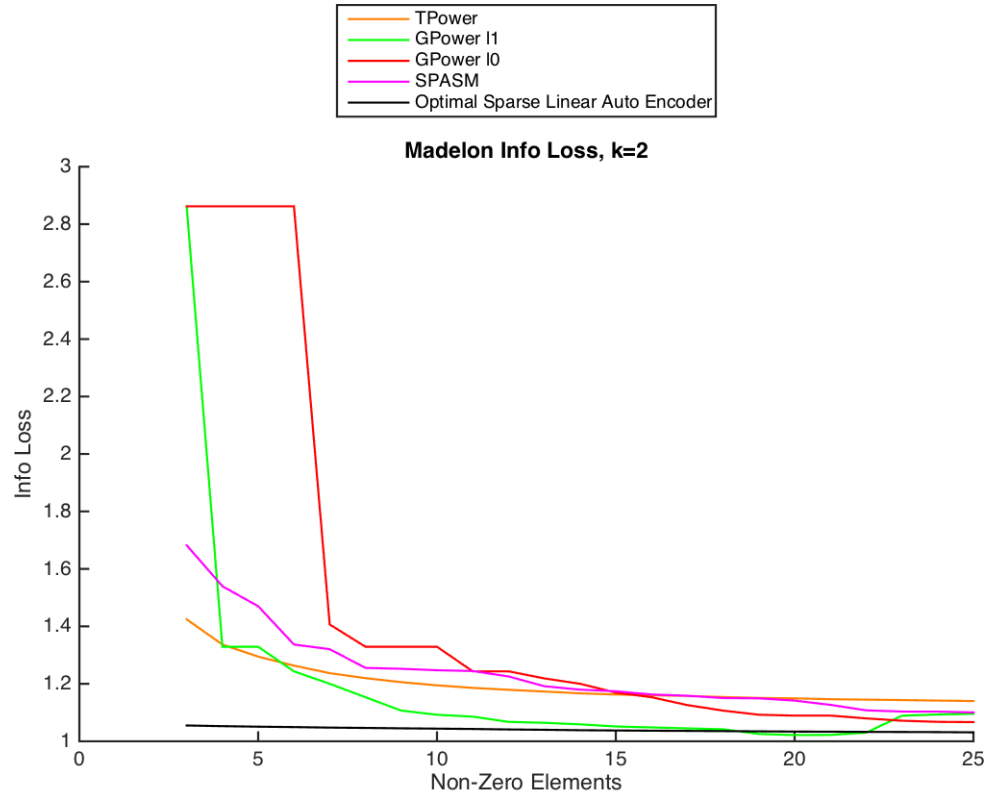

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Gisette Variance, k=6
Variance
Non-Zero Elements
0.25
0.2
0.15
0.1
0.05
0
6
8
10
12
14
16
18
20
22
24
26

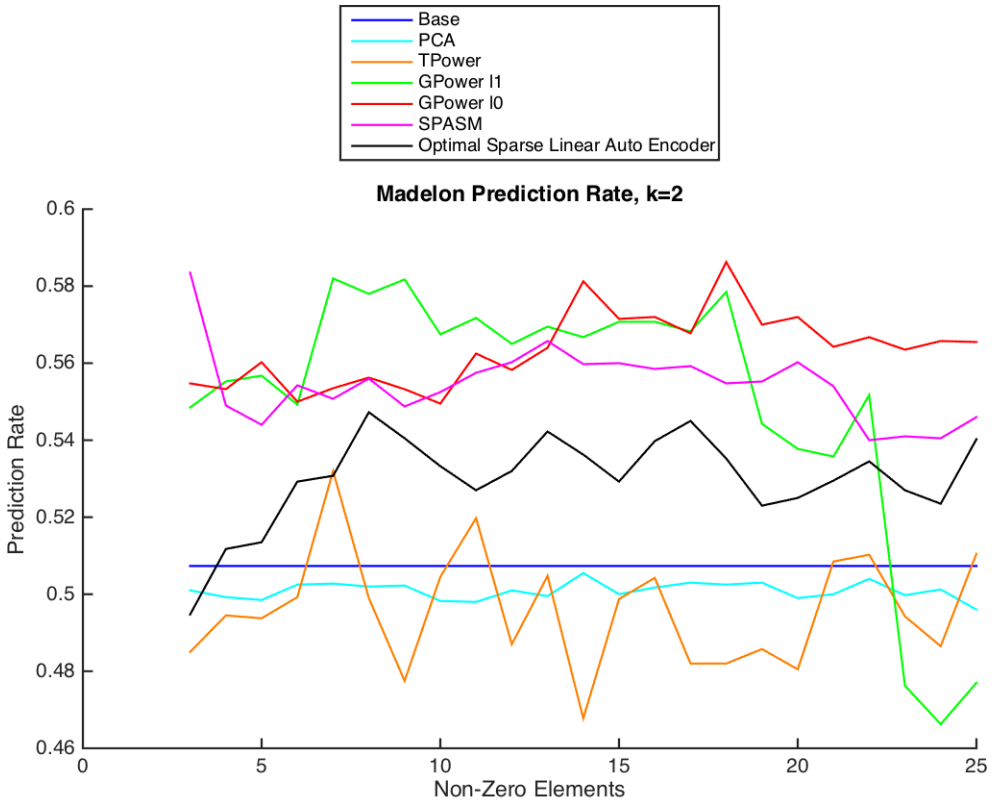
A.4 Madelon Results

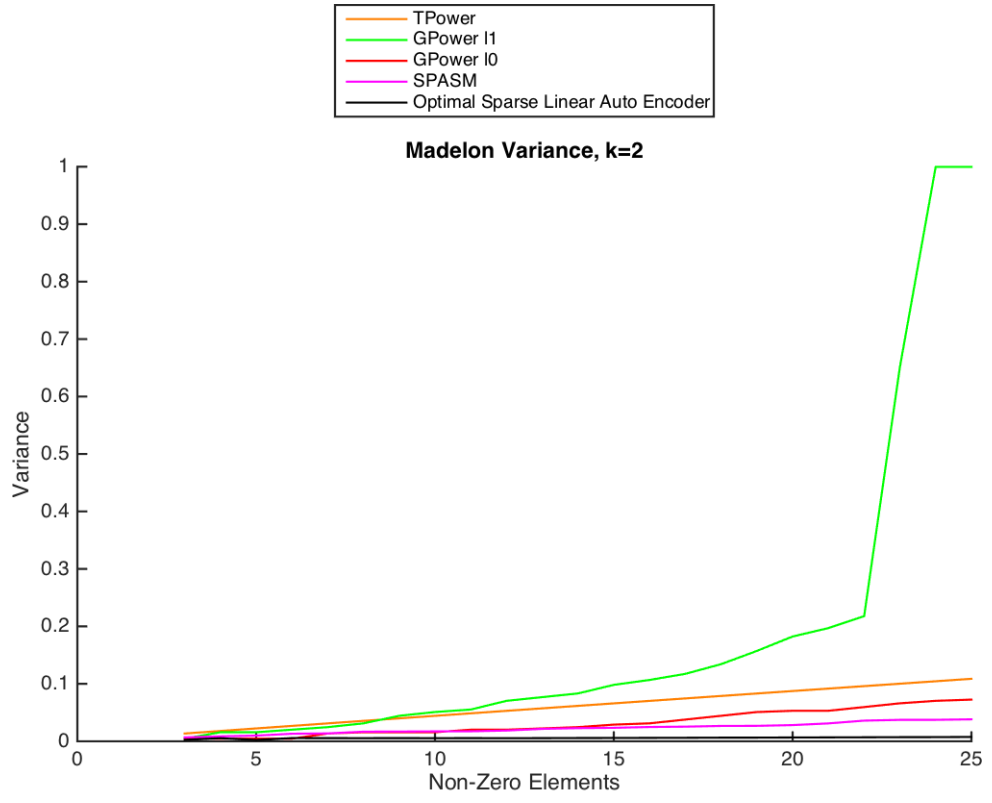


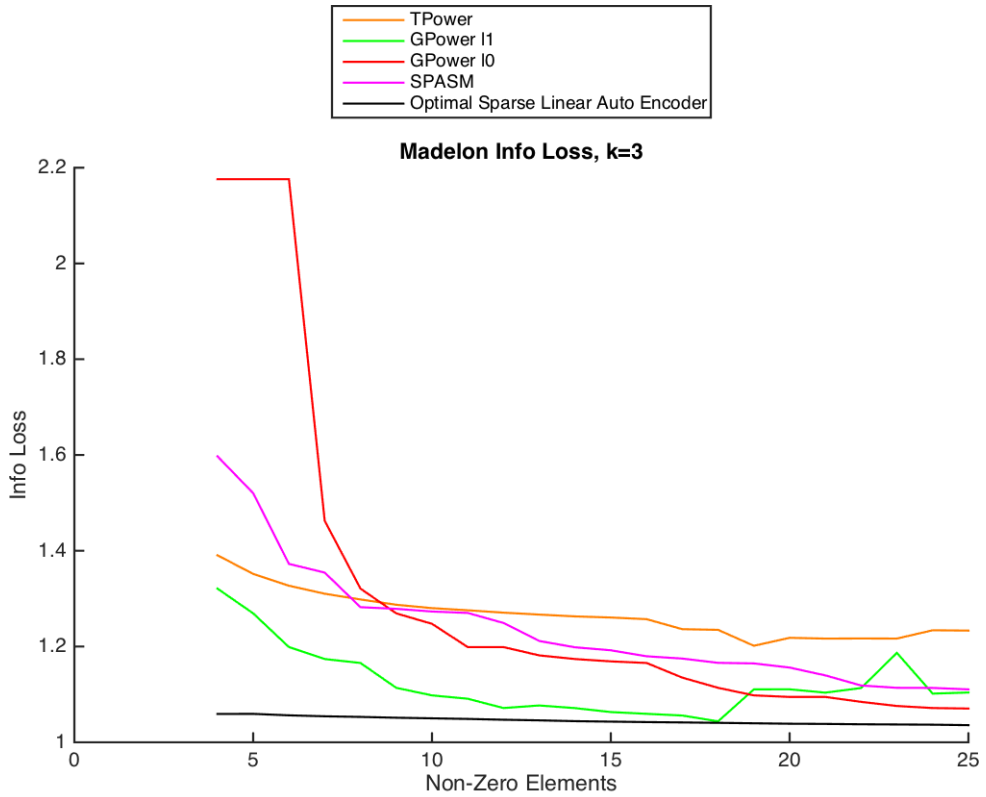


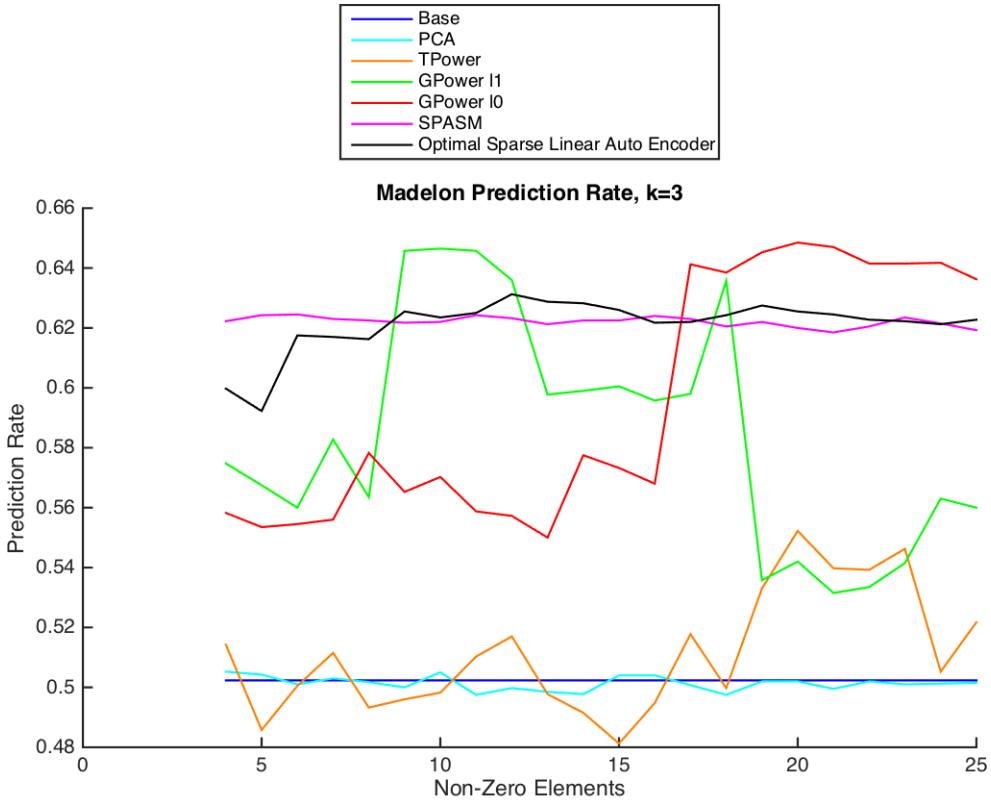

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Madelon Variance, k=1
Variance
Non-Zero Elements
0
0.05
0.1
0.15
0.2
0.25
0.3
0
5
10
15
20
25

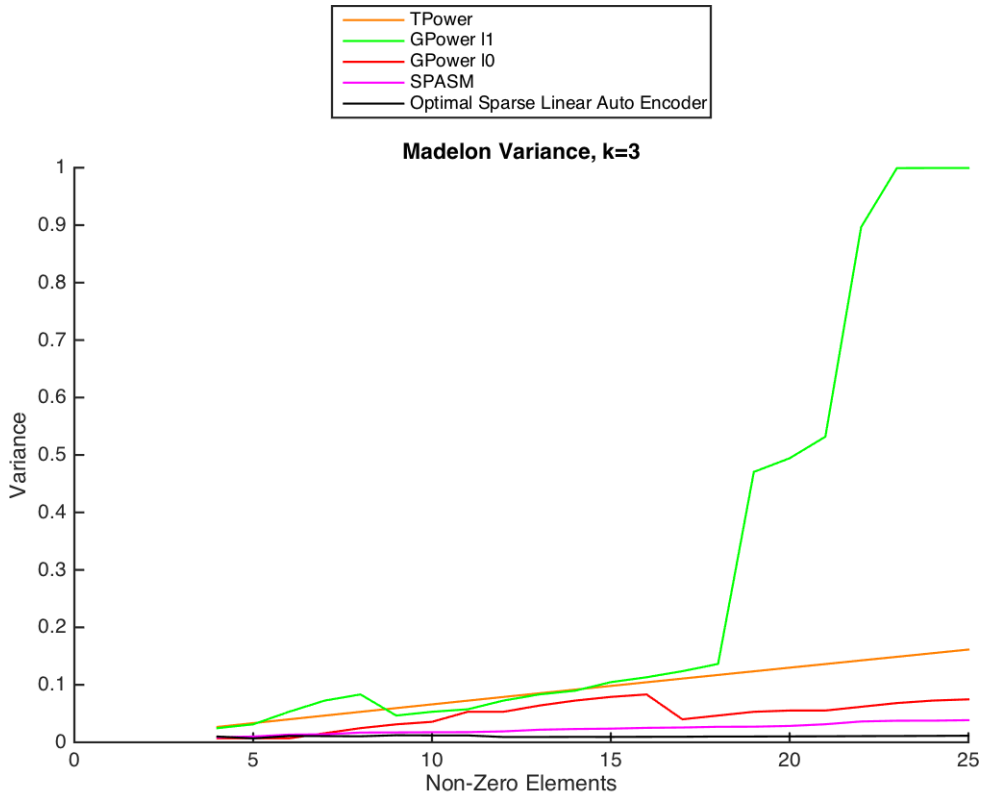

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Madelon Info Loss, k=2
Info Loss
Non-Zero Elements
1
1.2
1.4
1.6
1.8
2
2.2
2.4
2.6
2.8
3
0
5
10
15
20
25

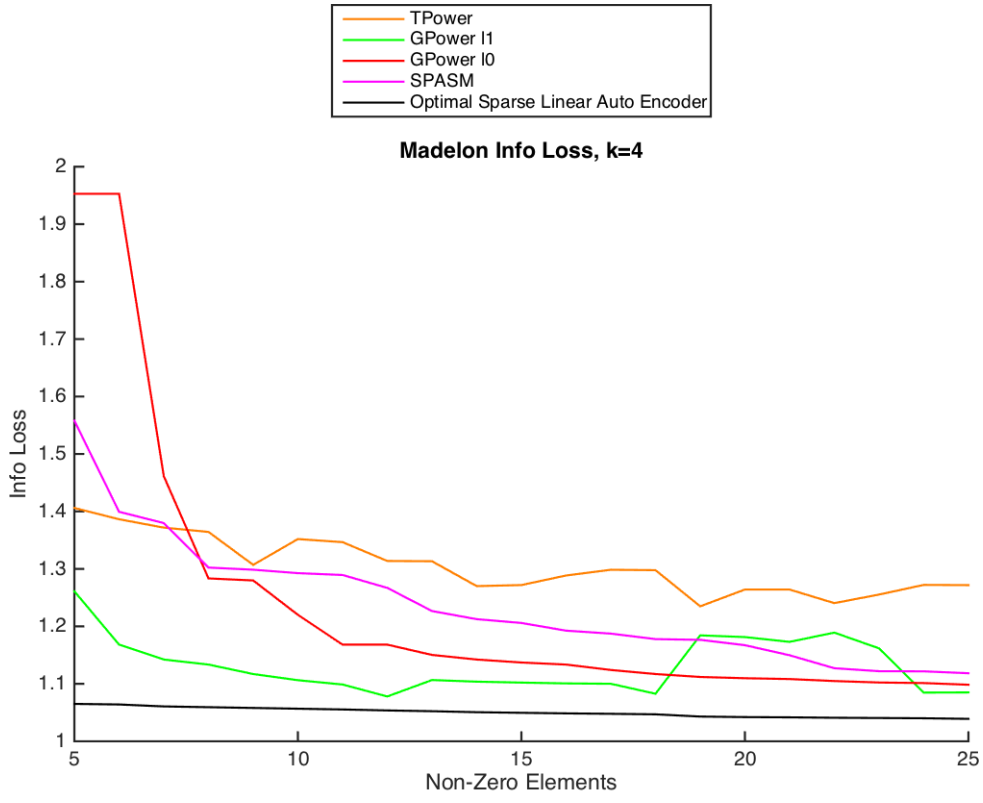

Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Madelon Prediction Rate, k=2
Prediction Rate
Non-Zero Elements
0.6
0.58
0.56
0.54
0.52
0.5
0.48
0.46
0
5
10
15
20
25

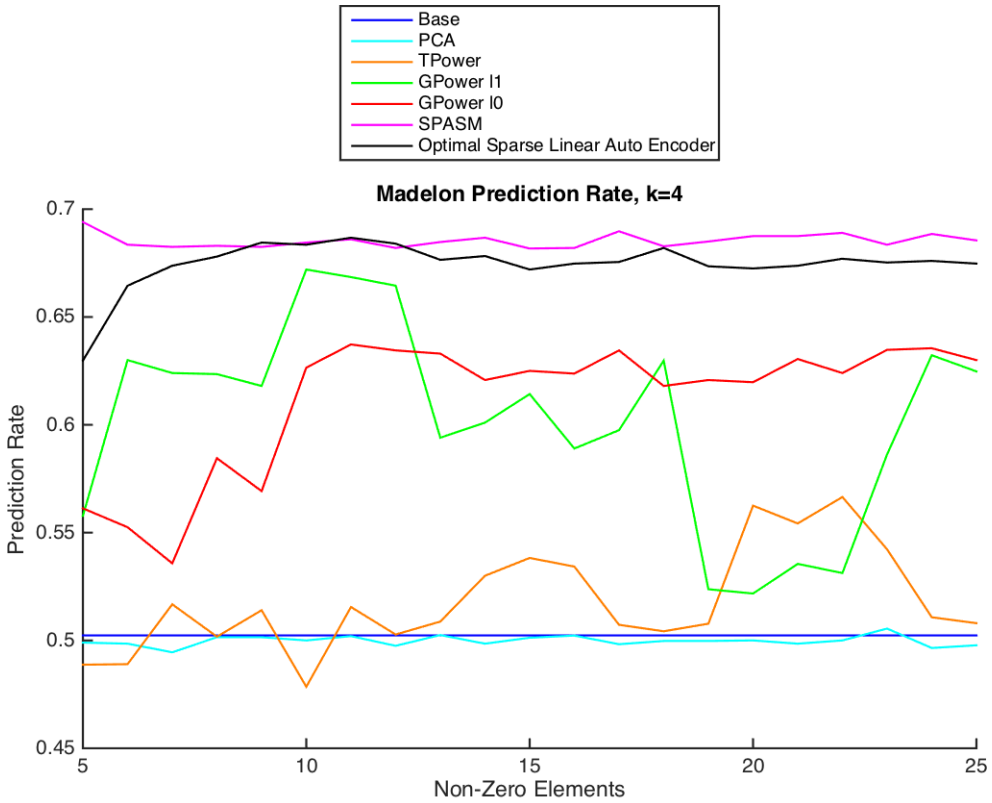

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Madelon Variance, k=2
Variance
Non-Zero Elements
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
5
10
15
20
25

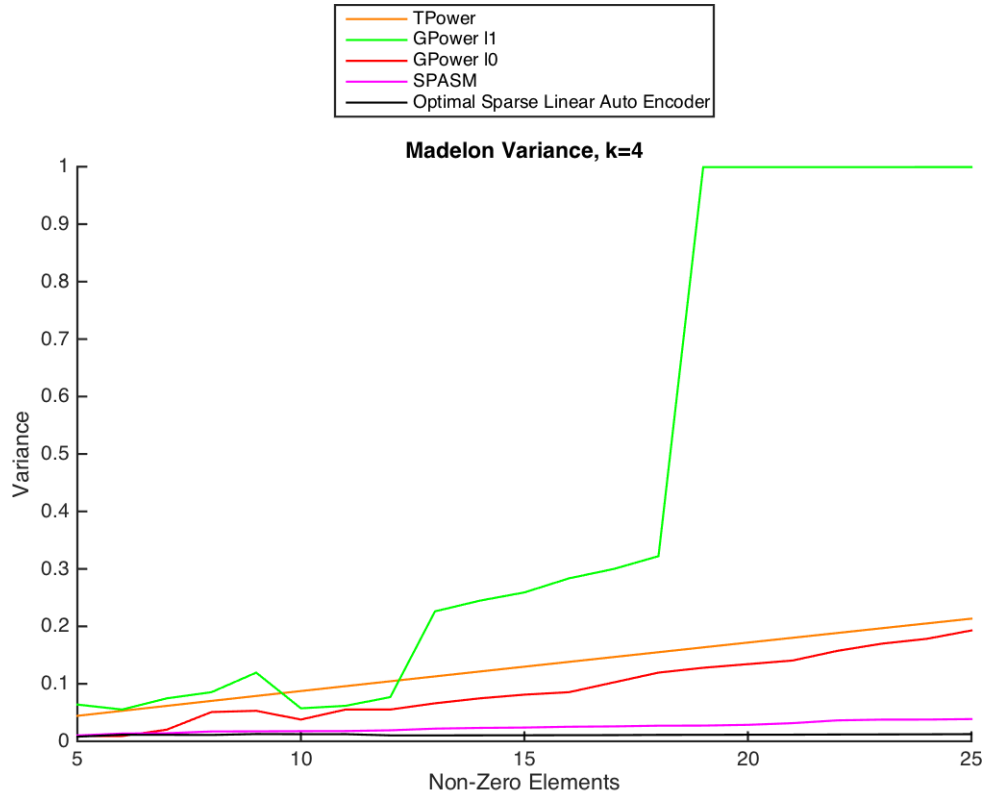

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Madelon Info Loss, k=3
Info Loss
Non-Zero Elements
2.2
2
1.8
1.6
1.4
1.2
1
0
5
10
15
20
25

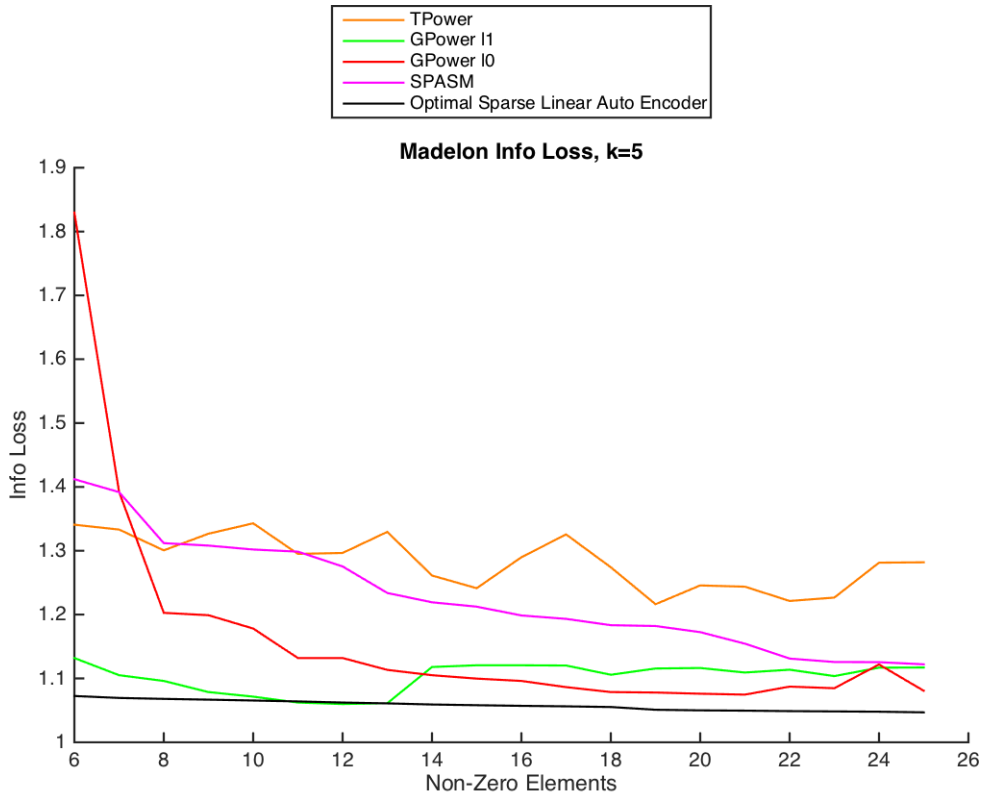

Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Madelon Prediction Rate, k=3
Prediction Rate
Non-Zero Elements
0.66
0.64
0.62
0.6
0.58
0.56
0.54
0.52
0.5
0.48
0
5
10
15
20
25

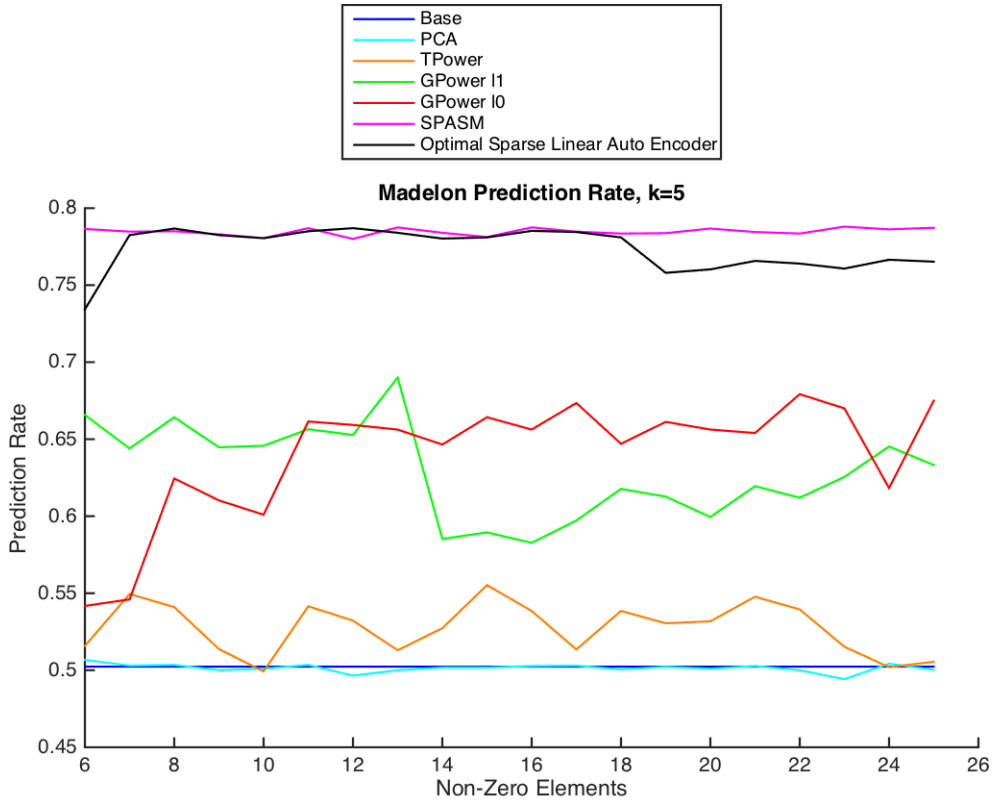

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Madelon Variance, k=3
Variance
Non-Zero Elements
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
5
10
15
20
25

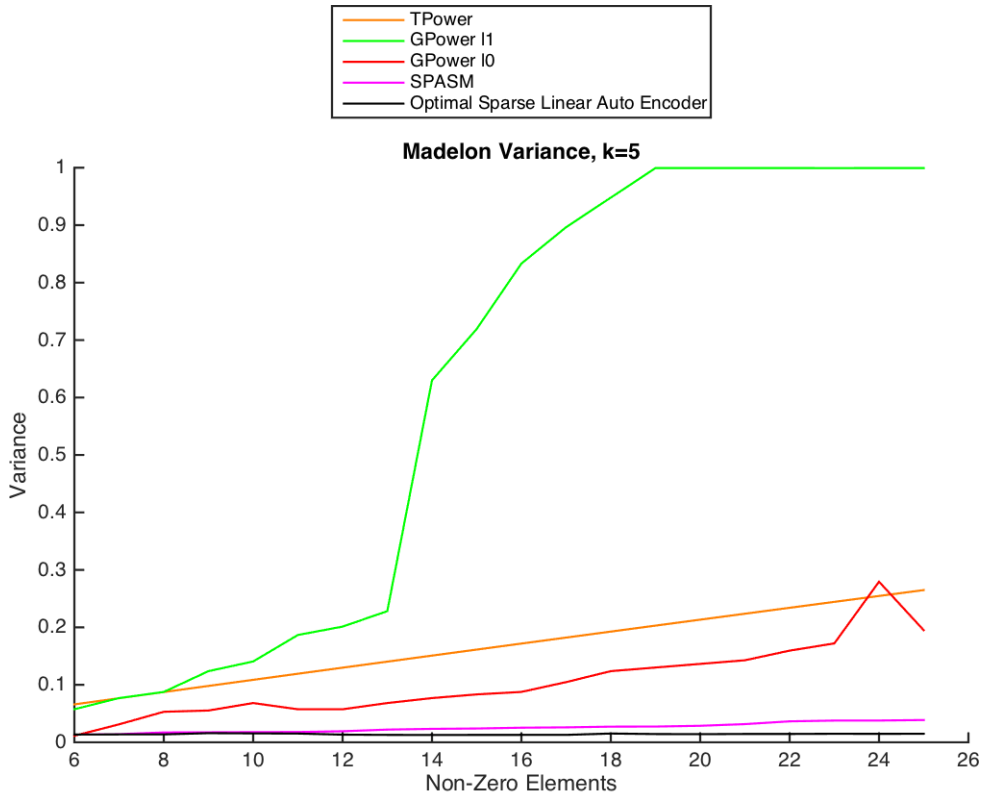

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Madelon Info Loss, k=4
Info Loss
Non-Zero Elements
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
5
10
15
20
25

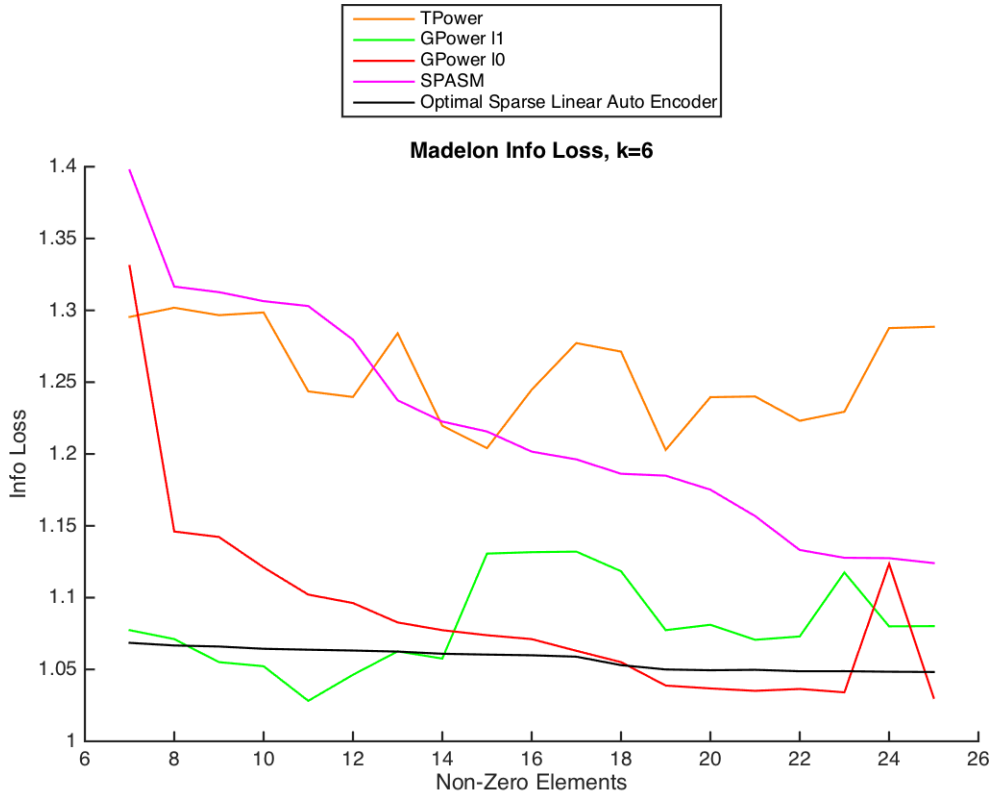

Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Madelon Prediction Rate, k=4
Prediction Rate
Non-Zero Elements
0.7
0.65
0.6
0.55
0.5
0.45
5
10
15
20
25

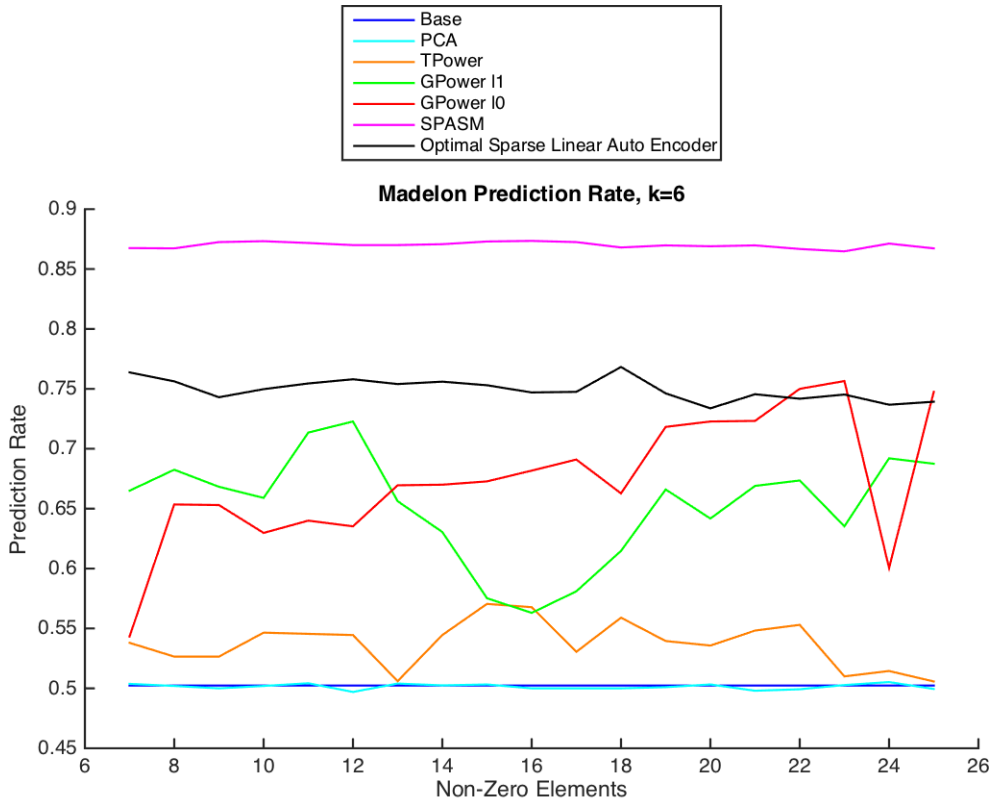

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Madelon Variance, k=4
Variance
Non-Zero Elements
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
5
10
15
20
25

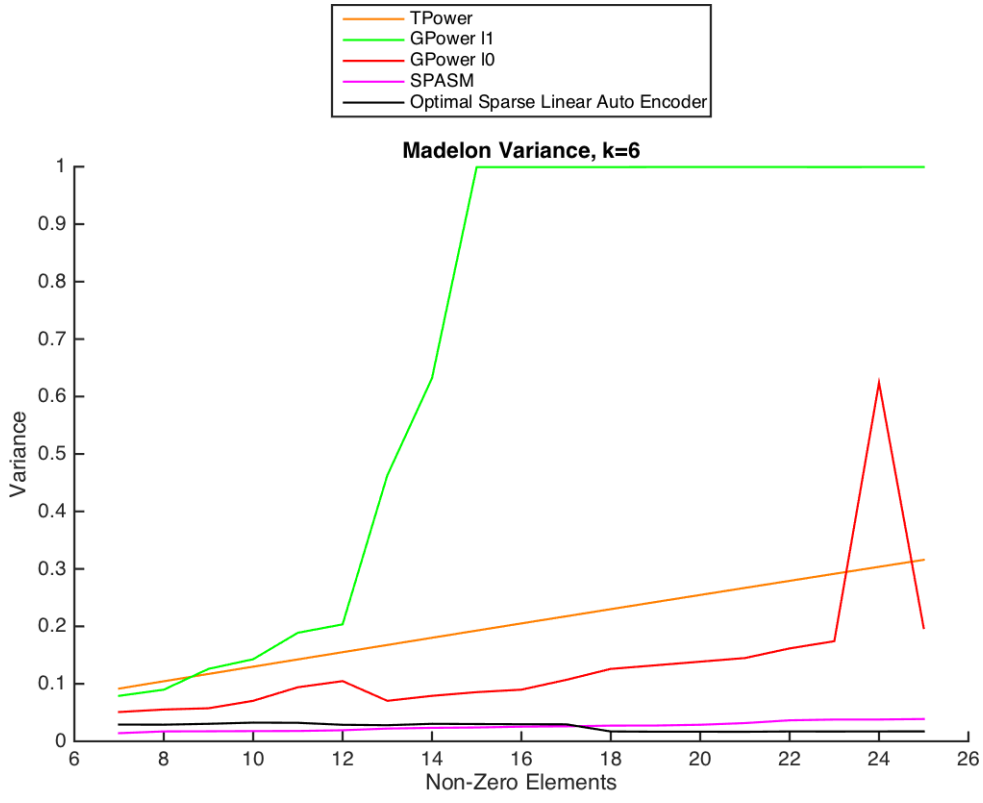

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Madelon Info Loss, k=5
Info Loss
Non-Zero Elements
1.9
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1
6
8
10
12
14
16
18
20
22
24
26


Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Madelon Prediction Rate, k=5
Prediction Rate
Non-Zero Elements
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0.45
6
8
10
12
14
16
18
20
22
24
26

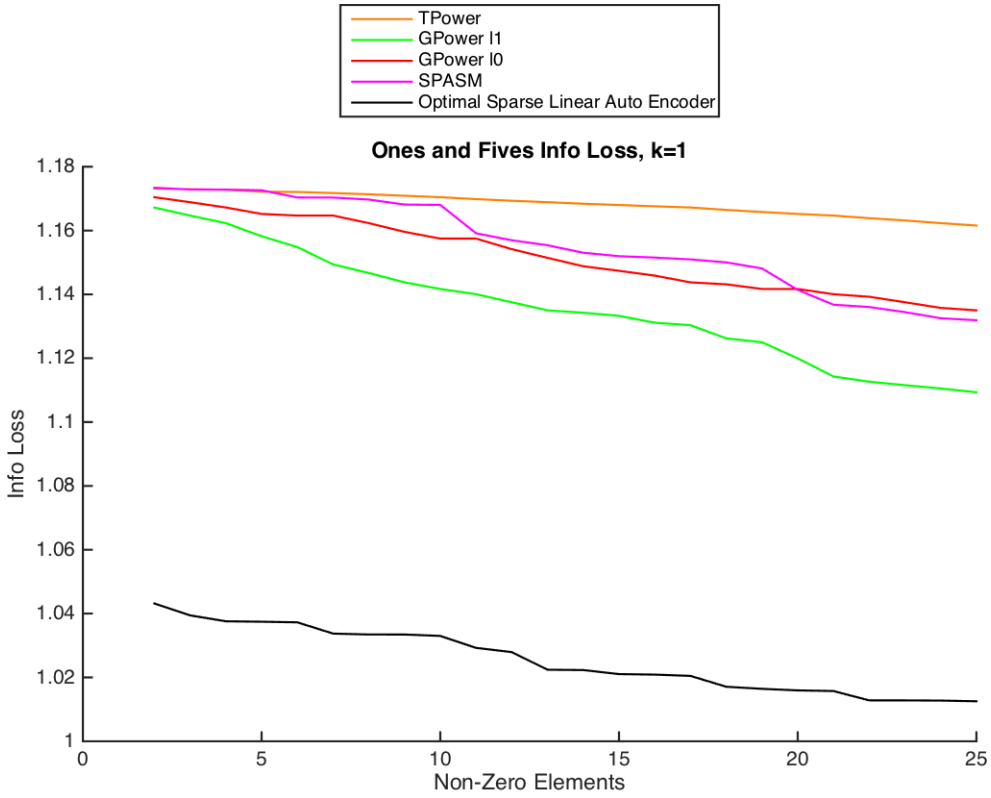

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Madelon Variance, k=5
Variance
Non-Zero Elements
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
6
8
10
12
14
16
18
20
22
24
26

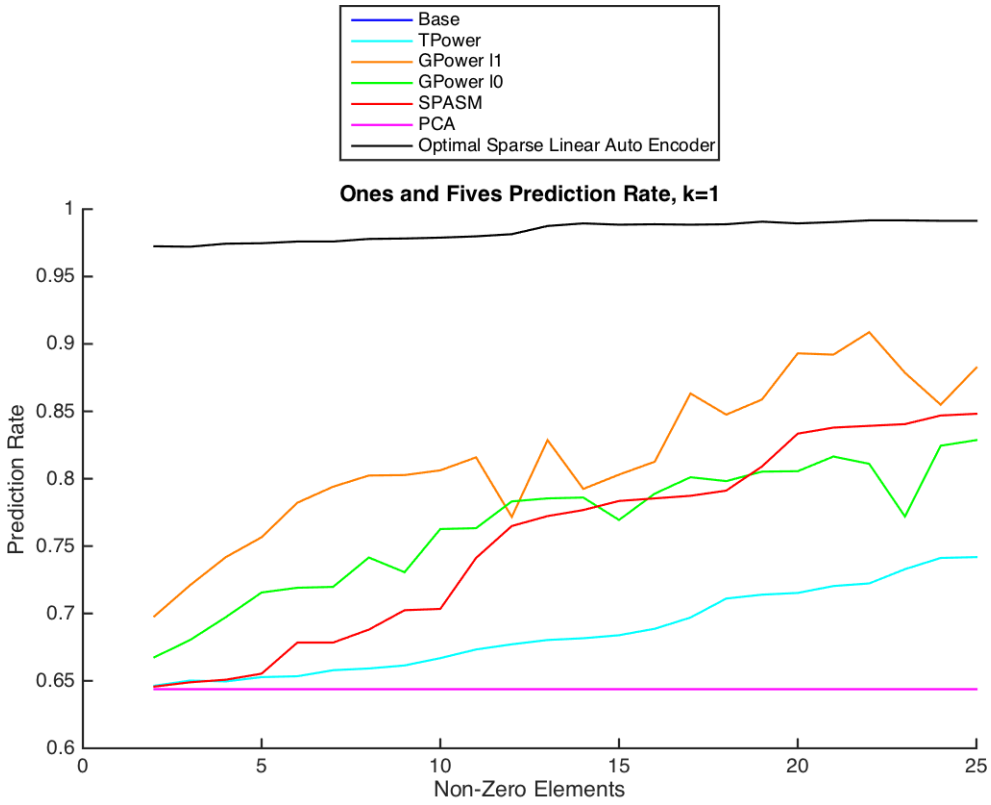

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Madelon Info Loss, k=6
Info Loss
Non-Zero Elements
1
1.05
1.1
1.15
1.2
1.25
1.3
1.35
1.4
6
8
10
12
14
16
18
20
22
24
26

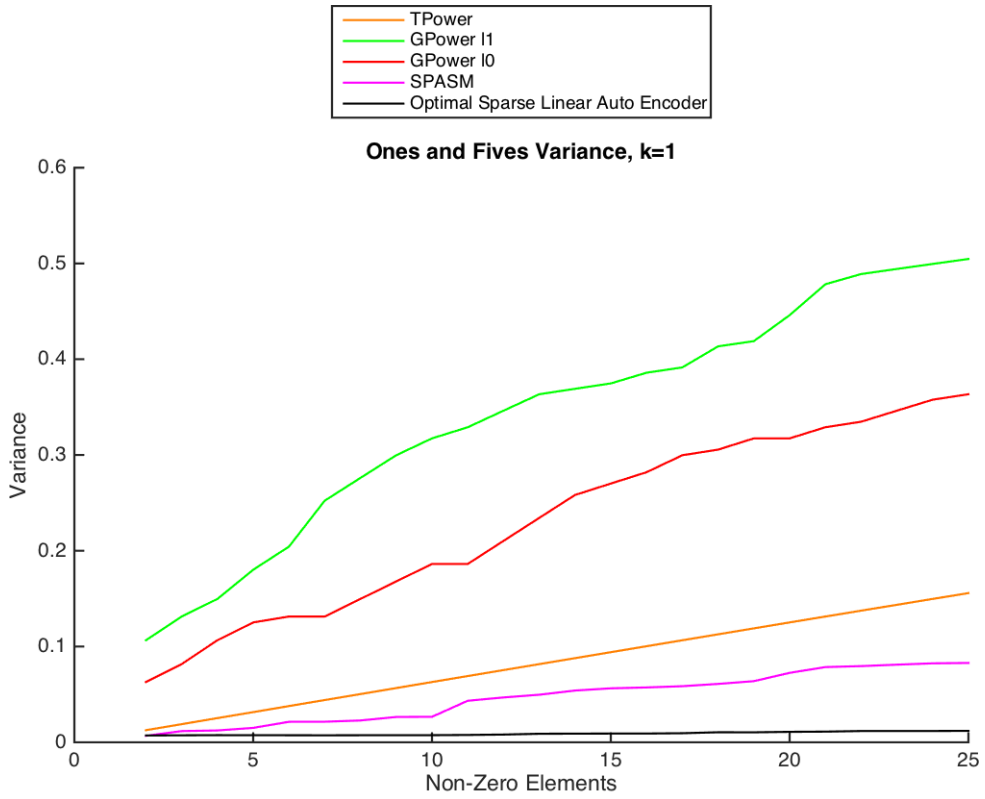

Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Madelon Prediction Rate, k=6
Prediction Rate
Non-Zero Elements
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0.45
6
8
10
12
14
16
18
20
22
24
26

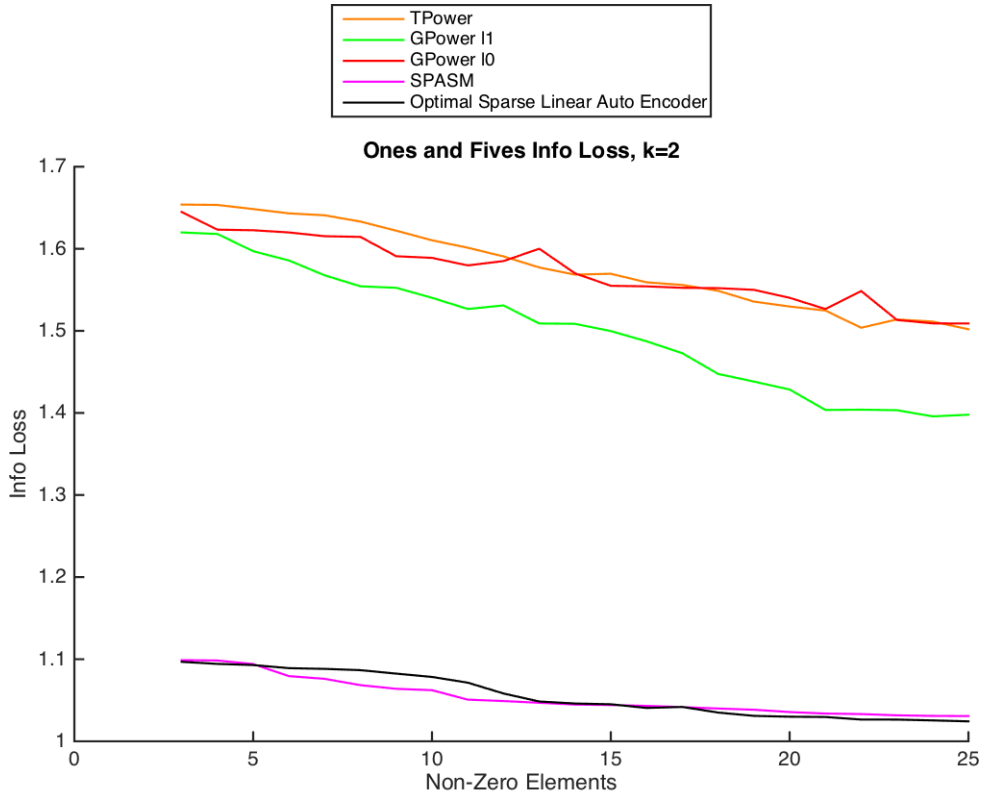

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Madelon Variance, k=6
Variance
Non-Zero Elements
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
6
8
10
12
14
16
18
20
22
24
26

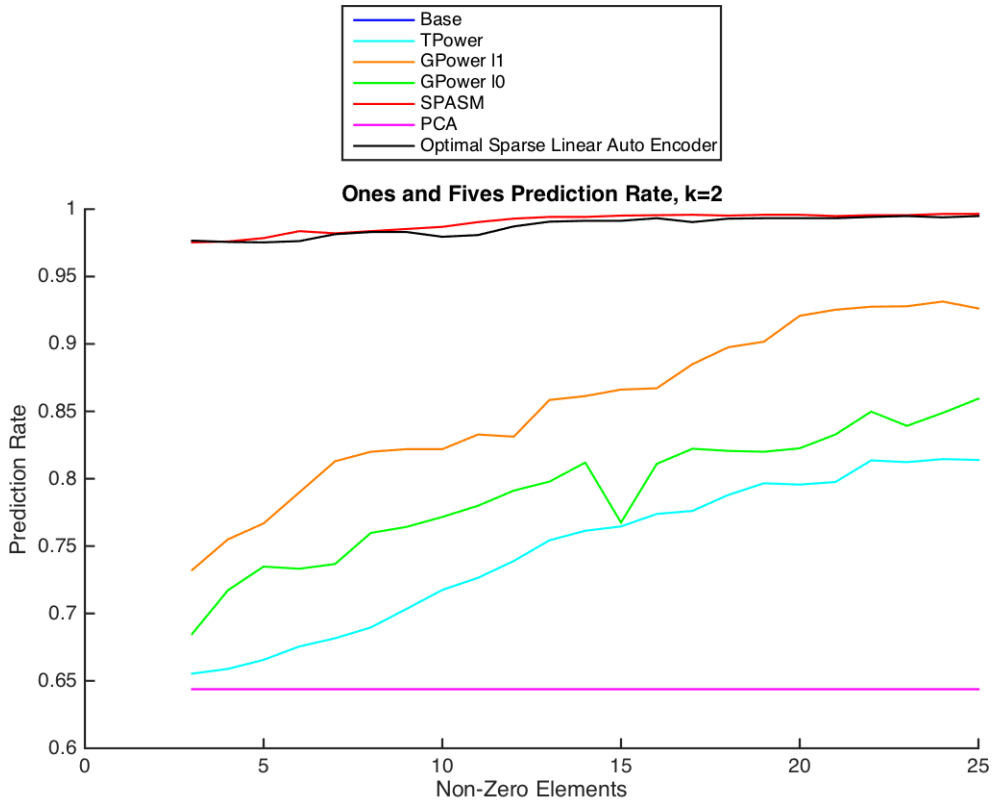
A.5 Ones and Fives Results

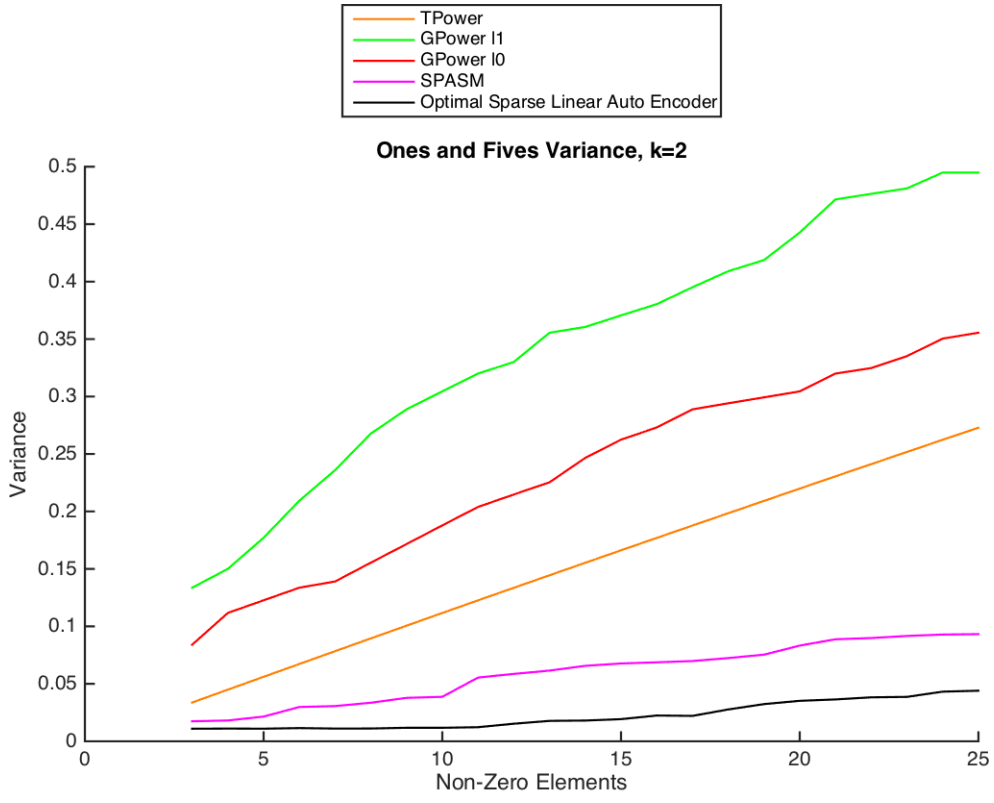


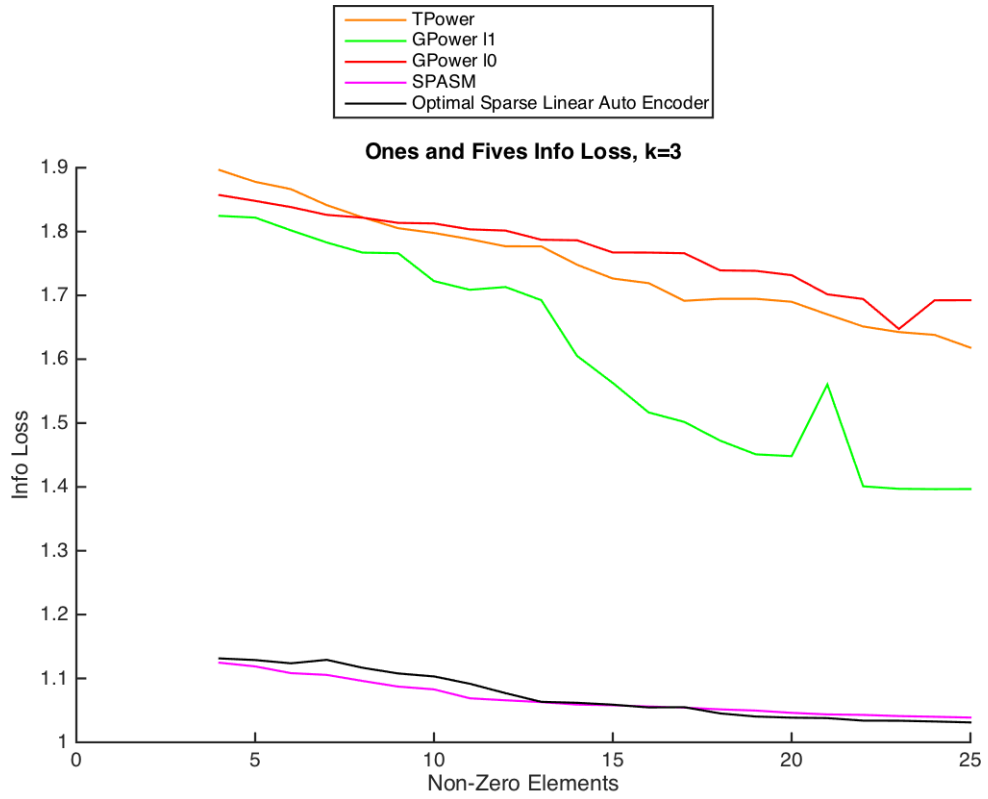


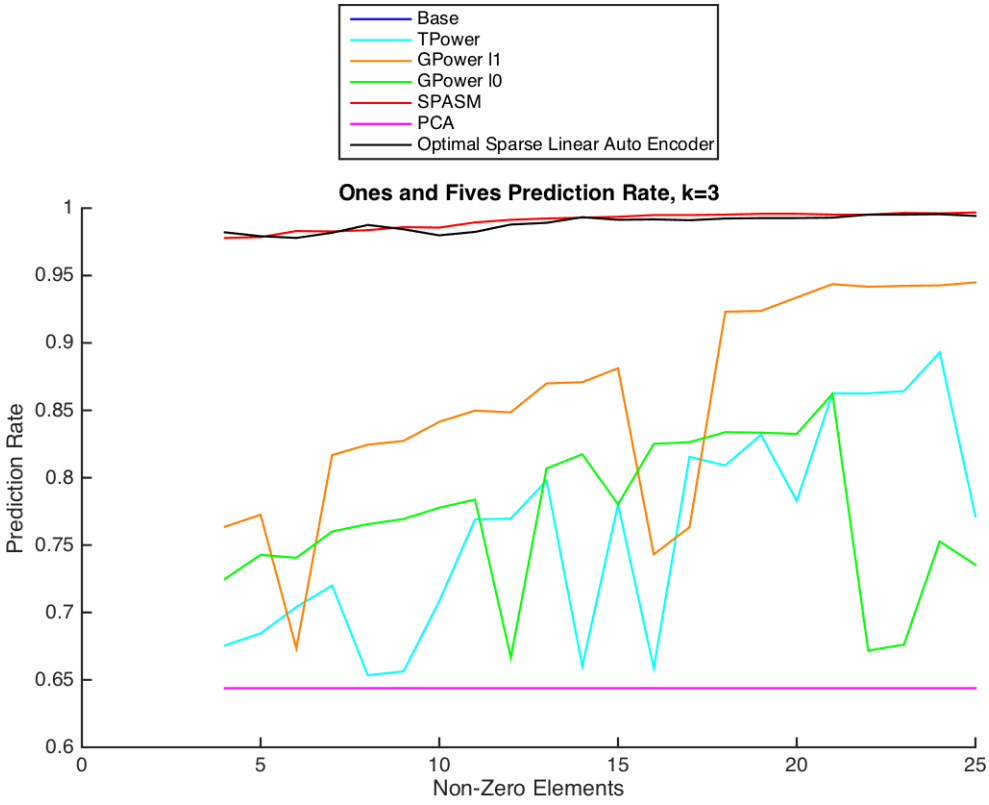

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Ones and Fives Variance, k=1
Variance
Non-Zero Elements
0
0.1
0.2
0.3
0.4
0.5
0.6
0
5
10
15
20
25

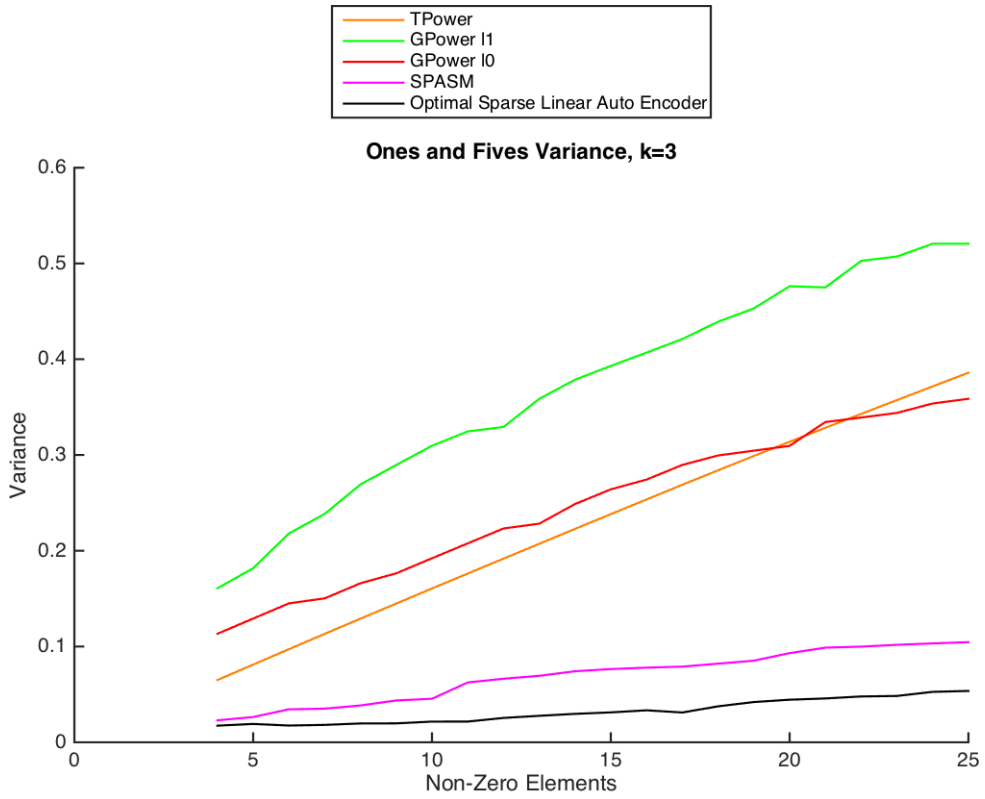

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Ones and Fives Info Loss, k=2
Info Loss
Non-Zero Elements
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
0
5
10
15
20
25

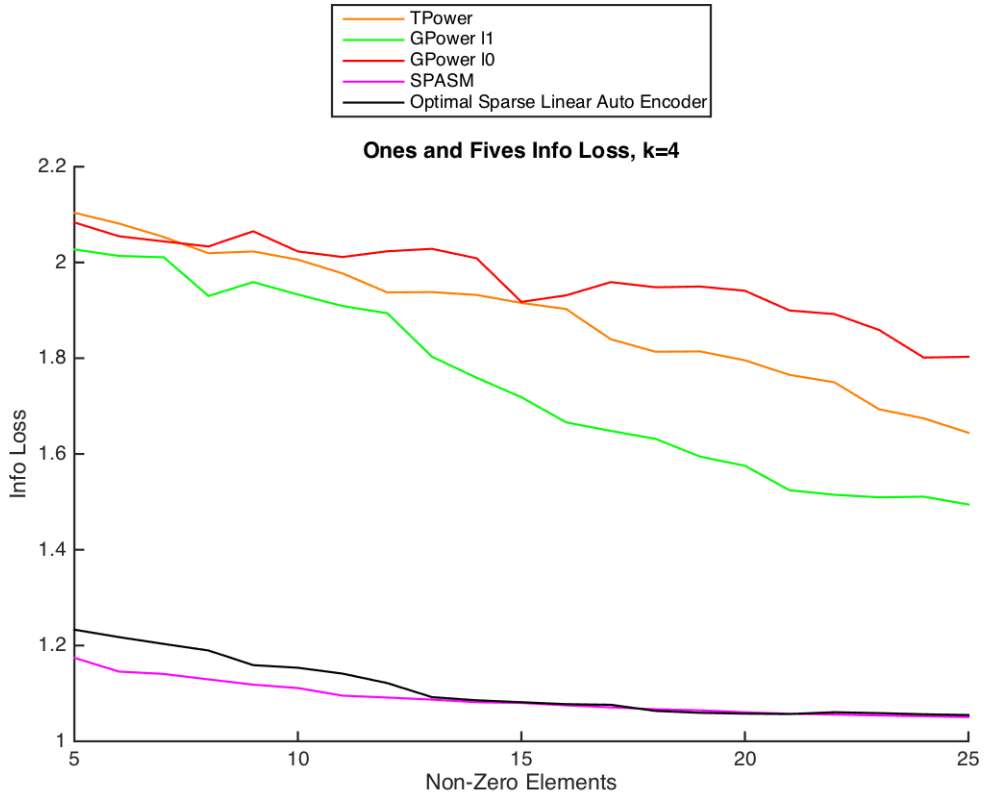

Base
TPower
GPower l1
GPower l0
SPASM
PCA
Optimal Sparse Linear Auto Encoder
Ones and Fives Prediction Rate, k=2
Prediction Rate
Non-Zero Elements
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0
5
10
15
20
25

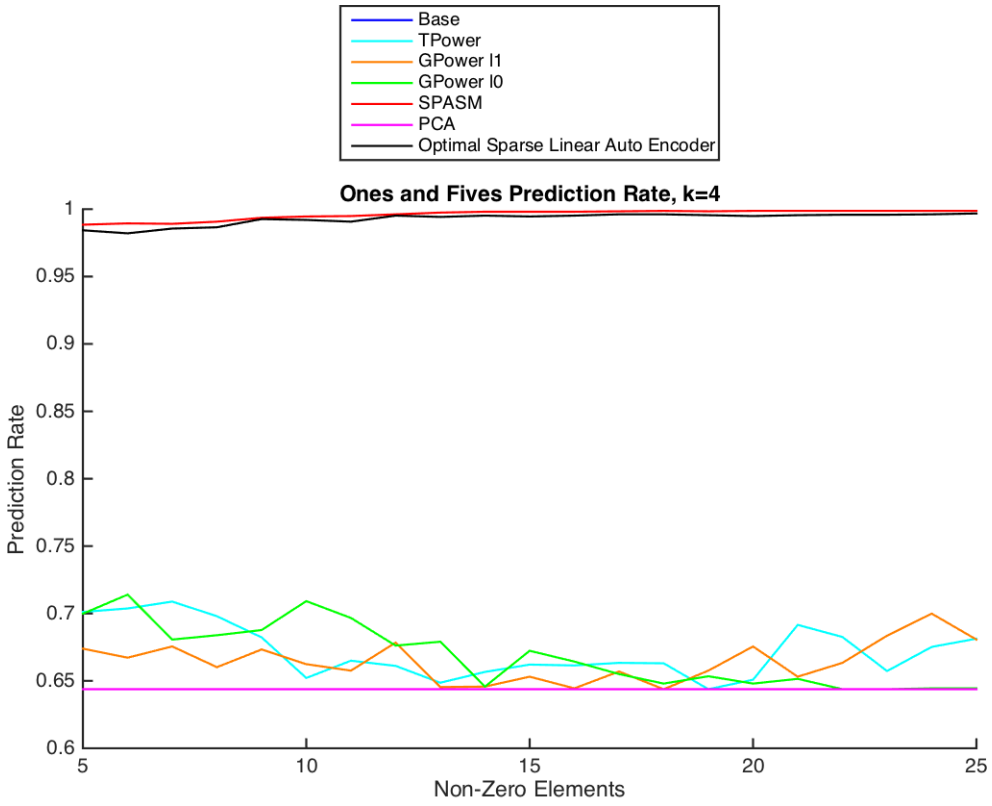

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Ones and Fives Variance, k=2
Variance
Non-Zero Elements
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
5
10
15
20
25

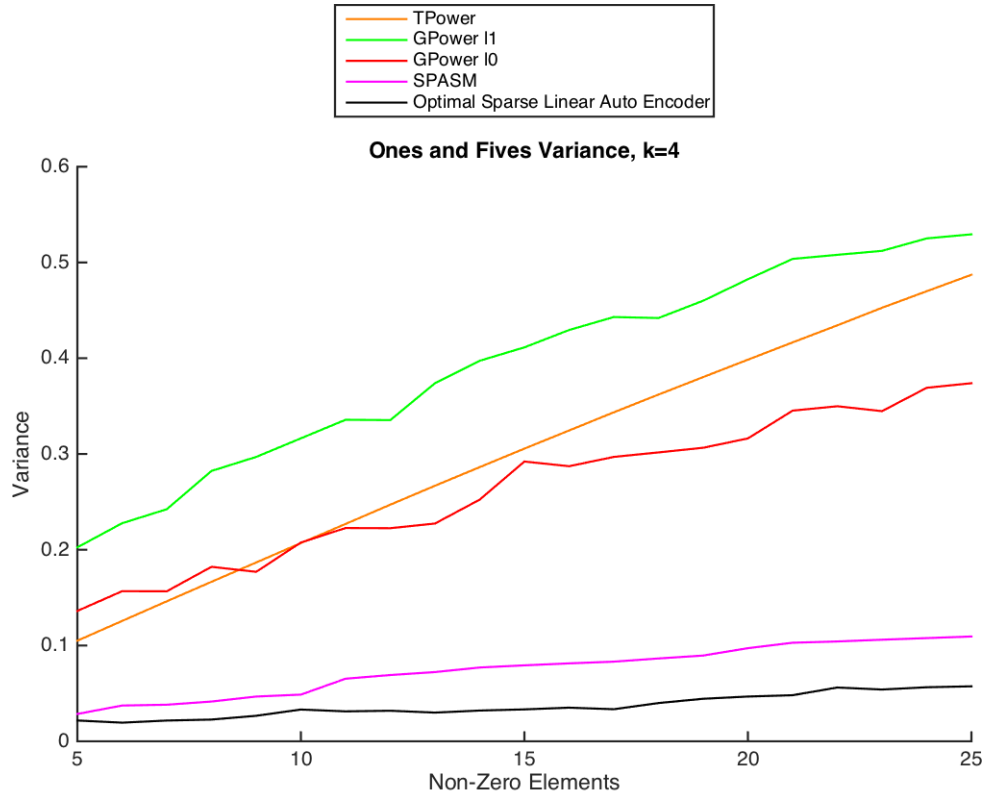

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Ones and Fives Info Loss, k=3
Info Loss
Non-Zero Elements
1.9
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1
0
5
10
15
20
25

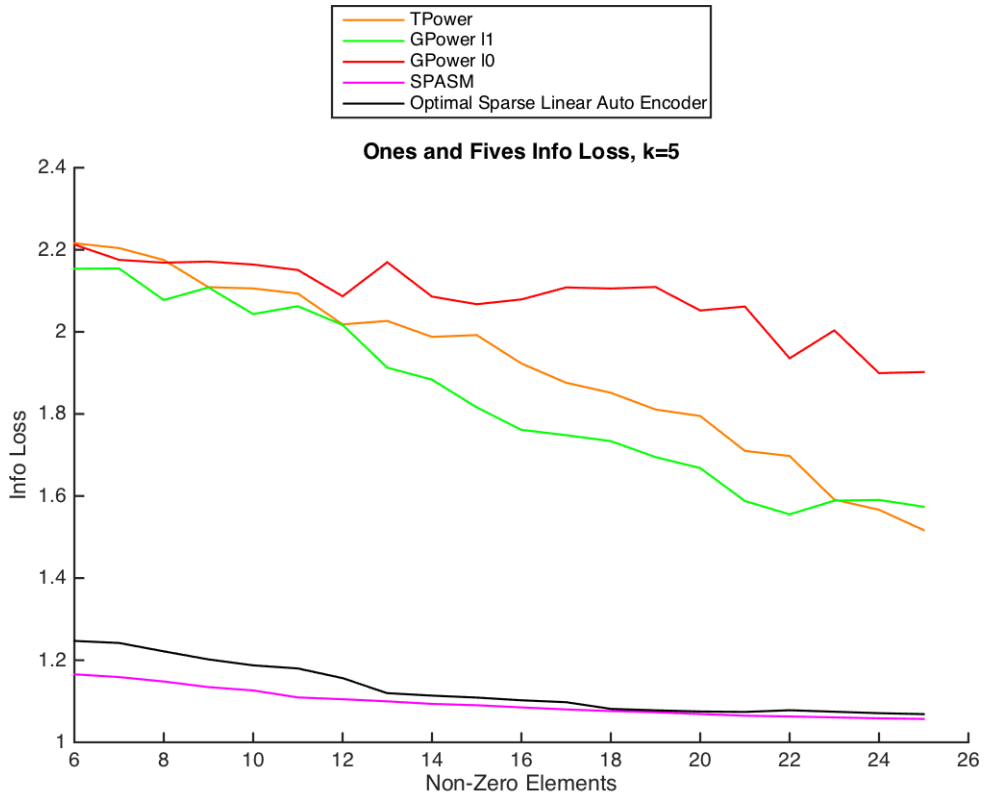

Base
TPower
GPower l1
GPower l0
SPASM
PCA
Optimal Sparse Linear Auto Encoder
Ones and Fives Prediction Rate, k=3
Prediction Rate
Non-Zero Elements
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0
5
10
15
20
25

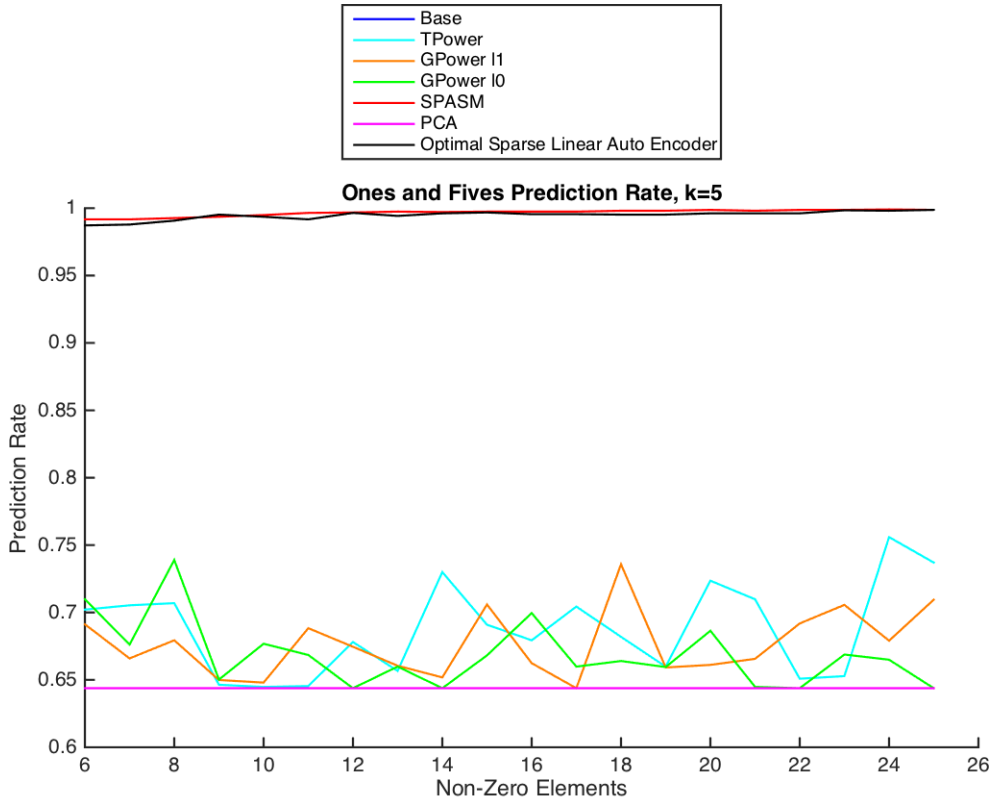

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Ones and Fives Variance, k=3
Variance
Non-Zero Elements
0
0.1
0.2
0.3
0.4
0.5
0.6
0
5
10
15
20
25

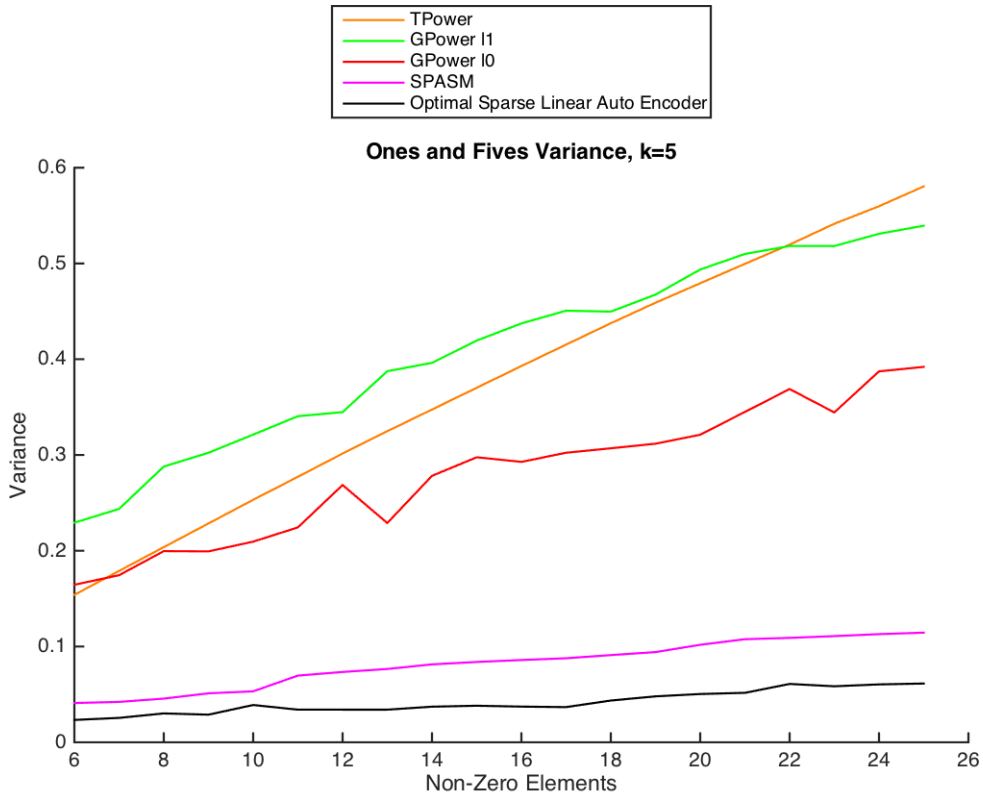

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Ones and Fives Info Loss, k=4
Info Loss
Non-Zero Elements
1
1.2
1.4
1.6
1.8
2
2.2
5
10
15
20
25

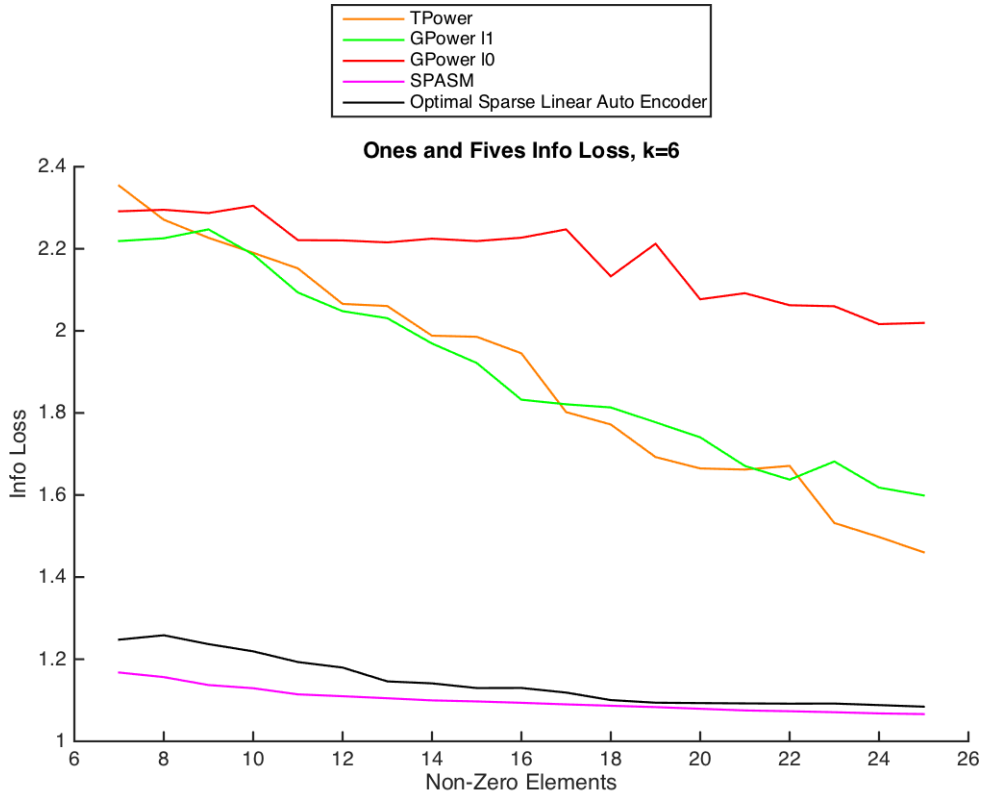

Base
TPower
GPower l1
GPower l0
SPASM
PCA
Optimal Sparse Linear Auto Encoder
Ones and Fives Prediction Rate, k=4
Prediction Rate
Non-Zero Elements
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
5
10
15
20
25

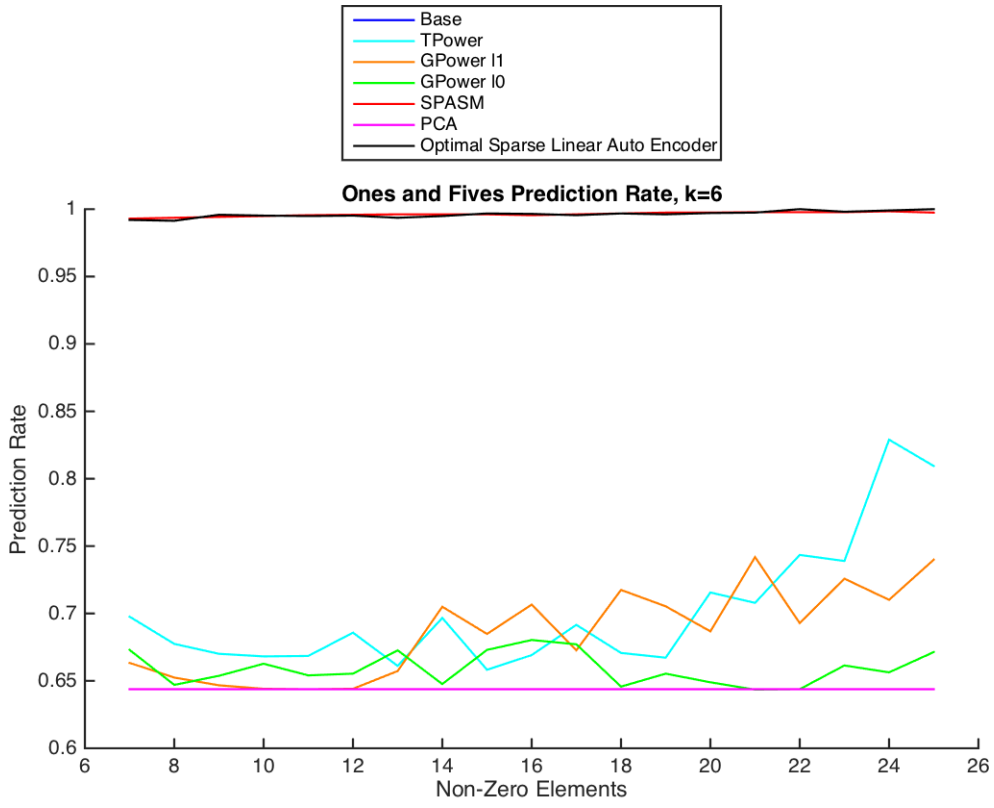

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Ones and Fives Variance, k=4
Variance
Non-Zero Elements
0.6
0.5
0.4
0.3
0.2
0.1
0
5
10
15
20
25

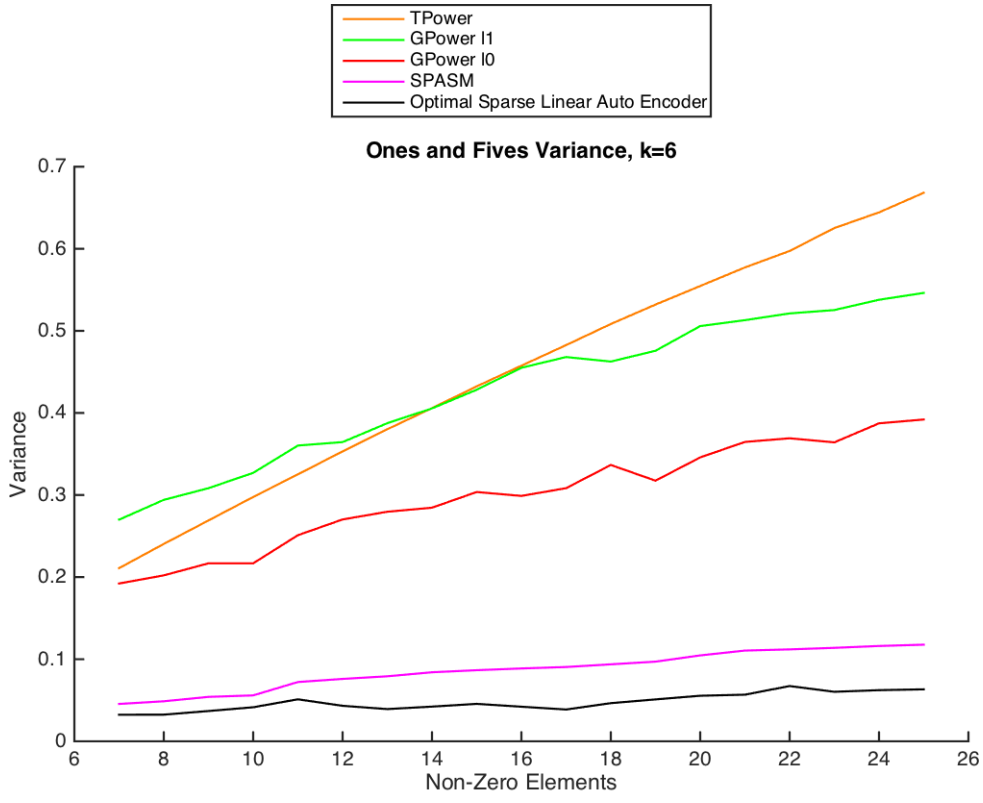

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Ones and Fives Info Loss, k=5
Info Loss
Non-Zero Elements
2.4
2.2
2
1.8
1.6
1.4
1.2
1
6
8
10
12
14
16
18
20
22
24
26


Base
TPower
GPower l1
GPower l0
SPASM
PCA
Optimal Sparse Linear Auto Encoder
Ones and Fives Prediction Rate, k=5
Prediction Rate
Non-Zero Elements
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
6
8
10
12
14
16
18
20
22
24
26

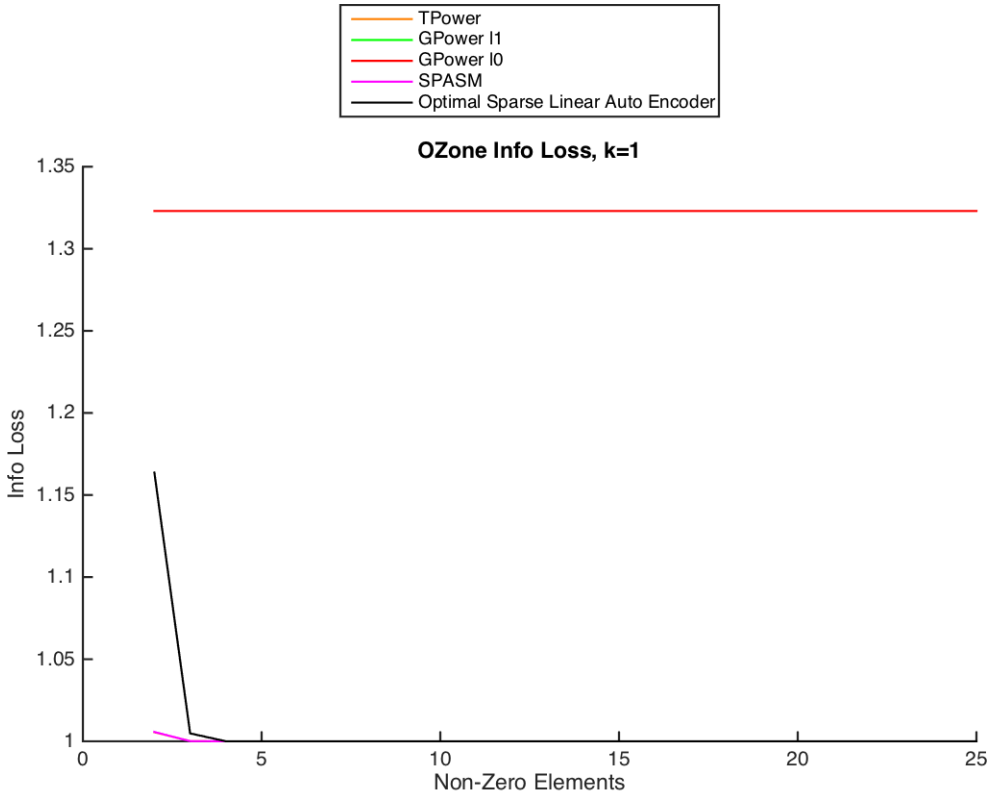

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Ones and Fives Variance, k=5
Variance
Non-Zero Elements
0
0.1
0.2
0.3
0.4
0.5
0.6
6
8
10
12
14
16
18
20
22
24
26

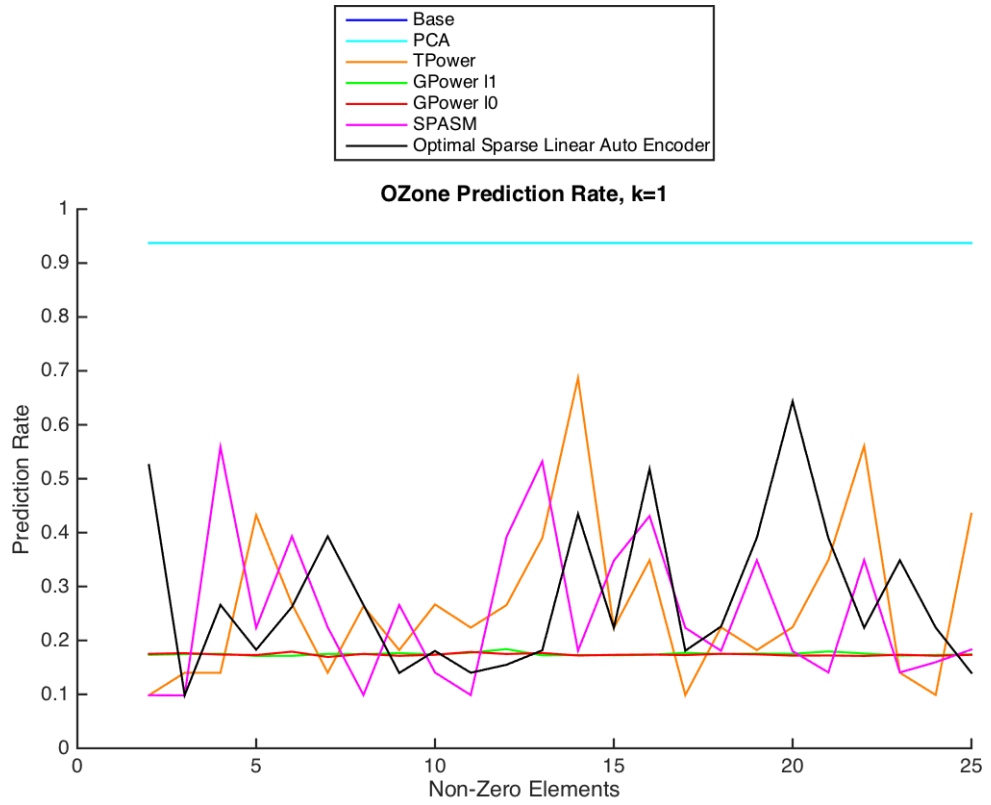

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Ones and Fives Info Loss, k=6
Info Loss
Non-Zero Elements
1
1.2
1.4
1.6
1.8
2
2.2
2.4
6
8
10
12
14
16
18
20
22
24
26

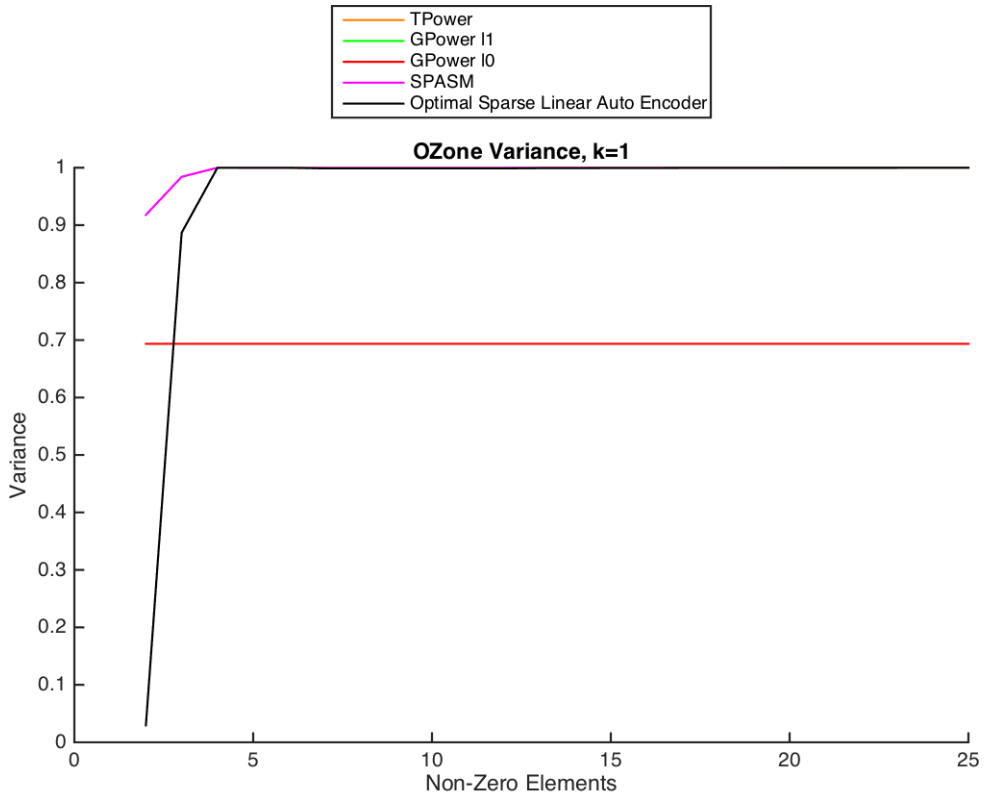

Base
TPower
GPower l1
GPower l0
SPASM
PCA
Optimal Sparse Linear Auto Encoder
Ones and Fives Prediction Rate, k=6
Prediction Rate
Non-Zero Elements
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
6
8
10
12
14
16
18
20
22
24
26

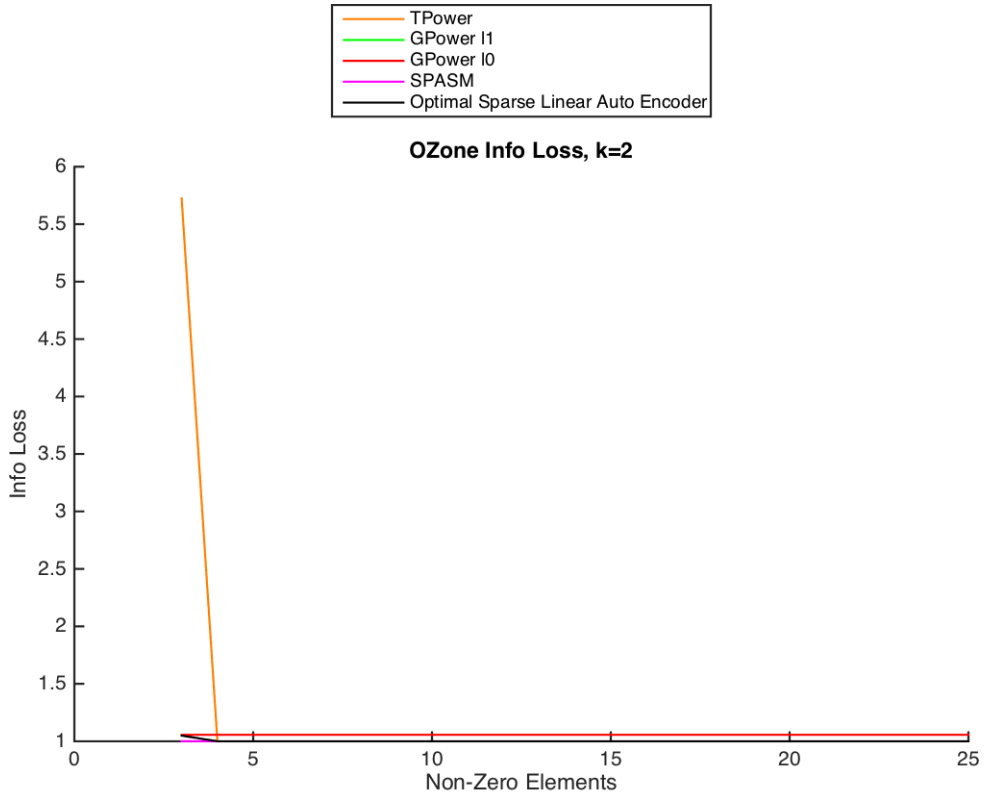

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Ones and Fives Variance, k=6
Variance
Non-Zero Elements
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
6
8
10
12
14
16
18
20
22
24
26

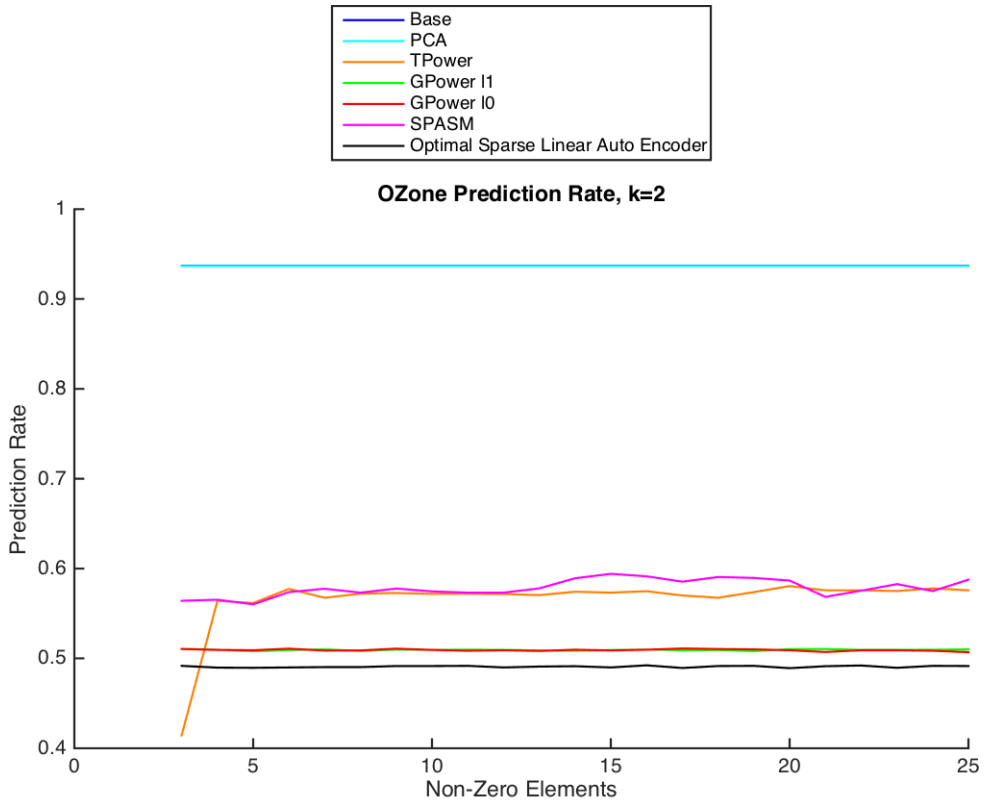
A.6 Ozone Results

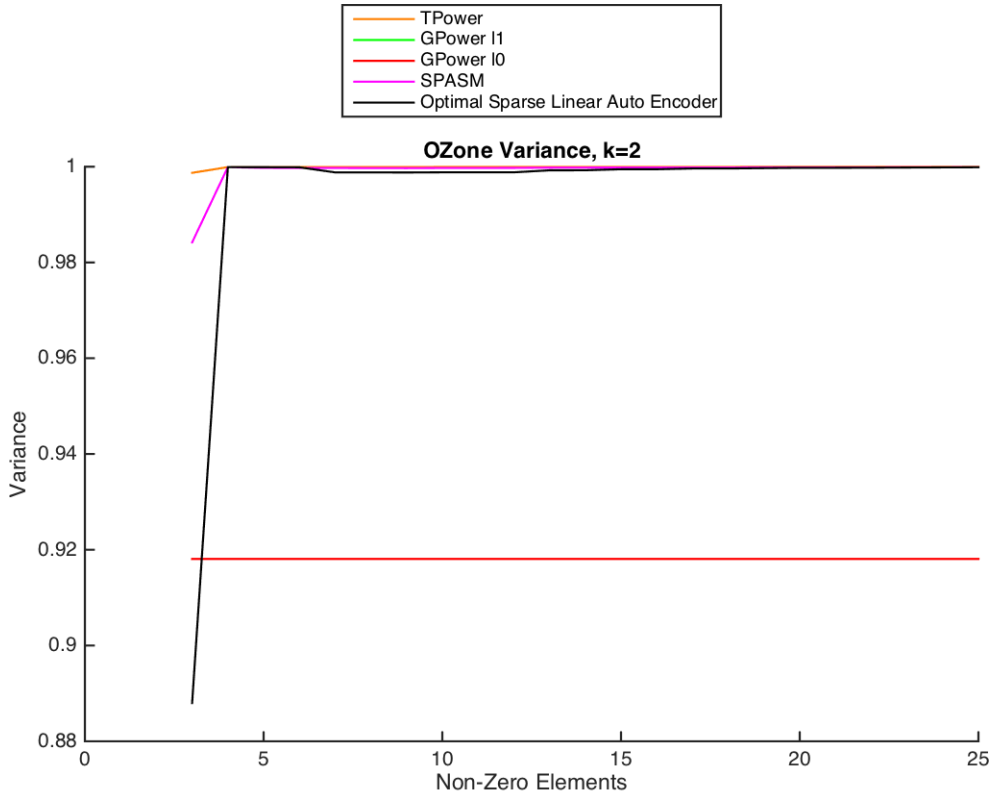


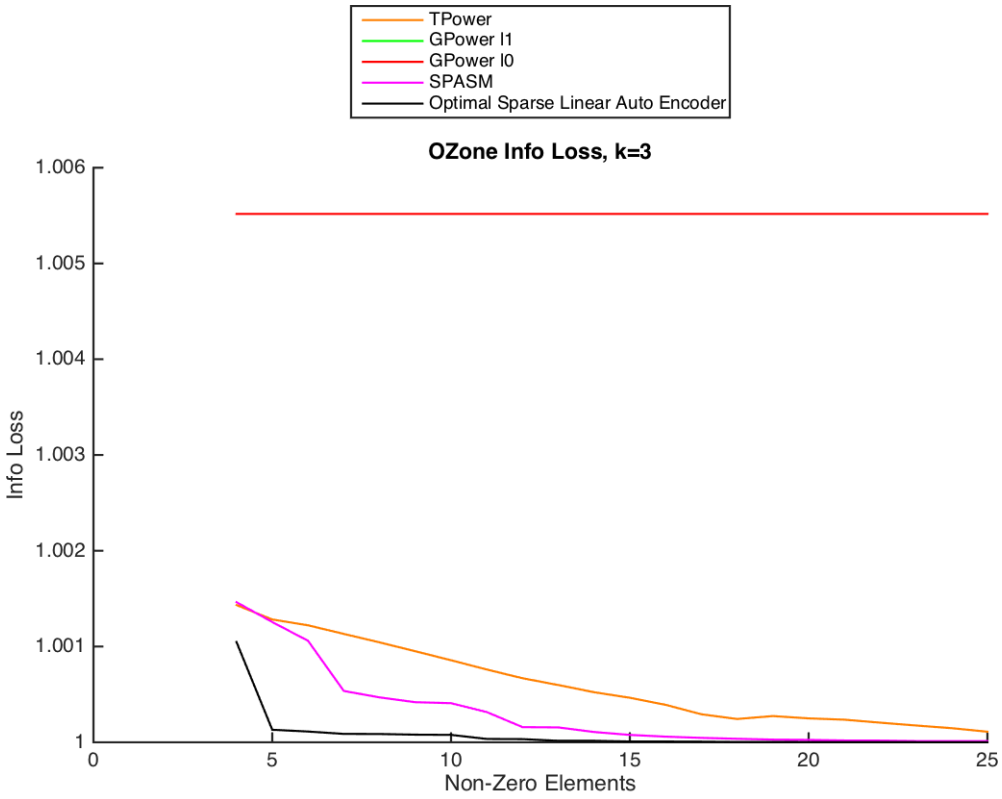


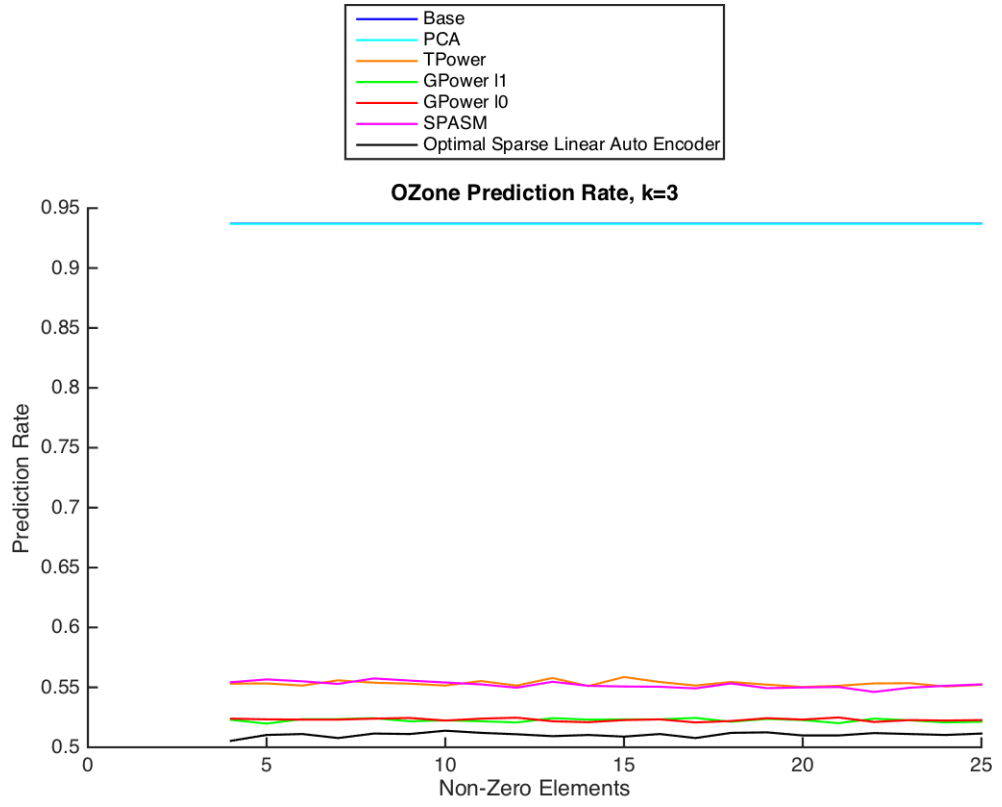

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
OZone Variance, k=1
Variance
Non-Zero Elements
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
5
10
15
20
25

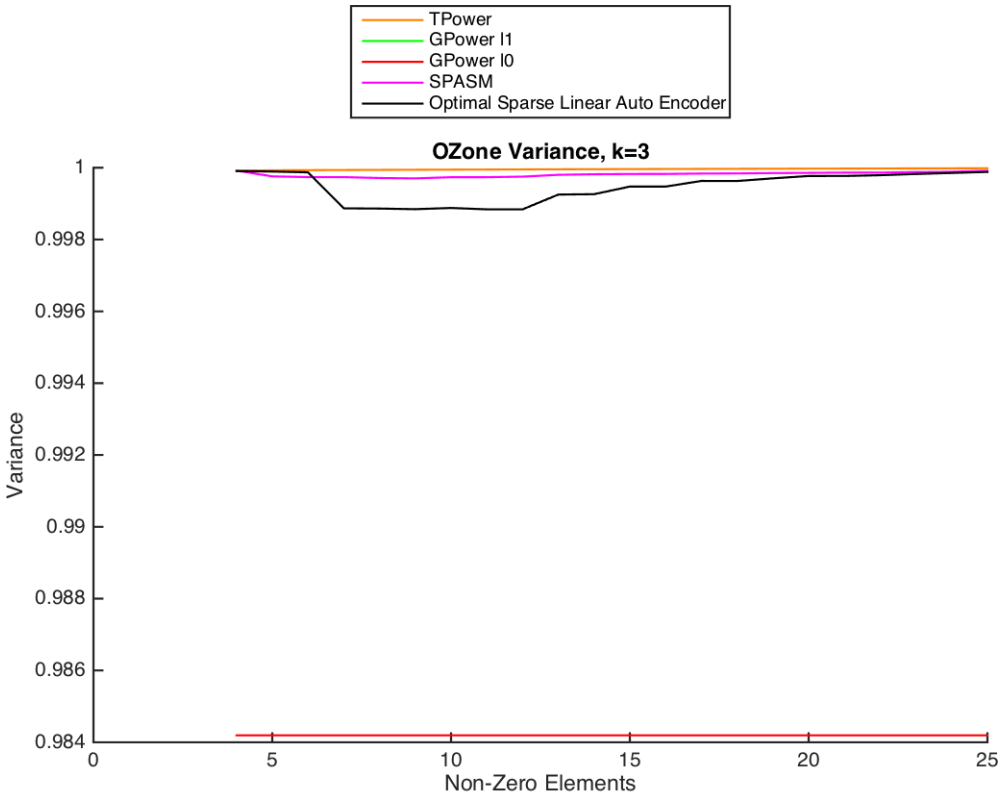

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
OZone Info Loss, k=2
Info Loss
Non-Zero Elements
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
0
5
10
15
20
25

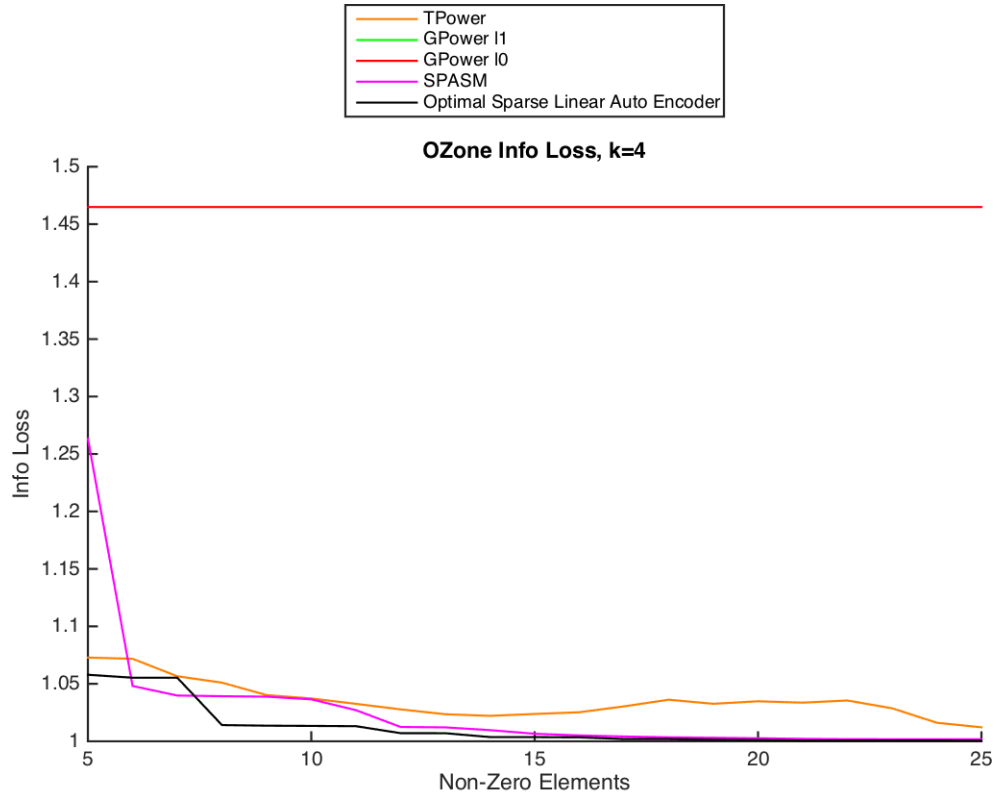

Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
OZone Prediction Rate, k=2
Prediction Rate
Non-Zero Elements
1
0.9
0.8
0.7
0.6
0.5
0.4
0
5
10
15
20
25

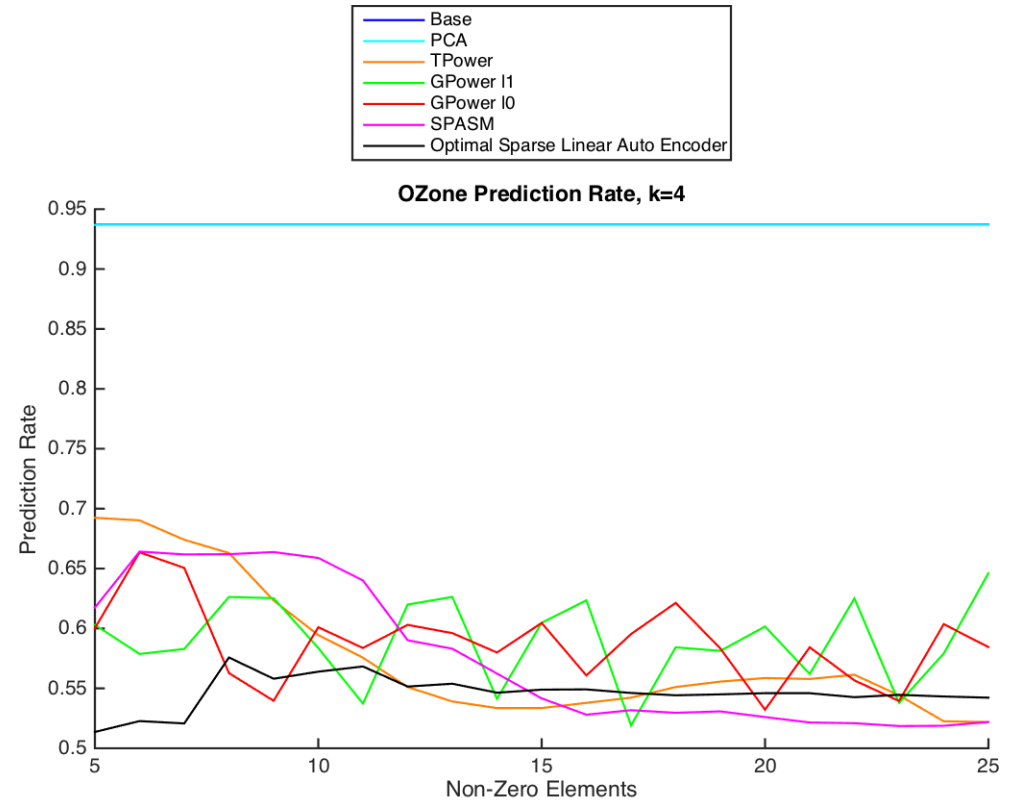

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
OZone Variance, k=2
Variance
Non-Zero Elements
1
0.98
0.96
0.94
0.92
0.9
0.88
0
5
10
15
20
25

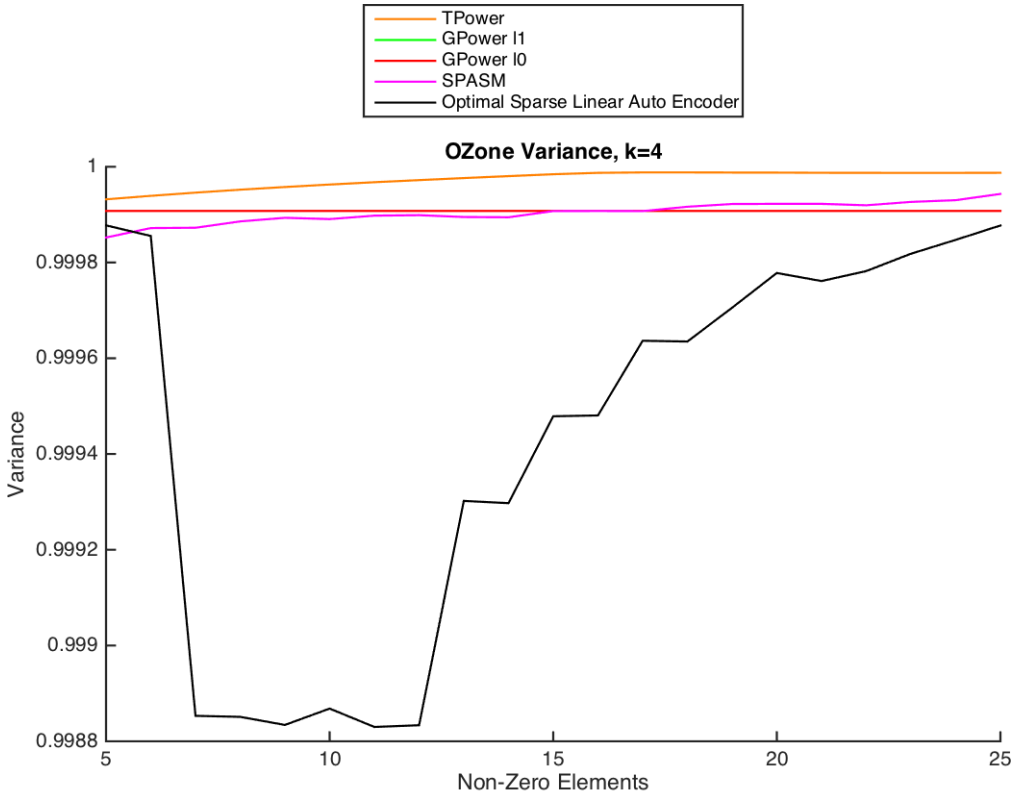

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
OZone Info Loss, k=3
Info Loss
Non-Zero Elements
1.006
1.005
1.004
1.003
1.002
1.001
1
0
5
10
15
20
25

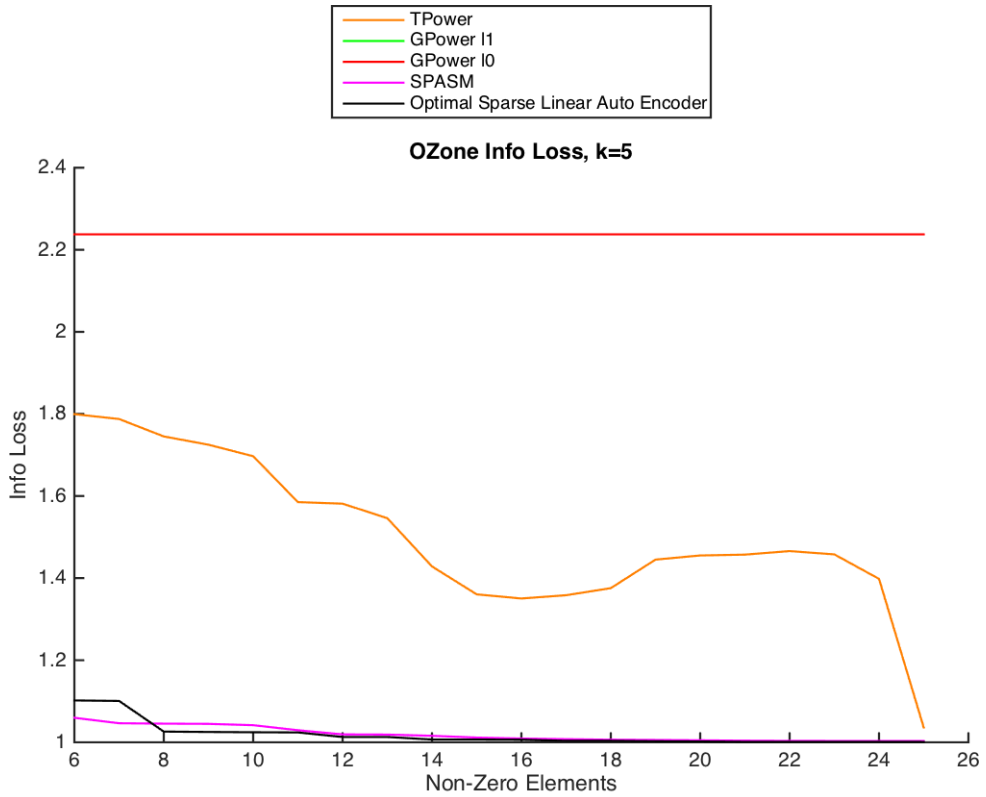

Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
OZone Prediction Rate, k=3
Prediction Rate
Non-Zero Elements
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0
5
10
15
20
25

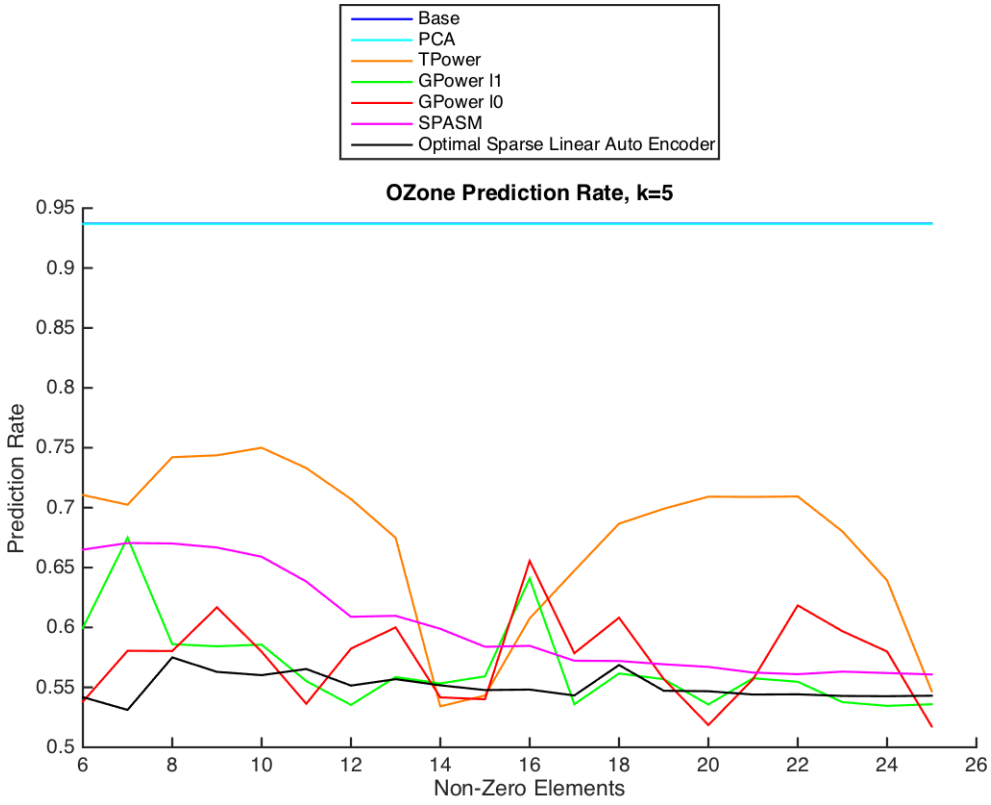

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
OZone Variance, k=3
Variance
Non-Zero Elements
1
0.998
0.996
0.994
0.992
0.99
0.988
0.986
0.984
0
5
10
15
20
25

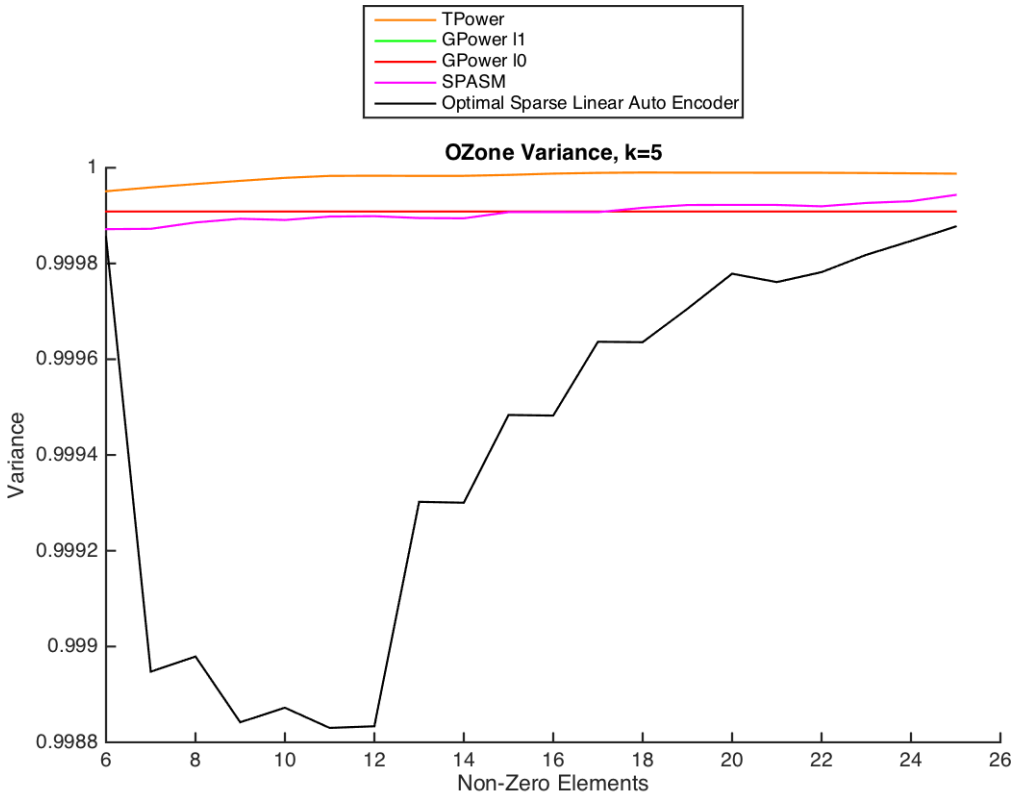

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
OZone Info Loss, k=4
Info Loss
Non-Zero Elements
1.5
1.45
1.4
1.35
1.3
1.25
1.2
1.15
1.1
1.05
1
5
10
15
20
25

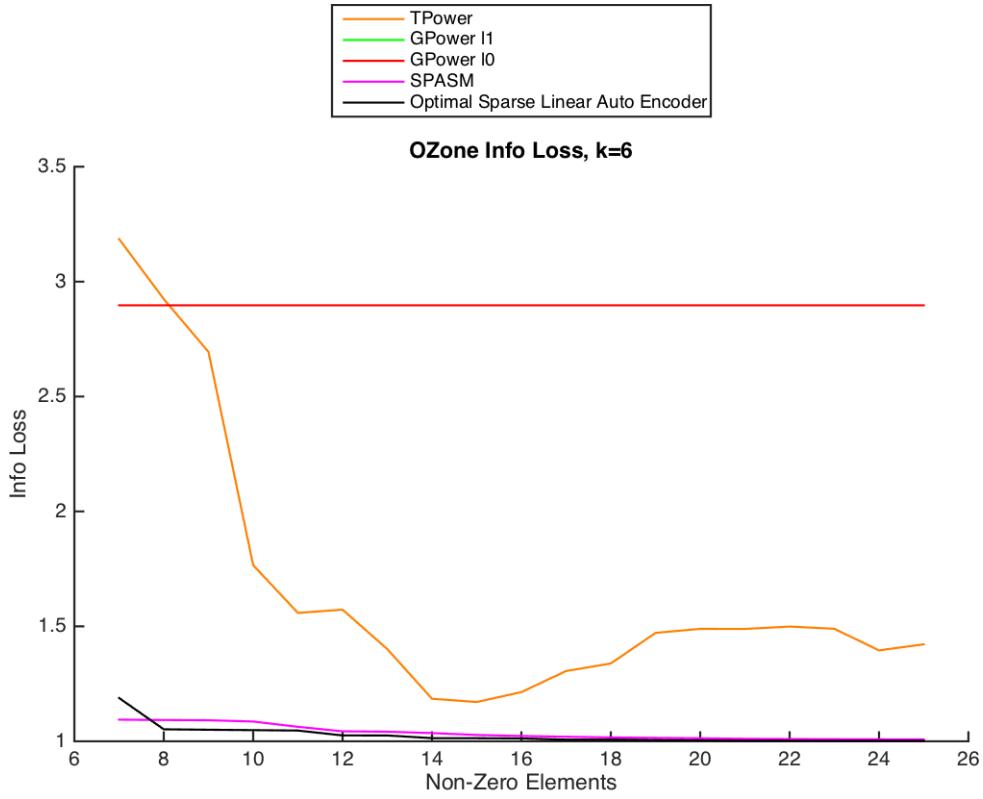

Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
OZone Prediction Rate, k=4
Prediction Rate
Non-Zero Elements
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
5
10
15
20
25

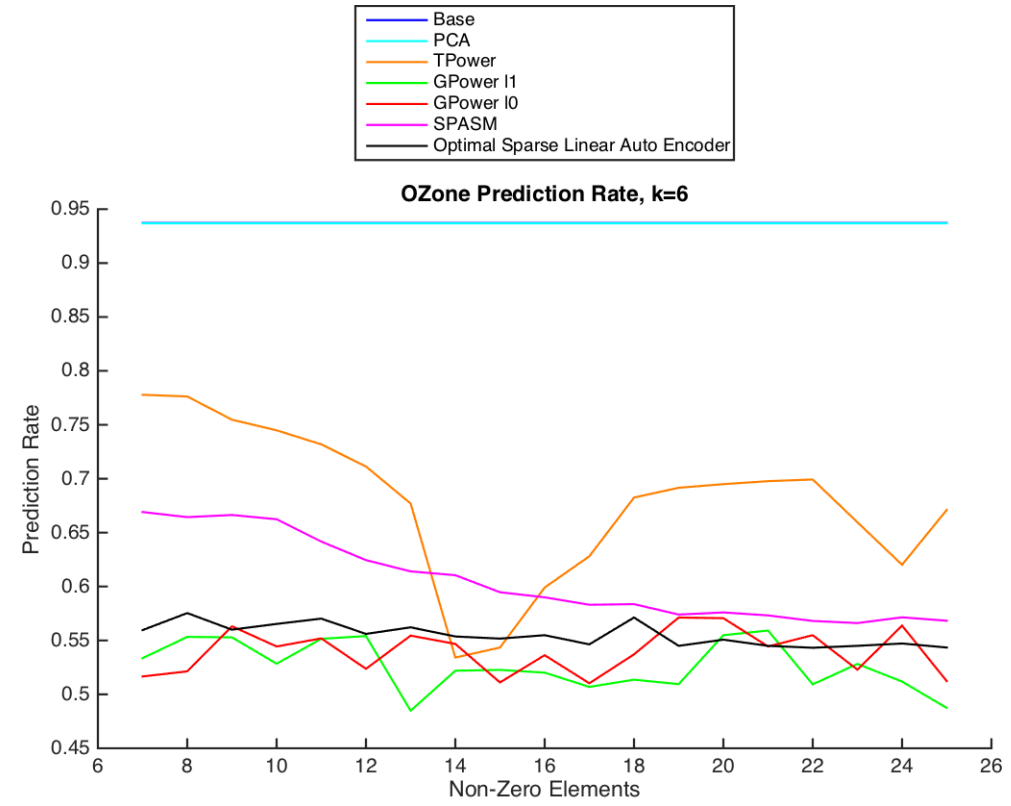

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
OZone Variance, k=4
Variance
Non-Zero Elements
1
0.9998
0.9996
0.9994
0.9992
0.999
0.9988
5
10
15
20
25

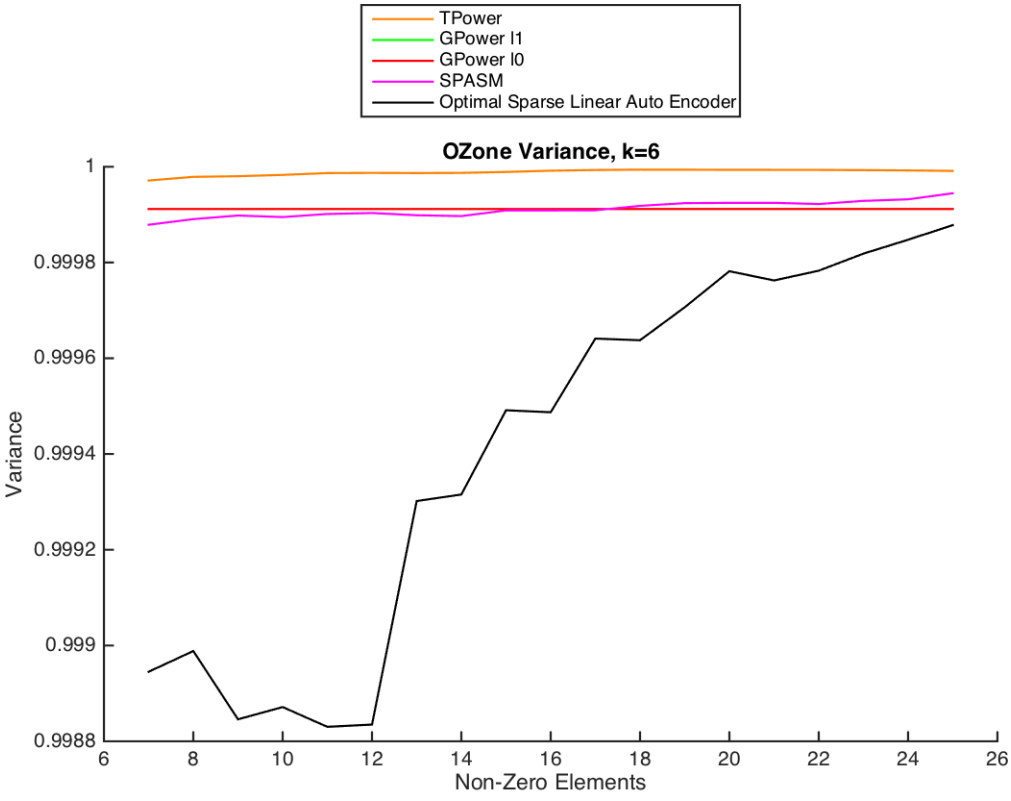

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
OZone Info Loss, k=5
Info Loss
Non-Zero Elements
2.4
2.2
2
1.8
1.6
1.4
1.2
1
6
8
10
12
14
16
18
20
22
24
26


Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
OZone Prediction Rate, k=5
Prediction Rate
Non-Zero Elements
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
6
8
10
12
14
16
18
20
22
24
26

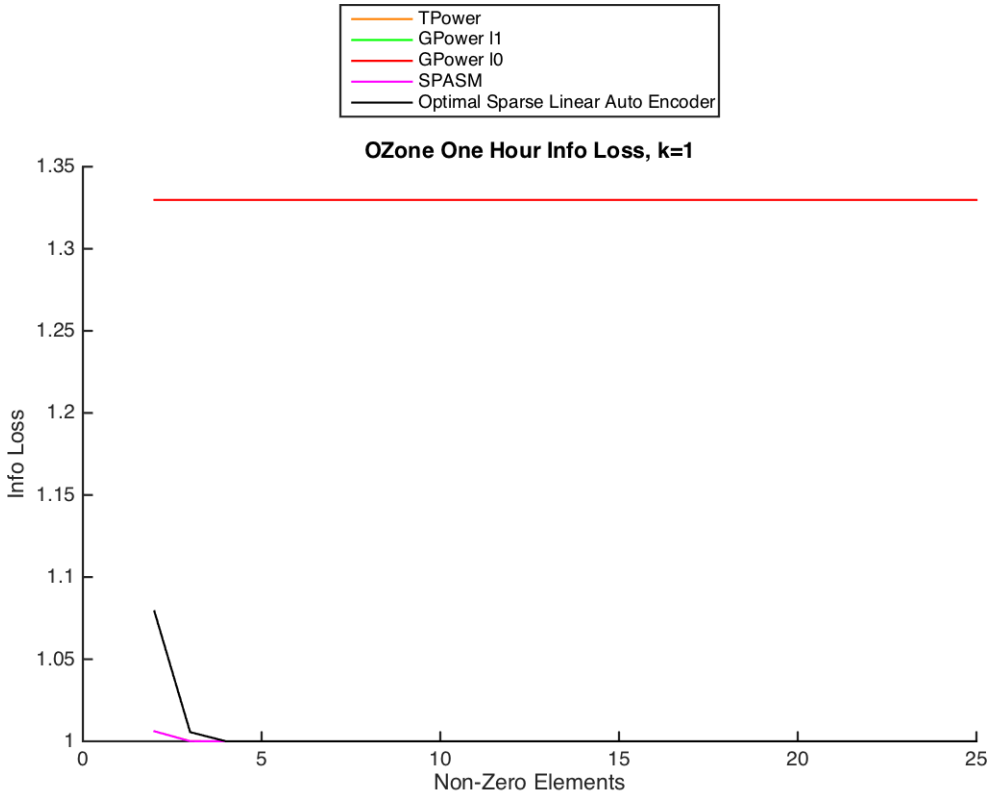

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
OZone Variance, k=5
Variance
Non-Zero Elements
1
0.9998
0.9996
0.9994
0.9992
0.999
0.9988
6
8
10
12
14
16
18
20
22
24
26

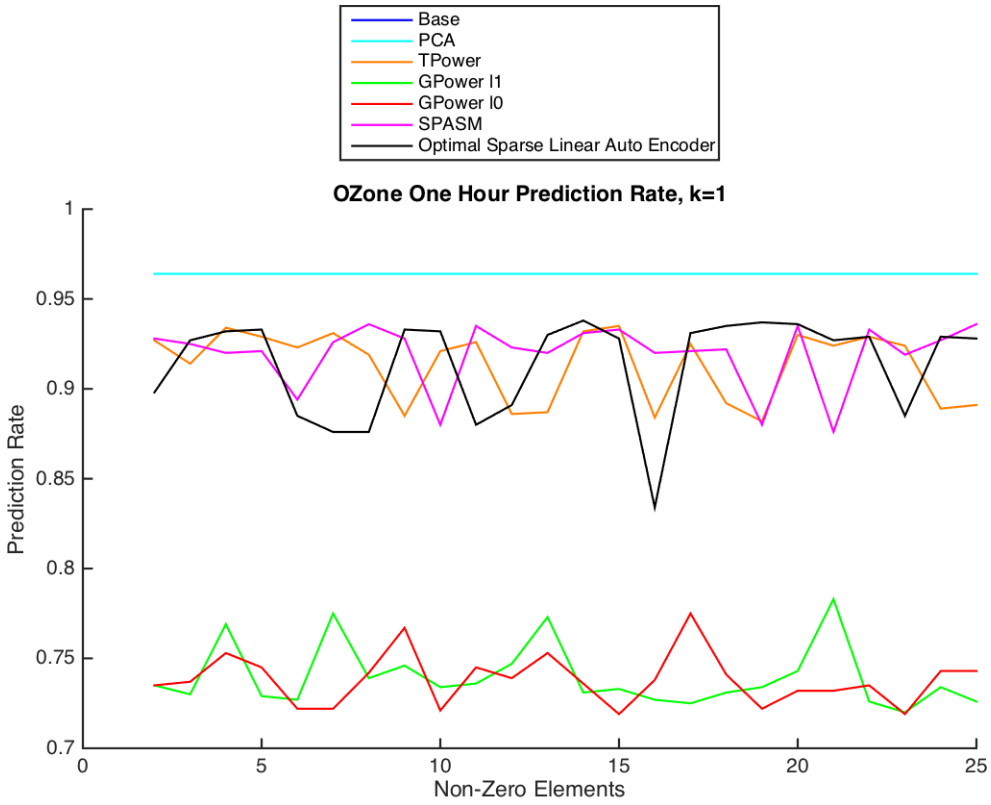

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
OZone Info Loss, k=6
Info Loss
Non-Zero Elements
3.5
3
2.5
2
1.5
1
6
8
10
12
14
16
18
20
22
24
26

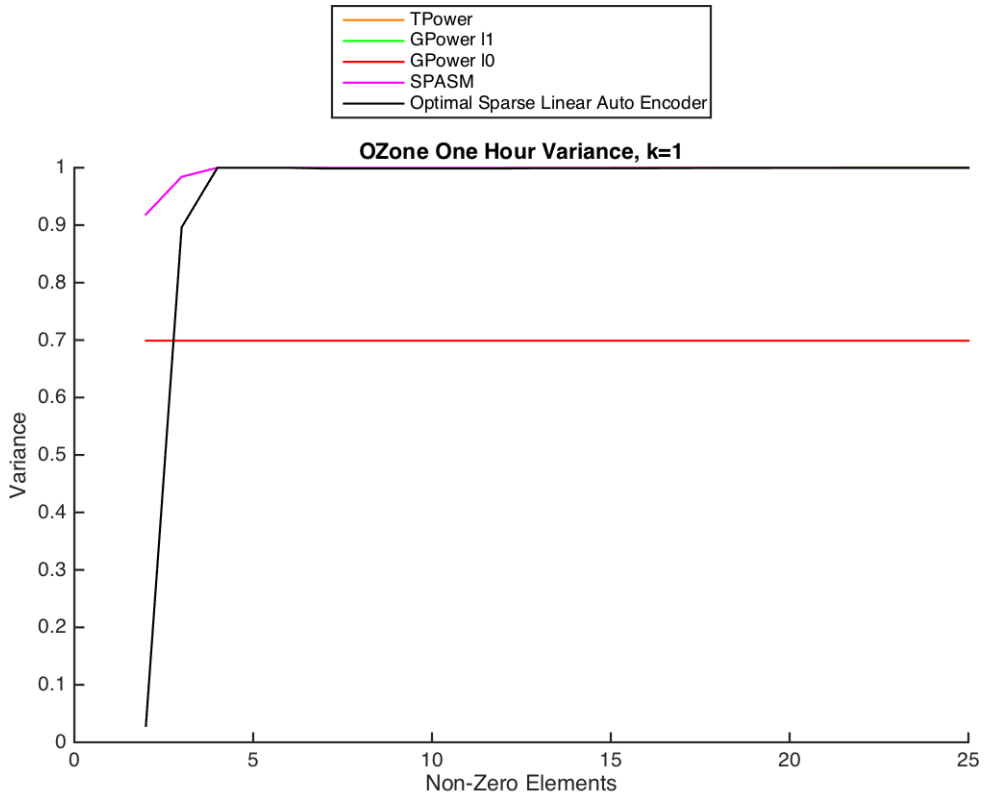

Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
OZone Prediction Rate, k=6
Prediction Rate
Non-Zero Elements
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0.45
6
8
10
12
14
16
18
20
22
24
26

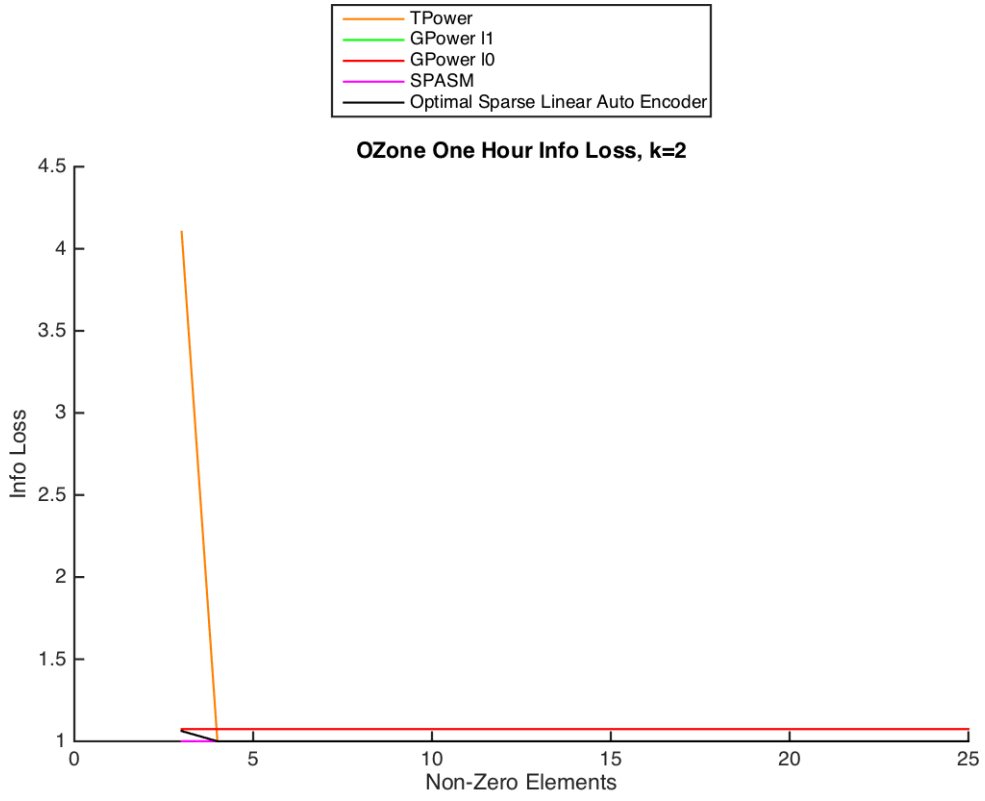

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
OZone Variance, k=6
Variance
Non-Zero Elements
1
0.9998
0.9996
0.9994
0.9992
0.999
0.9988
6
8
10
12
14
16
18
20
22
24
26

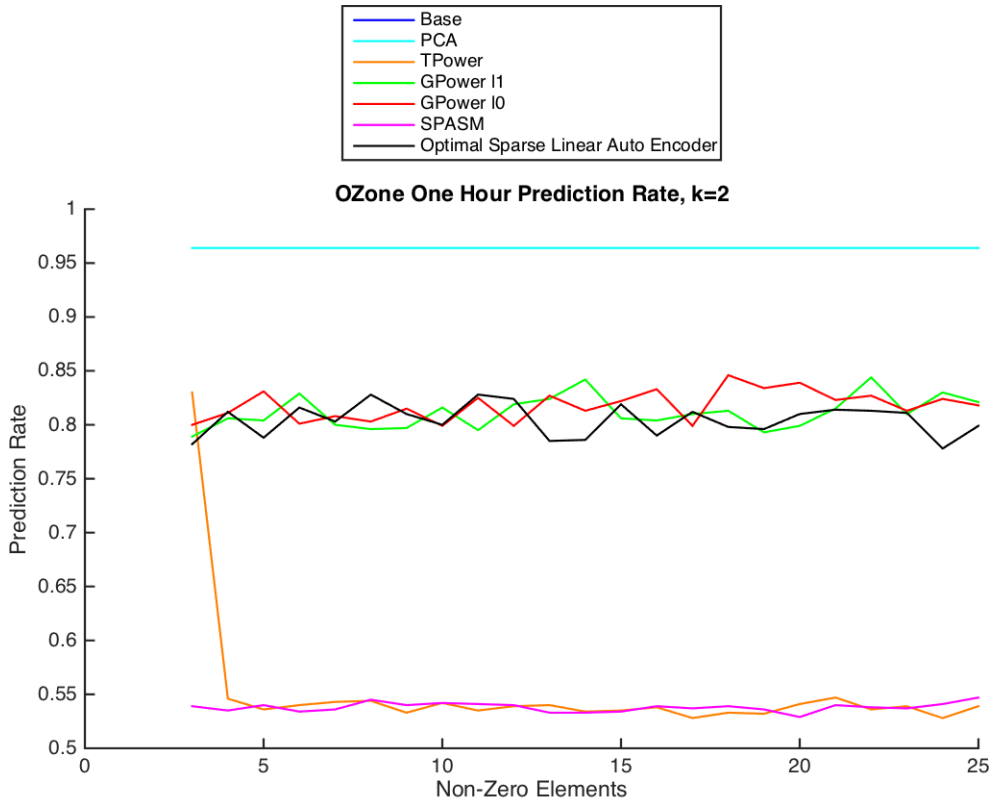
A.7 Ozone One Hour Results

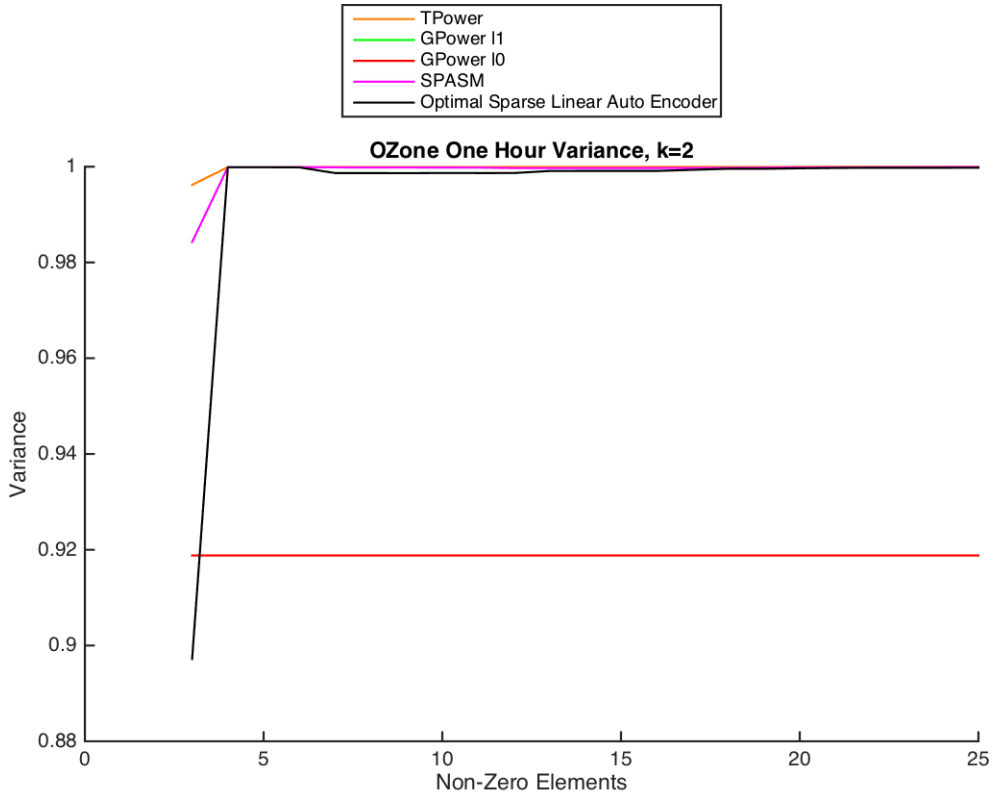


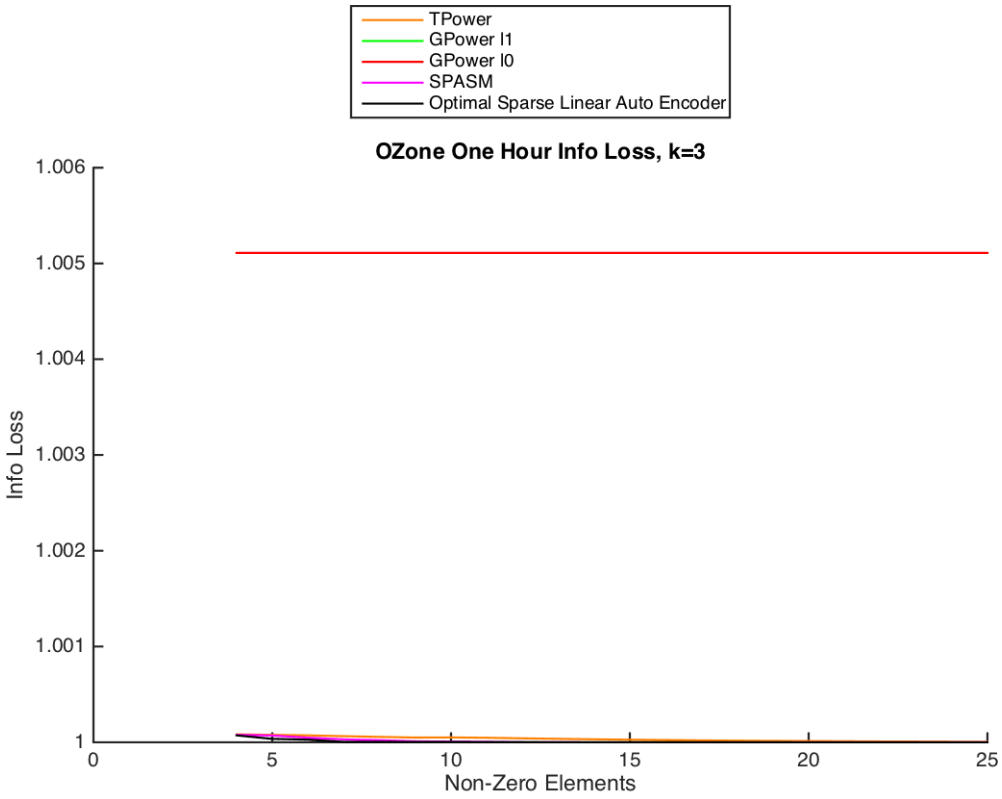


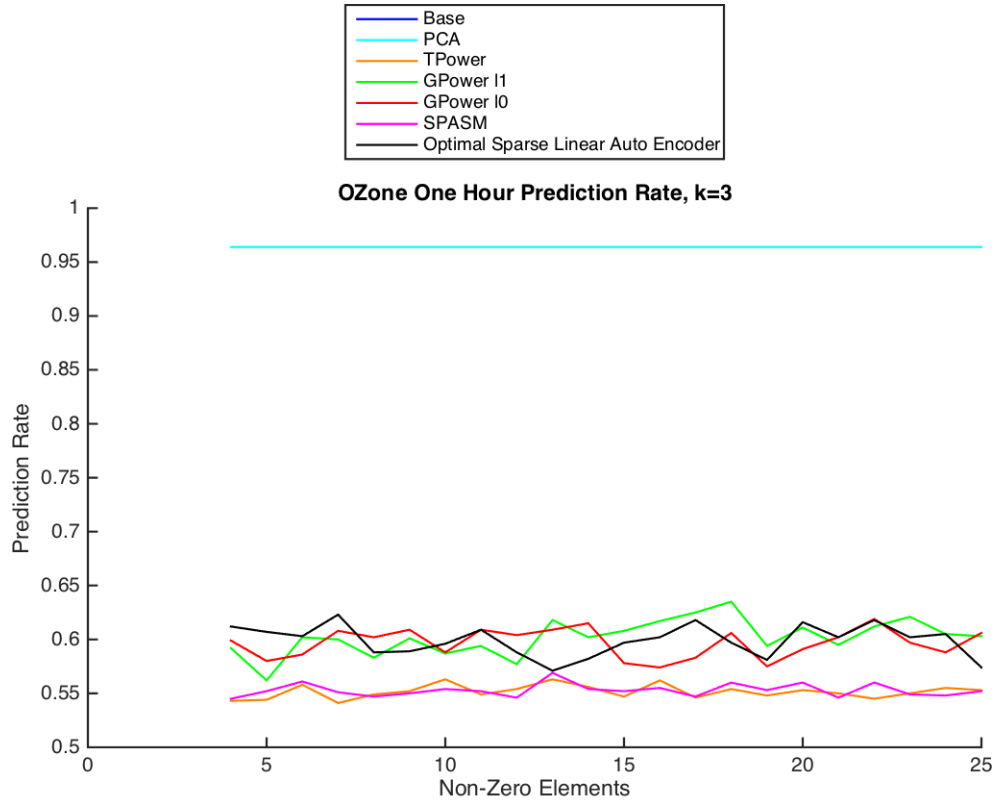

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
OZone One Hour Variance, k=1
Variance
Non-Zero Elements
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
5
10
15
20
25

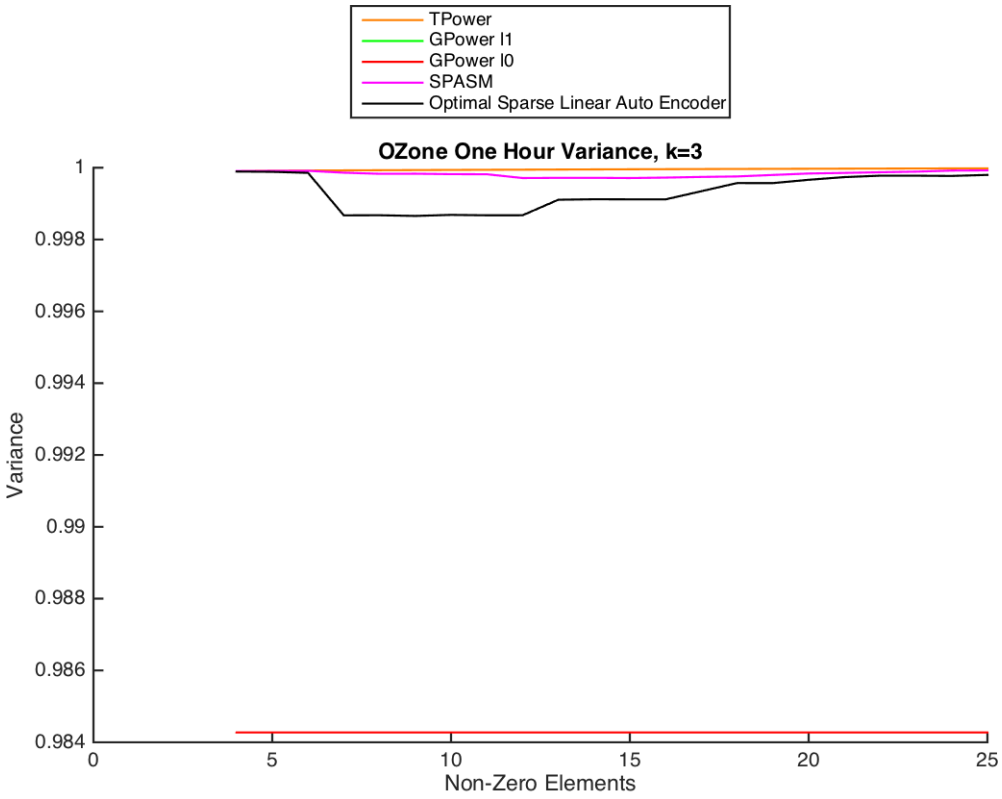

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
OZone One Hour Info Loss, k=2
Info Loss
Non-Zero Elements
1
1.5
2
2.5
3
3.5
4
4.5
0
5
10
15
20
25

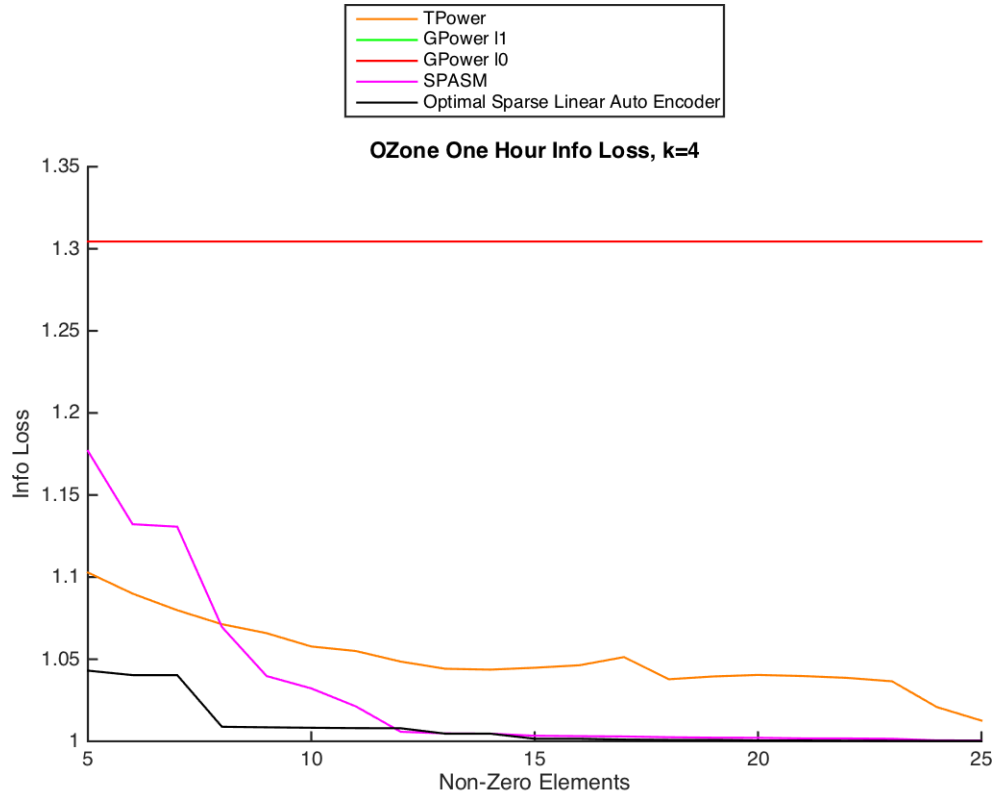

Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
OZone One Hour Prediction Rate, k=2
Prediction Rate
Non-Zero Elements
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0
5
10
15
20
25

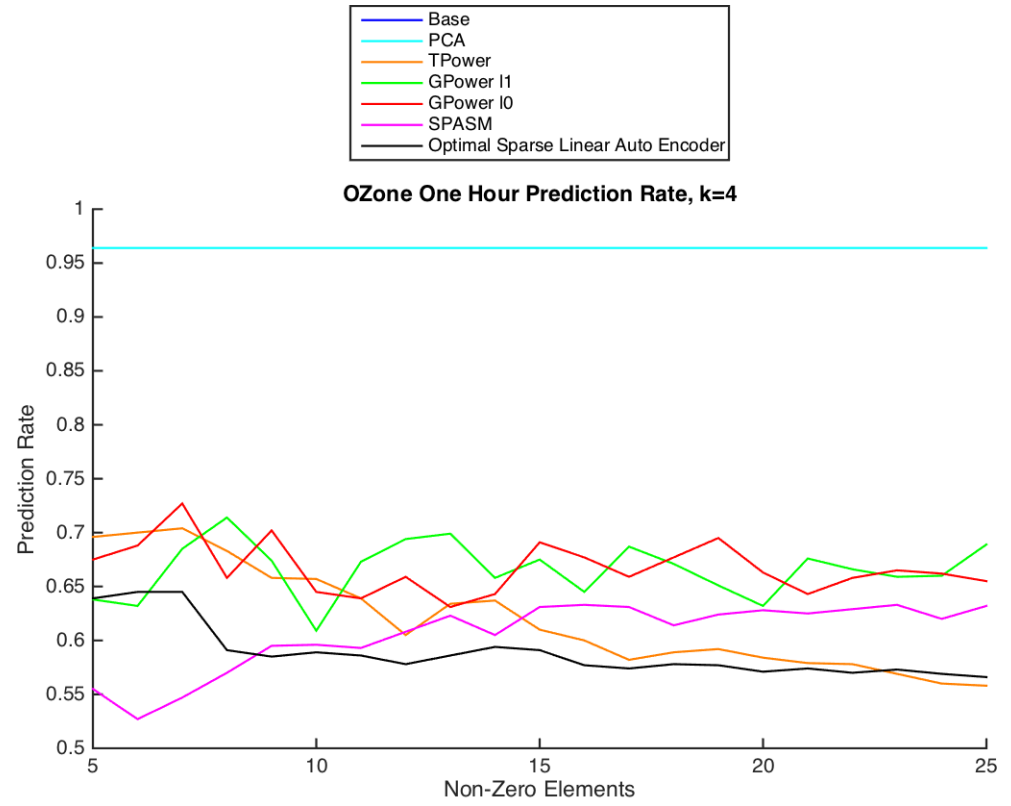

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
OZone One Hour Variance, k=2
Variance
Non-Zero Elements
1
0.98
0.96
0.94
0.92
0.9
0.88
0
5
10
15
20
25

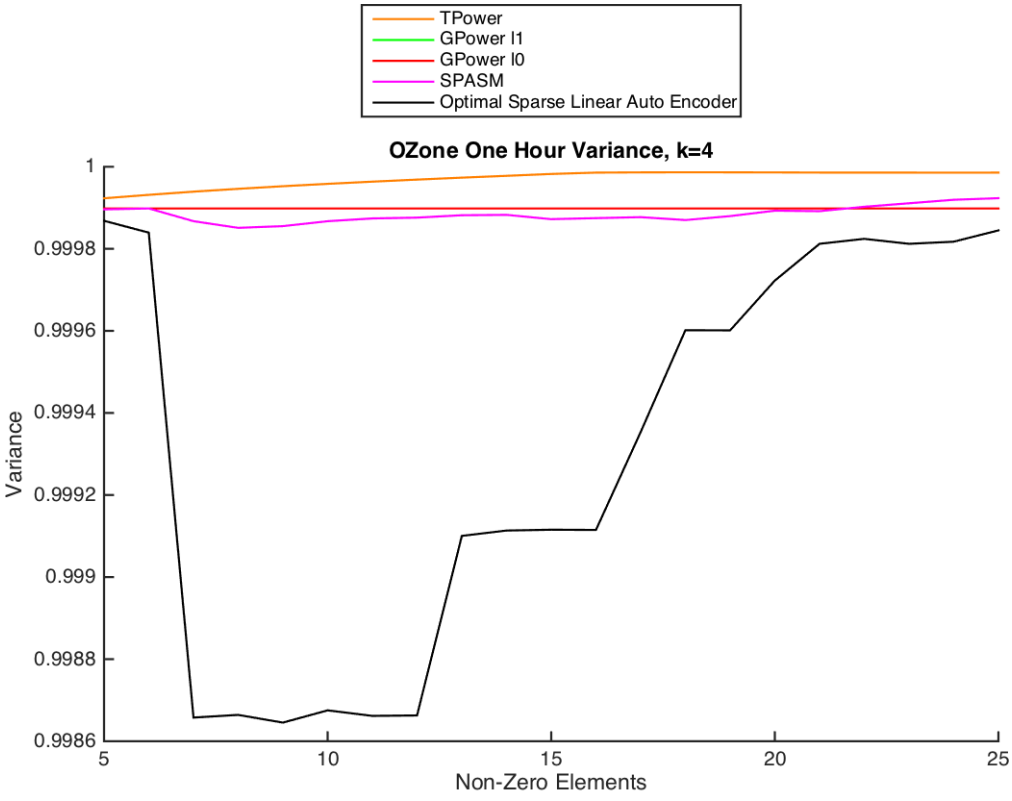

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
OZone One Hour Info Loss, k=3
Info Loss
Non-Zero Elements
1.006
1.005
1.004
1.003
1.002
1.001
1
0
5
10
15
20
25

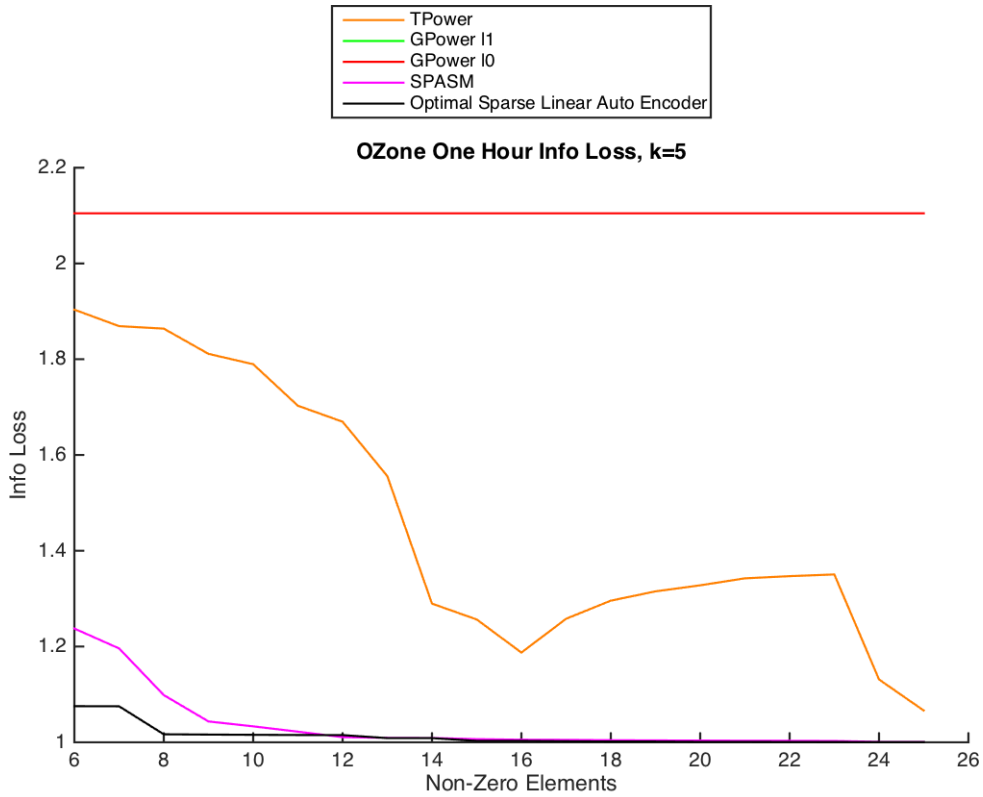

Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
OZone One Hour Prediction Rate, k=3
Prediction Rate
Non-Zero Elements
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0
5
10
15
20
25

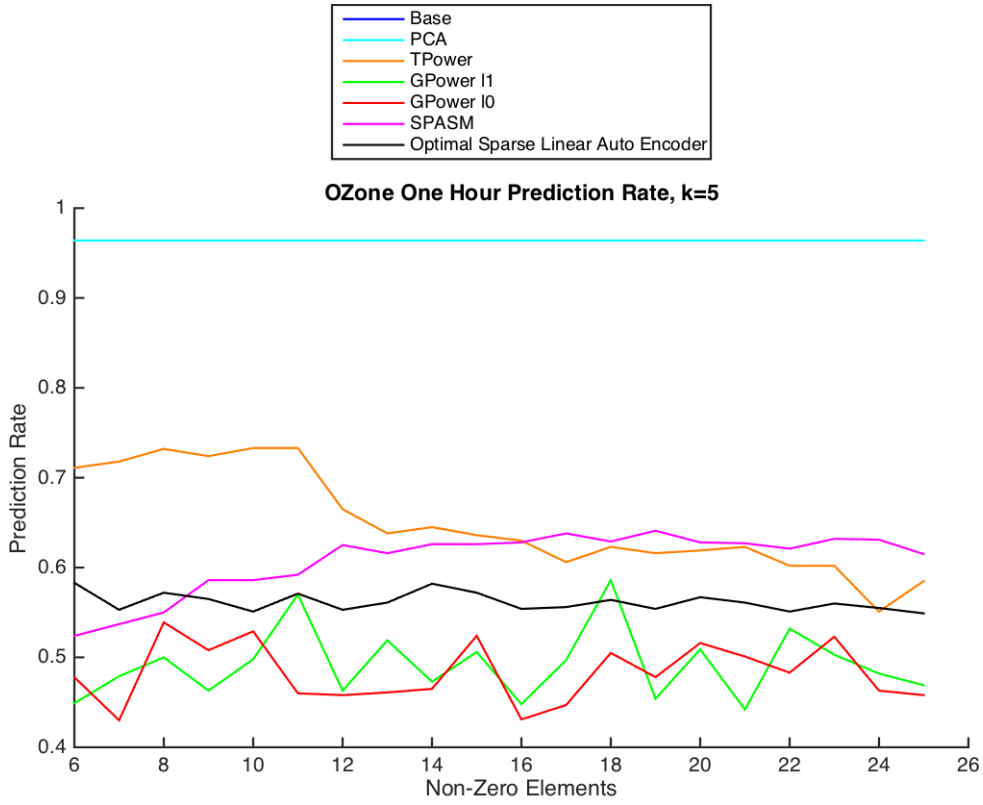

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
OZone One Hour Variance, k=3
Variance
Non-Zero Elements
1
0.998
0.996
0.994
0.992
0.99
0.988
0.986
0.984
0
5
10
15
20
25

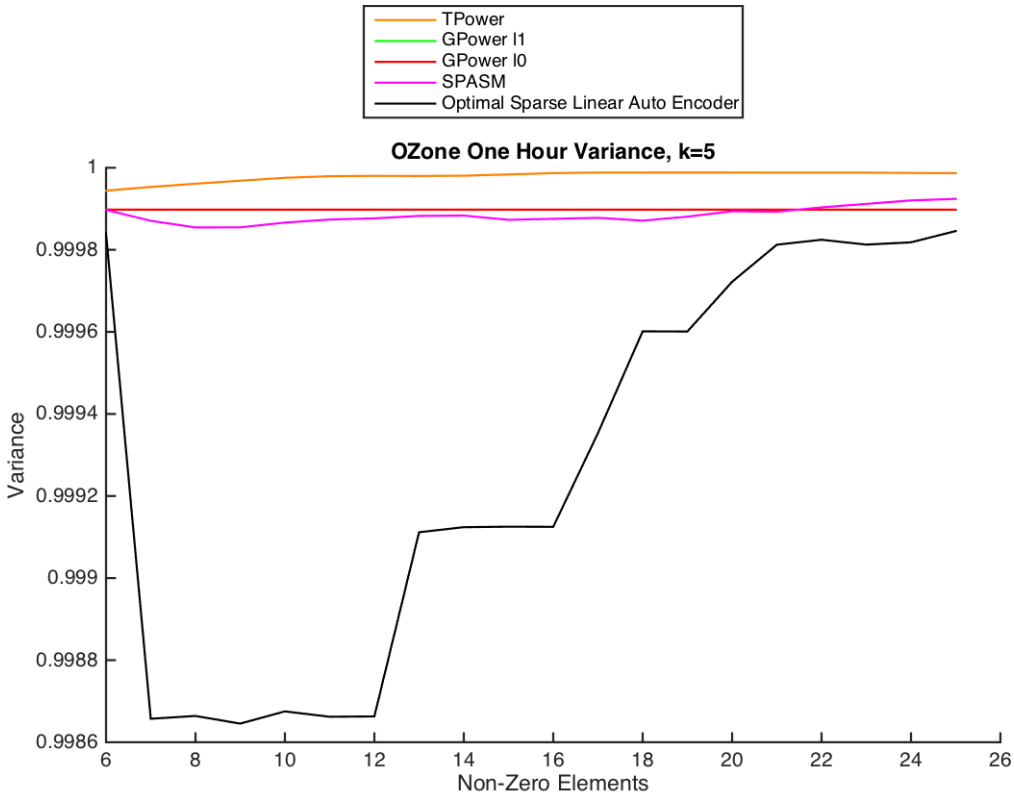

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
OZone One Hour Info Loss, k=4
Info Loss
Non-Zero Elements
1.35
1.3
1.25
1.2
1.15
1.1
1.05
1
5
10
15
20
25

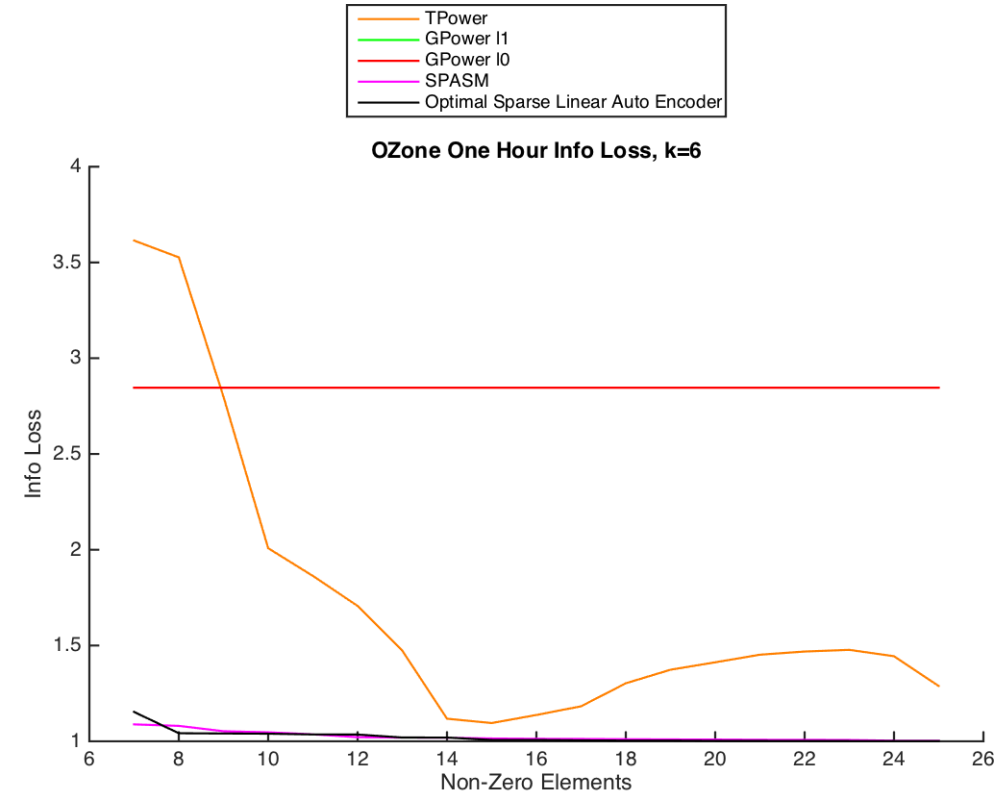

Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
OZone One Hour Prediction Rate, k=4
Prediction Rate
Non-Zero Elements
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
5
10
15
20
25

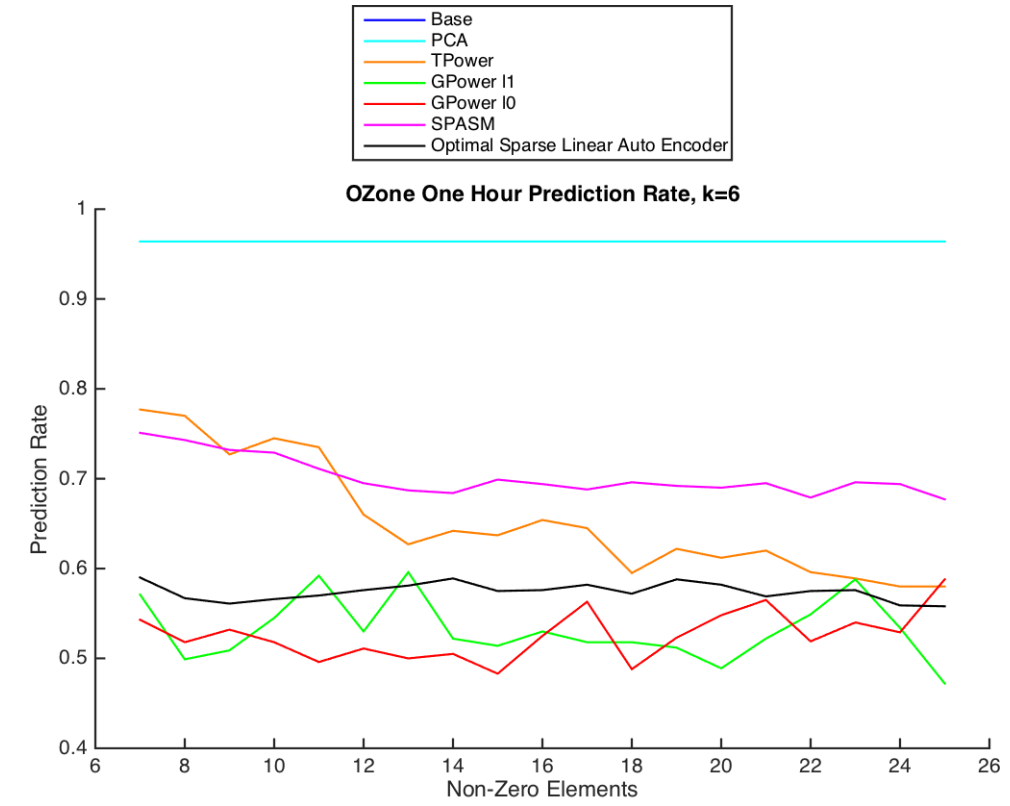

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
OZone One Hour Variance, k=4
Variance
Non-Zero Elements
1
0.9998
0.9996
0.9994
0.9992
0.999
0.9988
0.9986
5
10
15
20
25

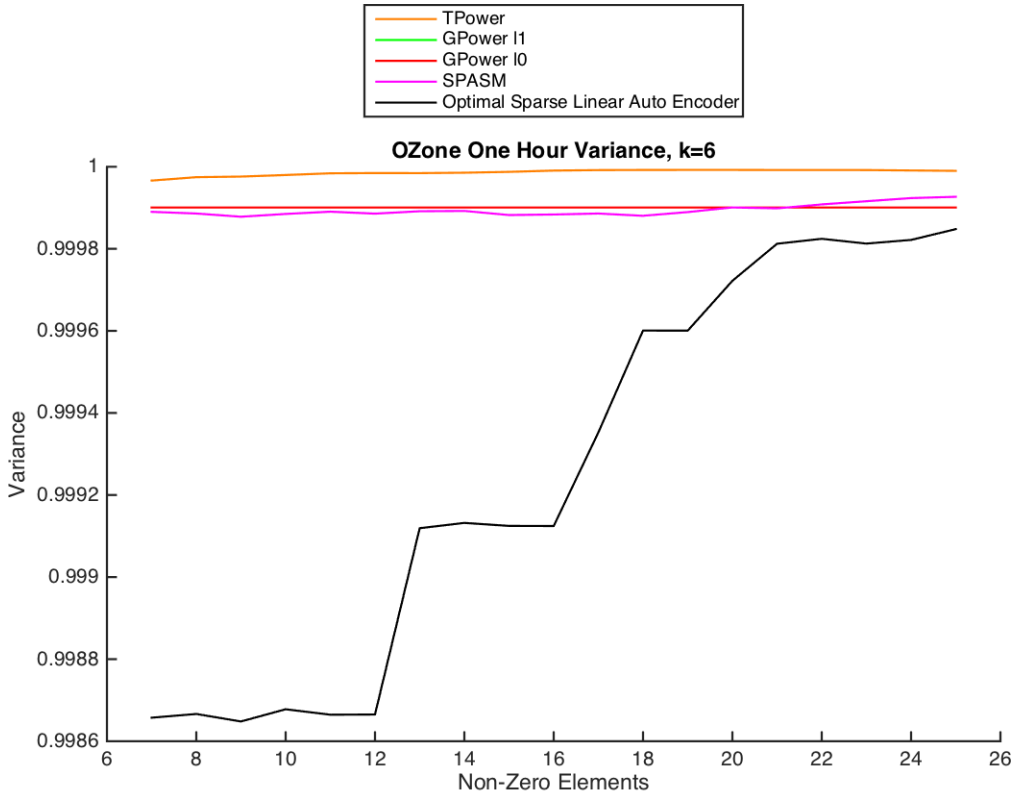

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
OZone One Hour Info Loss, k=5
Info Loss
Non-Zero Elements
2.2
2
1.8
1.6
1.4
1.2
1
6
8
10
12
14
16
18
20
22
24
26


Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
OZone One Hour Prediction Rate, k=5
Prediction Rate
Non-Zero Elements
1
0.9
0.8
0.7
0.6
0.5
0.4
6
8
10
12
14
16
18
20
22
24
26

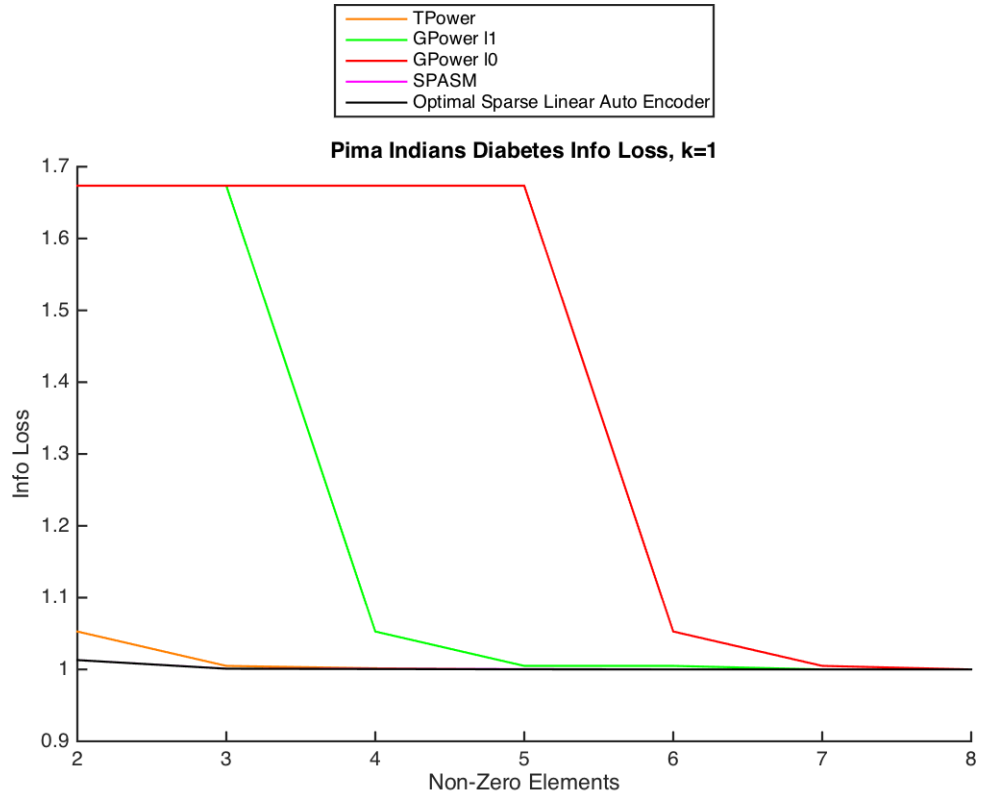

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
OZone One Hour Variance, k=5
Variance
Non-Zero Elements
1
0.9998
0.9996
0.9994
0.9992
0.999
0.9988
0.9986
6
8
10
12
14
16
18
20
22
24
26

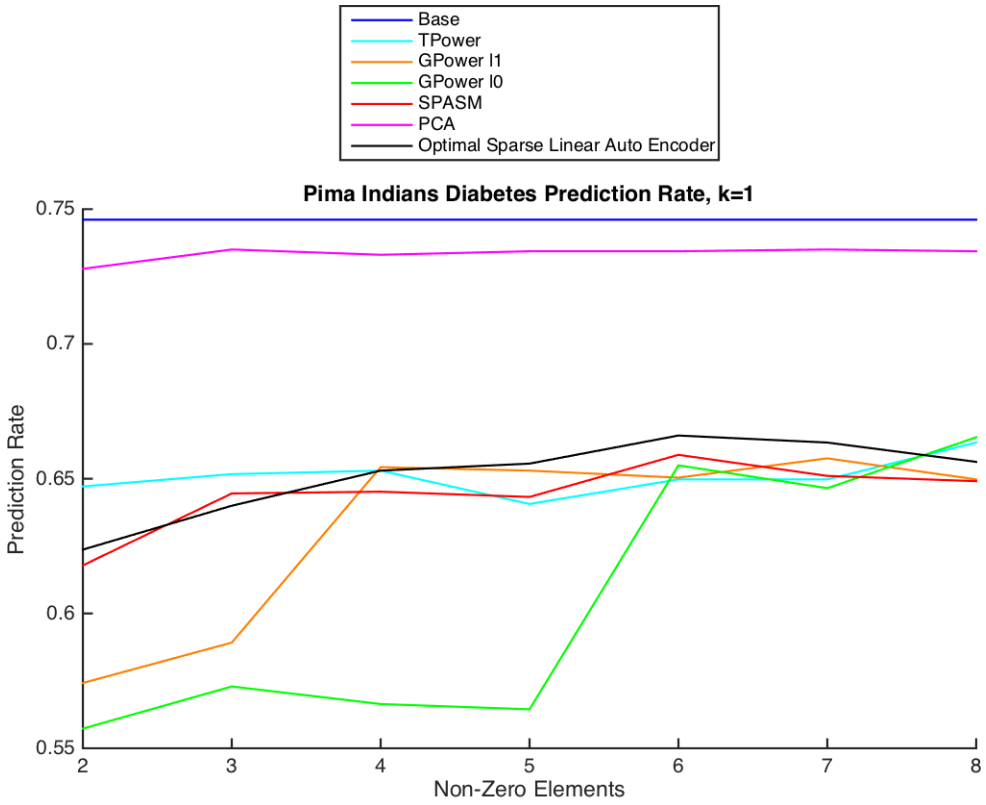

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
OZone One Hour Info Loss, k=6
Info Loss
Non-Zero Elements
4
3.5
3
2.5
2
1.5
1
6
8
10
12
14
16
18
20
22
24
26

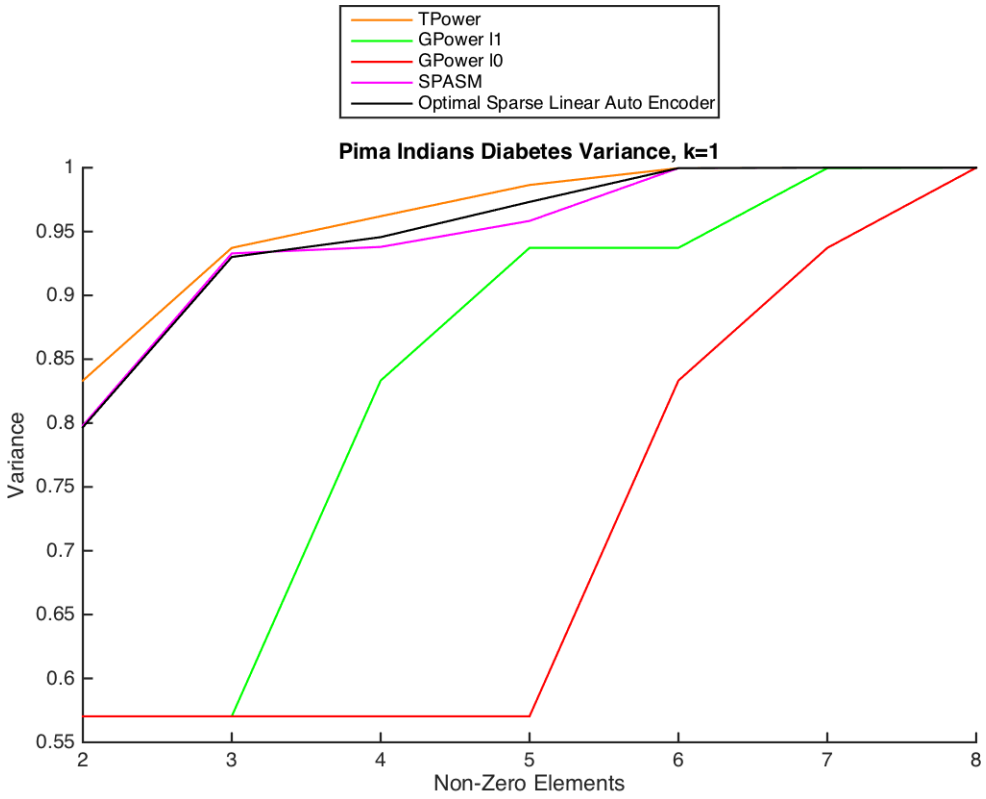

Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
OZone One Hour Prediction Rate, k=6
Prediction Rate
Non-Zero Elements
1
0.9
0.8
0.7
0.6
0.5
0.4
6
8
10
12
14
16
18
20
22
24
26

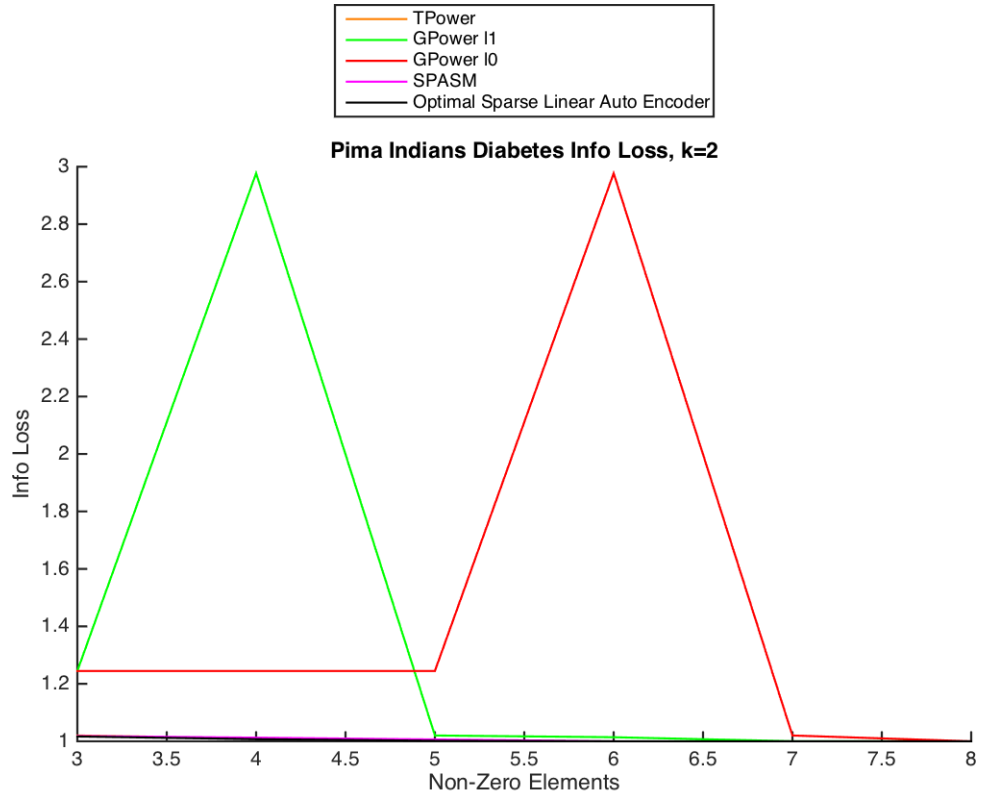

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
OZone One Hour Variance, k=6
Variance
Non-Zero Elements
1
0.9998
0.9996
0.9994
0.9992
0.999
0.9988
0.9986
6
8
10
12
14
16
18
20
22
24
26

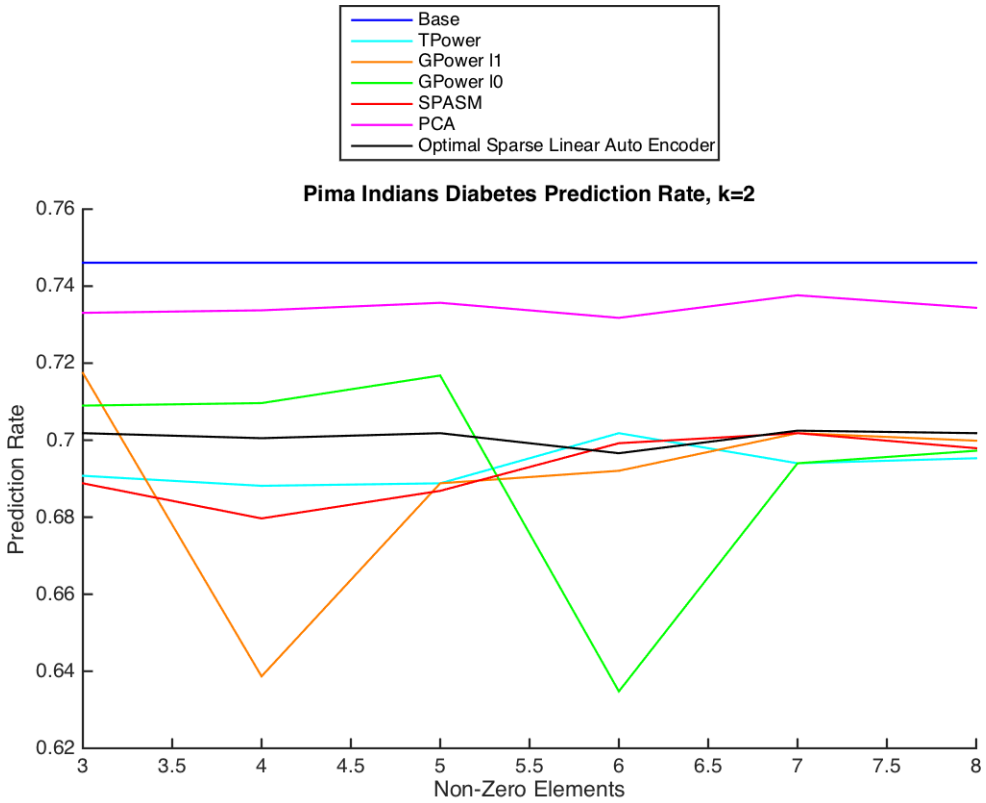
A.8 Pima Indians Diabetes Results

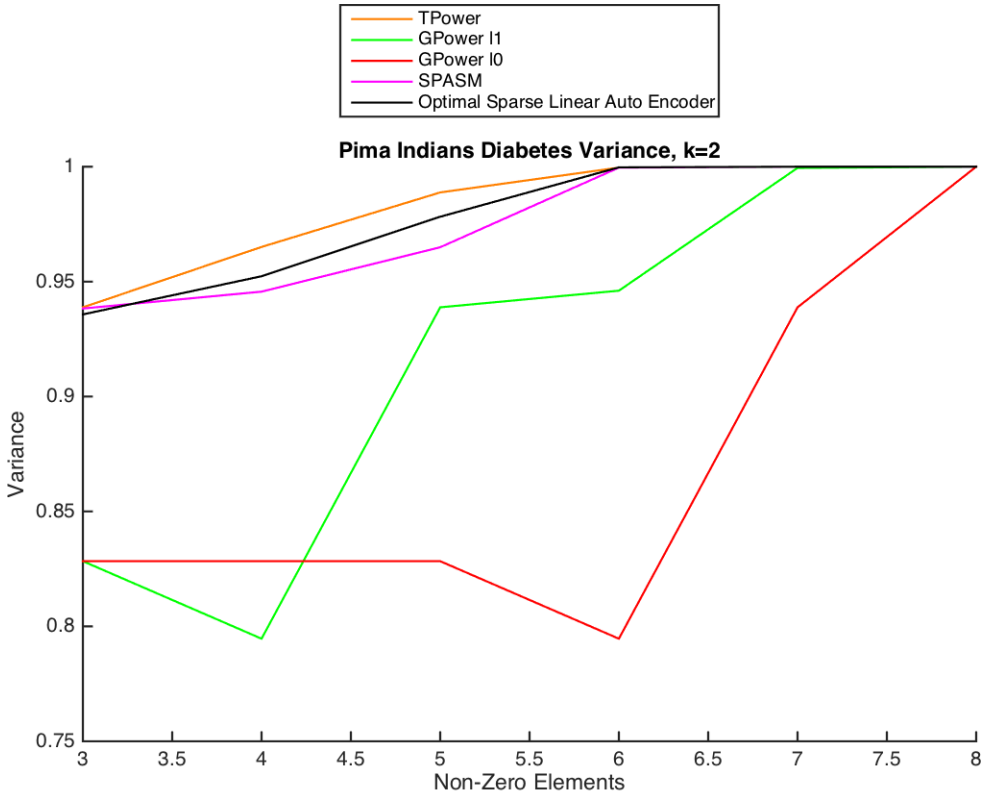


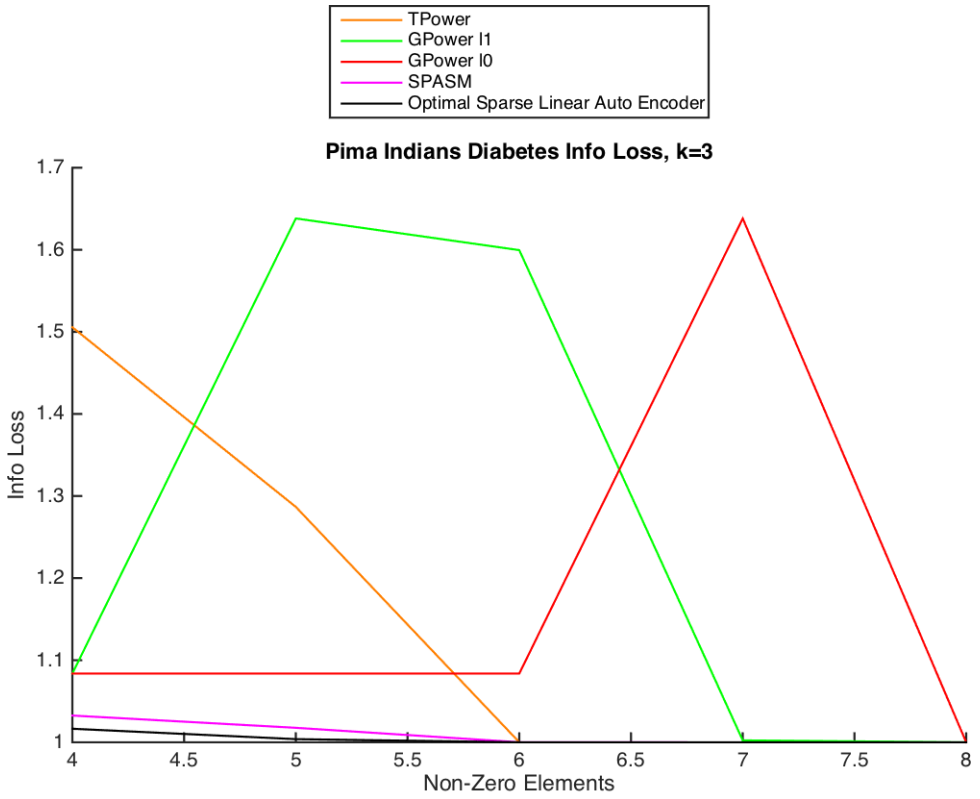


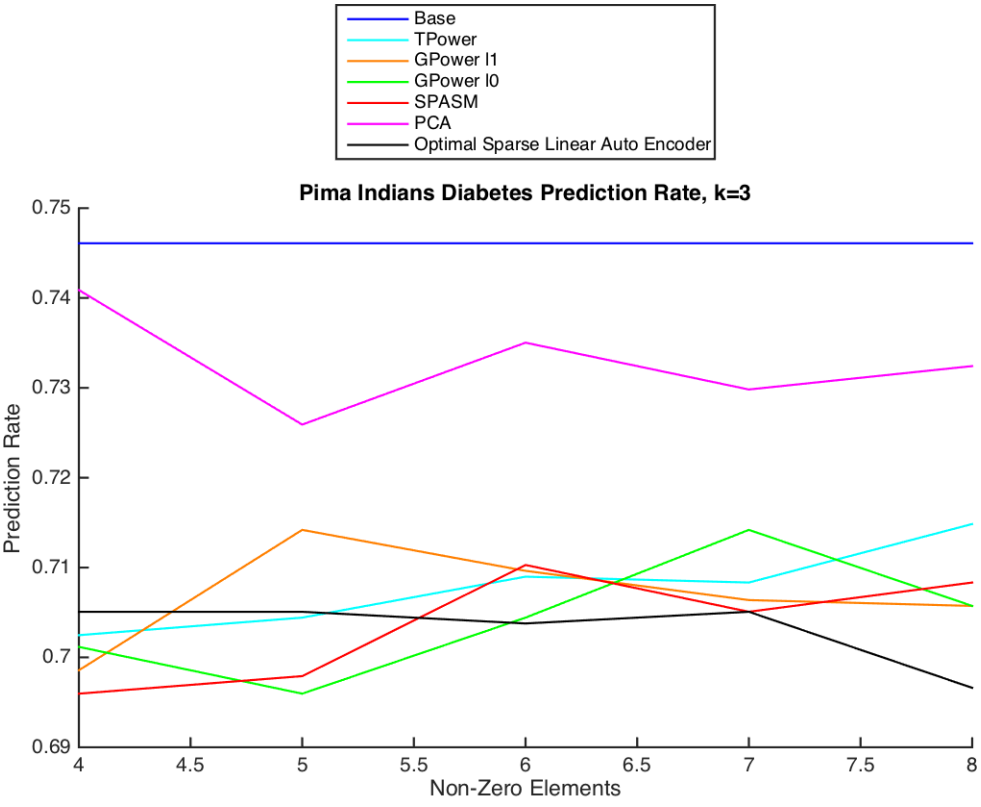

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Pima Indians Diabetes Variance, k=1
Variance
Non-Zero Elements
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
2
3
4
5
6
7
8

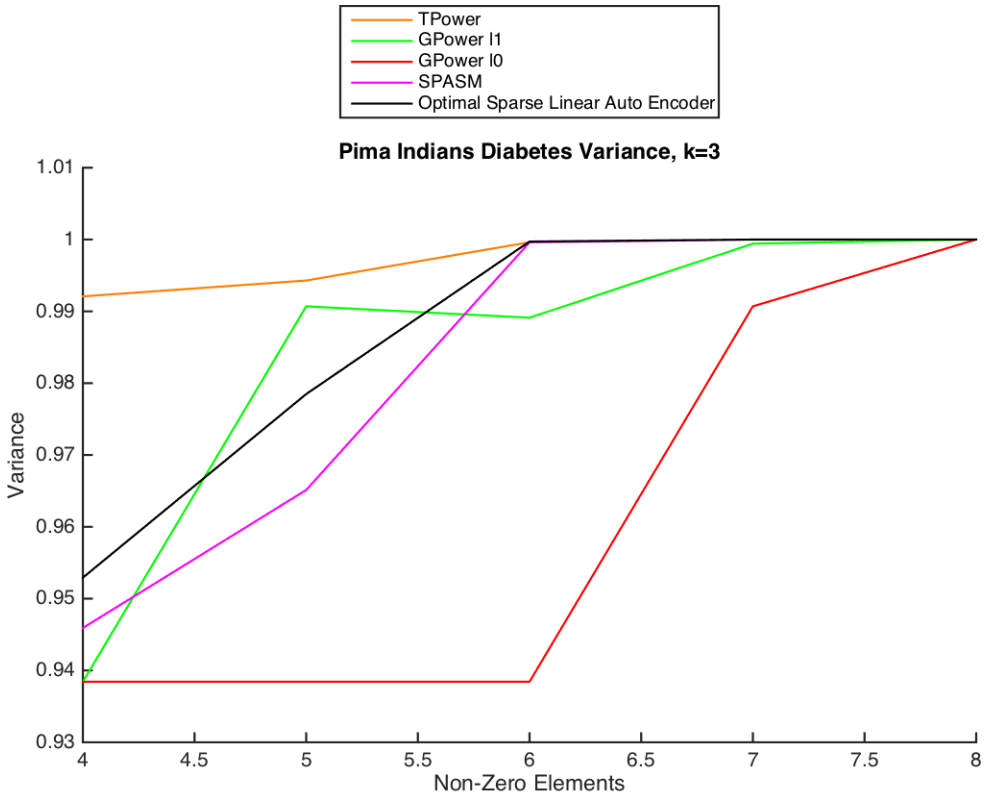

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Pima Indians Diabetes Info Loss, k=2
Info Loss
Non-Zero Elements
3
2.8
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
1
3
3.5
4
4.5
5
5.5
6
6.5
7
7.5
8

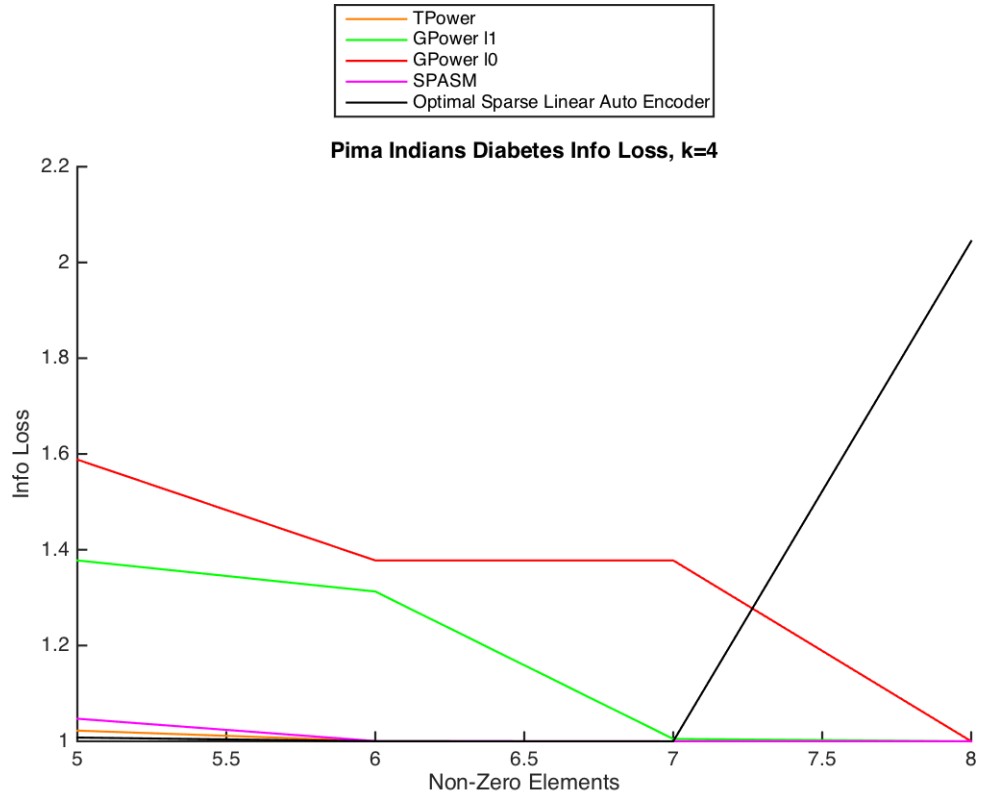

Base
TPower
GPower l1
GPower l0
SPASM
PCA
Optimal Sparse Linear Auto Encoder
Pima Indians Diabetes Prediction Rate, k=2
Prediction Rate
Non-Zero Elements
0.76
0.74
0.72
0.7
0.68
0.66
0.64
0.62
3
3.5
4
4.5
5
5.5
6
6.5
7
7.5
8

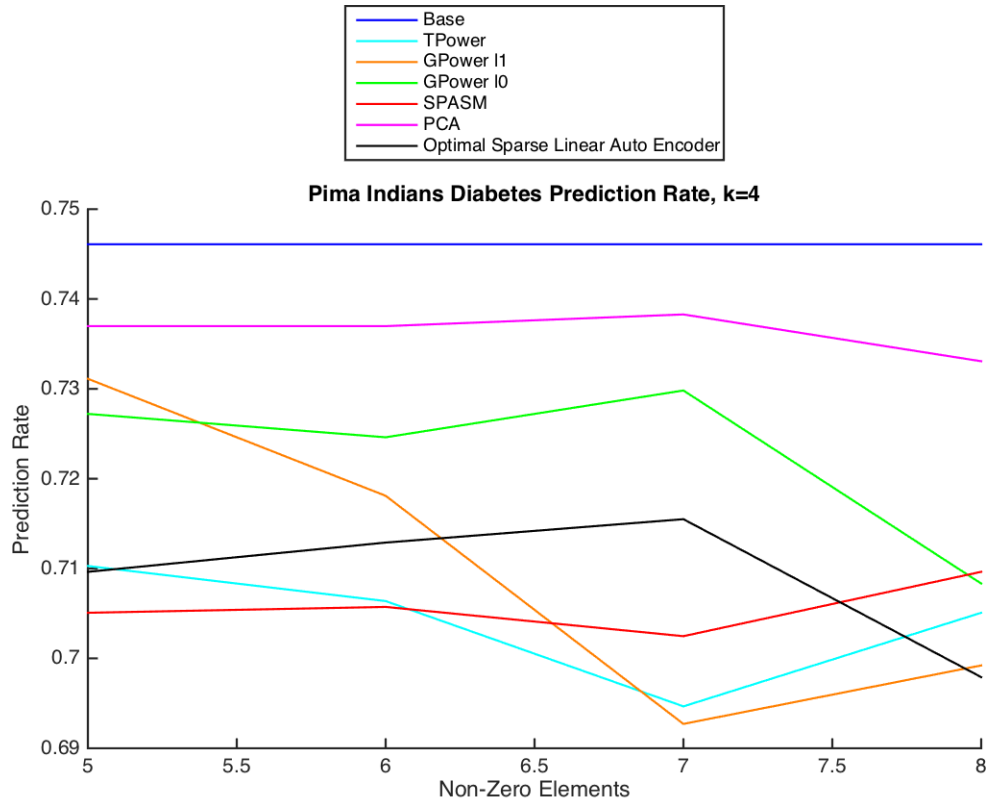

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Pima Indians Diabetes Variance, k=2
Variance
Non-Zero Elements
1
0.95
0.9
0.85
0.8
0.75
3
3.5
4
4.5
5
5.5
6
6.5
7
7.5
8

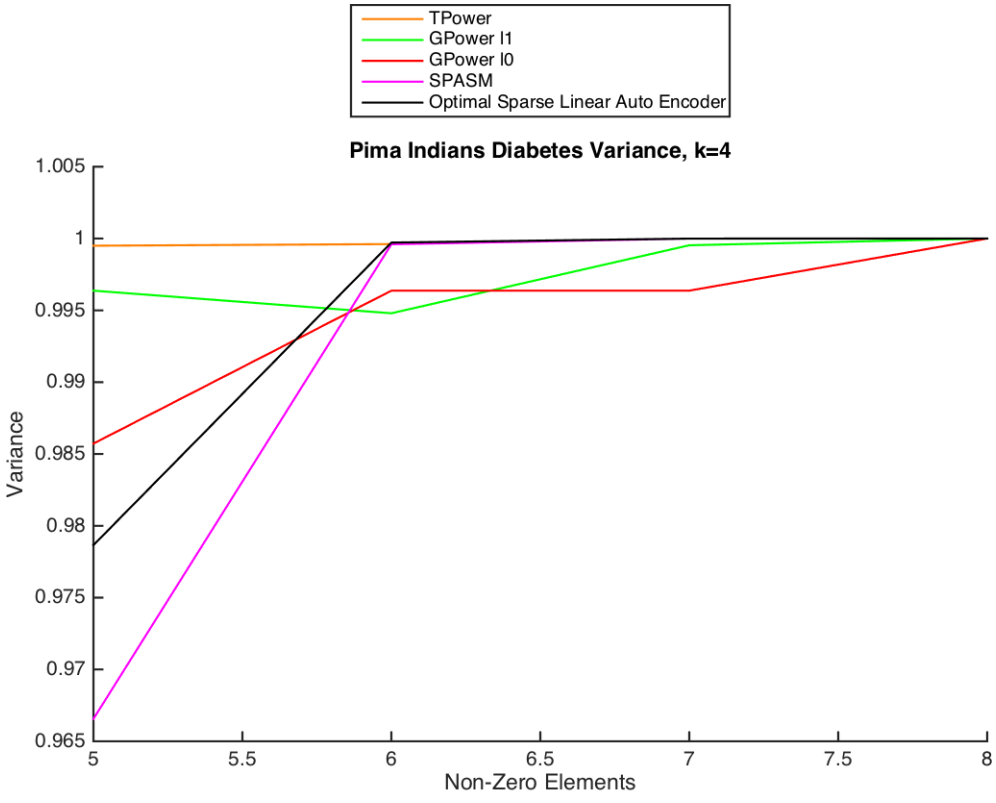

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Pima Indians Diabetes Info Loss, k=3
Info Loss
Non-Zero Elements
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1
4
4.5
5
5.5
6
6.5
7
7.5
8

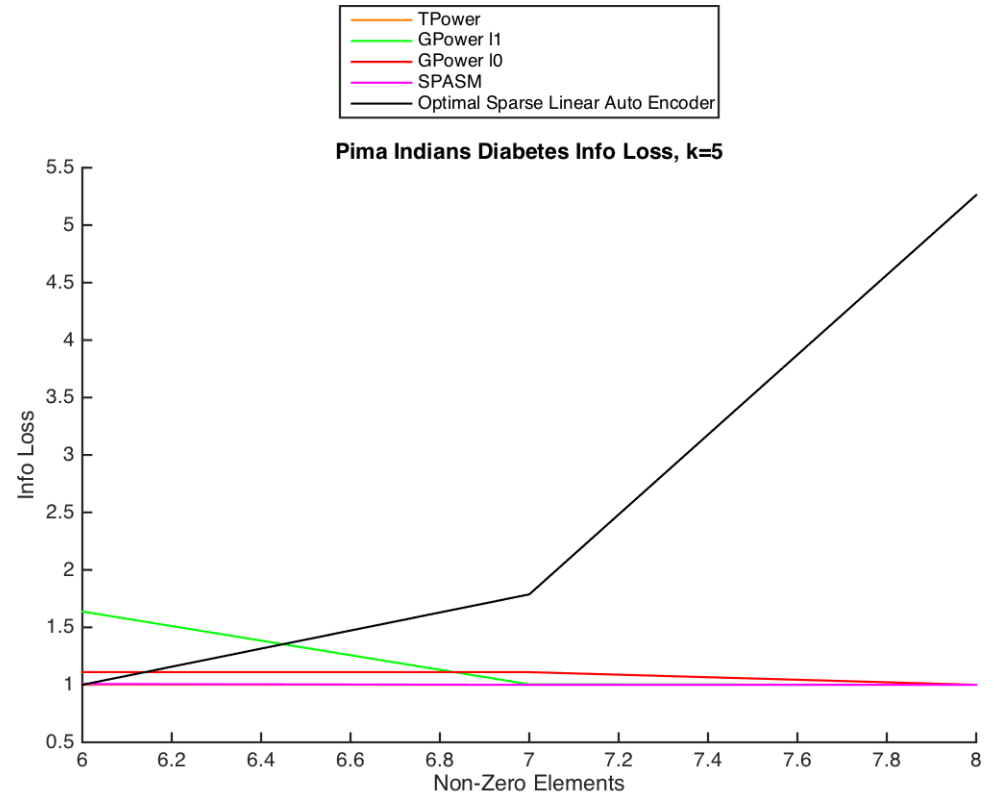

Base
TPower
GPower l1
GPower l0
SPASM
PCA
Optimal Sparse Linear Auto Encoder
Pima Indians Diabetes Prediction Rate, k=3
Prediction Rate
Non-Zero Elements
0.75
0.74
0.73
0.72
0.71
0.7
0.69
4
4.5
5
5.5
6
6.5
7
7.5
8

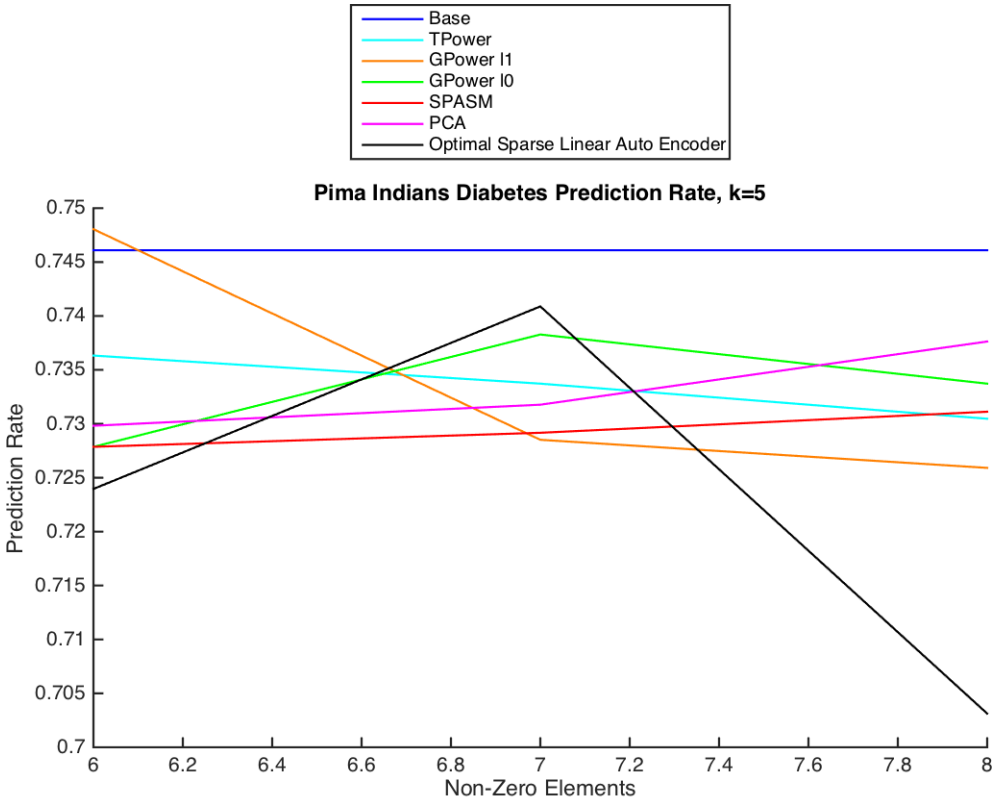

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Pima Indians Diabetes Variance, k=3
Variance
Non-Zero Elements
1.01
1
0.99
0.98
0.97
0.96
0.95
0.94
0.93
4
4.5
5
5.5
6
6.5
7
7.5
8

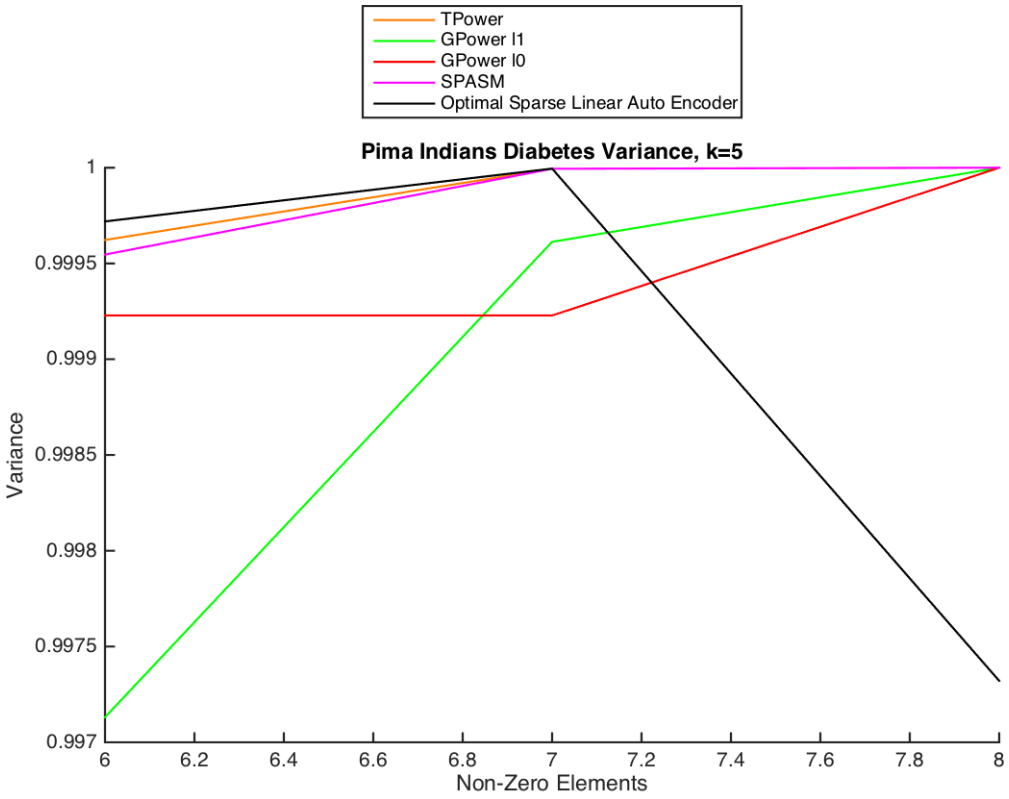

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Pima Indians Diabetes Info Loss, k=4
Info Loss
Non-Zero Elements
2.2
2
1.8
1.6
1.4
1.2
1
5
5.5
6
6.5
7
7.5
8

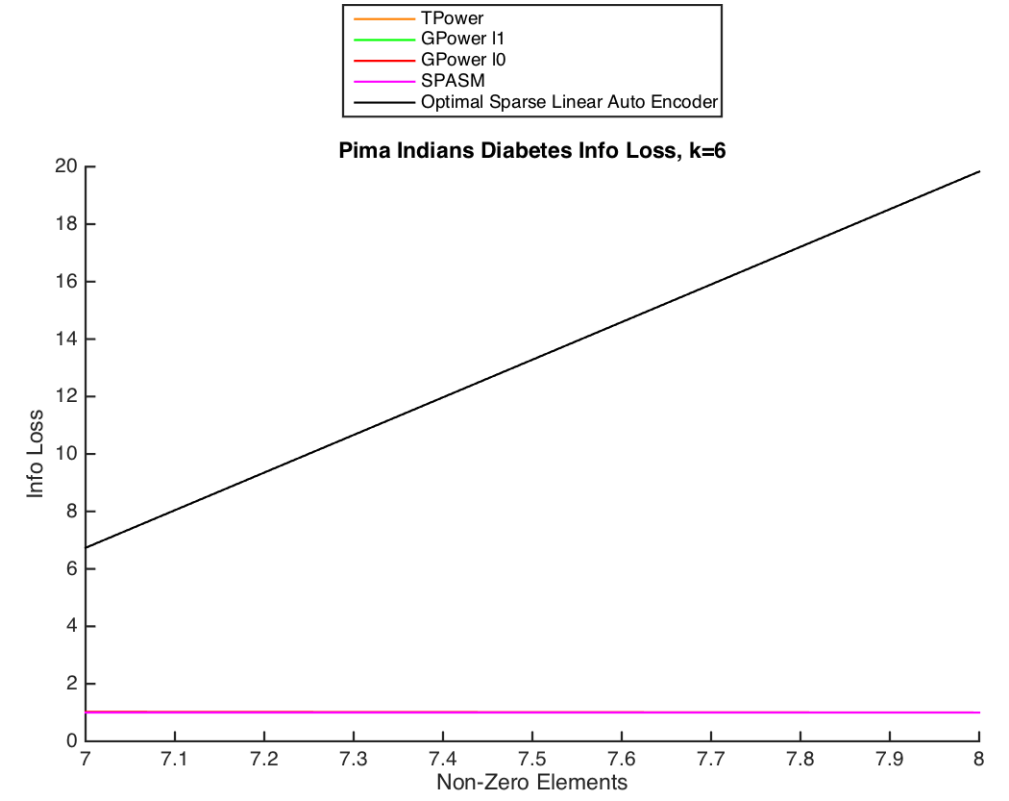

Base
TPower
GPower l1
GPower l0
SPASM
PCA
Optimal Sparse Linear Auto Encoder
Pima Indians Diabetes Prediction Rate, k=4
Prediction Rate
Non-Zero Elements
0.75
0.74
0.73
0.72
0.71
0.7
0.69
5
5.5
6
6.5
7
7.5
8

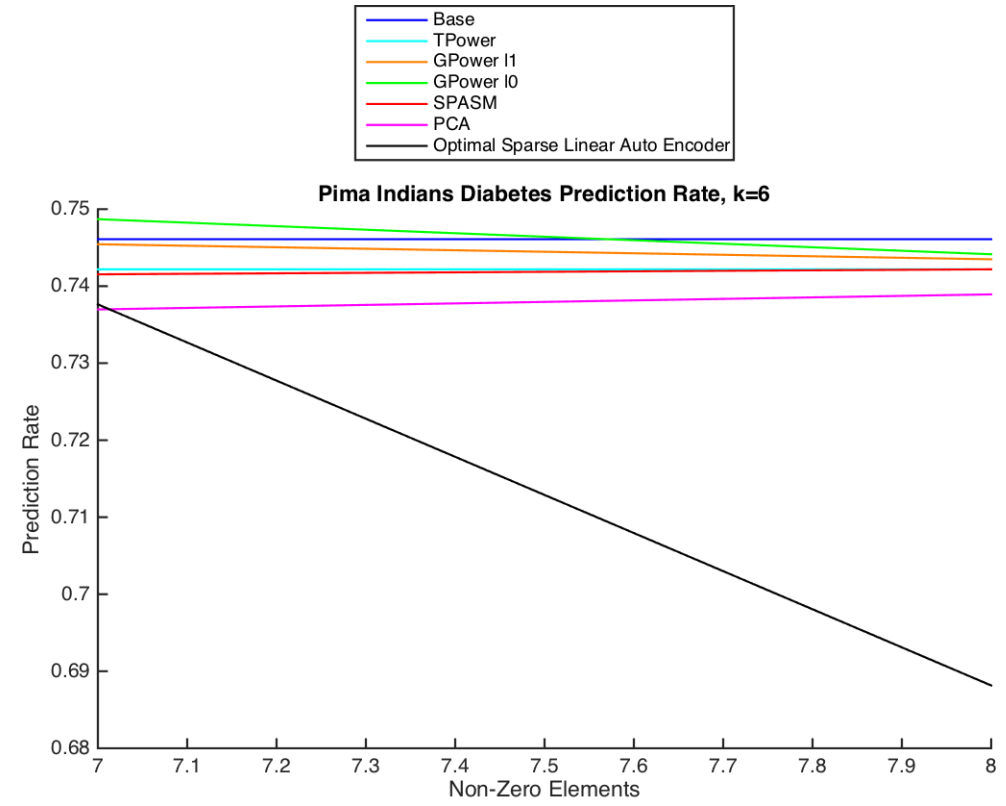

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Pima Indians Diabetes Variance, k=4
Variance
Non-Zero Elements
1.005
1
0.995
0.99
0.985
0.98
0.975
0.97
0.965
5
5.5
6
6.5
7
7.5
8

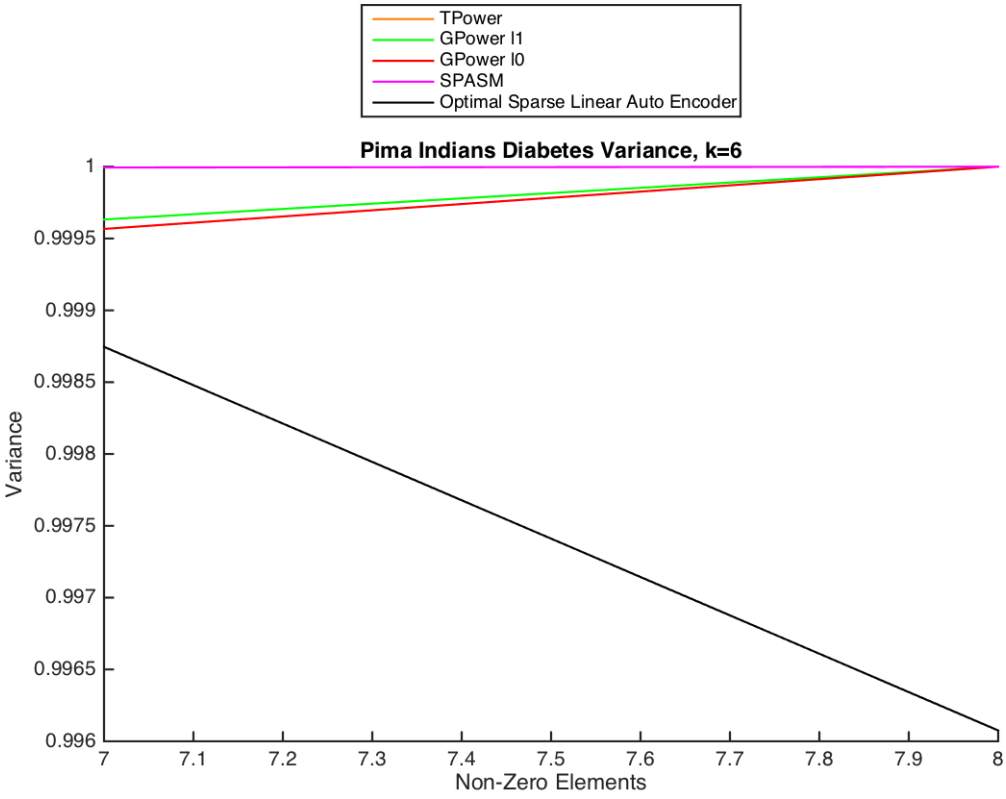

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Pima Indians Diabetes Info Loss, k=5
Info Loss
Non-Zero Elements
5.5
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
6
6.2
6.4
6.6
6.8
7
7.2
7.4
7.6
7.8
8


Base
TPower
GPower l1
GPower l0
SPASM
PCA
Optimal Sparse Linear Auto Encoder
Pima Indians Diabetes Prediction Rate, k=5
Prediction Rate
Non-Zero Elements
0.75
0.745
0.74
0.735
0.73
0.725
0.72
0.715
0.71
0.705
0.7
6
6.2
6.4
6.6
6.8
7
7.2
7.4
7.6
7.8
8

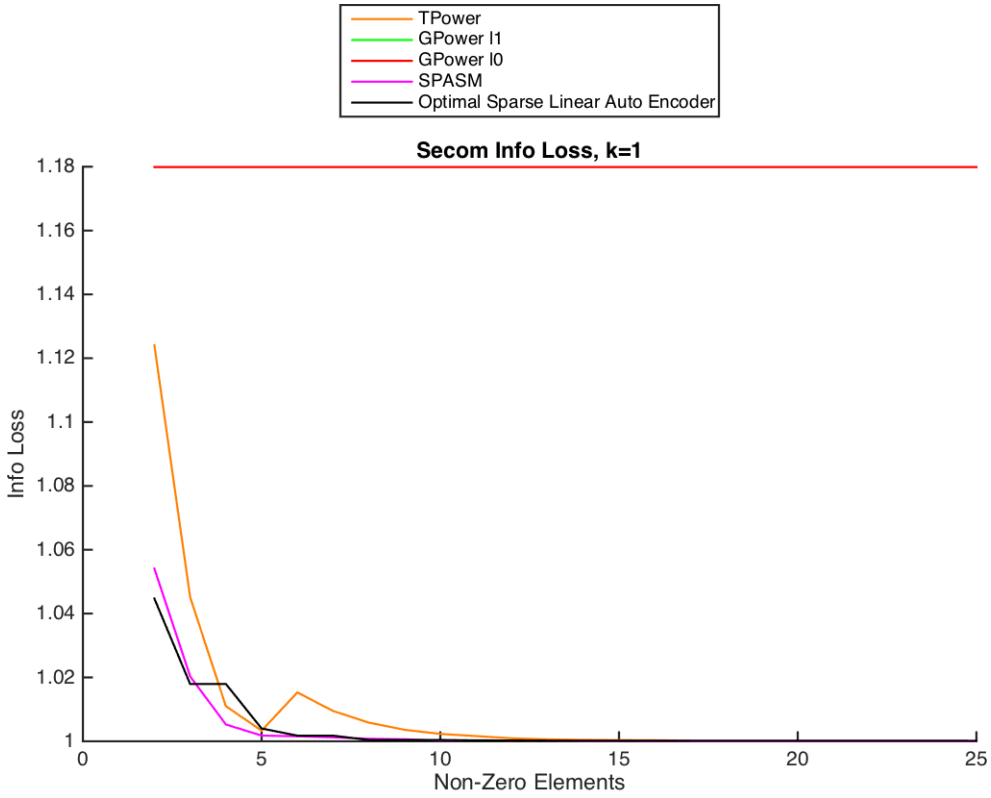

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Pima Indians Diabetes Variance, k=5
Variance
Non-Zero Elements
1
0.9995
0.999
0.9985
0.998
0.9975
0.997
6
6.2
6.4
6.6
6.8
7
7.2
7.4
7.6
7.8
8

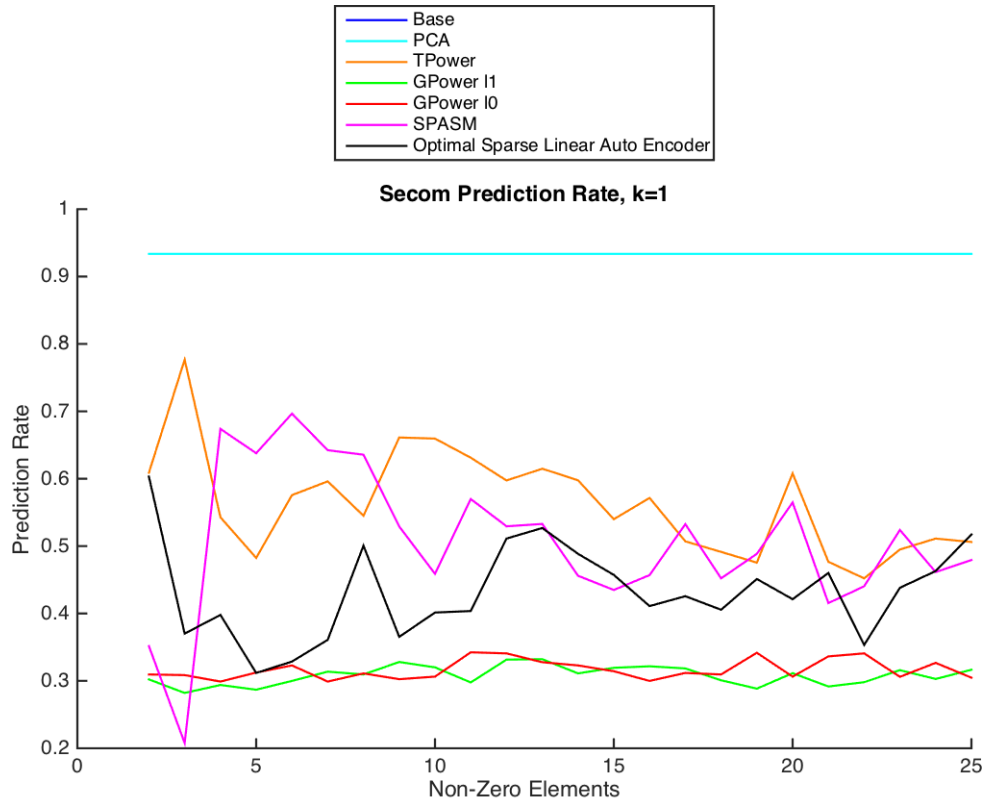

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Pima Indians Diabetes Info Loss, k=6
Info Loss
Non-Zero Elements
20
18
16
14
12
10
8
6
4
2
0
7
7.1
7.2
7.3
7.4
7.5
7.6
7.7
7.8
7.9
8

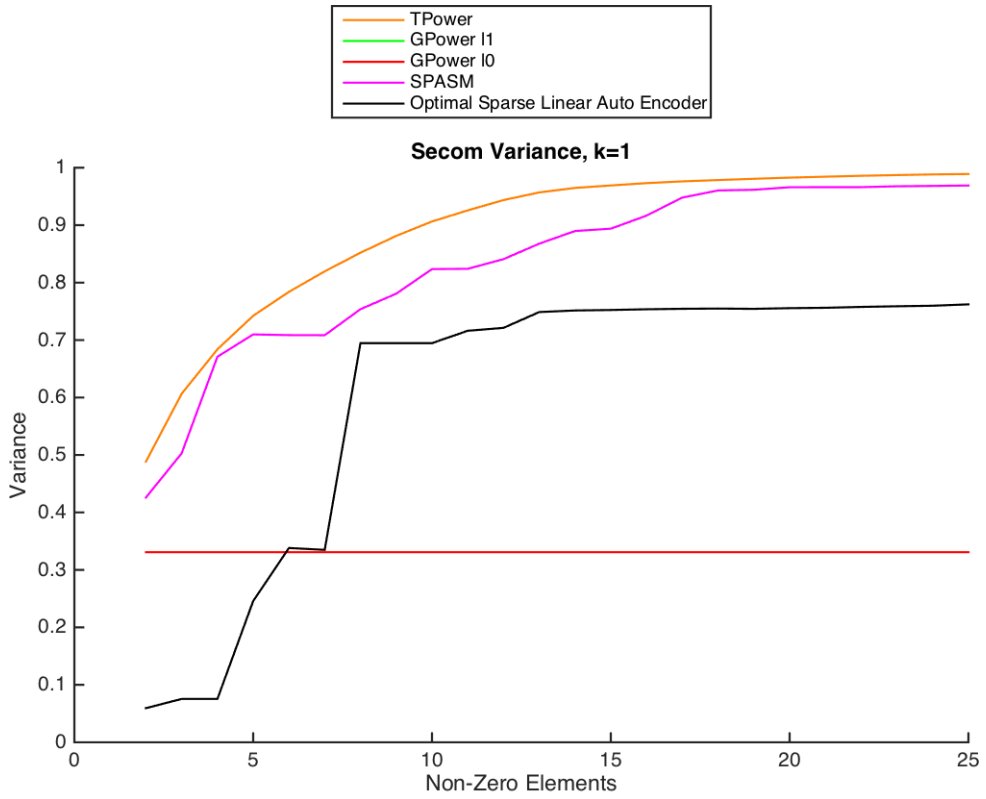

Base
TPower
GPower l1
GPower l0
SPASM
PCA
Optimal Sparse Linear Auto Encoder
Pima Indians Diabetes Prediction Rate, k=6
Prediction Rate
Non-Zero Elements
0.75
0.74
0.73
0.72
0.71
0.7
0.69
0.68
7
7.1
7.2
7.3
7.4
7.5
7.6
7.7
7.8
7.9
8

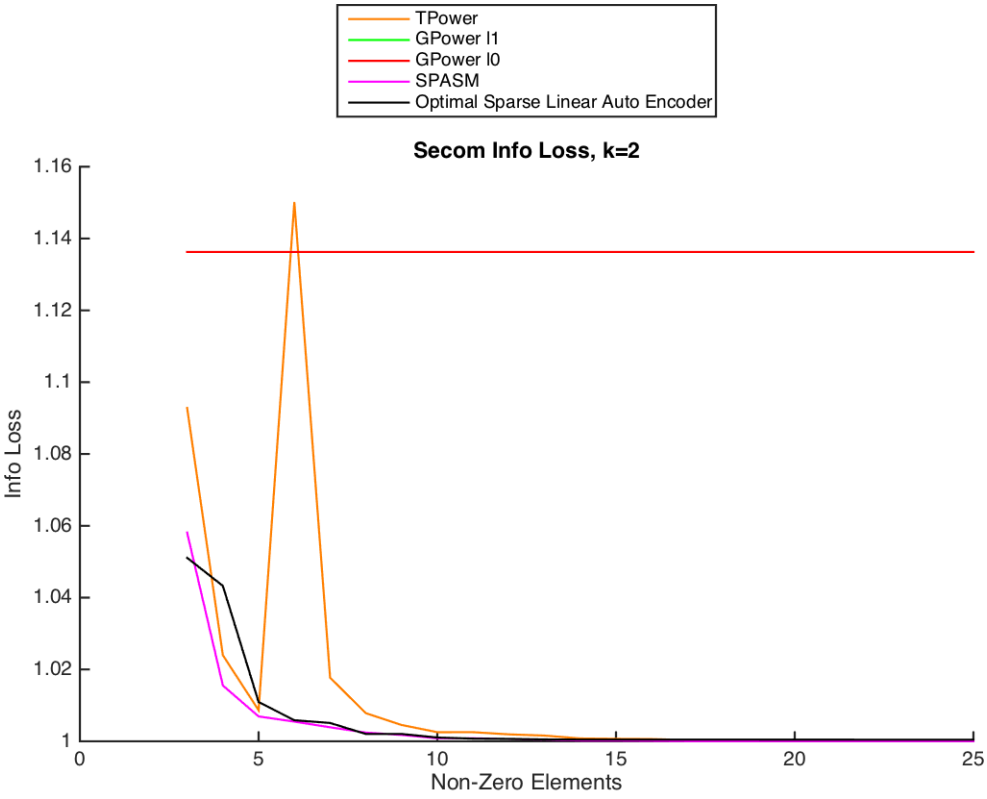

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Pima Indians Diabetes Variance, k=6
Variance
Non-Zero Elements
1
0.9995
0.999
0.9985
0.998
0.9975
0.997
0.9965
0.996
7
7.1
7.2
7.3
7.4
7.5
7.6
7.7
7.8
7.9
8

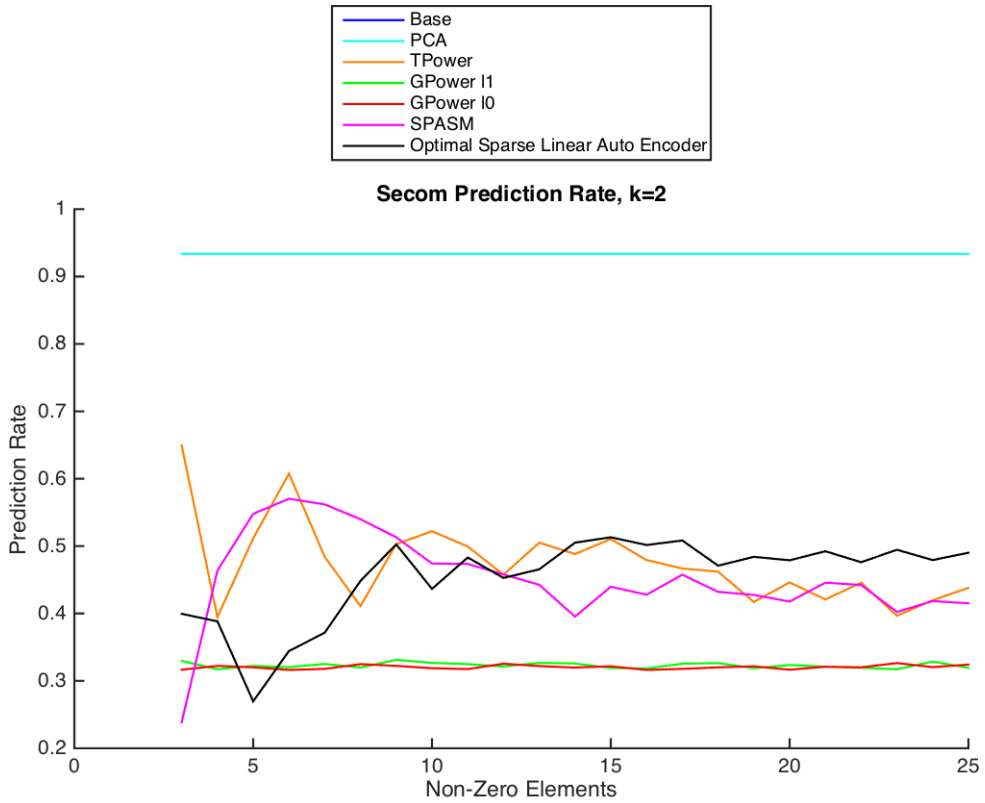
A.9 SECOM Results

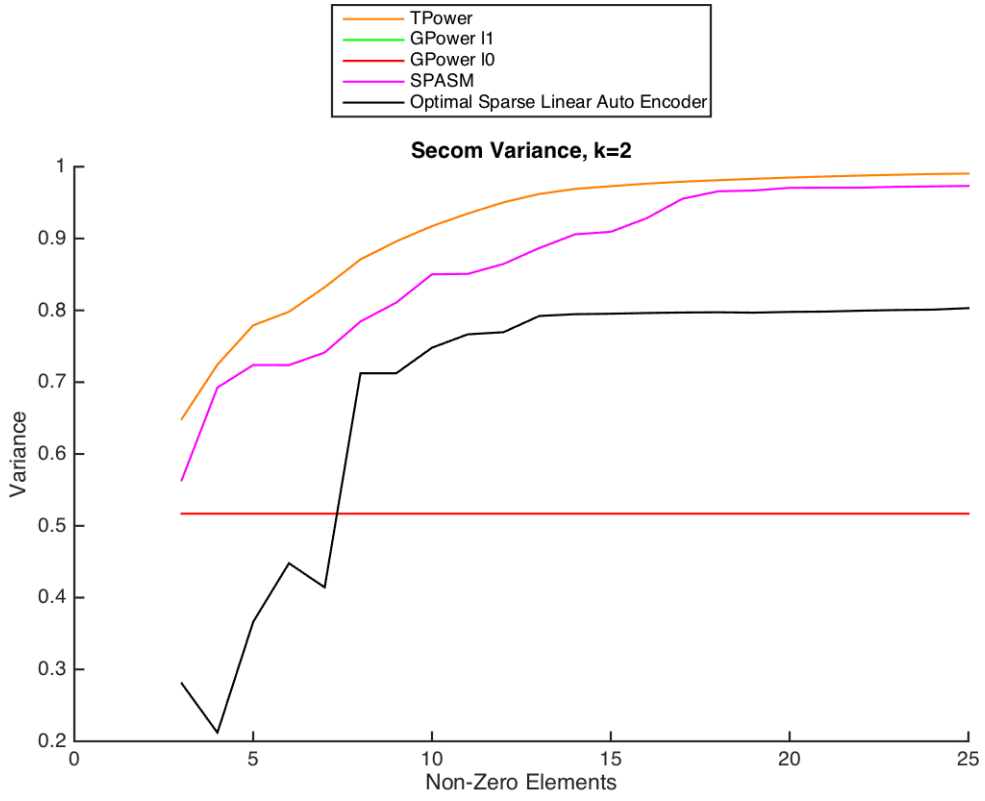


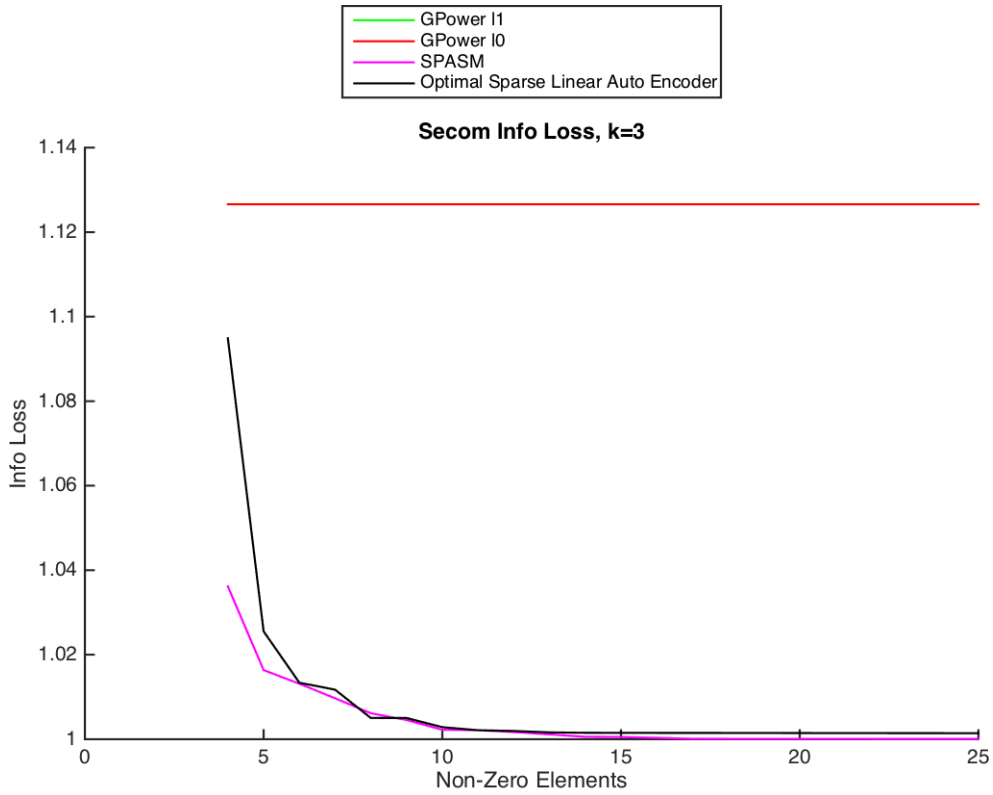


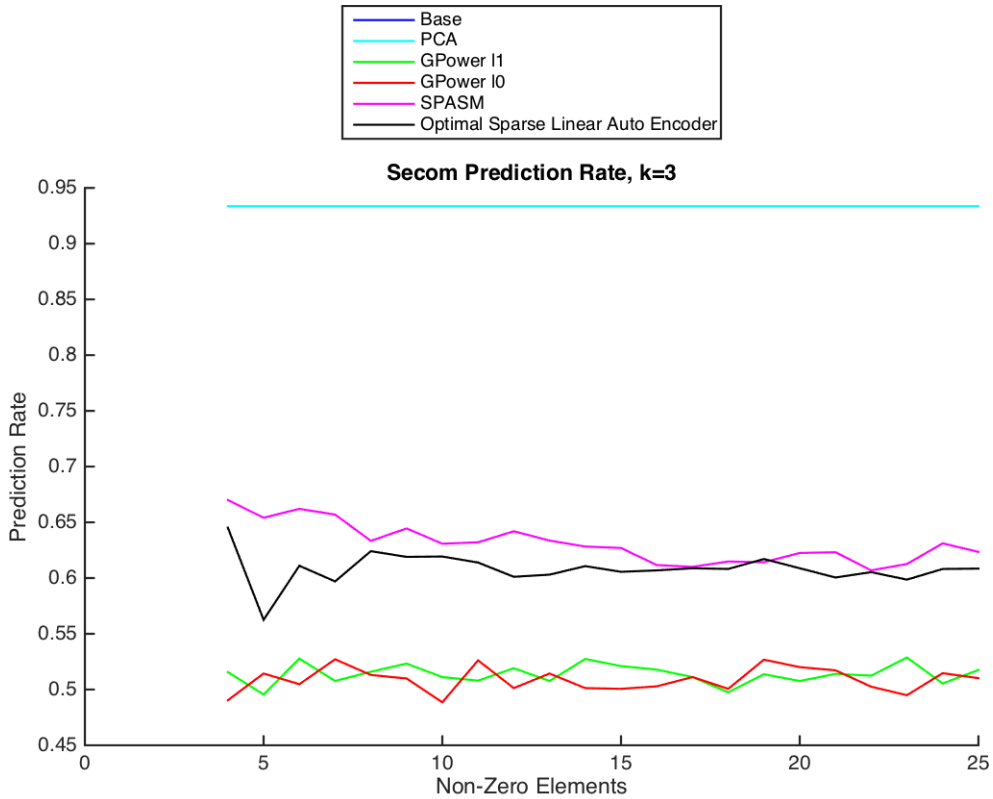

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Secom Variance, k=1
Variance
Non-Zero Elements
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
5
10
15
20
25

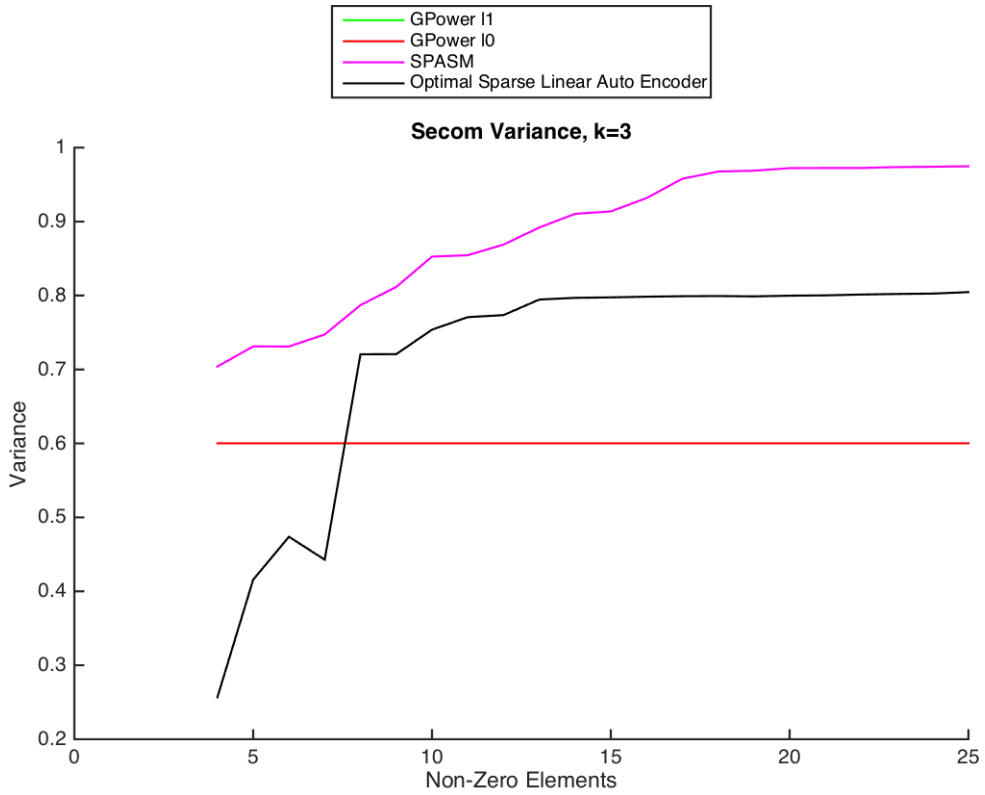

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Secom Info Loss, k=2
Info Loss
Non-Zero Elements
1
1.02
1.04
1.06
1.08
1.1
1.12
1.14
1.16
0
5
10
15
20
25

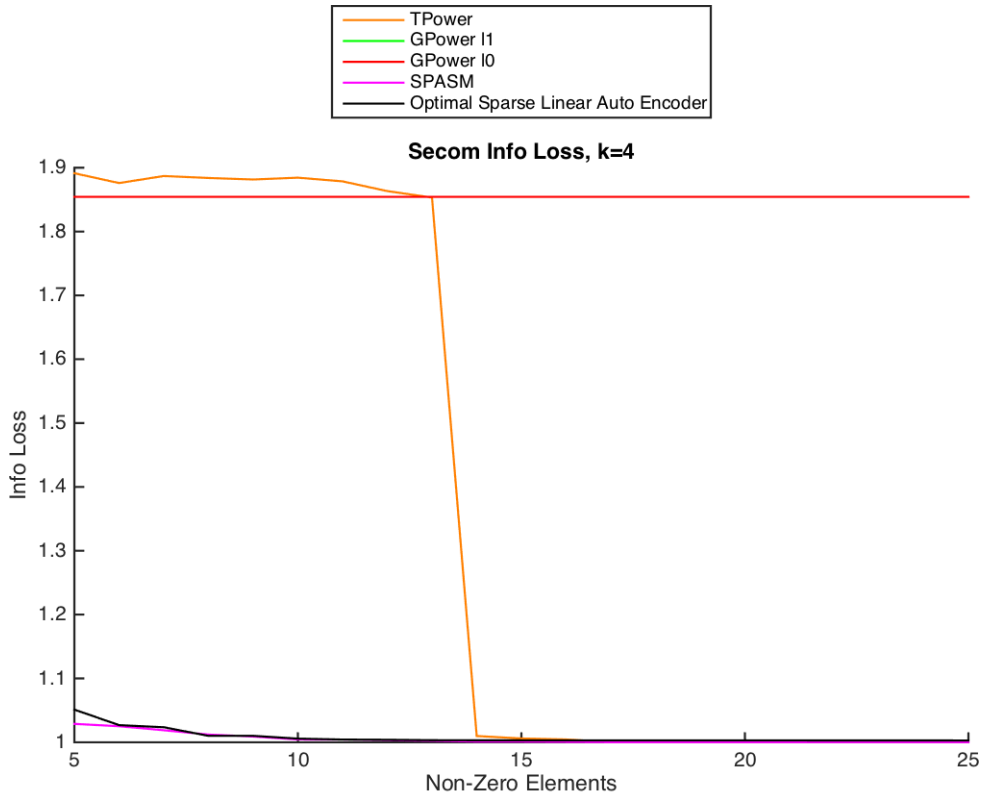

Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Secom Prediction Rate, k=2
Prediction Rate
Non-Zero Elements
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
5
10
15
20
25

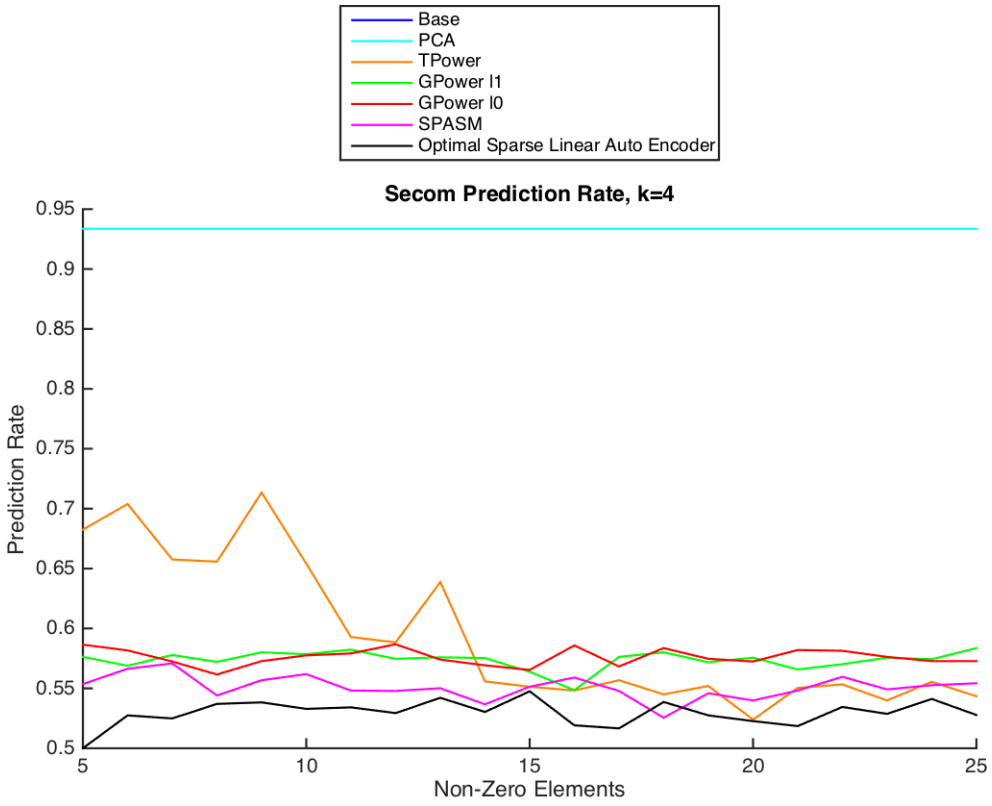

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Secom Variance, k=2
Variance
Non-Zero Elements
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
5
10
15
20
25

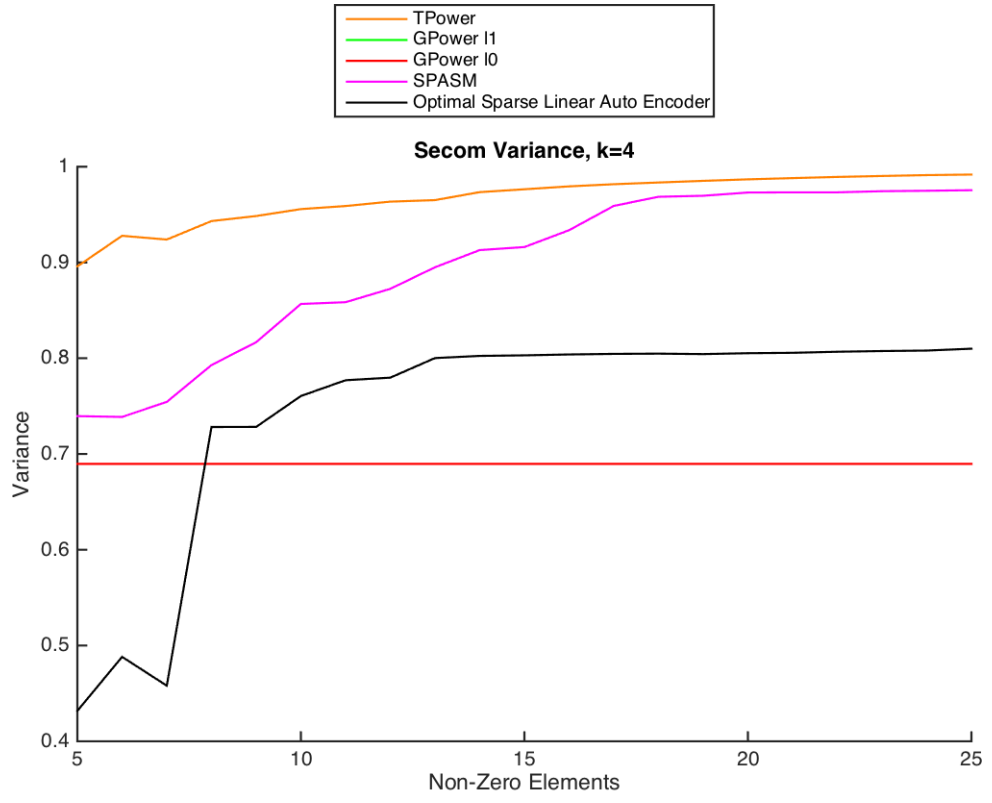

GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Secom Info Loss, k=3
Info Loss
Non-Zero Elements
1.14
1.12
1.1
1.08
1.06
1.04
1.02
1
0
5
10
15
20
25

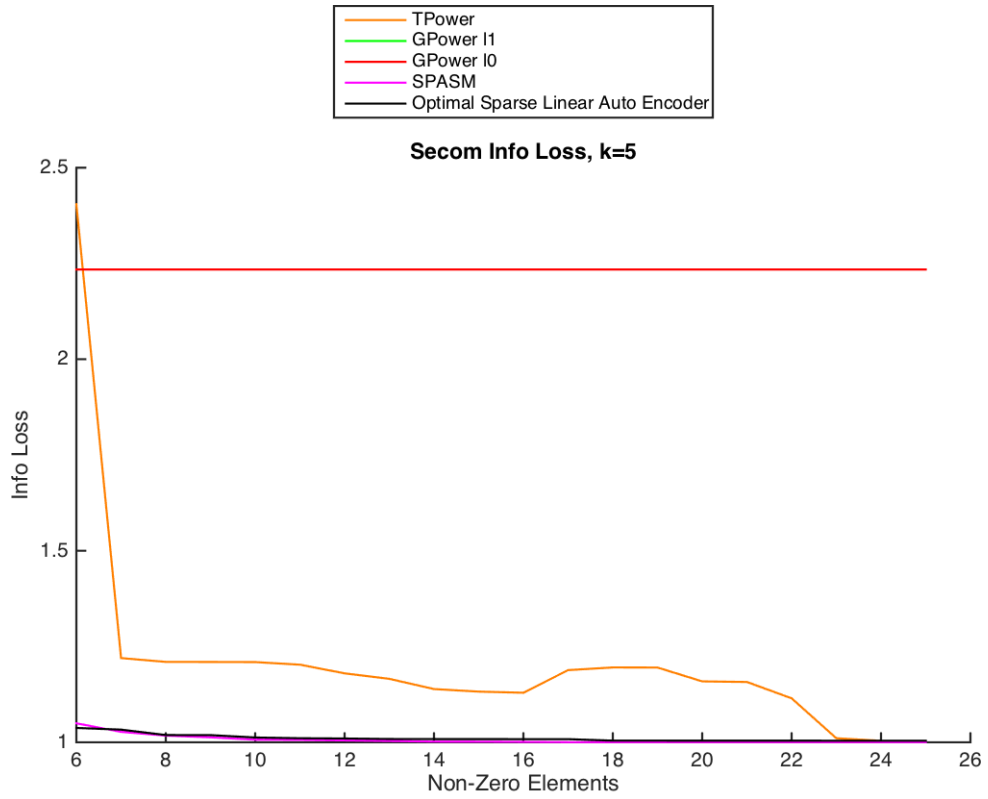

Base
PCA
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Secom Prediction Rate, k=3
Prediction Rate
Non-Zero Elements
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0
5
10
15
20
25

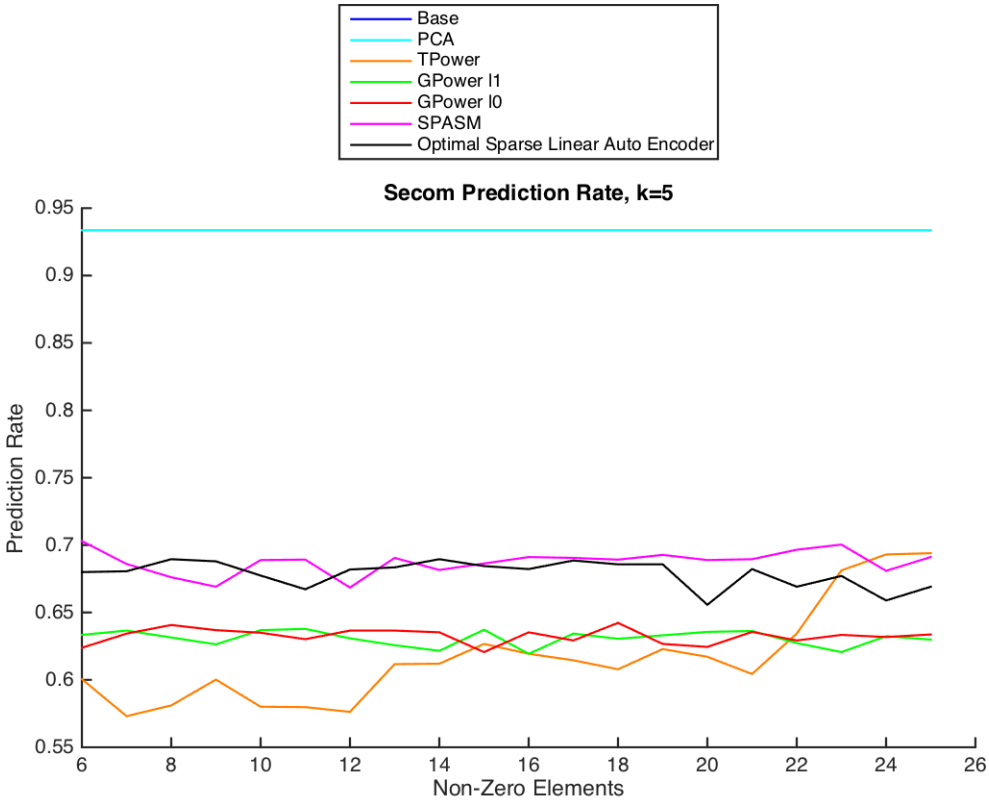

GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Secom Variance, k=3
Variance
Non-Zero Elements
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0
5
10
15
20
25

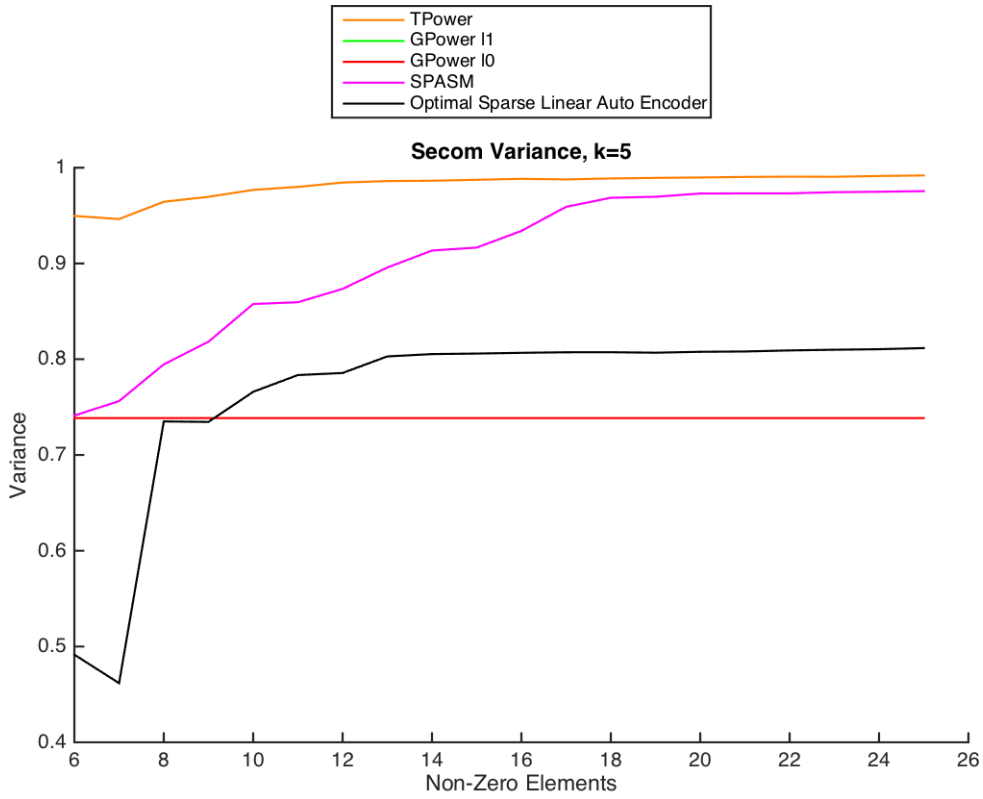

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Secom Info Loss, k=4
Info Loss
Non-Zero Elements
1.9
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1
5
10
15
20
25

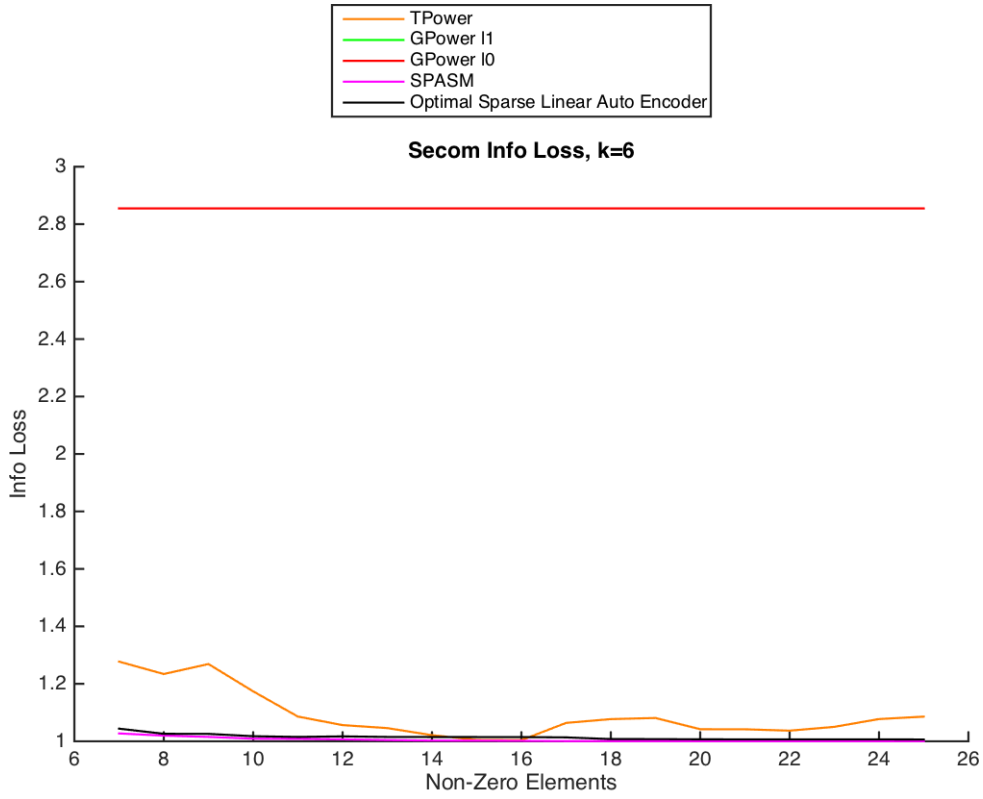

Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Secom Prediction Rate, k=4
Prediction Rate
Non-Zero Elements
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
5
10
15
20
25

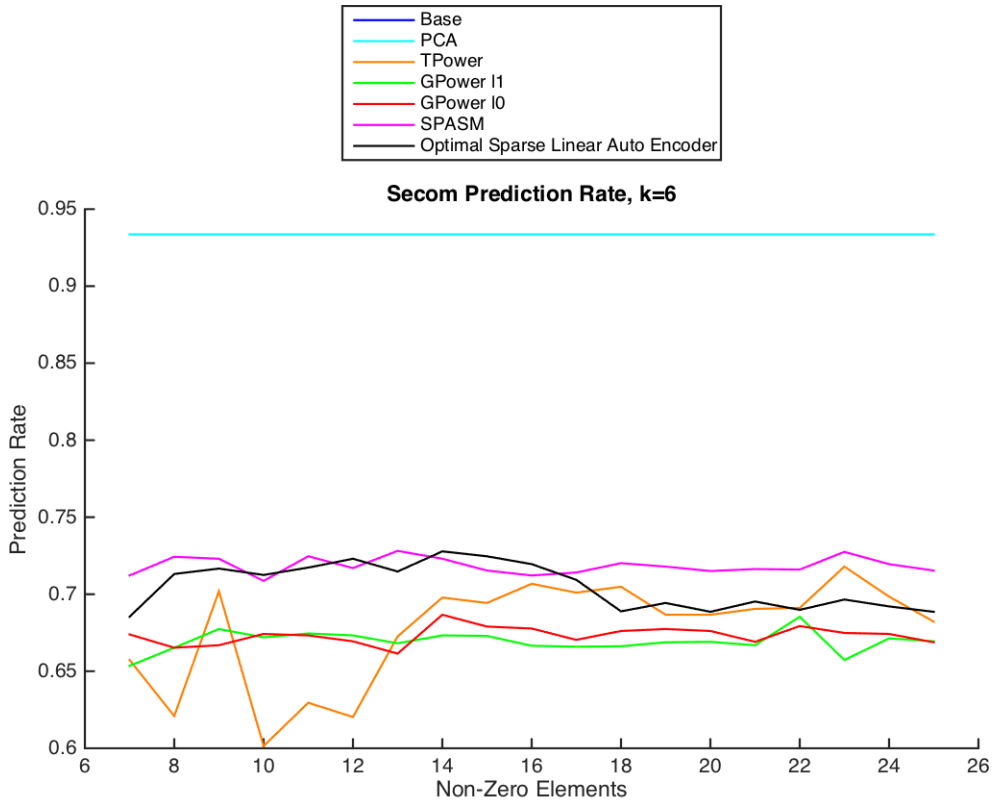

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Secom Variance, k=4
Variance
Non-Zero Elements
1
0.9
0.8
0.7
0.6
0.5
0.4
5
10
15
20
25

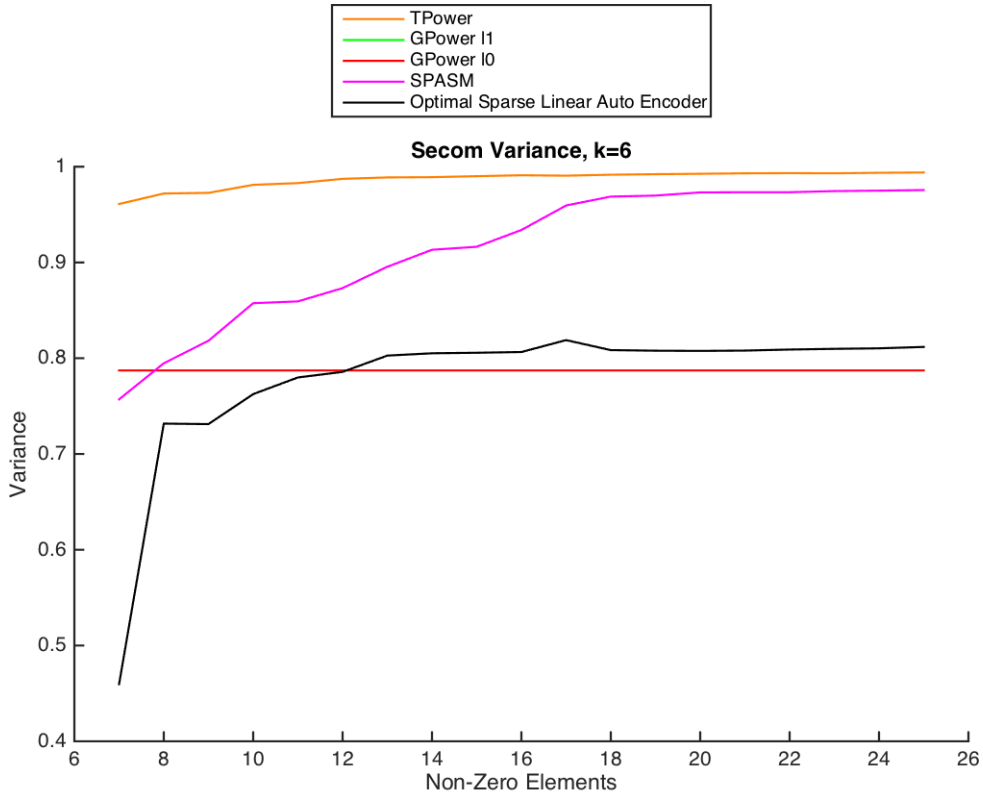

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Secom Info Loss, k=5
Info Loss
Non-Zero Elements
2.5
2
1.5
1
6
8
10
12
14
16
18
20
22
24
26


Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Secom Prediction Rate, k=5
Prediction Rate
Non-Zero Elements
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
6
8
10
12
14
16
18
20
22
24
26

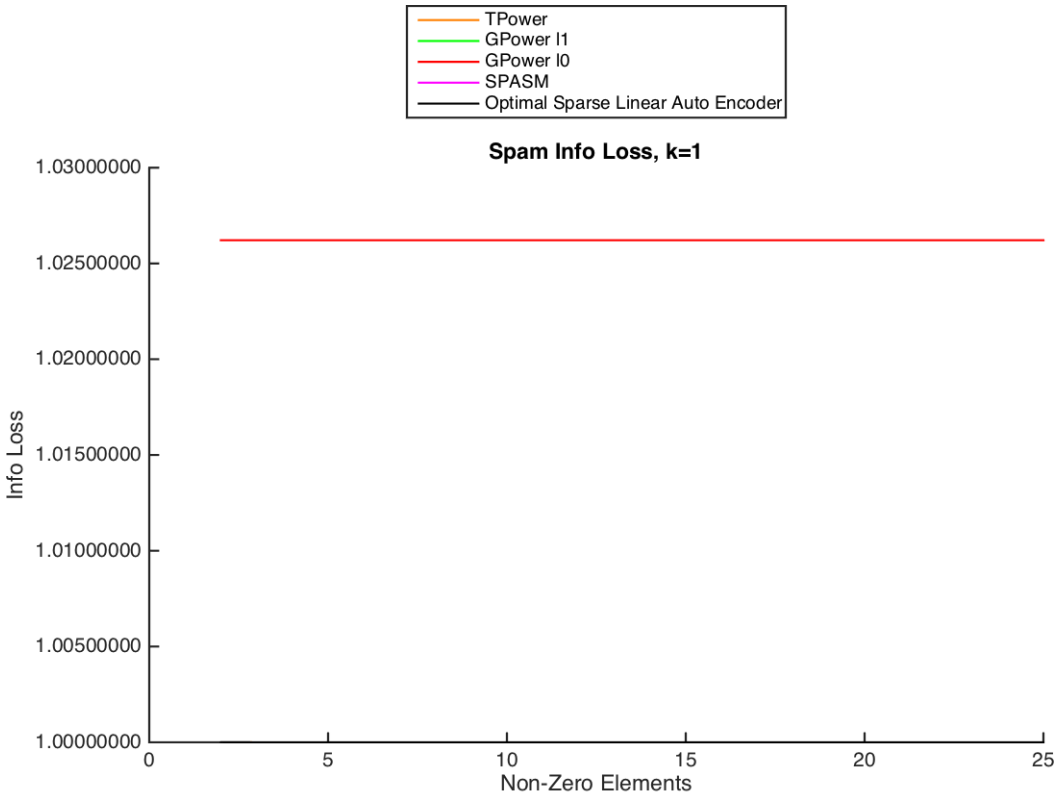

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Secom Variance, k=5
Variance
Non-Zero Elements
1
0.9
0.8
0.7
0.6
0.5
0.4
6
8
10
12
14
16
18
20
22
24
26

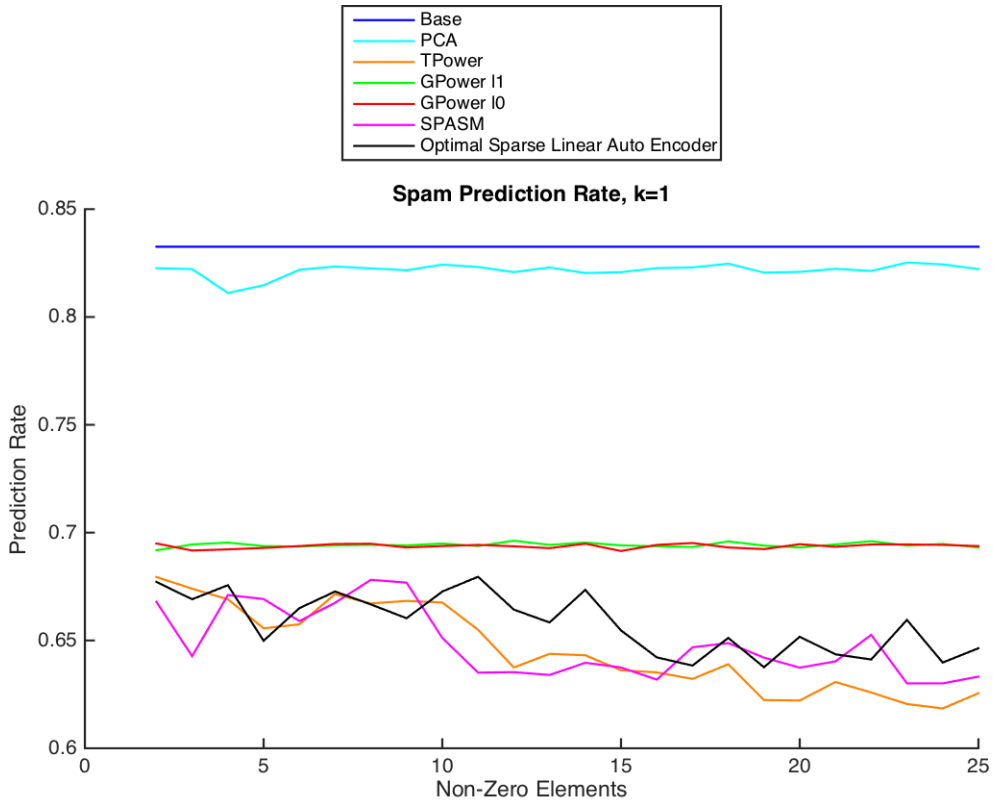

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Secom Info Loss, k=6
Info Loss
Non-Zero Elements
3
2.8
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
1
6
8
10
12
14
16
18
20
22
24
26

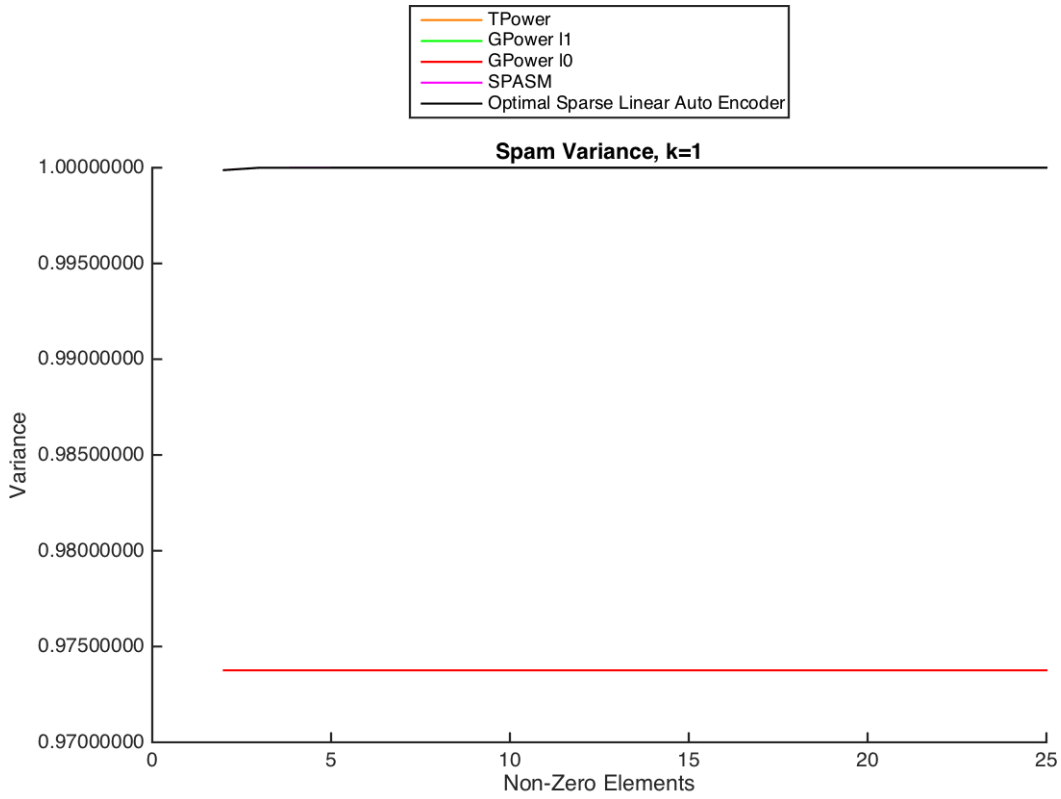

Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Secom Prediction Rate, k=6
Prediction Rate
Non-Zero Elements
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
6
8
10
12
14
16
18
20
22
24
26

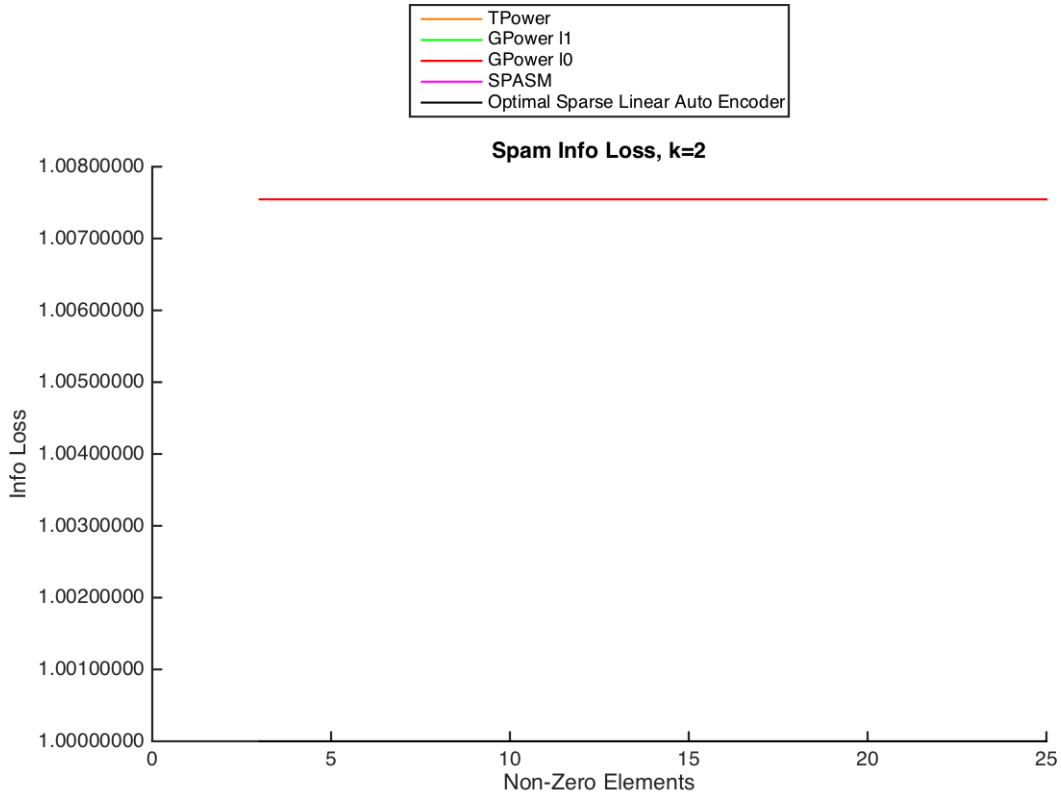

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Secom Variance, k=6
Variance
Non-Zero Elements
1
0.9
0.8
0.7
0.6
0.5
0.4
6
8
10
12
14
16
18
20
22
24
26

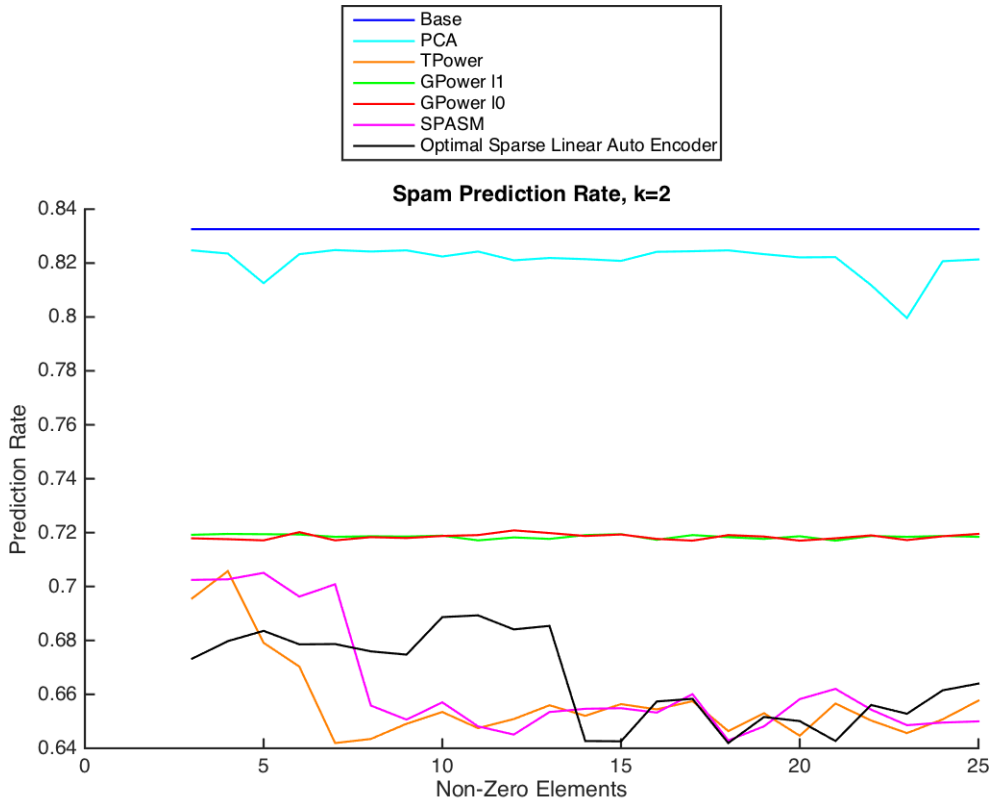
A.10 Spam Results

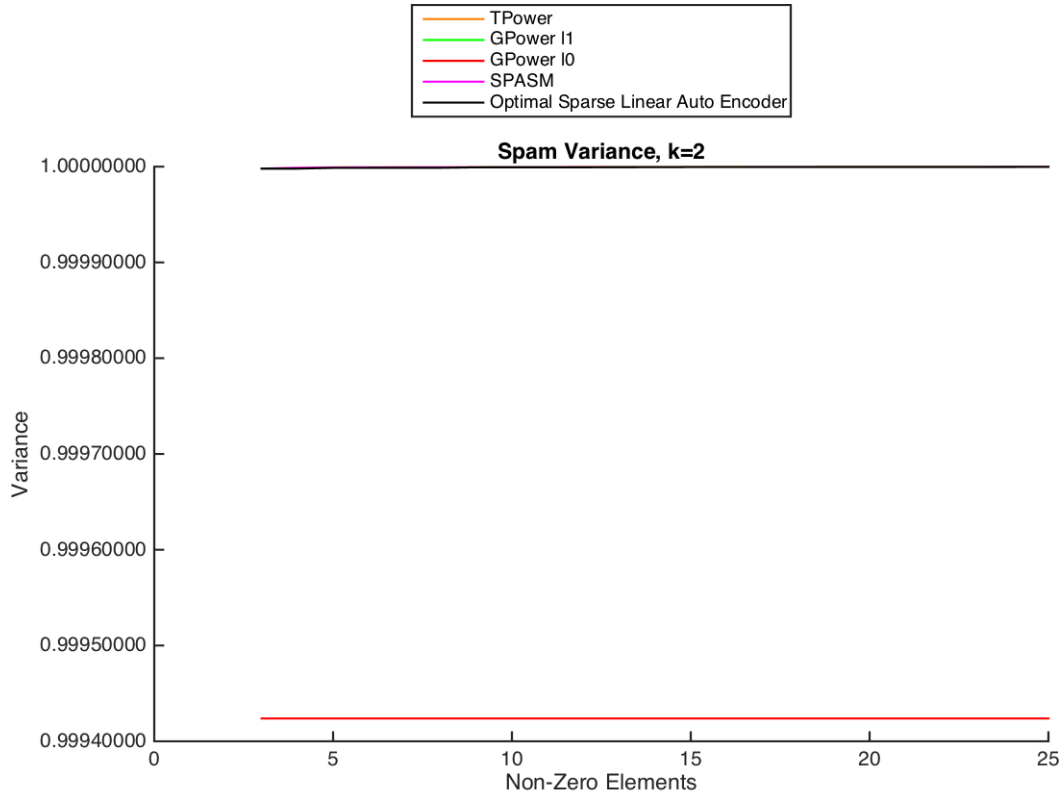


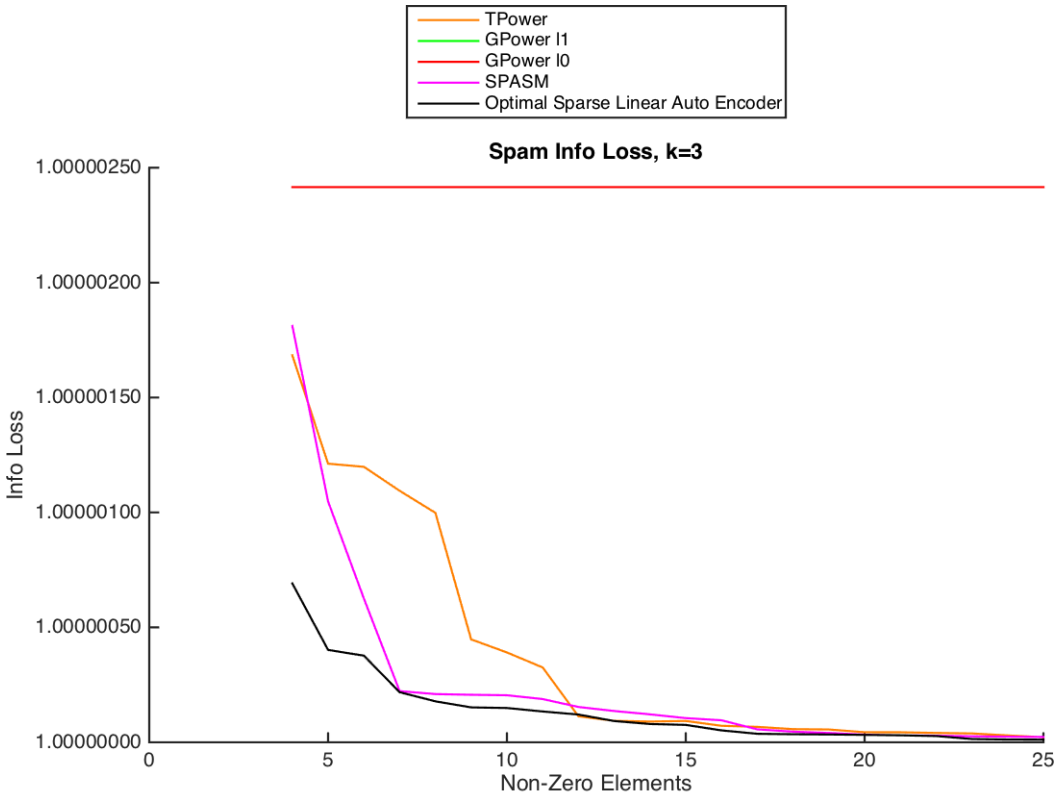


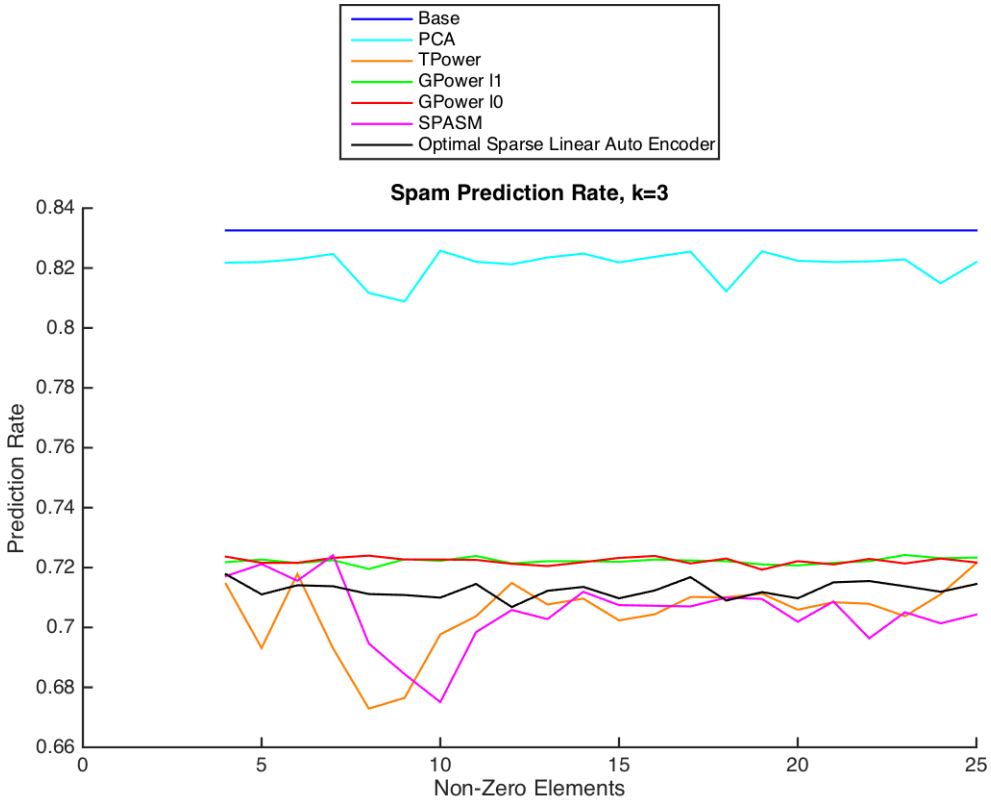

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Spam Variance, k=1
Variance
1.00000000
0.99500000
0.99000000
0.98500000
0.98000000
0.97500000
0.97000000
0
5
10
15
20
25
Non-Zero Elements

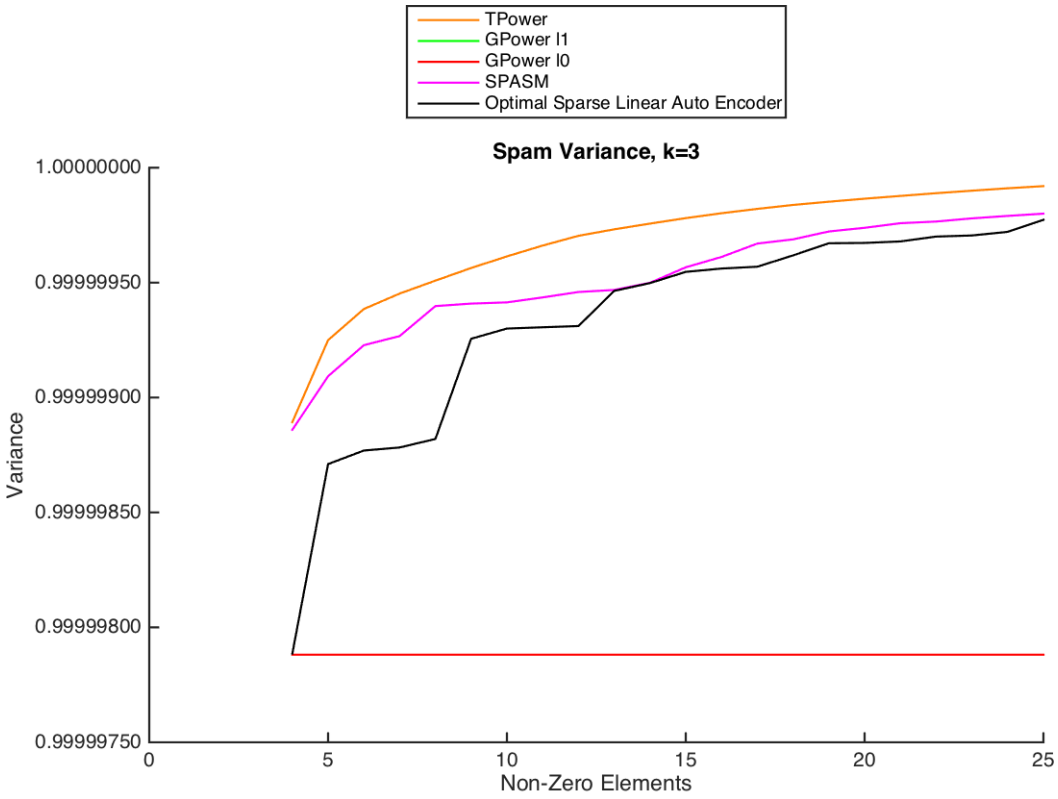

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Spam Info Loss, k=2
Info Loss
1.00800000
1.00700000
1.00600000
1.00500000
1.00400000
1.00300000
1.00200000
1.00100000
1.00000000
0
5
10
15
20
25
Non-Zero Elements

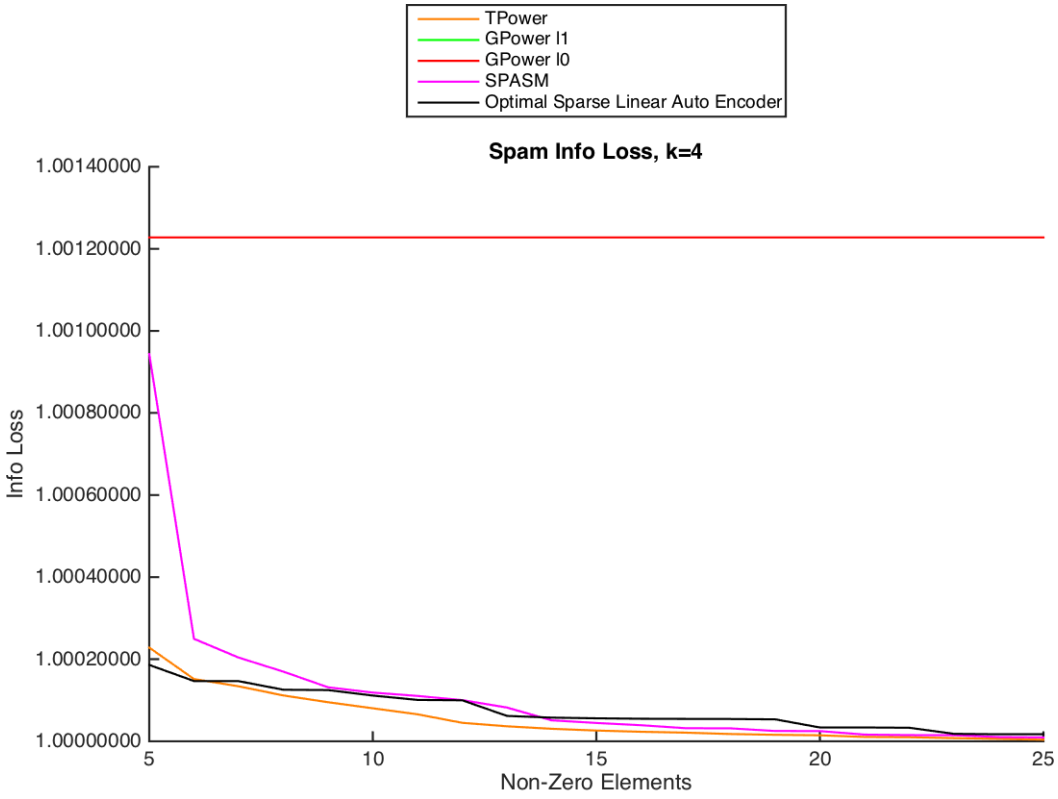

Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Spam Prediction Rate, k=2
Prediction Rate
Non-Zero Elements
0.84
0.82
0.8
0.78
0.76
0.74
0.72
0.7
0.68
0.66
0.64
0
5
10
15
20
25

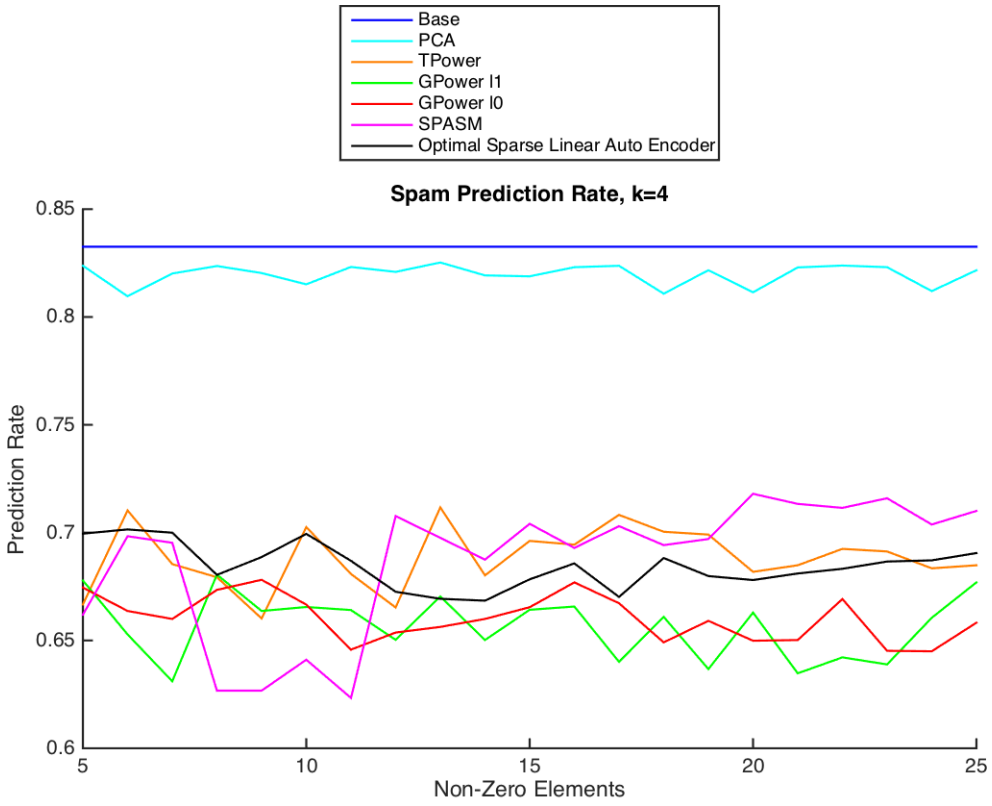

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Spam Variance, k=2
Variance
Non-Zero Elements
1.00000000
0.99990000
0.99980000
0.99970000
0.99960000
0.99950000
0.99940000
0
5
10
15
20
25

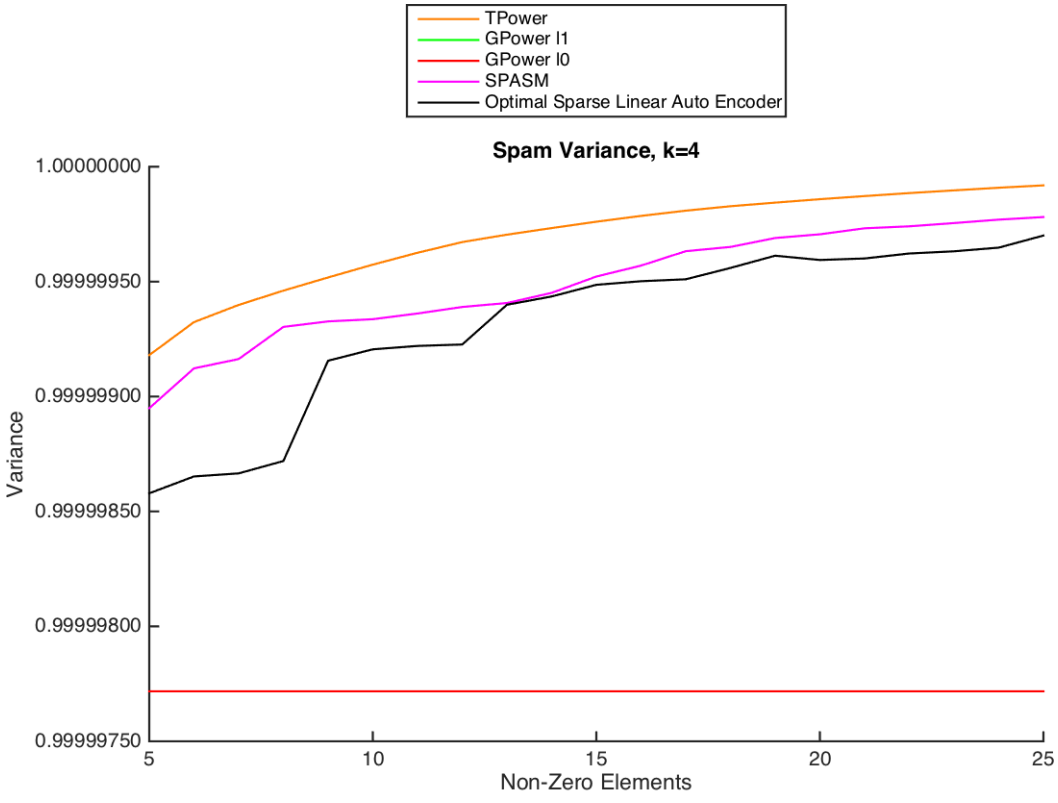

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Spam Info Loss, k=3
Info Loss
1.00000250
1.00000200
1.00000150
1.00000100
1.00000050
1.00000000
0
5
10
15
20
25
Non-Zero Elements

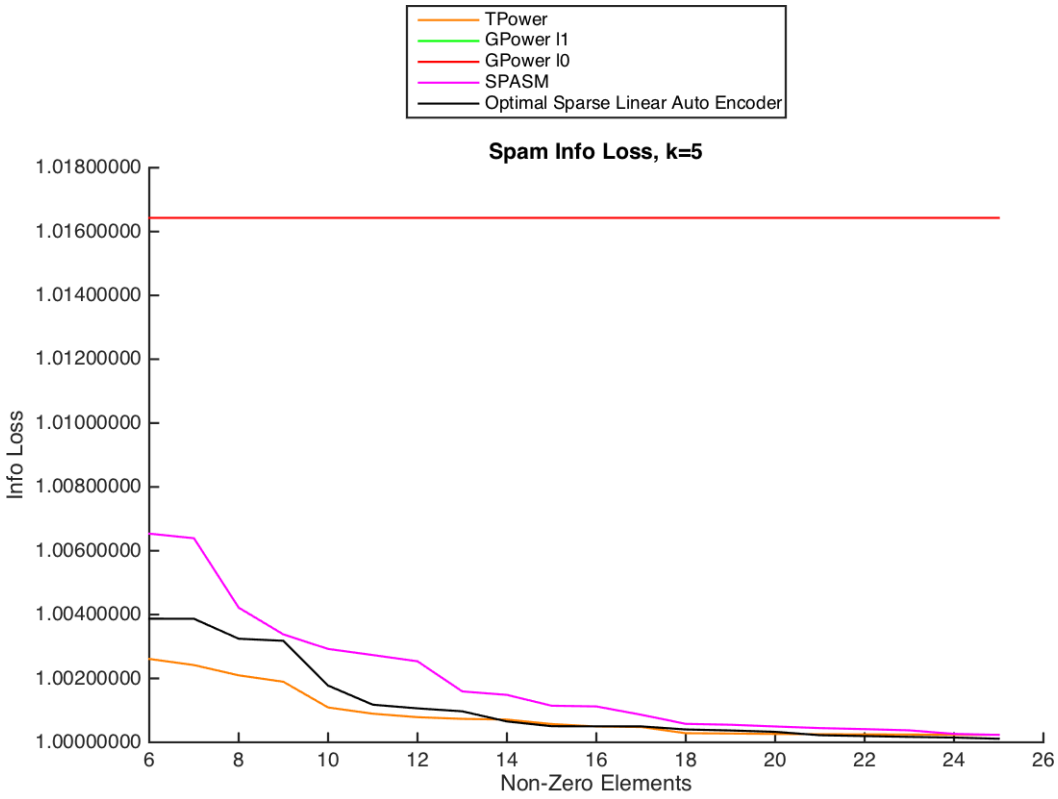

Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Spam Prediction Rate, k=3
Prediction Rate
0.84
0.82
0.8
0.78
0.76
0.74
0.72
0.7
0.68
0.66
0
5
10
15
20
25
Non-Zero Elements

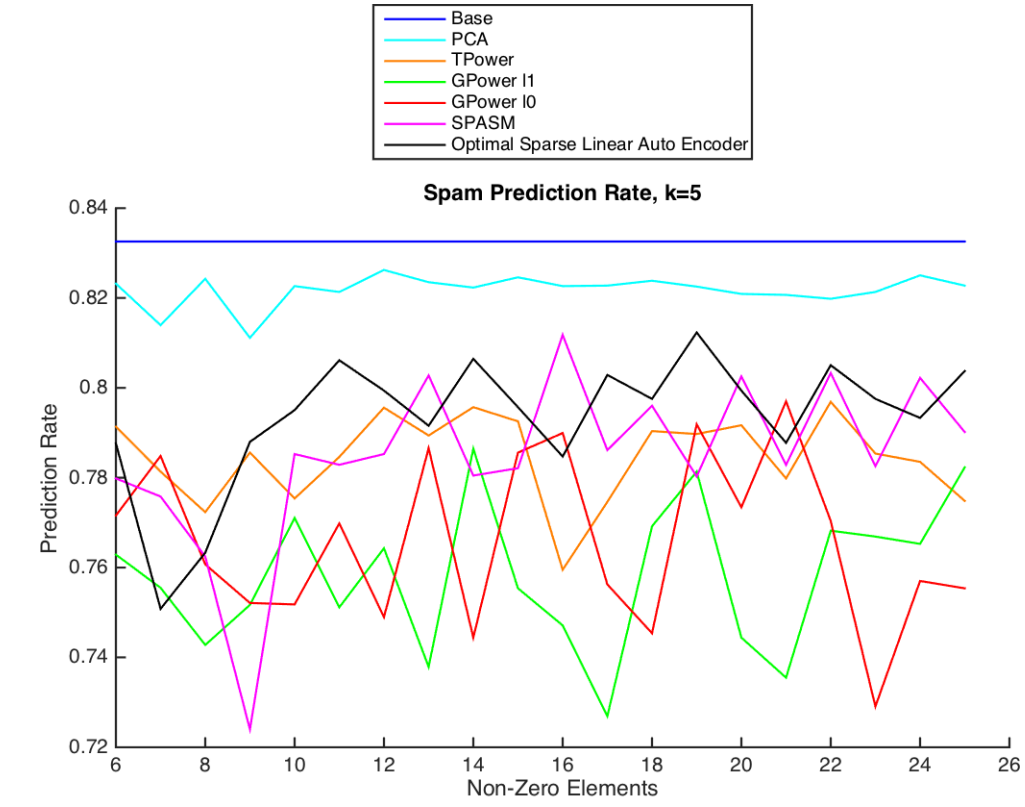

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Spam Variance, k=3
Variance
Non-Zero Elements
1.00000000
0.99999950
0.99999900
0.99999850
0.99999800
0.99999750
0
5
10
15
20
25

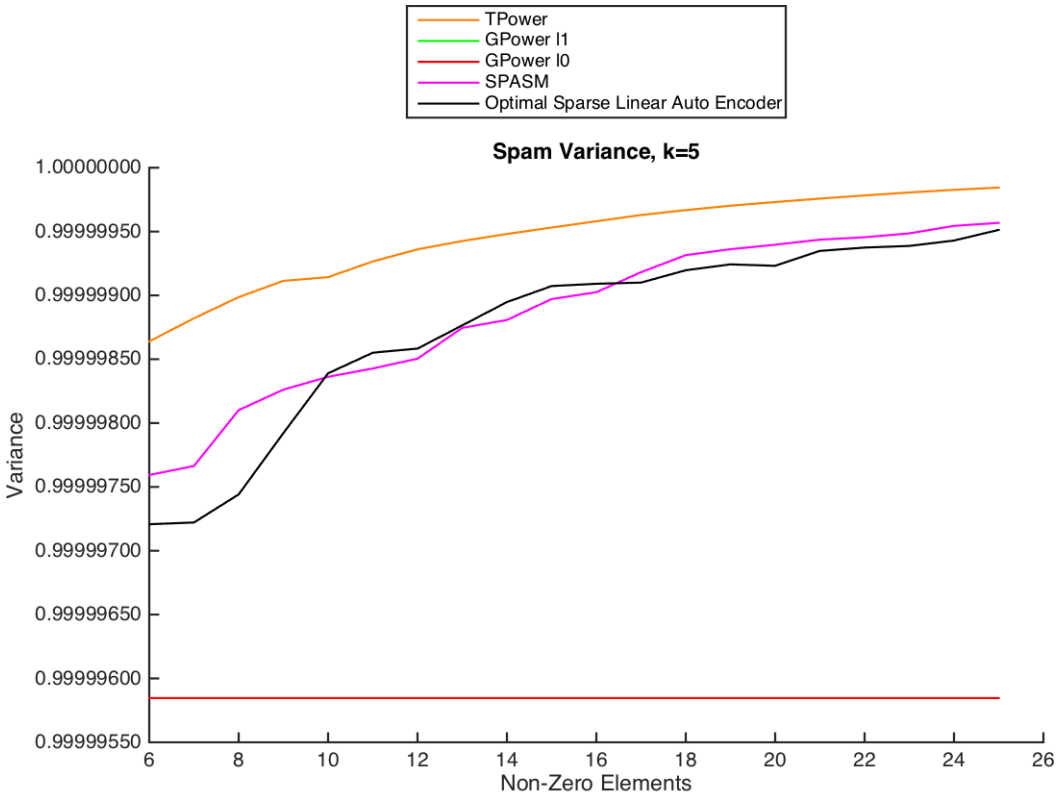

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Spam Info Loss, k=4
Info Loss
Non-Zero Elements
1.00140000
1.00120000
1.00100000
1.00080000
1.00060000
1.00040000
1.00020000
1.00000000
5
10
15
20
25

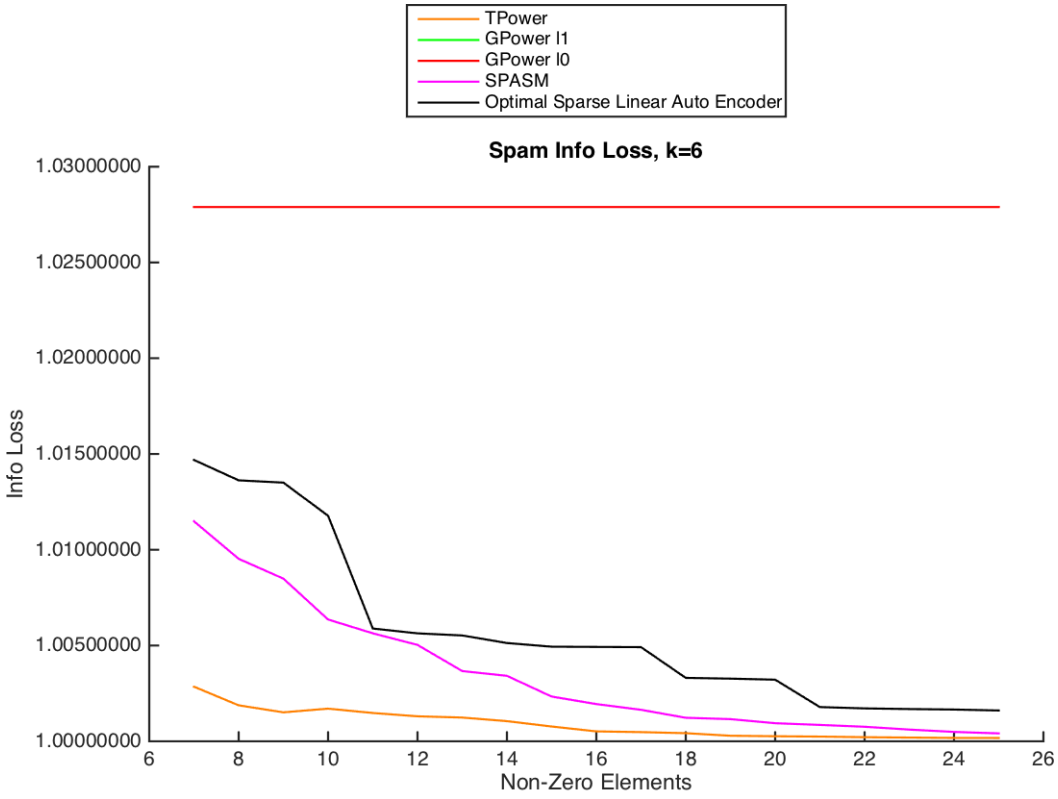

Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Spam Prediction Rate, k=4
Prediction Rate
Non-Zero Elements
0.85
0.8
0.75
0.7
0.65
0.6
5
10
15
20
25

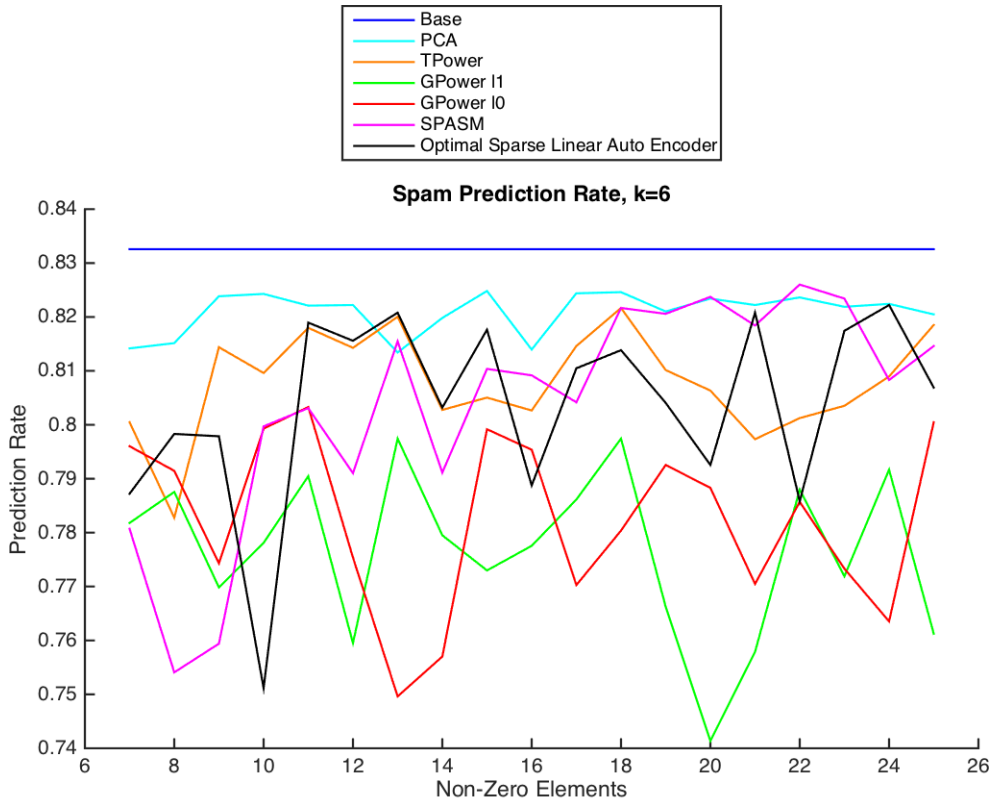

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Spam Variance, k=4
Variance
Non-Zero Elements
1.00000000
0.99999950
0.99999900
0.99999850
0.99999800
0.99999750
5
10
15
20
25

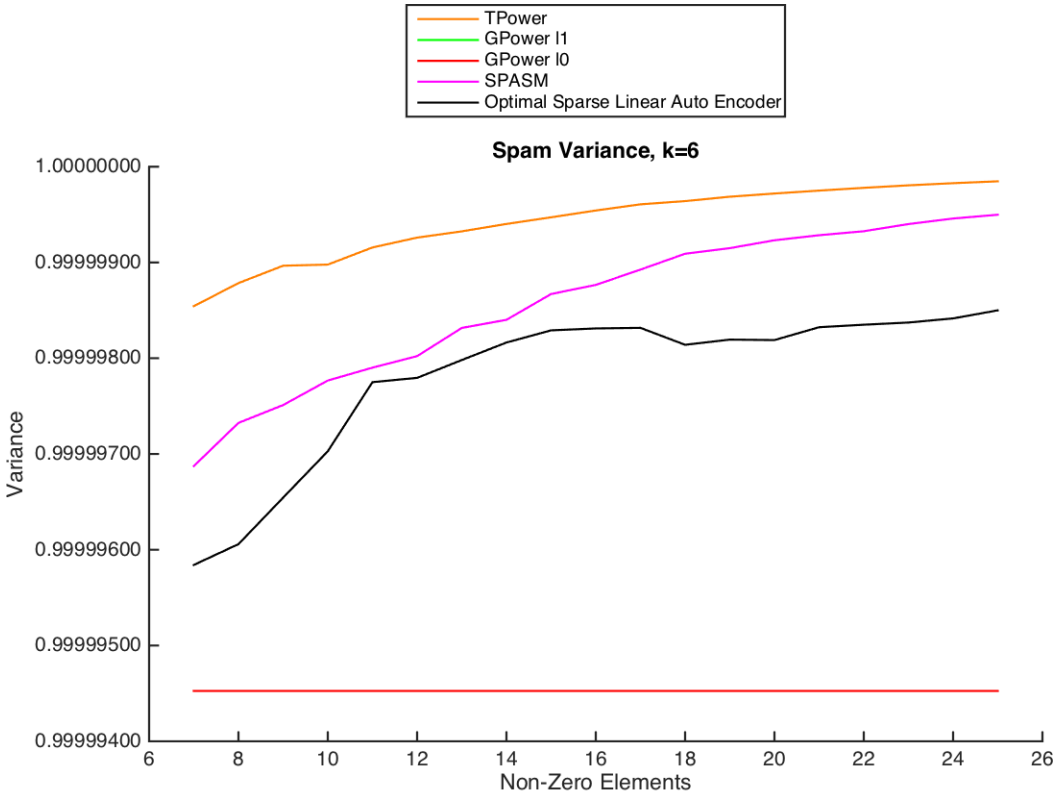

TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Spam Info Loss, k=5
Info Loss
Non-Zero Elements
1.01800000
1.01600000
1.01400000
1.01200000
1.01000000
1.00800000
1.00600000
1.00400000
1.00200000
1.00000000
6
8
10
12
14
16
18
20
22
24
26


Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Spam Prediction Rate, k=5
Prediction Rate
Non-Zero Elements
0.84
0.82
0.8
0.78
0.76
0.74
0.72
6
8
10
12
14
16
18
20
22
24
26


TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Spam Variance, k=5
Variance
Non-Zero Elements
1.00000000
0.99999950
0.99999900
0.99999850
0.99999800
0.99999750
0.99999700
0.99999650
0.99999600
0.99999550
6
8
10
12
14
16
18
20
22
24
26


TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Spam Info Loss, k=6
Info Loss
Non-Zero Elements
1.03000000
1.02500000
1.02000000
1.01500000
1.01000000
1.00500000
1.00000000
6
8
10
12
14
16
18
20
22
24
26


Base
PCA
TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Spam Prediction Rate, k=6
Prediction Rate
Non-Zero Elements
0.84
0.83
0.82
0.81
0.8
0.79
0.78
0.77
0.76
0.75
0.74
6
8
10
12
14
16
18
20
22
24
26


TPower
GPower l1
GPower l0
SPASM
Optimal Sparse Linear Auto Encoder
Spam Variance, k=6
Variance
Non-Zero Elements
1.00000000
0.99999900
0.99999800
0.99999700
0.99999600
0.99999500
0.99999400
6
8
10
12
14
16
18
20
22
24
26