bioRxiv preprint doi: https://doi.org/10.1101/207225; this version posted January 15, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

So you think you can PLS-DA?

Daniel Ruiz-Perez[1] and Giri Narasimhan[1]

***Abstract*— Partial Least-Squares Discriminant Analysis (PLS-DA) is a popular machine learning tool that is gaining increasing attention as a useful feature selector and classifier. In an effort to understand its strengths and weaknesses, we performed a series of experiments with synthetic data and compared its performance to its close relative from which it was initially invented, namely Principal Component Analysis (PCA). We demonstrate that even though PCA ignores the information regarding the class labels of the samples, this unsupervised tool can be remarkably effective as a dimensionality reducer and a feature selector. In some cases, it outperforms PLS-DA, which is made aware of the class labels in its input.**

Our experiments range from looking at the signal-to-noise ratio in the feature selection task, to considering many practical distributions and models for the synthetic data sets used. Our experiments consider many useful distributions encountered when analyzing bioinformatics and clinical data, especially in the context of machine learning, where it is hoped that the program automatically extracts and/or learns the hidden relationships.

I. INTRODUCTION

Partial Least-Squares Discriminant Analysis (PLS-DA) is a multivariate dimensionality-reduction tool [15], [2] that has been popular in the field of chemometrics for well over two decades [9], and has been recommended for use in *omics* data analyses. PLS-DA is gaining popularity in metabolomics and in other integrative omics analyses [19], [18], [14]. Both chemometrics and omics data sets are characterized by large volume, large number of features, noise and missing data [2], [8]. These data sets also often have lot fewer samples than features.

PLS-DA can be thought of as a "supervised" version of *Principal Component Analysis* (PCA) in the sense that it achieves dimensionality reduction but with full awareness of the class labels.

Besides its use as for dimensionality-reduction, it can be adapted to be used for feature selection [5] as well as for classification [12], [13], [16], [3].

As its popularity grows, it is important to understand that its role in discriminant analysis can be easily misused and misinterpreted [2], [4]. Since it is prone to the problem of *overfitting*, *cross-validation* is an important step in using PLS-DA as a feature selector and a classifier [17].

Furthermore, precious little is known about the performance of PLS-DA for different kinds of data. In this paper, we use a series of experiments to shed light on the strengths and weaknesses of PLS-DA vis-à-vis PCA, as well as the kinds of data distributions where PLS-DA is potentially useful and where it fares poorly.

[1]Bioinformatics Research Group (BioRG), School of Computing and Information Sciences, Florida International University, Miami FL 33199.

II. THEORETICAL BACKGROUND

The objective of dimensionality-reduction methods such as PCA and PLS-DA is to arrive at a linear transformation that transforms the data to a lower dimensional space with as small an error as possible. If we think of the original data matrix to be a collection of n m-dimensional vectors (i.e., $\mathbf{X}$ is a $n \times m$ matrix), then the above objective can be thought of as that of finding a $m \times d$ transformation matrix $\mathbf{A}$ that optimally transforms the data matrix $\mathbf{X}$ into a collection of n d-dimensional vectors $\mathbf{S}$. Thus, $\mathbf{S} = \mathbf{XA} + \mathbf{E}$, where $\mathbf{E}$ is the error matrix. The matrix $\mathbf{S}$, whose rows correspond to the transformed vectors, gives d-dimensional *scores* for each of the n vectors in $\mathbf{X}$.

In PCA, the transformation preserves (in its first principal component) as much variance in the original data as possible. In PLS-DA, the transformation preserves (in its first principal component) as much covariance as possible between the original data and its labeling. Both can be described as iterative processes where the error term is used to define the next principal component.

As mentioned earlier, PCA and PLS-DA are considered as unsupervised and supervised counterparts of dimensionality-reduction tools. The new features representing the reduced dimensions are referred to as *principal components*. A simple example can highlight the differences in their approaches. Figure 1 shows an example of a synthetic data set for which the principal component chosen by PCA points to the bottom right, while the one chosen by PLS-DA is roughly orthogonal to it pointing to the bottom left.

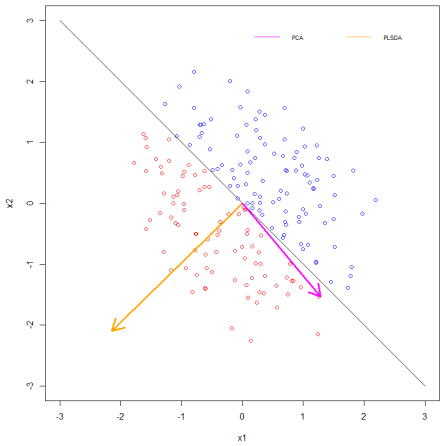


Fig. 1: Comparing the first principal component computed by PCA (pink) versus that computed by PLS-DA (orange)

bioRxiv preprint doi: https://doi.org/10.1101/207225; this version posted January 15, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

PCA : Figure 1 and the associated text gave an intuitive difference between PCA and PLS-DA. In what follows, we provide a brief algorithmic and mathematical explanation of the differences. Informally, the PCA algorithm calculates the first principal component along the first eigenvector, by minimizing the projection error (i.e., by minimizing the average squared distance from each data to its projection on the principal component, or maximizing the variance of the projected data). After that, the algorithm iteratively projects all the points to a subspace orthogonal to the last principal component and then repeats the process on the projected points, thus constructing an orthonormal basis of eigenvectors and principal components. An alternative formulation is that the principal component vectors are given by the eigenvectors of the non-singular portion of the covariance matrix $\mathbf{C}$, given by:

$$\mathbf{C} = \frac{1}{n-1}\mathbf{X}^T\mathbf{C_n}\mathbf{X}, \tag{1}$$

where $\mathbf{C_n}$ is the $n \times n$ centering matrix. The *loading vectors*, denoted by $\mathbf{L_1}, \ldots, \mathbf{L_n}$, are given in terms of the eigenvectors, $\mathbf{e_1}, \ldots, \mathbf{e_n}$ and the eigenvalues, $\lambda_1, \ldots, \lambda_n$, of $\mathbf{C}$ as follows:

$$\mathbf{L_i} = \sqrt{\lambda_i}\ \mathbf{e_i}, \quad i = 1, \ldots, n, \tag{2}$$

PLS-DA : As with PCA, the principal components of PLS-DA are linear combinations of features, and the number of these components defines the dimension of the transformed space. In a standard variant of PLS-DA, the components are required to be orthogonal to each other (although this is not necessary). This is employed in the package `mixOmics` [11]. In a manner similar to Eq. (1), the principal components of PLS-DA can be formulated as the eigenvectors of the non-singular portion of the covariance matrix $\mathbf{C}$, given by:

$$C = \frac{1}{(n-1)^2}\mathbf{X^T}\mathbf{C_n}\mathbf{y}\mathbf{y^T}\mathbf{C_n}\mathbf{X}, \tag{3}$$

where $\mathbf{y}$ is the class label vector.

The iterative process computes the transformation vectors (also, *loading vectors*) $\mathbf{a}_1, \ldots, \mathbf{a}_d$, which give the *importance* of each feature in that component. In iteration h, PLS-DA has the following objective:

$$\max_{(\mathbf{a}_h, b_h)} \operatorname{cov}(\mathbf{X}_h\mathbf{a}_h, \mathbf{y}_h b_h), \tag{4}$$

where b_h is the loading for the label vector $\mathbf{y}_h$, $\mathbf{X}_1 = \mathbf{X}$, and $\mathbf{X}_h$ and $\mathbf{y}_h$ are the residual (error) matrices after transforming with the previous $h-1$ components.

SPLS-DA: Variant of PLS-DA that make a *sparsity* assumption, i.e., that only a small number of features are responsible for driving a biological event or effect under study have been devised [6], [10] and shown to be successful with applications where the number of features far outnumber the number of samples [7]. Using lasso penalization, these methods add penalties (L_1 or L_0 norm) to better guide the feature selection and model fitting process and achieve improved classification by allowing to select a subset of the covariates instead of using all of them.

III. Experimental Results

In this section, we discuss a variety of experiments with synthetic data that will help us explain the strengths and weaknesses of PLS-DA in comparison to PCA.

A. Synthetic Data for the Experiments

The following describes a standard experimental setup. Clarifications are provided wherever the experiments differed from this norm. For each of the experiments, labeled synthetic data were generated as follows. Input consisted of the parameters n, the number of samples and m, the number of features. Every data set assumed that there was a *rule* (e.g., a linear inequality), which was a function of some subset of the m features (i.e., *signal* features), while the rest were considered as *noise* features. The input parameter also included the rule and consequently the set of signal features. This rule will be considered as the *ground truth*. PLS-DA was then applied to the data set to see how well it performed feature selection (i.e., identified the important features in the rule) or how well it classified (i.e., identified the labels of the data points). All experiments were executed using PCA and sparse PLS-DA (sPLS-DA). where the loading vector is only non-zero for the selected features. Both are available in the *mixOmics* R package, which was chosen because it is the implementation most used by biologists and chemists. The noise features of all points are generated from a random distribution that is specified as input to the data generator. The default is assumed to be the uniform distribution. The generation of the signal features is dictated by the rule that they satisfy.

B. Performance Metrics for the Experiments

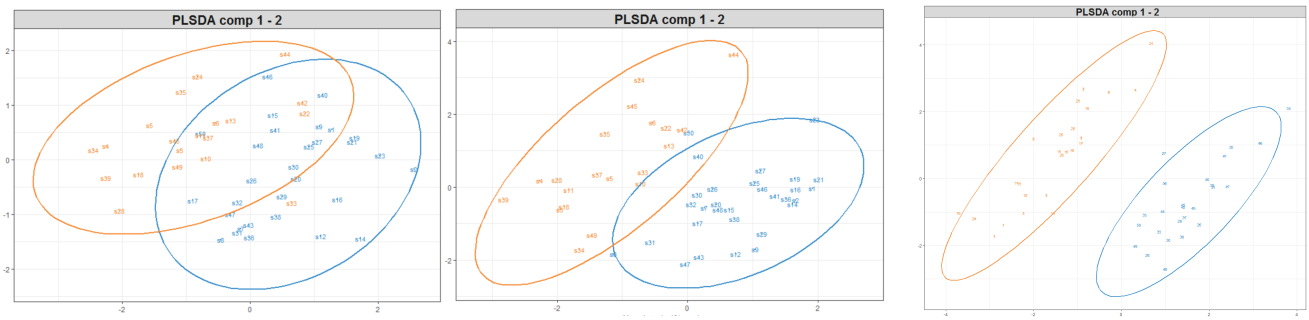
As is standard with experiments in machine learning, we evaluated the experiments by computing the following measures: true positives (tp), true negatives (tn), false positives (fp), false negatives (fn), precision ($tp \div (tp + fp)$), and recall ($tp \div (tp + fn)$). Note that in our case precision and recall are identical because $fn = fp$. Since tn is large in all our feature extraction experiments, some of the more sophisticated measures are skewed and therefore not useful. For example, the F1 score ($2 * tp \div (2 * tp + tn + fp + fn)$) will be necessarily low, while accuracy ($(tp + tn) \div (tp + tn + fp + fn)$) and specificity ($2 * tn \div (tn + fn)$) will be extremely high. When the number of noise features is low, precision could be artificially inflated. However, this is not a likely occurrence in real experiments.

All performance graphs are shown as 3D plots (except in Section III-C) where the z axis represents the performance measure of choice, while the x and y axes represent important parameters of the experiment, unless otherwise noted.

C. Experiments varying n/m

We first show how the performance of PLS-DA can be affected and the number of spurious relationships found can

bioRxiv preprint doi: https://doi.org/10.1101/207225; this version posted January 15, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.



(a) 50 Samples x 25 Features (b) 50 Samples x 50 Features (c) 50 Samples x 100 Features

Fig. 2: Separability of random points as the ratio of number of samples to number of features is decreased.

vary when the ratio of number of samples, n, to the number of features, m, is varied.

As described in Section III-A, we generated n random data points in m-dimensional space (from a uniform distribution) and labeled them randomly. The ratio n/m was reduced from 2:1 to 1:1 to 1:2. Given the data set, it is clear that any separation of the data found by any method is merely occurring fortuitously. When we have at least twice as many features as samples, PLS-DA readily finds a hyperplane that perfectly separates both merely by chance. As shown in Figure 2, the two randomly labeled groups of points become increasingly separable. These experiments only range in ratios from 2:1 to 1:2. In many current day omics data sets, this ratio can even exceed 1:1000. (For example, data sets with 50 samples and 50,000 genes are commonplace.)

If any separating hyperplane is used as a *rule* to discriminate blue points from orange points, then even though the apparent error rate (AE) decreases for this set, its ability to discriminate any new random points will remain dismal [1]. In fact, the cross-validation overall error rate for the first principal component in the three experiments shown in Figure 2 were 0.52 ± 0.036, 0.58 ± 0.05 and 0.60 ± 0.03, showing that even though separability increased, the errors remain unreasonably large.

D. Experiments using PLS-DA as a feature selector

In this section, we show our experiments with PLS-DA as a feature selector. We used 3 sets of methods for generating the synthetic points. In the first set, we consider point sets that are linearly separable. In the second we assume that the membership of the points in a class is determined by whether selected signal features lie in prespecified ranges. Finally, we perform experiments with clustered points.

1) Experiments with Linearly Separable Points: For these experiments we assume that the data consist of a collection of n random points with s signal features and $m - s$ noise features. They are labeled as belonging to one of two classes using a randomly selected linear separator given as a function of only the signal features. The experiments were meant to test the ability of PLS-DA (used as a feature extractor) to correctly identify the signal features. The performance scores shown in Figure 3 were averaged over 100 repeats. Note that the performance metric measured the number of signal features identified by PLS-DA as significant. The linear model used implements the following rule $\mathcal{R}_1$:

$$\mathcal{R}_1 : \quad \sum_{i=1}^{s} s_i \geq C \Rightarrow \text{class } 0, \text{else class } 1 \tag{5}$$

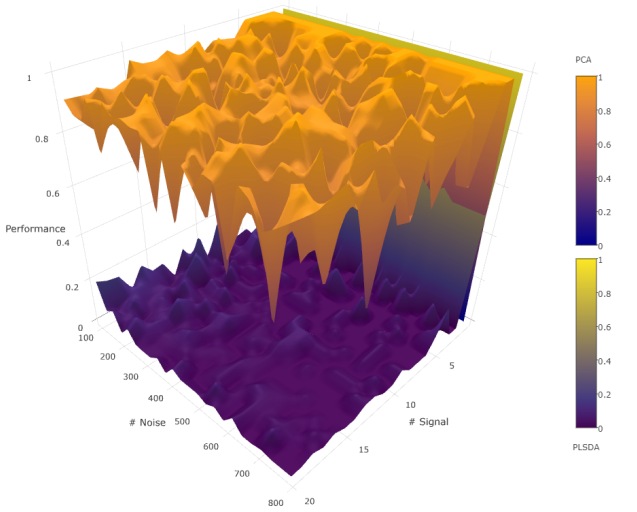
Two sets of experiments were performed. In the first set, n was fixed at 200, but s and m were varied. (See Figure 3 (a).) In the second set, s was fixed at 10, but n and m were varied. (See Figure 3 (b).) PCA consistently outperformed PLS-DA in all these experiments with linear relationships. Also, when the number of samples was increased, the performance of PCA improved, which makes sense because there is more data from which to learn the relationship. However, it did not help PLS-DA.

The *loading vector* is a reflection of what PCA or PLS-DA guessed as the linear relationship between the features. We, therefore, set out to verify how far was the linear relationship that was guessed by the tools used. Even if the tools picked many noise features, we wanted to see how they weighted the noise and signal features they picked. Toward this goal, we ran an extra set of experiments with the model shown above to see if the loading vector from PLS-DA indicated a better performance than what might be suggested by Figure 3. Note that ideally the loading vector should have 0s for the noise features and 1s for the signal features. We computed the cosine distance between the loading vector computed in the experiment and the vector reflected by the true relationship. As shown in Figure 4, we see that the loading vectors of both PCA and PLS-DA failed to reflect the true relationship. These experiments were performed using $n = 200$ averaged over 100 repetitions. Even though PCA successfully selected many of the signal features during feature selection, it was unable to get sufficiently close to the underlying linear relationship, perhaps because of the compositional nature of the signal variables, which gives rise to correlations.

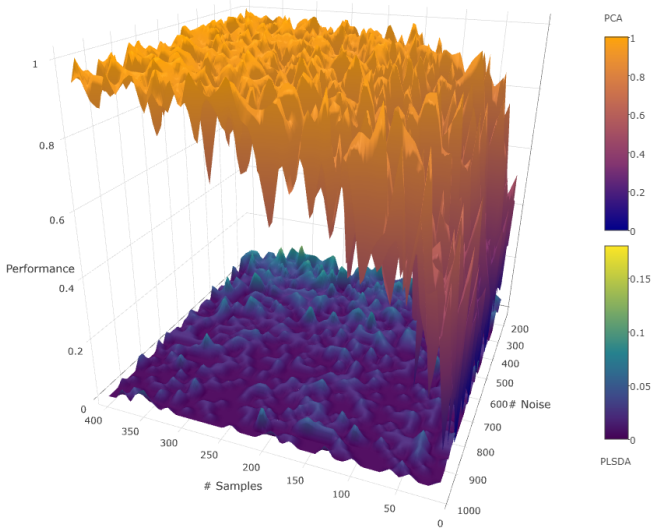
Other related experiments include the following:

- Changing the magnitude of constant in the inequality;
- Changing the relationship from a linear inequality to two linear equalities, i.e., the two classes of points lie on two randomly chosen hyperplanes in the subspace of the signal features.

bioRxiv preprint doi: https://doi.org/10.1101/207225; this version posted January 15, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.



(a) Signal[1,20] vs Noise[100,800]



(b) Samples[10,400] vs Noise[150,1000]

Fig. 3: Linear relationship Signal vs Noise and Samples vs Noise. Note that PCA is successful only because the features that are the signal are the only correlated variables.

- Constructing rules described as the disjunction of two simple rules, as shown below:

$$\mathcal{R}_2 : \sum_{i=1}^{\lfloor s/2 \rfloor} s_i = C_0 \vee \sum_{i=1}^{\lfloor s/2 \rfloor} s_i = C_0 \geq C \quad (6)$$
$$\Rightarrow \text{class } 0, \text{else class } 1$$

2) Interval model: In this set of experiments, we tried PCA and PLS-DA on data sets where the rules that determined class membership are often encountered in biological data sets. A typical rule assumes that class membership is determined by a select set of (signal) features being constrained to lie in specific ranges of values (intervals), while those that don't belong to the class are unconstrained. For example, children are diagnosed with *attention deficit and hyperactivity disorder* (ADHD) based on whether or not their performance levels fall in specified ranges on multiple psychological tests. We call this model as the *interval model*.

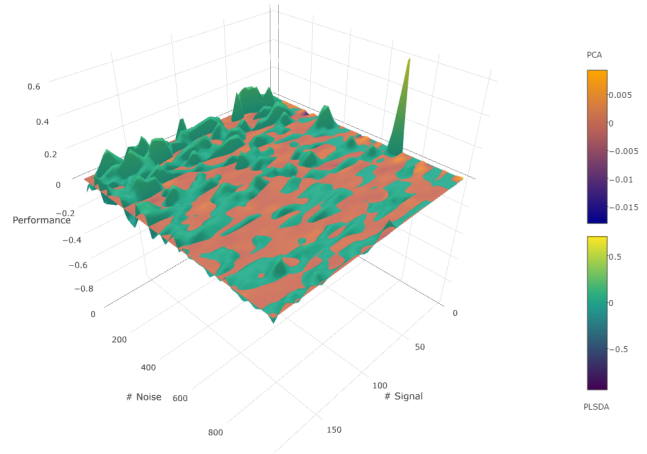
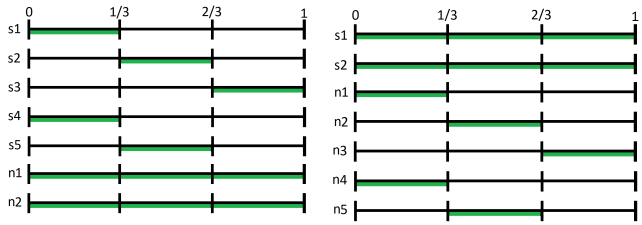


Fig. 4: Cosine distance to the real signal. Signal[2,200] vs Noise[0,1000]

We tried two different ways to generate data from this interval model. In the first version, we constrained the signal features. To generate such data sets, members of the class had signal features selected uniformly at random from prespecified intervals, while all other features were generated from a uniform distribution in the range $[0, 1]$. In the second version, we constrained the noise features. Here, members of the class had noise features selected uniformly at random from prespecified intervals, while all the signal features were generated from a uniform distribution in the range $[0, 1]$.



(a) Constraining signal (b) Constraining noise

Fig. 5: Two different ways of generating interval data

We divided the range $[0, 1]$ into subintervals of width $1/p$. Figure 5 shows an example with $p = 3$, but experiments were carried out with $p = 3, 5$ and 10. Depending on the experiment, signal and noise feature were assigned to either a subinterval of width $1/p$ or the entire interval of $[0, 1]$. Figure 5 shows shaded green areas that specify valid ranges for signal features (or noise features, as the case may be).

Figure 5a shows the setup of experiments with the first approach (constrained signal feature values) and was executed with 200 samples and repeated 100 times.

Figure 5b shows the setup of experiments with the second approach (constrained noise features)

bioRxiv preprint doi: https://doi.org/10.1101/207225; this version posted January 15, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

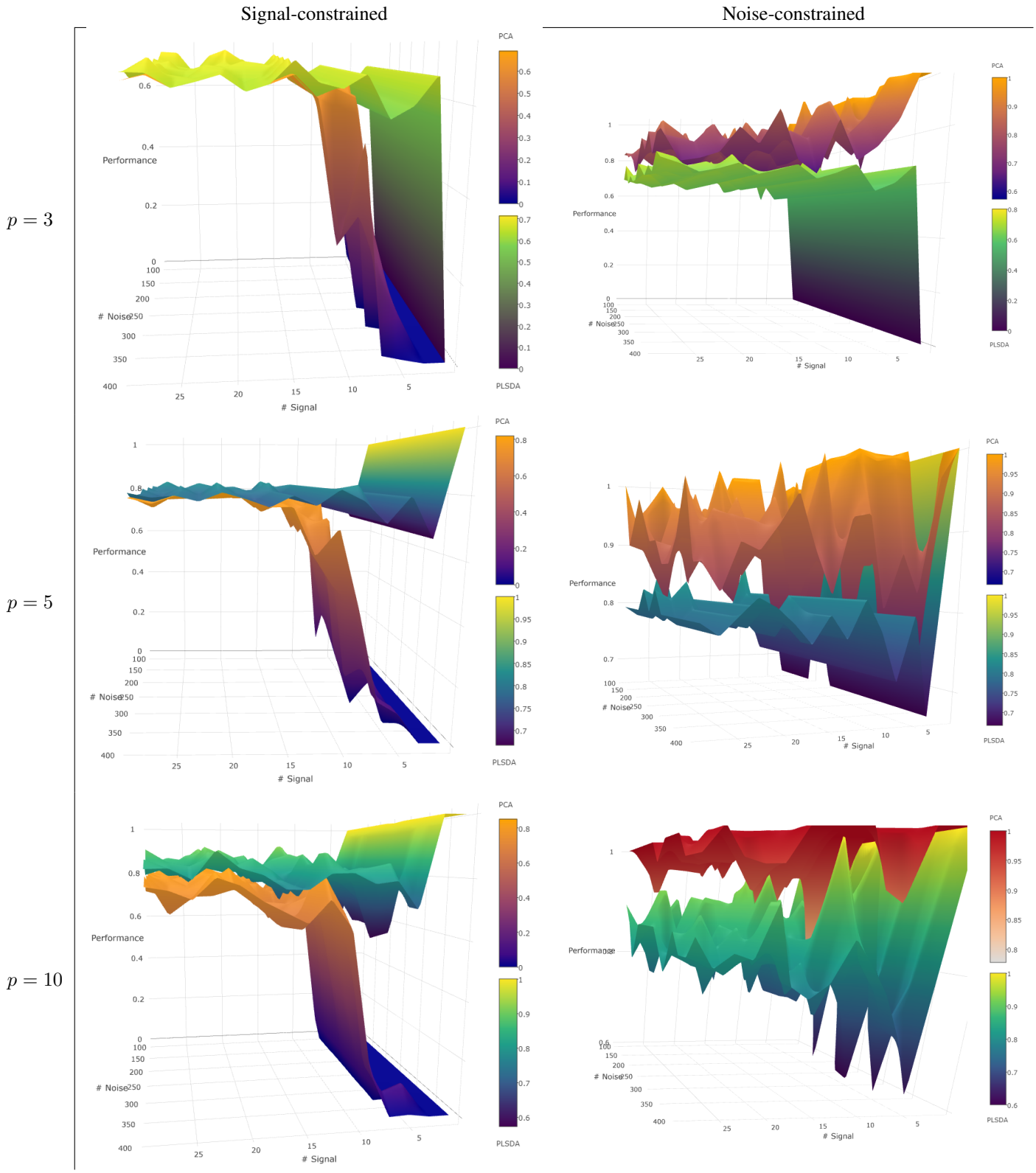


Fig. 6: Experiments with the interval model. Each plot shows the number of signal features ranging from 1 through 30 and the number of noise features ranging from 100 to 400

The results are shown in Figure 6. The axes for each of the 3d plot represent the number of signal features, number of noise features, and the measure of performance.

We note that for the first approach when the signal features are constrained, PLS-DA consistently outperforms PCA. This is because there is a strong correlation between the signal

bioRxiv preprint doi: https://doi.org/10.1101/207225; this version posted January 15, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

features for class members, and PLS-DA is able to detect that correlation. On the other hand, for the second approach where the noise features are constrained, PCA consistently outperforms PLS-DA.

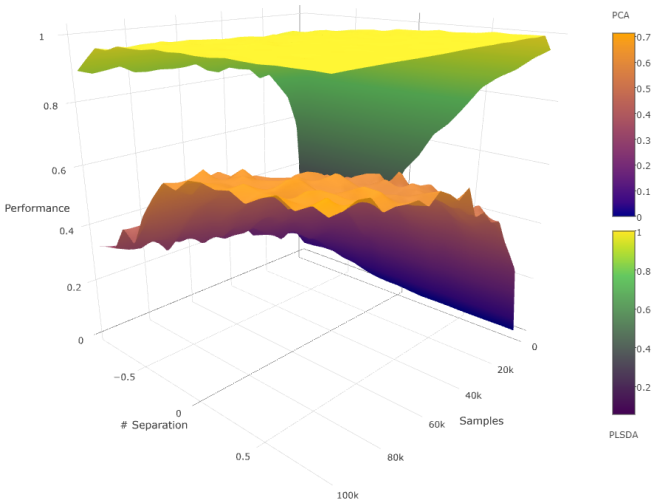
PLS-DA performs poorly when the number of signal features is 1 and $p = 3$, because the distribution of values for the single signal feature is not very different from the distribution of the noise features. On the other hand PLS-DA succeeds when the constraints for the intervals are stronger (when $p = 5$ or 10).

3) Cluster model: For these experiments, the signal features of the points were generated from a clustered distribution with two clusters separated by a prespecified amount. All noise features were generated from a uniform distribution with mean 0.

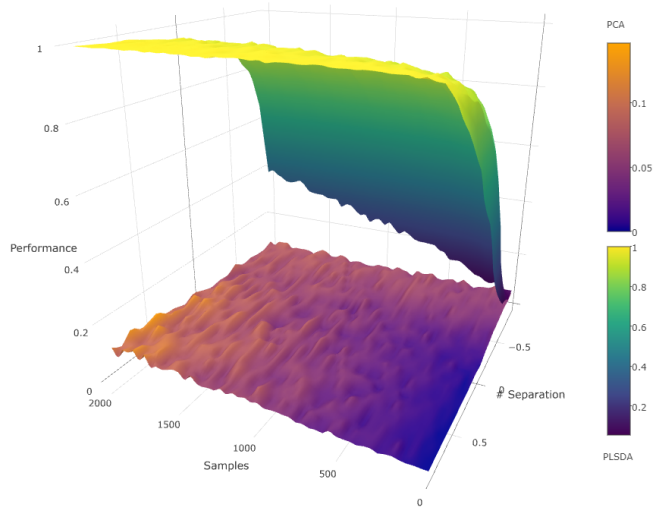
The R package *clusterGeneration* was used for this purpose, which also allows control over the separation of the clusters. Cluster separation between the clusters was varied in the range $[-0.9, 0.9]$. Thus when the points are viewed only with the noise features, they appear like a uniform cloud, and when viewed only with the signal features, the members of the two classes are clustered. Note that cluster separation of -0.9 will appear as indistinguishable clusters, while a separation of 0.9 will appear as well-separated clusters.

The experiments were executed with 10 signal and 200 noise features, averaged over 100 runs.

The executions with clustered data showed PLS-DA to be clearly superior to PCA. As shown in Figure 7, the difference narrows when the number of samples is made very large or the clusters are widely separated (i.e., cleanly separated data), but the difference remains significant. PLS-DA is able to select the correct hyperplane even with few samples and even when the separation between the clusters is low (values close to 0). PCA needs both an unreasonably large number of samples and very well separated clusters to perform respectably in comparison. However, data with high separation values are embarrassingly simple to analyze with a number of competing methods.



(a) Samples[10,100k] vs Separation[-0.9,0.9]



(b) Samples[10,2k] vs Separation[-0.9,0.9]

Fig. 7: Number of samples vs cluster separation

E. Experiments as a classifier

Our final set of experiments was to see how PLS-DA fared as a classifier with cross-validation. The results corroborated its performance as a feature selector. The following experiments were executed 100 times each, with 10 signal features. In all of the experiments carried out, there is a correspondence between a high performance as feature selector and a low cross-validation error.

As shown in Figure 8a, for the linear relationship model, its performance is around 0.5, which is no better than chance for a 2-class experiment. If the feature selection by PLS-DA was poor, then it should come as no surprise that the classifier fared equally poorly.

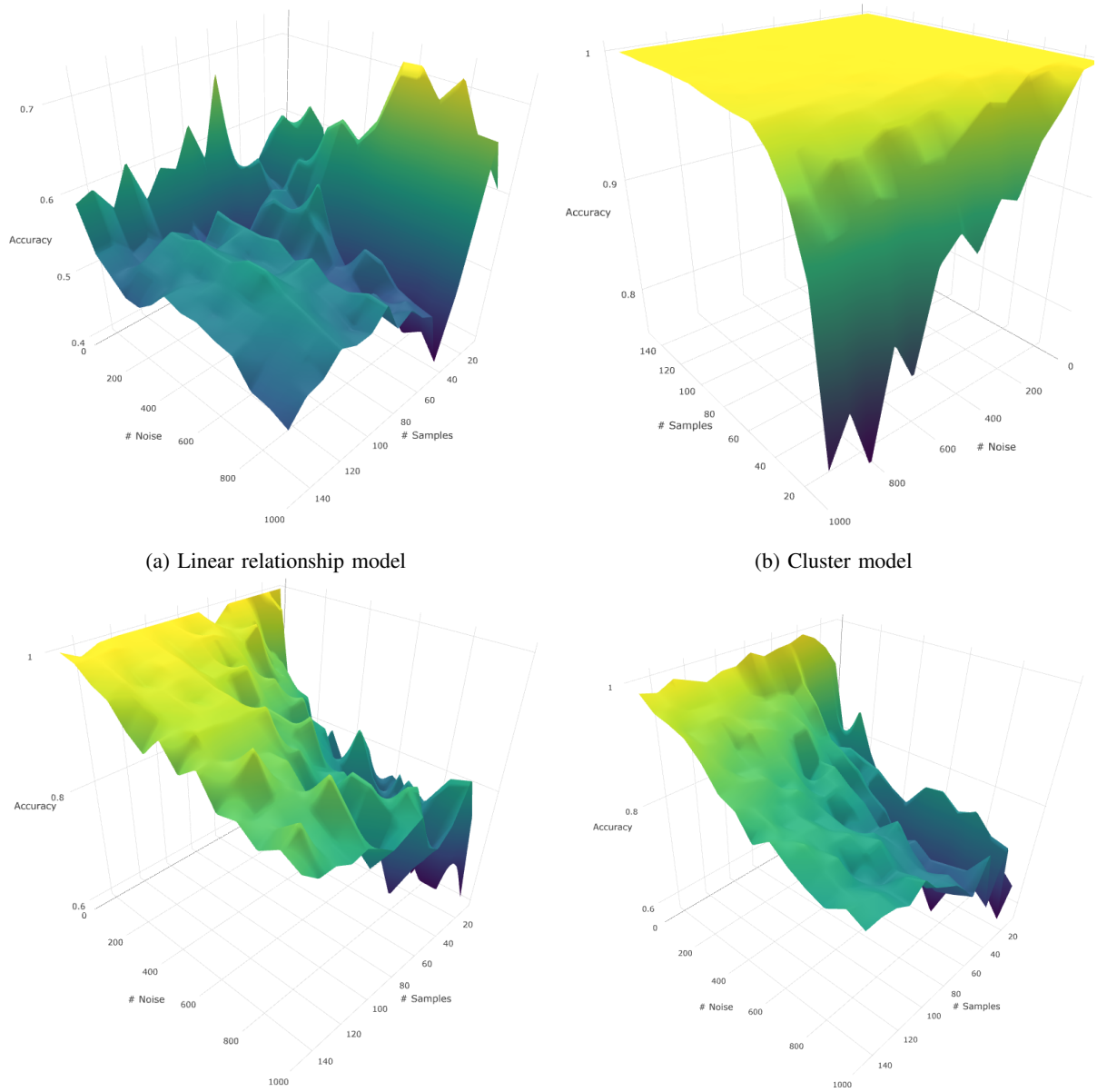
Figures 8c and 8d show the results of our experiments with the interval model. As in the case of the feature selection experiments, both versions performed roughly the same, classifying much better than chance and having their best performance when the number of samples was large and the number of noise features was low, again as expected.

For the results with the cluster model shown in Figure 8b, the cross-validation error is almost 0 in every case, except when the number of samples is low, which is again consistent with what we saw in the feature selection experiments. The performance gets noticeably worse when, in addition to a low number of samples, the number of noise features is large. This can be explained by observing that the signal is hidden among many irrelevant (noise) features, something that one has come to expect with all machine learning algorithms.

IV. Conclusions

The obvious conclusion from our experiments is that it is a terrible idea to use PLS-DA blindly with all data sets. In spite of its attractive ability to identify features that can separate the classes in the data set, it is clear that any data set with sufficiently large number of features is separable and

bioRxiv preprint doi: https://doi.org/10.1101/207225; this version posted January 15, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.



(a) Linear relationship model

(b) Cluster model

(c) Signal constrained 5 intervals model

(d) Noise constrained 5 intervals model

Fig. 8: Cross validation performance for the different models

that most of the separating hyperplanes are just "noise". Thus using it indiscriminately would turn into a "golden hammer", which is an oft-used, but inappropriate tool. Fortunately, the liberal use of cross-validation would readily point to when PLS-DA is being used ineffectively.

More significantly, our work sheds light on the kind of relationships and the kind of data models with which PLS-DA can be effective both as a feature selector as well as a classifier. In particular, we claim that when classes are determined by linear relationships, even simple unweighted relationships, PLS-DA provides almost no insight into the data. But, surprisingly, PLS-DA is reasonably effective when the classes have a clustered distribution on select (signal) features, even when these features are hidden among a large number of noise features. PLS-DA retains a strong performance even when the classes are contained in n-orthotopes (i.e., rectangular boxes in the subspace of the signal features).

In all of the experiments carried out there is a correspondence between high performance as feature selector and low cross-validation error. We have shown that just-by-chance good behaviors exist and different methods should be used for different datasets.

V. Future Work

There are two main directions for future improvements.

- What other models describe data sets in Bioinformatic and Clinical life sciences? And how does PLS-DA fare

bioRxiv preprint doi: https://doi.org/10.1101/207225; this version posted January 15, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

with such data sets? How does PLS-DA compare not just with its close relative, PCA, but with a wider suite of feature selection methods?

- Given the weaknesses of PLS-DA, can they be addressed by theoretically tweaking the method?

Supplementary Website

All the figures shown in this paper can be viewed interactively at the following URL, allowing for the plots to be rotated in all 3 dimensions: `http://biorg.cs.fiu.edu/plsda.html`

Acknowledgments

This work was partially supported by grants from the Department of Defense Contract W911NF-16-1-0494, NIH grant 1R15AI128714-01, and NIJ grant 2017-NE-BX-0001. The authors also thank the members of the Bioinformatics Reseach Group (BioRG) for their useful comments during the course of this research.

References

[1] Christophe Ambroise and Geoffrey J McLachlan. Selection bias in gene extraction on the basis of microarray gene-expression data. *Proceedings of the national academy of sciences*, 99(10):6562–6566, 2002.

[2] Matthew Barker and William Rayens. Partial least squares for discrimination. *Journal of chemometrics*, 17(3):166–173, 2003.

[3] Cristina Botella, Joan Ferré, and Ricard Boqué. Classification from microarray data using probabilistic discriminant partial least squares with reject option. *Talanta*, 80(1):321–328, 2009.

[4] Richard G Brereton and Gavin R Lloyd. Partial least squares discriminant analysis: taking the magic away. *Journal of Chemometrics*, 28(4):213–225, 2014.

[5] Christin Christin, Huub CJ Hoefsloot, Age K Smilde, Berend Hoekman, Frank Suits, Rainer Bischoff, and Peter Horvatovich. A critical assessment of feature selection methods for biomarker discovery in clinical proteomics. *Molecular & Cellular Proteomics*, 12(1):263–276, 2013.

[6] Hyonho Chun and Sündüz Keleş. Sparse partial least squares regression for simultaneous dimension reduction and variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(1):3–25, 2010.

[7] Dongjun Chung and Sunduz Keles. Sparse partial least squares classification for high dimensional data. *Statistical applications in genetics and molecular biology*, 9(1), 2010.

[8] Lennart Eriksson, Henrik Antti, Johan Gottfries, Elaine Holmes, Erik Johansson, Fredrik Lindgren, Ingrid Long, Torbjörn Lundstedt, Johan Trygg, and Svante Wold. Using chemometrics for navigating in the large data sets of genomics, proteomics, and metabonomics (gpm). *Analytical and bioanalytical chemistry*, 380(3):419–429, 2004.

[9] Johan Gottfries, Kaj Blennow, Anders Wallin, and CG Gottfries. Diagnosis of dementias using partial least squares discriminant analysis. *Dementia and Geriatric Cognitive Disorders*, 6(2):83–88, 1995.

[10] Kim-Anh Lê Cao, Simon Boitard, and Philippe Besse. Sparse pls discriminant analysis: biologically relevant feature selection and graphical displays for multiclass problems. *BMC bioinformatics*, 12(1):253, 2011.

[11] Kim-Anh Le Cao, Florian Rohart, Ignacio Gonzalez, Sebastien Dejean, Benoit Gautier, and Francois Bartolo. mixOmics: Omics data integration project. *R package, version*, 2017.

[12] Danh V Nguyen and David M Rocke. Classification of acute leukemia based on dna microarray gene expressions using partial least squares. *Methods of Microarray Data Analysis. Kluwer, Dordrecht*, pages 109–124, 2002.

[13] Danh V Nguyen and David M Rocke. Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics*, 18(1):39–50, 2002.

[14] Florian Rohart, Benoit Gautier, Amrit Singh, and Kim-Anh Le Cao. mixomics: an R package for omics feature selection and multiple data integration. *bioRxiv*, page 108597, 2017.

[15] Lars Ståhle and Svante Wold. Partial least squares analysis with cross-validation for the two-class problem: A monte carlo study. *Journal of chemometrics*, 1(3):185–196, 1987.

[16] Yongxi Tan, Leming Shi, Weida Tong, GT Gene Hwang, and Charles Wang. Multi-class tumor classification by discriminant partial least squares using microarray gene expression data and assessment of classification models. *Computational Biology and Chemistry*, 28(3):235–243, 2004.

[17] Johan A Westerhuis, Huub CJ Hoefsloot, Suzanne Smit, Daniel J Vis, Age K Smilde, Ewoud JJ van Velzen, John PM van Duijnhoven, and Ferdi A van Dorsten. Assessment of PLSDA cross validation. *Metabolomics*, 4(1):81–89, 2008.

[18] Bradley Worley, Steven Halouska, and Robert Powers. Utilities for quantifying separation in PCA/PLS-DA scores plots. *Analytical biochemistry*, 433(2):102–104, 2013.

[19] Bradley Worley and Robert Powers. Multivariate analysis in metabolomics. *Current Metabolomics*, 1(1):92–107, 2013.