

# Big Supervised Manifold Learning by Linear Optimal Low-Rank (LOL)

Joshua T. Vogelstein, Mauro Maggioni

## Abstract

In the 21st century high-dimensional observations abound. In classical statistics, it is well known that as the dimensionality of a dataset increase, the number of samples required to estimate properties thereof increases even faster. A number of techniques have been developed to stave off this “curse of dimensionality”, including dimensionality reduction and regularization. Mitigating this curse is especially demanding in supervised learning problems, such as classification. The challenge in finding a low-dimensional discriminant boundary in the face of high-dimensional observations is that direct searches are computationally infeasible. Therefore, previous art have taken one of two approaches. First, one can ignore the discriminant problem, and try to reduce the dimensionality of the data—for example, using principal components analysis or manifold learning techniques—in hopes that one has not discarded the discriminant information. Second, one directly solve the problem after making a number of overly restrictive assumptions of the data, such as independence and sparsity. Here, we demonstrate via a simple geometric argument that one can algebraically compute a low-dimensional embedding that is approximately optimal with high probability in many settings. The intuition is that we can project the data both in the directions of maximal variance (ignoring the classification task) and the linear direction of maximal discrimination (ignoring the variance). The result is an extremely computationally efficient supervised manifold learning method that we call “Linear Optimal Low-Rank” (LOL). LOL outperforms state-of-the-art algorithms in a wide variety of settings, including both those for which theoretical arguments predict it should, and in much more general settings for which we are unable as yet to obtain solid theoretical footing. This includes both simulated settings, where our intuition enables us to characterize a number of simple variants designed for more complex settings, as well as several benchmark problems. In particular, LOL outperforms several reference algorithms in terms of both efficiency and accuracy on said benchmarks. We therefore believe that LOL could be useful in myriad applications, and many extensions are readily available.

XXX no mention really of big sample sizes anywhere

XXX no mention of supervised manifold learning.

Supervised learning (SL)—the art and science of estimating statistical relationships using labeled training data—is a crucial tool in scientific discovery. SL has been enabled a wide variety of basic and applied findings, ranging from discovering biomarkers in omics data [1], to object recognition from images [2], and includes celebrated methods such as support vector machines, random forests, and deep networks [3]. A special case of SL is classification; a classifier predicts the ‘class’ of a novel observation via partitioning the space of observations into different classes. (for example, predicting male vs. female from MRI scans). In this big data age, the ambient (or observed) dimensionality of the observations is quickly ballooning, and with it, the ambient dimensionality of the discriminant boundary. While historical data may have consisted of only a few dimensions (e.g., height and weight), modern scientific datasets often consist of hundreds, thousands, or even millions of dimensions (e.g. genetics, neuroscience, omics). Regardless of the dimensionality, when scientists or analysts obtain new datasets, they must decide which of the myriad available boundary estimation methods (estimators) to use. When choosing an approach, there are many factors to consider across different levels of analysis, including both statistical and computational considerations.

The statistical goal of such problems is to estimate a discriminant boundary that optimally partitions the space. To do so, one must make a series of decisions to minimize the expected error. First, one can choose which assumptions to make about the *data generating mechanism*. These properties imply geometric constraints on the set of optimal discriminant boundaries. For example, if one assumes that the data are sampled from a Gaussian, and the two classes differ only in their means, then one can prove that the optimal discriminant boundary is linear, but we do not know the (potentially multidimensional) slope and intercept. Second, given such assumptions, one can then choose the *boundary search space* over which

one will search for the optimal boundary. For example, if we assume the optimal boundary is linear, we can search only for linear boundaries, and exclude multi-linear and quadratic boundaries. The minimum distance between the boundaries we search over and the optimal boundary is called the **model bias**, which we want to minimize by choosing a search space as close as possible to the optimal boundary space. Third, given our constrained search space, we can choose an *algorithm* to find an optimal boundary, such as Fishers Linear Discriminant, Linear Support Vector Machines, or Logistic Regression. Each such algorithm has different convergence properties as sample size increases. Importantly, for different data generating mechanisms, different algorithms may converge on different solutions. The additional distance between the expected estimated boundary and the optimal boundary is called the **estimator bias**, which we also want to minimize by choosing an algorithm that will select a boundary as close as possible to the truth. Different algorithms will also differ in the **variance** of their estimated boundaries, which is a function both of the size of the search space, and the idiosyncrasies of the algorithm. The two bias terms and the variance together determine expected square error, so, one also desires to minimize the variance of the estimated boundary.

In addition to the above statistical considerations, there are a number of computational considerations to consider which also depend on a series of choices. In this big data age, the practical importance of these considerations is quickly gaining on the importance of the various statistical considerations. First, the choice of *implementations* for a particular algorithm can dramatically change the computational properties of an algorithm. In particular, one desires an implementation that is **numerically stable**, regardless of the number of samples and dimensions. Second, one desires an implementation that minimizes computational complexity, both in terms of worst case **space** and **time**. Second, the *platform* upon which the implementation runs will determine a number of other factors, including both software dependencies and hardware choice. In particular the **scalability** of the implementation, that is, how large of a problem can be solved on a particular hardware configuration.

In addition to the above theoretical considerations, there are also a number of pragmatic ones. Using theory we can prove properties of certain choices with respect to various assumptions, and use those theoretical properties to compare different methodologies. But the theory is limited to certain relatively simple sets of assumptions, assumptions for which many real data sets and algorithms do not satisfy. Even when the assumptions are met, theory often merely provides bounds, without concrete numbers in any particular situations. Numerical experiments can therefore buttress the theoretical considerations. In particular, **simulated performance** on suites of simulated data, where we know the truth and can therefore compare the performance of different approaches to the optimal can provide insight into when our theories are working as expected, including in settings that extend beyond the theoretical constraints. Moreover, **real data performance**, both accuracy and time, provide insight into different methodologies when we do not know much about the true data generating mechanism.

Finally, in addition to the above mentioned quantitative criteria, there are also many qualitative criteria that merit consideration. First, the **intuitiveness** of the method can provide a warm fuzzy feeling because the ways in which a particular method can go wrong are better understood. Second, **hyper-parameter searches** can add significant computational overhead, as searching over many different hyper-parameters can exponentially increase the amount of time one spends estimating the optimal boundary. Third, whether the implementation is **free and open source software** (FOSS) impacts its availability, especially for non-profit organizations, developing world institutions, and citizen scientists. Fourth, the **ease of use**, which is largely determined by platform considerations. In particular, if running the method requires access to a supercomputer, or a Hadoop cluster, or non-free software (like MATLAB), or otherwise specialized software and hardware configurations, it may require more activation energy and costs.

It is quite a challenge to optimize along all of the above dimensions for any particular kind of exploitation task. Here, we first focus on the simplest big data supervised learning task: two-class classification, where each class is sampled from a Gaussian distribution and they only differ in their means, and the dimensionality  $p$  is greater than the sample size  $n$ , that is,  $p > n$ . Fishers Linear Discriminant (FLD), while asymptotically optimal (zero bias and variance as  $n$  goes to infinity), is numerically unstable when  $n < p$ , and therefore will not even run. A very natural fix to this is to first “pre-process” the data via a truncated Principal Components Analysis (tPCA), and then apply FLD (an approach called Fisherfaces []). While first running tPCA greatly reduces the variance relative to directly running FLD, and therefore makes the algorithm numerically

stable, it adds estimator bias because the dimension reduction performed by PCA does not consider the class labels, only the predictor variables, and the discriminant boundary is determined by the relationship between the class labels and predictor variables. Other “manifold learning techniques”, such as isomap [4], local linear embedding [5], Laplacian eigenmaps [6], Diffusion maps [7] and others are similarly “unsupervised”, and therefore, do not explicitly search for a discriminant boundary, and therefore may add significant bias. Adding supervision to these techniques so far has been difficult, methods proposed thus far embed the data in an unsupervised fashion first, again, potentially adding bias [8], and require substantial computations. Sparse methods (e.g., []) can use both the class labels and predictor variables, but they come with substantial costs. In particular, if certain very restrictive assumptions of the data are not satisfied, such as the restrictive isometry property, sparse methods add both significant estimator bias and variance. Moreover, they require much greater computational time than Fisherfaces.

Thus, a gap remains in the literature for a method that, under relatively general assumptions, statistically achieves both small bias and variance, and computationally numerically stable and requires minimal space and time, and is highly scalable. Importantly, the method should perform well both on simulated and real data experiments. And ideally, the method is intuitive, FOSS, and easy to use.

Here, we describe a novel methodology, called Linear Optimal Low-Rank (LOL), that satisfies all of the above desiderata. We developed LOL by considering the above described simple scenario, and realizing that the optimal dimension reduction for such a setting would have to combine both the means and the covariances, because together those two parameters determine the optimal decision boundary. While other methods have previously had a similar insight (in particular the “First Two Moment” (F2M) methods []), F2M methods only operate when  $n > p$ , and we are interested in the setting where  $p > n$ , and possibly  $p \gg n$ . Therefore, we decided to combine the means and covariances differently, essentially by first projecting the data onto the means, and then effectively computing tPCA on the “residuals”. Running FLD after projecting the data onto the subspace spanned jointly by the means and the top principal components comprises the LOL method.

We demonstrate LOL outperforms—or at least performs no worse than—Fisherfaces implementations along every statistical, computational, and implementational consideration mentioned above. This includes proving the improvement over Fisherfaces along both statistical considerations (bias and variance) for the simple Gaussian setting mentioned above, as well as the computational considerations: stability, time, space, and scalability under more general settings. Moreover, numerical experiments quantitatively demonstrate the improvement of LOL over both Fisherfaces and sparse methods on a wide range of simulated experiments that satisfy our theoretical assumptions. By virtue of LOL being built upon geometric intuition, it is easy to extend it along various dimensions. Additional numerical experiments demonstrate that practice matches our intuition, in particular, generalizations of LOL can easily be constructed to match different geometric assumptions of the data. Indeed, even when data are sampled from distributions outside our mean and covariance intuition, we see that LOL achieves smaller error than its competitors. In fact, we can extend our intuition outside of classification problems to other problems, including wide regression and hypothesis testing. A trivial modification of LOL leads to regression and testing procedures that also outperform the natural competitors. Computationally, LOL is always numerically stable, and requires no more computational space, and smaller computational time, than both Fisherfaces and sparse methods. Moreover, we provide a highly scalable implementation that efficiently runs on terabyte datasets with hundreds of thousands or millions of features. Finally, we test LOL against a set of standard methods on four different benchmark real datasets. For any of the four different settings, there was not a single method that outperformed LOL on both accuracy and time.

In addition to the above quantitative properties, LOL is intuitive, only has a single hyper-parameter which is exceedingly easy to tune, and we provide free and open source implementations, both in MATLAB and R, as well as pre-configured environments that will enable anybody to run LOL on commodity machines, including laptops, workstations, and cloud instances. LOL is not likely to necessarily be the best algorithm on any given new high-dimensional challenge problem. Rather, the arguments and methodology developed herein provide a framework for developing big supervised manifold learning algorithms to tackle the data science problems of the next century.

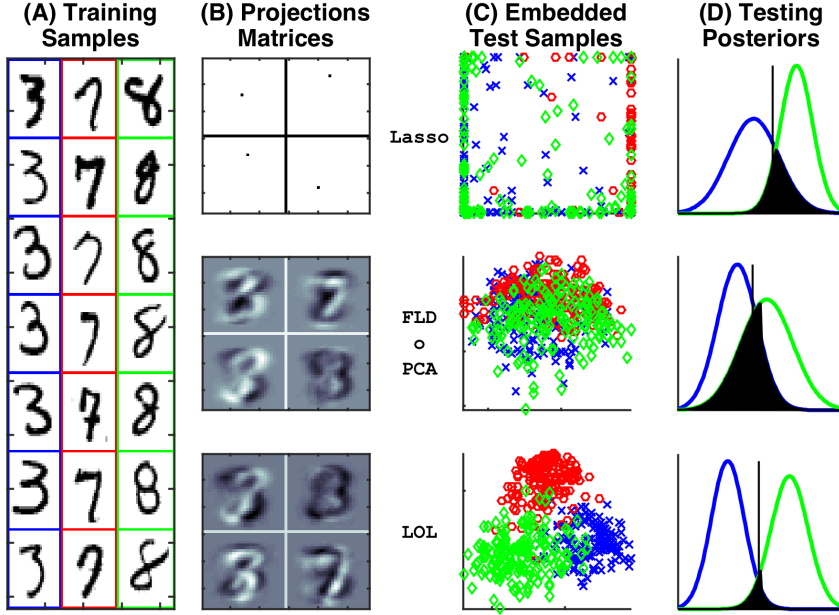


Figure 1: Illustrating three different classifiers—`LASSO` (top), `FLDPCA` (middle), and `LOL` (bottom)—for embedding images of the digits 3, 7, and 8 (from MNIST), each of which is  $28 \times 28 = 784$  dimensional. **(A)** Exemplars, boundary colors are only for visualization purposes. **(B)** The first four projection matrices learned by the three different approaches on 300 training samples. Note that `LASSO` is sparse and supervised, `PCA` is dense and unsupervised, and `LOL` is dense and supervised. **(C)** Embedding 500 test samples into the top 2 dimensions using each approach. Digits color coded as in (A). **(D)** The estimated posterior distribution of test samples after 5-dimensional projection learned via each method. We show only 3 vs. 8 for simplicity. The vertical line shows the classification threshold. The filled area is the estimated error rate: the goal of any classification algorithm is to minimize that area. Clearly, `LOL` exhibits the best separation after embedding, which results in the best classification performance.

## Results

### An Illustrative Real Data Example of Supervised Linear Manifold Learning

Pseudocode of any method that embeds data as part of classification proceeds as schematized in Figure 1: (A) obtain/select  $n$  training samples of the data, (B) learn a low dimensional projection, (C) project  $m$  testing samples onto the lower dimensional space, (D) classify the embedded testing samples using some classifier. We consider three different linear dimensionality reduction methods—`LASSO`, `PCA`, and `LOL`—each of which we compose with a classifier to form high-dimensional classifiers.<sup>1</sup>

To demonstrate the utility of `LOL`, we first consider one of the most popular benchmark datasets ever, the MNIST dataset [10]. This dataset consists of many thousands of examples of images of the digits 0 through 9. Each such image is represented by a  $28 \times 28$  matrix, which means that the observed (or ambient) dimensionality of the data is  $p = 784$ . Because we are motivated by the  $n \ll p$  scenario, we subsample the data to select  $n = 300$  examples of the numbers 3, 7, and 8. We then apply all three approaches to this subsample of the MNIST dataset, learning a projection, and embedding  $m = 500$  testing samples, and classifying the resulting embedded data.

`LASSO`, by virtue of being a sparse method, finds the pixels that most discriminate the 3 classes. The resulting embeddings mostly live along the boundaries, because these images are close to binary, and

<sup>1</sup>Although `LASSO` is not a 2-step method (where embedding is learned first, and then a classifier is applied), adaptive lasso [9] and its variants improve on lasso's theoretical and empirical properties, so we consider such an approach here.

therefore, images either have or do not have a particular pixel. Indeed, although the images themselves are nearly sparse (over 80% of the pixels in the dataset have intensity  $\leq 0.05$ ), a low-dimensional discriminant boundary does not seem to be sparse.  $\text{PCA}$ , on the other hand, finds the linear combinations of training samples that maximize the variance. This unsupervised linear manifold learning method results in projection matrices that indeed look like linear combinations of the three different digits. The goal here, however, is separating classes, not maximizing variability. The resulting embeddings are not particularly well separated, suggesting the the directions of discriminability are not the same as the directions of maximum variance.  $\text{LOL}$  is our newly proposed supervised linear manifold learning method (see below for details). The projection matrices it learns look qualitatively much like those of  $\text{PCA}$ . This is not surprising, as both are linear combinations of the training examples. The resulting embeddings however, look quite different. The three different classes are very clearly separated by even the first two dimensions. The result of these embeddings yields classifiers whose performance is obvious from looking at the embeddings:  $\text{LOL}$  achieves significantly smaller error than the other two approaches. This numerical experiment justifies the use of supervised linear manifold learning, we next investigate the performance of these methods in simpler simulated examples, to better illustrate when we can expect  $\text{LOL}$  to outperform other methods, and perhaps more importantly, when we expect this “vanilla” variant of  $\text{LOL}$  to fail.

## Linear Gaussian Intuition

The above real data example suggests the geometric intuition for when  $\text{LOL}$  outperforms its sparse and unsupervised counterparts. To further investigate, both theoretically and numerically, we consider the simplest setting that illustrates the relevant geometry. In particular, we consider a two-class classification problem, where both classes are distributed according to a multivariate normal distribution, the class priors are equal, and the joint distribution is centered, so that the only difference between the classes is their means (we call this the Linear Discriminant Analysis ( $\text{LDA}$ ) model; see Methods for details). To motivate  $\text{LOL}$ , and the following simulations, let’s consider what the optimal projection would be in this scenario. The optimal low-dimensional projection is analytically available as the dot product of the difference of means and the inverse covariance matrix,  $\mathbf{A}_* = \delta^T \Sigma^{-1} [\mathbf{1}]$  (see Methods for details).  $\text{PCA}$ , the dominant unsupervised manifold learning method, utilizes only the covariance structure of the data, and ignores the difference between the means. In particular,  $\text{PCA}$  would project the data on the top  $d$  eigenvectors of the covariance matrix.

**The key insight of our work is the following: we can combine the difference of the means and the covariance matrix in a simple fashion, rather than just the covariance matrix, to find a low dimensional projection.** Naïvely, this should typically improve performance, because in this stylized scenario, both are important. The  $\text{F2M}$  literature has a similar insight, but a different construction that requires the dimensionality to be smaller than the sample size. Formally, we implement our idea by simply concatenating the difference of the means with the top  $d$  eigenvectors of the covariance. This is equivalent to first projecting onto the difference of the means vector, and then projecting the residuals onto the first  $d$  principle components. Thus, it requires almost no additional computational time or complexity over that of  $\text{PCA}$ , rather, merely estimates the difference of the means. In this sense,  $\text{LOL}$  can be thought of as a very simple “supervised  $\text{PCA}$ ”.

Figure 2 shows three different examples of data sampled from the  $\text{LDA}$  model to geometrically illustrate this intuition. In each, we sample  $n = 100$  training samples in  $p = 1000$  dimensional space, so  $n \ll p$ . Figure 2(A) shows an example we call “stacked cigars”. In this example and the next, the covariance matrix is diagonal, so all ambient dimensions are independent of one another. Moreover, the difference between the means and direction of maximum variance are both large along the same dimensions (they are highly correlated with one another). This is an idealized setting for  $\text{PCA}$ , because  $\text{PCA}$  finds the direction of maximal variance, which happens to correspond to the direction of maximal separation. However, in the face of high-dimensional data,  $\text{PCA}$  does not weight the discriminant directions sufficiently, and therefore performs only moderately well.<sup>2</sup> Because all dimensions are independent, this is a good scenario for sparse methods. Indeed,  $\text{ROAD}$ , a sparse classifier designed for precisely this scenario, does an excellent job finding the most useful ambient dimensions.  $\text{LOL}$  does the best of all three approaches, by using both the difference

<sup>2</sup>When having to estimate the eigenvector from the data,  $\text{PCA}$  performs even worse. This is because when  $n \ll p$ ,  $\text{PCA}$  is an inconsistent estimator with large variance [13, 14]

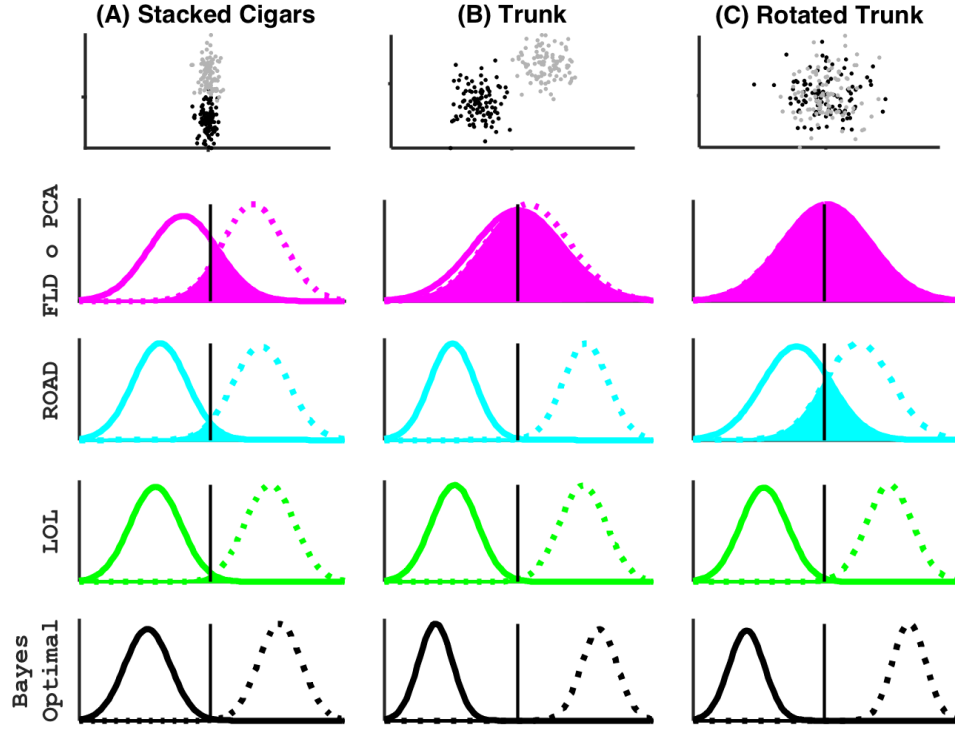


Figure 2: **LOL** achieves near optimal performance for a wide variety of distributions. Each point is sampled from a multivariate Gaussian; the three columns correspond to different simulation parameters (see Methods for details). In each of 3 simulations, we sample  $n = 100$  points in  $p = 1000$  dimensions. And for each approach, we embed into the top 20 dimensions. Note that we use the sample estimates, rather than the true population values of the parameters. In this setting, the results are similar. **(A)** The mean difference vector is aligned with the direction of maximal variance, making it ideal for both **PCA** to discover the discriminant direction and a sparse solution. **(B)** The mean difference vector is orthogonal to the direction of maximal variance, making **PCA** fail, but sparse methods can still recover the correct dimensions. **(C)** Same as (B), but the data are rotated. **Row 1:** A scatter plot of the first two dimensions of the sampled points, with class 0 and 1 as black and gray dots, respectively. **Row 2** **FLD o PCA**. **Row 3** **ROAD**, a sparse method designed specifically for this model [12]. **Row 4** **LOL**, our newly proposed method. **Row 5** the Bayes optimal classifier, which is what all classifiers strive to achieve. Note that **LOL** is closest to Bayes optimal in all three settings.



of the means and the covariance.

Figure 2(B) shows an example which is a worst case scenario for using PCA to find the optimal projection for classification. In particular, the variance is getting larger for subsequent dimensions,  $\sigma_1 < \sigma_2 < \dots < \sigma_p$ , while the magnitudes of the difference between the means are decreasing with dimension,  $\delta_1 > \delta_2 < \dots > \delta_p$ . Thus, for any truncation level, PCA finds exactly the *wrong* directions. ROAD is not hampered by this problem, it is also able to find the directions of maximal discrimination, rather than those of maximal variance. Again, LOL, by using both parameters, does extremely well.

Figure 2(C) is exactly the same as (B), except the data have been randomly rotated in all 1000 dimensions. This means that none of the original coordinates have much information, rather, linear combinations of them do. This is evidenced by observing the scatter plot, which shows that two dimensions clearly fail to disambiguate the two classes. PCA, being rotationally invariant, fails in this scenario as it did in (B). Now, there is no small number of ambient dimensions that separate the data well, so ROAD also fails. LOL is unperturbed by this rotation; in particular, it is able to “unrotate” the data, to find dimensions that optimally separate the two classes.

## Theoretical Confirmation

The above numerical experiments provide the intuition to guide our theoretical developments.

**Theorem 1.** *Under the LDA model, LOL is better than PCA.*

In words, it is better to incorporate the mean difference vector into the projection matrix. The degree of improvement is a function of the embedding dimension  $d$ , the ambient dimensionality  $p$ , and the parameters (see Methods for details and proof).

## How many dimensions to keep?

In the above numerical and theoretical investigations, we fixed  $d$ , the number of dimensions to embed into. Much unsupervised manifold learning theory typically focuses on finding the “true” intrinsic dimensionality of the data. The analogous question for supervised manifold learning would be to find the true intrinsic dimensionality of the discriminant boundary. However, in real data problems, typically, there is no perfect low dimensional representation because of noise.

Thus, in all the following simulations, the true ambient dimensionality of the data is equal to the dimensionality of the optimal discriminant boundary (given infinite data). In other words, there does not exist a discriminant space that is lower dimensional than the ambient space, so we cannot find the “intrinsic dimension” of the data or the discriminant boundary. Rather, we face a trade-off: keeping more dimensions reduces bias, but increases variance. The optimal bias/variance trade-off depends on the distribution of the data, as well as the sample size [15].

Consider again the rotated trunk example as well as a “Toeplitz” example, as depicted in Figures 3(A) and (B), respectively. In both cases, the data are sampled from the LDA model, and in both cases, the optimal dimensionality under finite samples depends on the particular approach, but is never the true dimensionality. Moreover, LOL dominates the other approaches, regardless of the number of dimensions used. Figure 3(C) shows a sparse example with “fat tails” to mirror real data settings better. The qualitative results are consistent with those of (A) and (B), even though the setting is no longer exactly the setting under which we have theoretical confirmation.

## Multiple Classes

LOL can trivially be extended to  $> 2$  class situations. Naïvely it may seem like we would need to keep all pairwise differences between means. However, given  $k$  classes, the set of all  $k^2$  differences is only rank  $k-1$ . In other words, we can equivalently find the class which has the maximum number of samples (breaking ties randomly), and subtract its mean from all other class means. Figure 3(D) shows a 3-class generalization of (A). While LOL uses the additional class naturally, many previously proposed high-dimensional FLD variants,

such as  $\text{ROAD}$ , natively only work for 2-classes.

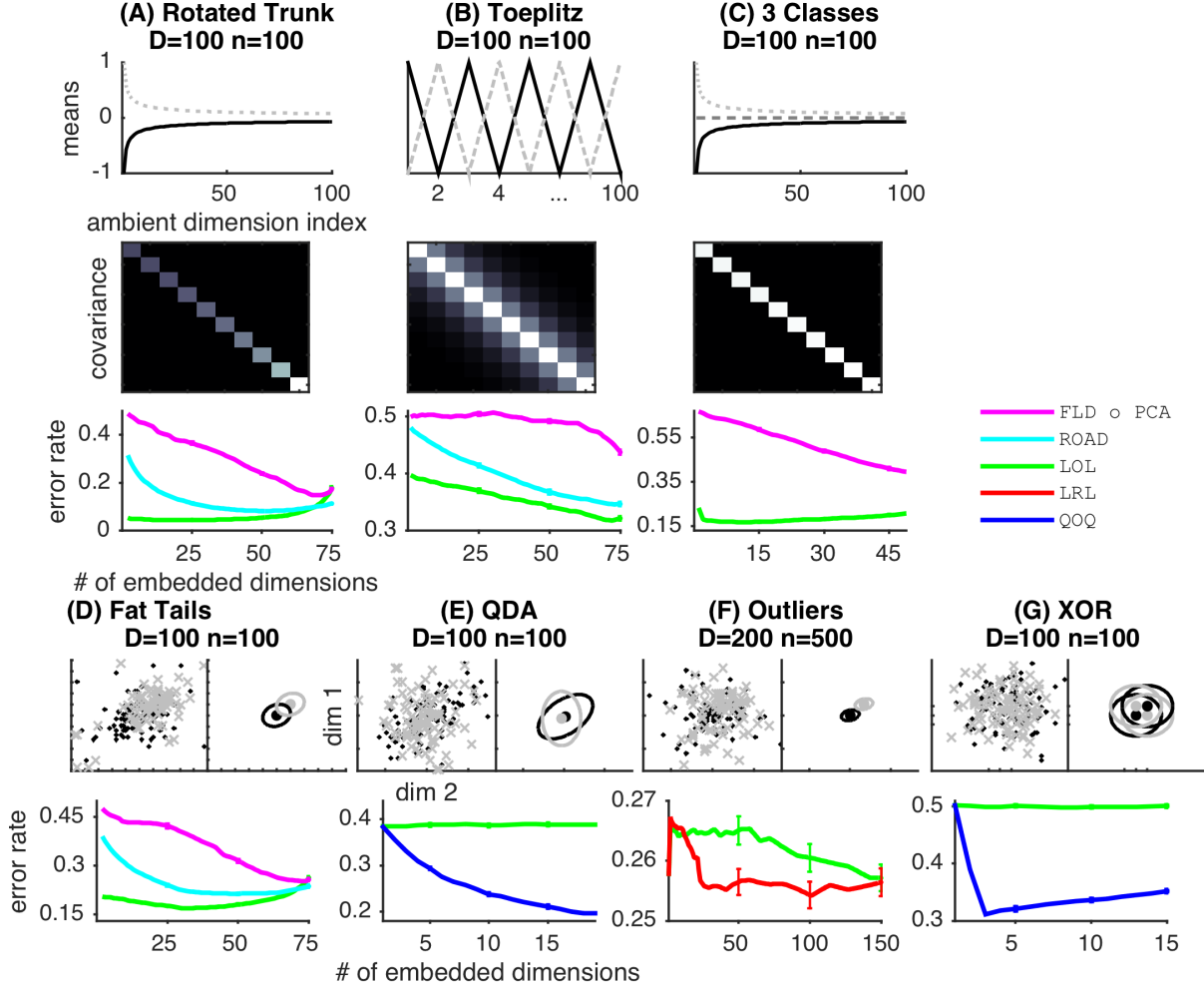


Figure 3: Seven simulations demonstrating that even when the true discriminant boundary is high-dimensional,  $\text{LOL}$  can find a low-dimensional projection that wins the bias-variance trade-off against competing methods. For the first three, the top panels depict the means (top), the shared covariance matrix (middle), and these examples satisfy the assumptions underlying Theorem 1. For the next four, the top panels depict a 2D scatter plot (left), mean and level set of one standard deviation of covariance matrix (right), and all four are examples for which our theory does not extend. For all seven simulations, the bottom panel shows misclassification rate as a function of the number of embedded dimensions, for several different classifiers. The simulations settings are as follows: **(A)** Rotated Trunk: same as Figure 2(C). **(B)** Toeplitz: another setting where mean difference is not well correlated with any eigenvector, and no ambient coordinate is particularly useful on its own. **(C)** 3 Classes:  $\text{LOL}$  naturally adapts to multiple classes. **(D)** Fat Tails: a common phenomenon in real data that is more general than our theory supports. **(E)** QDA:  $\text{QOQ}$ , a variant of  $\text{LOL}$  when each class has a unique covariance, outperforms  $\text{LOL}$ , as expected. **(F)** Outliers: adding high-dimensional outliers degrades performance of standard eigensolvers, but those can easily be replaced in  $\text{LOL}$  for a robust variants (called  $\text{LRL}$ ). **(G)** XOR: a high-dimensional stochastic generalization of XOR, demonstrating the  $\text{LOL}$  and  $\text{QOQ}$  work even in scenarios that are quite distinct from the original motivating problems. In all 7 cases,  $\text{LOL}$ , or the appropriate generalization thereof, outperforms unsupervised, sparse, or other methods. Moreover, the optimal embedding dimension is never the true discriminant dimension, but rather, a smaller number jointly determined by parameter settings and sample size.



## Generalizations of $\text{LOL}$

The simple geometric intuition which led to the development of  $\text{LOL}$  suggests that we can easily generalize  $\text{LOL}$  to be more appropriate for more complicated settings. We consider three additional scenarios:

**QDA** Sometimes, it makes more sense to model each class as having a unique covariance matrix, rather than a shared covariance matrix. Assuming everything is Gaussian, the optimal classifier in this scenario is called Quadratic Discriminant Analysis (QDA) [3]. Intuitively then, we can modify  $\text{LOL}$  to compute the eigenvectors separately for each class, and concatenate them (sorting them according to their singular values). Moreover, rather than classifying the projected data with  $\text{LDA}$ , we can then classify the projected data with QDA. Indeed, simulating data according to such a model (Figure 3(E)),  $\text{LOL}$  performs slightly better than chance, regardless of the number of dimensions we use to project, whereas QOQ (which denotes we estimate eigenvectors separately and then use QDA on the projected data) performs significantly better regardless of how many dimensions it keeps.

**Outliers** Outliers persist in many real data sets. Finding outliers, especially in high-dimensional data, is both tedious and difficult. Therefore, it is often advantageous to have estimators that are robust to certain kinds of outliers [16–18].  $\text{PCA}$  and eigenvector computation are particularly sensitive to outliers [19]. Because  $\text{LOL}$  is so simple and modular, we can replace typical eigenvector computation with a robust variant thereof, such as the geometric median subspace embedding [20]. Figure 3(F) shows an example where we generated  $n/2$  training samples according to the simple  $\text{LDA}$  model, but then added another  $n/2$  training samples from a noise model.  $\text{LRL}$  (our robust variant of  $\text{LOL}$  that simply replaces the fragile eigenvector computation with a robust version), performs better than  $\text{LOL}$  regardless of the number of dimensions we keep.

**XOR** XOR is perhaps the simplest nonlinear problem, the problem that led to the demise of the perceptron, prior to its resurgence after the development of multi-layer perceptrons [21]. Thus, in our opinion, it is warranted to check whether any new classification method can perform well in this scenario. The classical (two-dimensional) XOR problem is quite simple: the output of a classifier is zero if both inputs are the same (00 or 11), and the output is one if the inputs differ (01 or 10). Figure 3(G) shows a high dimensional and stochastic variant of XOR. This simulation was designed such that standard classifiers, such as support vector machines and random forests, achieve chance levels (not shown).  $\text{LOL}$ , performs moderately better than chance, and QOQ performs significantly better than chance, regardless of the chosen dimensionality. This demonstrates that our classifiers developed herein, though quite simple and intuition, can perform well even in settings where the data are badly modeled by our underlying assumptions. This mirrors previous findings where the so-called “idiots’s Bayes” classifier outperforms more sophisticated classifiers [11]. In fact, we think of our work as finding intermediate points between idiot’s Bayes (or naïve Bayes) and  $\text{FLD}$ , by enabling degrees of regularization by changing the dimensionality used.

## Computational Efficiency

In many applications, the main quantifiable consideration in whether to use a particular method, other than accuracy, is computational efficiency. Because implementing  $\text{LOL}$  requires only highly optimized linear algebraic routines—including computing moments and singular value decomposition—rather than the costly iterative programming techniques currently required for sparse or dictionary learning type problems. To quantify the computational efficiency of  $\text{LOL}$  and its variants, Figure 4 shows the wall time it takes to run each method on the stacked cigars problem, varying the ambient dimensionality, embedded dimensionality, and sample size. Note that for completeness, we include two additional variants of  $\text{LOL}$ :  $\text{LAL}$  and  $\text{LFL}$ .  $\text{LFL}$  (short for Linear Fast Low-rank) replaces the standard  $\text{SVD}$  algorithm with a randomized variant, which can be much faster in certain situations [22].  $\text{LAL}$  (short for Linear Approximate Low-rank) goes even one step further, replacing  $\text{SVD}$  with random projections [23]. This variant of  $\text{LOL}$  is the fastest, its runtime is least sensitive to  $(p, d, n)$ , and its accuracy is often commensurate (or better) than other variants of  $\text{LOL}$ . The runtime of all the variants of  $\text{LOL}$  are quite similar to  $\text{FLD} \circ \text{PCA}$ . Given, given  $\text{LOL}$ ’s improved accuracy, and nearly identical simplicity, it seems there is very little reason to not use  $\text{LOL}$  instead of  $\text{FLD} \circ \text{PCA}$ .

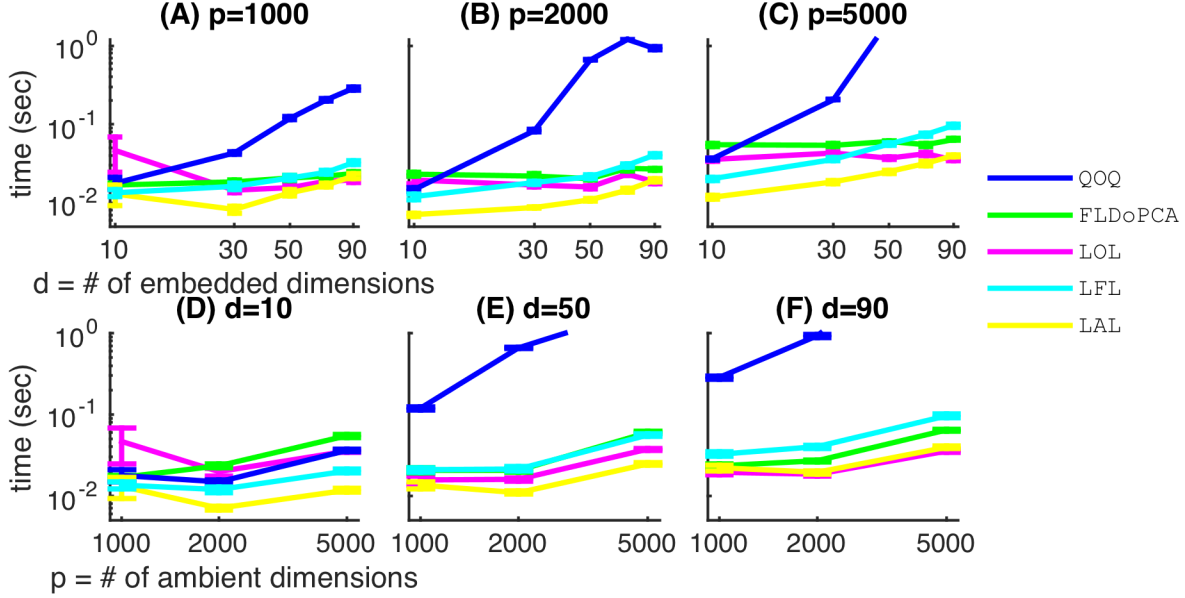


Figure 4: Computational efficiency of various low-dimensional projection methods. In all cases,  $n = 100$ , and we used the “stacked cigars” simulation parameters. We compare  $\text{PCA}$  with the projection steps from  $\text{LOL}$ ,  $\text{QOQ}$ ,  $\text{LRL}$ ,  $\text{LFL}$ , and  $\text{LAL}$ , for different values of  $(p, d)$ . The addition of the mean difference vector is essentially negligible. Moreover, for small  $d$ , the  $\text{LFL}$  is advantageous.  $\text{LAL}$  is always fastest, and its performance is often comparable to other methods (not shown).

## Benchmark Real Data Applications

To more comprehensively understand the relative advantages and disadvantages of  $\text{LOL}$  with respect to other high-dimensional classification approaches, in addition to evaluating its performance in theory, and in a variety of numerical simulations, it is important to evaluate it also on benchmark datasets. For these purposes, we have selected four commonly used high-dimensional datasets (see Methods for details). For each, we compare  $\text{LOL}$  to (i) support vector machines (SVM), (ii)  $\text{ROAD}$ , (iii) lasso, (iv) and random forest (RF). Because in practice all these approaches have “hyperparameters” to tune, we consider several possible values for SVM, lasso, and  $\text{LOL}$  (but not RF, as its runtime was too high). Figure 5 shows the results for all four datasets.

Qualitatively, the results are similar across datasets:  $\text{LOL}$  achieves high accuracy and computational efficiency as compared to the other methodologies. Considering Figure 5(A) and (B), two popular sparse settings, we find that  $\text{LOL}$  can find very low dimensional projections with very good accuracy. For the prostate data, with a sufficiently non-sparse solution for  $\text{ROAD}$ , it slightly outperforms  $\text{LOL}$ , but at substantial computational cost, in particular,  $\text{ROAD}$  takes about 100 times longer to run on this dataset. Figure 5(C) and (D) are 10-class problems, so  $\text{ROAD}$  is no longer possible. Here, SVM can again slightly outperform  $\text{LOL}$ , but again, requiring 100 fold additional computational time. In all cases, the beloved random forest classifier performs subpar.

## Extensions to Other Supervised Learning Problems

The utility of incorporating the mean difference vector into supervised machine learning for wide data extends beyond merely classification. In particular, hypothesis testing can be considered as a special case of classification, with a particular loss function. Therefore we apply the same idea to a hypothesis testing scenario. The multivariate generalization of the t-test, called Hotelling’s Test, suffers from the same problem as does the classification problem; namely, it requires inverting an estimate of the covariance matrix. To

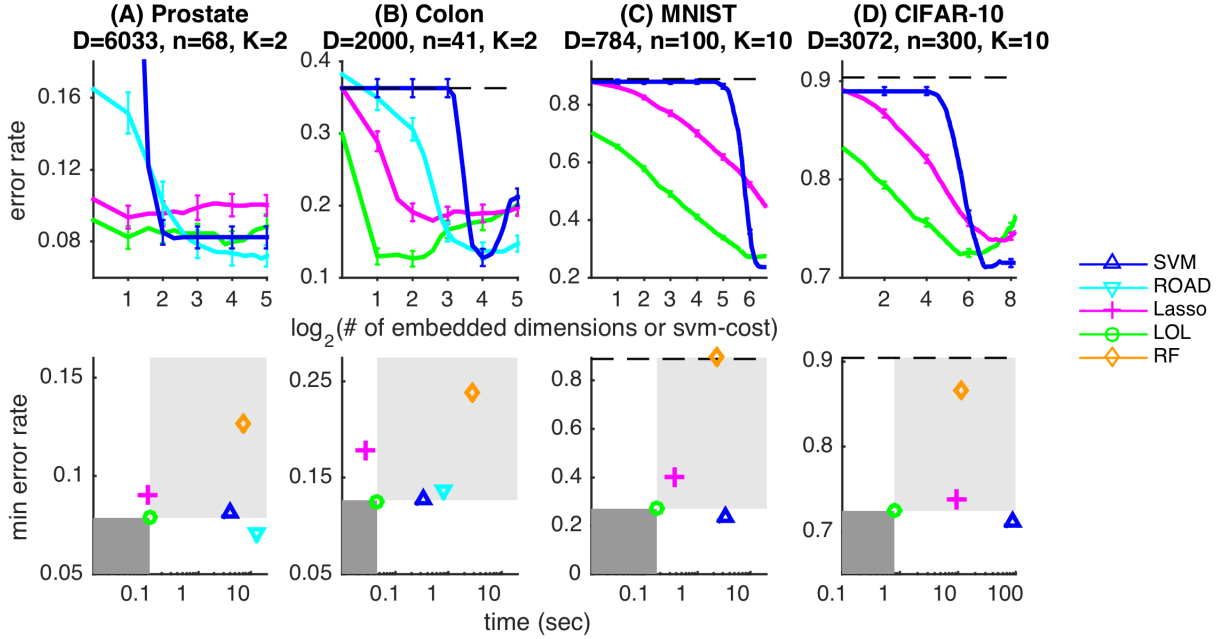


Figure 5: For four standard datasets, we benchmark `LOL` (green circles) versus standard classification methods, including support vector machines (blue up triangles), `ROAD` (cyan down triangles), `LASSO` (magenta pluses), and random forest (orange diamonds). Top panels show error rate as a function of  $\log_2$  number of embedded dimensions (for `LOL`, `ROAD`, and `LASSO`) or cost (for `SVM`). Bottom panels show the minimum error rate (over hyperparameters) achieved by each of the five algorithms versus time. The lower left dark gray (upper right light gray) rectangle is the area in which any algorithm is *better* (worse) than `LOL` in terms of both accuracy and efficiency. **(A)** Prostate: a standard sparse dataset. 1-dimensional `LOL` does very well, although `ROAD` keeps  $2^5$  ambient coordinates to slightly improves performance, at a significant cost of compute time (two orders of magnitude), with minimal additional interpretability. **(B)** Colon: another standard sparse dataset. Here, 2-4 dimensions of `LOL` outperforms all other approaches considered. **(C)** MNIST: 10 image categories here, so `ROAD` is not possible. `LOL` does very well regardless of the number of dimensions kept. `SVM` marginally improves on `LOL` accuracy, at a significant cost in computation (two orders of magnitude). **(D)** CIFAR-10: a higher dimensional and newer 10 category image classification problem. Results are qualitatively similar to (C). Note that, for none of the problems is there an algorithm ever performing better and faster than `LOL`; rather, most algorithms typically perform worse and slower (though some are more accurate and much more computationally expensive).

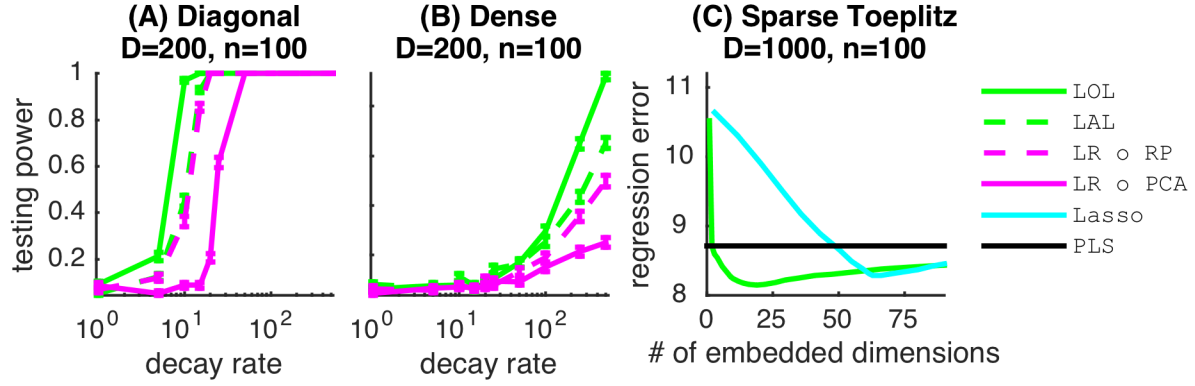


Figure 6: The intuition of including the mean difference vector is equally useful for other supervised manifold learning problems, including testing and regression. (A) and (B) show two different high-dimensional testing settings, as described in Methods. Power is plotted against the decay rate of the spectrum, which approximates the effective number of dimensions.  $\text{LOL}$  composed with Hotelling’s test outperforms the random projections variants described in [24], as well as several other variants. (C) shows a high-dimensional regression settings, as described in Methods.  $\log_{10}$  mean squared error is plotted against the number of embedded dimensions. For regression,  $\text{LOL}$  composed with linear regression outperforms  $\text{LASSO}$  (cyan), as well as partial least squares (PLS; black). In the legend, ‘A’ denote either ‘linear regression’ (in (C)), or ‘Hotelling’ (in (A) and (B)). These three simulation settings therefore demonstrate the generality of this technique.

mitigate this issue in the hypothesis testing scenario, prior art applied similar tricks as they have done in the classification setting. One particularly nice and related example is that of Lopes et al. [24], who addresses this dilemma by using random projections to obtain a low-dimensional representation, following by applying Hotelling’s Test in the lower dimensional subspace. Figure 6(A) and (B) shows the power of their test alongside the power of the same approach, but using the  $\text{LOL}$  projection rather than random projections. The two different simulations include the simulated settings considered in their manuscript (see Methods for details). The results make it clear that the  $\text{LOL}$  test has higher power for essentially all scenarios. Moreover, it is not merely the replacing random projections with  $\text{PCA}$  (solid magenta line), nor simply incorporating the mean difference vector (dashed green line), but rather, it appears that  $\text{LOL}$  for testing uses both modifications to improve performance.

High-dimensional linear regression is another supervised learning method that can utilize this idea. Linear regression, like classification and Hotelling’s Test, requires inverting a singular matrix as well. By projecting the data only a lower dimensional subspace first, followed by linear regression on the low-dimensional data, we can mitigate the curse of high-dimensions. To choose the projection matrix, we partition the data into  $K$  partitions, based on the percentile of the target variable, we obtain a  $K$  class classification problem. Then, we can apply  $\text{LOL}$  to learn the embedding. Figure 6(C) shows an example of this approach, contrasted with lasso and partial least squares, in a sparse simulation setting (see Methods for details).  $\text{LOL}$  is able to find a better low-dimensional projection than lasso, and performs significantly better than PLS, for essentially all choices of number of dimensions to embed into.

## Discussion

We have introduced a very simple, yet new, methodology to improve performance on supervised learning problems with wide data. In particular, we have proposed a supervised manifold learning procedure, the utilizes both the difference of the means, and the covariance matrices. This is in stark contrast to previous approaches, which only utilize the covariance matrices (or kernel variants thereof), or solve a difficult optimization theoretic problem. In addition to demonstrating the accuracy and numerical efficiency of  $\text{LOL}$  on simulated and real classification problems, we also demonstrate how the same idea can also be used for

other kinds of supervised learning problems, including regression and hypothesis testing. Theoretical guarantees suggest that this line of research is promising and can be extended to other more general settings and tasks.

## Related Work

One of the first publications to compose `FLD` with an unsupervised learning method was the celebrated Fisherfaces paper [25]. The authors showed via a sequence of numerical experiments the utility of embedding with `PCA` prior to classifying with `FLD`. We extend this work by adding a supervised component to the initial embedding. Moreover, we provide the geometric intuition for why and when this is advantageous, as well as show numerous examples demonstrating its superiority.

For example, one flavor of strategies, collectively called “manifold learning”, typically assumes that the space that contains the signal is a linear or nonlinear combination of the original space (a rather general assumption) [26]. More recently a large number of nonlinear manifold learning methods have been developed (e.g., [4–7, 27–30]), many of which can be thought of as kernel versions of PCA [31], also called “spectral methods”.

A complimentary flavor of strategies for dealing with wide data is regularization or shrinkage [11, 32, 33], including smoothing [34] and more recently *sparsity* [35]. Sparse methods assume that signal lives in a small subset of the ambient dimensions (e.g., [12, 34, 36–43]). This assumption, when accurate, can lead to substantial improvements in accuracy with relatively moderate increase in computational cost [44]. Moreover, it is comparably easy to incorporate supervised information.

Such manifold learning methods, while exhibiting both strong theoretical [26, 30, 45] and empirical performance, are fully unsupervised. Thus, in classification problems, they discover a low-dimensional representation of the data, ignoring the labels. This can be highly problematic when the discriminant dimensions and the directions of maximal variance in the learned manifold are not aligned (see Figure 1 for an example). Supervised dimensionality reduction techniques, therefore, combine the best of both worlds, explicitly searching for low-dimensional discriminant boundaries. A set of methods from the statistics community is collectively referred to as “sufficient dimensionality reduction” (SIR) or “first two moments” (F2M) methods [46–50]. These methods are theoretically elegant, but typically require the sample size to be larger than the number of observed dimensions (although see [51] for some promising work). Other approaches formulate an optimization problem, such as projection pursuit [52], empirical risk minimization [53], or supervised dictionary learning [54]. These methods are limited because they are prone to fall into local minima, they require costly iterative algorithms, and lack any theoretical guarantees [53]. Thus, there remains a gap in the literature: a supervised learning method with theoretical convergence guarantees appropriate when the dimensionality is orders of magnitude larger than the sample size.

## Next Steps

The `LOL` idea, appending the mean difference vector to convert unsupervised manifold learning to supervised manifold learning, has many potential applications. We have presented the first few. Incorporating additional nonlinearities via kernel methods [55], ensemble methods such as random forests [56], multiscale methods [30], and more scalable implementations [57], are all of immediate interest.

ALGORITHMS	$p < n$	2-Class LDA MODELS, $p > n$			GENERALIZED MODELS, $p > n$					OTHER TASKS, $p > n$	ALGORITHM PROPERTIES							ALGORITHMS	REF	NOTES
	LDA model	delta & cov aligned	& cov misaligned	delta & cov misaligned rotated	Fat Tails	>2 Class	QDA Model	Outlier Model	Non- linear	Testing Regress	Fast	Simple	Theory	Open Source	Dense	2-step/ DL	Super- vised			
kNN	X	X	X	X	X	X		X	X	X		X	X	X	X		X	kNN	1	not considered
HCT	X	X	X								X	X	X			X	X	HCT	14	
LDA o Trace-Ratio	X	X	X	X							X	X	X		X	X	X	LDA o Trace-Ratio	4	
F2M / SDR	X			X		X							X	X	X	X	X	F2M / SDR	10	needs $n > p$
Deep Learning	X				X	X	X	X	X	X				X	X	X	X	Deep Learning	11	best when $n \gg p$
Vowpal Wabbit	X				X	X		X		X	X			X	X	X	X	Vowpal Wabbit	5	best when $n \gg p$
Ridge Regression	X										X	X	X	X	X		X	Ridge Regression	1	regression
Partial Least Squares										X	X						X	Partial Least Squares	1	regression
Naive Bayes	X	X	X			X					X	X	X	X	X		X	Naive Bayes	1,12	
HDDA	X	X	X	X		X								X	X		X	HDDA	6	
ROAD	X	X										X	X	X	X		X	ROAD	7,8,9	
LASSO	X	X				X				X	X	X	X	X		X	X	LASSO	3	
kSVM	X	X	X	X	X	X	X	X	X				X	X	X		X	kSVM	1	
Random Forest	X												X	X	X		X	Random Forest	1, 15	
LDA	X										X	X	X	X			X	LDA	1	
LDA o PCA	X	X				X				X	X	X	X	X	X	X	1/2	LDA o PCA	2	
LOL	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	LOL	*	
FIGURE		2A	2B, 3A	2C, 3B	3C	3D	3E	3F	3G	5A, 5D	5C, 5D	6								

Figure 7: Table of algorithms and their properties for high-dimensional data. Gray elements indicate that results are demonstrated in the Figure labeled in the bottom row. 'X' denotes relatively good performance for a given setting, or has the particular property.



## A Theoretical Background

### I.A The Classification Problem

Let  $(X, Y)$  be a pair of random variables, jointly sampled from  $F := F_{X,Y} = F_{X|Y}F_Y$ . Let  $X$  be a multivariate vector-valued random variable, such that its realizations live in  $p$  dimensional Euclidean space,  $x \in \mathbb{R}^p$ . Let  $Y$  be a categorical random variable, whose realizations are discrete,  $y \in \{0, 1, \dots, C\}$ . The goal of a classification problem is to find a function  $g(x)$  such that its output tends to be the true class label  $y$ :

$$g^*(x) := \operatorname{argmax}_{g \in \mathcal{G}} \mathbb{P}[g(x) = y].$$

When the joint distribution of the data is known, then the Bayes optimal solution is:

$$g^*(x) := \operatorname{argmax}_y f_{y|x} = \operatorname{argmax}_y f_{x|y}f_y = \operatorname{argmax}_y \{\log f_{x|y} + \log f_y\} \quad (1)$$

Denote expected misclassification rate of classifier  $g$  for a given joint distribution  $F$ ,

$$L_g^F := \mathbb{E}[g(x) \neq y] := \int \mathbb{P}[g(x) \neq y] f_{x,y} dx dy,$$

where  $\mathbb{E}$  is the expectation, which in this case, is with respect to  $F_{X,Y}$ . For brevity, we often simply write  $L_g$ , and we define  $L_* := L_{g^*}$ .

### I.B Linear Discriminant Analysis ( $LDA$ )

Linear Discriminant Analysis ( $LDA$ ) is an approach to classification that uses a linear function of the first two moments of the distribution of the data. More specifically, let  $\mu_j = \mathbb{E}[F_{X|Y=j}]$  denote the class conditional mean, and let  $\Sigma = \mathbb{E}[F_X^2]$  denote the joint covariance matrix, and  $\pi_j = \mathbb{P}[Y = j]$ . Using this notation, we can define the  $LDA$  classifier:

$$g_{Lda}(x) := \operatorname{argmin}_y \frac{1}{2}(x - \mu_0)^T \Sigma^{-1}(x - \mu_0) + \mathbb{I}\{Y = y\} \log \pi_y,$$

where  $\mathbb{I}\{\cdot\}$  is one when its argument is true, and zero otherwise. Let  $L_{Lda}^F$  be the misclassification rate of the above classifier for distribution  $F$ . Assuming equal class prior and centered means,  $\pi_0 = \pi_1$  and  $(\mu_0 + \mu_1)/2 = 0$ , re-arranging a bit, we obtain

$$g_{Lda}(x) := \operatorname{argmin}_y x^T \Sigma^{-1} \mu_y.$$

In words, the  $LDA$  classifier chooses the class for whom the projection of an input vector  $x$ , onto  $\Sigma^{-1} \mu_y$ , is maximized. When there are only two classes, this further simplifies to

$$g_{2-Lda}(x) := \mathbb{I}\{x^T \Sigma^{-1} \delta > 0\},$$

where  $\delta = \mu_0 - \mu_1$ . Note that the equal class prior and centered means assumptions merely changes the threshold constant from 0 to something else.

### I.C $LDA$ Model

A statistical model is a family of distributions indexed by a parameter  $\theta \in \Theta$ ,  $\mathcal{F}_\theta = \{F_\theta : \theta \in \Theta\}$ . Consider the special case of the above where  $F_{X|Y=y}$  is a multivariate Gaussian distribution,  $\mathcal{N}(\mu_y, \Sigma)$ , where each class has its own mean, but all classes have the same covariance. We refer to this model as the  $LDA$  model. Let  $\theta = (\pi, \mu, \Sigma)$ , and let  $\Theta_{C-Lda} = (\Delta_C, \mathbb{R}^{p \times C}, \mathbb{R}_{>0}^{p \times p})$ , where  $\mu = (\mu_1, \dots, \mu_C)$ ,  $\Delta_C$  is the  $C$  dimensional simplex, that is  $\Delta_C = \{x : x_i \geq 0 \forall i, \sum_i x_i = 1\}$ , and  $\mathbb{R}_{>0}^{p \times p}$  is the set of positive definite  $p \times p$  matrices. Denote  $\mathcal{F}_{Lda} = \{F_\theta : \theta \in \Theta_{Lda}\}$ , dropping the superscript  $C$  for brevity where appropriate. The following lemma is well known:

**Lemma 1.**  $L_{Lda}^F = L_*^F$  for any  $F \in \mathcal{F}_{Lda}$ .

*Proof.* Under the  $LDA$  model, the Bayes optimal classifier is available by plugging the explicit distributions into Eq. (1).  $\square$

## B Projection Based Classifiers

Let  $A \in \mathbb{R}^{d \times p}$  be an orthonormal matrix, that is, a matrix that projects  $p$  dimensional data into a  $d$  dimensional subspace, where  $AA^T$  is the  $d \times d$  identity matrix, and  $A^T A$  is symmetric  $p \times p$  matrix with rank  $d$ . The question that motivated this work is: what is the best projection matrix that we can estimate, to use to “pre-process” the data prior to applying  $LDA$ . Projecting the data  $x$  onto a low-dimensional subspace, and the classifying via  $LDA$  in that subspace is equivalent to redefining the parameters in the low-dimensional subspace,  $\Sigma_A = A \Sigma A^T \in \mathbb{R}^{d \times d}$  and  $\delta_A = A \delta \in \mathbb{R}^d$ , and then using  $g_{Lda}$ . When  $C = 2$ ,  $\pi_0 = \pi_1$ , and  $(\mu_0 + \mu_1)/2 = 0$ , this amounts to:

$$g_A^d(x) := \mathbb{I}\{(Ax)^T \Sigma_A^{-1} \delta_A > 0\}, \text{ where } A \in \mathbb{R}^{d \times p}. \quad (2)$$

Let  $L_A^d := \int \mathbb{P}[g_A(x) = y] f_{x,y} dx dy$ . Our goal therefore is to be able to choose  $A$  for a given parameter setting  $\theta = (\pi, \delta, \Sigma)$ , such that  $L_A$  is as small as possible (note that  $L_A$  will never be smaller than  $L_*$ ).

Formally, we seek to solve the following optimization problem:

$$\begin{aligned} & \underset{A}{\text{minimize}} && \mathbb{E}[\mathbb{I}\{x^T A^T \Sigma_A^{-1} \delta_A > 0\} \neq y] \\ & \text{subject to} && A \in \mathbb{R}^{p \times d}, \quad AA^T = I_{d \times d}, \end{aligned} \quad (3)$$

where  $I_{u \times v}$  is the  $u \times v$  identity matrix identity, that is,  $I(i, j) = 1$  for all  $i = j \leq \min(u, v)$ , and zero otherwise. Let  $\mathcal{A}^d = \{A : A \in \mathbb{R}^{d \times p}, AA^T = I_{d \times d}\}$ , and let  $\mathcal{A}_* \subset \mathcal{A}$  be the set of  $A$  that minimize Eq. (3), and let  $A_* \in \mathcal{A}_*$  (where we dropped the superscript  $d$  for brevity). Let  $L_A^* = L_{A_*}$  be the misclassification rate for any  $A \in \mathcal{A}_*$ , that is,  $L_A^*$  is the Bayes optimal misclassification rate for the classifier that composes  $A$  with  $LDA$ .

In our opinion, Eq. (3) is the simplest supervised manifold learning problem there is: a two-class classification problem, where the data are multivariate Gaussians with shared covariances, the manifold is linear, and the classification is done via  $LDA$ . Nonetheless, solving Eq. (3) is difficult, because we do not know how to evaluate the integral analytically, and we do not know any algorithms that are guaranteed to find the global optimum in finite time. This has led to previous work using a surrogate function [? ]. We proceed by studying a few natural choices for  $A$ .

### II.A Bayes Optimal Projection

**Lemma 2.**  $\delta^T \Sigma^{-1} \in \mathcal{A}_*$

*Proof.* Let  $B = (\Sigma^{-1} \delta)^T = \delta^T (\Sigma^{-1})^T = \delta^T \Sigma^{-1}$ , so that  $B^T = \Sigma^{-1} \delta$ , and plugging this in to Eq. (2), we obtain

$$\begin{aligned} g_B(x) &= \mathbb{I}\{xB^T \Sigma_B^{-1} \delta_B > 0\} \\ &= \mathbb{I}\{x^T \Sigma^{-1} \delta \times (\Sigma_B^{-1} \delta_B) > 0\} && \text{plugging in } B \\ &= \mathbb{I}\{x^T \Sigma^{-1} \delta > 0\} && \text{because } \Sigma_B^{-1} \delta_B > 0. \end{aligned}$$

In other words, letting  $B$  be the Bayes optimal projection recovers the Bayes classifier, as it should. Or, more formally, for any  $F \in \mathcal{F}_{Lda}$ ,  $L_{\delta^T \Sigma^{-1}} = L_*$   $\square$

## II.B Principle Components Analysis (PCA) Projection

Principle Components Analysis (PCA) finds the directions of maximal variance in a dataset. PCA is closely related to eigendecompositions and singular value decompositions (SVD). In particular, the top principle component of a matrix  $X \in \mathbb{R}^{p \times n}$ , whose columns are centered, is the eigenvector with the largest corresponding eigenvalue of the centered covariance matrix  $XX^T$ . SVD enables one to estimate this eigenvector without ever forming the outer product matrix, because SVD factorizes a matrix  $X$  into  $USV^T$ , where  $U$  and  $V$  are orthonormal  $p \times n$  matrices, and  $S$  is a diagonal matrix, whose diagonal values are decreasing,  $s_1 \geq s_2 \geq \dots > s_n$ . Defining  $U = [u_1, u_2, \dots, u_n]$ , where each  $u_i \in \mathbb{R}^p$ , then  $u_i$  is the  $i^{\text{th}}$  eigenvector, and  $s_i$  is the square root of the  $i^{\text{th}}$  eigenvalue of  $XX^T$ . Let  $A_d^{\text{PCA}} = [u_1, \dots, u_d]$  be the truncated PCA orthonormal matrix.

The PCA matrix is perhaps the most obvious choice of a orthonormal matrix for several reasons. First, truncated PCA minimizes the squared error loss between the original data matrix and all possible rank  $d$  representations:

$$\operatorname{argmin}_{A \in \mathbb{R}^{d \times p}: AA^T = I_{d \times d}} \|X - A^T A\|_F^2.$$

Second, the ubiquity of PCA has led to a large number of highly optimized numerical libraries for computing PCA (for example, LAPACK [58]).

Moreover, let  $U_d = [u_1, \dots, u_d] \in \mathbb{R}^{p \times d}$ , and note that  $U_d^T U_d = I_{d \times d}$  and  $U_d^T U_d = I_{p \times d}$ . Similarly, let  $USU^T = \Sigma$ , and  $US^{-1}U^T = \Sigma^{-1}$ . Let  $S_d$  be the matrix whose diagonal entries are the eigenvalues, up to the  $d^{\text{th}}$  one, that is  $S_d(i, j) = s_i$  for  $i = j \leq d$  and zero otherwise. Similarly,  $\Sigma_d = US_dU^T = U_d S_d U_d^T$ .

Let  $g_{\text{PCA}}^d := g_{A_{\text{PCA}}^d}$ , and let  $L_{\text{PCA}}^d := L_{A_{\text{PCA}}^d}$ . And let  $g_{\text{LDA}}^d := \mathbb{I}\{x \Sigma_d^{-1} \delta > 0\}$  be the regularized LDA classifier, that is, the LDA classifier, but sets the bottom  $p - d$  eigenvalues to zero.

**Lemma 3.**  $L_{\text{PCA}}^d = L_{\text{LDA}}^d$ .

*Proof.* Plugging  $U_d$  into Eq. (2) for  $A$ , and considering only the left side of the operand, we have

$$\begin{aligned} (Ax)^T \Sigma_A^{-1} \delta_A &= x^T A^T A \Sigma^{-1} A^T A \delta, \\ &= x^T U_d U_d^T \Sigma^{-1} U_d U_d^T \delta, \\ &= x^T U_d U_d^T U S^{-1} U U_d U_d^T \delta, \\ &= x^T U_d I_{d \times p} S^{-1} I_{p \times d} U_d^T \delta, \\ &= x^T U_d S_d^{-1} U_d^T \delta, \\ &= x^T \Sigma_d^{-1} \delta. \end{aligned}$$

□

The implication of this lemma is that if one desires to implement Fisherfaces, rather than first learning the eigenvectors and then learning LDA, one can instead directly implement regularized LDA by setting the bottom  $p - d$  eigenvalues to zero.

## II.C Linear Optimal Low-Rank (LOL) Projection

The basic idea of LOL is to use both  $\delta$  and the top  $d$  eigenvectors. Most naïvely, we could simply concatenate the two,  $A_{\text{LOL}}^d = [\delta, A_{\text{PCA}}^{d-1}]$ . Recall that eigenvectors are orthonormal. To maintain orthonormality, we could easily apply Gram-Schmidt,  $A_{\text{LOL}}^d = \text{ORTH}([\delta, A_{\text{PCA}}^{d-1}])$ . Both in practice and in theory (as will be shown below), this orthogonalization step does not matter much.

to ensure that they are balanced appropriately, we normalize  $\delta$

each vector in  $\delta$  to have norm unity. Formally, let  $\tilde{\delta}_j = \delta_j / \|\delta_j\|$ , where  $\delta_j$  is the  $j^{\text{th}}$  difference of the mean vector (remember, the number of vectors is equal to  $C - 1$ , where  $C$  is the total number of classes),

and let  $\mathbf{A}_{\text{LoI}}^d = [\tilde{\delta}, \mathbf{A}_{\text{PCA}}^{d-(C-1)}]$ . The eigenvectors are all normalized and orthogonal to one another; to impose orthogonality between  $\tilde{\delta}$  and the eigenvectors, we could use any number of numerically optimized algorithms. However, in practice, orthogonalizing does not matter very much, so we do not bother. We formally demonstrate this below.

## C Theoretical Properties of LDA based Classifiers

### III.A $\text{LDA}$ is rotationally invariant

For certain classification tasks, the ambient coordinates have intrinsic value, for example, when simple interpretability is desired. However, in many other contexts, interpretability is less important [59]. When the exploitation task at hand is invariant to rotations, then we have no reason to restrict our search space to be sparse in the ambient coordinates, rather, for example, we can consider sparsity in the eigenvector basis. Fisherfaces is one example of a rotationally invariant classifier, under certain model assumptions. Let  $\mathbf{W}$  be a rotation matrix, that is  $\mathbf{W} \in \mathcal{W} = \{\mathbf{W} : \mathbf{W}^\top = \mathbf{W}^{-1} \text{ and } \det(\mathbf{W}) = 1\}$ . Moreover, let  $\mathbf{W} \circ F$  denote the distribution  $F$  after transformation by an operator  $\mathbf{W}$ . For example, if  $F = \mathcal{N}(\mu, \Sigma)$  then  $\mathbf{W} \circ F = \mathcal{N}(\mathbf{W}\mu, \mathbf{W}\Sigma\mathbf{W}^\top)$ .

**Definition 1.** A rotationally invariant classifier has the following property:

$$L_g^F = L_g^{\mathbf{W} \circ F}, \quad F \in \mathcal{F}.$$

In words, the Bayes risk of using classifier  $g$  on distribution  $F$  is unchanged if  $F$  is first rotated, for any  $F \in \mathcal{F}$ .

Now, we can state the main lemma of this subsection:  $\text{LDA}$  is rotationally invariant.

**Lemma 4.**  $L_{\text{LDA}}^F = L_{\text{LDA}}^{\mathbf{W} \circ F}$ , for any  $F \in \mathcal{F}$ .

*Proof.*  $\text{LDA}$  simply becomes thresholding  $\mathbf{x}^\top \Sigma^{-1} \delta$ . Thus, we can demonstrate rotational invariance by demonstrating that  $\mathbf{x}^\top \Sigma^{-1} \delta$  is rotationally invariant.

$$\begin{aligned} (\mathbf{W}\mathbf{x})^\top (\mathbf{W}\Sigma\mathbf{W}^\top)^{-1} \mathbf{W}\delta &= \mathbf{x}^\top \mathbf{W}^\top (\mathbf{W}\mathbf{U}\mathbf{S}\mathbf{U}^\top \mathbf{W}^\top)^{-1} \mathbf{W}\delta && \text{by substituting } \mathbf{U}\mathbf{S}\mathbf{U}^\top \text{ for } \Sigma \\ &= \mathbf{x}^\top \mathbf{W}^\top (\tilde{\mathbf{U}}\tilde{\mathbf{S}}\tilde{\mathbf{U}}^\top)^{-1} \mathbf{W}\delta && \text{by letting } \tilde{\mathbf{U}} = \mathbf{W}\mathbf{U} \\ &= \mathbf{x}^\top \mathbf{W}^\top (\tilde{\mathbf{U}}\mathbf{S}^{-1}\tilde{\mathbf{U}}^\top) \mathbf{W}\delta && \text{by the laws of matrix inverse} \\ &= \mathbf{x}^\top \mathbf{W}^\top \mathbf{W}\mathbf{U}\mathbf{S}^{-1}\mathbf{U}^\top \mathbf{W}^\top \mathbf{W}\delta && \text{by un-substituting } \mathbf{W}\mathbf{U} = \tilde{\mathbf{U}} \\ &= \mathbf{x}^\top \mathbf{U}\mathbf{S}^{-1}\mathbf{U}^\top \delta && \text{because } \mathbf{W}^\top \mathbf{W} = \mathbf{I} \\ &= \mathbf{x}^\top \Sigma^{-1} \delta && \text{by un-substituting } \mathbf{U}\mathbf{S}^{-1}\mathbf{U}^\top = \Sigma \end{aligned}$$

□

One implication of this lemma is that we can reparameterize without loss of generality. Specifically, defining  $\mathbf{W} := \mathbf{U}^\top$  yields a change of variables:  $\Sigma \mapsto \mathbf{S}$  and  $\delta \mapsto \mathbf{U}^\top \delta := \delta''$ , where  $\mathbf{S}$  is a diagonal covariance matrix. Moreover, let  $\mathbf{d} = (\sigma_1, \dots, \sigma_D)^\top$  be the vector of eigenvalues, then  $\mathbf{S}^{-1}\delta' = \mathbf{d}^{-1} \odot \tilde{\delta}$ , where  $\odot$  is the Hadamard (entrywise) product. The  $\text{LDA}$  classifier may therefore be encoded by a unit vector,  $\tilde{\mathbf{d}} := \frac{1}{m} \mathbf{d}^{-1} \odot \tilde{\delta}'$ , and its magnitude,  $m := \|\mathbf{d}^{-1} \odot \tilde{\delta}\|$ . This will be useful later.

### III.B Rotation of Projection Based Linear Classifiers $g_A$

By a similar argument as above, one can easily show that:

$$\begin{aligned}
 (AWx)^T(AW\Sigma W^T A^T)^{-1}AW\delta &= x^T(W^T A^T)(AW)\Sigma^{-1}(W^T A^T)(AW)\delta \\
 &= x^T Y^T Y \Sigma^{-1} Y^T Y \delta \\
 &= x^T Z \Sigma^{-1} Z^T \delta \\
 &= x^T (Z \Sigma Z^T)^{-1} \delta = x^T \tilde{\Sigma}_d^{-1} \delta,
 \end{aligned}$$

where  $Y = AW \in \mathbb{R}^{d \times p}$  so that  $Z = Y^T Y$  is a symmetric  $p \times p$  matrix of rank  $d$ . In other words, rotating and then projecting is equivalent to a change of basis. The implications of the above is:

**Lemma 5.**  $g_A$  is rotationally invariant if and only if  $\text{span}(A) = \text{span}(\Sigma_d)$ . In other words, PCA is the only rotationally invariant projection.

### III.C Simplifying the Objective Function

Recalling Eq. (2), a projection based classifier is effectively thresholding the dot product of  $x$  with the linear projection operator  $P_A := A^T \Sigma_A^{-1} \delta_A \in \mathbb{R}^p$ , and let  $P_* := P_{A_*}$ . Unfortunately, the nonlinearity in in Eq. (3) makes analysis difficult. However, because of the linear nature of the classifier and projection matrix operator, an objective function that is simpler to evaluate is available. Define  $\angle(P, P') = \frac{P^T P'}{\|P\|_2 \|P'\|_2} \in (0, 1)$ , and consider

$$\begin{aligned}
 &\underset{A}{\text{minimize}} && -\angle(P_A, P_*), \\
 &\text{subject to} && A \in \mathbb{R}^{p \times d}, \quad AA^T = I_{d \times d}.
 \end{aligned} \tag{4}$$

**Lemma 6.** The solution to Eq. (4) is also the solution to Eq. (3) for any given  $d$ .

*Proof.* The minimum of Eq. (4) is clearly  $A = \Sigma^{-1} \delta$ , which is also the minimum of Eq. (3).  $\square$

**Remark 1.**  $\angle$  is merely the angle between two vectors, and is therefore scale invariant. In other words,  $\angle(P_A, P) = \angle(P_{cA}, P)$ , for any  $c > 0$ .

Given the above, we can evaluate various choices of  $A$  in terms of their induced projection operator  $P_A$  and the angle between said projection operators and the Bayes optimal projection operator.

**Lemma 7.**  $\angle(P_A, P_*) < 1 \implies L_A > L_*$

*Proof.* If  $\angle(P_A, P_*) < 1$ , then there exists an  $x$  such that  $\mathbb{I}\{x^T P_A > 0\} \neq \mathbb{I}\{x^T P_* > 0\}$ , and therefore,  $L_A > L_*$ .  $\square$

**Conjecture 1.**

$$\angle(P_A, P_*) \leq \angle(P_B, P_*) \implies L_A \leq L_B.$$

#### III.C(1) When $d = 1$

**Remark 2.** If  $A \in \mathcal{A}_*$  and  $B \notin \mathcal{A}_*$ , then  $\angle(A^T, A_*^T) = 1$  and  $\angle(B^T, P_*^T) < 1$ , and therefore  $\angle(A^T, A_*^T) > \angle(B^T, A_*^T)$ .

**Conjecture 2.** When  $d = 1$ :

$$\angle(A^T, A_*^T) \leq \angle(B^T, A_*^T) \implies \angle(P_A, P_*) \leq \angle(P_B, P_*).$$

*Proof.* I believe so, but i don't see how to prove it. A little arithmetic shows that the left hand side means that:

$$A \Sigma^{-1} \delta > B \Sigma^{-1} \delta.$$

Similarly, the right hand side means that:

$$\delta \mathbf{A}^\top \mathbf{A} \boldsymbol{\Sigma}^{-1} \mathbf{A}^\top \mathbf{a} > \delta \mathbf{B}^\top \mathbf{B} \boldsymbol{\Sigma}^{-1} \mathbf{B}^\top \mathbf{b},$$

where we substituted  $\mathbf{a} := \mathbf{A} \boldsymbol{\Sigma}^{-1} \boldsymbol{\delta}$  and  $\mathbf{b} := \mathbf{B} \boldsymbol{\Sigma}^{-1} \boldsymbol{\delta}$ .

But I do not see how to go any further. □



## III.D PCA versus LOL

We would like to prove that  $\text{LOL}$  is always better than  $\text{PCA}$ , when using one or the other to project the data onto a low dimensional space, followed by classifying with  $\text{LDA}$ , under a wide variety of settings. Formally, we would like to prove  $\mathbb{P}[L_{\text{LOL}}^d \leq L_{\text{PCA}}^d]$  is big. To do so, we ask a sequence of increasingly sophisticated questions that have the following form:

1. Under which parameter settings is  $\text{LOL}$  better than  $\text{PCA}$ ?
2. How often do those parameter settings arise, under various statistical models of the parameters?

Recall that for the  $C$ -class classification problem, the parameter from which we sample the data is  $\theta_c = (\pi_c, \mu_c, \Sigma_c)$ , where

- the class probabilities are non-negative and sum to unity:  $\pi = (\pi_1, \dots, \pi_C) \in \Delta_C := \sum_{c \in \mathcal{C}} \pi_c = 1$  and  $\pi_c \geq 0 \forall c \in \mathcal{C}$ ,
- the class means are  $p$ -dimensional vectors:  $\mu_c \in \mathbb{R}^p$  is the class  $c$  mean vector, and
- the class covariances are positive definite  $p \times p$  real matrices:  $\Sigma_c \in \mathbb{R}_+^{p \times p}$  is the class conditional covariance matrix (and  $\mathbb{R}_+^{p \times p}$  is the set of positive definite real  $p \times p$  matrices).

The fully unconstrained parameter space is therefore  $\Theta = \{\Delta_C \times (\mathbb{R}^p, \mathbb{R}_+^{p \times p})^C\}$ .

When using a projection based classifier, we also have the hyperparameter  $d$  which specifies the dimensionality of the low-dimensional projection.

We thus try to answer the above two questions for increasingly relaxed constraints on the parameter space and  $d$ . To start, we assume that:

1. we have only two classes,  $C = 2$ ;
2. each class has the same covariance matrix,  $\Sigma_0 = \Sigma_1$ .

For simplicity (but without loss of generality), we will also assume that the two classes have centered means and equal priors, that is,  $(\mu_0 + \mu_1)/2 = 0$  and  $\pi_0 = \pi_1 = 1/2$  (relaxing this assumption merely changes the threshold of the classifier). Let the parameter space defined by these constraints be denoted  $\Theta' = \{\mathbb{R}^{2p}, \mathbb{R}_+^{p \times p}\}$ . In Section III.D, we will always assume  $\theta \in \Theta'$ , unless otherwise specified.

Given the above, we consider the following sequence of relaxations, first when  $d = 1$ , and then when  $d < p$ :

1. the covariance is a scaled identity matrix, that is,  $\Sigma = kI$ ;
2. the covariance is a diagonal matrix, that is,  $\Sigma = S$ , where  $S_{ij} = \sigma_i$  when  $i = j$  and is zero otherwise;
3. the covariance is an arbitrary positive definite matrix, that is  $\Sigma \in \mathbb{R}_+^{p \times p}$ .

After those, extensions to the multiclass and/or different covariance matrix setting might then also be explored.

III.D(1)  $d = 1$ 

In this section, we will only consider  $d = 1$ , meaning that

- $\text{LOL}$  is simply  $\delta$ ,
- $\text{PCA}$  is simply  $u_1$ , the eigenvector corresponding to the largest eigenvalue of  $\Sigma$ .

## III.D(1).1 Scaled Identity Covariance Matrix

**Lemma 8.**  $\text{LOL}$  better than  $\text{PCA}$  almost always when  $\Sigma = kI$ .

*Proof.* To prove this statement, we first show how to define  $A_{\text{LOL}}^1$  and  $A_{\text{PCA}}^1$  in this setting.

- $A_{\text{LOL}}^1$  is simply  $\delta$ , regardless of the covariance matrices and priors.
- When  $\Sigma = kI$ ,  $A_{\text{PCA}}^1$  is a random vector, because the first principal component of a scaled identity matrix can equally be defined as any basis vector.

Thus, only when the first eigenvector of the covariance matrix is randomly assigned to  $\delta$  will  $\text{PCA}$  work as

well as LOL.  $\square$

Given the conditions under which LOL is better than PCA, we next ask how often that happens, under a reasonable model assumption.

**Lemma 9.**  $\mathbb{P}[L_{LoL}^1 \leq L_{PCA}^1] = 1$  when  $\Sigma = kI$  and  $\delta \sim \mathcal{N}(\mu_\delta, \Sigma_\delta)$ .

*Proof.* When  $\delta \sim \mathcal{N}(\mu_\delta, \Sigma_\delta)$ , the probability that it exactly equals any vector  $x \in \mathbb{R}^d$  is real, this includes, of course,  $u_1$ , the first eigenvector of  $kI$ . Thus,  $\mathbb{P}[\delta \propto u_1] = 0$ , no matter how  $u_1$  is chosen (as long as it is not chosen using the knowledge of the value of  $\delta$ ). This implies that  $\mathbb{P}[\angle(u_1^T, A_*^T) = 1] = 0$ , and therefore,  $\mathbb{P}[L_{u_1} \geq L_*] = 1$ .  $\square$

### III.D(1).2 Diagonal covariance matrix

Now consider a simple generalization of the above scenario, namely,  $\Sigma = S$  is diagonal. We want to know under what conditions is LOL better than PCA, and how often that happens under a reasonable model.

Let  $s$  be the singular values of  $\Sigma$ . When  $\Sigma$  is diagonal, its diagonal elements are  $s_1, \dots, s_p$ , and we call that matrix  $S$ . Recalling that  $s_i = \lambda_i^2$ , where  $\lambda_i$ 's are eigenvalues, and when a matrix is diagonal,  $s_i = \lambda_i^2$ , we have  $A_* = S^{-1}\delta = s^{-1} \odot \delta := \tilde{\delta} = (\delta_1/\lambda_1^2, \dots, \delta_p/\lambda_p^2)$ .

**Lemma 10.** LOL is better than PCA whenever  $1 - \delta_1 < s_1/\delta_1 \sum_{i=2}^p \delta_i^2/s_i$ . A consequence of this is that LOL is always better than PCA whenever  $\delta_1 > 1$ . This means that when the variables are independent

*Proof.* Recalling that  $u_i = e_i$ , where  $e_i$  is a vector of all zeros except a one in the  $i^{th}$  element, and let  $u^i = u_i$  for simplicity here, we can show that:

$$\angle(\delta, \tilde{\delta}) = \sum_i \delta_i \tilde{\delta}_i = \sum_i \delta_i s_i^{-1} \delta_i = \sum_i \delta_i^2 / s_i$$

and

$$\angle(u^1, \tilde{\delta}) = \sum_i u_i^1 \tilde{\delta}_i = \sum_i u_i^1 s_i^{-1} \delta_i = \sum_i u_i^1 \delta_i / s_i = u_1^1 \delta_1 / s_1 = \delta_1 / s_1.$$

Thus,  $\angle(\delta, \tilde{\delta}) > \angle(u_1, \tilde{\delta})$  any time that  $\sum_i \delta_i^2 / s_i > \delta_1 / s_1$ . Note that at a minimum, it must be that  $\delta_1 < 1$ .

Conjectures 1 and 2 demonstrate that if the angle is better, then Bayes error is also better, and this completes the proof.  $\square$

If Conjecture 1 and 2 are true, then the above also implies that LOL's Bayes error is also better than PCA's, in this setting. Numerical experiments suggest this is true, and moreover, that when randomly sampling  $\delta$  and  $S$ , LOL always does better than PCA.

**Conjecture 3.** LOL is almost always better than PCA whenever  $s_i \stackrel{iid}{\sim} \mathcal{U}(0, 1)$  and  $\delta_i \stackrel{iid}{\sim} \mathcal{N}(0, 1)$  for all  $i \in [p]$ .

*Proof.* We start with the very simple scenario of  $p = 1$  and condition on  $s$  to build intuition. In this setting we have:

$$\mathbb{P}[\delta^2/s > \delta/s] = \mathbb{P}[\delta^2/s - \delta/s > 0]. \quad (5)$$

Let  $z = x + y$ , where  $x = \frac{\delta^2}{s}$  and  $y = \frac{\delta}{-s}$ . So,  $x$  has a generalized chi-squared distribution and  $y$  has a normal distribution. A bit of algebra reveals the explicit form of this distribution, however, it includes an integral that we do not know how to evaluate. Thus, instead of solving exactly for the above, we provide a bound.

Bennett's inequality states given  $X_1, \dots, X_n$ , assume:

- each  $X_i$  is independent of all others

### III.D PCA versus LOL

### C THEORETICAL PROPERTIES OF LDA BASED CLASSIFIERS

- (without loss of generality)  $\mathbb{E}[X_i] = 0 \forall i$
- $|X_i| < a$  almost surely for all  $i$
- $\sigma^2 = \frac{1}{n} \sum_i \mathbb{V}(X_i)$ .

Then, for any  $t \geq 0$

where  $h(u) = (1 + u) \log(1 + u) - u$ .

Recalling that  $\mathbb{V}(\sum_i X_i) = \sum_i \mathbb{V}(X_i)$  for independent random variables, and that

1.  $\mathbb{V}(\delta/s) = s^{-2}$
2.  $\mathbb{V}(\delta^2/s) = h(s)$

where  $\Phi$  is the cumulative distribution function for a standard normal random variable.

Note that re-introducing randomness into  $s$

Coming soon. “almost always” and “better” mean with probability 1 and better in the sense of the angle between  $A_{\text{LoL}}^1$  and the Bayes vector is larger than that of  $A_{\text{PCA}}^1$  and the Bayes vector. I believe this also implies Bayes error is better, but we have not yet shown that.  $\square$

#### III.D(1).3 Arbitrary Covariance Matrix

#### III.D(2) $d \geq 1$

#### III.D(3) Scaled Identity Covariance Matrix

#### III.D(4) Diagonal Covariance Matrix

#### III.D(5) Arbitrary Covariance Matrix

## D Asymptotic Theory

In real data problems, the true joint distribution is unknown. Instead, what is provided is a set of training data. We therefore assume the existence of  $n$  training samples, each of which has been sampled identically and independently from the same distribution,  $(\mathbf{X}_i, Y_i) \stackrel{iid}{\sim} F_{\mathbf{X}, Y}$ , for  $i = 1, 2, \dots, n$ . We can use these training samples to then estimate  $f_{x|y}$  and  $f_y$ . Plugging these estimates in to Eq. (??), we obtain the Bayes plugin classifier:

$$\hat{g}_n^*(\mathbf{x}) := \operatorname{argmax}_y \hat{f}_{\mathbf{x}|y} \hat{f}_y. \quad (6)$$

Under suitable conditions, it is easy to show that this Bayes plugin classifiers performance is asymptotically optimal. Formally, we know that:  $L_{\hat{g}_n^*} \rightarrow L_{g^*}$ .

When the parameters, and we want to use a linear approach, we can implement a Bayes plug-in LDA, which we call Fisher's Discriminant Analysis (FLD) [60]. Under the two-class, equal prior, and centered means assumption, we have

$$\hat{g}_n^*(\mathbf{x}) := \mathbb{I}\{\mathbf{x}^\top \hat{\Sigma}^{-1} \hat{\delta} > 0\}, \quad (7)$$

and  $L_{\hat{g}_n}$  is the misclassification rate for an estimated classifier,  $\hat{g}_n$ . Unfortunately, when  $p \gg n$ , the estimate of the covariance matrix  $\Sigma$  will be low-rank, and therefore, not invertible (because an infinite number of solutions all fit equally well). In such scenarios, we seek alternative methods, even in the LDA model.

We would like to prove:

**Lemma 11.**  $L_{\hat{g}_n}^d \rightarrow L_*$  for any  $\theta \in \Theta'$ .

**Lemma 12.**  $\mathbb{P}[L_{\hat{g}_n}^d \rightarrow L_*] > 0$  for any  $\theta \in \Theta'$

## E Finite Sample Theory

It would be awesome to prove something like:

**Theorem 2.**

$$\mathbb{P}[\hat{L}_{LoL}^d - \hat{L}_{PcA}^2 > 0] < f(\theta, d, n),$$

which would state that  $LoL$  is better than  $PcA$ , again, under suitable assumptions.

**Theorem 3.**

$$\mathbb{P}[\mathbf{P}_{PcA}^\top \mathbf{P}_* - \mathbf{P}_{LoL}^\top \mathbf{P}_* > t \|\mathbf{P}_A\| \|\mathbf{P}_*\|] < f(t, p, d),$$

which would state that  $LoL$  is better than  $PcA$ , again, under suitable assumptions.

In terms of distributions of the above, it seems that perhaps we could start simple. Assume for the moment that  $\delta, \mathbf{u}_1, \dots, \mathbf{u}_p \stackrel{iid}{\sim} \mathcal{N}(\mu_p, \Sigma_p)$ , and let  $\mathbf{\Lambda} = (\mathbf{u}_1, \dots, \mathbf{u}_p)^\top$ , and  $\Sigma = \mathbf{\Lambda}^\top \mathbf{\Lambda}$ .

The reason the above is probabilistic is because it is under certain assumptions on the *distributions* of  $\delta$ ,  $\Sigma$ , and  $\mathbf{A}$ .

Perhaps even simpler is to start with specific assumptions about  $\delta$ ,  $\Sigma$ , and  $\mathbf{A}$ . Because  $LoL$  is rotationally invariant, I believe that we can assert, without loss of generality, that  $\Sigma = \mathbf{S}$ , where  $\mathbf{S}$  is a diagonal matrix with diagonal entries  $\sigma_1, \dots, \sigma_p$ , where all  $\sigma_j > 0$ . Now, the optimal projection  $\Sigma^{-1}\delta$  is just a simple dot product,  $\mathbf{d}^\top \delta$ , where  $\mathbf{d} = \text{diag}(\mathbf{S}) \in \mathbb{R}^p$ .

For example, letting  $\mathbf{A} = \mathbf{U}_d$ , and letting  $\mathbf{U}_i = \mathbf{e}_i$  be the unit vector, with zeros everywhere except a one in the  $i^{th}$  position, we have

$$\mathbf{P}_A^\top \mathbf{P}_* = \delta^\top \mathbf{U}_d^\top \mathbf{U}_d \Sigma^{-1} \mathbf{U}_d^\top \mathbf{U}_d \Sigma^{-1} \delta \delta^\top \Sigma_d \Sigma^{-1} \Sigma_d \Sigma^{-1} \delta = \delta^\top \Sigma^{-2} \delta.$$

So, we want to understand the probability that  $\alpha_{PcA}$  is small under different parameter settings,  $\theta \in \Theta$ .

## F Bibliography

- [1] J. T. Vogelstein, Y. Park, M. Zlatić, and C. E. Priebe, “Discovery of Brainwide Neural-Behavioral Maps via Multiscale Unsupervised Structure Learning,” *Science*, vol. 344, pp. 386 — 392, 2014. 1
- [2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105. [Online]. Available: <http://papers.nips.cc/paper/4824-imagenet-classification-w> 1
- [3] T. Hastie, R. Tibshirani, and J. H. Friedman, “The Elements of Statistical Learning: Data Mining, Inference, and Prediction,” *Beijing: Publishing House of Electronics Industry*, 2004. 1, 9
- [4] J. B. Tenenbaum, V. de Silva, J. C. Langford, and V. D. Silva, “A global geometric framework for nonlinear dimensionality reduction.” *Science*, vol. 290, no. 5500, pp. 2319–23, dec 2000. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/11125149> 3, 13
- [5] S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding.” *Science (New York, N.Y.)*, vol. 290, no. 5500, pp. 2323–6, dec 2000. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/11125150> 3
- [6] M. Belkin and P. Niyogi, “Laplacian Eigenmaps for Dimensionality Reduction and Data,” *Neural Computation*, vol. 15, pp. 1373–1396, 2003. 3
- [7] R. R. Coifman and S. Lafon, “Diffusion maps,” *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 5–30, jul 2006. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S1063520306000546> 3, 13
- [8] M. Belkin, I. Matveeva, and P. Niyogi, “Regularization and semi-supervised learning on large graphs,” in *COLT*, 2004. 3
- [9] H. Zou, “The Adaptive Lasso and Its Oracle Properties,” pp. 1418–1429, 2006. 4
- [10] Y. LeCun, C. Cortes, and C. Burges, “MNIST handwritten digit database.” [Online]. Available: <http://yann.lecun.com/exdb/mnist/> 4
- [11] P. J. Bickel and E. Levina, “Some theory for Fisher’s linear discriminant function, ‘naive Bayes’, and some alternatives when there are many more variables than observations,” *Bernoulli*, vol. 10, no. 6, pp. 989–1010, dec 2004. [Online]. Available: <http://projecteuclid.org/euclid.bj/1106314847> 5, 9, 13
- [12] J. Fan, Y. Feng, and X. Tong, “A road to classification in high dimensional space: the regularized optimal affine discriminant,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 74, no. 4, pp. 745–771, sep 2012. [Online]. Available: <http://doi.wiley.com/10.1111/j.1467-9868.2012.01029.x> 6, 13
- [13] J. Baik and J. W. Silverstein, “Eigenvalues of large sample covariance matrices of spiked population models,” *Journal of Multivariate Analysis*, vol. 97, no. 6, pp. 1382–1408, 2006. 5
- [14] D. Paul, “Asymptotics of sample eigenstructure for a large dimensional spiked covariance model,” *Statistica Sinica*, vol. 17, no. 4, p. 1617, 2007. [Online]. Available: <http://www3.stat.sinica.edu.tw/statistica/oldpdf/A17n418.pdf> 5
- [15] G. V. Trunk, “A problem of dimensionality: a simple example.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, no. 3, pp. 306–307, 1979. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4766926> 7
- [16] P. J. Huber, *Robust Statistics*, ser. Wiley Series in Probability and Statistics. Wiley, 1981, vol. 82, no. 3. [Online]. Available: <http://doi.wiley.com/10.1002/9780470434697> 9
- [17] P. J. Rousseeuw and K. V. Driessen, “A Fast Algorithm for the Minimum Covariance Determinant Estimator,” *Technometrics*, vol. 41, no. 3, pp. 212–223, mar 1999. [Online]. Available: <http://amstat.tandfonline.com/doi/abs/10.1080/00401706.1999.10485670{#}.VTejtq1Viko>



- [18] D. Ferrari and Y. Yang, "Maximum Lq-likelihood estimation," *IEEE Transactions on Nuclear Science*, vol. 57, no. 3, pp. 1077–1084, 2010. [Online]. Available: <http://arxiv.org/abs/1002.4533> 9
- [19] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust Principal Component Analysis?" *Journal of the ACM*, vol. 58, no. 3, pp. 1–37, dec 2009. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1970392.1970395http://arxiv.org/abs/0912.3599> 9
- [20] T. Zhang and G. Lerman, "A Novel M-Estimator for Robust PCA," *Journal of Machine Learning Research*, vol. 15, pp. 749–808, 2014. [Online]. Available: <http://jmlr.org/papers/v15/zhang14a.html> 9
- [21] C. M. Bishop, *Pattern Recognition and Machine Learning*, ser. Information science and statistics. Springer, 2006. [Online]. Available: <http://www.library.wisc.edu/selectedtocs/bg0137.pdf> 9
- [22] N. Halko, P. G. Martinsson, and J. A. Tropp, "Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions," *SIAM Review*, vol. 53, no. 2, pp. 217–288, jan 2011. [Online]. Available: <http://epubs.siam.org/doi/abs/10.1137/090771806> 9
- [23] E. J. Candès, "Compressive sampling," *Proceedings of the International Congress of Mathematicians, Madrid, Spain*, vol. 3, pp. 1433–1452, 2006. 9
- [24] M. Lopes, L. Jacob, and M. J. Wainwright, "A More Powerful Two-Sample Test in High Dimensions using Random Projection," in *Neural Information Processing Systems*, 2011, pp. 1206–1214. [Online]. Available: <http://papers.nips.cc/paper/4260-a-more-powerful-two-sample-test-in-high-dimensions-using-random-projection> 12
- [25] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997. 13
- [26] V. de Silva and J. B. Tenenbaum, "Global Versus Local Methods in Nonlinear Dimensionality Reduction," in *Neural Information Processing Systems*, 2003, pp. 721–728. 13
- [27] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, no. 1, pp. 59–69, 1982. [Online]. Available: <http://link.springer.com/10.1007/BF00337288> 13
- [28] C. M. Bishop, M. Svensén, and C. K. I. Williams, "GTM: The Generative Topographic Mapping," *Neural Computation*, vol. 10, no. 1, pp. 215–234, jan 1998. [Online]. Available: [#}.VMIRX17F-oU](http://www.mitpressjournals.org/doi/abs/10.1162/089976698300017953)
- [29] Z. Zhang and H. Zha, "Principal Manifolds and Nonlinear Dimensionality Reduction via Tangent Space Alignment," *SIAM Journal on Scientific Computing*, vol. 26, no. 1, pp. 313–338, jan 2004. [Online]. Available: <http://epubs.siam.org/doi/abs/10.1137/S1064827502419154>
- [30] W. K. Allard, G. Chen, and M. Maggioni, "Multi-scale geometric methods for data sets II: Geometric Multi-Resolution Analysis," *Applied and Computational Harmonic Analysis*, vol. 32, no. 3, pp. 435–462, may 2012. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S1063520311000868> 13
- [31] J. Ham, D. D. Lee, S. Mika, and B. Schölkopf, "A kernel view of the dimensionality reduction of manifolds," in *Twenty-first international conference on Machine learning - ICML '04*. New York, New York, USA: ACM Press, jul 2004, p. 47. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1015330.1015417> 13
- [32] J. H. Friedman, "Regularized Discriminant Analysis," *Journal of the American Statistical Association*, vol. 84, no. 405, pp. 165–175, mar 1989. [Online]. Available: <http://www.tandfonline.com/doi/abs/10.1080/01621459.1989.10478752> 13
- [33] C. Bouveyron, S. Girard, and C. Schmid, "High-dimensional data clustering," *Computational Statistics & Data Analysis*, vol. 52, no. 1, pp. 502–519, sep 2007. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0167947307000692> 13

- [34] D. M. Witten and R. Tibshirani, "Covariance-regularized regression and classification for high-dimensional problems." *Journal of the Royal Statistical Society. Series B, Statistical methodology*, vol. 71, no. 3, pp. 615–636, feb 2009. [Online]. Available: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2806603&tool=pmcentrez&rendertype=abstract> 13
- [35] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?" *Vision Research*, vol. 37, no. 23, pp. 3311–3325, dec 1997. [Online]. Available: [http://dx.doi.org/10.1016/S0042-6989\(97\)00169-7](http://dx.doi.org/10.1016/S0042-6989(97)00169-7) 13
- [36] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society. Series B*, vol. 58, pp. 267–288, 1996. 13
- [37] D. L. Donoho and J. Jin, "Higher criticism thresholding: Optimal feature selection when useful features are rare and weak." *Proceedings of the National Academy of Sciences of the United States of America*, vol. 105, no. 39, pp. 14790–5, sep 2008. [Online]. Available: <http://www.pnas.org/content/105/39/14790><http://www.pnas.org/content/105/39/14790.short>
- [38] J. Fan and J. Lv, "Sure independence screening for ultrahigh dimensional feature space," *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, vol. 70, no. 5, pp. 849–911, 2008.
- [39] R. Li, W. Zhong, and L. Zhu, "Feature Screening via Distance Correlation Learning," *Journal of the American Statistical Association*, vol. 107, no. 499, pp. 1129–1139, sep 2012. [Online]. Available: <http://www.tandfonline.com/doi/abs/10.1080/01621459.2012.695654>
- [40] R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu, "Diagnosis of multiple cancer types by shrunken centroids of gene expression." *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 10, pp. 6567–6572, may 2002. [Online]. Available: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=124443&tool=pmcentrez&rendertype=abstract>
- [41] J. Fan and Y. Fan, "High-dimensional classification using features annealed independence rules," *The Annals of Statistics*, vol. 36, no. 6, pp. 2605–2637, dec 2008. [Online]. Available: <http://projecteuclid.org/euclid.aos/1231165181>
- [42] L. Clemmensen, T. Hastie, D. M. Witten, and B. Ersbøll, "Sparse Discriminant Analysis," *Technometrics*, vol. 53, no. 4, pp. 406–413, nov 2011. [Online]. Available: <http://pubs.amstat.org/doi/abs/10.1198/TECH.2011.08118>
- [43] Q. Mai and H. Zou, "A Note On the Connection and Equivalence of Three Sparse Linear Discriminant Analysis Methods," *Technometrics*, vol. 55, pp. 243–246, 2013. [Online]. Available: <http://www.tandfonline.com/doi/abs/10.1080/00401706.2012.746208> 13
- [44] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least Angle Regression," *Annals of Statistics*, vol. 32, no. 2, pp. 407–499, 2004. [Online]. Available: <http://arxiv.org/abs/math/0406456> 13
- [45] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, sep 1936. [Online]. Available: <http://www.springerlink.com/content/9v4274h33h75lq24/> 13
- [46] K.-C. Li, "Sliced Inverse Regression for Dimension Reduction," *Journal of the American Statistical Association*, vol. 86, no. 414, pp. 316–327, jun 1991. [Online]. Available: <http://www.tandfonline.com/doi/abs/10.1080/01621459.1991.10475035> 13
- [47] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method arXiv : physics / 0004057v1 [ physics . data-an ] 24 Apr 2000," *Neural Computation*, pp. 1–16, 1999.
- [48] A. Globerson and N. Tishby, "Sufficient Dimensionality Reduction," *Journal of Machine Learning Research*, vol. 3, no. 7-8, pp. 1307–1331, oct 2003. [Online]. Available: <http://www.crossref.org/jmlr/{-}DOI.html>

- [49] R. D. Cook and L. Ni, "Sufficient Dimension Reduction via Inverse Regression," *Journal of the American Statistical Association*, vol. 100, no. 470, pp. 410–428, jun 2005. [Online]. Available: <http://amstat.tandfonline.com/doi/abs/10.1198/016214504000001501{#}.U6tH3Y1dUts>
- [50] K. Fukumizu, F. R. Bach, and M. I. Jordan, "Dimensionality Reduction for Supervised Learning with Reproducing Kernel Hilbert Spaces," *Journal of Machine Learning Research*, vol. 5, pp. 73–99, 2004. 13
- [51] R. D. Cook, L. Forzani, and A. J. Rothman, "Prediction in abundant high-dimensional linear regression," *Electronic Journal of Statistics*, vol. 7, pp. 3059–3088, 2013. [Online]. Available: <https://projecteuclid.org/euclid.ejs/1387207935> 13
- [52] P. J. Huber, "Projection Pursuit," *The Annals of Statistics*, vol. 13, no. 2, pp. 435–475, jun 1985. [Online]. Available: <http://projecteuclid.org/euclid.aos/1176349519> 13
- [53] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples," *The Journal of Machine Learning Research*, vol. 7, pp. 2399–2434, dec 2006. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1248547.1248632http://dl.acm.org/citation.cfm?id=1248632> 13
- [54] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach, "Supervised Dictionary Learning," in *Advances in Neural Information Processing Systems*, 2009, pp. 1033–1040. [Online]. Available: <http://papers.nips.cc/paper/3448-supervised> 13
- [55] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. Mullers, "Fisher discriminant analysis with kernels," in *Neural Networks for Signal Processing IX: Proceedings of the 1999 IEEE Signal Processing Society Workshop (Cat. No.98TH8468)*. IEEE, 1999, pp. 41–48. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=788121> 13
- [56] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001. 13
- [57] C.-C. Chang and C.-J. Lin, "LIBSVM," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1–27, apr 2011. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1961189.1961199> 13
- [58] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. D. Croz, A. Greenbaum, S. Hammerling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide: Third Edition*. SIAM, 1999. [Online]. Available: <https://books.google.com/books?hl=en{&}lr={&}id=AZlvEnr9gCgC{&}pgis=1> 17
- [59] L. Breiman, "Statistical Modeling : The Two Cultures," *Statistical Science*, vol. 16, no. 3, pp. 199–231, 2001. 18
- [60] R. A. Fisher, "Theory of Statistical Estimation," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 22, no. 05, pp. 700–725, oct 1925. [Online]. Available: [http://journals.cambridge.org/abstract{\\_-}S0305004100009580](http://journals.cambridge.org/abstract{_-}S0305004100009580) 24