METRIC PRINCIPLE COMPONENT ANALYSIS:
ON IDENTIFYING IMPORTANT SUBSPACES FOR APPROXIMATION

Thomas C. H. Lux
Tyler H. Chang

Department of Computer Science
Virginia Polytechnic Institute & State University
2000 Torgersen Hall, Blacksburg, VA, USA
{tchlux,thchang}@vt.edu

ABSTRACT

Many modern data science problems for approximation have high dimension while the true underlying phenomenon can be accurately represented from a fraction of the provided information. A variety of dimension reduction techniques exist, but most target unsupervised applications rather than approximation problems or only accommodate a small set of approximation problems. This paper presents metric principle component analysis (MPCA), an extension of standard principle component analysis (PCA) that accounts for metric variation in applied approximation problems like classification and regression. MPCA is theoretically motivated, shown to be a superset of classical PCA, and its usefulness as a dimension reduction technique for approximation problems is demonstrated. Initial results reveal both that MPCA can be significantly better in some cases, and that in other applications it performs similarly to standard PCA.

Keywords: principle component analysis, classification, regression, approximation, metric learning

1 INTRODUCTION

Dimension reduction is an important problem in data science and, more generally, function approximation. Consider a multivariable function $f : \mathbb{R}^d \to (S,m)$ where (S,m) denotes a metric space with metric m. In the context of data science, (S,m) could represent a discrete classification space, a continuous real-valued space, or could be descriptive of some graph structure. In this paper, the supervised learning problem is considered, where a finite set $X \subset \mathbb{R}^d$ of data points are given along with labeled function values $Y = \{y_i = f(x_i) | x_i \in X\}$. There are many algorithms in the field of data science, mathematics, and machine learning for predicting general functions such as multivariate spline approximations, neural networks, clustering algorithms, support vector machines/regressors, and more. One particularly common yet effective class of solutions to these problems are distance based approximations such as k-nearest neighbor (KNN). Though effective for many classes of problems, these methods depend on the premise that distance in the input space can be associated with change in the output class or value. While this assumption holds for most of the common "big data" problems, it is often the case that some number of directions in the input space have no bearing on the output class/value. Without applying dimension reduction, these directions will increase the cost of any distance based approximation techniques, in some cases making the prediction techniques computationally intractable. Furthermore, such "meaningless directions" cannot improve prediction accuracy, and often decrease accuracy by disassociating distance from change in class/value.

In this paper, a novel technique for performing dimension reduction is proposed that considers how the output $f(x)$ changes given the data X and function value/labels Y. Using this information, an orthonormal basis is constructed for a new subspace of $\mathbb{R}^d$ where every direction is associated with some change in f, thereby increasing the accuracy of distance based methods while often significantly reducing the dimension. The technique is model agnostic, meaning that it can be applied as a "pre-conditioning" step before any distance based approximation. The effectiveness of the technique is demonstrated on one theoretical and three real-world problems, using four different distance based approximation algorithms as well as a multilayer perceptron (MLP).

The remainder of this paper is structured as follows: Section 2 considers existing dimension reduction techniques for approximation problems, Section 3 explains the methodology of computing MPCA and provides a theoretical example, Section 4 shows results on real-world problems, Section 5 discusses the implications of results, and finally Section 6 concludes.

2 BACKGROUND

2.1 Related Work

One of the most standard methods for performing dimension reduction is correlation analysis, whereby a general model[1] is fit[2] to the data and the coefficients of each term are analyzed to determine whether that term has any effect on the output. After dropping all terms that have no statistically significant effect on the function value, it can be inferred that only the directions associated with the remaining terms affect the output class or function value. These techniques are highly effective when an appropriate model is chosen, but are highly model dependent and generally fail for a black-box output function f. For example, this technique is most commonly applied by fitting a linear model to labeled data. Then, if the function f is not itself approximately linear, the above technique can fail to produce any meaningful results.

Principle component analysis (PCA) is another commonly used dimension reduction technique. An explanation of Principle Component Analysis and its applications can be found in Shlens (2014), Powell and Lehe (2014). Given a finite set of unlabeled points $X \subset \mathbb{R}^d$, PCA identifies an ordered set of orthonormal basis vectors $\{v_1, \ldots, v_d\}$, such that the projection of X onto any subspace defined by the span of $\{v_1, \ldots, v_k\}$, $k \leq d$ has maximal variance. Equivalently, PCA is the dimension reduction technique that minimizes reconstruction error for X with respect to mean squared error (MSE). In its standard application, PCA is an unsupervised technique, and does not consider the effect of each dimension on output class or function values.

Recently a modified variant of PCA was proposed which incorporates function values into the subspace basis and the technique is referred to as *supervised PCA* (Barshan et al. 2011). However this technique relies on the coefficients of a linear fit and faces the same nonlinear approximation weaknesses (as will be demonstrated in a theoretical example).

Other notable techniques include linear discriminant analysis (LDA) and neural network autoencoders. LDA depends heavily on linear separability of the output classes, and therefore can fail for a general function f. In their most naive implementation, autoencoders are equivalent to standard PCA. However, they can be modified to also consider output labels, though no convergence guarantees can be made and the results are purely heuristic. Other feature weighting techniques have been studied for clustering and generic learning algorithms (Modha and Spangler 2003, Wettschereck et al. 1997), and in general a lot of work has been done on feature selection given a model (Li et al. 2017, Tenenbaum et al. 2000). Those survey papers extensively

[1] The model may be a multivariable linear or quadratic model, possibly with interaction terms.

[2] Most commonly fits are performed via least-squares approximation.

cover existing techniques and approaches for weighting (continuous valued) and selecting (binary operation) features of data for making accurate approximations.

2.2 Approximation Techniques

Four distance based approximation techniques are used to validate the proposed dimension reduction technique and additionally a classic neural network is tested. The first approximation algorithm is k-nearest neighbor using the 2-norm distance with $k = 1$, and the second is the same method with $k = 10$. By sorting points in a tree, k-nearest neighbors is able to scale linearly with the input dimension and near logarithmically with respect to number of input points. The second technique is the modified linear Shepard's method called LSHEP (Thacker et al. 2010). LSHEP incorporates a local linear fit into the the original Shepard's method (Shepard 1968), an inverse distance weighting technique that scales linearly with dimension and linearly with number of points. The final distance based technique uses the Delaunay triangulation, a piecewise linear interpolant based on a simplicial mesh of the same name (Chang et al. 2018). Of the above techniques, Delaunay is the most computationally expensive, scaling approximately linearly with number of points, but growing prohibitively expensive in dimension greater than 50. The last technique for general comparison is a classic neural network called a Multi-Layer Perceptron (MLP). This work uses the implementation available at (Pedregosa et al. 2011) while choosing the Rectified Linear Unit (ReLU) activation function and Stochastic Gradient Descent (SGD) error minimizer. 1000 gradient steps are allowed to be taken and the default number of hidden nodes (100) is used with one hidden layer.

3 METHODOLOGY

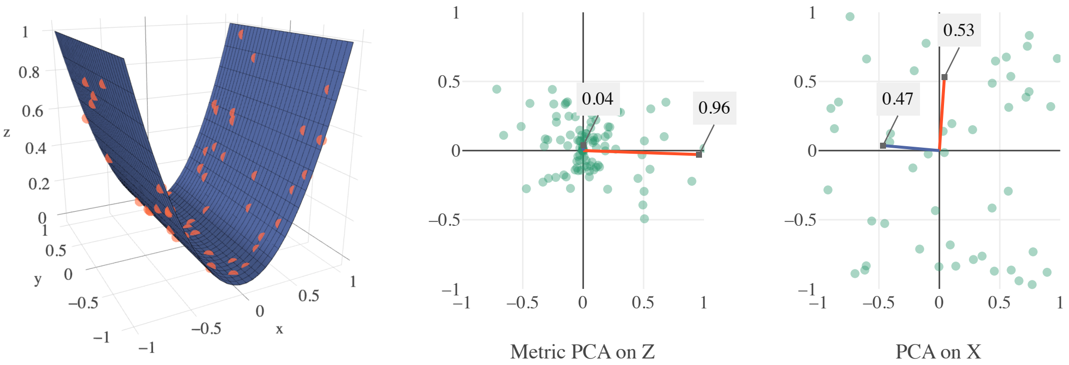
In this section, a novel application of Principle Component Analysis (PCA) is proposed in order to perform dimension reduction in a supervised learning (approximation) context. The method will be referred to as Metric PCA (MPCA).

Let $f : \mathbb{R}^d \to (S,m)$ be an arbitrary function of interest, where (S,m) is a metric space and d is a positive integer. Let $X \subset \mathbb{R}^d$ be a finite set of points with known function values (labels) $Y = \{y_i = f(x_i) | x_i \in X\}$. Consider the set Z of vectors

$$Z = \left\{ \frac{(x_i - x_j)\ m(y_i, y_j)}{||(x_i - x_j)||_2^2} : \quad x_i, x_j \in X, \quad y_i, y_j \in Y \right\},$$

with covariance matrix $Cov(Z)$. Note that since $Cov(Z)$ is symmetric, its Eigenvectors are orthogonal and form a basis for $\mathbb{R}^d$. Note that if the original set X contains n points, then the new set Z must contain $\mathscr{O}(n^2)$ points, since it consists of all vectors *between* points in X, rescaled by change in their corresponding function values (in Y).

An Eigenvector decomposition is performed on $Cov(Z)$, then the points are reduced into the lower-dimensional space spanned by the first k Eigenvectors ranked by the magnitude of their corresponding Eigenvalues (i.e., PCA on the new set Z). A *magnitude* is assigned to each vector by computing the *total variation* of the function along each Eigenvector and normalizing the sum of magnitudes to be one. In practice, instead of performing an Eigenvector decomposition on $Cov(Z)$, a singular value decomposition is performed on the matrix $Z^\star$, whose rows are transposed vectors in Z. Then, $Z^\star = U \Sigma V^T$, and the columns of V are exactly the Eigenvectors of $Cov(Z)$.



Metric PCA on Z

PCA on X

Figure 1: Depicted above from left to right are source data for an approximation problem (red dots, on left) and a true underlying function (blue mesh, on left), the constructed point set Z with metric principle components (middle), and the source point set X with standard principle components (right). Notice that the MPCA components are ordered and weighted in proportion to their relevance to the true function, while the PCA components are unrelated to the underlying function.

3.1 Equivalence of PCA and MPCA Under Euclidean Distance Metric

In this section it is shown that in the special case where $m(y_i, y_j) = \|x_i - x_j\|_2$, MPCA is identical to PCA.

Given X as stated above, if the set of all pairwise difference vectors Z is considered, then $PCA(Z) = PCA(X)$. Denote Z_j to be the subset of Z about vertex $x_j \in X$, $Z_j = \{x_j - x_i : x_i \in X\}$. Consider the first principle component of X, $v_1 \in \mathbb{R}^d$. It is noted that the first principle component of Z is the same, and will follow that all remaining components are identical.

$$\begin{aligned}
\max_{\|v\|_2=1} \|Zv\|_2^2 &= \max_{\|v\|_2=1} \sum_{z\in Z} \langle z, v\rangle^2 = \max_{\|v\|_2=1} \sum_{Z_j \subset Z} \sum_{z \in Z_j} \langle z, v\rangle^2 \\
&\leq \sum_{Z_j \subset Z} \max_{\|v\|_2=1} \sum_{z\in Z_j} \langle z, v\rangle^2 = \sum_{Z_j\subset Z} \sum_{z\in Z_j} \langle z, v_1\rangle^2 \\
&= \sum_{z\in Z} \langle z, v_1\rangle^2 = \|Zv_1\|_2^2 \\
\implies & \max_{\|v\|_2=1} \|Zv\|_2 = \|Zv_1\|_2.
\end{aligned}$$

After removing the first component from each vector in Z, the same technique can be reapplied to find the second component. This methdology can be applied to the remaining components in an inductive fashion, to show total equivalence of the principle components for X and Z.

3.2 Analytic Demonstration

Consider the following example. In this, $X^{(50\ \times\ 2)}$ is a random sample of points drawn from the unit cube and the values come from a function $f : \mathbb{R}^2 \to \mathbb{R}$, defined as $f(x,y) = x^2$ and note that this only depends on the x direction. The results of applying PCA and MPCA to this problem can be seen in Figure 1. It is clear that the example set of points are not distributed according the direction of greatest change in the underlying function. This causes PCA to produce distinctly different vectors from MPCA, noting that the

Methods	PCA, MPCA
Approximators	KNN (1), KNN (10), LSHEP, Delaunay, MLP
Number of samples (for MPCA)	n, 10n
Fraction of original dimension (for reduction)	1/3, 1/10, 1/100

Table 1: All possible combinations of the above settings are considered in experiments. Four distance based techniques are tested as well as a classical neural network. Since MPCA requires the construction of a pairwise difference set of points, random samples of size n and $10n$ are drawn to prevent drastic increase in the amount of data.

standard principle components are not only irrelevant to approximation, they are ordered incorrectly in terms of importance. It should be noted that this symmetric underlying function would cause *supervised PCA* to fail, since no meaningful linear fit can be produced.

3.3 Evaluation

Given theoretical interest in MPCA, it is important to evaluate the performance of MPCA by applying it to real approximation problems. In this section, MPCA is applied to three such real problems, two image classification problems (MNIST, CIFAR10) and one regression problem (Yelp rating prediction). First, the performance of four common approximation algorithms applied to the raw training data is evaluated using four-fold cross validation. Then, the dimension is reduced to various percentages of the total dimension using both MPCA and PCA and the performance of each algorithm is reevaluated. Recall that when X contains n data points, the transformed set Z contains $\mathscr{O}(n^2)$ points. Therefore, MPCA requires the singular value decomposition of a matrix $Z^\star$ with d columns and $\mathscr{O}(n^2)$ rows. Since this is not computationally feasible when n is large, a random sample of vectors in Z can be taken. Throughout Section 4, all combinations of the values listed in Table 1 are tested.

4 RESULTS

Yelp: Yelp (2018) is a collection of 479 American-style restaurant ratings from Las Vegas. Most of the data is composed of categorical descriptors, there are 63 features. This is a regression problem, where the algorithm predicts the star rating of a restaurant on 0-5 scale with .5 intervals.

MNIST: LeCun, Yann and Cortes, Corinna and Burges, Christopher J.C. (2008) provide a collection 60,000 images that are randomly reduced to 10,000 black and white images, each with shape [28 x 28] having 784 channels. The data poses a classification problem with 10 unique classes, which are 10 digits handwritten.

CIFAR10: Krizhevsky and Hinton (2009) share a collection of 50,000 images that are randomly reduced to 10,000 color images, each with shape [32 x 32 x 3] having 3072 channels in total. The data poses a classification problem with 10 unique classes that are common objects seen in each image.

The first 12 components produced by MPCA for the image classification problems can be seen in Figures 2 and 3. Notice how the first components are drastically different even though both are computer vision classification tasks. This is expected, because recognizing the differences between handwritten digits and generic objects (with 3-dimensional orientations) are drastically different problems.

The prediction errors with varying amounts of dimension reduction by MPCA and PCA are presented in Figure 4. The results show that common real approximation problems seem to have data variance that aligns well with the underlying phenomenon being modeled. This result is somewhat expected, as data sets are

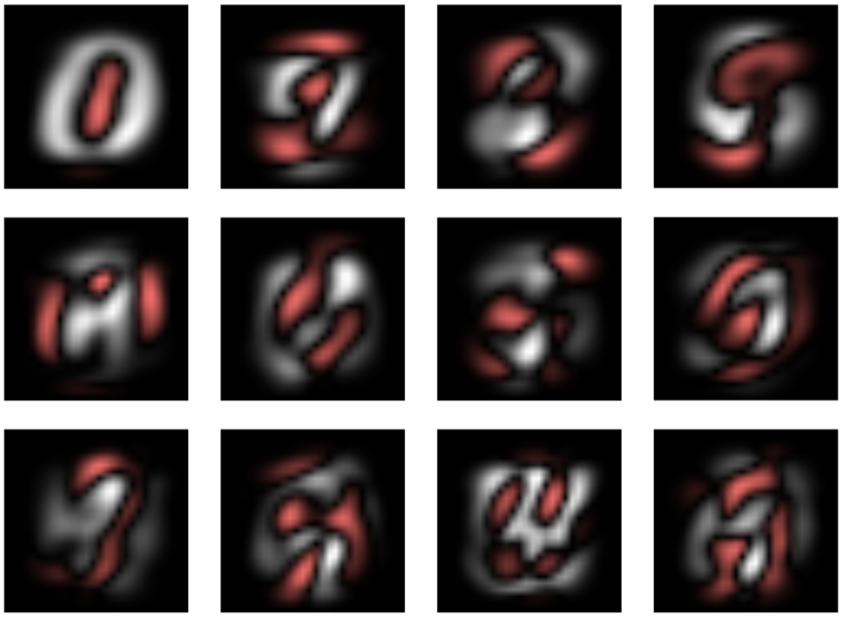


Figure 2: Above are the first 12 components produced by MPCA over the MNIST hand-written digit classification data set. The original MNIST data is in strictly black and white, these images are normalized to unit 2-norm for visualization purposes. The red coloring indicates large magnitude negative values, black is neutral, and white is large positive values. Notice the resemblance of these components to natural digits.

often collected in a way that assumes each piece of information is specifically relevant to the underlying phenomenon being approximated.

MPCA is used with varying levels of dimension reduction to improve KNN predictions in Table 2. For each approximation problem, a different amount of reduction produces the best approximation accuracy. The Yelp data requires the smallest subspace, MNIST is near the middle, and the CIFAR10 data requires the largest subspace that is not the raw images.

Finally, results combining all algorithms, data sets, and both PCA and MPCA are presented in Table 3. Notice that the empirical results demonstrate that MPCA is not guaranteed to be the best reduction technique even for strictly distance based prediction techniques. On the *Yelp* data, PCA still produces better results for 10-nearest neighbors. It is suspected that the worse performance of MPCA is due to the reduction on Z and the increased variability introduced by the random sampling and metric scaling.

These results should not be considered exhaustive, nor should these results devalue MPCA accordingly. In almost all cases the differences between the best PCA and MPCA predictors are less than 1%. Aside from performing similarly, MPCA has the added benefit of accommodating less careful data collection procedures.

5 DISCUSSION

The error results for reduced-dimension problems using PCA and Metric PCA (MPCA) are remarkably similar. In almost all cases the MPCA reduction causes an improvement in prediction accuracy for nearest neighbor predictions, though the improvement is small. Although an analytic example demonstrates the potentially large difference between MPCA and PCA, in practice the two appear to often be nearly equivalent.

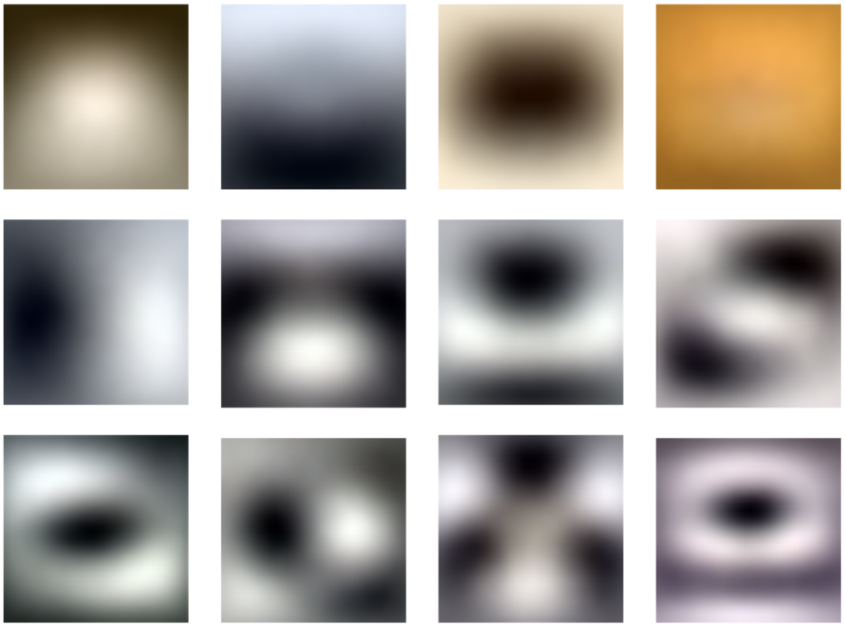


Figure 3: Above are the first 12 components produced by MPCA over the CIFAR10 general image classification data set. These images are normalized to unit 2-norm before visualization. Notice that the components strongly resemble the early terms in a Fourier decomposition, suggesting that low frequency spatial oscillations contain the most information in images, followed by more varied spatial patterns.

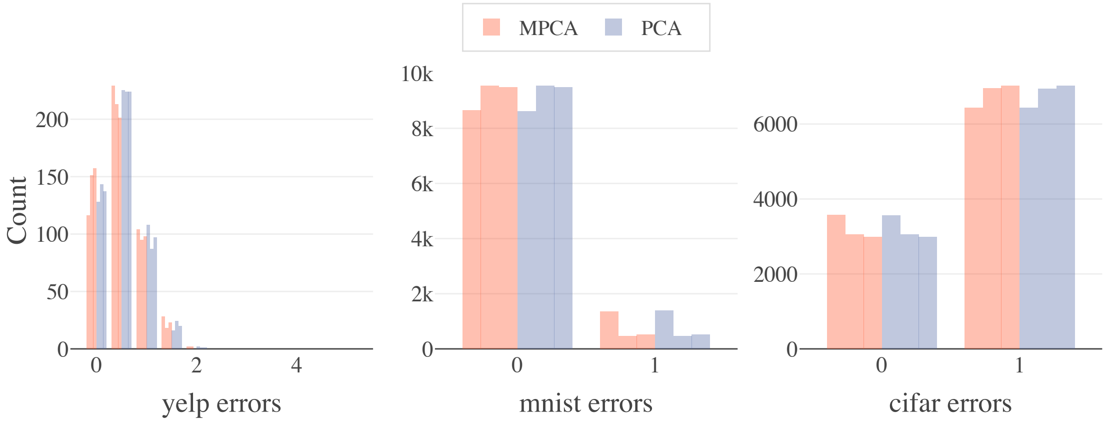


Figure 4: The prediction errors of KNN are computed via 4-fold cross validation. From left to right, the three bars within each group show the performance of KNN with $\{1/100, 1/10, 1/3\}$ of all features respectively. In general, MPCA and PCA perform similarly for all three prediction tasks.

Data Name	Raw data	1/3 MPCA components	1/10 MPCA components	1/100 MPCA components
Yelp 64 dim	0.493 (stars)	0.493 (stars)	0.496 (stars)	0.530 (stars)
MNIST 784 dim	5.29%	5.26%	4.63%	13.43%
CIFAR10 3072 dim	70.99%	70.23%	69.46%	62.46%

Table 2: This table shows the prediction accuracy of a KNN model when provided the raw data, and varying size subspaces from MPCA for the three real approximation problems. In each problem, the optimal size subspace varies. For Yelp, the smallest size subspace is best, while for CIFAR10, the largest subspace is best.

One could speculate that this is a contrived phenomenon, the only data that is kept in curated sets is data that represents a phenomenon of interest. MPCA is constructed to disregard data that is not useful, but for the data sets chosen in this work the underlying data is already distributed according to the functions being modeled.

6 CONCLUSION

The proposed technique, Metric PCA, demonstrates strong potential as a dimension reduction strategy for approximation problems. Analytically, Metric PCA is not susceptible to adverse data conditions that may cause PCA to disregard important dimensions for approximation. Empirically, the results obtained by MPCA and PCA are quite similar, suggesting that many curated data sets have the property that data variance corresponds closely to variance in the underlying function. Analytic and empirical results combined demonstrate the effectiveness of MPCA as a robust dimension reduction strategy for approximation.

REFERENCES

Barshan, E., A. Ghodsi, Z. Azimifar, and M. Z. Jahromi. 2011. “Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds”. *Pattern Recognition* vol. 44 (7), pp. 1357–1371.

Chang, T. H., L. T. Watson, T. C. Lux, B. Li, L. Xu, A. R. Butt, K. W. Cameron, and Y. Hong. 2018. “A polynomial time algorithm for multivariate interpolation in arbitrary dimension via the Delaunay triangulation”. In *Proceedings of the ACMSE 2018 Conference*, pp. 12. ACM.

Krizhevsky, A., and G. Hinton. 2009. “Learning multiple layers of features from tiny images”.

LeCun, Yann and Cortes, Corinna and Burges, Christopher J.C. 2008, Apr. “THE MNIST DATABASE”.

Li, J., K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu. 2017. “Feature selection: A data perspective”. *ACM Computing Surveys (CSUR)* vol. 50 (6), pp. 94.

Modha, D. S., and W. S. Spangler. 2003. “Feature weighting in k-means clustering”. *Machine learning* vol. 52 (3), pp. 217–237.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duches-

Data Set	**Approximator**	**Reduction Technique**	**Reduced Dimension**	**Lowest Mean Error**
Yelp	Delaunay	PCA	21	0.512
	KNN (1)	MPCA	6	0.610
	KNN (10)	PCA	6	0.493
	LSHEP	raw	63	0.678
	MLPRegressor	PCA	1	0.515
MNIST	Delaunay	PCA	8	0.132
	KNN (1)	MPCA	78	0.046
	KNN (10)	MPCA	78	0.051
	LSHEP	PCA	8	0.139
	MLPRegressor	PCA	261	0.133
CIFAR10	Delaunay	PCA	31	0.631
	KNN (1)	MPCA	31	0.681
	KNN (10)	MPCA	31	0.643
	LSHEP	PCA	31	0.650
	MLPRegressor	raw	3072	0.897

Table 3: In this expanded set of results, the optimal reduction technique and subspace size is presented for each algorithm and for each real approximation problem. Notice that MPCA is not the most common *best* reduction technique, which is counterintuitive. This suggests that the data provided for each approximation problem is in fact representative of the underlying phenomenon being modeled. It also suggests that the noise introduced in randomly sampling pairwise vectors for MPCA may inhibit its performance on approximation problems that are well-represented by available data.

nay. 2011. “Scikit-learn: Machine Learning in Python”. *Journal of Machine Learning Research* vol. 12, pp. 2825–2830.

Powell, Victor and Lehe, Lewis 2014, Oct. “Principal Component Analysis explained visually”.

Shepard, D. 1968. “A two-dimensional interpolation function for irregularly-spaced data”. In *Proceedings of the 1968 23rd ACM national conference*, pp. 517–524. ACM.

Shlens, J. 2014. “A tutorial on principal component analysis”. *arXiv preprint arXiv:1404.1100*.

Tenenbaum, J. B., V. De Silva, and J. C. Langford. 2000. “A global geometric framework for nonlinear dimensionality reduction”. *science* vol. 290 (5500), pp. 2319–2323.

Thacker, W. I., J. Zhang, L. T. Watson, J. B. Birch, M. A. Iyer, and M. W. Berry. 2010. “Algorithm 905: SHEPPACK: Modified Shepard algorithm for interpolation of scattered multivariate data”. *ACM Transactions on Mathematical Software (TOMS)* vol. 37 (3), pp. 34.

Wettschereck, D., D. W. Aha, and T. Mohri. 1997. “A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms”. *Artificial Intelligence Review* vol. 11 (1-5), pp. 273–314.

Yelp 2018, Nov. “American Style Restaurant Ratings in Las Vegas Nevada”. Online https://www.yelp.com/dataset.

AUTHOR BIOGRAPHIES

THOMAS C. H. LUX is a Ph.D. candidate in computer science at Virginia Polytechnic Institute and State University (VPI&SU) studying under Dr. Layne Watson. His research interests include computational science, approximation theory, optimization, and artificial intelligence. His email address is tchlux@vt.edu.

TYLER H. CHANG is a Ph.D. candidate at VPI&SU studying computer science under Dr. Layne Watson.