A Comparison of Dimensionality Reduction Methods for Large Data

ASHLEY BABJAC and SCOTT EMRICH, University of Tennessee, Knoxville, USA

CCS Concepts: • **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

Additional Key Words and Phrases: Autoencoders, LASSO, PCA, dimensionality reduction, SVM, modeling

ACM Reference Format:
Ashley Babjac and Scott Emrich. 2022. A Comparison of Dimensionality Reduction Methods for Large Data. 1, 1 (April 2022), 7 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Dimensionality reduction is a common task in most machine learning frameworks. When dealing with extremely large datasets it is imperative to reduce noise and other confounding factors that can either complicate or introduce error when training a model. In bioinformatics, this is traditionally approached using one of two methods: Principal Components Analysis (PCA) or LASSO (Least Absolute Shrinkage and Selection Operator). A primary reason for their general acceptance in the biology community is their ability to be easily interpreted.

More recently, deep learning methods such as autoencoders and transformers have been making their way into more bioinformatics applications. These models have been recognized as being able to extract signal from hard to model datasets using a wide variety of features including available metadata. One such example of this was using autoencoders for predicting microbiomes from environmental factors [3]. In another more recent example, Google's BERT transformer model was used to perform text analysis on DNA sequences [4]. We continue this trend by applying autoencoders for dimensionality reduction on different but also hard to model abundance matrices.

1.1 Related Work

Similar work has been done in trying to understand biological associations via dimensionality reduction. One such tool is MetAML which pairs feature selection (LASSO or Elastic Net) with machine learning classifiers (Support Vector Machines (SVM) and Random Forests (RF)) [8]. Their results show that combining feature selection with a machine learning classifier significantly improves performance for metagenomic prediction from microbiomes. The unnamed approach of [3] is similar to our approach in that they use autoencoders to develop an initial latent space based on species presence/absence; however, their only goal was to use environmental data as the only input for predicting the composition of root microbiomes. No "tradition" feature selection or ordination is considered, and only maize (corn) data is used to train and assess their framework. Mian [5] is a newly available web-based microbiome visualization

Authors' address: Ashley Babjac, ababjac@utk.edu; Scott Emrich, semrich@utk.edu, University of Tennessee, Knoxville, 2704 Kingston Pike, Knoxville, Tennessee, USA, 37919-4618.

Manuscript submitted to ACM

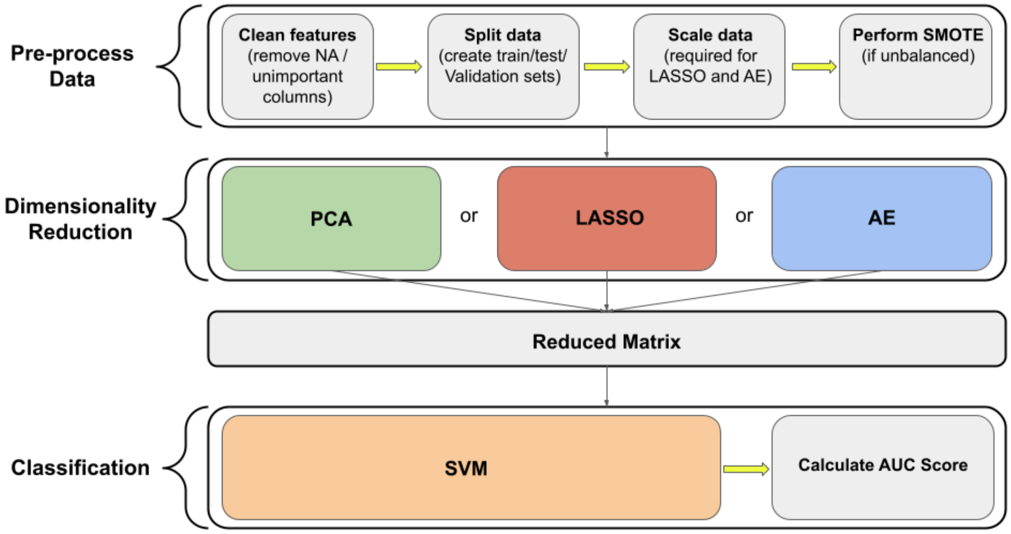
Pre-process Data
Clean features (remove NA / unimportant columns)
Split data (create train/test/ Validation sets)
Scale data (required for LASSO and AE)
Perform SMOTE (if unbalanced)
Dimensionality Reduction
PCA
or
LASSO
or
AE
Reduced Matrix
Classification
SVM
Calculate AUC Score

Fig. 1. Our model pipeline for computing desired labels via dimensionality reduction and SVM.

and machine learning platform. In addition to regression models used by [3] and similar classifiers, they provide customizable multi-layer perceptrons (MLPs). An MLP-based approach typically did worse on maize root microbiome data than an autoencoder-based approach and, in addition, Mian only uses simple feature selection methods such as univariate feature selection. In our prior work, we showed on one of the three datasets considered here that such feature selection does not work as well as LASSO-based feature selection [9].

1.2 Our Approach and Contributions

In this paper we explicitly compare pseudo-dimensionality reduction using autoencoders to the more traditionally accepted methods of PCA and LASSO. Specifically, we use an autoencoders' latent space representation in combination with classification algorithms, similar to how MetAML [8] pairs either elastic net or LASSO-based feature selection with classification. We compare the feature selection approaches with and without additional metadata, and assess the resulting prediction performance.

The primary contributions of this work are: (i) developing a model architecture for using autoencoders and SVM for improved classification performance across multiple large and sparse datasets with different use cases, (ii) using this model architecture to improve speed/memory consumption when performing prediction tasks on large data. We conclude based on these results that autoencoder-based dimensionality reduction outperforms previously used methods PCA and LASSO in terms of both classification and runtime.

2 MATERIALS AND METHODS

2.1 Datasets

We use three different datasets for comparing dimensionality reduction and classification. Each dataset consists of what we define as “metadata” (features that describe information about each sample), and “features” (the actual features for each sample. We will define the specifics for each of these data / use cases below.

The first dataset being analyzed comes from the TARA oceans project [10], and is comprised of 124 samples by 9,305 features, 36 of which are metadata. For these data, the metadata are variables about the ocean sample (i.e. depth, pressure, etc.) and the features are the observed gene abundances at each site grouped based on individual Kegg Ontology terms. We predict the ocean region where each sample was collected (9 ocean regions total) and refer to these data as “TARA” throughout. It is important to note that the TARA labels (ocean regions) are highly unbalanced. Since TARA has such a small number of samples we perform SMOTE (synthetic minority-oversampling technique) [1] in order to re-balance the data during the pre-processing step of our pipeline (see Architecture).

The second dataset comes from soil rhizosphere samples generated from a well-planned experimental treatment cite: this came back with a pretty minor revision. Can provide details (and will be referred to as “Rhizo”). It is made up of 62 samples by 13,822 features, 3 of which are treated as metadata (habitat, irrigation, marker_gene). The features themselves are sparse OTU tables which correspond to labels drawn from a focused experimental design where two different genotypes (with either high or low drought_tolerance, respectively) are grown in a common garden under two different irrigation schemes (normal or reduced). This is a more challenging prediction task than irrigation since the drought tolerance label is based on the host poplar tree and not each individual soil sample. We previously analyzed these data using LASSO in [9]. These should be challenging data for a deep learning-inspired framework as the time and resource-intensive nature of this experiment yielded at most 16 samples per genotype/treatment, e.g., drought tolerant with reduced irrigation. It should be ideal for considering metadata, however, due to the confounding factor of irrigation.

The final dataset is generated from GEM (Genomic Earth’s Microbiomes) project data [7]. Using a custom analysis pipeline, each microbiome-sourced genome/species was mapped to NCBI’s RefSeq collection – because all of these curated genomes must be culture-able this implies all matches can been reproduced at least one known environment. More simply, if it exists in RefSeq it is labeled “cultured”, and if it does not exist in culture it is labeled “uncultured.” There are 52,515 samples by 4,401 features. The features are broken up into 12 metadata that are taxonomy related, i.e., phylum, class, genus, species, taxonomic distance to a known cultured organism, 71 pathway features that represent coarse-grained COG abundances (aggregated into higher biochemical-pathways), and 4,318 annotation features which are fine-grained COG abundances. We predict the cultured/uncultured status as described above. Note that although we assume any genome with no representative is unculturable—and that is the default state for most organisms (get citation from Drew)—this is still an inference. A more correct label would be “unknown.” These GEM data are the ideal input for an autoencoder-based framework on prior work [3] because: (i) We have tens of thousands of samples (genomes); and (ii) more importantly, **there is no known model that can predict culurability available to microbial ecologists**. There have been prior efforts to associate microbiomes with diseases (e.g., [6, 8] or specific plants (e.g., [2]) but to the best of our knowledge no one has trained a model using the types of genes/pathways within specific organisms.

2.2 Models for Dimensionality Reduction

We evaluate three different models for dimensionality reduction: PCA, LASSO, and autoencoders.

PCA (principal component analysis) is a model that works by projecting data points into hyperplanes that best approximate the original data via least squares error. PCA can either project into a K dimensional space (i.e. create K components to represent the data), or it can account for K% of variability (by creating N components). For our model, we chose to account for 90% of variability based on preliminary testing. This amount of variability created a small enough number of components to be comparable while still accounting for the majority of variability in the data across all of our datasets.

LASSO is a modified form of linear regression that uses a cost function to tune the model based on an α hyperparameter. The lower the α parameter, the more LASSO begins to approximate linear regression, whereas higher α values tend to cause coefficients to trend towards 0. Unlike PCA and autoencoders, where we encode the entire dataset into a lower dimensional space, LASSO reduces (or increases) feature importance based on the determined regression co-efficients. For our model, we tested α values in range (1, 10, 0.1). As above this specific range was chosen based on preliminary testing where α values lower than 1 tended to not perform as well when the important coefficients were passed to SVM. The best value of α was dataset dependent and was used in each run to create the matrix passed to SVM based on 5-fold cross validation.

Autoencoders are a subset of neural networks that involve an input layer (referred to as the encoder) and an output layer (referred to as the decoder). Autoencoders, in general, condense the original input data down to a small number of layers (referred to as the latent space) and then try to recreate that input from the latent space representation. For our model, we use the latent space representation as the reduced matrix to pass to SVM. There are multiple hyperparameters that can be tuned for autoencoders. We find the best combination by varying the following parameters via grid search with 5-fold cross validation and training of 10 epochs: {latent dimensions: [10, 50, 100, 200], activation function: [relu, sigmoid, tanh], loss: [MAE, binary crossentropy], optimizer: [SGD, Adam]}. We use the best combination of parameters to encode the latent space for each dataset and pass this reduced representation to SVM.

2.3 Architecture

Dimensionality reduction does not perform classification; therefore, we pass the dimensionality reduced feature vectors through SVM to perform classification on our labels and assess the results. Figure 1 shows our model architecture. The specifics of our pipeline are as follows: (i) pre-process the data; this involves cleaning the data to remove any unknown values, calling SMOTE if required on unbalanced data, and scaling the data based on model requirements, (ii) perform dimensionality reduction using PCA, LASSO or autoencoders as described in section 2.2; at the end of this step a reduced matrix is created based on the specified algorithm, (iii) perform SVM based on the reduced matrix; this is called with probabilities set to true in order to compute the AUC and plot the ROC curve. Specifically, SVM is tuned with a grid search of the following parameters: {C value: [0.1, 1, 10, 100, 1000], gamma: [1, 0.1, 0.01, 0.001, 0.0001], kernel: [rbf, linear]}. We perform 5-fold cross validation and parallelize the SVM computation across 5 cores. We do note that this architecture could easily be applied to other models besides SVM in order to fit other regression or classification tasks. All of our code and figures from the full analysis are available on GitHub at ababjac/microbial-steen-project.

2.4 Metrics for Evaluation

To evaluate our models we look at both prediction performance and speed/memory consumption (reported as wall clock time and memory in Kilobytes) during runtime. To evaluate prediction we use AUC (area under the curve), which is a number between 0 and 1 calculated as the area underneath the ROC curve (showing true vs false positive rate across all classification thresholds). For balanced data, we would expect a minimum AUC of 0.5 if we predicted everything as true

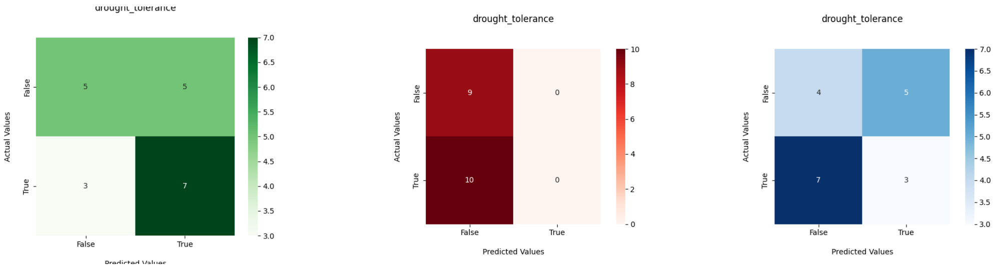

Fig. 2. Comparing PCA, LASSO, and Autoencoders results on SVM classification for feature vectors containing no metadata from GEM.

for a baseline comparison. Because it is known that the metadata features contain the bulk of important information correlated with the prediction labels, we perform the analysis described in Section 2.3 both including and excluding the metadata features for each dataset in order to compare dimensionality reduction methods when given data with low signal.

3 RESULTS

We typically expect models of biological features with metadata to perform better than models without metadata because these additional features often contain a large percentage of information which is highly correlated with the labels of interest. Table 1 shows the AUC scores across models with and without metadata for all combinations of dimensionality reduction techniques. As can be seen, models containing metadata typically do slightly better than models without (except Rhizo for autoencoders where there is an increase in performance). It is significant to note that when removing the metadata, autoencoders typically have the smallest decrease in performance (1-2 percentage points as compared to 5+ percentage points for other models). As stated previously, we actually see an increase in performance for Rhizo when using autoencoders from an AUC of 0.500 with metadata to 0.590 without. For GEM, we see a small decrease in AUC from 0.98 to 0.96, and for TARA similarly a microscopic decrease in AUC from 0.996 to 0.994. This marginally small decrease in performance shows that autoencoders are able to understand important relationships from just the data itself with no additional context.

Figure 2 shows the GEM data predictions for all models (note: right now this figure shows Rhizo because GEM is still re-running so I don't have the new figures yet. Will finish out this paragraph once I get final GEM results.

Additionally, we assess the timing and memory performance across all three dimensionality reduction techniques **in combination** with SVM. Table 2 shows the wall clock time in hours:minutes:seconds and well as the reported memory in Kilobytes. We can see that for small datasets (TARA and Rhizo) PCA by far performs the fastest, followed by LASSO, followed by autoencoders. This is because of the overhead introduced by LASSO and autoencoder grid search and cross-validation (which is not required for PCA). Autoencoders have far more grid search parameters than LASSO and must also train for multiple epochs in addition to the cross-validation which adds additional upfront compute time. However, autoencoders perform far better at scale when combined with SVM. This is likely due in part to the nature of autoencoders latent space in which the data gets reshaped into a new N-layer space. This smaller space is much more robust when passed to SVM which has an easier time separating the data for classification when compared to PCA/LASSO (kernel dependent).

	TARA			Rhizo			GEM		
DR Type	PCA	Lasso	AE	PCA	Lasso	AE	PCA	Lasso	AE
With Metadata	**0.997**	0.984	0.996	**0.530**	0.500	0.500	N/A	**1.000**	0.980
Without Metadata	0.990	0.976	**0.994**	0.470	0.500	**0.590**	N/A	0.700	**0.960**

Table 1. Table showing the AUC scores for each combination of dimensionality reduction and data. The TARA AUC score is averaged across the 9 site prediction scores.

	TARA			Rhizo			GEM		
DR Type	PCA	Lasso	AE	PCA	Lasso	AE	PCA	Lasso	AE
Timing	0:1:6.11	0:34:34.37	1:49:30.40	0:0:11.36	0:6:29.03	0:14:11.59	>1 week	5 days	3 days
Memory	0	0	0	0	0	0	N/A	N/A	N/A

Table 2. Table showing the timing and memory statistics averaged across 5 runs using our pipeline with no metadata. The timing is the elapsed time formatted as hours:minutes:seconds, and the memory is the memory consumption in Kilobytes as reported by /usr/bin/time on Linux.

4 DISCUSSION

One key distinction between our model and previous work is the non-reliance on metadata features to extract signal for prediction. Our model using autoencoders and SVM actually performs better with just the raw data in some cases than using combined metadata and features. In other cases, we see just a small reduction in performance accuracy where we see much larger reductions using other models. For GEM specifically, LASSO performs extremely well with the metadata. Upon further inspection of the features, we can see that two of the features “cultured.level” (the level at which something has been cultured) and “taxonomic.dist” (the distance from another species based on taxonomy) are highly correlated with the cultured/uncultured status label that we are predicting. LASSO basically can only pick out few important features to predict with making it highly dependent upon the information contained within the metadata. This explains the large decrease in performance (1.00 to 0.70 AUC) when the metadata is removed, because LASSO has trouble deciding which raw abundance vectors are important.

Conversely, autoencoders encode the information contained across all features into a small contained latent space representation. While this representation is not necessarily interpretable, it has been shown to contain more information in the reduced dimensionality than PCA or LASSO. This is shown in both Rhizo (increased AUC from 0.50 to 0.59) and GEM (small decrease in AUC from 0.98 to 0.96).

Similarly, PCA simply reduces down variability mathematically rather than learning the data into a condensed latent space. This works well for smaller datasets (TARA and Rhizo) when including the metadata because PCA is able to make multiple components based on the signal in the metadata combined with the feature vectors. However, PCA had significant trouble creating meaningful components without the metadata. For GEM, PCA reduced down the entire abundance matrix into 1 components (100% of variability).

In terms of the runtimes...

just starting to put some ideas down here...

5 CONCLUSION

ACKNOWLEDGMENTS

REFERENCES

[1] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16 (2002), 321–357.

[2] Zhiyu Deng, Jinming Zhang, Junya Li, and Xiujun Zhang. 2021. Application of Deep Learning in Plant–Microbiota Association Analysis. *Frontiers in Genetics* 12 (2021).

[3] Beatriz García-Jiménez, Jorge Muñoz, Sara Cabello, Joaquín Medina, and Mark D Wilkinson. 2021. Predicting microbiomes through a deep latent space. *Bioinformatics* 37, 10 (2021), 1444–1451.

[4] Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V Davuluri. 2021. DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome. *Bioinformatics* 37, 15 (02 2021), 2112–2120. https://doi.org/10.1093/bioinformatics/btab083 arXiv:https://academic.oup.com/bioinformatics/article-pdf/37/15/2112/39622303/btab083.pdf

[5] Boyang Tom Jin, Feng Xu, Raymond T Ng, and James C Hogg. 2021. Mian: interactive web-based microbiome data table visualization and machine learning platform. *Bioinformatics* 38, 4 (11 2021), 1176–1178.

[6] Seung Jae Lee and Mina Rho. 2022. Multimodal deep learning applied to classify healthy and disease states of human microbiome. *Scientific Reports* 12, 1 (2022), 824.

[7] Stephen Nayfach, Simon Roux, Rekha Seshadri, Daniel Udwary, Neha Varghese, Frederik Schulz, Dongying Wu, David Paez-Espino, I-Min Chen, Marcel Huntemann, et al. 2021. A genomic catalog of Earth's microbiomes. *Nature biotechnology* 39, 4 (2021), 499–509.

[8] Edoardo Pasolli, Duy Tin Truong, Faizan Malik, Levi Waldron, and Nicola Segata. 2016. Machine learning meta-analysis of large metagenomic datasets: tools and biological insights. *PLoS computational biology* 12, 7 (2016), e1004977.

[9] Owen Queen and Scott J. Emrich. 2021. LASSO-based feature selection for improved microbial and microbiome classification. In *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. 2301–2308.

[10] Shinichi Sunagawa, Silvia G Acinas, Peer Bork, Chris Bowler, Damien Eveillard, Gabriel Gorsky, Lionel Guidi, Daniele Iudicone, Eric Karsenti, Fabien Lombard, et al. 2020. Tara Oceans: towards global ocean ecosystems biology. *Nature Reviews Microbiology* 18, 8 (2020), 428–445.