How Does Noise Effect PCA

Some Interesting Results and Guesses at Why

Ethan Walker

BYU

Table of Contents

The General Idea

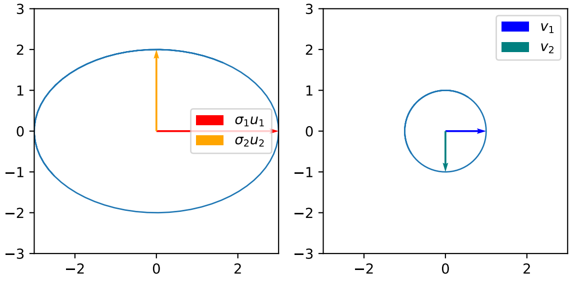
- What is Principal Component Analysis?
    - PCA is a dimension reduction method
- Why do we Use PCA
    - PCA can drastically simplify the computations required in machine learning by reducing the dimension of the data
- What is Noise?
    - When recording data there is always error, and variation in that error
    - This error is called noise

The general goal in this project is to examine how well PCA and PCA aided machine learning techniques perform when the data is perturbed by random noise of different types

PCA

The motivation behind PCA can be explained using the SVD. In the SVD, we order the singular values from largest to smallest and we arrange the singular vectors to align with those singular values.

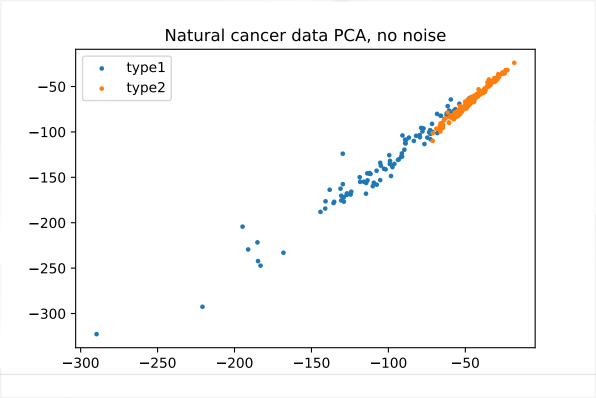
Recall that singular vectors can be thought of as orthonormal bases for a linear transformation. In this case, the singular vectors associated with a certain data set are the orthonormal bases that describe that data. Not only this, but the singular values are the weights given to each direction within that basis.



PCA

Because of these features of the SVD we can extract large amounts of information by taking the singular vectors associated with the largest singular values. This is, in a way, "projecting" the data into a subspace which holds a disproportionate amount of information.

This is especially useful when the singular values are very different in size, i.e. discarding the vectors associated with the smallest singular values causes minimal information loss.



The Breast Cancer Dataset

The dataset that I decided to use was the Breast Cancer Wisconsin Diagnostic Dataset. It was created in 1995 by Nick Street and is a standard toy dataset. This dataset is relatively high dimensional, as far as proof of concept goes, as it has 30 features; using PCA to reduce it to only 2 principal components will show how useful PCA is.

The dataset was created by examining the characteristics of cells of a breast mass. These include things like radius, texture, smoothness, etc. The different masses are categorized as either malignant or benign.

Noise

Certain types of noise make sense in certain contexts.

For this project I decided to add several different kinds of noise, each of which may apply in different ways. The types of noise I added are all based on different common continuous probability distribution functions, they are

Types of Noise

- Normal/Gaussian
  - This is the most common noise we expect. It is generally not a bad assumption to claim that some natural random process is normally distributed.
- Student-t
  - This distribution is used to estimate the mean of a normally distributed random variable. It is like the normal distribution with thicker tails.
- Beta
  - Defined on the interval [0,1].
  - Generally applies to random variables that have finite support.
  - It can take on many shapes within its finite support.
- Gamma
  - This is a PDF that is only defined on the positive reals, it in some ways a generalization of the Poisson distribution to continuous variables.
- Uniform

ML

There are many good machine learning classifiers, but some of my favorites, which include XGBoost and LightGMB, are somewhat computationally heavy. Instead I choose one that retains a strong interpretation of data being spatially distributed. K-Nearest Neighbors is a method that builds a sort of map using training data, then attempts to classify new data based on its position within this map. This method allows me to pull on the strengths of machine learning at scale, while also preserving the ability to analyze the spatial interpretations of data that PCA lends itself to.

I also used a random forest classifier, simply to provide a different perspective on the ML results.

K-Neighbors baseline score: 0.6175

Random Forest baseline score: 0.6175

Normal Results Random Forest

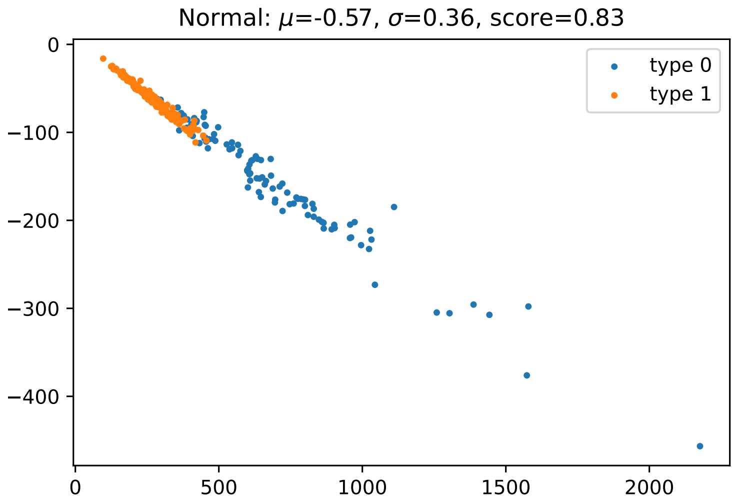
Best Results			Worst Results		
μ	σ	Score	μ	σ	Score
-0.43	0.36	0.923	-0.86	0.57	0.382
0.57	0.36	0.912	-0.86	0.71	0.382
-0.57	1.00	0.905	0.29	0.43	0.382
-0.14	0.21	0.905	0.43	1.00	0.372

Normal Results K-Neighbors

Best Results			Worst Results		
μ	σ	Score	μ	σ	Score
-0.86	0.36	0.919	-0.43	0.86	0.382
-0.71	0.29	0.916	0.71	0.86	0.382
-0.29	0.43	0.912	0.29	0.57	0.382
0.86	0.43	0.912	-0.29	0.93	0.382

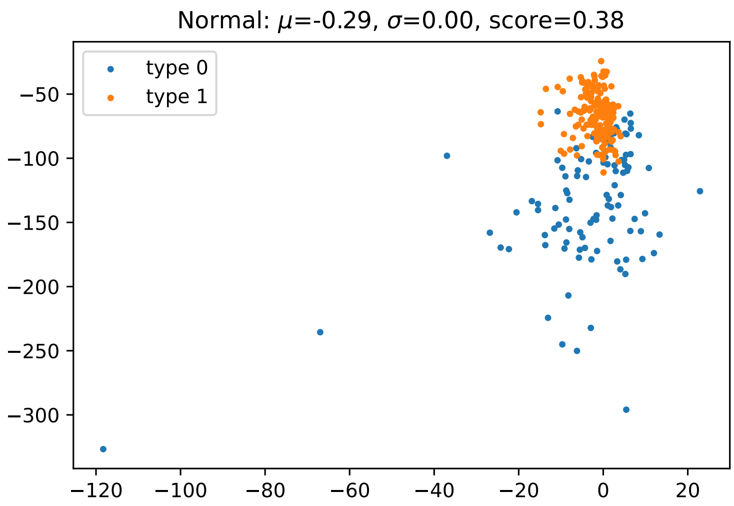
Normal Results PCA Image

Here are some examples of the transformed data



Normal Results PCA Image

Here are some examples of the transformed data



Student’s-t Results Random Forest

Best Results		Worst Results	
df	Score	df	Score
3.10	0.902	3.45	0.382
5.54	0.902	6.51	0.382
2.05	0.874	4.07	0.382
2.96	0.863	2.61	0.312

Student’s-t Results K-Neighbors

Best Results		Worst Results	
df	Score	df	Score
4.49	0.874	5.54	0.382
6.37	0.863	5.61	0.382
4.00	0.856	3.31	0.382
2.82	0.821	6.23	0.382

Beta Results Random Forest

Best Results			Worst Results		
α	β	Score	α	β	Score
3.95	3.60	0.919	1.85	0.45	0.382
3.25	0.45	0.912	2.90	2.55	0.382
0.45	1.50	0.909	3.60	3.25	0.382
1.50	1.85	0.895	2.90	2.90	0.112

Beta Results K-Neighbors

Best Results			Worst Results		
α	β	Score	α	β	Score
4.65	3.60	0.912	3.60	2.90	0.393
0.10	1.85	0.898	0.80	0.80	0.389
2.20	2.55	0.898	4.30	4.30	0.382
3.95	3.60	0.895	0.10	0.45	0.382

Gamma Results Random Forest

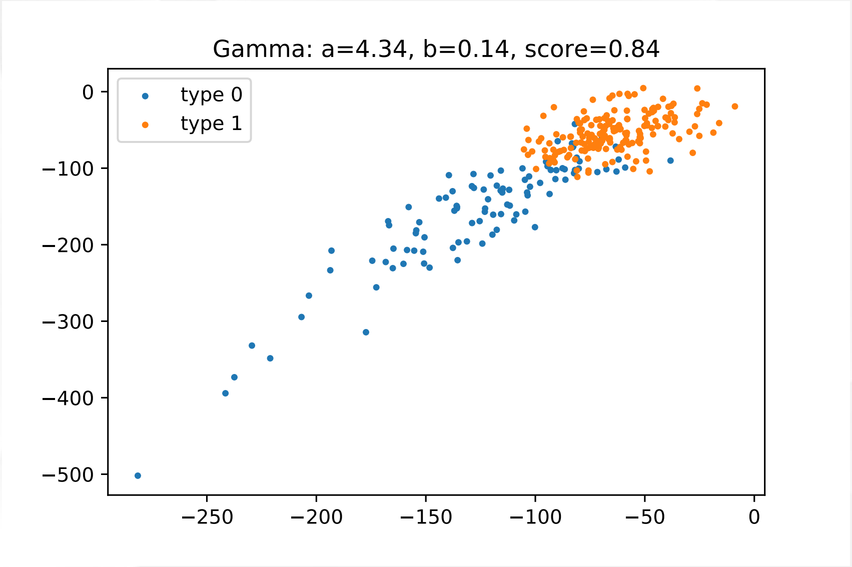
Best Results			Worst Results		
α	β	Score	α	β	Score
4.34	0.66	0.919	1.51	0.23	0.382
0.10	0.13	0.916	5.05	1.24	0.382
0.81	0.45	0.912	5.05	0.13	0.382
4.34	1.24	0.902	10.00	0.13	0.126

Gamma Results K-Neighbors

Best Results			Worst Results		
α	β	Score	α	β	Score
0.81	0.14	0.905	5.05	0.13	0.382
1.51	0.34	0.902	3.64	1.24	0.382
5.05	0.45	0.898	8.59	0.14	0.382
9.29	10.00	0.898	3.64	0.12	0.382

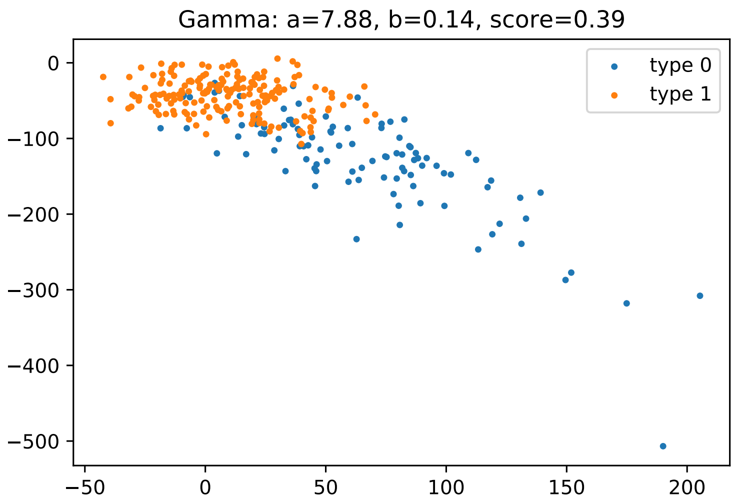
Gamma Results PCA Image

Here are some examples of the transformed data



Gamma Results PCA Image

Here are some examples of the transformed data



Uniform Results Random Forest

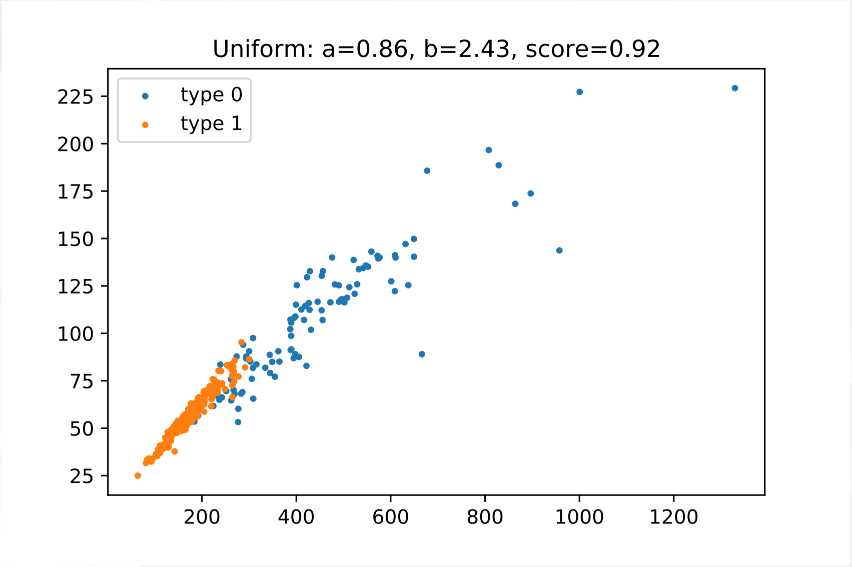
Best Results			Worst Results		
a	*b*	Score	*a*	*b*	Score
1.00	1.50	0.919	0.29	1.96	0.382
1.86	3.64	0.905	1.57	3.57	0.382
1.57	2.61	0.905	1.71	3.07	0.382
2.00	3.14	0.902	2.00	4.00	0.382

Uniform Results K-Neighbors

Best Results			Worst Results		
a	*b*	Score	*a*	*b*	Score
0.43	1.14	0.909	0.00	1.79	0.386
0.29	2.29	0.909	1.71	3.39	0.382
1.00	2.36	0.909	2.00	3.14	0.382
0.86	2.64	0.905	1.29	3.29	0.382

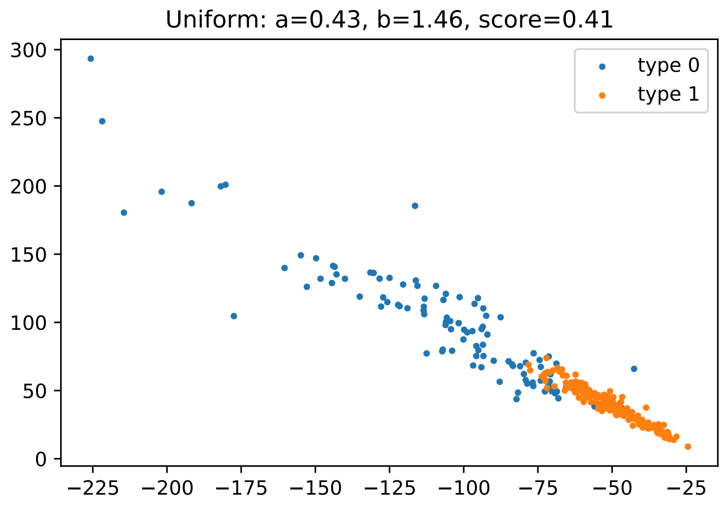
Uniform Results PCA Image

Here are some examples of the transformed data



Uniform Results PCA Image

Here are some examples of the transformed data



Conclusion

I learned a lot about PCA and testing in the project.

- Examining this stuff is hard
  - Be careful about how you write code
  - Be careful about what you think you can do
- I got way more data than I knew what to do with
  - Find good ways to manage data
  - Patterns are hard to find
- Be sure to work at scale, because sometimes things don't appear until you do them at scale
  - First I worked with the iris data set, but that was too low dimensional for PCA to make a difference
- There are things that we can't see that make a big difference
  - Sometimes noise really helped
  - Sometimes it makes things way worse
- I need to do a lot more work before I can make any kind of useful conclusions, but I did learn a lot
- What I didn't do but should have
  - More narrow
  - More Trials
  - Less Ambitious