Foundations + Context

Dimension Reduction

Dimension reduction is a common technique used for working with large data sets with the goal moving data from a high-dimensional feature space to a low-dimensional feature space while retaining key features of the original data. Approaches to dimension reduction can generally broken into one of two categories; feature selection and feature extraction. Feature selection techniques remove unneeded features from the data. For example, feature selection before building a regression model will often include remove predictors with low correlation to the response variable and predictors with high correlation. Feature extraction techniques will create a new set of features that capture relevant information about the original data. For example, a feature extraction technique may create linear combinations of our original predictors. It is important to note that feature selection will return a subset of variables from the original data set while feature extraction will return a set of new variables.

Curse of Dimensionality

At first, dimension reduction may seem like a counter intuitive approach to take when working with data. It seems that the more features we have, the more we should be able to learn about our data. However, when working with large datasets, the curse of dimensionality comes into play. The curse of dimensionality is a term used to refer to a variety of issues that working with high dimensional data presents. As the dimension of feature space increases, data tends to become increasingly sparse making trends in data more difficult to recognize. One way to think of how this problem manifests is to consider how the ratio of the volume of ball and the hypercube containing the ball changes as the dimension d increases.

The volume of a ball, B, of radius r in n dimensions is given by

$$\frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2}+1)}r^n$$

The volume of the hypercube,C, containing this ball (with sides of length $2r$) is given by

$$(2r)^n$$

Taking the ratio we have

$$\frac{\text{Vol}(B)}{\text{Vol}(C)} = \frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2}+1)\cdot 2^n} \text{ so } \lim_{n\to\infty}\frac{\text{Vol}(B)}{\text{Vol}(C)} = 0$$

Consider the case where $r = 1$,

```
library(gt)

ratio_df <- data.frame(n = c(1,2,5,10,25,100), Ratio = c('1', '0.7854', '0.1645', '2.49*e-3', '2.854e-1

gt(ratio_df)
```

n	Ratio
1	1
2	0.7854
5	0.1645

10	2.49*e-3
25	2.854e-11
100	1.868e-70

Thus, we can expect the number of points within distance 1 of a point will decrease as the dimension of data increases.

Another problem that arises from high dimensional data is computational time. The computation complexity of fitting a linear regression model to an $n \times p$ matrix of predictors is $O(np^2 + p^3)$ (many programs will not perform the full matrix inversion so the complexity is usually smaller). This will quickly balloon for large p. More troubling in high dimensions is that for p predictors, there are 2^p possible combinations of predictors making techniques such as *Best Subset Selection* computationally prohibitive.

Principal Component Analysis

Principal component analysis(PCA) is a commonly used unsupervised learning technique. PCA works by finding the principle components of a dataset, these can be thought of as the direction along which the data varies the most in the feature space. For those of you with a background in linear algebra, the principal components will be the eigenvectors of the covariance matrix in order of the norm of the eigenvalues.

Sections to add PCA +Linear Regression? PCR PLS L^1 Sparsification