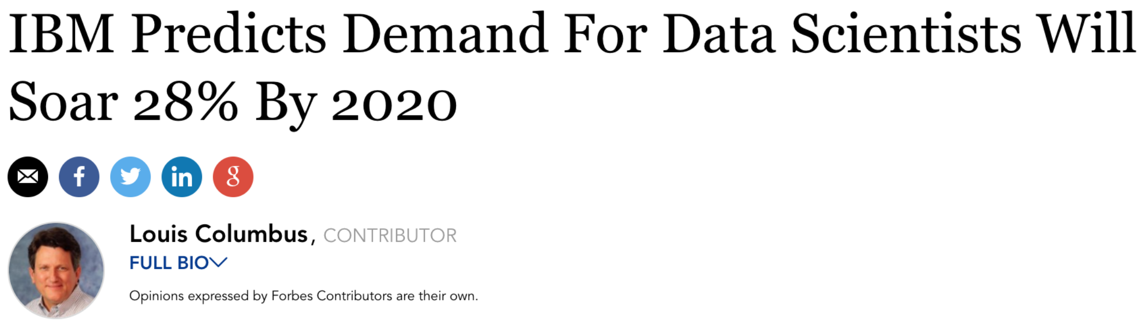
Post01 - Principal Component Analysis, Explained… Roughly

Jonathan Stuart

10/31/2017

Introduction

Clearly, Data Science is the new black. It's what everyone wants to do, and for good reason. Check out this headline from a recent article appearing on Forbes.com.



IBM Predicts Demand For Data Scientists Will Soar 28% By 2020

Louis Columbus, CONTRIBUTOR

FULL BIO

Opinions expressed by Forbes Contributors are their own.

alt text

The article relates some pretty exciting facts for those considering careers in data:

- Jobs that require *machine learning* pay an average of $114,000
- Annual demand for data scientists and engineers will reach 700,000 per year by 2020 (that's a LOT of new jobs per year)
- On average, data science jobs remain unfilled for 5 days longer than the national average.

Put those facts together and we see that Data Science is an area in which many new jobs paying high wages are rapidly opening up to a field of people not yet large enough to fill them all. Excellent news for someone working toward a degree that'll get their foot in the door. Or is it?

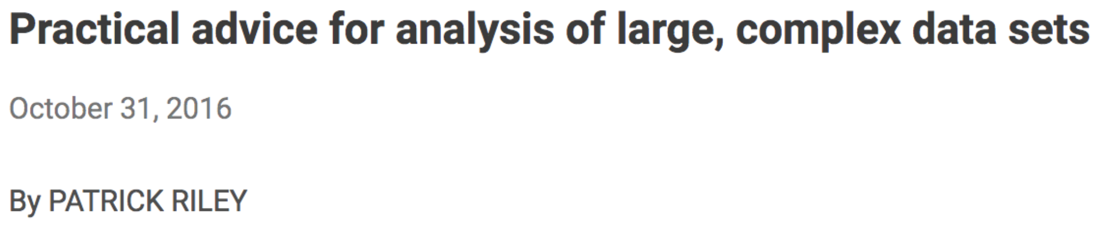
Being such an attractive area to work in, many, many people are planning to move into this arena. This begs the question: how does one become a competitive applicant? What skill set or knowledge base will make someone "right for the job," a preferred candidate? These questions, paired with an introductory exposure to Principal Component Analysis have formed the backbone of the research question for this blog post.

Motivation

To understand the motivation behind this post, let's first look at some comments from a Google employee on data analysis, then an example of how Principal Component Analysis might be useful, in practice.

Google Guy

Exactly a year ago, Patrick Riley of Google made the following blog post:



Practical advice for analysis of large, complex data sets

October 31, 2016

By PATRICK RILEY

In it, he talks about how in his 11 years of running Google's data science team for Google Search logs, there was one particular document he produced that, surprisingly to him, got the most reads. The document was titled "Good Data Analysis," and its subject bears directly upon our research question here: What skill set or knowledge base will make someone "right for the job" in the data science market?

Patrick broke the article into the following general areas:

- Technical: manipulation and examination of data.
- Process: Recommendations about how to approach a project, conceptually.
- Social: Communicating your insights about your data.

For our purposes, we're most interested in his comments on the technical aspects of data analysis. In Stat133, we've been learning quite a bit about how to manipulate and examine data. Prof. Sanchez has emphasized a few times that knowing your dataset is an important first step in the data analysis cycle. In his post, Patrick talks about some ways to above and beyond the simple examination of distributions by coming up with novel ways to slice your data, plot your data, and examine the interrelationships of subsets of your data. He goes briefly into some statistical concepts that might help one do this, and it is here where a discussion of Principal Component Analaysis fits nicely.

Principal Component Analysis: A First Pass

In order to understand the context in which PCA might be useful, let's consider a simple an example. Say you're a University Administrator. Being a University Administrator, you're going to be concerned about the long-term health of your particular institution. How to attract more students, how to attract more faculty and staff, how to retain students, faculty and staff and improve the quality of their experience, how to improve course offerings, how to improve educational-outcomes and post-graduate employment opportunities, among others, might all be questions that you would be interested in asking. Say your institution has 30,000 undergraduates, 11,000 graduates, 2,400 faculty members, and 130,000 staff members (including your top-notch National Laboratory).

(Berkeley, we're talking about Berkeley.)

Okay, so that's basically 200,000 people. Now let's think about the number of questions we'd have to ask each group of people in order to develop an accurate picture of their campus experience. Let's also keep in mind that some questions, like campus safety, etc., might apply to all three groups people, students, faculty and staff. We might ask the students about tuition costs and career goals, while we might be more concerned about what the faculty and staff think about retirement plans, etc. If we want to ask the students, say, 30 unique questions to get a sense of how to better serve them, the faculty 25 unique questions, the staff 20 unique questions, and on top of that as 10 questions that apply to all three groups, we're going end up with a 200000x85 matrix.

How would you then extract anything meaningful from that data set? Now remember…you're an administrator. Your livelihood rests absolutely on your determination of effective metrics of and plans to improve upon general satisfaction. Hmm. Wouldn't it be great if you could somehow take the 17,000,000 data points isolate those most significant?

Background

Categorically, we can look at where PCA fits more broadly into the discipline of statistical analysis. *Principal Component Analysis* is actually the simplest and most straight forward of a group of approaches that fall under the category of multivariate analysis. Multivariate analysis is the branch of statistics concerned with the simultaneous description of more than one random variable. This category of multivariate analysis also includes *factor analysis* and *canonical correlation analysis*. Briefly, *factor analysis* has to do with attempting to explain the variations observed among multiple variables by identifying and describing auxiliary, unobserved variables. Through the inference of what are called *latent variables*, descriptions of the observed variables can be achieved through descriptions of the auxiliary, *latent variables*. Canonical correlation analysis seems to be a bit more involved. It, briefly, has to do with creating whole vectors of correlated random variables and examining linear combinations of those variables with the goal being to maximize the correlation between linear combinations.

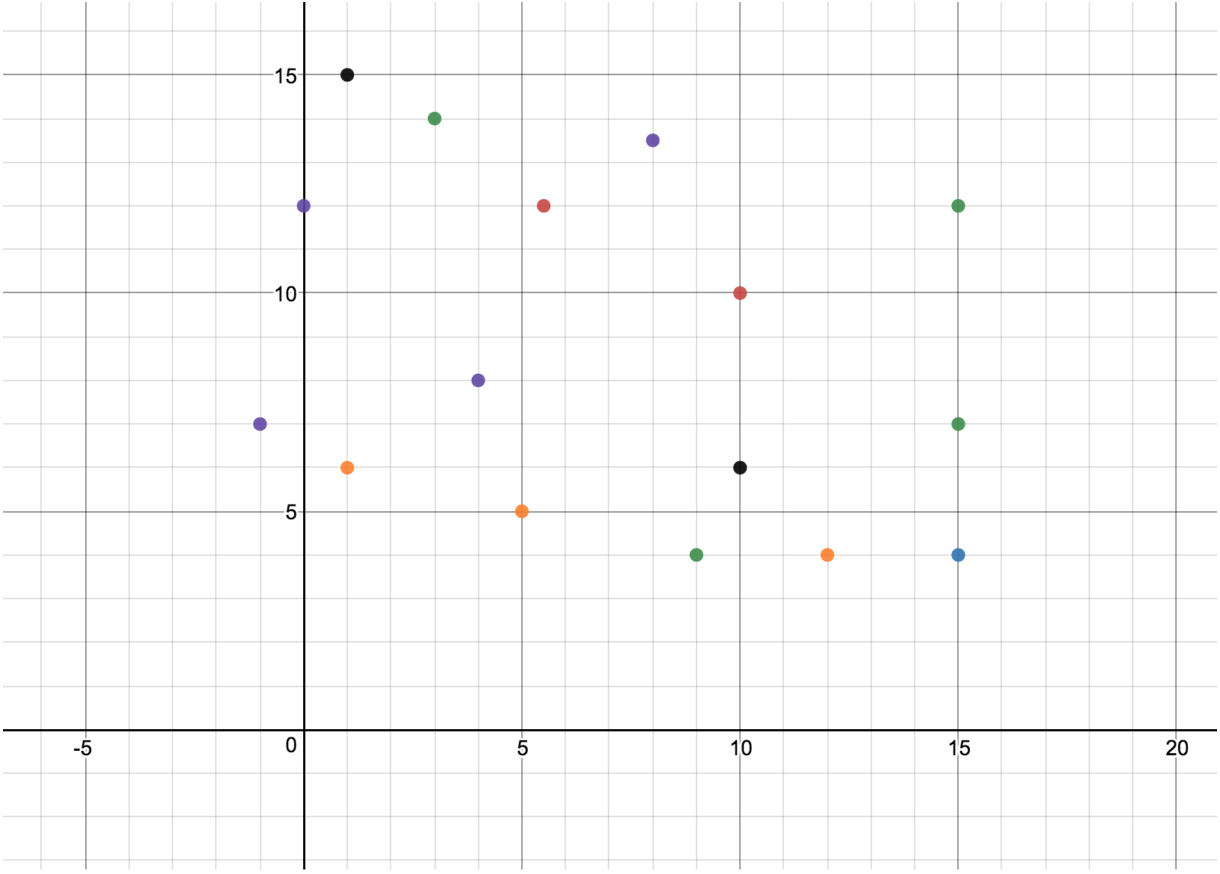
In general, principal component analysis, factor analysis and canonical correlation analysis all have, at the root, methods of interpreting variance and correlation. Variance, statistically, is a measurement of how far particular values in a set of values lie from the mean of all those values, while correlation measures the degree to which outcomes associated with one variable are related to outcomes of other variable or variables.

Discussion

One thing I kept coming across when reading about PCA was the idea of reducing the number of dimensions under consideration. Isn't that exactly what we want to do in our simple example of the university administrator, and what we wanted to do in HW03? How is it that PCA does this? Let's investigate.

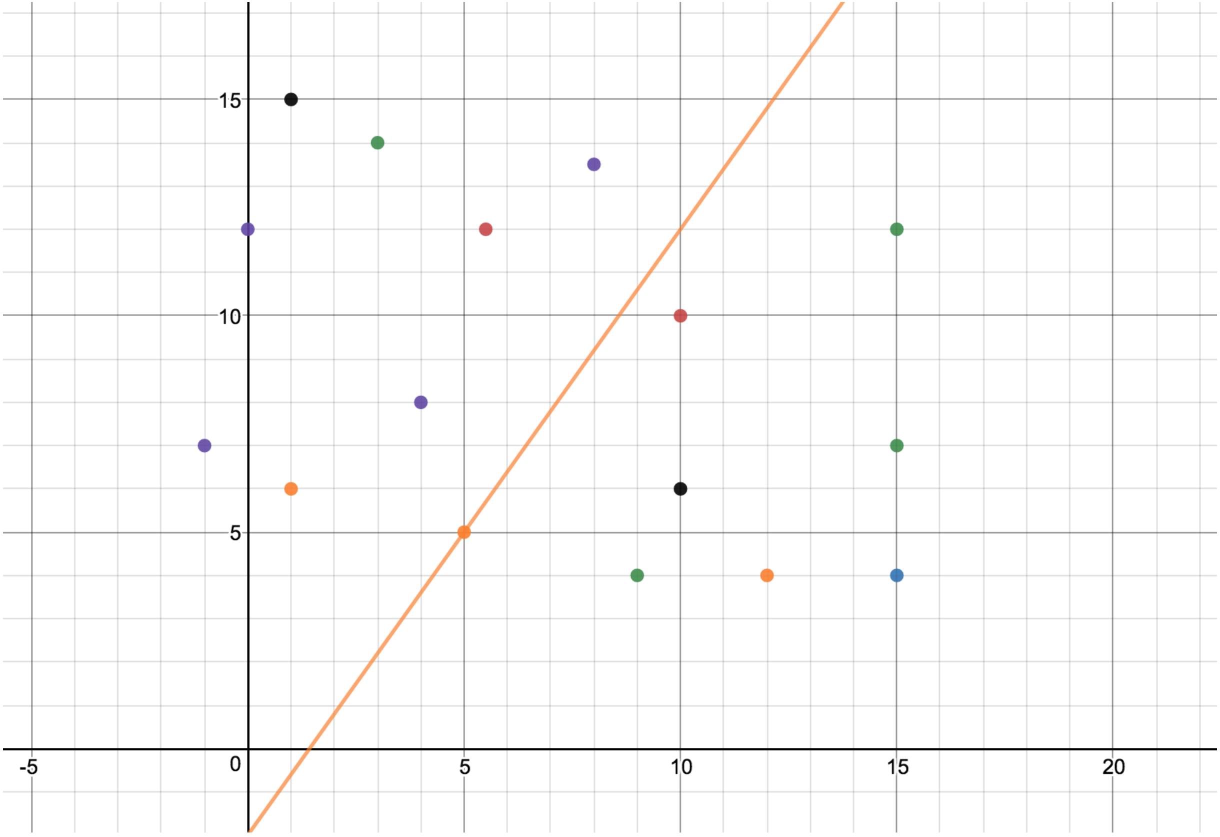
Principal Component Analysis: The Meat

So, without further ado, here's what's happening when performing a principal component analysis. Say we have a bunch of data points that end up looking something like this when graphed against the X, Y axes.

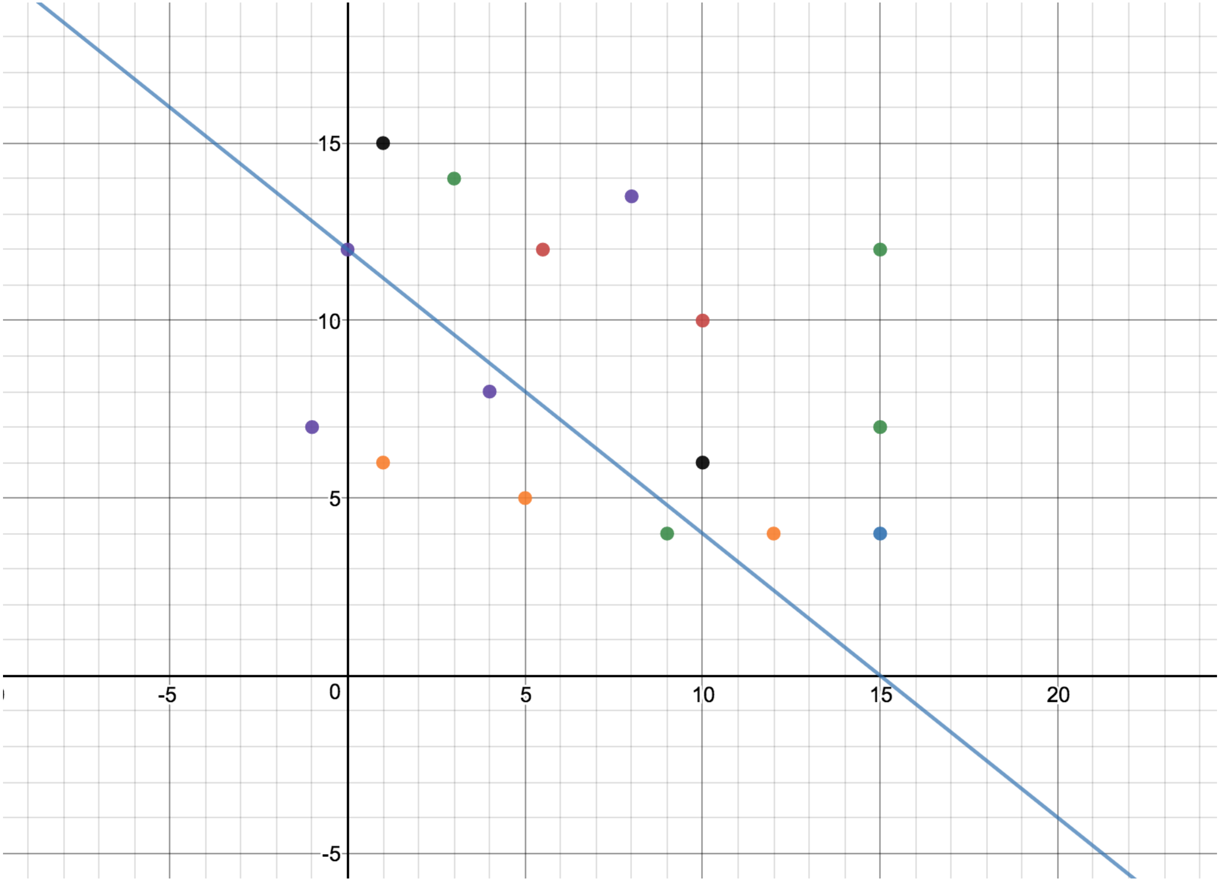


Given this set of points, we may want to devise some metric that best describes the most number of points, right? In our simple example above, this would be akin to coming up with one administrative policy that fits the needs of the largest number of individuals. In order to do that, we would need to devise a description for these points that captures the most amount of variation. In plain English, we would want to find a way to capture as many points as possible under a single description. Statistically, that means maximizing the variance. Graphically, that means maximizing the distance from as many points as possible to a particular line; we're trying to find a line where the points are most spread out.

Our first attempt might look something like this.



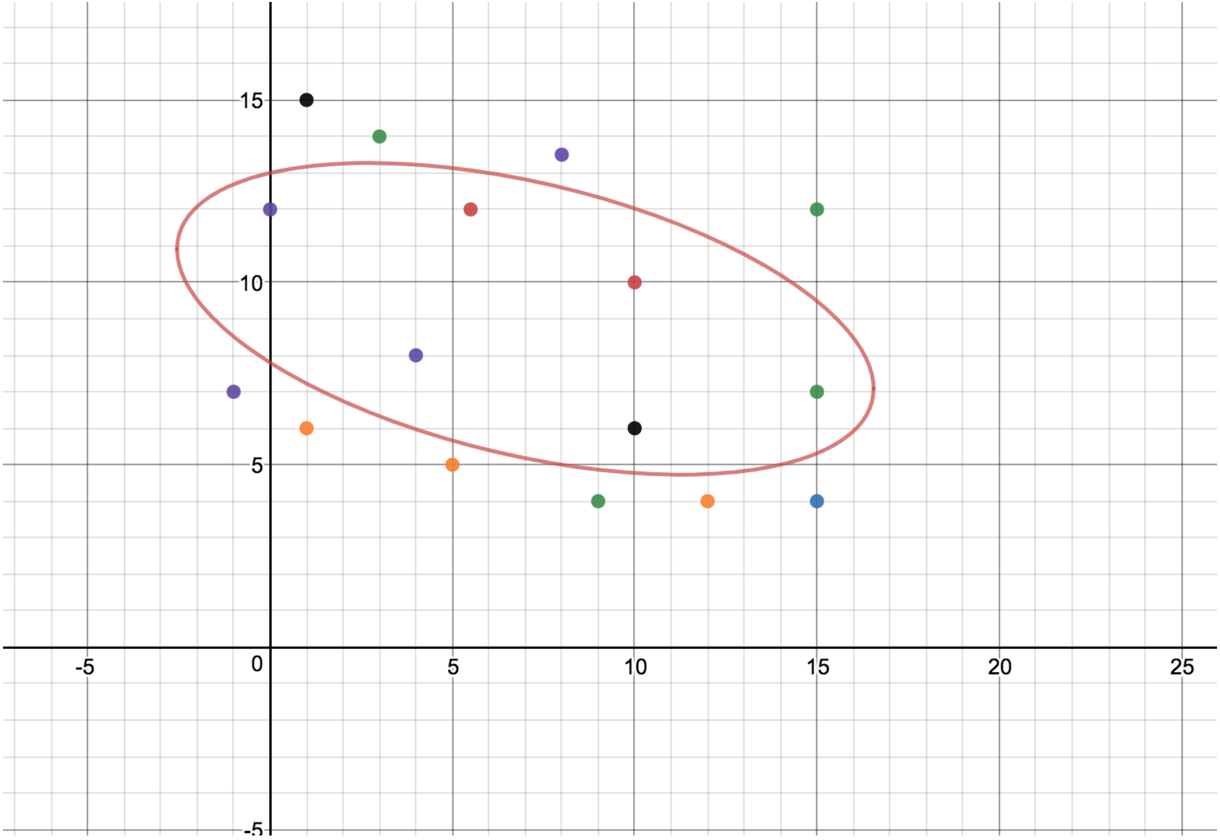
That's not bad, but some of the points are still kind of close to the line. Maybe we could try this?



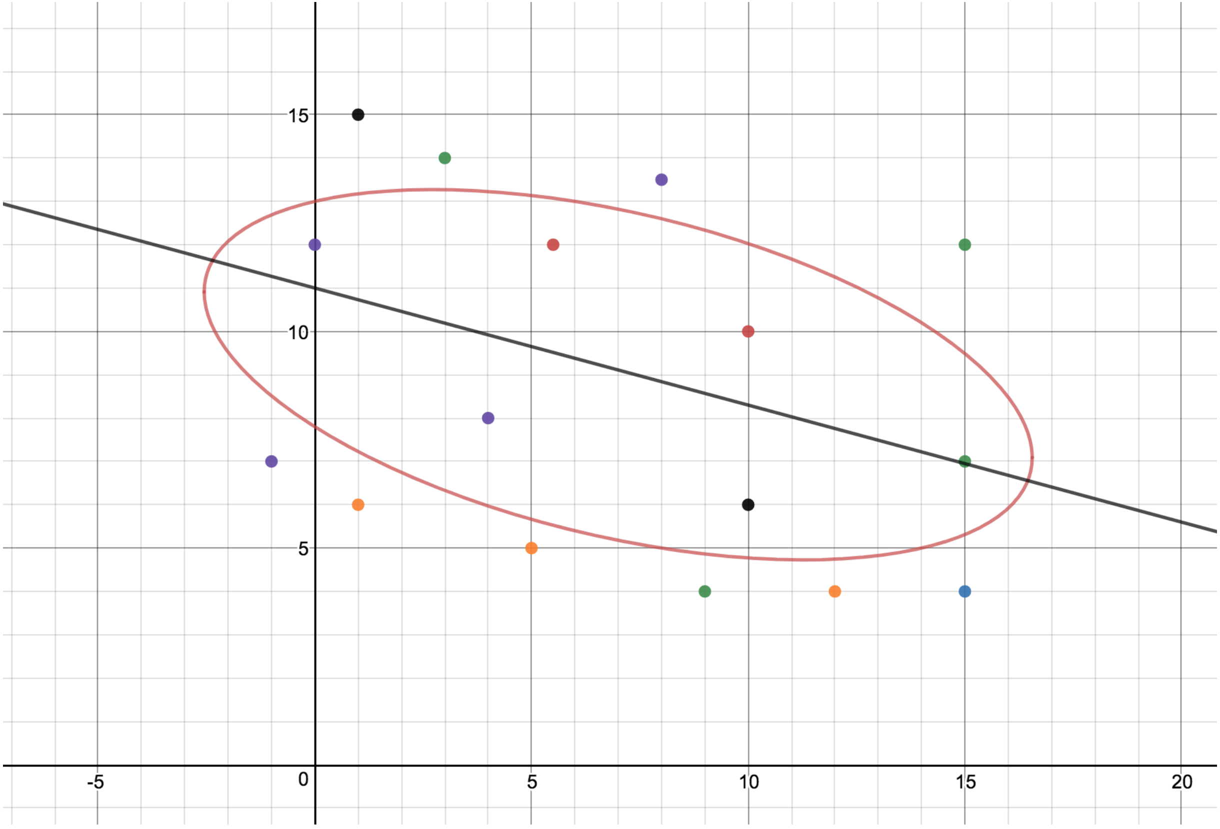
I guess we could go on and on and on like this, but there's actually a better way. This is where eigenvectors and eigenvalues come into play. An eigenvector is simply a vector that points in a particular direct. The corpus of linear algebra allows us to show that every eigenvector corresponds to a particular eigenvalue, and the statistical concepts allow us to show that eigenvalue associated with a particular eigenvector is a measurement of the amount of variation "captured" by a line drawn along the direction of that eigenvector. In the context of our graphical example, "capturing" variation has to do with maximizing the total distance of all the points from a given line, and in the context of our simple example, "capturing" variation has to do with formulating a policy to satisfy as many survey respondents as possible.

In this way, the principal component analysis built on eigenvector and eigenvalue analysis helps us to expose the underlying structure of a given data set. By first identifying the eigenvector that captures the most amount a variation among the data points, we can create the first of two new axes against which to examine the data. Let's see how.

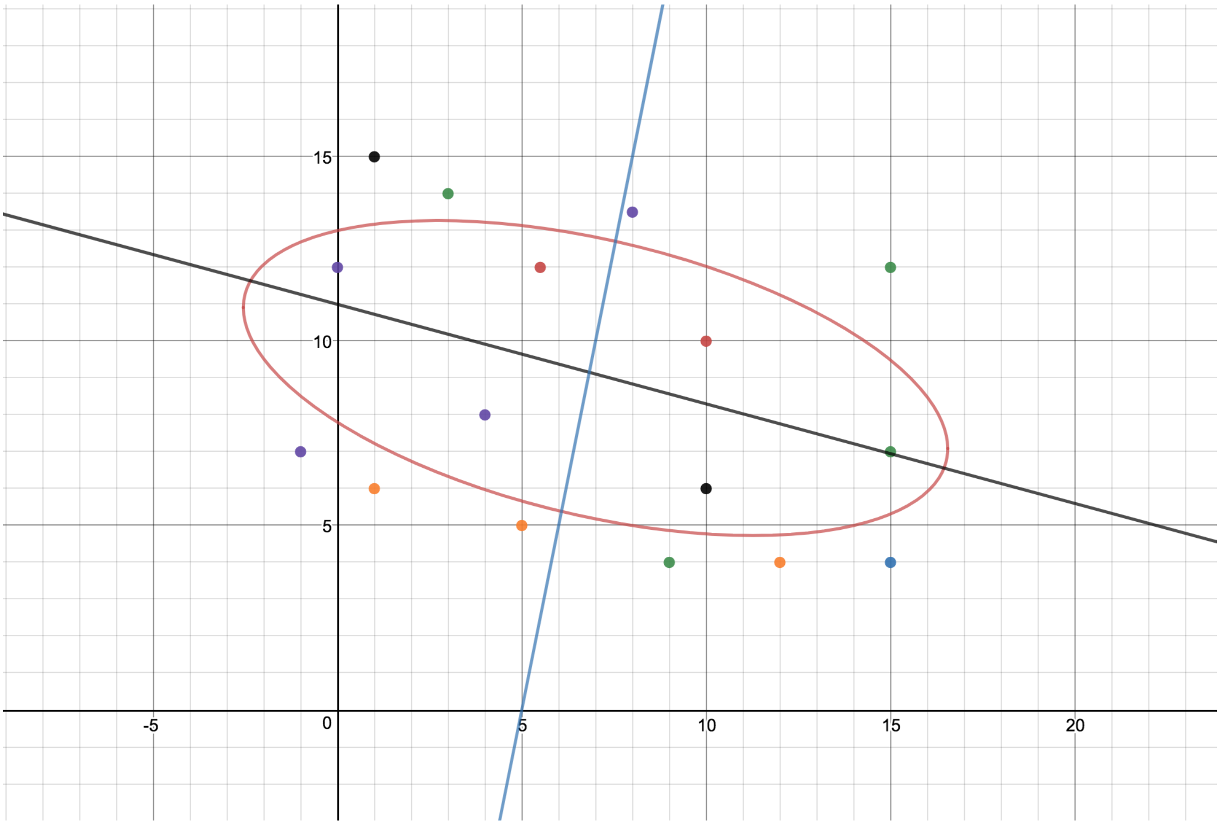
The following shows an ellipse fitted to the data, for visualization purposes.



Next, we have a line through, approximately, the major axis of the ellipse. This would be the line whose direction was in accord with the first eigenvector, and the eigenvalue would give us a value for variation.

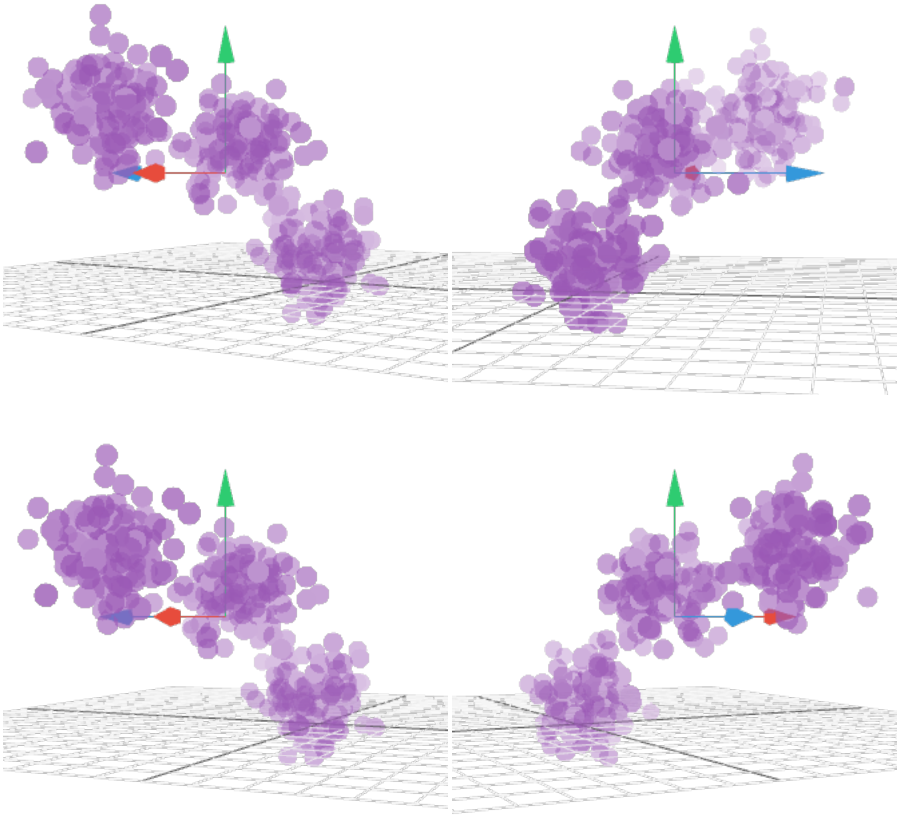


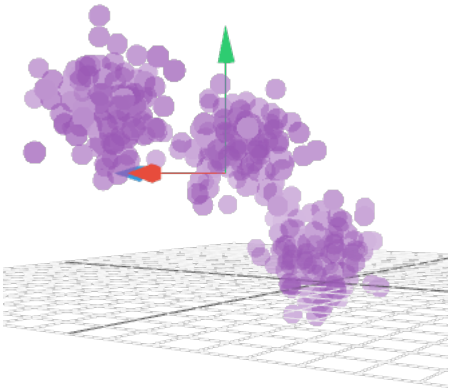
Now let’s look at the line moving in the direction of the second eigenvalue.



It’s no accident that the two lines are at right angles to each other. Being at right angels allows the first two principal components to capture the largest possible variation among all the points in the data set. These two lines also give us a new axis against which to view the data set. If we treated the black line as our new x-axis and the blue line as our new y-axis, we would now have a more useful axis to frame our data.

Now, return to the example of our administrator. The 45 questions would be akin to 45 dimensions. In the progression above, we started out with two. Looking at the following progression of images, try to form an idea of the complexity of a 3-dimensional data set.





Moving from a 3-d to a 45-d data set might be tough conceptually, but that's essentially what we've got to do when analyzing large data sets. Here, in this context, we see the use and value of PCA. Exactly as above, the first principal component will strike a line in the direction that "captures" the most variation between the data points, i.e. maximizes the distances from them to a line. The second principal component will strike a line orthogonal to the first, giving us the ability to re-frame a 45-dimensional data set more useful, in 2 dimensions.

HW03 - NBA Rankings

Let's briefly back at HW03. These are the variables we had to perform PCA on…there were 10 of them.

```
##  [1] "points3_made" "points2_made" "free_throws"  "off_rebounds"
##  [5] "def_rebounds" "assists"      "steals"       "blocks"      
##  [9] "turnovers"    "fouls"
```

The values for each of these variables for each of the NBA teams are stored in an R object, a dataframe, called `pca_frame`.

When we ran the following code,

```
pca <- prcomp(pca_frame, scale. = TRUE)
```

we relied on the built-in `prcomp()` function to calculate our eigenvectors and eigenvalues for us. Calling `prcomp()` and storing its output in `pca` created an R object that we then extracted values from to carry out our principal component analysis. Let's do a little looking around in the context of the examples given above.

```
names(pca)
```

```
## [1] "sdev"     "rotation" "center"   "scale"    "x"
```

These are the sets of values stored in `pca`. When doing HW03, we were most interested in the `rotation` list…it looks like this.

```
pca$rotation
```

```
##                    PC1         PC2         PC3          PC4         PC5
## points3_made 0.1121782 -0.65652993  0.28806873 -0.042637313  0.28657624
## points2_made 0.3601766  0.32892544 -0.06763180 -0.347710703 -0.15173866
## free_throws  0.3227564 -0.17651228  0.39157491  0.147596178 -0.21363792
## off_rebounds 0.3029366  0.35931603  0.33884845 -0.288483019 -0.16571824
## def_rebounds 0.3719432 -0.12808273  0.15026131 -0.492969442  0.26476256
## assists      0.3125312 -0.44134618 -0.26294129 -0.088066602 -0.36972525
## steals       0.3447256 -0.03540585 -0.48554101  0.177578661 -0.33549491
## blocks       0.3162237  0.06131890 -0.48869371  0.003935374  0.65459381
## turnovers    0.3353958 -0.02169833  0.08910421  0.532117541 -0.04471763
## fouls        0.3072548  0.28954426  0.26469871  0.454751471  0.26814214
##                      PC6         PC7         PC8         PC9         PC10
## points3_made -0.028435666  0.38167878  0.18027569 -0.20631322  0.409762462
## points2_made -0.088714347  0.07302430 -0.47216199 -0.35836740  0.499011524
## free_throws  -0.487342521 -0.62732220  0.07726675 -0.08283563 -0.006875686
## off_rebounds  0.283093235  0.13535335  0.64646479 -0.14735551 -0.124601143
## def_rebounds  0.066309015 -0.04926346 -0.23787252  0.64632050 -0.168579984
## assists       0.176019008  0.11785039 -0.18235775 -0.34086739 -0.547385461
## steals       -0.303664534  0.25883825  0.32703573  0.41596580  0.246739300
## blocks       -0.009954065 -0.30799231  0.23947533 -0.27071160 -0.057627209
## turnovers     0.675777660 -0.18850849 -0.14308362  0.13524769  0.250947823
## fouls        -0.298848473  0.47268121 -0.21462859 -0.04367200 -0.335087245
```

We calculated the eigenvalues like this,

```
eigenvals <- data.frame(
  eigenvalue = round(pca$sdev^2, 4)
)
eigenvals
```

```
##    eigenvalue
## 1      4.6959
## 2      1.7020
## 3      0.9795
## 4      0.7717
## 5      0.5341
## 6      0.4780
## 7      0.3822
## 8      0.2603
## 9      0.1336
## 10     0.0627
```

From this table, we found that, clearly, the eigenvalue associated with PC1 captures the most amount of variation within the data set; that's why that eigenvalue is the largest. The second largest is eigenvalue 2, corresponding to PC2.

Since the `rotation` portion of `pca` contains the eigenvectors, the following code allows us to examine the relative impact of each of the variables on the the first two principal components; the larger the contribution to the eigenvector, the more influential a categorical variable.

```
pca$rotation[,1:2]
```

```
##                    PC1         PC2
## points3_made 0.1121782 -0.65652993
## points2_made 0.3601766  0.32892544
## free_throws  0.3227564 -0.17651228
## off_rebounds 0.3029366  0.35931603
## def_rebounds 0.3719432 -0.12808273
## assists      0.3125312 -0.44134618
## steals       0.3447256 -0.03540585
## blocks       0.3162237  0.06131890
## turnovers    0.3353958 -0.02169833
## fouls        0.3072548  0.28954426
```

In concluding our discussion of PCA, we should key into the fact that "the system of variables with reduced dimension carries the main trends of the data and is easier to interpret and visualize than the original data. We may begin with a large number of variables; however, through PCA, we are able to represent most of the features of the data in a just a few variables." - https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3193798/. That is, essentially, roughly, the primary usefulness of Principal Component Analysis.

Conclusions

So what's the main takeaway? Yes, data science is cool, yes a bunch of people are rushing into it, and yes, there is a body of knowledge necessary to succeed in it. The good news is that we're working on acquiring that body of knowledge, even as we speak. Broadly, "PCA is a method of extracting information from data that keeps only what is most important and finds the underlying trends." - https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3193798/. It is simply the first of many statistical approaches to data representation and interpretation that we'll learn on the road to becoming data scientists.

PCA is broadly applicable…incredibly broadly. It has uses in Finance, Biology, Engineering, Neuroscience, tech usage/adoption rates, and basically anywhere you can find a data set that begs description. No matter the underlying subject of a data set, PCA is one of many tools we have available to help us understand it.

I included the piece from the Google Guy to further emphasize that, even after the degree is earned and the job is secured, we all have a long road ahead of us full of constant learning, and constant improvement of our skill sets and knowledge bases.

References

Here we have a list of the references used while putting together this post.

- Websites
    - Wikipedia Pages
        - https://en.wikipedia.org/wiki/Variance
        - https://en.wikipedia.org/wiki/Principal_component_analysis
    - Principal Component Analysis
        - http://setosa.io/ev/principal-component-analysis/
        - http://support.minitab.com/en-us/minitab/17/topic-library/modeling-statistics/multivariate/principal-components-and-factor-analysis/what-is-pca/
        - https://georgemdallas.wordpress.com/2013/10/30/principal-component-analysis-4-dummies-eigenvectors-eigenvalues-and-dimension-reduction/
    - Data Science
        - http://www.unofficialgoogledatascience.com/2016/10/practical-advice-for-analysis-of-large.html
        - https://www.forbes.com/sites/louiscolumbus/2017/05/13/ibm-predicts-demand-for-data-scientists-will-soar-28-by-2020/#27db518f7e3b
    - Journal Articles
        - https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3193798/