Introduction to *Urban* Data Science

Dimensionality Reduction

(EPA1316)

Lecture 12

Trivik Verma

Last Time

- The need to group data
- Geodemographic analysis
- Non-spatial clustering
- Regionalization

Today

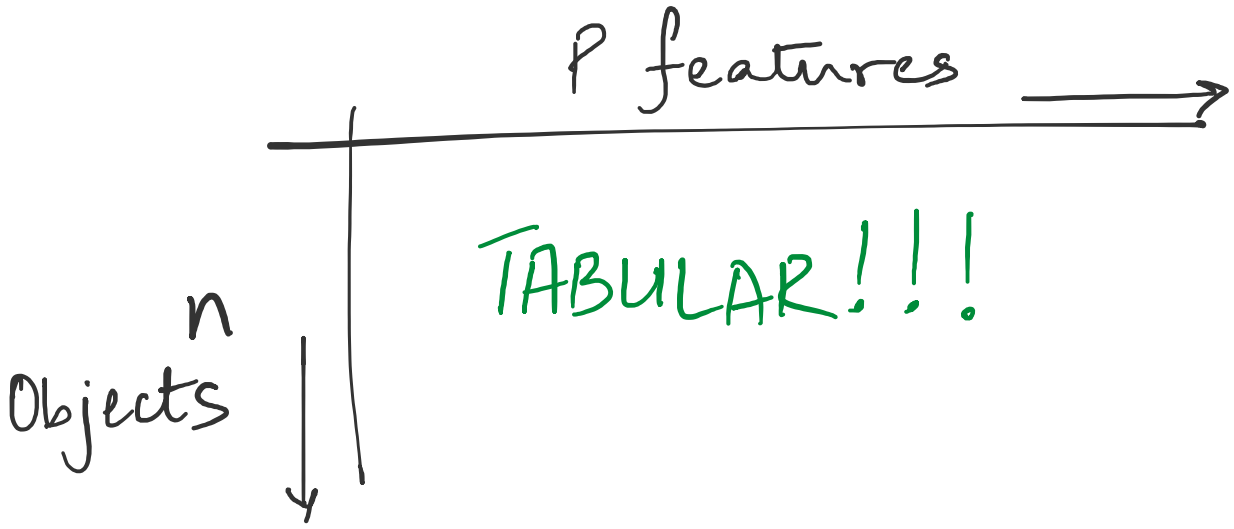
- Big Data and High Dimensionality
- A Framework For Dimensionality Reduction
- Principal Components Analysis (PCA)

Big Data and High Dimensionality

What is ‘Big Data’?

In the world of Data Science, the term *Big Data* gets thrown around a lot. We discussed this in Lecture 01-02

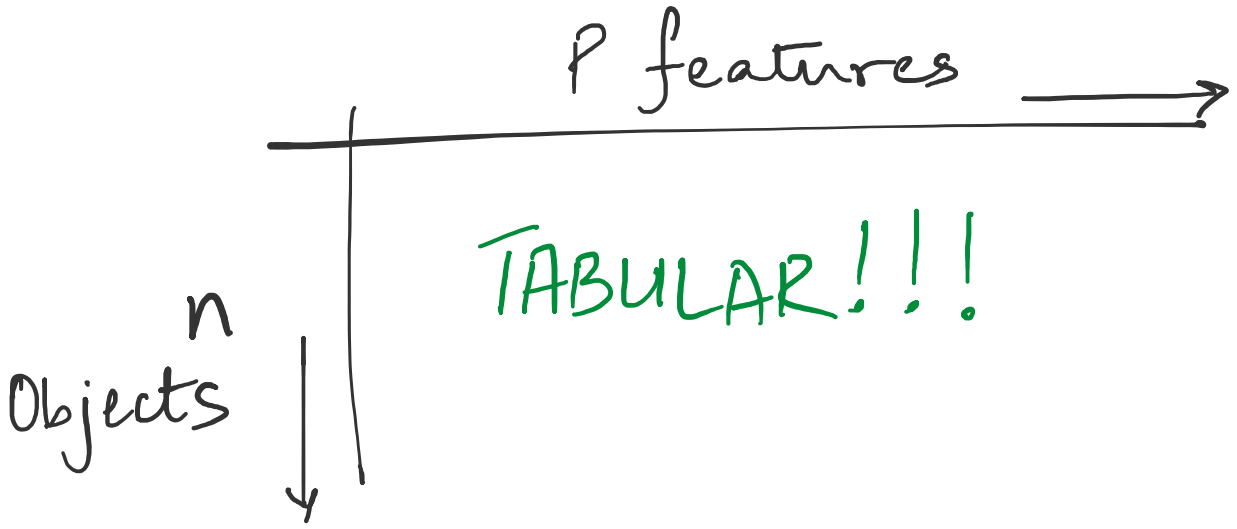
A rectangular data set has two dimensions: number of observations (*n*) and the number of predictors (*p*). Both can play a part in defining a problem as a *Big Data* problem.



What is ‘Big Data’?

What are some issues when:

- *n* is big (and *p* is small to moderate)?
- *p* is big (and *n* is small to moderate)?
- *n* and *p* are both big?



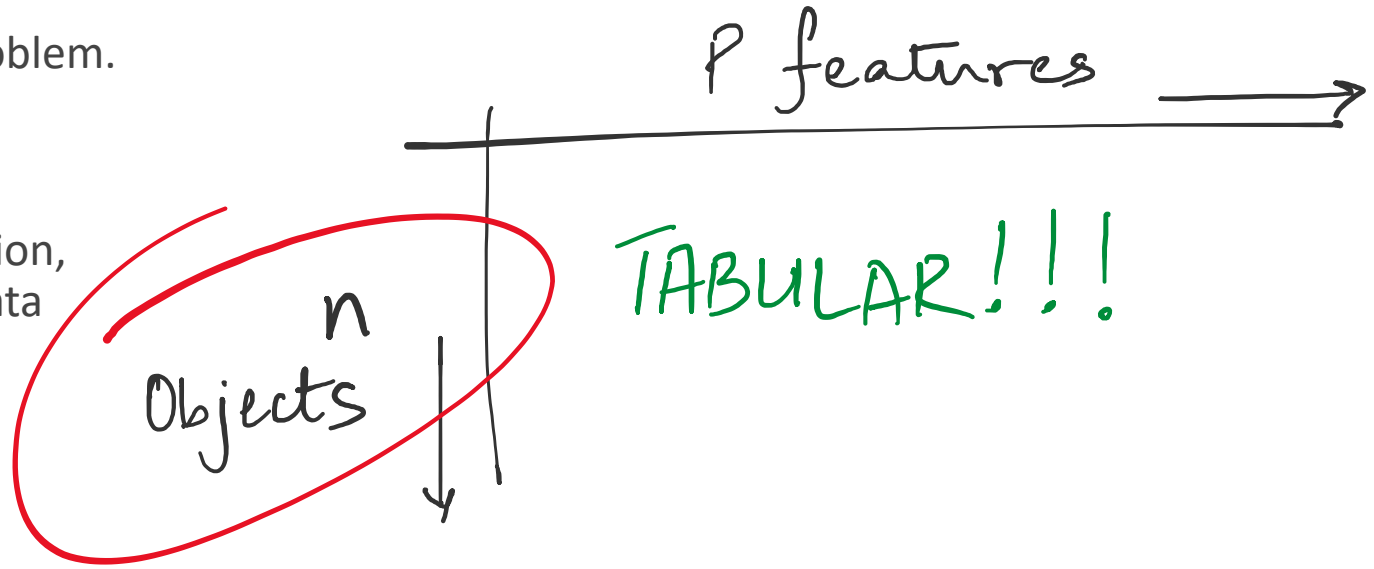
When *n* is big

When the sample size is large, this is typically not much of an issue from the statistical perspective, just one from the computational perspective.

- Algorithms can take forever to finish. Estimating the coefficients of a regression model, especially one that does not have closed form (like LASSO), can take a while.
- If you are tuning a parameter or choosing between models, this exacerbates the problem.

What can we do to fix this computational issue?

- Perform 'preliminary' steps (model selection, tuning, etc.) on a subset of the training data set. 10% or less can be justified



Keep in mind, big *n* doesn't solve everything

- The era of Big Data (aka, large *n*) can help us answer lots of interesting scientific and application-based questions, but it does not fix everything.

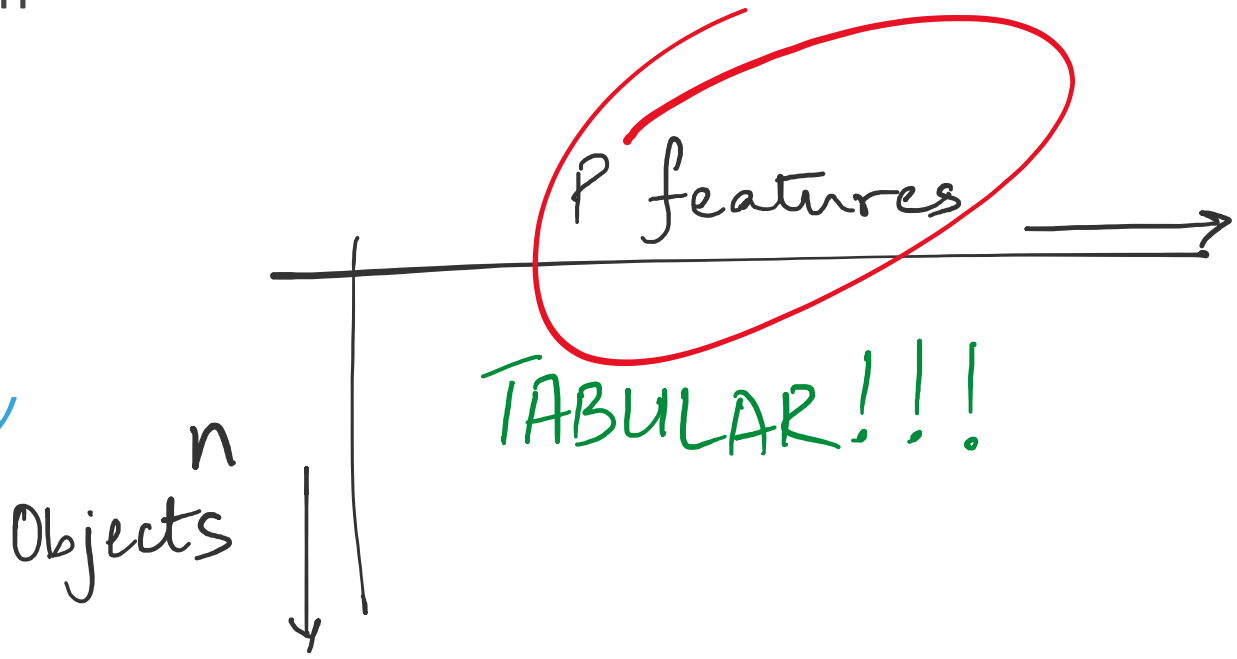
- Remember the adage: "**crap in = crap out**". If the data are not representative of the population, then modeling results can be terrible. Random sampling ensures representative data.

When p is big

When the number of predictors is large (in any form: interactions, polynomial terms, etc.), then lots of issues can occur.

- Matrices may not be invertible (issue in regression).
- Multicollinearity is likely to be present
- Models are susceptible to overfitting

This situation is called *High Dimensionality* and needs to be accounted for when performing data analysis and modeling.

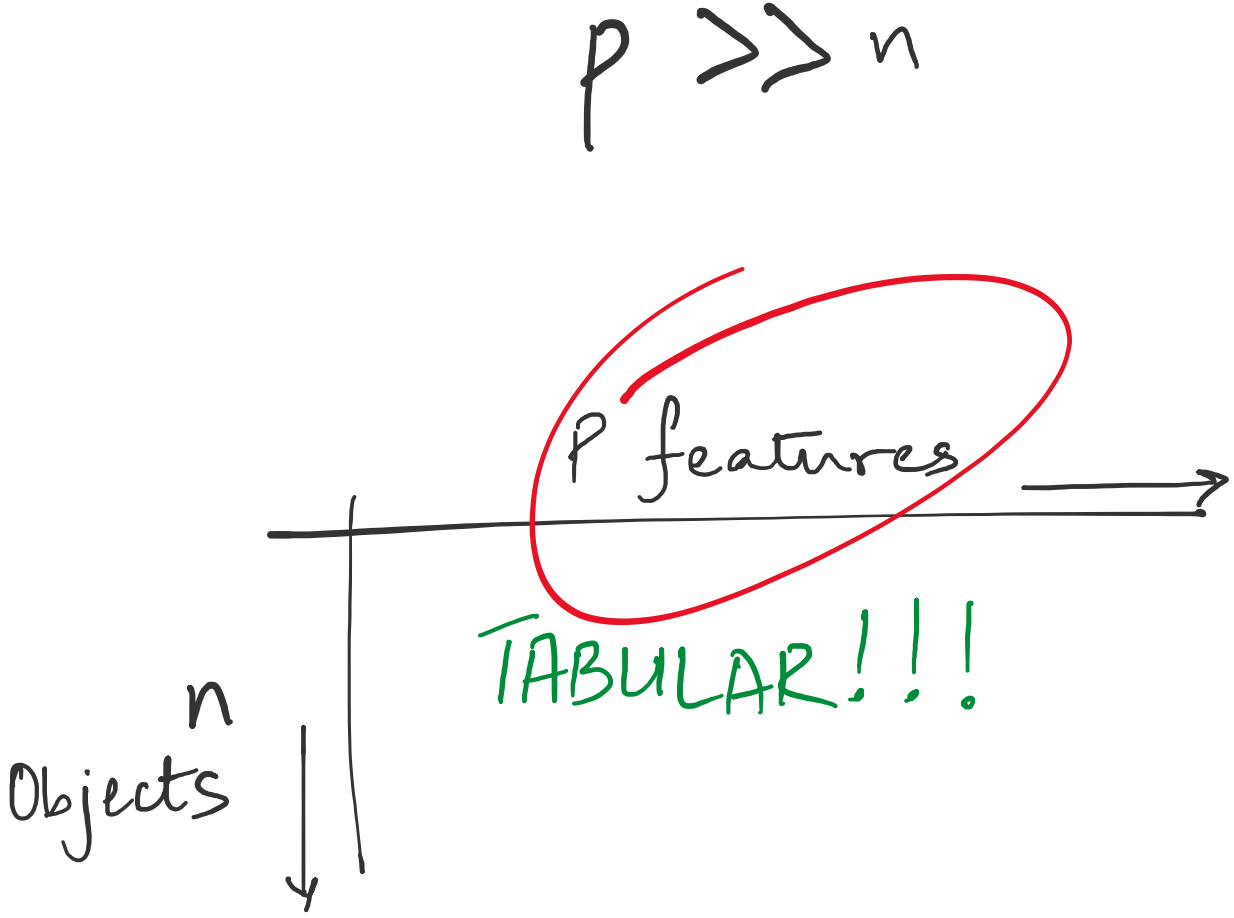


When Does High Dimensionality Occur?

This can occur when we consider lots of interaction terms. But this can also happen when the number of main effects is high.

<u>For example:</u>

- When we are performing polynomial regression with a high degree and many predictors.
- When the predictors are 155 census variables (and possible interactions) in a compounded urban problem.
- When the predictors are the counts of all English words appearing in a text.



A Framework For Dimensionality Reduction

One way to reduce the dimensions of the feature space is to create a new, smaller set of predictors by taking linear combinations of the original predictors.

- We choose $Z_1, Z_2, \ldots, Z_m$, where $m \leq p$ and where each Z_i is a linear combination of the original p predictors

$$Z_i = \sum_{j=1}^{p} \phi_{ji} X_j$$

- for fixed constants ϕ_{ji}. Then we can build a linear regression regression model using the new predictors

$$Y = \beta_0 + \beta_1 Z_1 + \cdots + \beta_m Z_m + \varepsilon$$

- Notice that this model has a smaller number (m+1 < p+1) of parameters.

A Framework For Dimensionality Reduction (cont.)

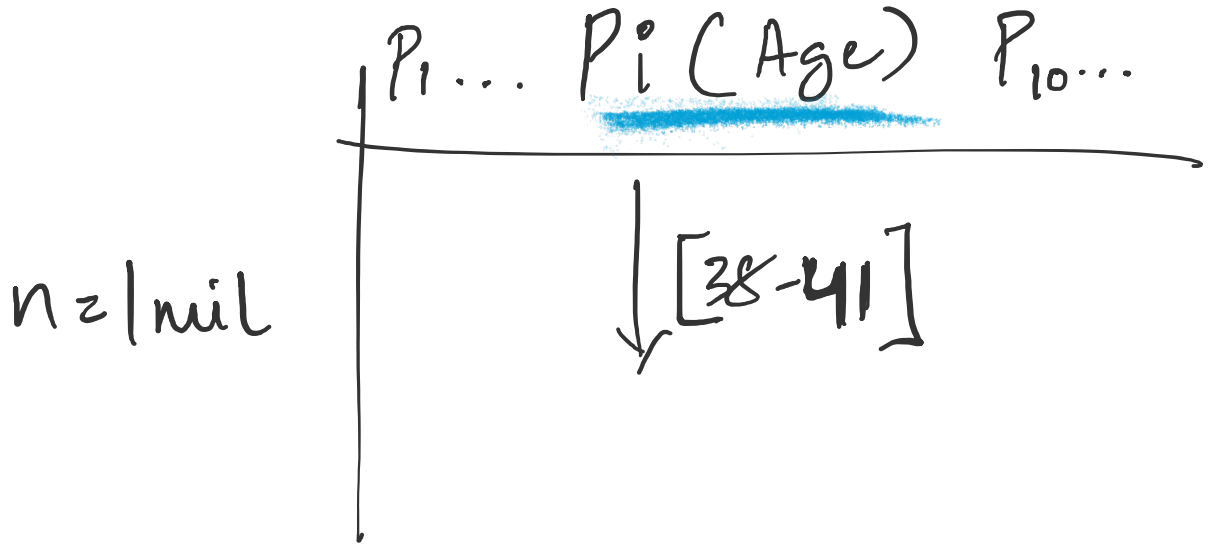
A method of dimensionality reduction includes 2 steps:

- Determine an optimal set of new predictors $Z_1, \ldots, Z_m$, for $m < p$.
- Express each observation in the data in terms of these new predictors. The transformed data will have m columns rather than p.

Thereafter, we can fit a model using the new predictors.

The method for determining the set of new predictors can differ according to application. We will explore a way to create new predictors that captures the *essential* variations in the observed predictor set.

Why *Variations*?

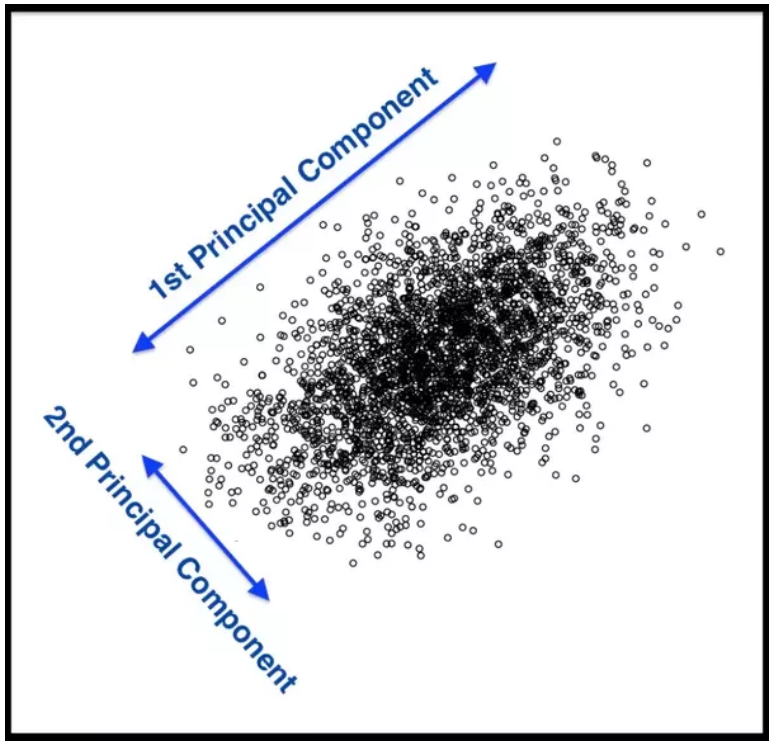


What do we mean by an optimal predictors set?

Principal Components Analysis (PCA)

Principal Components Analysis (PCA)

Principal Components Analysis (PCA) is a method to identify a new set of predictors, as linear combinations of the original ones, that captures the 'maximum amount' of variance in the observed data.



Q: What should I do to address the problem of a large sample size (n = 10 million records)?

A. Select an appropriate model using all the data
B. Design a polynomial regression to fit all the data
C. Calculate the variance of the data to understand its statistical properties
D. Use a subset of the training data for model tuning

Break

WATER

WALK

COFFEE OR TEA

MAKE FRIENDS

Q: If I have a Covid-19 database of n = 10 patients and p = 100 properties describing their gene sequencing, which problem(s) do I run into?

A. High dimensionality
B. Overfitting
C. Multicollinearity
D. All the above

Imagine a big family dinner, where everybody starts asking you about **PCA**.

First you explain it to your great-grandmother; then to you grandmother; then to your mother; then to your spouse; finally, to your daughter (who is a mathematician). Each time the next person is less of a layman.

Here is how the conversation might go.

Great-grandmother: I heard you are studying "Pee-See-Ay". I wonder what that is...

You: Ah, it's just a method of summarizing some data. Look, we have some wine bottles standing here on the table. We can describe each wine by its colour, by how strong it is, by how old it is, and so on. We can compose a whole list of different characteristics of each wine in our cellar. But many of them will measure related properties and so will be redundant. If so, we should be able to summarize each wine with fewer characteristics!

This is what PCA does.

Grandmother: This is interesting! So this PCA thing checks what characteristics are redundant and discards them?

You: Excellent question, granny! No, PCA is not selecting some characteristics and discarding the others. Instead, it constructs some *new* characteristics that turn out to summarize our list of wines well. Of course these new characteristics are constructed using the old ones; for example, a new characteristic might be computed as wine age minus wine acidity level or some other combination like that (we call them *linear combinations*).

In fact, PCA finds the best possible characteristics, the ones that summarize the list of wines as well as possible (among all conceivable linear combinations). This is why it is so useful.

Mother: Hmmm, this certainly sounds good, but I am not sure I understand. What do you actually mean when you say that these new PCA characteristics "summarize" the list of wines?

You: I guess I can give two different answers to this question. First answer is that you are looking for some wine properties (characteristics) that strongly differ across wines. Indeed, imagine that you come up with a property that is the same for most of the wines. This would not be very useful, wouldn't it? Wines are very different, but your new property makes them all look the same! This would certainly be a bad summary. Instead, PCA looks for properties that show as much variation across wines as possible.

The second answer is that you look for the properties that would allow you to predict, or "reconstruct", the original wine characteristics. Again, imagine that you come up with a property that has no relation to the original characteristics; if you use only this new property, there is no way you could reconstruct the original ones! This, again, would be a bad summary. So PCA looks for properties that allow to reconstruct the original characteristics as well as possible.

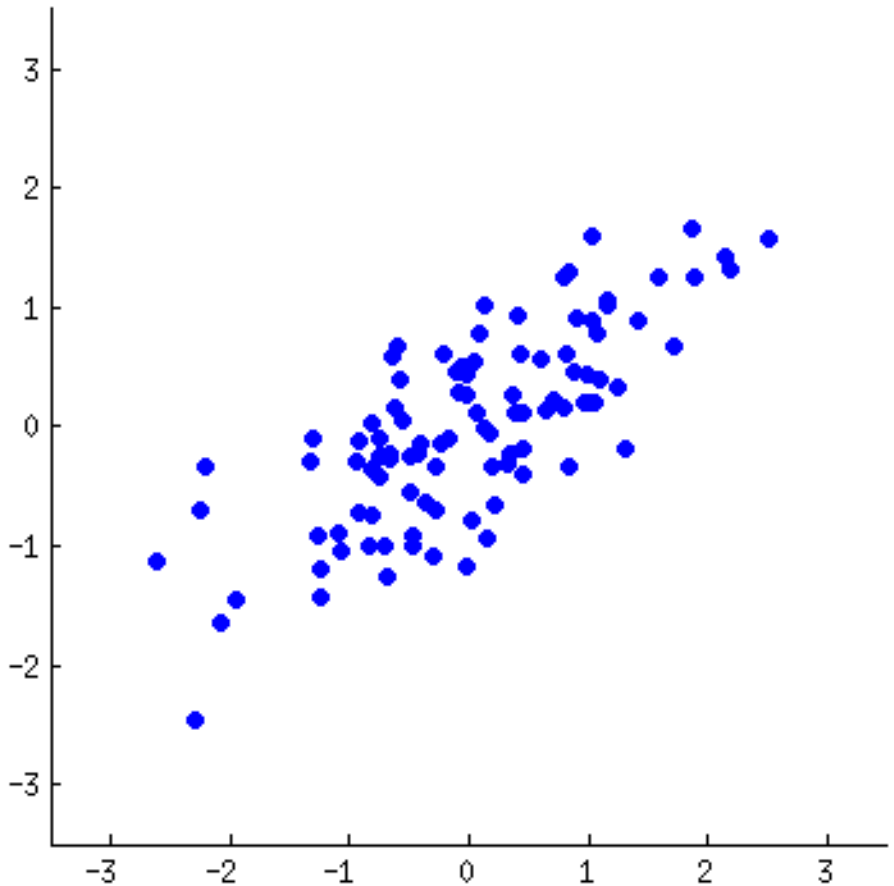
Surprisingly, it turns out that these two aims are equivalent and so PCA can kill two birds with one stone.

Spouse: But honey, these two "goals" of PCA sound so different! Why would they be equivalent?

You: Hmmm. Perhaps I should make a little drawing *(takes a napkin and starts scribbling)*. Let us pick two wine characteristics, perhaps wine darkness and alcohol content -- I don't know if they are correlated, but let's imagine that they are. Here is what a scatter plot of different wines could look like:

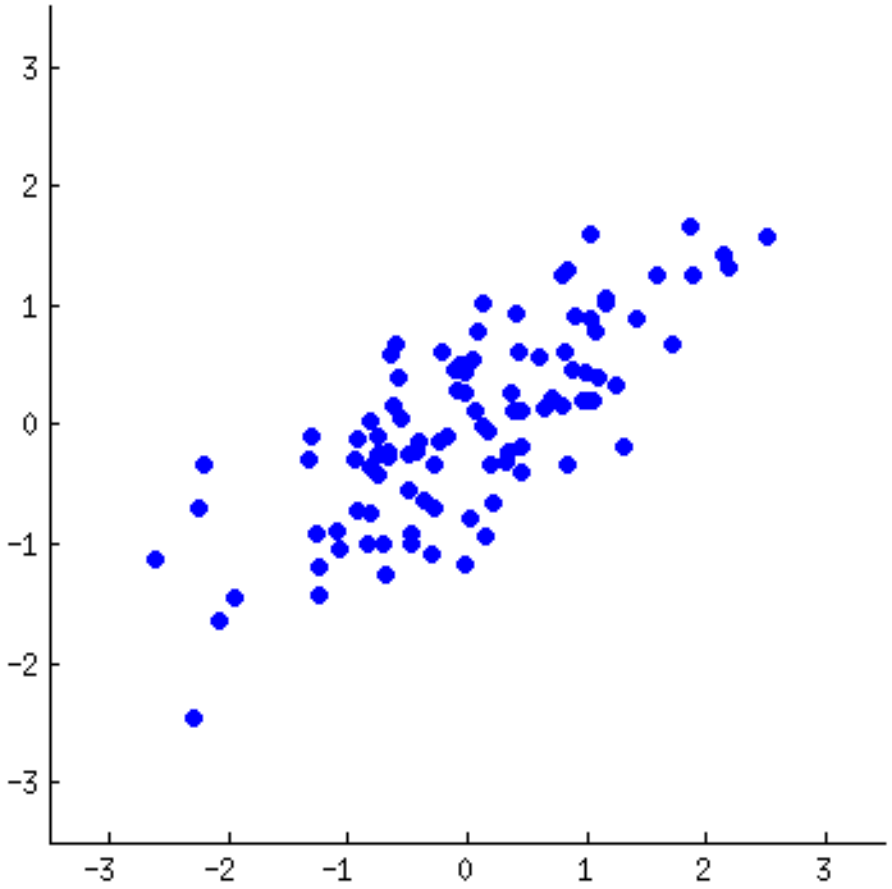
Spouse: But honey, these two "goals" of PCA sound so different! Why would they be equivalent?

You: Hmmm. Perhaps I should make a little drawing *(takes a napkin and starts scribbling)*. Let us pick two wine characteristics, perhaps wine darkness and alcohol content -- I don't know if they are correlated, but let's imagine that they are. Here is what a scatter plot of different wines could look like:



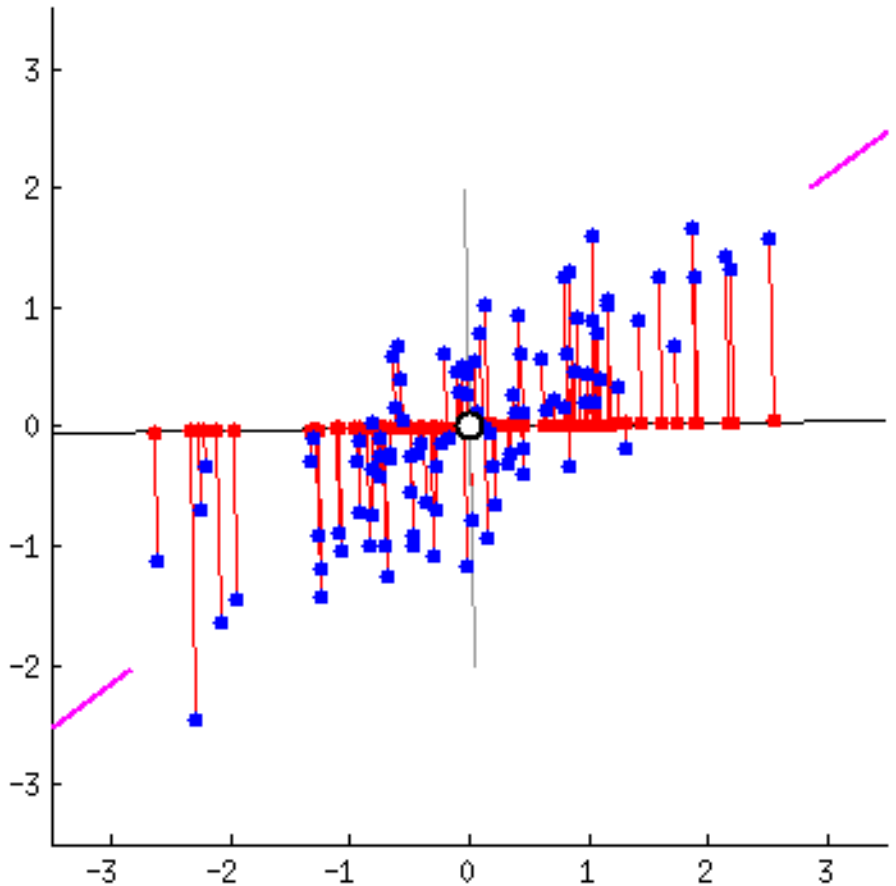
Spouse: But honey, these two "goals" of PCA sound so different! Why would they be equivalent?

You: Each dot in this "wine cloud" shows one particular wine. You see that the two properties (x and y on this figure) are correlated. A new property can be constructed by drawing a line through the centre of this wine cloud and projecting all points onto this line. This new property will be given by a linear combination $w1x+w2y$, where each line corresponds to some particular values of $w1$ and $w2$.



Spouse: But honey, these two "goals" of PCA sound so different! Why would they be equivalent?

You: Now look here very carefully -- here is how these projections look like for different lines (red dots are projections of the blue dots):



Spouse: But honey, these two "goals" of PCA sound so different! Why would they be equivalent?

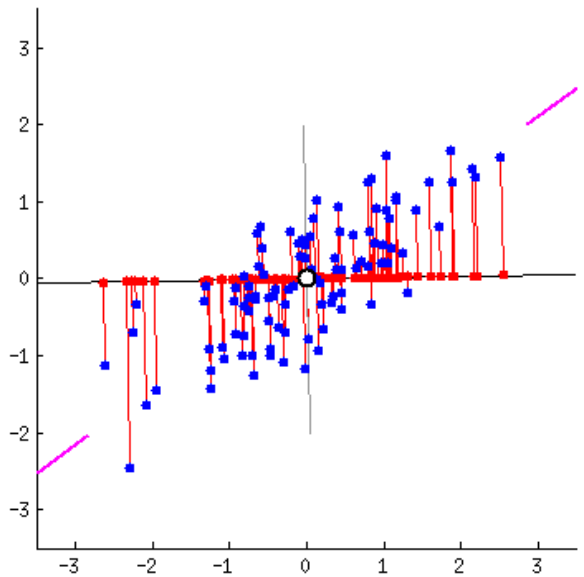
You: As I said before, PCA will find the "best" line according to two different criteria of what is the "best". First, the variation of values along this line should be maximal. Pay attention to how the "spread" (we call it "variance") of the red dots changes while the line rotates; can you see when it reaches maximum? Second, if we reconstruct the original two characteristics (position of a blue dot) from the new one (position of a red dot), the reconstruction error will be given by the length of the connecting red line. Observe how the length of these red lines changes while the line rotates; can you see when the total length reaches minimum?

If you stare at this animation for some time, you will notice that "the maximum variance" and "the minimum error" are reached at the same time, namely when the line points to the magenta ticks I marked on both sides of the wine cloud. This line corresponds to the new wine property that will be constructed by PCA.

By the way, PCA stands for "principal component analysis" and this new property is called "first principal component". And instead of saying "property" or "characteristic" we usually say "feature" or "variable".

Daughter: Very nice, mama! I think I can see why the two goals yield the same result: it is essentially because of the Pythagoras theorem, isn't it? Anyway, I heard that PCA is somehow related to eigenvectors and eigenvalues; where are they on this picture?

You: Brilliant observation. Mathematically, the spread of the red dots is measured as the average squared distance from the centre of the wine cloud to each red dot; as you know, it is called the *variance*. On the other hand, the total reconstruction error is measured as the average squared length of the corresponding red lines. But as the angle between red lines and the black line is always 90◦, the sum of these two quantities is equal to the average squared distance between the centre of the wine cloud and each blue dot; this is precisely Pythagoras theorem. Of course this average distance does not depend on the orientation of the black line, so the higher the variance the lower the error (because their sum is constant).



PCA (cont.)

PCA produces a list of *p principle components* $Z_1,\ldots, Z_p$ such that

- Each Z_i is a linear combination of the original predictors, and its vector norm is 1
- The Z_i's are pairwise orthogonal
- The Z_i's are ordered in decreasing order in the amount of captured observed variance.

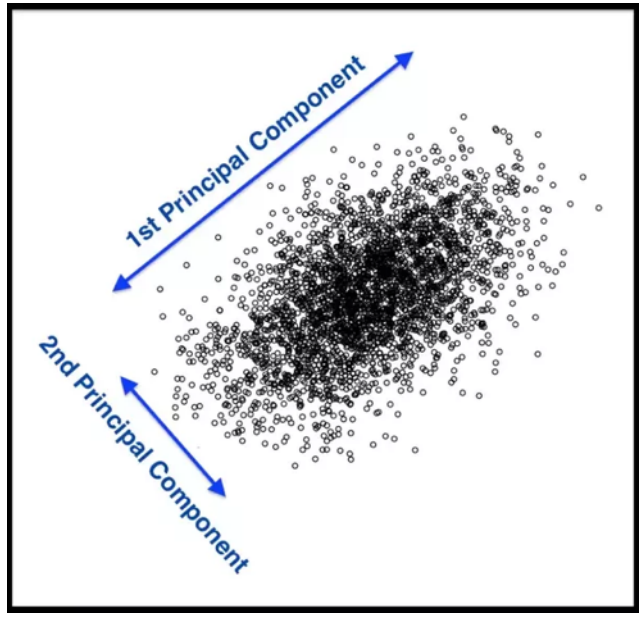
That is, the observed data shows more variance in the direction of Z_1 than in the direction of Z_2.

To perform dimensionality reduction we select the top *m* principle components of PCA as our new predictors and express our observed data in terms of these predictors.

The Intuition Behind PCA

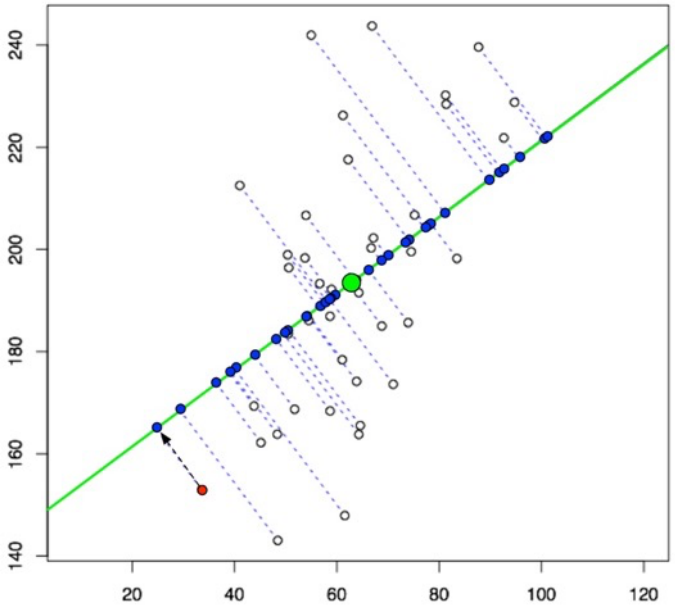
Top PCA components capture the most of amount of variation (interesting features) of the data.

Each component is a linear combination of the original predictors - we visualize them as vectors in the feature space.

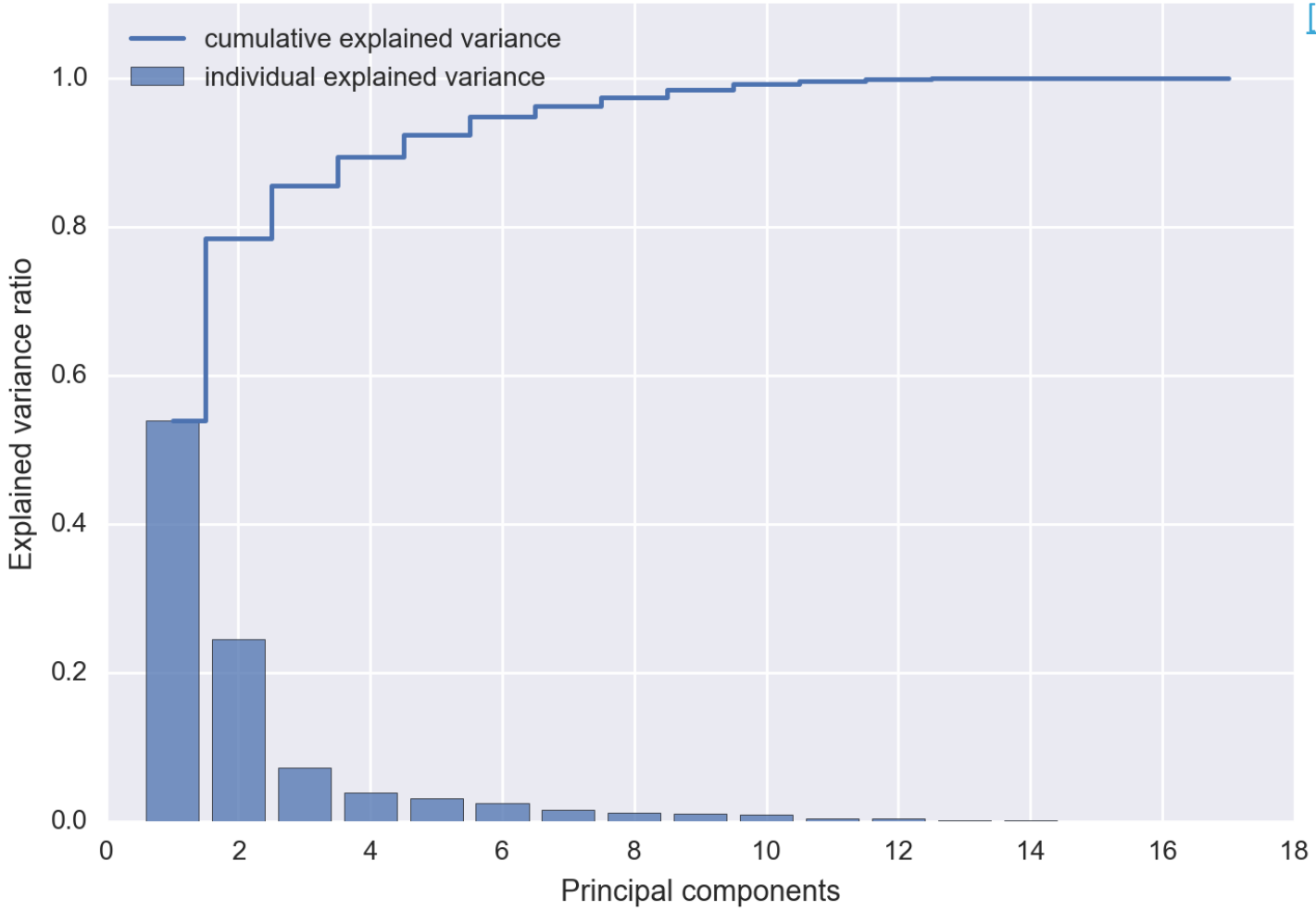


The Intuition Behind PCA (cont.)

Transforming our observed data means projecting our dataset onto the space defined by the top *m* PCA components, these components are our new predictors.



What is a good value for m?



[source]

Q: I have all the census tract information from my village. We have 10, 000 residents and about 50, 000 predictors available for each resident. Which kind of algorithm would I use to avoid overfitting the data?

A. Multivariate Regression
B. Dimensionality Reduction
C. K-Nearest Neighbours
D. A multicollinearity analysis

PCA example in sklearn

PCA is easy to perform in Python using the `decomposition.PCA` function in the `sklearn` package.

In [11]:
```
X = heart_df[['Age','RestBP','Chol','MaxHR']]

# create/fit the 'full' pca transformation
pca = PCA().fit(X)

# apply the pca transformation to the full predictor set
pcaX = pca.transform(X)

# convert to a data frame
pcaX_df = pd.DataFrame(pcaX, columns=[['PCA1' , 'PCA2', 'PCA3', 'PCA4']])

# here are the weighting (eigen-vectors) of the variables (first 2 at least)
print("First PCA Component (w1):",pca.components_[0,:])
print("Second PCA Component (w2):",pca.components_[1,:])

# here is the variance explained:
print("Variance explained by each component:",pca.explained_variance_ratio_)
```

```
First PCA Component (w1): [ 0.03839966  0.05046168  0.99798051 -0.0037393 ]
Second PCA Component (w2): [ 0.180616    0.10481151 -0.01591307 -0.9778237 ]
Variance explained by each component: [0.74831735 0.15023974 0.0852975  0.01614541]
```

What's the difference: Standardize vs. Normalize (Lecture 04)

What is the difference between Standardizing and Normalizing a variable?

- Normalizing means to bound your variable's observations between zero and one. Good when interpretations of "percentage of max value" makes sense.
- Standardizing means to re-center and re-scale your variable's observations to have mean zero and variance one. Good to put all your variables on the same scale (have same weight) and to turn interpretations into "changes in terms of standard deviation."

When to Standardize vs. Normalize

When should you do each?

- Normalizing is only for improving interpretation (and dealing with numerically very large or small measures). Does not improve algorithms otherwise.
- Standardizing can be used for improving interpretation and should be used for specific algorithms. Clustering or PCA.

*Note: you can standardize without assuming things to be [approximately] normally distributed!

It just makes the interpretation nice if they are normally distributed.

A few notes on using PCA

PCA is an unsupervised algorithm. It is done independent of the outcome variable.

- Note: the components as predictors might not be ordered from best to worst!

PCA is not so good because:

1. Direct Interpretation of coefficients in PCR is completely lost. So do not do if interpretation is important.
2. Will not improve predictive ability of a model.

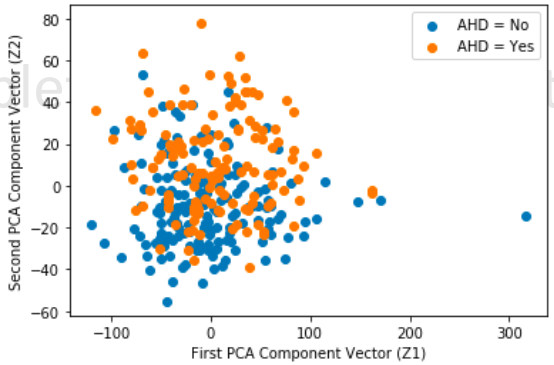
A few notes on using PCA

PCA is an unsupervised algorithm. It is done independent of the outcome variable.

- Note: the components as predictors might not be ordered from best to worst!

PCA is not so good because:

1. Direct Interpretation of coefficients in PCR is comple[illegible]t do if interpretation is important.
2. Will not improve predictive ability of a model.



PCA is great for:

1. Reducing dimensionality in very high dimensional settings.
2. Visualizing how predictive your features can be of your response, especially in the classification setting.
3. Reducing multicollinearity, and thus may improve the computational time of fitting models.

For next class..

Finish Lab 07 to practice programming

Submit Homework 07 for peer review on Peer

Submit Assignment 3 – due in **Week 7** on Sunday at **2330**

See "To do before class" for every lecture (~ 1 hour of self study)

Read paper for **Discussion** session before every Friday

Post questions on the **Discussion** forum on Brightspace (especially on **Clustering** for this week)