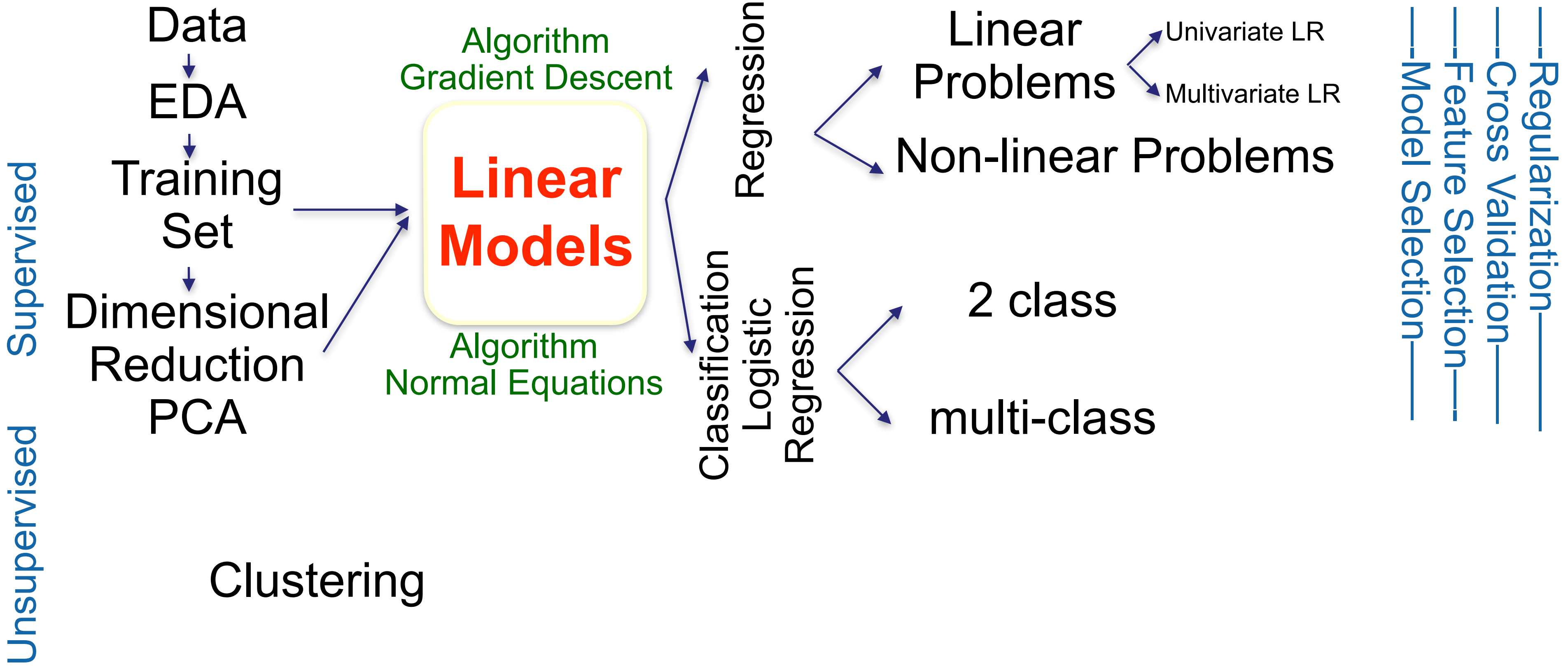
GENERAL ASSEMBLY

INTRO TO DATA SCIENCE
LECTURE 9: DIMENSIONALITY REDUCTION

WHERE ARE WE ON THE DATA SCIENCE ROAD-MAP?



Key Concepts - Motivation

- Dimensionality Reduction
- Removing data redundancy
  - e.g. 2 variables, highly co-linear, reduced to a single variable
- Data Compression
- Data Visualization

Key Concepts - Dimensionality Reduction - Dominant Algorithm

- Principal Components Analysis

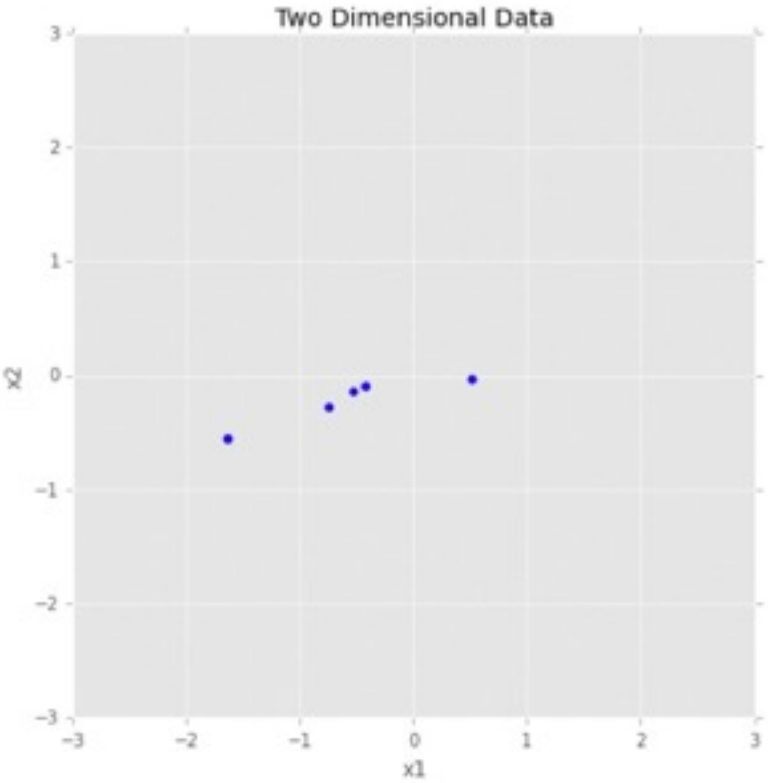
A technique whose purpose is to reduce the dimensionality of a dataset (reduce the number of features), while retaining most of the information of the original dataset

- By far the most popular and commonly used algorithm

- PCA does NOT require data labels, and in this regard could be considered an unsupervised learning algorithm

KEY CONCEPTS - WHAT PCA DOES

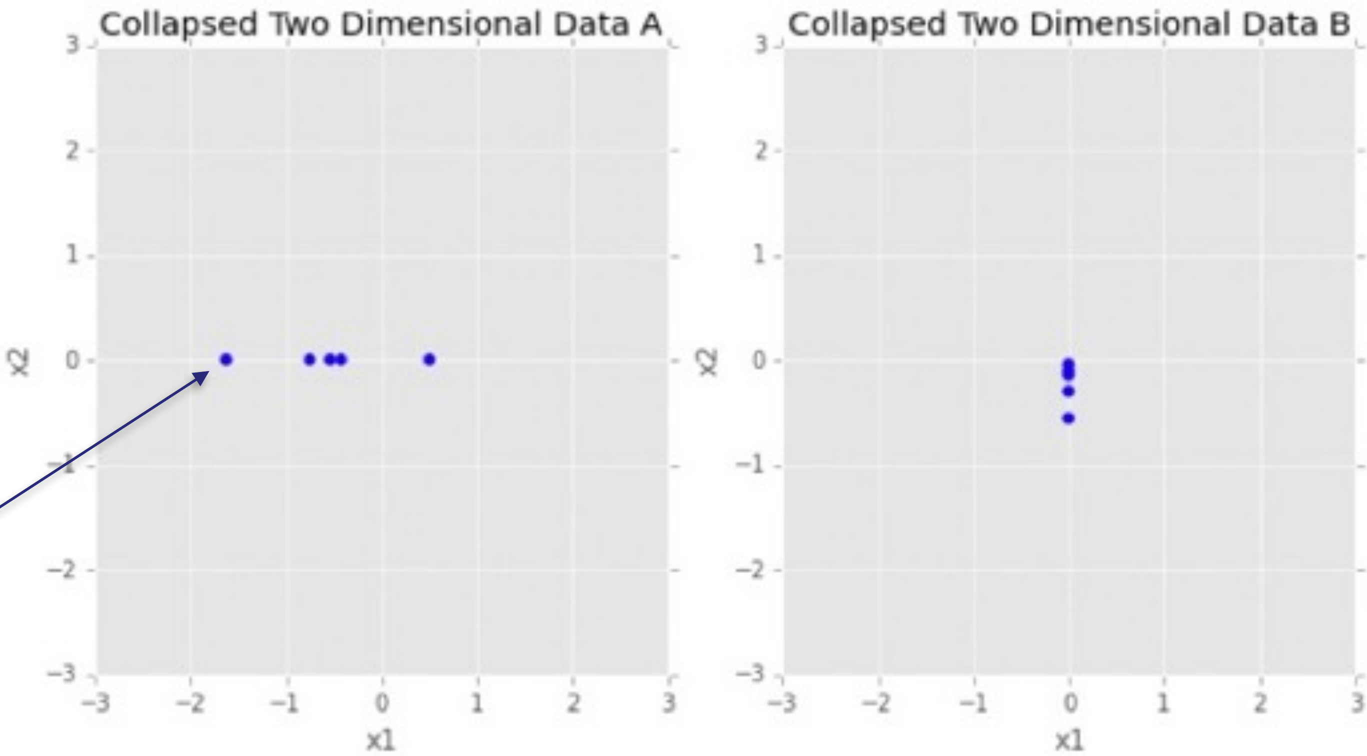
- 5 2-dimensional points
- We want to convert this into 5 1-dimensional points



Key Concepts - What PCA Does

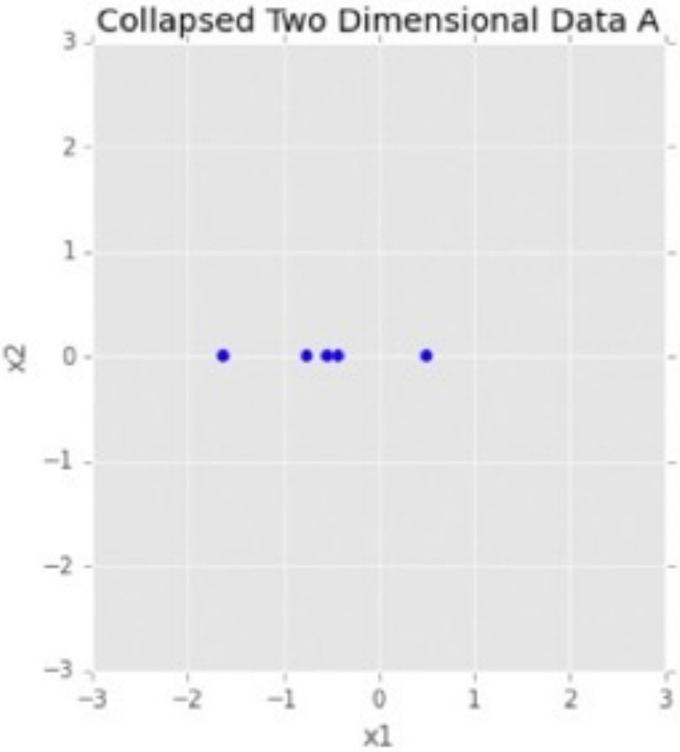
- We could, for example, collapse onto either axis

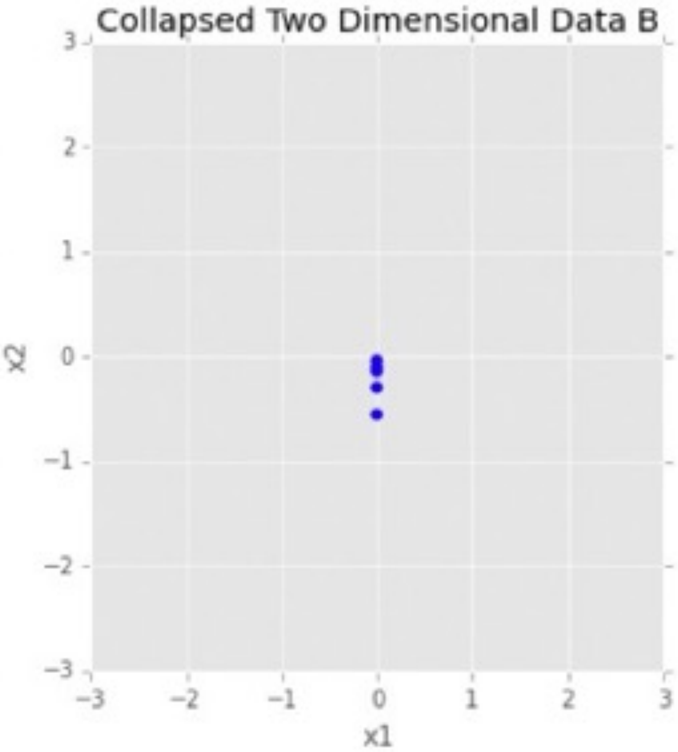
The point (-1.56, 0.24) becomes (-1.56)



KEY CONCEPTS - WHAT PCA DOES

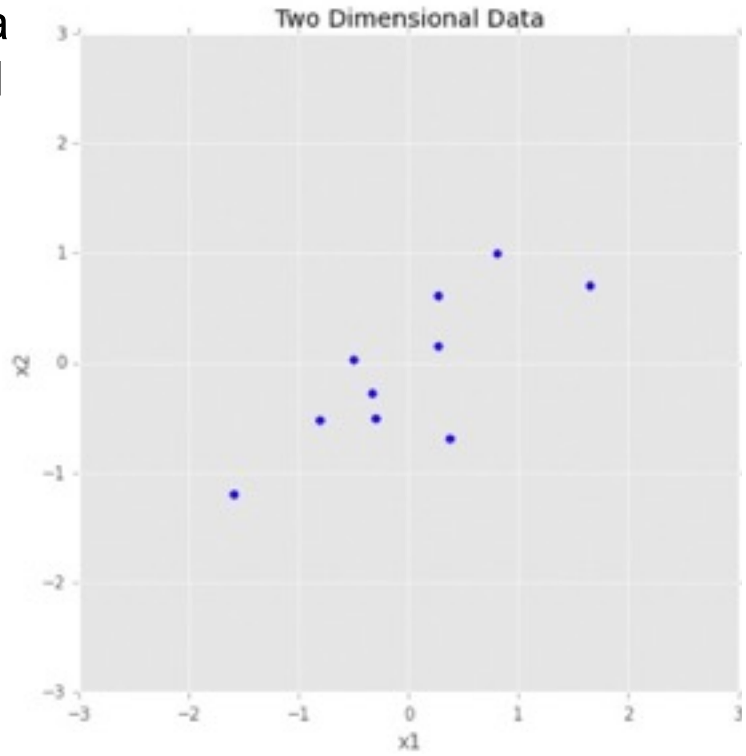
- Clearly projecting the points onto the x-axis yields a better set of 1-D points

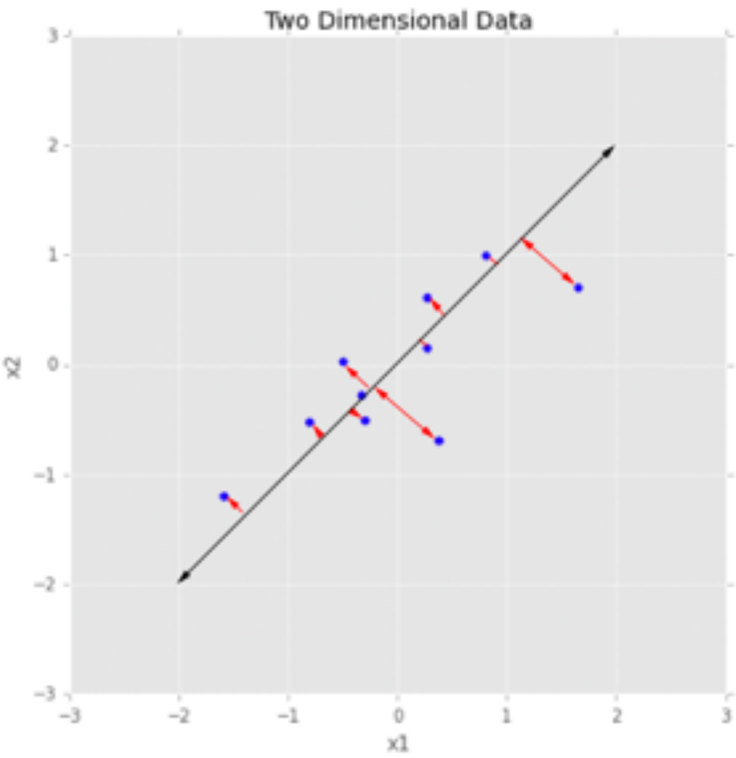




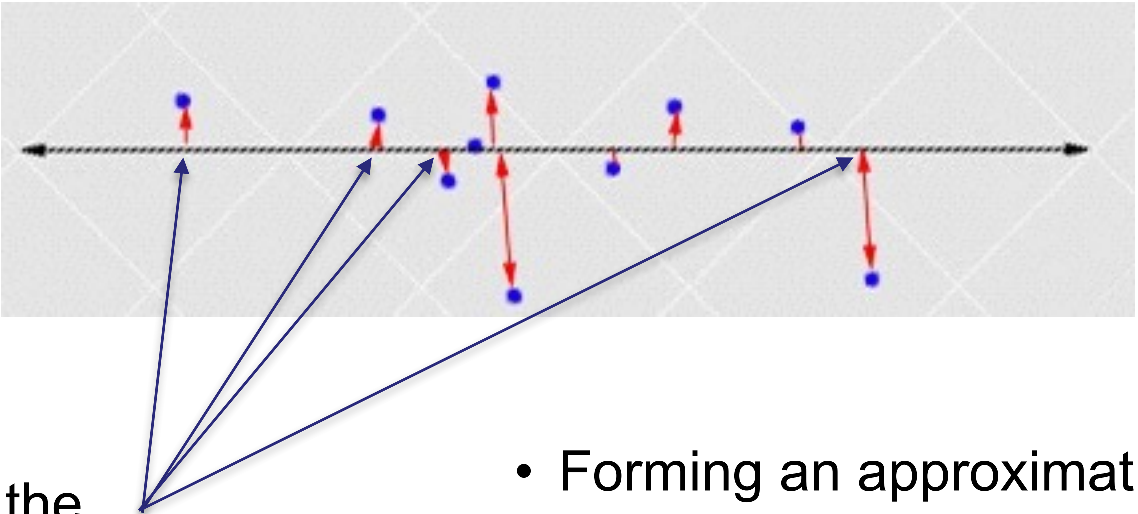
KEY CONCEPTS - WHAT PCA DOES

- But there is a more optimal solution
- Project the data onto a line whose direction is along the maximal variance of the data
- PCA finds such a line





KEY CONCEPTS - WHAT PCA DOES

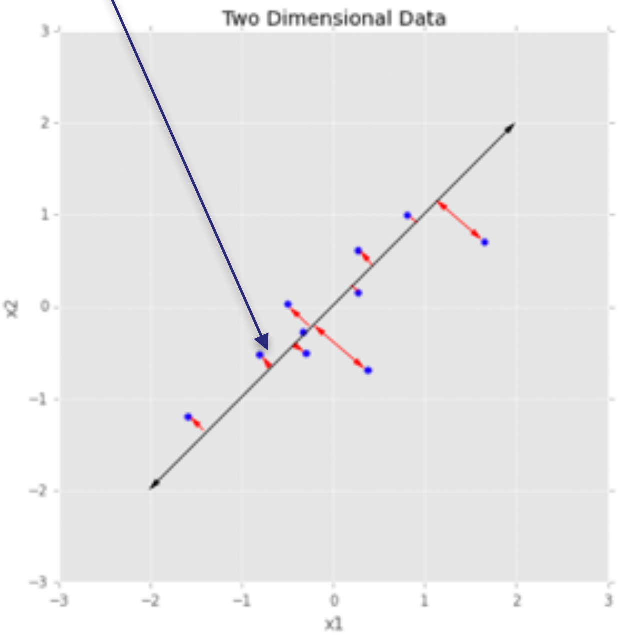


- The transformed numbers appear on the line

- Forming an approximation to the original training set

- PCA finds the optimal projection through the data, and works by minimizing the sum of squares of the projection errors. The red lines in the diagram.

  - PCA requires the data to be zeroed, i.e. subtract off the mean. It also works better if all the features are of a similar scale.



Key Concepts - Most Common Usage

- In our examples we tend to see small dimensions being reduced to 1-D or 2-D, but, in a object recognition project, for example, you might use this technique to reduce the dimensionality from 1000-D to 100-D, or even 10K to 1000!!

- But, obviously, to visualize data we need it to be no more than 3 dimensional. So it can be very useful for visualization.

KEY CONCEPTS - THE ALGORITHM

- Pre-processing:
  - mean normalization
  - plus/minus feature scaling

- In the 2-D case the algorithm finds a vector, or a line direction, that minimizes the sum of the squares of the projection errors of the data

- Algorithmic details require knowledge of linear algebra

Key Concepts - The Algorithm

- Uses a technique called Singular Value Decomposition (SVD)

- To reduce data from N-dimensions to K-dimensions:
  - Compute the covariance matrix of the data
  - Compute the eigenvectors of the covariance matrix (Σ)

- SVD will decompose the covariance matrix of the data into 3 matrices, such that svd(Σ) -> U * S * V

- Σ is an N x N matrix
- U is an N x N matrix, whose K columns are the vectors we want.

KEY CONCEPTS - THE ALGORITHM

- Taking the first K columns of the U matrix gives us the vectors that we need

- These are the K directions that we want to project the data onto

- To obtain the lower dimensional representation of the data we form a matrix from the K vectors of the U matrix (sometimes called the U_reduce matrix)

- We then multiply the transpose of U_reduce by the data

KEY CONCEPTS - SINGULAR VALUE DECOMPOSITION

- Be aware of SVD, because it is one of the most elegant algorithms in linear algebra

- Decomposes a matrix into 3 other matrices, U, S, V
  - S is a real-valued diagonal matrix
  - The diagonal values are called singular values

- Uses:
  - solving sets of simultaneous equations
  - matrix inversion
  - finding eigenvalues of a matrix
  - finding the rank of a matrix

- Particular famous because of it’s numerical stability

Key Concepts - PCA As a Compression Algorithm

- Lossy compression, meaning when you reconstruct the data in the original dimensionality, some information has been lost and cannot be recovered.

KEY CONCEPTS - CHOOSING K

- K is also referred to as the number of principal components
- It is a parameter of the model
- Typical value of k, is to choose the smallest value of k such that 99% of the total variance in the data is retained

KEY CONCEPTS - CHOOSING K

- Suppose you have N features and find K principal components such that K = N
- Each principal component is orthogonal to all the other principal components, this means that the principal components have a zero covariance with each other
- Once the data is transformed then the ‘new’ features also have zero covariance (and hence are uncorrelated)
- Each principal component accounts for some proportion of the variance in the data

Key Concepts - Choosing K

- Principal components are ordered such that the first is responsible for more of the variance than the others

- The idea, then, is to use the first K principal components such that they, combined, account for 99% of the variance in the data

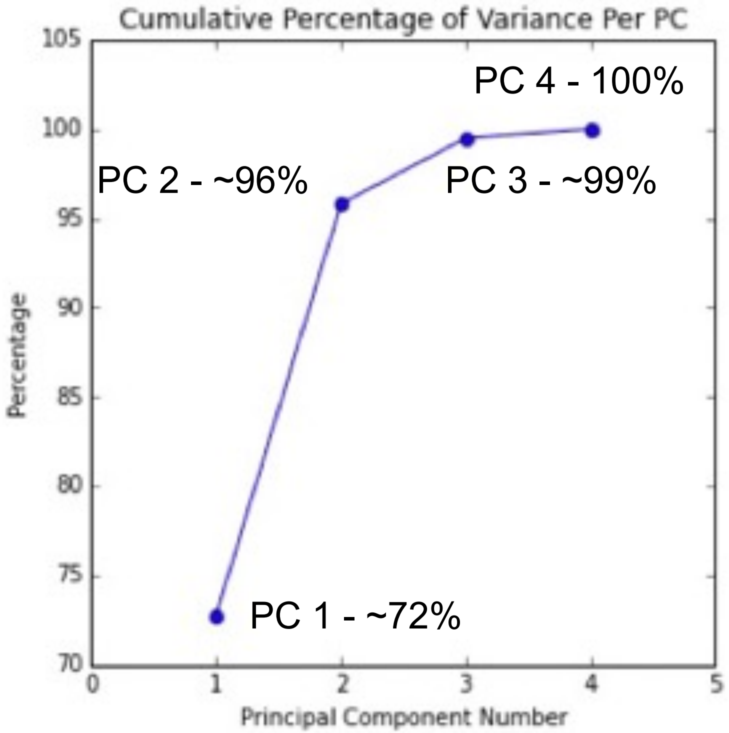
- Some other commonly used numbers are 95%, 90%, or even 85%

KEY CONCEPTS - CHOOSING K

- Can also derive the percentage from the S matrix in the SVD algorithm, or
- Check for a ‘knee in the curve’, or
- But practically you can search for K based on predictive performance
    - Remember it’s a parameter to be found on the training set only

KEY CONCEPTS - EXPLAINED VARIANCE

- Example: a 4 dimensional dataset
- Using PCA we have found the 4 principal components
- The plot shows the amount of the total variance contributed by each principal component
- Obviously being 4-D data, the cumulative sum of 4 principal components will be 100%



KEY CONCEPTS - UNCORRELATED INPUTS

- The PCA transformed features are uncorrelated

- PCA can, therefore, be used if multi-collinearity between input features is a problem - aka remove data redundancy

```
ppf = pd.DataFrame(X_transform)
ppf.corr()
```

	0	1	2	3
0	1.000000e+00	2.246315e-16	7.486036e-17	2.263402e-16
1	2.246315e-16	1.000000e+00	-8.705601e-16	1.003804e-16
2	7.486036e-17	-8.705601e-16	1.000000e+00	1.503160e-16
3	2.263402e-16	1.003804e-16	1.503160e-16	1.000000e+00

Key Concepts - Disadvantage

- The transformed data have no units
- What do your input features mean??
- Not helpful if you are seeking to explain the input-output relationship of a model, in terms of specific input features

KEY CONCEPTS - HOW NOT TO USE PCA

- PCA should not be used to ‘cure’ over-fitting, by reducing the number of features

- Over-fitting should always be addressed using regularization

- PCA is throwing away some information, but if you choose K based on trying to address over-fitting you may end up discarding important information

KEY CONCEPTS - HOW NOT TO USE PCA

- When designing your model do NOT plan to use PCA from the outset

- Build and test your model without using PCA first

- Only use PCA when you can identify a specific reason to use PCA
    - speed (input dimension 10K)
    - compression (memory constraints)
    - multi-colinearity
    - addressing input space dimensionality issues (curse of dimensionality)

KEY CONCEPTS - SKLEARN

- from sklearn.decomposition import PCA
- PCA has a ‘fit’ method, a ‘transform’ method and a ‘fit_transform’ method
- In general you can reduce N dimensional data down onto K dimensions, where K < N, and K > 0
- PCA finds K vectors onto which you can project the data
- PCA is a linear transformation

KEY CONCEPTS - SKLEARN

- In the sklearn PCA object you can specify n_components
- This specify the number of components, or K, that you would like
- or you can specify a fraction, between 0 and 1, to have the algorithm return the number of components that satisfies the percentage variance you have entered as the fraction

KEY CONCEPTS - PCA AND LINEAR REGRESSION

- PCA and Linear Regression are DIFFERENT
- In LR you are trying to predict y
- No such concept in PCA