**Lesson 4: Introduction to Dimensionality Reduction and Principal Component Analysis**

Present 3: Lesson Objectives

By the end of this lesson, you will be able to:

* Understand use cases and motivations for dimensionality reduction.
* An overview of the landscape of dimensionality reduction techniques.
* Understand principal component analysis (PCA) in more detail.
* Apply PCA algorithm to datasets.

Present 4: Motivation

**Motivation**

With the advent of the internet and social media, the amount of data that is being collected increased dramatically. Every minute of the day, 3.5 million text messages are sent and 3.6 million searches are conducted on Google (1). The data is growing not only in terms of depth but also breath; meaning that there are more variables to explore and learn from. Just to provide a few examples:

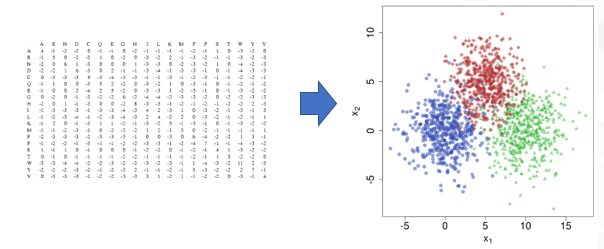
* Retailers capturing data using cameras and tracking shopper behaviour through computer vision algorithms
* TV Set top boxes that collect viewing behaviour every second
* Internet browsing behaviour where each website visit may be a different variable
* A self-driving car that uses a combination of cameras, sensors and LIDAR to navigate autonomously
* Text data that is generated by people through social media posts or tweets

Present 5: Discussion

As self-driving cars, internet of things, robots and virtual assistants become more common and embedded in our lives, the data in the world will continue to grow in both size and complexity. From a machine learning perspective, this is a welcome development since the performance of the models depend greatly on the size and quality of the data. For any particular machine learning task, one of the aims of the data scientist is to include more variables that are potentially predictive of a target outcome. However having lots of variables may also have non-negligible downsides:

Present 6: Interpretability

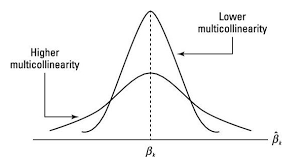
* **Interpretability:** If there are thousands or even millions of variables, it will be almost impossible to identify which variables to include in a model through feature exploration. We may also want to reduce the dimensionality so that we can make visual observations of the data in a 2 or 3 dimensional space.



Present 7: Multicollinearity

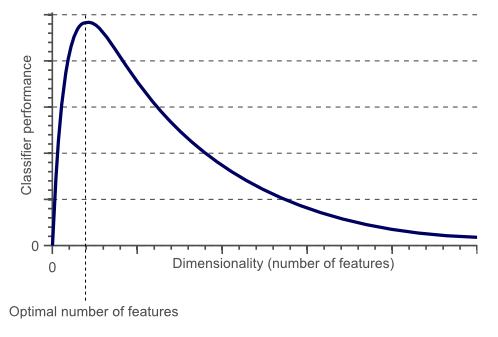
* **Multicollinearity:** Variables in a data set may actually be very highly correlated, with the potential of adversely affecting the performance of the model. To provide an example from a regression model, imagine that A and B are perfectly correlated variables used to predict variable Y. When we train our regression model, the following weights may be allocated to the variables:
  + A = 0.01, B = 0.99
  + A = 0.99, B = 0.01
  + A = 0.5, B = 0.5

From this simple example, we can see that our interpretation of the parameters will be potentially wrong. Now imagine that we have hundreds of variables with messy weights, it will be very hard to generate insights from the model and create accurate simulations.



Present 8: Curse of Dimensionality

* **Curse of dimensionality:** As the number of dimensions/variables increase in our dataset, the volume of the space increases faster and the dataset becomes sparse. According to Hughes Phenomenon, as the number of features increases, so does the performance of a classifier, until it reaches a point where the performance gradually drops. It is advised to have at least 5 training examples for each dimension in the representation. (2)



Present 9: Motivation

Due to the above mentioned reasons, it may be a good idea to reduce the dimensionality of a dataset before embarking on training a predictive model or building a data driven product. To summarize the benefits of dimensionality reduction:

1. Avoid multi-collinearity in order to set weights of the parameters in a model accurately
2. Reduce the storage space and computational challenges when processing the data
3. Visualize data in a 2D or 3D space to generate insights
4. Prevent curse of dimensionality

Present 10: Discussion

**Example use cases for dimensionality reduction**

In this chapter we will go through a couple of real world examples that make use of dimensionality reduction algorithms:

Present 11: Facial Recognition with Principle Component Analysis (PCA)

|  |  |
| --- | --- |
|  | **Facial Recognition:** One important use case for dimensionality reduction is computer vision exercises such as facial recognition. In one particular facial recognition methodology called Eigenface approach, the PCA algorithm (that we will cover in more detail) is used to match images with associated identities. In first step, data is split into training and test images where each image in the training set also has an identity label associated with it. PCA algorithm is employed on the pixels of the image to create a reduced representation of the image through feature extraction so that we end up with the feature vector for each person. The trained PCA algorithm is then fitted on the images of the test set to create another reduced representation. The final step is to match the most similar representations generated from the test set to the representations in the training set, thereby recognizing the identity of the person from the image. Please feel free to read the article for more information, especially after we cover PCA algorithm in more detail. (3) |

Present 12: Text Categorization

In this real world example, any type of text based documents can be categorized using a bag of words approach and then applying a dimensionality reduction algorithm.

* E-mails as an example can be grouped into spam or not spam by counting the occurrence of each word in an email where words such as “lottery” or “pay” would be better features for spam classification than “girl ” or “boy”.
* The next step would be to compute the low-dimensional projection of the bag-of-words vectors and use these vectors in classification algorithms such as Logistic Regression or Random Forest instead of original emails.
* Using the projections instead of original emails would enable the training to run much faster and overfitting will be reduced.

**Overview of Dimensionality Reduction Techniques**

In this chapter we will provide an overview of the two main approaches to dimensionality reduction and briefly describe the most prominent algorithms for each approach.

Present 14: Non-negative matrix factorization

**Non-negative matrix factorization (NMF):** Non-negative matrix factorization is a group of algorithms in multivariate analysis and linear algebra where a matrix V is factorized into (usually) two matrices W and H, with the property that all three matrices have no negative elements. This non-negativity makes the resulting matrices easier to inspect. NMF finds applications in such fields as astronomy, computer vision, document clustering, recommender systems, and bioinformatics. We will get a chance to work with this algorithm in Chapter 7.

Present 15: Linear discriminant analysis

**Linear discriminant analysis (LDA):** LDA is a generalization of Fisher's linear discriminant, a method to find a linear combination of features that characterizes or separates two or more classes of objects or events. LDA is also closely related to principal component analysis (PCA) in that they both look for linear combinations of variables which best explain the data. LDA explicitly attempts to model the difference between the classes of data where PCA on the other hand does not take into account any difference in class. LDA has been used in many applications such as bankruptcy prediction, facial recognition, biomedical studies classifying diseases and topic analysis from textual data. This algorithm will also be used in Chapter 7.

Present 16: Autoencoder

**Autoencoders:** Autoencoders are a deep learning method employing feed-forward neural networks with a bottle-neck hidden layer. The basic idea behind autoencoders is to encode information (as in compress, not encrypt) automatically, hence the name.

* The entire network always resembles an hourglass like shape, with smaller hidden layers than the input and output layers.
* AEs are also always symmetrical around the middle layer(s) (one or two depending on an even or odd amount of layers).
* The smallest layer(s) is|are almost always in the middle, the place where the information is most compressed (the chokepoint of the network). We will cover autoencoders in more detail in the next chapter.

Present 17: T-distributed Stochastic Neighbor Embedding (t-SNE)

**T-distributed Stochastic Neighbor Embedding**: T-SNE is a non-linear dimensionality reduction technique well-suited for embedding high-dimensional data for visualization in a low-dimensional space of two or three dimensions. T-SNE has been used for visualization in a wide range of applications, including computer security research, music analysis, cancer research, bioinformatics, and biomedical signal processing. We will cover T-SNE in more detail in the sixth chapter.

Present 19: PCA Overview

Principle Component Analysis: PCA is mathematically defined as an orthogonal linear transformation that transforms the data to a new coordinate space such that the greatest variance (basically how widely the data is spread out) by some projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on.

PCA can be thought of as fitting an n-dimensional ellipsoid to the data, where each axis of the ellipsoid represents a principal component. If some axis of the ellipsoid is small, then the variance along that axis is also small.

Present 20: PCA Step 1- Calculate Covariance Matrix

To find the axes of the ellipsoid, we compute the covariance matrix of the data. Covariance is a measure of the joint variability of two random variables. If the greater values of one variable mainly correspond with the greater values of the other variable, and the same holds for the lesser values, (i.e., the variables tend to show similar behavior), the covariance is positive.



Consider the 2x2 covariance matrix below:



This means that the variance of variable x is 1 and variance of y is 0.7 and the covariance between them is 0.5.

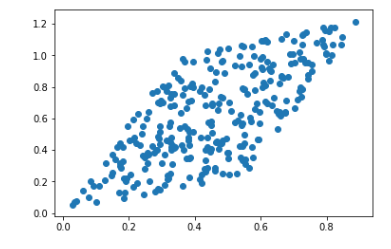
Present 21: PCA – Step 2: Orthogonal Transformation

Then the next step is to calculate the eigenvalues and corresponding eigenvectors of this covariance matrix. The mathematics of the calculation can become calculated, however for our purposes to understand the intuition behind the algorithm we should know that eigenvectors can tell us about the direction of maximum variance in our data and eigenvalues will tell us which eigenvectors are the most important. It follows that the direction of the first principal component is given by the first eigenvector of the covariance matrix. The second principal component direction (the direction orthogonal to the first component that has the largest projected variance) is the eigenvector corresponding to the second largest eigenvalue and so on.

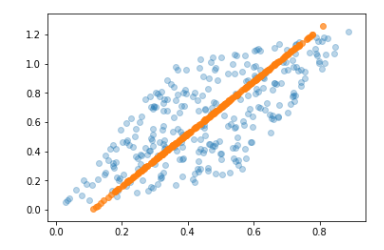
Present 22: Exercise 1: Applying PCA algorithm

In this activity, we will use PCA algorithm and apply it on a sample data to understand the intuition behind it.

Visualize the data on a scatter plot:



Visualize where our data would sit if we transformed it back to its original space:



Present 23: Exercise 2: Applying PCA algorithm for Facial Recognition

In this exercise we will apply the PCA algorithm to the faces of some famous people. As discussed earlier, this is a real world example of how PCA is being used to reduce dimensionality of images where data size is significantly reduced so it is easier to train machine learning algorithms while maintaining the most important features of the image data.

Comparison of original image and transformed image with PCA with 200 dimensions:



Present 24: Activity 1: Applying PCA algorithm for Facial Recognition

In this activity, please pair up with someone else in the class.

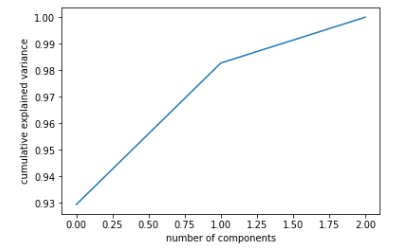
First we will load the Iris dataset which consists of 50 samples from each of three species of Iris (Iris setosa, Iris virginica and Iris versicolor). Four features were measured from each sample: the length and the width of the sepals and petals, in centimeters.

Then the aim will be to:

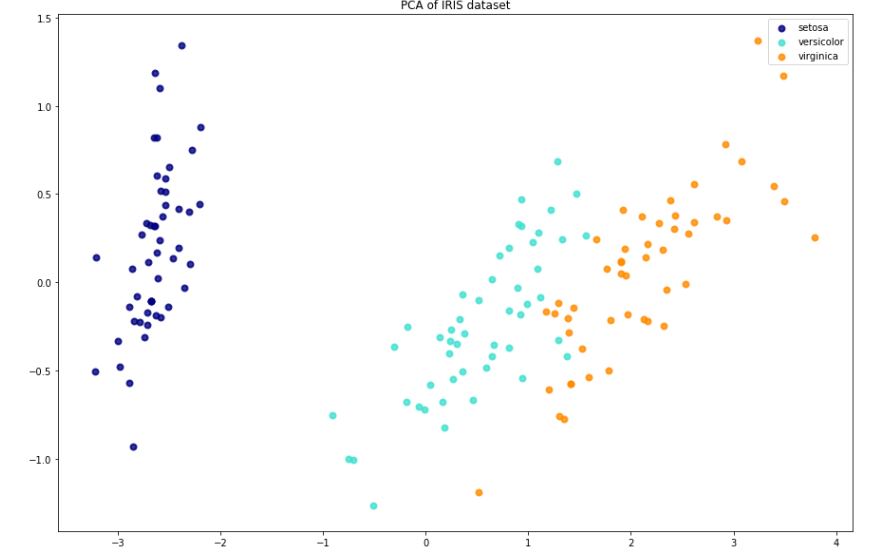
1a) Fit the PCA algorithm with 3 components and visualize the explained variance ratio by number of components. Therefore we should have the explained variance ratio on one axis, and the number of components on the second axis.

1b) Fit the PCA algorithm with 2 components and visualize the data on a two dimensional space.

Present 25: Activity 1: Solution 1



Present 26: Activity 1: Solution 2



Present 27: Discussion

Now that we know about the PCA algorithm, can you think of any more use cases where PCA can be potentially applied?

Present 28: Summary

In this lesson, we

* Explored some use cases and motivations for dimensionality reduction.
* Provided an overview of the landscape of dimensionality reduction techniques.
* Understand principal component analysis (PCA) in more detail.
* Apply PCA algorithm to different datasets.

*1) Data Never Sleeps 5.0 (2018). Retrieved from* [*https://www.domo.com/learn/data-never-sleeps-5?aid=ogsm072517\_1&sf100871281=1*](https://www.domo.com/learn/data-never-sleeps-5?aid=ogsm072517_1&sf100871281=1)

*2) Koutroumbas, Sergios Theodoridis, Konstantinos (2008). Pattern Recognition - 4th Edition. Burlington. Retrieved 8 January2018.*

*3) Liton Chandra Paul , Abdulla Al Sumam (2012). Face Recognition Using Principal Component Analysis Method. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 1, Issue 9, November 2012*