Chapter 8: Dimensionality Reduction

Aly Khaled

August 23, 2021

1 The Curse of Dimensionality

- The more dimensions the training set has, the greater the risk of **overfitting** it.
- One solution to the curse of dimensionality could be to **increase** the size of the training set to reach a sufficient density of training instances.
- Unfortunately, the number of training instances required to reach a given density grows **exponentially** with the number of dimensions.

2 Approaches for Dimensionality Reduction

- There are two main approaches to reducing dimensionality **Projection** and **Manifold**.

2.1 Projection

- In most real world problems, training instances are not spread out uniformly across all dimensions. As a result, all training instances lie within (or close to) a much lower-dimensional subspace of the high dimensional space.
-

2.2 Manifold Learning

- More generally, a d-dimensional manifold is a part of an n-dimensional space (where d $<$ n) that locally resembles a d-dimensional hyperplane.
- Many dimensionality reduction algorithms work by modeling the manifold on which the training instance lie: this is called **Manifold Learning**. It relies on the **manifold assumption** also called **manifold hypothesis**.
- The manifold assumption is often accompanied by another implicit assumption: that the task at hand will be simpler if expressed in the lower dimensional space of the manifold.
- Reducing the dimensionaliy of your training set before training a model will usually speed up training, but it may not always lead to a better or simpler solution; it all depends on the dataset.

3 Dimensionality reduction algorithms:

- There are many dimensionality reduction algorithms like: PCA, Kernel PCA, LLE.

3.1 PCA

- **Principal Component Analysis (PCA)** is by far the most popular dimensionality reduction algorithm.
- It first identify the hyperplane that lies closet to the data, and then it projects the data onto it.
- Before we can project the training set onto a lower-dimensional hyperplane, we first need to choose the right hyperplane by choosing the axis that preserves the **maximum amount of variance** or the axis that **minimizes the mean squared distance between the original dataset and its projection onto that axis**.
- PCA identifies the axis that accounts for the largest amount of variance in the training set.

3.2 Kernel PCA

-

3.3 LLE

-

Papers and Reports to read later

- Karl Pearson, "On Lines and Planes of Closet Fit to System of Points in Space".
- Bernhard Scholkopf et al., "Kernel Principal Component Analysis" in Lecture Notes in Computer Science.
- Gokhan H. Bakir et al., "Learning to Find Pre-Images", Proceeding of the 16th international Conference on Neutral Information Processing Systems.
- Sam T. Roweis and Lawrence K. Saul, "Nonlinear Dimensionality Reduction by Locally Embedding"