MLT Week-1

Sherry Thomas
21f3001449

Contents

Abstract

The week provides an introduction to Machine Learning and subsequently delves into the syllabus with a focus on unsupervised learning. The two primary areas of study covered are representation learning and Principal Component Analysis (PCA).

Introduction to Machine Learning

Machine Learning is a sub-field of artificial intelligence concerned with the design of algorithms and statistical models that allow computers to learn from and make predictions or decisions based on data, without being explicitly programmed. It utilizes mathematical optimization, algorithms, and computational models to analyze and understand patterns in data and make predictions about future outcomes.

It can be further explained as follows:

- Why: Machine Learning is used to automate tasks that would otherwise require human intelligence, to process vast amounts of data, and to make predictions or decisions with greater accuracy than traditional approaches. It also has surged in popularity in recent years.
- Where: Machine Learning is applied in various fields such as computer vision, natural language processing, finance, and healthcare, among others.Where: Machine Learning is applied in various fields such as computer vision, natural language processing, finance, and healthcare, among others.

- What: Machine Learning departs from traditional procedural approaches, instead it is driven by data analysis. Rather than memorizing specific examples, it seeks to generalize patterns in the data. Machine Learning is not based on magic, rather it relies on mathematical principles and algorithms.

Broad Paradigms of Machine Learning

1. **Supervised Learning**:Supervised Machine Learning is a type of machine learning where the algorithm is trained on a labeled dataset, meaning that the data includes both inputs and their corresponding outputs. The goal of supervised learning is to build a model that can accurately predict the output for new, unseen input data. Few examples:

- Linear regression for predicting a continuous output
- Logistic regression for binary classification problems
- Decision trees for non-linear classification and regression problems
- Support Vector Machines for binary and multi-class classification problems
- Neural Networks for complex non-linear problems in various domains such as computer vision, natural language processing, and speech recognition

2. **Unsupervised Learning**: Unsupervised Machine Learning is a type of machine learning where the algorithm is trained on an unlabeled dataset, meaning that only the inputs are provided and no corresponding outputs. The goal of unsupervised learning is to uncover patterns or relationships within the data without any prior knowledge or guidance. Few examples:

- Clustering algorithms such as K-means, hierarchical clustering, and density-based clustering, used to group similar data points together into clusters
- Dimensionality reduction techniques such as Principal Component Analysis (PCA), used to reduce the number of features in a dataset while preserving the maximum amount of information
- Anomaly detection algorithms used to identify unusual data points that deviate from the normal patterns in the data

3. **Sequential learning**: Sequential Machine Learning (also known as time-series prediction) is a type of machine learning that is focused on making predictions based on sequences of data. It involves training the model on a sequence of inputs, such that the predictions for each time step depend on the previous time steps. Few examples:

- Time series forecasting, used to predict future values based on past trends and patterns in data such as stock prices, weather patterns, and energy consumption
- Speech recognition, used to transcribe speech into text by recognizing patterns in audio signals
- Natural language processing, used to analyze and make predictions about sequences of text data

Representation Learning

Representation learning is a sub-field of machine learning that focuses on learning meaningful and compact representations of complex data for tasks such as dimensionality reduction, clustering, and classification.

Given a dataset $\{x_1, x_2, \ldots, x_n\}$ where $x_i \in \mathbb{R}^d$, we need to find a representation for it with the minimum reconstruction error.

Let w, where $||w|| = 1$, be the best linear representation of the dataset.

The representation is given by,

$$\frac{(x_i^T w)}{w^T w} w$$
$$\text{But } ||w|| = 1$$
$$\therefore \text{ Projection } = (x_i^T w)w$$

The reconstruction error is given by,

$$\text{Reconstruction Error}(f(w)) = \frac{1}{n}\sum_{i=1}^{n} ||x_i - (x_i^T w)w||^2$$

where $x_i - (x_i^T w)w$ is the residue and can be given by x'.

Our objective is to minimize the reconstruction error. Minimizing it, we get,

$$\begin{aligned}
\min_{w \in ||w||=1} f(w) &= \frac{1}{n}\sum_{i=1}^{n} -(x_i^T w)^2 \\
\therefore \max_{w \in ||w||=1} f(w) &= \frac{1}{n}\sum_{i=1}^{n} (x_i^T w)^2 \\
&= w^T(\frac{1}{n}\sum_{i=1}^{n} x_i x_i^T)w \\
\max_{w \in ||w||=1} f(w) &= w^T C w
\end{aligned}$$

where $C = \frac{1}{n}\sum_{i=1}^{n} x_i x_i^T$ is the Covariance Matrix and $C \in \mathbb{R}^{d\times d}$.

From the above equation we find that w is the eigenvector corresponding to the largest eigenvalue λ of C.

w is also called the First Principal Component of the dataset.

Potential Algorithm

With this information, we can give the following algorithm:\ Given a dataset $\{x_1, x_2, \ldots, x_n\}$ where $x_i \in \mathbb{R}^d$,

- Center the dataset

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$
$$x_i = x_i - \mu \qquad \forall i$$

- Find the best representation $w \in \mathbb{R}^d$ and $||w|| = 1$.
- Replace $x_i = x_i - (x_i^T w)w \quad \forall i$
- Repeat the first three steps until residues become zero and we obtain $w_2, w_3, \dots, w_d$.

But is this the best way? How many w do we need for optimal compression?

Principal Component Analysis

Principal Component Analysis (PCA) is a technique used to reduce the dimensionality of a dataset by identifying the most important features, or principal components, that explain the maximum amount of variance in the data. PCA accomplishes this by transforming the original dataset into a new set of variables that are uncorrelated and ordered by their importance in explaining the variance in the data. This can be helpful for visualizing high-dimensional data or for preprocessing data before performing machine learning tasks.

Following the potential algorithm from above, using the $\{w_1, w_2, \dots, w_d\}$, we get,

$$\forall i \quad x_i - ((x_i^T w_1)w_1 + (x_i^T w_2)w_2 + \dots + (x_i^T w_d)w_d) = 0$$
$$\therefore x_i = (x_i^T w_1)w_1 + (x_i^T w_2)w_2 + \dots + (x_i^T w_d)w_d$$

From the above equation, we can say that we can represent the data using $\{x_i^T w_1, x_i^T w_2, \dots, x_i^T w_d\}$, which are constants, and $\{w_1, w_2, \dots, w_d\}$ which are vectors.

Therefore, data which was initially $d \times n$ can now be represented with $d(d+n)$ elements which doesn't seem great at first.

But if the data lies in a lower sub-space, the residues can be zero without the need for d principal components.

If the data can be represented using only k principal components, where $k << d$, the data now can be compressed from $d \times n$ to $k(d+n)$.

Approximate Representation

If the data can be approximately represented by a lower sub-space, is it enough that we use only those k projections? How much variance should be covered?

Given a centered dataset $\{x_1, x_2, \dots, x_n\}$ where $x_i \in \mathbb{R}^d$, let C be its covariance matrix, and let $\{\lambda_1, \lambda_2, \dots, \lambda_d\}$ be its eigenvalues, which are non-negative because the covariance matrix is a positive semi-definite matrix, placed in descending order, and let $\{w_1, w_2, \dots, w_d\}$ be its corresponding eigenvectors of unit length.

The eigen equation for the covariance matrix can be given by,

$$
\begin{aligned}
Cw &= \lambda w \\
w^T C w &= w^T \lambda w \\
\therefore \lambda &= w^T C w \qquad \{w^T w = 1\} \\
\lambda &= \frac{1}{n} \sum_{i=1}^{n} (x_i^T w)^2
\end{aligned}
$$

Therefore, as the mean is zero, λ gives the variance captured by the eigenvector w.

A good rule of thumb is that the variance captured by P.C.A. should be at least 95%. If the first K eigenvectors capture the required variance, this is given by,

$$\frac{\displaystyle\sum_{k=1}^{K} \lambda_k}{\displaystyle\sum_{i=1}^{d} \lambda_i} \geq 0.95$$

Hence, the higher the variance captured, the lower is the error obtained.

P.C.A. Algorithm

Given a centered dataset $\{x_1, x_2, \ldots, x_n\}$ where $x_i \in \mathbb{R}^d$, let C be its covariance matrix.\ The algorithm is as follows:

- Step 1: Find the eigenvalues and eigenvectors of C. Let $\{\lambda_1, \lambda_2, \ldots, \lambda_d\}$ be its eigenvalues placed in the descending order, and let $\{w_1, w_2, \ldots, w_d\}$ be its corresponding eigenvectors of unit length.
- Step 2: Calculate K, where K is the number of required top eigenvalues and eigenvectors, according to the required variance that needs to be covered.
- Step 3: Project the data onto the eigenvectors, and obtain the required representation as a linear combination of those projections.

In short, we can say that P.C.A. is a dimensionality reduction technique that finds combination of features that are de-correlated (independent of each other).

Credits

- Professor Arun Rajkumar: The content as well as the notations are from his slides and lecture.

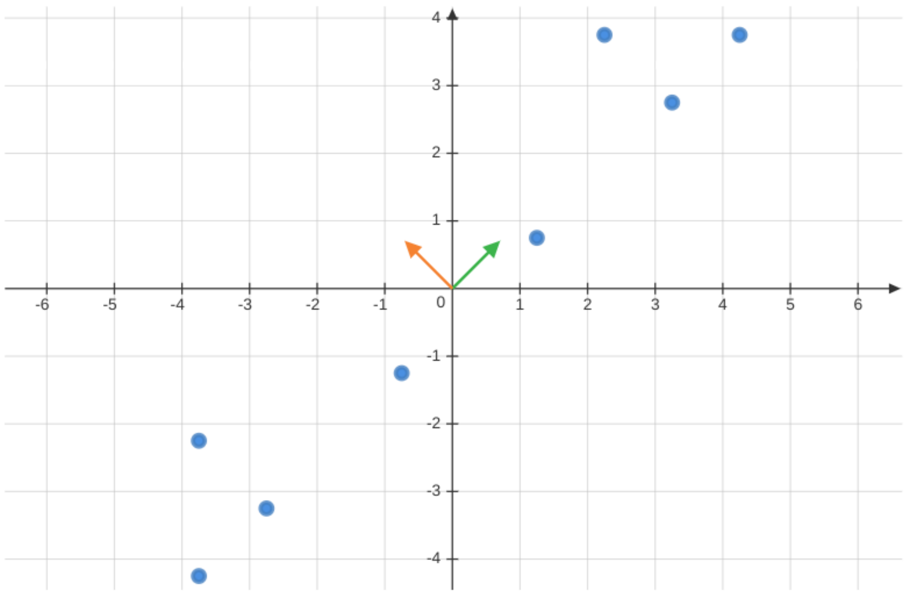

Figure 1: The dataset depicted in the diagram has two principal components: the green vector represents the first PC, whereas the red vector corresponds to the second PC.