Machine Learning

Third assignment solutions

Mobin Nesari
Dr. Hadi Farahani
May 18, 2023

Question 1:

What is the curse of dimensionality and how does it affect clustering?

Answer:

The "curse of dimensionality" is a term used to describe the negative effects that arise when working with high-dimensional data. As the number of dimensions in the data increases, the volume of the space increases exponentially, and the number of data points required to maintain a certain level of statistical significance grows exponentially as well. This results in sparsity of data points, making it difficult to draw meaningful conclusions from the data.

In the context of clustering, the curse of dimensionality can make it more challenging to find natural clusters in high-dimensional data. This is because when there are many dimensions, the distance between any two points in the dataset becomes increasingly similar, leading to a situation where all points appear to be equidistant from one another. As a result, traditional clustering algorithms that rely on measuring distances between points may struggle to identify meaningful clusters in such high-dimensional spaces.

To mitigate the curse of dimensionality in clustering, one approach is to use dimensionality reduction techniques such as Principal Component Analysis (PCA), t-SNE, or UMAP to reduce the number of dimensions in the data while preserving as much of the original variability as possible. Another strategy is to use clustering algorithms specifically designed for high-dimensional data, such as subspace clustering methods that attempt to identify clusters within smaller subspaces of the high-dimensional space.

Question 2:

In what cases would you use regular PCA, incremental PCA, randomized PCA, or random projection?

Answer:

PCA (Principal Component Analysis) is a popular technique for dimensionality reduction in machine learning. However, there are several variants of PCA that can be used depending on the specific requirements of the problem and the computational resources available.

Here are some general guidelines on when to use each variant of PCA:

- **Regular PCA**: This is the standard approach to PCA and is suitable when we have access to all the data at once. Regular PCA is used when we have complete datasets that can fit into memory and can be processed efficiently.
- **Incremental PCA**: This is a variant of PCA that can handle large datasets that cannot fit into memory. Incremental PCA processes data in small batches instead of processing all the data at once. It is useful when we have streaming data or when available computational resources are limited.
- **Randomized PCA**: This is a fast approximation algorithm for PCA that provides an approximate solution with less computational effort. Randomized PCA uses random projections to reduce the dimensionality of the data and then applies regular PCA on the projected data. It is useful when we need to perform PCA on very high-dimensional data where the computation time required by regular PCA is prohibitive.
- **Random Projection**: Random projection is a simple technique in which we randomly project high-dimensional data onto a lower-dimensional space. This technique can be used as a quick-and-dirty way to

reduce the dimensionality of data when computational resources are limited or when the accuracy of the projection is not critical.

In summary, the choice of which variant of PCA to use depends on the size and characteristics of the data, as well as the computational resources available and the specific requirements of the problem at hand.

Question 3:

Does it make sense to chain two different dimensionality reduction algorithms?

Answer:

It is possible to chain two different dimensionality reduction algorithms, but whether or not it makes sense to do so depends on the specific problem and the characteristics of the data.

There are situations where chaining multiple dimensionality reduction techniques can be beneficial. For example, if the original data is very high-dimensional and contains a lot of noise or irrelevant features, we might first apply a feature selection method (such as Lasso or Random Forest Feature Importance) to reduce the number of input variables, followed by PCA to further reduce the dimensionality and extract the most important patterns in the data.

However, there are also situations where chaining multiple dimensionality reduction techniques can be detrimental. This is because each technique imposes its own assumptions and biases on the data, and these assumptions may not always be compatible with one another. For example, if one technique assumes that the data is normally distributed while another assumes non-normality, the combination of these two techniques may result in distorted or incorrect representations of the data.

Moreover, chaining multiple dimensionality reduction techniques can make it more difficult to interpret the resulting transformed data. It may be unclear which features or patterns in the original data correspond to the output of each individual technique, making it harder to explain and reason about the results.

Therefore, before chaining multiple dimensionality reduction algorithms, it is important to carefully consider the specific problem and characteristics of the data, and to evaluate the impact of combining the techniques on the performance and interpretability of the resulting models.

Question 4:

What are the main assumptions and limitations of PCA?

Answer:

PCA (Principal Component Analysis) is a popular technique for dimensionality reduction in machine learning. Like any other technique, PCA has certain assumptions and limitations that need to be considered when applying it to a dataset.

Here are the main assumptions and limitations of PCA:

Assumptions:

- **Linearity**: PCA assumes that the underlying relationships between variables in the data are linear. If the relationships are non-linear, PCA may not be able to effectively capture the variance in the data.
- **Normality**: PCA assumes that the variables in the data are normally distributed. If the variables have a skewed distribution, this can affect the ability of PCA to correctly identify the principal components.
- **Homoscedasticity**: PCA assumes that the variance of the variables in the data is constant across all values. If the variance is not constant, this

can lead to incorrect estimates of the principal components.

Limitations:

- **Orthogonality**: PCA produces orthogonal components, which may not always be the most meaningful representation of the data. In some cases, non-orthogonal components may provide a better representation of the underlying patterns in the data.
- **Variance capture**: PCA aims to capture the maximum amount of variance in the data with a limited number of components. However, the amount of variance captured by each component may not necessarily correspond to its importance or relevance to the problem at hand.
- **Interpretability**: The principal components produced by PCA are often difficult to interpret directly, especially if there are many variables in the original dataset. This can make it challenging to extract meaningful insights from the results of PCA.

In conclusion, while PCA is a powerful and widely used technique for dimensionality reduction, it makes several assumptions about the underlying structure of the data, and has certain limitations that should be taken into account when applying it to a specific problem.

Question 5:

How can clustering be used to improve the accuracy of the linear regression model?

Answer:

Clustering and linear regression are separate techniques used for different purposes. Clustering is an unsupervised technique used to group similar data points together, while linear regression is a supervised technique used to model the relationship between independent and dependent variables.

However, clustering can be used to improve the accuracy of linear regression in certain situations, such as when dealing with heteroscedastic data (data with varying variance) or non-linear relationships between variables.

Here are some ways clustering can be used to improve the accuracy of linear regression:

- **Feature engineering**: Clustering can be used to identify groups of similar features that can be combined into a single feature. This can reduce the dimensionality of the data and improve the accuracy of the linear regression model.
- **Outlier detection**: Clustering can be used to identify outliers in the data that may be affecting the accuracy of the linear regression model. By removing these outliers, the model can be made more accurate.
- **Non-linear relationships**: Clustering can be used to identify non-linear relationships between variables that may not be captured by linear regression. Once these non-linear relationships are identified, they can be incorporated into the linear regression model using polynomial regression or other non-linear regression techniques.
- **Heteroscedasticity**: Clustering can be used to identify groups of data points with similar variances. This can help address the issue of heteroscedasticity, which can lead to inaccurate predictions in linear regression.

Therefore, while clustering and linear regression are separate techniques, clustering can be used to complement linear regression and improve its accuracy in certain situations.

Question 6:

How is entropy used as a clustering validation measure?

Answer:

Entropy is a measure of the level of disorder or uncertainty in a system. In the context of clustering, entropy can be used as a validation measure to evaluate the quality of a clustering solution by measuring the degree of homogeneity within clusters and separation between clusters.

The entropy-based clustering validation measure is calculated as follows:

1) For each cluster, compute the proportion of data points that belong to each class.

2) Compute the entropy for each cluster using the formula: $H = -\sum_i p_i \log(p_i)$ where p_i is the number of data points in the ith class within the cluster.

3) Compute the weighted average entropy across all clusters using the formula: $H_{wt} = \sum (\frac{n_i}{N}) \times H_i$ where n_i is the number of data points in the ith cluster, N is the total number of datapoints, and H_i is the entropy of the ith cluster.

The resulting value of H_{wt} ranges from 0 to 1, with lower values indicating better clustering solutions. A value of 0 indicates that all datapoints within a cluster belong to the same class, while a value of 1 indicates that all classes are equally represented within each cluster.

In practice, the entropy-based clustering validation measure can be used to compare different clustering solutions and select the one with the lowest entropy value. It can also be used as a stopping criterion for hierarchical clustering algorithms, where the algorithm is stopped when the entropy value falls below a certain threshold.

Question 7:

What is label propagation? Why would you implement it, and how? (Extra Point)

Answer:

Label propagation is a semi-supervised learning technique used for classification tasks. It is based on the idea that data points that are close to each other in feature space are likely to have the same label. The goal of label propagation is to propagate labels from labeled data points to unlabeled data points in order to classify them.

The label propagation algorithm works by constructing a graph where each node represents a data point and the edges between nodes represent their similarity or proximity. The labels of the labeled data points are initially set as the "known" labels, and the algorithm propagates these labels to the unlabeled data points by taking into account the labels of neighboring data points in the graph.

There are several reasons why one might choose to implement label propagation:

- **Semi-supervised learning**: Label propagation can be used in situations when only a small portion of the data is labeled, while the majority of the data is unlabeled. In such cases, label propagation can help make use of the available labeled data to make predictions on the unlabeled data.
- **Large datasets**: Label propagation can be applied to large datasets since it scales well with the number of data points.
- **Non-linear relationships**: Label propagation can capture non-linear relationships between data points that may not be captured by linear models.

To implement label propagation, one would typically follow these steps:

1) Construct a graph based on the pairwise similarities or distances between the data points. This can be done using methods such as k-Nearest Neighbors or Gaussian Kernel functions.

2) Assign labels to the labeled data points, and initialize the labels of the unlabeled data points to some default value (such as 0)

3) Propagate the labels from the labeled data points to the unlabeled data points based on the similarities between data points in the graph. This can be done iteratively, using an update rule that takes into account the labels of neighboring data points.

4) Repeat step 3 until convergence, or until a maximum number of iterations is reached.

5) Use the propagated labels to make predictions on new data points.

In summary, label propagation is a semi-supervised learning technique that can be used to propagate labels from labeled data points to unlabeled data points in order to classify them. It can be useful in situations when only a small portion of the data is labeled or when dealing with large datasets.

Question 8:

You are going to work on the Supermarket dataset for predictive marketing. Your task is to use clustering algorithms to segment the customers into distinct groups based on their shopping behavior and demographics.

- Explore and preprocess the dataset. This may involve handling categorical variables and normalizing or scaling numerical features and feature engineering.
- Use K-means clustering to identify the optimal number of clusters. Experiment with different values of K and use metrics

such as the elbow method and silhouette score to evaluate the performance of the clustering.

- Visualize the clusters and analyze their characteristics. This may involve plotting the clusters in 2D or 3D using PCA or t-SNE.
- Experiment with other clustering algorithms such as DBSCAN or hierarchical clustering, and compare their performance with K-mean.
- Try to reduce data dimensionality using PCA before training your model, use different numbers of components and report their effects. (Extra Point)
- Interpret the results and provide insights to the store owners. What are the distinct customer segments that have been identified? How can the store owners use this information to improve their marketing strategy, product offerings, or customer experience? (Extra Point)