ML Notes without the BS

vivian sedov

December 22, 2022

Contents

1 Types of Learning

- Supervised Learning
  - Algo is trained on labeled data
  - meaning that the data has been labeled with the correct output
  - *Goal:* is to make predictions about new unseen data based on patterns learned from the training data.
  - Examples
    - Linear Regerssion
    - Logistic Regression
    - Support vector Machines
    - Classficiation problems : Binary or Multi Class
- Unsupervised Learning
  - The algo is not given any labeled data and must find the patterns and relashinships on its own.
  - *Goal:* Is to discover structure in the data and to identify relationships bsed on that structure.
  - Examples:
    - Clustering
    - Dimensionality reduction [Really cool concept]
- Reinforcement Learning
  - The learning algo is not given any labeled data and must find patterns and relationships in the data on its won.
  - *Goal :* is to discover structure, but based on its own ability to generalise and understand the current scope of what it can learn from.
  - Examples:
    - Self driving cars
    - Those small little hoovers that move around on their own. Question your self how do they map everything.

1.1 Supervised Learning

Due to this module purely focusing on this, i will pick certain info that i deem viable

Features

- Components of samples are features
- Each feature can be descrete or continuous $[0, \infty)$

Protocols used in supervised Learning There are two core protocols that are used within supervised learning: Batch and Online — Just google this, the explanation on the slides are bad.

Batch Learning In batch learning, there are two core steps, and stages that you have to work with

- Training: Exploration stage - You analyse the data, and the training set - and find some viable explanation for why it works
- Exploitation stage: Pretty much saying does this hypothesis work with repsect to what the training data had presented.

Definition 1.1 – Induction Induction is a form of learning that involves making generalizations based on specific examples or observations. In induction, the goal is to learn a general rule or model that can be applied to new, unseen data. This is the approach used in many supervised learning algorithms, where the algorithm is trained on a labeled dataset and then makes predictions about new, unseen data based on the patterns learned from the training data.

has a model

Definition 1.2 – Transduction Transduction is a form of learning that involves making predictions about a specific instance based on the available information. In transduction, the goal is to make a prediction about a specific case rather than learning a general rule that can be applied to new data. This is the approach used in many unsupervised learning algorithms, where the algorithm is given a dataset but is not told what the correct output should be. Instead, the algorithm must find patterns and relationships in the data on its own and use that information to make a prediction about a specific instance.

No model

The main difference between induction and transduction is that induction involves learning a general rule or model that can be applied to new data, while transduction involves making a prediction about a specific instance based on the available information.

2 Introduction To NN

Definition 2.1 – Nearest Neighbor Algorithm The nearest neighbor algorithm is a method for classification that assigns a new data point to the class of the nearest training data point. This algorithm is based on the idea that similar data points are likely to belong to the same class.

Within binary classfication problems, the label space is often taken to be 0 or 1.

2.1 How does it work ?

Here is a mathematical example of how the nearest neighbor algorithm works: Suppose we have a training set of data points with known classes, represented as a matrix $X \in \mathbb{R}^{n \times m}$, where each row corresponds to a data point and each column corresponds to a feature. The class labels for the data points are stored in a vector $y \in \mathbb{R}^n$.

Now, suppose we have a new data point, represented as a row vector $x \in \mathbb{R}^m$, that we want to classify. To classify this point using the nearest neighbor algorithm, we need to find the data point in the training set that is closest to x. We can measure the distance between x and each data point in the training set using a distance metric, such as the Euclidean distance:

$$d(x, X_i) = \sqrt{\sum_{j=1}^{m} (x_j - X_{i,j})^2}$$

where X_i is the i-th row of the matrix X, and x_j and $X_{i,j}$ are the j-th elements of the vectors x and X_i, respectively.

Once we have computed the distances between x and all the data points in the training set, we can find the nearest neighbor by selecting the data point with the smallest distance:

$$\hat{y} = y_i, where\ i = \operatorname{argmin}_{j=1}^{n} d(x, X_j)$$

In this equation, $\hat{y}$ is the predicted class for the new data point x, and y_i is the class of the nearest neighbor.

2.1.1 Step by step guide on how NN works

Suppose we have the following training set of data points, where each row corresponds to a data point and each column corresponds to a feature:

$$X = \begin{bmatrix} 1 & 1\,2 & 2\,3 & 3\,4 & 4 \end{bmatrix}$$

The class labels for the data points are stored in the following vector:

$$y = \begin{bmatrix} 0\ 1\ 0\ 1 \end{bmatrix}$$

Now, suppose we have a new data point that we want to classify, represented as the following row vector:

$$x = \begin{bmatrix} 1.5 & 1.5 \end{bmatrix}$$

To classify this point using the nearest neighbor algorithm, we need to find the data point in the training set that is closest to x. We can measure the distance between x and each data point in the training set using the Euclidean distance, as shown in the previous equation.

For example, the distance between x and the first data point in the training set is:

$$d(x, X_1) = \sqrt{(1.5-1)^2 + (1.5-1)^2} = \sqrt{0.5^2 + 0.5^2} = 0.7071067811865475$$

We can compute the distances between x and all the other data points in the training set in a similar way. Once we have computed the distances, we can find the nearest neighbor by selecting the data point with the smallest distance:

$$\hat{y} = y_i, where\ i = \text{argmin}_{j=1}^{n} d(x, X_j)$$

In this case, the nearest neighbor is the first data point, since it has the smallest distance to x. Therefore, the predicted class for the new data point x is $\hat{y} = y_1 = 0$.

This is transduction : No model was formed

2.2 KNN

KNN - K - nearest Neighbors - is a non parametric, instance based supervised learning algorithm. It is used for both classfication and gression tasks.

The basic idea of KNN is to use the information from the K nearest Nehigbors of a given data point and make a prediction, given the current set of data that we work with.

Definition 2.2 Mathematical definition of KNN is defined :

$$\hat{y} = \frac{1}{k}\sum_{i=1}^{k} y_i$$

For example, consider a dataset with two labels, 0 and 1, and a feature set containing two features, x_1 and x_2. Let's say we want to predict the label of a new data point with feature values $x_1 = 3$ and $x_2 = 4$. We can use KNN to make this prediction by finding the K nearest neighbors to the new data point and averaging their labels.

x$_1$	x$_2$	label
2	3	0
3	5	1
4	2	0

Let $K = 3$:

In this case, the predicted label for the new data point would be $\hat{y} = \frac{1}{3}(0+1+0) = 13 = 0.33$ We then look at how close the values are for each point, and compare them , in this instance it would be 0.

Cosine Distance, Eulcidian Distance KNN typically uses the Euclidean distance as the measure of distance between data points. The Euclidean distance between two points in a Euclidean space is defined as the square root of the sum of the squares of the differences between the coordinates of the points.

For example, in two-dimensional space, the Euclidean distance between points (x_1, y_1) and (x_2, y_2) is given by:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Other distance measures, such as the Manhattan distance or the cosine similarity, can also be used with KNN, but the Euclidean distance is the most commonly used.

It is important to note that the choice of distance measure can have a significant impact on the performance of KNN, so it is important to consider which distance measure is most appropriate for the given data and task.

Better Example Consider the following data:

x$_1$	x$_2$	label
1	2	0
2	3	1
3	1	0
4	5	1
5	4	0

Suppose we want to predict the label of a new data point with feature values $x_1 = 2.5$ and $x_2 = 3.5$, and we want to use the Euclidean distance to find the nearest neighbors. Which is normally used.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

For example, to find the distance between the new data point and the first data point in the table, we would use:

$$d = \sqrt{(2.5 - 1)^2 + (3.5 - 2)^2} = \sqrt{1.5^2 + 1.5^2} = 2.12$$

We can repeat this process for each of the other data points to find the distances between the new data point and all of the existing data points.

x$_1$	x$_2$	label	Distance
1	2	0	2.12
2	3	1	0.71
3	1	0	1.80
4	5	1	2.83
5	4	0	2.24

We can then sort the data points by distance and select the K nearest neighbors (where K is the number of neighbors that we want to use in the prediction). For example, if we set K=3, the three nearest neighbors to the new data point would be the second, third, and fifth data points in the table.

Using these nearest neighbors, we can then make a prediction for the label of the new data point using the KNN formula:

$$\hat{y} = \frac{1}{K} \sum_{i=1}^{K} y_i = \frac{1}{3}(1 + 0 + 0) = \frac{1}{3} = 0.33$$

Since this value is closer to 0 than 1, we would predict that the label for the new data point is 0.

3 Conformal Prediction

Definition 3.1 Conformal prediction is a method of making predictions that is designed to provide a measure of confidence or uncertainty in the prediction. It is based on the idea of creating a prediction region, or "conformal prediction set," around a new data point, which contains a range of possible values for the prediction. The prediction is considered "valid" if the true value of the data point falls within this region.

The mathematical formula for conformal prediction is as follows:

$$P(\text{true label} \in \text{prediction set}) \geq 1 - \alpha$$

Here, $P(\text{true label} \in \text{prediction set})$ is the probability that the true label of the data point falls within the prediction set, and α is a user-specified confidence level, typically set to a small value such as 0.1 or 0.05.

Conformal prediction is used to provide a measure of uncertainty in predictions, which can be useful in situations where it is important to know how confident we are in a prediction. For example, in medical diagnosis, it may be important to know how certain we are that a patient has a particular disease, in order to make informed treatment decisions. In such cases, conformal prediction can be used to provide a range of possible diagnoses, along with a corresponding confidence level.

Conformal prediction is also used in situations where the data is noisy or uncertain, as it can provide a way to account for this uncertainty in the predictions. This is especially useful when working with complex or high-dimensional data, where the relationship between the features and the target variable may be difficult to model accurately.

Properties

- Guaranteed validity - with respect to probability of error
- Commonplace in stats - Confidence Intervals, prediction intervals.

3.1 Explanation of Algo

The high level idea behind conformal prediction is to create a prediction set or region, around a new data point that contains a rage of possible values for the prediction. This prediction set is constructed using the conformity scores and the p_value of the data points of the training and test values

- Given a training set of data points $z_1, z_2, ..., z_n$ and a new test sample $x*$, we first compute the conformity scores, or p-values, for each possible label y for the test sample.
- The conformity score for a particular label y is calculated as the number of training data points with conformity scores less than or equal to the conformity score of the test sample, divided by the total number of training data points plus one. This is represented by the formula:
$$p(y) = \frac{i = 1, ..., n+1 | \alpha_y^i \leq \alpha_y^{n+1}}{n+1}$$

where $\alpha_y^1, ..., \alpha_y^n, \alpha_y^{n+1}$ are the conformity scores corresponding to $z_1, ..., z_n, (x*, y)$.

- Given a significance level $\epsilon > 0$ (our target probability of error), we can then compute the corresponding prediction set $\Gamma_\epsilon = y \in Y | p(y) > \epsilon$. This is the set of labels that are considered "valid" predictions for the test sample, as they have a probability of error less than or equal to ϵ.
- To make a prediction for the test sample, we can choose the label with the highest conformity score, or we can select the label with the highest conformity score that falls within the prediction set Γ_ϵ. This allows us to balance the trade-off between accuracy and confidence in the prediction.

Overall, the goal of conformal prediction is to provide a measure of confidence or uncertainty in the prediction, by constructing a prediction set that contains a range of possible values for the prediction, rather than a single point estimate. This can be useful in situations where it is important to know how confident we are in a prediction, or where the data is noisy or uncertain.

3.1.1 Special Cases

Most cases it is important to understand how strange our data is. This refers to how different a data point is from the rest of the training data. A dta point with a high degree of strangeness is one that is significantly different from the other data points in the training set.

The special cases are of the following: Because the slides are shit

- If the test sample has the highest conformity score. Of all the data points in the training set. This means that it is the most strange data point. In this case the conformity score of the test sample will be $\frac{1}{n+1}$ Which is the smallest possible value.
- If the test sample has the second highest conformity score of all the data points in the training set, this means that it is the second most strange data. In this case it will have a conformity score of the test $\frac{2}{n+1}$
- If the test sample has the lowest conformity score of all the data points in the training set, this means that it is the most "conforming" data point. In this case, the conformity score of the test sample will be 1.

These special cases illustrate how the conformity score of the test sample is influenced by its relationship to the conformity scores of the training data points. In general, the higher the conformity score of the test sample, the more "strange" it is relative to the training data, and the lower the conformity score, the more "conforming" it is. This is important to consider when constructing the prediction set, as the conformity scores of the test sample and the training data points are used to determine which labels are considered "valid" predictions for the test sample.

3.1.2 Assumptions towards Conformity

The assumptions of machine learning refer to the assumptions that are made about the data and the learning process when building and using machine learning models. These assumptions can vary depending on the type of model and the specific problem being solved, but they generally include assumptions about the structure of the data, the relationship between the features and the target variable, and the nature of the learning process itself.

Conformal prediction is a method of making predictions that is designed to provide a measure of confidence or uncertainty in the prediction. It is based on the idea of creating a prediction region, or "conformal prediction set," around a new data point, which contains a range of possible values for the prediction. The prediction is considered "valid" if the true value of the data point falls within this region.

Conformal prediction based on nearest neighbors is a specific approach to conformal prediction that uses the k-nearest neighbors (KNN) algorithm to find the nearest neighbors of a new data point and use their labels to make a prediction. The conformity scores of the data points are calculated using the KNN distances, and the prediction set is constructed based on these conformity scores.

The validity and efficiency of conformal predictors refer to the properties of the prediction set that is constructed around a new data point. Validity refers to the probability that the true label of the data point falls within the prediction set, which is typically set to a user-specified value such as 0.1 or 0.05. Efficiency refers to the size of the prediction set, with smaller prediction sets being considered more efficient.

The validity of conformal predictors is ensured by the property that the predictor makes a mistake with probability at most ϵ, provided that the labeled samples are independently and identically distributed (IID). This means that the probability of the true label falling outside of the prediction set is at most ϵ.

The efficiency of conformal predictors can be traded off against their validity. In general, smaller prediction sets are more efficient, but they may also be less accurate, as they may not contain a wide enough range of possible values for the prediction. On the other hand, larger prediction sets may be more accurate, but they may also be less efficient. It is important to consider both validity and efficiency when using conformal prediction, in order to find the optimal balance between these two properties.

3.2 Conformity Measure for KNN (K=1)

There are several different conformity measures that can be used with the 1-nearest neighbor (1-NN) algorithm for conformal prediction. Here is an explanation of two of these measures:

The distance to the nearest sample of a different class: This conformity measure is based on the idea that a data point that is more similar to data points from a different class is more likely to be classified correctly. In other words, if the nearest neighbor of a data point belongs to a different class, this may be a good indication that the data point is correctly classified.

The conformity score for this measure can be calculated as the distance between the data point and its nearest neighbor of a different class. For example, if the data point is classified as class A and its nearest neighbor is classified as class B, the conformity score would be the distance between the two points.

This conformity measure can be represented mathematically as follows:

$$\alpha = d(\mathbf{x}, \mathbf{x}_{\text{NN}}^{\text{diff}})$$

where α is the conformity score, $\mathbf{x}$ is the data point, and $\mathbf{x}_{\text{NN}}^{\text{diff}}$ is the nearest neighbor of a different class.

One over the distance to the nearest sample of the same class: This conformity measure is based on the opposite idea, that a data point that is more similar to data points from the same class is more likely to be classified correctly. In other words, if the nearest neighbor of a data point belongs to the same class, this may be a good indication that the data point is correctly classified.

The conformity score for this measure can be calculated as the reciprocal of the distance between the data point and its nearest neighbor of the same class. For example, if the data point is classified as class A and its nearest neighbor is also classified as class A, the conformity score would be the reciprocal of the distance between the two points.

This conformity measure can be represented mathematically as follows:

$$\alpha = \frac{1}{d(\mathbf{x}, \mathbf{x}_{\text{NN}}^{\text{same}})}$$

where α is the conformity score, $\mathbf{x}$ is the data point, and $\mathbf{x}_{\text{NN}}^{\text{same}}$ is the nearest neighbor of the same class.

Both of these conformity measures are based on the distance between the data point and its nearest neighbors, with the first measure using the distance to the nearest neighbor of a different class and the second measure using the distance to the nearest neighbor of the same class. These measures can be useful for assessing the confidence of the