Tutorial: Applications of KNN-variants in Classification, Regression, and Data Generation

November 24, 2023

KNN

k-Nearest Neighbour (KNN) algorithms are part of the supervised learning methodologies that utilize a distance metric to classify, predict, or generate new samples. The algorithm stores a set of input-output pairs which define clusters based on the number of unique classes. The local decision boundary between clusters is defined by a distance metric between samples of different labels. The boundaries are piece-wise linear class borders parameterized by all inputs. Given a new input sample without a label, the sample is classified as part of the cluster with the smallest distance between the current input's features and the features of all the samples in the cluster Cover [1967].

Although the algorithm is straightforward, it requires expensive storage and computation, because the boundaries between clusters are recalculated with each new sample, which requires looping through all the $N-1$ samples in the dataset to calculate the distances. The algorithm is also sensitive to outliers, which can greatly weigh the cluster's distance metric.

KNN can be applied as a supervised learning method in classification tasks: given a dataset with multi-dimensional inputs and categorical labels, it classifies unlabelled data with an existing label; regression tasks: given a dataset with multi-dimensional inputs and continuous labels, it predicts unlabelled data with a continuous value; sample generation tasks: given a dataset with multi-dimensional inputs and either continuous or categorical labels, it generates new samples within the cluster.

This project is in part a theoretical introduction to the three applications of the KNN algorithm formulated in this report and in part a Python tutorial of the three applications on an imbalanced dataset classification problem and on an artificially generated dataset prediction problem. The code for the tutorial can be found at https://github.com/SarahEmaAllam/ContemporaryStats.

Let the dataset D be comprised of input vectors $x_i^n \in \mathbb{R}^n$, where i is the index and n is the dimensionality of the vector, with a corresponding output label $y_i \in \mathbb{R}$. If S_{x_i} is the set of the nearest data vectors (neighbours) of x_i, then $S_{x_i} \subseteq D$ so that $|S_{x_i}| = k$. Then, we can define every neighbour vector in S_{x_i} as vectors within a certain distance of x, and the non-neighbouring vectors as part of the set $D \setminus S_{x_i}$ so that $\forall (x_h, y_h) \in D \setminus S_{x_i}$ when $h \neq j, i$. The distance between x_i and x_{ij} is defined as:

$$d(x_i, x_h) \geq \max_{(x_{ij}, y_{ij}) \in S_{x_i}} d(x_i, x_{ij})$$

The label y_i of x_i is then defined as the most frequent label $y_{ij} : (x_{ij}, y_{ij}) \in S_{x_i}$. The distance d is usually the Minkowski distance:

$$d(x_i, x_{ij}) = (\sum_{j=1}^{k} |x_i - x_{ij}|^p)^{1/p}$$

Particular cases of Minkowski distance are the Manhattan distance with $p = 1$ and the Euclidean distance with $p = 2$.

As an application for the KNN to demonstrate its properties with data analysis, we will use the DREAM challenge Aghaeepour N and Scheuermann [2011]. DREAM is a supervised classification problem consisting of 359 total input samples, with only 179 labeled samples, and 186 input features. Therefore, the training set is 179 samples and the test set is 180 samples. The problem is to correctly classify the remaining 180 samples as patients with Acute Myeloid Leukemia (AML) or healthy patients (a binary problem). In addition, the dataset is highly imbalanced, with a majority of the samples pertaining to the healthy class. There are only 20 AML samples in the 180 test set.

Classification

KNN classification assumes that the class labels y for the inputs x are categorical, and therefore classifies a new input x_i with the predicted class label $\hat{y}_i$ which is the majority of the labels in the cluster to which x_i was assigned.

Algorithm 1 KNN Classification Algorithm

```
Ensure: class labels $\hat{y}$
Require: training data $(x_j, y_j)_{j=1...N}$, test data $\hat{x}_{ii=1...N}$, neighbours parameter k, $d(x_i, x_j)$ distance metric
    foreach $\hat{x}_i$ do
        foreach $\hat{x}_j$ do
            $d(x_i, x_j) = (\sum_{j=1}^{k} |x_i - x_j|^p)^{1/p}$
        select $S_{x_i} = (x_{ij}, y_{ij})_{j=1...k}$ with $k$ smallest distances $d(x_i, x_j)$
        $\hat{y}_i \leftarrow mode(y_{ij} : (x_{ij}, y_{ij}) \in S_{x_i})$
    =0
```

Regression

KNN regression assumes that the class labels y for the inputs x are continuous, and therefore predicts for a new input x_i a value $\hat{y}_i$ which is the average $\hat{y}_{ij}$ values in the cluster to which x_i was assigned.

Algorithm 2 KNN regression Algorithm

```
Ensure: class labels $\hat{y}$
Require: training data $(x_j, y_j)_{j=1...N}$, test data $\hat{x}_{ii=1...N}$, neighbours parameter k, $d(x_i, x_j)$ distance metric
    foreach $\hat{x}_i$ do
        foreach $\hat{x}_j$ do
            $d(x_i, x_j) = (\sum_{j=1}^{k} |x_i - x_j|^p)^{1/p}$
        select $S_{x_i} = (x_{ij}, y_{ij})_{j=1...k}$ with $k$ smallest distances $d(x_i, x_j)$
        $\hat{y}_i \leftarrow \frac{1}{2}(\sum_{j=1}^{k} y_{ij} \in S_{x_i})$
```

Sample Generation

KNN can be used for sample generation with Synthetic Minority Over-sampling Technique (SMOTE). Instead of sampling with replacement, SMOTE creates a synthetic sample based on the cluster features in order to increase the number of samples in the minority class and therefore balance the classes. A new synthetic sample is generated by randomly choosing a sample within a class and taking the difference between each feature vector of the chosen sample and one of the nearest neighbour. The difference in the feature vectors is multiplied with a random value between 0 and 1 in order to produce variability within the synthetic data Chawla et al. [2002].

Algorithm 3 SMOTE with KNN Algorithm

```
Ensure: $(N/100) * T$ synthetic samples $(\hat{x}_t, \hat{y}_t)$
Require: minority class data $(x_t, y_t)_{t=1...T}$, neighbours parameter k, $N\%$ percentage of SMOTE data to be generated
    $n \leftarrow$ features of $x_t$
    while $N \neq 0$ do
        foreach $\hat{x}_t$ do
            $karray \leftarrow k$ nearest neighbour of $x_t$
            $x_{tj} \leftarrow random(karray, 1)$
            foreach $n$ do
                $dif \leftarrow x_t^n - x_{tj}^n$
                $var \leftarrow random([0, 1], 1)$
                $\hat{x_t}^n \leftarrow x_t^n + var * dif$
            $N \leftarrow N - 1$
            $\hat{y}_t \leftarrow mode(y_{tj} : (x_{tj}, y_{tj}) \in S_{x_t})$
    =0
```

References

The FlowCAP Consortium The DREAM Consortium Holger Hoos Time R Mosmann Ryan Brinkman Raphael Gottardo Aghaeepour N, Finak G and Richard Scheuermann. Dream6 - flowcap2 molecular classification of acute myeloid leukaemia challenge. 1:228–238, 2011.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16 (1):321–357, jun 2002. ISSN 1076-9757.

and Hart Peter Cover, Thomas. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions*, 13:21–27, 1967.