CHAPTER 3

Research Designs and Methods

This chapter explains the proposed research methodologies for incorporating uncertainty into string grammar k-nearest neighbor. Nevertheless, we implemented 7 algorithms for the proposed research methodologies.

3.1 String Grammar Fuzzy K-Nearest Neighbor

Given $i = 1, 2, \ldots, C$, $a = 1, 2, \ldots, K$, $j = 1, 2, \ldots, N$, $p = 1, 2, \ldots, C$, and $q = 1, 2, \ldots, C$, where C is the number of classes and K is the number of nearest neighbors, now we are ready to create 7 algorithms , namely sgFKNN1, sgFKNN2, sgFKNN3, sgFKNN4, sgFKNN5, sgFKNN6, and sgFKNN7, as follows.

3.1.1 sgFKNN1

In the first algorithm, sgFKNN1, the K closest strings of objects are identified. The membership value in [24] is modified by using the Levenshtein distance instead of the Euclidean distance and used as a membership value of string $\mathbf{x}$ in class i as follows:

$$u_i(\mathbf{x}) = \frac{\sum_{a=1}^{K} u_{ia} \left[\dfrac{\exp\left(-m\sqrt{C}Lev\left(\mathbf{x}-\mathbf{x}_a^q\right)\right)}{\beta} \right]}{\sum_{a=1}^{K} \left[\dfrac{\exp\left(-m\sqrt{C}Lev\left(\mathbf{x}-\mathbf{x}_a^q\right)\right)}{\beta} \right]} \tag{3.1}$$

where u_{ia} is the membership value of training string of object $\mathbf{x}_a^q$ in class i

m is the fuzzifier

C is the number of classes in the training dataset.

β is modified from [24] and calculated as

$$\beta = \frac{\sum_{q=1}^{C}\sum_{a=1}^{N_q} Lev\left(\mathbf{x}_a^q, \mathbf{x}_{med}\right)}{\sum_{q=1}^{C} N_q} \tag{3.2}$$

where the median string $\mathbf{x}_{med}$ in a set of strings $\mathbf{x}$ can be calculated as [17, 29]

$$\mathbf{x}_{med} = \underset{a \in \mathbf{X}}{\arg\min} \sum_{k=1}^{N_1+N_2+\ldots+N_C} Lev(\mathbf{x}_a, \mathbf{x}_k) \tag{3.3}$$

Then, the decision rule is as following:

$$x \text{ is assigned to class } i \text{ if } u_i(\mathrm{x}) > u_a(\mathrm{x}) \text{ for } a \neq i. \tag{3.4}$$

In our experiment section, since we know the class that the training string of object $\mathbf{x}_a^q$ represents, we set u_{ia} = 1 for $\mathbf{x}_a^q$ in class q and 0 for all the other classes. The time complexity of sgFKNN1 is approximately $O\left(N^2 \log N\right)$.

Nevertheless, we also set u_{ia} on the fuzzy initialization in stardard dataset represent. We used equation 2.7 for fuzzy initialization.

3.1.2 sgFKNN2

Next, we create another algorithm called sgFKNN2 which is implemented with the membership value modified from [8] as

$$u_i\left(\mathbf{x}\right) = \frac{\sum_{a=1}^{\mathrm{K}} u_{ia}\left(1 \Big/ Lev\left(\mathbf{x} - \mathbf{x}_a^q\right)\right)^{2/m-1}}{\sum_{a=1}^{\mathrm{K}}\left(1 \Big/ Lev\left(\mathbf{x} - \mathbf{x}_a^q\right)\right)^{2/m-1}}. \tag{3.5}$$

The parameters m and the membership u_{iq} are set in the same way as in equation 3.1. The time complexity of sgFKNN2 is approximately $O\left(N^2 \log N\right)$.

3.1.3 PEC (Possibilistic Entropy based Clustering) - sgFKNN3

We also implemented our sgFKNN3 with the membership value modified from [29] as

$$u_i(\mathbf{x}) = \frac{\sum_{a=1}^{K} u_{ia}\left(e^{-\beta_i Lev(\mathbf{x}-\mathbf{x}_a^q)}\right)}{\sum_{a=1}^{K}\left(e^{-\beta_i Lev(\mathbf{x}-\mathbf{x}_a^q)}\right)} \tag{3.6}$$

where β_i is computed

$$\beta_i = (1+\alpha)\frac{\sum_{j=1}^{N} u_{ij}\left(Lev(\mathbf{x}-\mathbf{x}_{ij})\right)^2}{\sum_{j=1}^{N} u_{ij}\left(Lev(\mathbf{x}-\mathbf{x}_{ij})\right)^4} \tag{3.7}$$

where $\alpha \geq 0.5$

u_{ij} is the membership value of the training string of object $\mathbf{x}_j$ in class i

$\mathbf{x}_{ij}$ is the string of object j in class i.

Note that one constraint is that $\sum_{j=1}^{N} u_{ij}\left(Lev(\mathbf{x}-\mathbf{x}_{ij})\right)^4 > 0$, i.e. the divider cannot be zero.The time complexity of sgFKNN3 is approximately $O\left(N^2 \log N\right)$.

3.1.4 VFC (Vector Fuzzy C-Means) - sgFKNN4

Next, we implemented the forth algorithm, sgFKNN4, with the membership value modified from [30] as

$$u_i(\mathbf{x}) = \frac{\sum_{a=1}^{K} u_{ia}\left(1 \Big/ \left(\sum_{p=1}^{C}\frac{Lev(\mathbf{x}-\mathbf{x}_{med\,a}^{\;i})}{Lev(\mathbf{x}-\mathbf{x}_{med\,a}^{\;p})}\right)^{2/m-1}\right)}{\sum_{a=1}^{K}\left(1 \Big/ \left(\sum_{p=1}^{C}\frac{Lev(\mathbf{x}-\mathbf{x}_{med\,a}^{\;i})}{Lev(\mathbf{x}-\mathbf{x}_{med\,a}^{\;p})}\right)^{2/m-1}\right)}\; ; K \leq C \tag{3.8}$$

where $\mathbf{x}_{med\,a}^{\;i}$ is the median in class i when a = 1 to K

C is the number of classes then $K \le C$.

Note that one constraint is that $\left(\sum_{p=1}^{C} \frac{Lev(\mathbf{x} - \mathbf{x}_{med\,a}^{\;i})}{Lev(\mathbf{x} - \mathbf{x}_{med\,a}^{\;p})} \right)^{2/m-1} \neq 0$, i.e. the divider cannot be zero. The time complexity of sgFKNN4 is approximately $O(N \log N)$.

3.1.5 NFE (New Fuzzy Entropy) – sgFKNN5

The fifth algorithm, sgFKNN5, is implemented with the membership value modified from [31] as

$$u_i(\mathbf{x}) = \frac{\sum_{a=1}^{K} u_{ia} \left(\sum_{p=1}^{C} e^{-\frac{1}{\gamma}\left(Lev\left(\mathbf{x} - \mathbf{x}_{med\,a}^{\;p}\right) - Lev\left(\mathbf{x} - \mathbf{x}_{med\,a}^{\;i}\right)\right)} \right)}{\sum_{a=1}^{K} \left(\sum_{p=1}^{C} e^{-\frac{1}{\gamma}\left(Lev\left(\mathbf{x} - \mathbf{x}_{med\,a}^{\;p}\right) - Lev\left(\mathbf{x} - \mathbf{x}_{med\,a}^{\;i}\right)\right)} \right)} \; ; K \le C \tag{3.9}$$

Note that one constraint is that $\sum_{p=1}^{C} e^{-\frac{1}{\gamma}\left(Lev\left(\mathbf{x} - \mathbf{x}_{med\,a}^{\;p}\right) - Lev\left(\mathbf{x} - \mathbf{x}_{med\,a}^{\;i}\right)\right)} \neq 0$, i.e. the divider cannot be zero.

where $\gamma = 1$ (the degree of fuzziness) is a weighting exponent used for controlling the degree of fuzziness and the membership function same the FCM or used for controlling the compromise between the intra-cluster scattering error and the fuzzy entropy. The time complexity of sgFKNN5 is approximately $O(N \log N)$.

3.1.6 RGB (Rule Generation Based) - sgFKNN6

For the sixth algorithm, sgFKNN6 is implemented with the membership value modified from [32] , i.e.,

$$u_i(\mathbf{x}) = \frac{\sum_{a=1}^{K} u_{ia} \left[\exp\left(-\frac{(Lev(\mathbf{x} - \mathbf{x}_{med\,a}^{\;\;i}))^2}{2(\beta)^2} \right) \right]}{\sum_{a=1}^{K} \left[\exp\left(-\frac{(Lev(\mathbf{x} - \mathbf{x}_{med\,a}^{\;\;i}))^2}{2(\beta)^2} \right) \right]} \quad ; K \le C \qquad (3.10)$$

and β can be set the same way as in equation 3.2. The time complexity of sgFKNN6 is approximately $O(N \log N)$.

3.1.7 PCMed – sgFKNN7

Finally, we implemented sgFKNN7 with the membership value modified from [33] , i.e.,

$$u_i(\mathbf{x}) = \frac{\sum_{a=1}^{K} u_{ia} \left[\frac{1}{1 + \left(\frac{Lev(\mathbf{x} - \mathbf{x}_a^q)}{\eta_i} \right)^{\frac{1}{m-1}}} \right]}{\sum_{a=1}^{K} \left[\frac{1}{1 + \left(\frac{Lev(\mathbf{x} - \mathbf{x}_a^q)}{\eta_i} \right)^{\frac{1}{m-1}}} \right]} \quad ; \quad \eta_i \ne 0 \quad \forall i \qquad (3.11)$$

where $$\eta_i = P \frac{\sum_{a=1}^{K} u_{ia}^m Lev(\mathbf{x} - \mathbf{x}_a^q)}{\sum_{a=1}^{K} u_{ia}^m} \quad ; \sum_{a=1}^{K} u_{ia}^m \ne 0 \quad \text{and} \quad P = 1 \qquad (3.12)$$

The time complexity of sgFKNN7 is approximately $O\left(N^2 \log N\right)$.