Project Machine Learning

Effect of feature weighting in Fuzzy-Rough nearest neighbour classification algorithms

Victor Miclotte and Bram Verbeke

November 2017

Fuzzy Rough Set Theory

- Fuzzy-part
- Rough-part
- Combining those two

Feature Weights

- Calculating weights

TODO

- Implementation
- Calculating feature weights
- Performance comparison

Fuzzy set theory

Suppose we want to make a set of all real numbers close to 10.

We could suggest a function:

$$\begin{aligned} A:\ & \mathbb{R} \mapsto \{0,1\} \\ & x \mapsto 1 \text{ if } x \in [9,11] \\ & x \mapsto 0 \text{ else} \end{aligned}$$

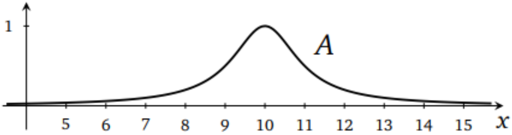
Problem:

- There is no difference between 9 and 9.5.
- But there is a huge difference between 8.99 and 9.

Fuzzy set theory

Solution:

$$A : \mathbb{R} \mapsto [0, 1]$$



We will call this a **membershipfunction**.

Rough set theory

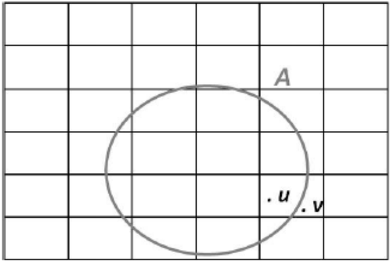
Problem:

	feature 1	feature 2	feature 3	feature 4	eyecolor
Victor	x_1	x_2	x_3	x_4	brown
Bram	x_1	x_2	x_3	x_4	green

We say this data is **incomplete**

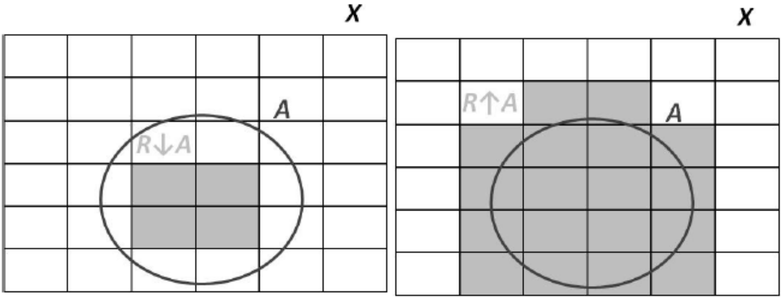
Rough set theory

Problem: Incomplete data



Rough set theory

Solution:



Lower approximation $R \downarrow A$
Upper approximation $R \uparrow A$

Fuzzy Rough set theory

Now we combine the two new concepts.

We grant each object a degree of membership to the lower and upper approximation.

Algorithm 3: The fuzzy-rough nearest neighbour algorithm

Input: X, the training data; $\mathcal{C}$, the set of decision classes; y, the object to be classified

Output: Classification for y

```
begin
    N ← getNearestNeighbours(y,K)
    τ ← 0, Class ← ∅
    foreach C ∈ 𝒞 do
        if ((R↓C)(y) + (R↑C)(y))/2 ≥ τ then
            Class ← C
            τ ← ((R↓C)(y) + (R↑C)(y))/2
        end
    end
    output Class
end
```

Calculating weights: 1

1. Suppose we leave one feature out of the data.
2. Calculate the accuracy.
3. If the accuracy is high, the left-out feature won't be very important.
4. Let's give this feature the weight 1 - acc

Calculating weights: 2

1. Suppose we take only one feature out of the data.
2. Calculate the accuracy (acc) from this one feature.
3. If the accuracy is high, the feature will be very important.
4. Let's give this feauture the weight acc.

Pros and cons: 1 & 2

Pros

1. Easy to implement and to understand
2. Works well for a small amount of features

Cons

1. Sometimes a combination of features gives a lot of information while one of those features on its own doesn't
2. Many features will require a lot of iterations of the algorithm
3. Correlated and redundant features will get similar weights while we might want to only use one of those features (feature selection)

Solutions

1. Consider multiple features at once to reduce the amount of iterations of the algorithm
2. Use feature selection to avoid redundancy

Calculating weights: 3 (Entropy)

1. Average amount of information produced by a probabilistic stochastic source of data
2. $H(X) := E\left[-\log(P(X))\right] = -\sum_{x} P(x) \log P(x)$
3. $H(X|Y) := \sum_{x,y} P(x,y) \log \frac{P(x,y)}{P(y)}$
4. Information gain:

$$H(C) - H(C|A) =$$
$$\sum_{a} P(a) \sum_{c} P(c|a) \log P(c|a) - \sum_{c} P(c) \log P(c)$$

Calculating weights: 4

1. Kullback-Leibler measure $\mathcal{KL}(C|a_{ij}) = \sum_c P(c|a_{ij}) \log \frac{P(c|a_{ij})}{P(c)}$
2. Weight of feature i, $wavg(i)$, is weighted average of the KL measures across the feature values.
3. $wavg(i) = \sum_j \frac{\#(a_{ij})}{N} \mathcal{KL}(C|a_{ij})$
4. $\#(a_{ij})$ is the number of instances that have value a_{ij} for feature i
5. N is the total number of training instances

Calculating weights: 5

1. Using a genetic algorithm, we get the optimal weight vector.

TODO: Implementation

1. Programming language: java
2. Making use of the WEKA java library

TODO: Calculating feature weights

1. Implement feature weighting methods
2. Look for more feature weighting techniques

TODO: Performance comparison

1. Compare the accuracy of the algorithms with and without every type of feature weighting
2. Look for other relevant performance measurements to compare the algorithms