Assignment 8,9 + Classifiers

(SNLP Tutorial 9)

Vilém Zouhar, Awantee Deshpande, Julius Steuer

22nd, 24th June

Assignment 8

- Exercise 1: Feature Selection (DF, PMI)
- Exercise 2: χ^2
- Exercise 3: Author identification
- Bonus: Features for clustering

Decision Trees

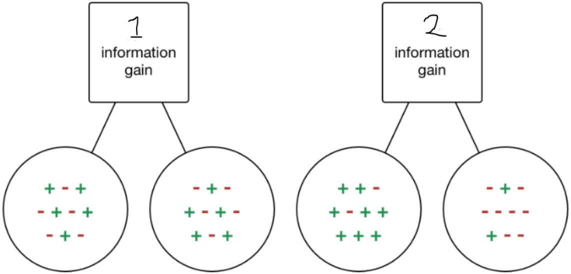
- Can be used for classification as well as regression
- Algorithm

```
function DECISION-TREE-LEARNING(examples, attributes, parent_examples) returns a tree
    if examples is empty then return PLURALITY-VALUE(parent_examples)
    else if all examples have the same classification then return the classification
    else if attributes is empty then return PLURALITY-VALUE(examples)
    else
        $A \leftarrow \underset{a \in attributes}{\mathbf{argmax}}$ IMPORTANCE(examples)
        tree ← a new decision tree with root test $A$
        for each value $v_k$ of $A$ do
            $exs \leftarrow \{e : e \in examples$ and $e.A = v_k\}$
            subtree ← DECISION-TREE-LEARNING(exs, attributes − A, examples)
            add a branch to tree with label $(A = v_k)$ and subtree subtree
        return tree
```

What is plurality value?
What is importance?

Decision Trees - Questions

- Which of the 2 splits has a better information gain?



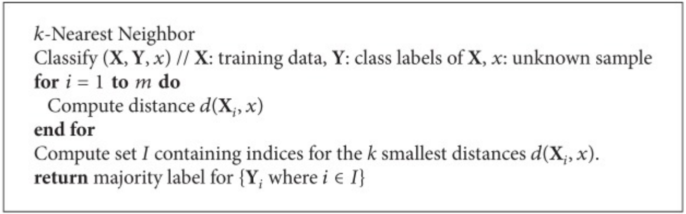
- What are the pros and cons of decision trees?
- How to avoid overfitting?

Naïve Bayes

TODO

kNN

- Algorithm



```
k-Nearest Neighbor
Classify (X, Y, x) // X: training data, Y: class labels of X, x: unknown sample
for $i = 1$ to $m$ do
  Compute distance $d(\mathbf{X}_i, x)$
end for
Compute set $I$ containing indices for the $k$ smallest distances $d(\mathbf{X}_i, x)$.
return majority label for $\{\mathbf{Y}_i \text{ where } i \in I\}$
```

Source:https://www.researchgate.net/figure/Pseudocode-for-KNN-classification_fig7_260397165

Questions

- What are the training and test computation times for kNN?
- What are the pros and cons of kNN classifiers?
- Can kNN be used for regression?
- How will the value of k affect the bias?

SVM

- Find a boundary that maximizes the distance to closest vectors
- If not possible, find one that minimizes the error
- Add the kernel trick

Perceptron

- Binary classification
- Linear boundary in feature space
- $\hat{y} = \text{sign}(wx + b)$

Algorithm:

```
$w_0 = \overrightarrow{0}$
For every data point $x_i$
    $\hat{y}_i = \text{sign}(w_k x_i + b)$
    if $\hat{y}_i \neq y_i$:
        $w_{k+1} = w_k - \hat{y}_i \cdot x$
    else:
        $w_{k+1} = w_k$
```

- TODO: illustration
- TODO: advantages/disadvantages

Useful Python Implementations

Resources

1. UdS SNLP Class, WSD: https://teaching.lsv.uni-saarland.de/snlp/
2. Decision Trees: https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html
3. kNN Example: https://iq.opengenus.org/text-classification-using-k-nearest-neighbors/