

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

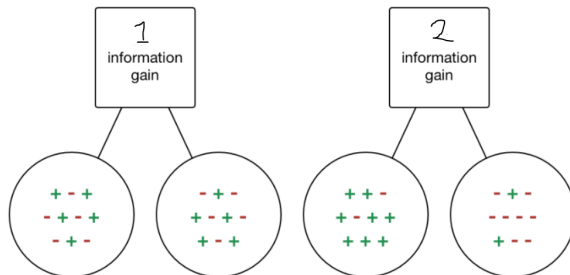
```
function DECISION-TREE-LEARNING(examples, attributes, parent_examples) returns a tree
  if examples is empty then return PLURALITY-VALUE(parent_examples)
  else if all examples have the same classification then return the classification
  else if attributes is empty then return PLURALITY-VALUE(examples)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

- Based on Bayes Theorem
- Algorithm

Given a set of features $\{x_1 \dots x_n\}$ and class label y

For each x_i in x and y_i in y :

Find $P(x_i | y_i)$

For all y_i in y :

Find $P(y_i)$

For a new sample $x' = \{x'_1 \dots x'_n\}$

For every y in y_i :

$P(y_i | x') = \prod P(y_i | x'_i)$ for x'_i in $\{x'_1 \dots x'_n\}$

Assign y_i with maximum $P(y_i | x')$ as the class label.

Questions

- Why is Naive Bayes naive?
- What are the pros and cons?

- 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  $\{\mathbf{Y}_i \text{ where } i \in I\}$ 
```

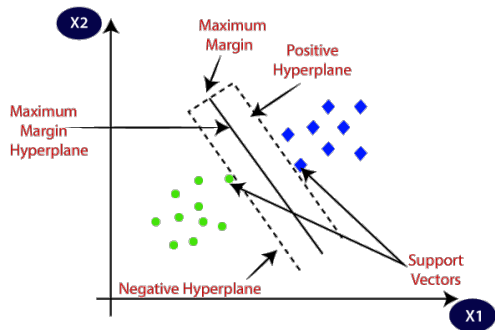
Source: https://www.researchgate.net/figure/Pseudocode-for-KNN-classification_fig7_260397165

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?

SVM

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



Questions

- What are the pros and cons of SVMs?
- Can SVMs be used for regression?

Perceptron

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

Algorithm:

- $w_0 = \vec{0}$
- 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:
- ▶ ★ $w_{k+1} = w_k$

Questions

- What are the pros and cons of simple perceptrons?
- Can we extend this to non-linear data?

Common Evaluation Measures

- Confusion matrix
- Precision = $\frac{TP}{TP+FP}$ (out of those marked as 1, how many are actually 1?)
- Recall = $\frac{TP}{TP+FN}$ (out of all 1s, how many are marked 1?)
- F-measure = $\frac{2 \cdot P \cdot R}{P+R}$ (weighted average of precision and recall)
- Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$

Useful Python Implementations

- https://scikit-learn.org/stable/supervised_learning.html
- Decision Trees: <https://scikit-learn.org/stable/modules/tree.html>
- Naive Bayes: https://scikit-learn.org/stable/modules/naive_bayes.html
- K Nearest Neighbour: <https://scikit-learn.org/stable/modules/neighbors.html>
- SVMs: <https://scikit-learn.org/stable/modules/svm.html>
- Perceptron:
https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Perceptron.html
- Evaluation metrics: https://scikit-learn.org/stable/modules/model_evaluation.html

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 Naive Bayes Example: <https://medium.com/analytics-vidhya/naive-bayes-classifier-for-text-classification-556fabaf252b>
- ④ kNN Example: <https://iq.opengenus.org/text-classification-using-k-nearest-neighbors/>
- ⑤ SVM: <https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/>
- ⑥ Perceptron
<https://machinelearningmastery.com/perceptron-algorithm-for-classification-in-python/>