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

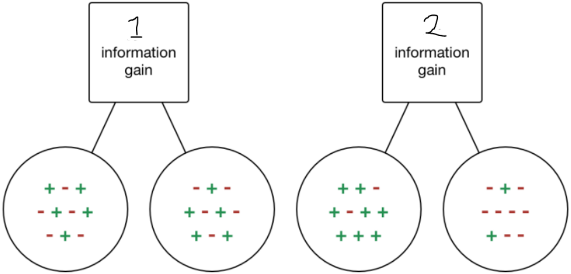
- What is a decision tree?

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?

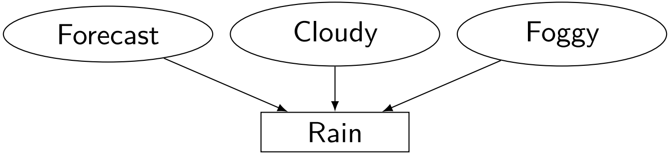
- How to avoid overfitting?
- How to use decision trees for regression?

- Algorithm

Given a set of features {x1..xn} and class label y

Naïve Bayes

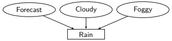
- Formula?
- $p(y = \text{Will rain}|x) = p(y_j|x) = \frac{p(x|y_j)p(y_j)}{p(x)} \propto p(x|y_j)p(y_j) \approx p(y_j)\prod_i p(x_i|y_j)$
- $\rightarrow \arg\max_{y_j} p(y_j)\prod_i p(x_i|y_j)$



- Why is Naive Bayes naive?
- How is the prior of e.g. 90% probability of not raining (overall) modelled?
- What are the pros and cons?

Naïve Bayes

- Formula?
- $p(y = \text{Will rain}|x) = p(y_j|x) = \frac{p(x|y_j)p(y_j)}{p(x)} \propto p(x|y_j)p(y_j) \approx p(y_j)\prod_i p(x_i|y_j)$
- $\rightarrow \arg\max_{y_j} p(y_j)\prod_i p(x_i|y_j)$



- Why is Naive Bayes naive?
- How is the prior of e.g. 90% probability of not raining (overall) modelled?
- What are the pros and cons?

- In Naïve Bayes we artificially flatten the network so that the observed variable is directly dependent to all causes and there are no other dependencies.
- The formula shows where the approximation is taking place.
- A practical example why this is naïve is that the variable *Rain* is heavily dependent on the *Cloudy* variable but as well on the *Foggy*, which in turn is almost the same thing as *Cloudy*. And if we put both all these in the formula, then we assign higher weight to the concept of *cloudyness* than to *forecast*.

kNN

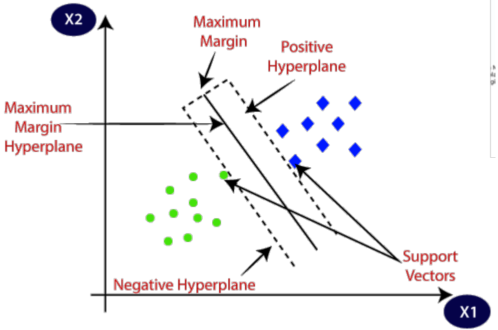
k-Nearest Neighbor
Classify $(\mathbf{X}, \mathbf{Y}, x)$ // $\mathbf{X}$: training data, $\mathbf{Y}$: class labels of $\mathbf{X}$, x: unknown sample
for $i = 1$ **to** m **do**
  Compute distance $d(\mathbf{X}_i, x)$
end for
Compute set I containing indices for the k smallest distances $d(\mathbf{X}_i,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

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

- 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



- 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:

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

- 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** $= \frac{TP}{TN+FN}$ (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** $= \frac{TP+TN}{TP+TN+FP+FN}$

Common Evaluation Measures

- **Confusion matrix**
- **Precision** $= \frac{TP}{TP+FP}$ (out of those marked as 1, how many are actually 1?)
- **Recall** $= \frac{TP}{TN+FN}$ (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** $= \frac{TP+TN}{TP+TN+FP+FN}$

Precision - TP/PREdicted true values, Recall - TP/REal values

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

Assignment 9

- Exercise 1: Text classification
- Bonus: Support Vector Machines

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/
7. Maximum Entropy Classifier: http://cseweb.ucsd.edu/~elkan/254/ari_talk.pdf