Report for Assignment 2 - ML

Longfei Zhao, u5976992, 22/8/2017

Q2. Clustering with Naive Bayes

Pseudocode

Pseudocode is mixtured with some python syntax. Note: For `D={(y, x⃗)}`, `label y` is a list of selected classes for each documents. `x⃗` contains all tokens and their count in those documents.

```
recompute(k, labels, docs):
    class <- [] * k
    for curClass in 1..k:
        class[curClass] <- TrainNB(curClass, doc[labels == curClass])
    return class

reassign(class, docs):
    k <- length(class)
    n <- length(docs)
    labels <- [] * n
    for idx in 1..n:
        p <- [0] * k
        for curClass in 1..k:
            p[curClass] += log(class[curClass]['docsCount'] / n)
            base <- class[curClass]['docsCount'] + class[curClass]['sum']
            for token, _ in doc['tokens']:
                p[curClass] += log((class[curClass]['tokens'][token] + 1) / base)
        labels[idx] = argmax(p)
    return labels

newCluster(k, docs):
    class <- [] * k
    seedDocs <- selectRandomSeed(k, docs)
    for doc in seedDocs, curClass in 1..k:
        class[curClass] <- TrainNB(curClass, doc)
    labels <- reassign(class, docs)
    while stopping criterion has not been met:
        class <- recompute(k, labels, docs)
        labels <- reassign(class, docs)
    return class
```

Convergence Analysis

The original K-means and this Naive Bayes variant won't always converge to the same answers, because those two algorithm's main idea are totally different. For K-means, document will be considered as a vector in a n-Dimension space. We cluster them by considering spatial distance. For Naive Bayes, all tokens will summed to compute the probability. For example, assume document `doc1` contains `token1` 1 time and `token2` 100 time, document `doc2` contains `token1` 100 time and `token2` 1 time and class `class1` contains `token1` as many as `token2`. For K-means, they are obviously different while they will be same in Naive Bayes.

Q3. Logistic Regression

1. the number of male who take the offer is in direct proportion to log(offered money)
2. 0.9
3. 9
4. e^2.21724, e^2.21724 / (1 + e^2.21724)
5. 3.47619
6. proportion of women taking the deal is less than man, for all given money. Generally, woman value the privacy more than man.
7. a little more than 4 thousands
8. about 8 thousands
9. In this case, money and people who take the deal are not based on linear regression. Logistic regression is more robust.

Q4. Hierarchical Classification

Algorithm Description

- Training Part

If a document is in a class `class1`, it is also in all Ancestors. We can simply add all those data in trainNB().

- Test Part

Pseudocode is shown in below. We classify the document from the root to the last level. In each loop, get all classes in this level and then treat it as a sample classification. Note that I think there may need a one more class standing that this document is not fit all children classes and belongs to the parent class or maybe we could set a minimum for this level, if all classes not exceed this bound then this document belongs to parent class.

```
def hierarchicallyClassify(root, doc):
    parent <- root
    while:
        classInThisLevel <- parent.childrenClass()
        res <- classify(classInThisLevel, doc)
        if res == -1:
            parent.containDocs.append(doc)
            break
        else:
            parent <- classInThisLevel[res]
            if parent.children.isEmpty():
                parent.containDocs(doc)
                break

def classify(classInThisLevel, doc):

    //return the class that doc fit most or if all class are not meet the require,
return -1 that means this doc belongs to the parent class and no more downwards
classify.
```

Complexity

- Training Part

We know that the Complexity of single class classification is `O(m * Lavg + b * v)` (in same branching, document belongs to just one class). For any document, it can belong to at most depth `d` classes. Therefore, the final complexity is `O(m * Lavg + d * b * v)`.

- Test Part

We could assume that same branching as a single classification. The complexity is `O(b * Ma)`. There are depth `d`, so the final complexity is `O(d * b * Ma)`.

- feasibility

Yes, because there are no more than linear growth.