Comp6490 Assignment 3

Guyue HU
u5608260

Q2

(1) **pseudo code**:

```
modified_K_means():
K = 2
initialize(D)
loop
    nb = trainNB(D)
    relabel(D)
until not changed
```

```
initialize(D):
random set label y ∈ {1, 2} for all documents in D
```

```
trainNB(D):
D1 = all documents labelled 1
D2 = all documents labelled 2
p(c = 1) = |D1| / |D|
p(c = 2) = |D2| / |D|
for all features $t_i$
    p($t_i$ | c = 1) = (number of $t_i$ from documents from D1) /
                       (number of all features in D1)
    p($t_i$ | c = 2) = (number of $t_i$ from documents from D2) /
                       (number of all features in D2)
```

```
relabel(D):
for all documents $d_i$ in D
    p1_t = 1
    p2_t = 1
    for all terms $t_j$ in $d_i$
        p1_t = p1_t * p($t_j$| c = 1)
        p2_t = p2_t * p($t_j$| c = 2)
    y = $argmax_k$($p_k$_t * p(c = k))
```

```
8       set label of $d_i$ as y
```

(2) No. We cannot expect the classifiers always converge to the same answer, because the classifiers converges to local (not global) optimals, and the answers are influenced by the initial settings.

Q3

$$P(c|d) = \frac{1/dist(c|d)}{\sum_{i=1}^{k} 1/dist(c_i|d)} \ (c_1, \ldots, c_k \in C)$$

Q4

(1) **Training**: For each leaf node, set the prototype of this node as the mediod of the feature vectors of all the documents assigned to this node. For each intermediate node, set the prototype of this node as the weighted average of the prototype of all its children nodes.

Test: For an unseen new test document d, starting from all the children of the root node $c_1, \ldots, c_k$ calculate $dist(c, d)$, which is the distance from the prototype of c to the feature vector of d. Assign d to c_i with the smallest distance. For all the children nodes of c_i, recursively loop the process, until d is assigned to a node with no child node.

(2) **Training time complexity**: $\Theta\ (mv+n)$
justification: assuming there are n_1 leaf nodes and n_2 intermediate nodes, where $n_1 + n_2 = n$. for all leaf nodes, the time complexity of summing up the feature vectors of all documents is $\Theta(mv)$ and the time complexity of calculate the mediod is $\Theta(n_1)$. For intermediate nodes, the optimal time complexity of calculating the weighted average of the children nodes is $\Theta(n_2)$ (if proper information is recorded when calculating lower level nodes). The total time complexity is
$\Theta(mv + n_1 + n_2) = \Theta\ (mv+n)$

Testing time complexity: $\Theta(bdv)$
justification: The test document is assigned from top down. At the beginning, the document is assigned to the root node. If the document is assigned to a node, the

time complexity of calculating the distance from the document to all the children node is $\Theta(bv)$. Because the depth of tree is at most d, this process would loop at most d times. The total time complexity is $\Theta(bdv)$

Yes the algorithm would be computationally feasible for large-scale training and testing. Because all the quantities m, n, b, d, v are uncorrelated, the time complexities of training and testing are linear in each of these quantities.

- end of assignment 3