

INF2B CW2 - Task 1

s1620208 - 13th April 2018

MySqDist

My implementation of the k-NN classification uses an efficient MySqDist implementation.

```
def MySqDist(X, Y):  
    XX = (X ** 2).sum(axis=1)[:, np.newaxis]  
    YY = (Y ** 2).sum(axis=1)[np.newaxis, :]  
  
    return XX - 2 * X.dot(Y.T) + YY
```

This makes use of numpy's automatic broadcasting as well as easy axis manipulation.

np.argpartition

Another way I speed up my implementation is by using *np.argpartition*, instead of *np.argsort*.

```
idx = DI.argpartition(maxK, axis=1)[:, :maxK]
```

This correctly *partitions* the array so that the provided index is in the correct position (leaving items larger to the right, and items larger to the left, unsorted). This is much faster because items on both sides *aren't sorted*, so it doesn't need to reshuffle all of the items

Subpartitioning

Because `np.argsort` provides numbers either side of a given number (at a specific index), and doesn't sort both sides, there is still a necessity to sort the resultant *idx*.

Instead of doing this I wanted to evaluate whether subpartitioning would be faster than sorting the *idx*. Unfortunately calling `argsort` on top of `argsort` did not work well (the resultant numbers were completely wrong) — perhaps it is necessary to first call *partition*, and then call *argsort* the second time round. In the end, I ran out of time to do this, and settled on just sorting *idx*.

Reducing for-loops

Currently, in the vital areas, my code only makes one for loop for knn classification and one for loop for calculating the confusion matrix.

The knn classification file uses a for loop to assign columns in the resultant *Cpreds* variable, and this involves converting indexes to labels, as well as calculating the modes.

The confusion calculation uses a for loop to sum up all the occurrences of pairs.

Results

Time taken: 10.180000 seconds

k	N	Nerrs	acc
1	7800	1965	0.748077
3	7800	1637	0.790128
5	7800	1505	0.806923
10	7800	1348	0.827179
20	7800	1301	0.833205