Web Search And Text Mining - A3
Anurag Kyal, 903040738

Solution 1a.
If the user wants to 'find pages like this one', the query optimization algorithm should have the maximum focus on the relevant document and minimum weight on the non-relevant document. It is quite unclear from the question how relevant is the current query as the document the user wants the other results to be like could be one of the bad results for the current query in which case we might want to weight the current query less. Assuming that the current query is not very relevant, the values would be like 0,1,0.

Solution 1b.
q_m would be the same as the original query when β and γ are 0 and α is weighed 1. In this case, the relevant and non relevant documents would have no effect on the new query.

In other cases, the q_m could be anywhere between the vector of original query and the centroid of the relevant document set in the Vector Space Model. If $\beta > \alpha$ and we consider that $\gamma = 0$ ie. the non relevant documents have no effect on the new query, then the new query will be closer to the the centroid of the relevant documents than q_0 and farther otherwise.

Solution 2a.
Reasons why relevance feedback is not very useful:

- It takes a long time and effort for the users to mark the results. It is generally undesirable to achieve search results in 2 steps.
- The search query formed after inclusion of terms from the feedback becomes computationally very expensive and time consuming.
- The method requires the judged documents to be a good number. If the number of judged documents is small, this method is bound to perform poorly.

Solution 2b.
For the relevance feedback algorithms to work well, it is required that the documents classified as non-relevant be alike so and form a single cluster. In practice, the set of negative documents is expected to be not related and thus not suited for the algortihm.

Choosing a single most irrelevant methods works well because by making the query distant from this document tends to eliminate the most documents that are similar to the most irrelevant document and thus increase performance. If instead a bunch of non relevant documents were selected, it would result in the problem described above and might perform poorly.

Solution 3.
Boosting and AdaBoost both work on the concept of ensembling many weak learners to form a learner which is expected to perform better. They work by initializing equal weights to all samples and train the classifier as we may assume. The classifiers are tested to check the errors and the

weights of the classifiers for the voting process is re-adjusted according to this result. The better classifier is given more power. Please note that the weight of the classifier is different from the weight of the examples used for training.

In the AdaBoost algorithm, in each iteration, the weights of the samples are adjusted for training the classifiers but the labels of the functions remain untouched. The Gradient Boosting operates in a slightly different way by combining boosting with Gradient Descent. In the gradient boosting algorithm, instead of adjusting the weights of the examples, we try to adjust the labels of the examples so that in the next iteration, the classifier tries to get closer to the desired label.

Solution 4.

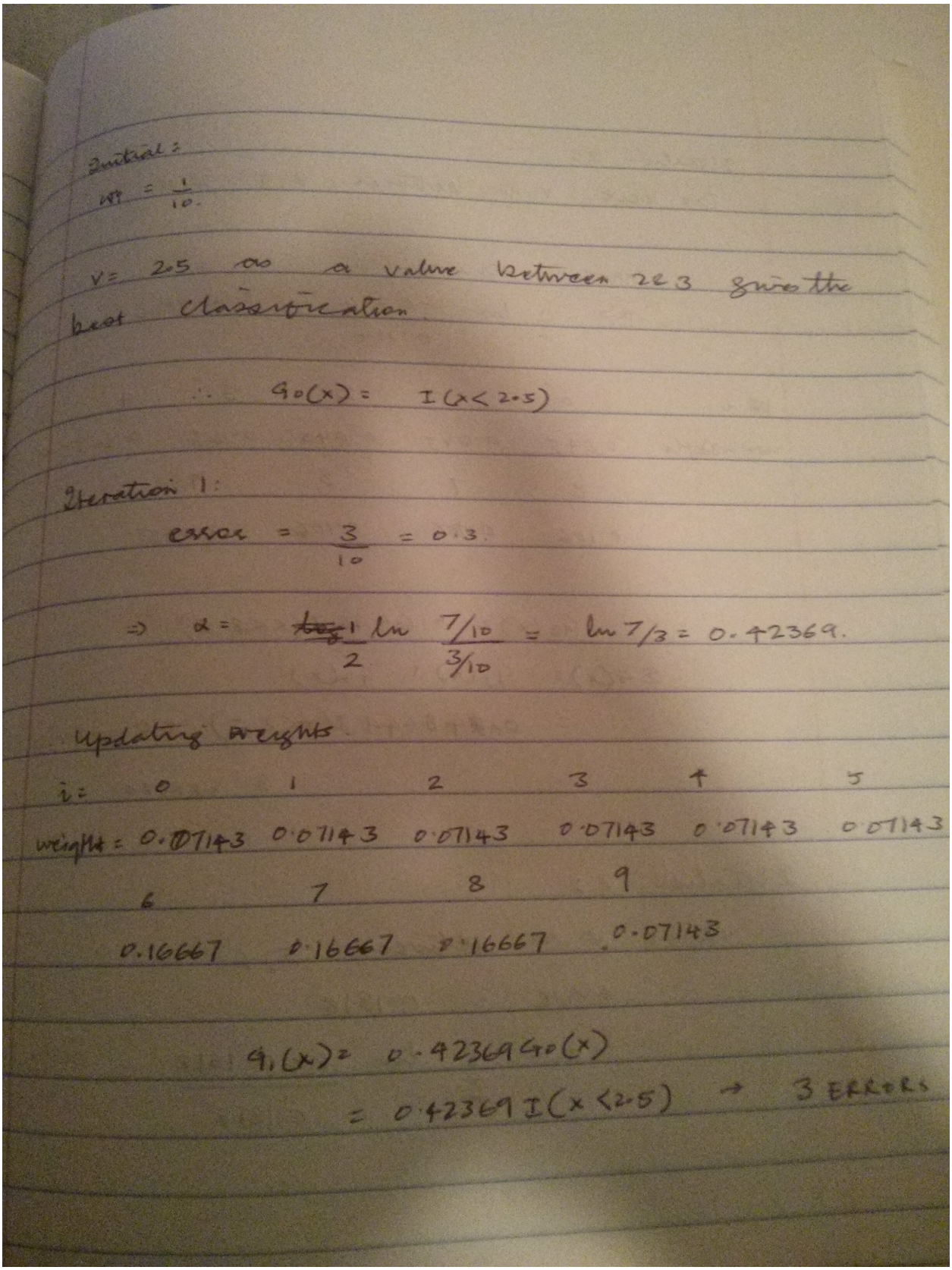

Figure 1: UDP Connection

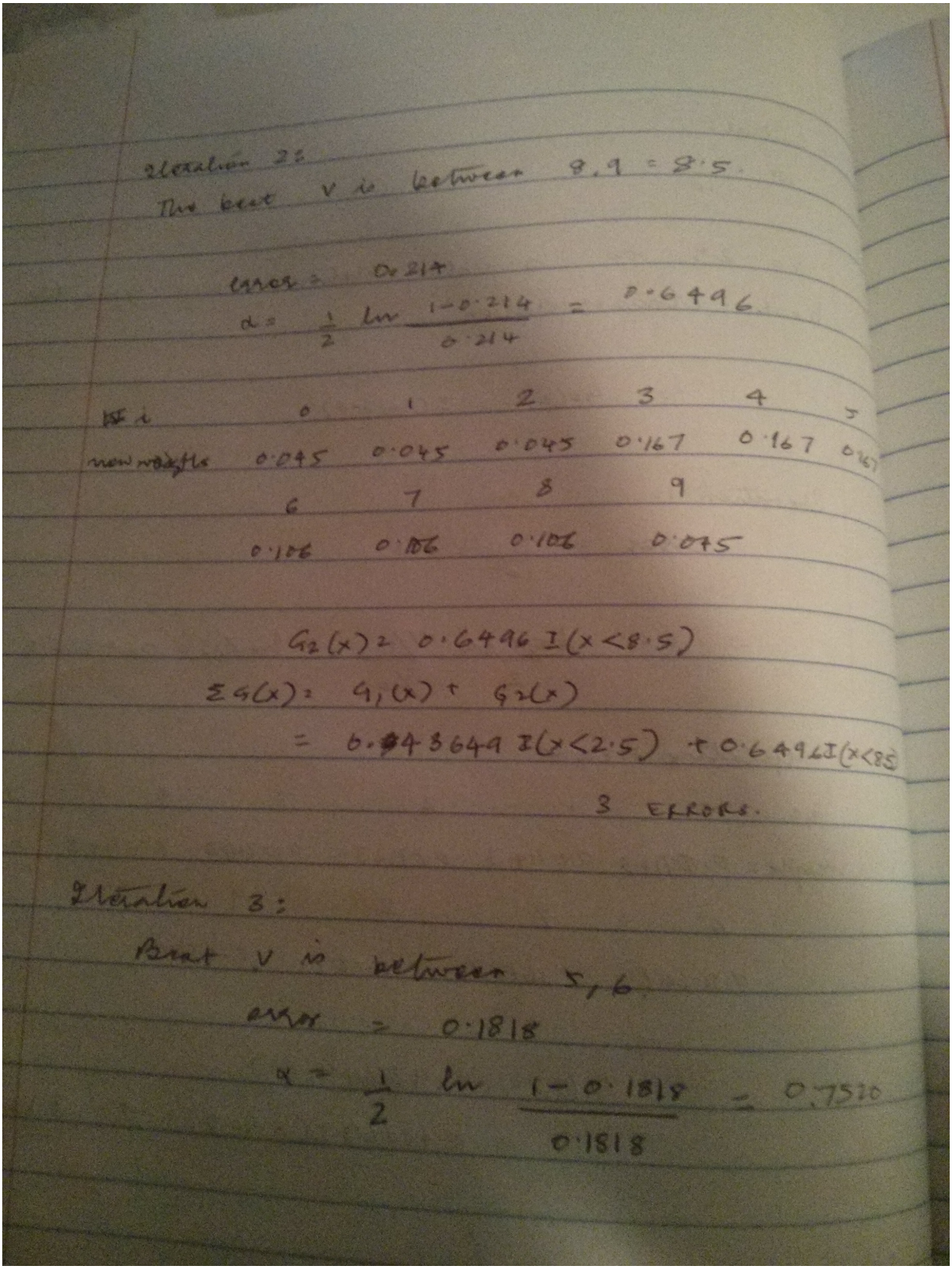

Figure 2: UDP Connection

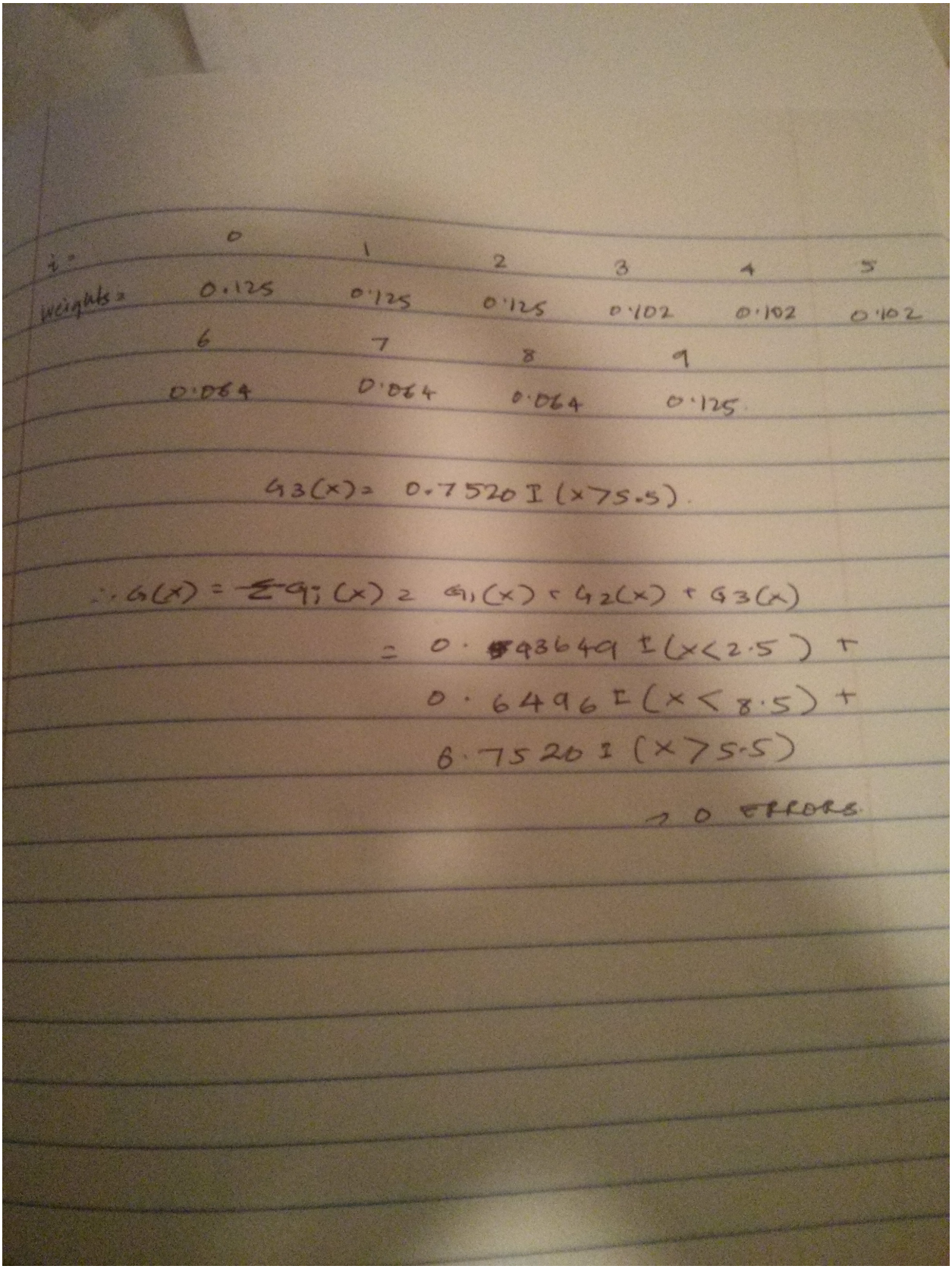

Figure 3: UDP Connection