Part III - Further Experiments

Dimension Reduction

kNN is known to suffer from the "curse of dimensionality", where the density of the training vectors is too low to accurately measure similarity. In the models used so far, with each unique term representing one dimension of the vector space, we may have been using a sub-optimal classifier.
To remedy this, we can reduce the dimensionality of the term-vector space by removing terms from the model altogether, since many terms may not particularly be useful for classifying a movie. To consider only the x most common words in a model, select the `Top Words` option when building a model, and adjust the number of terms to include.

Exercise 6

In this exercise, use the list `actrom100` with all preprocessing options, and set the training split to 60% and let $k = 11$.

a) Build models using the top 25, 100, 200, 500 and 1000 words only.

   i) Test these models with a kNN classifier, making a note of the results from each.
   ii) Between which pairs of values (e.g. 100 to 200) does the accuracy remain the same, despite the extra information?
   iii) Out of these values, which provides the best balance between dimension reduction and information loss?
   iv) Does the Rocchio algorithm give similar results? At what number of terms does it begin to decrease in accuracy, if at all? Check this against the performance on a model with all terms included.

b) Now build a model with all preprocessing options, except stopword removal, and include only the top 25 terms. This is effectively a "stopword only" classifier.

   i) Will this perform better than random chance (50% accuracy)?
   ii) Run both classifiers on this model. If it performs better than or worse than (or equal to) random chance, give a possible explanation for this behaviour.
   iii) Does the accuracy converge towards 50% when k is increased? Why or why not?

Data Distribution

We now explore the effect of different distributions of data on the performance of the classifier.

Exercise 7

a) Build two new models, with all preprocessing options, from the lists `actthril` and `actcom`. The first contains Action and Thriller movies, and the second contains Action and Comedy movies. Use a 60% training split in all experiments.

   i) Run a kNN classifier on each model, plotting a range of k-values in each case.

   ii) On average, which model gives better performance? Why do you think this is?

   iii) Now try with a Rocchio classifier - are the results similar to kNN?

   iv) Suppose you were to run a classifier on a model with Crime and Mystery movies - would you expect it to perform better or worse than a Crime / Family model, and why?

b) Next, build a model in the same way, using the `actall` list, of 100 Action movies and 100 non-Action movies, taken from all of the remaining genres.

   i) Run kNN with $k = 11$ and a training split of 60%, and compare it to the Action / Thriller and Action / Comedy models from part a). Does it outperform either of them?

   ii) Run both algorithms, with $k = 11$ in kNN, and plot the effect of the training set size in each case.

   iii) Which algorithm is more stable relative to the distribution of the data? Why do you think this is?

c) In addition to a description, each movie also comes with a "tagline", which is designed to sell the movie in a few words. We can append these words to the description of a movie by using the `Taglines` preprocessing option when building the model. To also add the movie's title to the description text, use the `Titles` preprocessing option.

   i) Build three models with `actrom100`, with all linguistic preprocessing options turned on. In the first, include the title in the description text, in the second, include the tagline, and in the third use both title and tagline.

   ii) Using a kNN classifier, compare the performance of these augmentations against the base model, plotting your results relative to the value of k.

   iii) Overall, which model performs the best, and which is the most stable?

   iv) With 1NN, does the `Titles` option improve the accuracy over the base model? Why or why not?

Exercise 8

a) Suppose you have implemented your own version of the kNN algorithm, and you want to make sure everything is functioning as expected. One way to do this is to create a custom dataset, and run experiments which you know in advance will or will not work. If the implemented algorithm matches the theoretical result, the classifier is likely to be robust and bug-free.

   Can you think of some custom datasets, based around movies, that you could build for this purpose?

b) One example would be to use sequels. In the movie list `sequels.lst`, there are 40 Action/Comedy movies, each accompanied by one of their sequels (for a total of 80 movies). If the training split is set to 50%, the original movies will make up the training set, with each sequel in the testing set.

   For example, Taken will be in the training set, and Taken 2 will be in the testing set. Use this split in all of the following experiments.

   i) How would you expect a 1NN classifier to perform on this test?

   ii) Build a model with the `sequels` list and all preprocessing options, then run a 1NN classifier to verify this result.

   iii) Plot the effect of k on this model. What trend would you expect to see?

   iv) Plot the effect of k on an older model, such as `actrom100`. If the pattern is different to `sequels`, why?

   v) Next, add the `Titles` and `Taglines` options to the `sequels` model. Would you expect the performance to increase or decrease? Why? Do the results agree with your expectations?

   vi) Next, run the Rocchio classifier on the same dataset. Does it score higher or lower than 1NN?

   vii) If you were to increase the amount of movie-sequel pairs in the model, which algorithm would perform better as the size gets larger?

c) In this question, we consider an alternate formulation of the `sequels` dataset, `sequelsmerged`. In this version, the sequels are paired, rather than split. So when the training size is set to 50%, there will be 40 movie-sequel pairs in both the training set and testing set.

   For example, Taken and Taken 2 will both be in the training set, while both X-Men and X-Men: Apocalypse will be in the testing set. Again, build a model with a 50% split and all preprocessing options for these experiments.

   i) Run this model with a 1NN classifier. Does this exhibit the same behaviour as with `sequels`? Why or why not?

   ii) Next, run a Rocchio classifier in the same configuration. Does the result agree with what you would expect?

   The performance of each algorithm will likely change with this new model. The model change could:

   i) Affect both algorithms for the same reason.

   ii) Affect each algorithm for different reasons.

   In both cases, find a possible explanation for the change in performance.

Customising Movies

With the `--custom` argument, you can provide your own description for a movie, in place of its official one. To use this in a model, modify the file `customisation.txt` in the `customisation` folder, and enter the descriptions in the following format:

```
[name];[description]
[name];[description]
...
```

Where `name` is the movie name without any spaces or punctuation, and in lowercase letters, e.g. `toystory3`. Next, when building a model, check the `Customisation` box to apply your custom descriptions.

Exercise 9

In this exercise, use models with all preprocessing options, and set $k = 11$.

a) The movie list `actrommatrix` contains 100 Action movies (including The Matrix) and 100 Romance movies, and the customisation file contains the official description of The Matrix (with a backup provided in `matrix_original.txt`, so it can be edited and restored).

   In this file, modify the description of The Matrix, such that it would be classified as a Romance movie in a kNN classifier (to check this, build a custom model as above, and run the `Find Neighbours` command).

   Do this in the following ways:

   i) By exchanging individual words with one that expresses a similar meaning.

      For example, changing "group of insurgents" to "group of people" may decrease the association with Action movies.

   ii) By modifying only a single word in the description.

   iii) By completely rewriting the description in your own words.

   iv) By replacing the description with a small number of keywords which best describe the movie.

   Use the commands from previous exercises to analyse the dataset and select appropriate terms.

b) Next, use the list `actcommatrix`, with Comedy movies instead of Romance, and rewrite the description so it is placed in the Comedy class.

c) What will happen if you leave the description empty, so that in the model, the movie has no terms? What will it be classified as, and why? Test this yourself.

If you haven't seen The Matrix:

d) Watch The Matrix.