Tutorial 11: Connectionist and Statistical Language Processing

Clustering

Date: 30.01.2002
Student's name:

1 The Task: Discovering Ambiguous Verbs

We will again use the Weka machine learning package in this tutorial. The files for the tutorial reside in:

```
/courses/connectionism.winter.01/tutorial11/
```

The task is to use co-occurrence vectors (as discussed in the last lecture) to discover verbs that are ambiguous as to their part of speech.

The co-occurrence vectors are contained in the file `vectors.arff`, and were generated as follows (Lowe and McDonald 2000). First a set of context words was created consisting of the 500 most frequent words in the British National Corpus (BNC; 100 million words of British English), excluding function words (determiners, prepositions, conjunctions, etc.). Then 222 verbs were selected at random from the BNC and its co-occurrence vectors with the context words were computed, based on a window of 10 words. The resulting vectors were normalized to make them comparable.

The file `vectors.arff` contains the output of this procedure encoded using the following attributes:

(1) `WORD`: the verb that is represented by the instance.
(2) `a_few ... young`: 500 attributes representing the co-occurrence of the instance with the context words.

To have a gold standard to evaluate the clustering output against, there is also an attribute `CAT` that specifies how ambiguous a given verb is in a WordNet, a large lexical database. There are four values that `CAT` can take:

(3) a. `verb`: only occurs as verb.
    b. `verbnoun`: occurs as verb or noun.
    c. `verbadj`: occurs as a verb or adjective.
    d. `verbnounadj`: occurs verb, noun, or adjective.

The task in this tutorial is to use clustering on the context vectors of the 222 verbs, and evaluate the resulting clusters against the attribute `CAT`. The hope is that verbs that exhibit the same ambiguity will cluster together.

2 Visualizing the Clusters

First we will attempt to visualize the clusters generated by the *k*-means clustering algorithm to get a feeling for what the algorithms does.

Start the Weka Explorer and open the file `vectors.arff`. Click on the button `None` under `Preprocess` to deselect all attributes (there are 502 of them). Then select the attributes `WORD`, `a_few`, and `able`. Now click on the button `Apply filter` to generate a working relation with these three attributes.

Click on `Cluster` to select the clustering mode of Weka. Under `Clusterer` select `SimpleKMeans` to choose the *k*-means clustering algorithm. Leave `seed` at 10, and set `numClusters` to 4. In the `Cluster mode` field make sure that `Use training set` is ticked, as well as `Store clusters for visualization`. Then press `Start` to compute the clusters.

Inspect the output of *k*-means and report the centroids (means) for the four clusters. The algorithm also outputs a word for each centroid. Do you have an idea how this word might be computed?

Now click with the right mouse button on the result labeled `SimpleKMeans` in the `Result list`. Chose `Visualize cluster assignments`. This brings up the visualization window. Now select `a_few` to be plotted on the x-axis, `able` to be plotted on the y-axis, and `Cluster` to be plotted as color.

This visualizes the clusters in a two-dimensional space defined by the values of the two attributes. Describe the pattern you see.

To check the behavior of an individual attribute, select `Instance_number` on the x-axis, the attribute on the y-axis, and `Cluster` as color. Compare the the two attributes `a_few` and `able` this way. Which one separates the clusters more accurately?

This way of visualizing clusters works well for two attributes. Is it possible to visualize the whole data set with 500 attributes?

3 Evaluating the Clusters

Now we will work on the whole data set with 500 attributes. Go back to `Preprocess` and click on the `All` button to select all attributes. Then press `Apply filter` and click on `Cluster`.

Under `Clusterer` again select `SimpleKMeans`. Set `seed` to 10 and `numClusters` to 4 (as we have four target classes). In the `Cluster mode` field tick `Classes to cluster evaluation` and select the attribute `WORD`. Now click on `Start`.

First perform an *intuitive* evaluation of the clusters, as discussed in the lecture. Inspect the resulting clusters and see which verb is assigned which cluster. Is there an intuitive resemblance to our target classification (unambiguous verbs, verbs that can be also nouns, verbs that can be also adjectives, and verbs that can be also both nouns and adjectives)?

Now change the k, the number of clusters that k-means comes up with. This can be done by changing `numClusters` in the `Clusterer` window. Start with $k = 2$ and work your way up to $k = 5$. Evaluate each results intuitively. Do you get better clusters with certain values of k? Do you think this is a sensible way of evaluating the clusters?

Now select `CAT` in `Classes to cluster evaluation` to use the category information in WordNet to evaluate the clusters. Again try values from $k = 2$ to $k = 5$ and report the respective accuracies. Which k gives you the best accuracy? Why is this a more sensible way of evaluating the performance of k-means on this data set?

4 Comparing Against a Baseline and Against a Classifier

In the last question, we determined the performance of *k*-means on the task of discovering clusters of ambiguous nouns. We will now compare the results to the performance using other machine learning methods.

Assume that this is a classification task with `CAT` as the target class. Compute two baselines: chance and frequency. How does the accuracy of *k*-means compare to these baselines? Is this a fair comparison?

Now train a decision tree on the same data set, with `CAT` as the target attribute. Chose the `Classify` panel and select in the `Classifier` box `j48.J48` (i.e., use C4.5 for decision tree induction). Train the classifier and report its precision on the training set. How does it compare to the precision of *k*-means? Again, is this a fair comparison?

References

Lowe, Will, and Scott McDonald. 2000. The direct route: Mediated priming in semantic space. In Lila R. Gleitman and Aravid K. Joshi, eds., *Proceedings of the 22nd Annual Conference of the Cognitive Science Society.* Mahwah, NJ: Lawrence Erlbaum Associates.