COMP30027: Project 1

Q1, Q2, Q3, Q6, Q7

What is labelled data worth to Naive Bayes?

Authors:
Chirag Rao Sahib 836011
Maleakhi Agung Wijaya 784091

March 28, 2018

Question 1

Since we're starting off with random guesses, it might be surprising that the unsupervised NB works at all. Explain what characteristics of the data cause it to work pretty well (say, within 10% Accuracy of the supervised NB) most of the time; also, explain why it utterly fails sometimes

Although the unsupervised classifier is not provided with labelled training data the classifier is still able to discover basic patterns since the inherent structure, and class-attribute relationship is maintained. It is important to note that the datasets provided were not randomly generated, and as with any real-life observation are likely to contain clusters; described by a particular probability distribution. However a key difference is that the unsupervised version is likely to invert the detected pattern due to the lack of class labels. The unsupervised NB is detecting a pattern but unsure as to which class it relates to; resulting in the need for class 'swapping'. The crux of why the unsupervised NB works at all, is that the approximate distribution of the unlabelled training data can still be modelled without the class/cluster labels.

The unsupervised NB utterly fails when the initial class probabilities are uniformly distributed. When the initial distribution is uniform (or close to it) the probability of each class (for an instance) is set to 1 / c, where c is the number of classes. After iterating, the new class distributions will be exactly the same as the previous iteration because the classifier randomly selects either class as both are equally likely. Essentially, with a uniform initialisation of class distributions, unsupervised NB will not diverge across the iterations. To a lesser extent, a recurring point of failure for all NB classifiers is also applicable to the unsupervised version - when there is insufficient data to accurately describe the relationship between attributes.

```
Accuracy (supervised)
2018S1-proj1_data/breast-cancer-dos.csv | Avg. Measure: 0.7552447552447551
2018S1-proj1_data/car-dos.csv | Avg. Measure: 0.8738425925925926
2018S1-proj1_data/hypothyroid-dos.csv | Avg. Measure: 0.9522605121719886
2018S1-proj1_data/mushroom-dos.csv | Avg. Measure: 0.991999015263417

Accuracy (unsupervised)
2018S1-proj1_data/breast-cancer-dos.csv | Avg. Measure 0.7027972027972028
2018S1-proj1_data/car-dos.csv | Avg. Measure 0.7002314814814815
2018S1-proj1_data/hypothyroid-dos.csv | Avg. Measure 0.9522605121719886
2018S1-proj1_data/mushroom-dos.csv | Avg. Measure 0.8548744460856721
```

The screenshot below depicts the state in which unsupervised NB utterly fails with a uniform initialisation. (and the resulting confusion matrix)

	age	menopause	tumor-size	inv-nodes	node-caps	deg-malig	breast	breast-quad	irradiat	recurrence-events	no-recurrence-events
0	40-49	premeno	15-19	0-2	yes	3	right	left_up	no	0.5	0.5
1	50-59	ge40	15-19	0-2	no	1	right	central	no	0.5	0.5
2	50-59	ge40	35-39	0-2	no	2	left	left_low	no	0.5	0.5
3	40-49	premeno	35-39	0-2	yes	3	right	left_low	yes	0.5	0.5
4	40-49	premeno	30-34	3-5	yes	2	left	right_up	no	0.5	0.5

	Recurrence-Events Predicted	No-Recurrence-Events Predicted
Recurrence-Events Actual	81	0
No-Recurrence-Events Actual	196	0

Question 2

When evaluating supervised NB across the four different datasets, you will observe some variation in effectiveness (e.g. Accuracy). Explain what causes this variation. Describe and explain any particularly suprising results.

It is fairly clear that datasets with fewer instances, namely breast-cancer and car, consistently have lower accuracies. We hypothesise that the difference in accuracies between the first and last two datasets is due to both insufficient variability in the available data, and the assumption of conditional independence. Similar results were observed when the classifier was trained on the test data and when using holdout, indicating the observed variation is independent of these. Note for the holdout method the classifier was run 10 times and the average accuracy was calculated.

The accuracy for the 'breast cancer' and 'car' datasets was only around 75% and 85% respectively. 'breast cancer' and 'car' likely do not contain enough instances (286 and 1728, respectively) to capture the inherent variability of the data. Although Naive Bayes functions correctly for datasets with fewer instances, the accuracy suffers as the probabilistic relationship cannot be accurately described. On the other hand, a particularly surprising result is the accuracy achieved with the 'hypothyroid' and 'mushroom' datasets, which contained around 3000 and 8000 instances respectively. Our argument is accentuated by the proportional increase in accuracy of these datasets; between 95-99%.

Another source of error may arise from the NB classifiers assumption of conditional independence between attributes; which is generally untrue. We hypothesize that datasets with lower accuracy ('breast cancer' and 'cars') contain numerous dependent attributes, thus violating a key assumption of the Naive Bayes classifier. This assumption does not usually have a large effect since the posterior probabilities are usually over-or-underestimated resulting in the overall probability being underestimated. Considering this it is likely that the 'mushroom' and 'hypothyroid' datasets have attributes that are less strongly correlated than those of the other datasets.

```
Using holdout, averaged over 10 runs
2018S1-proj1_data/breast-cancer-dos.csv | Avg. Measure: 0.7441493476497205
2018S1-proj1_data/car-dos.csv | Avg. Measure: 0.8604608212498983
2018S1-proj1_data/hypothyroid-dos.csv | Avg. Measure: 0.9521428931492114
2018S1-proj1_data/mushroom-dos.csv | Avg. Measure: 0.9922275854371312
```

Question 3

Evaluating the model on the same data that we use to train the model is considered to be a major mistake in Machine Learning. Implement a hold–out (hint: check out numpy.shuffle()) or cross–validation evaluation strategy. How does your estimate of Accuracy change, compared to testing on the training data? Explain why. (The result might surprise you!)

It should first be noted that the inherent randomness of holdout (selecting training and test instances randomly) will consequently produce varying accuracy on each run. The holdout method was run 10 times and the average accuracy was calculated. Implementing holdout generally results in a slight reduction in accuracy when compared to testing on the training data. The random partitioning method is likely to produce test data that is not representative of the training data or vice versa.

As discussed in Question 2, datasets with a smaller number of instances tend to experience a reduction in accuracy due to insufficient variability. The amount of data available for training is further reduced with the holdout method. Depending on the chosen ratio this may result in an accurate model, but the test data is unlikely to be representative of all data; as mentioned in week03a. For example the 'breast cancer' dataset contains only 286 instances, randomly selecting 57 instances as test data (20%) provides no guarantee that the distribution of the test data is similar to that of the training data and hence definitely not an accurate representation of the unseen data.

Furthermore the reduction in accuracy is expected if one considers that testing on the training data can be likened to "telling the classifier what the correct answers are, and then asking whether it can come up with the correct answers". Once the correct answers are taken away we can expect the accuracy of the classifier to decrease.

```
Using holdout, averaged over 10 runs
2018S1-proj1_data/breast-cancer-dos.csv | Avg. Measure: 0.7414519588356774
2018S1-proj1_data/car-dos.csv | Avg. Measure: 0.8561387053264783
2018S1-proj1_data/hypothyroid-dos.csv | Avg. Measure: 0.9489565375703217
2018S1-proj1_data/mushroom-dos.csv | Avg. Measure: 0.9928177353617211

Training on test data
2018S1-proj1_data/breast-cancer-dos.csv | Avg. Measure: 0.7552447552447551
2018S1-proj1_data/car-dos.csv | Avg. Measure: 0.8738425925925926
2018S1-proj1_data/hypothyroid-dos.csv | Avg. Measure: 0.9522605121719886
2018S1-proj1_data/mushroom-dos.csv | Avg. Measure: 0.991999015263417
```

Question 6

Rather than evaluating the unsupervised NB classifier by assigning a class deterministically, instead calculate how far away the probabilistic estimate of the true class is from 1 (where we would be certain of the correct class), and take the average over the instances. Does this performance estimate change, as we alter the number of iterations in the method? Explain why.

Due to the nature of an unsupervised NB classifier in which classes may be 'swapped', an assumption must first be made of how to define the 'probabilistic estimate of the true class'. It would be inaccurate to simply define this as the probability of the true class after the iterations, as the probabilities may in fact be 'swapped'. Hence we assume that the class with highest probability is representative of the true class.

Under this assumption, in any dataset with no more than 2 classes we can observe a reduction in delta (how far away the probabilistic estimate of the true class is from 1) as the number of iterations increases; i.e. an inverse relationship exists between delta and the number of iterations. The observed increase in performance (lower delta) can be attributed to the convergence of the probabilistic class estimates as the number of iterations increase. This process maximises the likelihood through repeated estimations of the posterior probabilities, until convergence. (Further discussed in Question 7)

On the other hand the delta value for the dataset 'car' which contains 4 classes is consistently observed around 0.73. This can most likely be explained by the additional complexity in regards to class 'swapping' for a dataset with 4 classes; perhaps the assumption above does not hold.

```
unsupervised delta testing 5 iterations
2018S1-proj1_data/breast-cancer-dos.csv | Delta Avg.: 0.14704164732757008
2018S1-proj1_data/car-dos.csv | Delta Avg.: 0.7350380661994881
2018S1-proj1_data/hypothyroid-dos.csv | Delta Avg.: 0.05463283416580999
2018S1-proj1_data/mushroom-dos.csv | Delta Avg.: 0.0012226083886554336

unsupervised delta testing 10 iterations
2018S1-proj1_data/breast-cancer-dos.csv | Delta Avg.: 0.04936186522295407
2018S1-proj1_data/car-dos.csv | Delta Avg.: 0.7334206524625354
2018S1-proj1_data/hypothyroid-dos.csv | Delta Avg.: 0.0012487003184797482
2018S1-proj1_data/mushroom-dos.csv | Delta Avg.: 0.00036122349489423777

unsupervised delta testing 15 iterations
2018S1-proj1_data/breast-cancer-dos.csv | Delta Avg.: 0.034543243039314116
2018S1-proj1_data/car-dos.csv | Delta Avg.: 0.7339012545186191
2018S1-proj1_data/hypothyroid-dos.csv | Delta Avg.: 4.6463412735792215e-14
2018S1-proj1_data/mushroom-dos.csv | Delta Avg.: 0.0
```

Question 7

Explore what causes the unsupervised NB classifier to converge: what proportion of instances change their prediction from the random assignment, to the first iteration? From the first to the second? What is the latest iteration where you observe a prediction change? Make some conjecture(s) as to what is occurring here.

By printing out the confusion matrix for every iteration we observe that the proportion of instances changing prediction decreases as the number of iterations increase. This follows from Question 6 as the delta performance measure also decreases as the number of iterations increase. Both observations assist in illustrating the convergence of the unsupervised NB classifier. Below is a summary of our results for each dataset:

Dataset	Proportion change (initial to first)	Proportion change (first to second)	Latest iteration of prediction change
breast-cancer	$34/286 = 0.119$	$24/286 = 0.084$	9
car	$520/1728 = 0.301$	$2/1728 = 0.001$	2
hypothyroid	$302/3163 = 0.095$	$150/3163 = 0.047$	9
mushroom	$4192/8124 = 0.516$	$784/8124 = 0.097$	9

For most of the datasets the latest iteration in which we observe a prediction change is the 9th iteration. As with every other question 'car' is once again an outlier which is mostly explained by it having 4 classes versus 2. It is worth noting that with 'cars' the predictions actually continue changing slightly, after around 6 iterations.

Through each iteration the classifier refines the prediction as it leans more and more towards a particular class; in each subsequent iteration this 'leaning' is represented by an increase in the corresponding class probability. Towards the end of the convergence process, class probabilities that are comparatively large will continue increasing while the smaller probabilities decrease proportionately. For example the gap between $P(instance|class1) = 0.99$ and $P(instance|class2) = 0.01$ will keep increasing. However the confusion matrix is unlikely to change from the 'latest iteration' where a prediction change is observed, since the difference between the two values is sufficiently large. This behaviour is similar to that of the Expectation-maximization algorithm.[1]

[1] http://cs229.stanford.edu/notes/cs229-notes8.pdf

Example depicting the change in class distributions between the initial and final iteration.

```
initial class distributions (random)
   0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17  hypothyroid  negative
0  M  f  f  f  f  f  f  f  f  f  f  f  y  y  y  y  y  n     0.123867  0.876133
1  F  t  f  f  f  f  f  f  f  f  f  f  y  y  y  y  y  n     0.533972  0.466028
2  M  f  f  f  f  f  f  f  f  f  f  f  y  y  y  y  y  n     0.435870  0.564130
3  F  f  f  f  f  f  f  f  f  f  f  f  y  y  y  y  y  n     0.395154  0.604846
4  M  f  f  f  f  f  f  f  f  f  f  f  y  y  y  y  y  n     0.785055  0.214945

final iteration (nine)
   0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17   hypothyroid  negative
0  M  f  f  f  f  f  f  f  f  f  f  f  y  y  y  y  y  n  3.338978e-13  1.0
1  F  t  f  f  f  f  f  f  f  f  f  f  y  y  y  y  y  n  3.683326e-13  1.0
2  M  f  f  f  f  f  f  f  f  f  f  f  y  y  y  y  y  n  3.338978e-13  1.0
3  F  f  f  f  f  f  f  f  f  f  f  f  y  y  y  y  y  n  8.439631e-13  1.0
4  M  f  f  f  f  f  f  f  f  f  f  f  y  y  y  y  y  n  3.338978e-13  1.0
```