F20DL - Data Mining CW2 Report

F20DL_2018-2019
Data Mining and Machine Learning
Taught by Diana Bental and Ekaterina Komendantskaya

Mobeen Aftab - H00220767
Jonathan Mendoza - H00251229
Owen Welch - H00235788

The task in this coursework is to work with a large ‘emotion recognition’ dataset, created by the research group of Pierre-Luc Carrier and Aaron Courville. The data set (of 35887 examples) consists of 48x48 pixel grayscale images of faces and the overarching goal is to categorize each face based on the emotion shown in the facial expression in one of seven categories and then perform various data mining activities on that data such as attempting to recognise new data once trained off this training set.

Source Code and Other Documentation

Source code used for this course work can be found on our private Github repository. Other supplementary documentation is in our Google Drive.

Task 1: Naïve Bayes Networks

We ran the BayesNet algorithm on the main fer2018 dataset as well as the fer2018-emotion datasets using K2 algorithm to learn the architecture with max number of parents = 1. We wrote a script to help us do this: bayes1.sh.

From CW1, we had datasets which consisted only of top attributes gathered from attribute selection from each fer2018-emotion dataset. We ran the BayesNet algorithm with the same settings as above. Bayes1.sh was also used. We used 3 fold cross validation to evaluate.

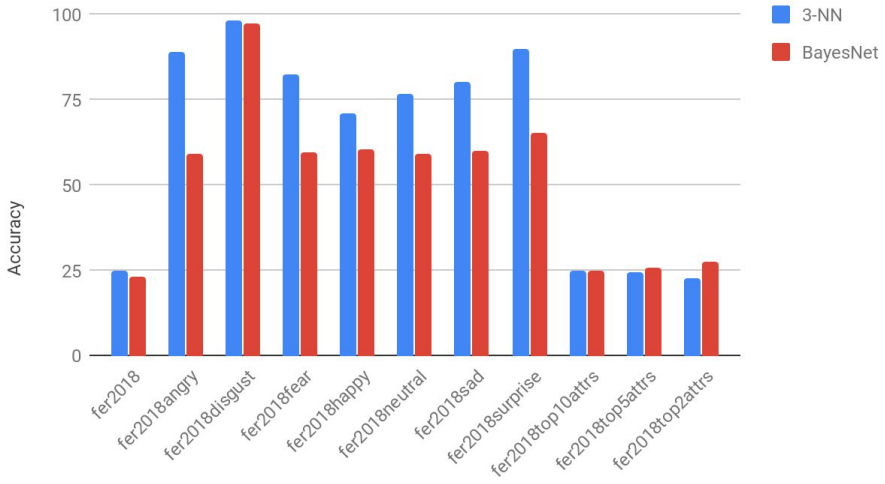

Figure 1 - Shows accuracy of both BayesNet and 3-NN on all datasets

The folder with the script, processed data, and raw data can be found here

Task 2: Naïve Bayes Networks - Conclusions

Using K2 with number of parents=1, the learned network structures are Naive Bayes networks. All non-class attributes have the class attribute as the parent.

As shown in Figure 1, There were no significant differences in accuracy on the main datasets. In the individual emotion datasets, we observed that the accuracies have decreased. However, as illustrated using a subset of our results, the time it takes to classify new instances of data is significantly quicker using the BayesNet model as opposed 3-NN. This supports our findings in CW1 that K-NN, in terms of classification speed, is not efficient on larger datasets. So Bayes Nets are useful for performance tuning.

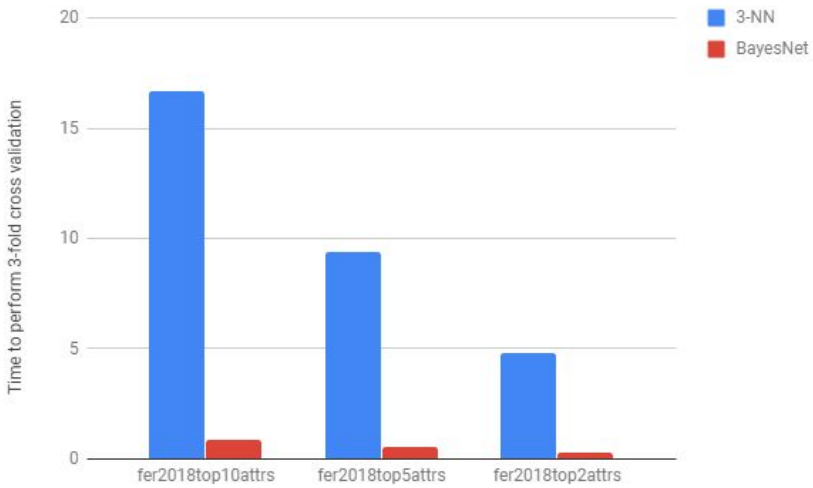

Figure 2 - Shows that classifying new instances using BayesNet is significantly faster than 3-NN

Task 3: Complex Bayesian Network Architectures

Like in Task 1, we wrote a script called bayes2.sh to automate the running of BayesNet using different search algorithms. Figure 3 shows our results. K2(1) refers to K2 with max number of parents as 1 (the results from task 1). K2(2) and HillClimber(2) refers to K2 and HillClimber algorithms with max num. of parents = 2.

We found out that initial ordering of the attributes affect the results of using K2. So we ran K2 several times with random ordering. The results were indeed different, but the differences were less than 1%.

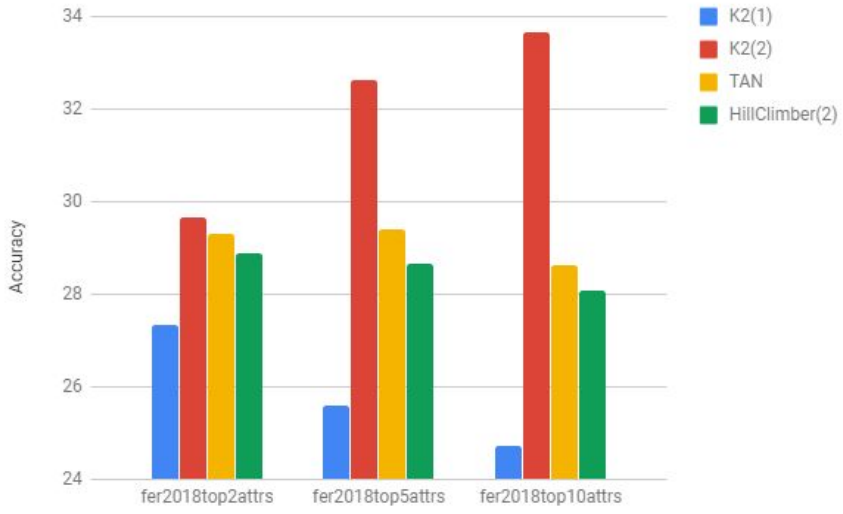

Figure 3 - Using more complex search algorithms to learn the network structure results in higher accuracies

Task 4: Complex Bayesian Network Architectures - Conclusions

Using more complex Bayesian networks, you could discover which attributes could be conditionally dependent of each other. In our case, it seems that a more complex architecture have lead to increased accuracies in the smaller datasets. However when attempting to use more complex search algorithms on the full dataset (fer2018 with all attributes), the model takes too long to be made; thus, attempting to improve our model through the use of more complex search algorithms on datasets with as many attributes does not seem to be feasible.

Task 5: Clustering using k-means

Both unsupervised and supervised K means on all of the emotion datasets results in two clusters with a ratio of 51% and 49% instances [2]. `weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 7 -A`

1	File	Accuracy (%)
2	fer2018	19.1
3	fer2018angry	50.9
4	fer2018disgust	51.3
5	fer2018fear	54.1
6	fer2018happy	50.8
7	fer2018neutral	51.6
8	fer2018sad	52.6
9	fer2018surprise	55.3

A Bash script kmeans.shwas used to run these tests via Weka CLI. Running the same K means algorithm, k = 7 on the selected best attributes produces the following results.

14	SelectKBest	Accuracy
15	Top 2	19.4
16	Top 5	19.5
17	Top 10	19.2

Task 6: Clustering using k-means - Conclusions

Running K-means with number of clusters as 2 on the emotion datasets showed accuracies of around 51%. This is slightly better than assigning classes purely randomly on a binary attribute (50%). Increasing the seed from 10 to 100 results in minimal change. Results here.

Compared to the previous steps 3-NN and bayes Nets results in accuracy increases from clustering. 3-NN and BayesNet found that disgust is the easiest emotion to detect while Kmeans found surprise to be the easiest. All algorithms showed a accuracy drop in happy and when testing the main fer2018 file, however all accuracies increased when running fer2018 on selectKbest attributes. Full results table can be found here.

Task 7: Alternate clustering methods

Below is a table of hand selecting and running various algorithms to find the optimal number of clusters and get better results. We found the following [3]:

	Algorithm	Top 2	Top 5	Top 10
2	EM	19 clusters 9.82	21 clusters 17.8	N/A
3	Kmeans k=10	10 clusters 14.1	10 clusters 14.2	10 clusters 15
4	Kmeans k=7	7 clusters 19.4	7 clusters 19.5	7 clusters 19.2
5	Kmeans k=5	5 clusters 22.6	5 clusters 22.1	5 clusters 22.1
6	Canopy	22 clusters 16.4	21 clusters 17.8	20 clusters 18.4
7	Farthest First	7 clusters 21.6	7 clusters 22.2	7 clusters 23.4
8	Filtered Cluster Kmeans	7 clusters 19.4	7 clusters 19.5	7 clusters 19.3
9	K Means Manhattan Distance	7 clusters 18.8	7 clusters 19.1	7 clusters 19.2
10		Most Accurate	Least Accurate	

The full weka settings and results of each test is available here.

Task 8: Alternate clustering methods - Conclusions

When running different clustering algorithms [4] and altering results we found the following from our test results:

- Increasing the clusters to 10 reduces accuracy while decreasing the clusters to 5 has improved accuracy from 7 clusters.
- When using EM weka to determine the number of clusters resulted in a reduced accuracy but greater number of clusters. Em algorithm can self determine an optimal number of clusters using cross validation and produce a probability distribution for each instance.
- Using other algorithms such as canopy to infer the number of clusters resulted in weka finding 7 as the optimal number of clusters. Canopy is less computationally expensive used for preprocessing data before using K-means algorithm or the Hierarchical clustering algorithm [1].
- In situations when there are more then 7 clusters weka assigns those clusters no classes and only uses 7 class clusters. The accuracy in this situation are similar to kmeans where k =7.
- Furthest First is modeled after kmeans, its can be used as a preprocessor for kmeans.
- Kmeans is the simples clustering algorithm, computationally faster [5] and may produce tighter clusters than hierarchical clustering if K is small.

Comparing these results we found that k2(2) has produces the highest accuracy overall. The full results table is available here. Generally speaking Bayes Nets and 2-KNN is more accurate than any clustering algorithms.

References

[1] McCallum, A., Nigam, K. and Ungar, L. (2018). *Efficient Clustering of High-Dimensional Data Sets with Application to Reference Matching*. [online] Kamalnigam.com. Available at: http://www.kamalnigam.com/papers/canopy-kdd00.pdf [Accessed 6 Nov. 2018].

[2] Jain, S., Aalam, M. and Doja, M. (2018). *K-MEANS CLUSTERING USING WEKA INTERFACE*. [online] Pdfs.semanticscholar.org. Available at: https://pdfs.semanticscholar.org/d3c6/5d73902c8a24865cca446e99ee8c0b0566a6.pdf [Accessed 7 Nov. 2018].

[3] Mishra, S. (2018). *Unsupervised Learning and Data Clustering – Towards Data Science*. [online] Towards Data Science. Available at: https://towardsdatascience.com/unsupervised-learning-and-data-clustering-eeecb78b422a [Accessed 7 Nov. 2018].

[4] Sharma, N., Bajpai, A. and Litoriya, R. (2018). *Comparison the various clustering algorithms of weka tools*. [online] Pdfs.semanticscholar.org. Available at: https://pdfs.semanticscholar.org/eca2/5eb78be04ffe09b029dd1d36f5ba66749f29.pdf [Accessed 7 Nov. 2018].

[5] E.B Fawlkes and C.L. Mallows. A method for comparing two hierarchical clusterings. Journal of the American Statistical Association, 78:553–584, 1983

Levels of contribution:

Mobeen Aftab - 33.33 %
Jonathan Mendoza - 33.33 %
Owen Welch - 33.33 %