Frankie Homewood - Student Number: 166952

1. Introduction

This analysis seeks to produce a method of classifying images from a partially analysed output of CaffeNET[4]. One data set has been labelled by three humans and deemed to be either "memorable" or "not memorable". These labels are what the classification algorithm seeks to automate, with a prediction of '1' representing the memorable class and '0' being the non-memorable class.

In addition to the task of constructing a classifier, there is the complication of having to account for the differences between the testing data and the training data. The testing data is sourced from images taken within the city of Brighton, Sussex. However the testing criteria, used to assess the efficiency of the classifier, is taken from images of London. As such, extra caution is taken for this difference in domain.

In order to construct this classification algorithm, we analyse the 4096 features extracted from CaffeNET as well as 512 GIST features, utilising pre-processing methods to accommodate for the change of scenery.

2. Approach

Here we employ a three stage pre-processing technique for analysing the data. Firstly the input data is standardised. Instead of feature selection, principal component analysis is used to extract optimum features from the data. The final stage of pre-processing, is the use of a clustering algorithm. This is to account for observed differences between training and testing data sets, so the clusters can be used to construct a composite data set with a shape that more closely resembles the testing data. A small subsection of the training data is then taken aside for model validation after the classifier has been trained.

After the clustering, the pre-processing of the information is complete and the training of a classifier can begin. In this analysis, the choice of classifier is a multi-layer perceptron (MLP), this MLP has a five layer structure such that an arbitrary decision boundary can be formed such that there is a high limit to the complexity of the decision boundary formed.

The principal component analysis process involves finding the set of eigenvectors, resulting in as set of composite features for the training data. Organising these features by those with the highest combined variance allows us to reduce the dimensionality by minimising information loss. When dimensionality needs to be reduced as in our case, the eigenvectors with the lowest variance are dropped leaving only the composite PCA features that carry the most information.

The k-means algorithm used is as a means of clustering the testing data, this is an algorithm that does not rely on any external training samples [2]. A set number of centres is assigned a random location in feature space, and a decision boundary drawn linearly between them. The centre of each cluster is then moved to the mean location of the data contained within the cluster. This process is iterated until the centres converge to the optimum location in feature space.

The MLP classifier used here has it's prediction power rooted in linear algebra where each layer acts as a matrix transformation on the layer before it, this reduces a trained MLP model down to a specific linear equation, making it trivial to implement once trained. To train an MLP there are two distinct phases, the forward pass and the backward pass[3]. In the forward pass we supply the MLP with a sample or a batch of the training data and perform the linear algebra to find out the model's predictions for the data. Subsequently, the predictions are compared to the target label and the difference between the two values is quantified in a loss statistic. After this step comes the backward pass, where the parameters within the model are progressively tweaked and back-propagated through the entire structure of the MLP to reduce the loss of the sample[3]. Upon iteration of this, on a large data set the MLP will generalise to any sample resembling the training data and produce accurate predictions.

3. Methodology

3.1. Pre-processing

Three data sets were provided as a basis for this analysis. The first was a small sample containing all of the features and also the human prediction for the image. The second is a larger set of data in which for each sample some of the values of the features are missing this selection also has the human predictions, these two sets were combined into one larger testing set for which a sample could be set aside for later validation checks. For this analysis a sample of 10% of the training data was used as a validation set. Finally there is a data set which doesn't have any predictions, this is the sample that is used to test the classification, thus is the testing data. The classifier's prediction for the testing labels are what is to be submitted as an assessment of the efficacy of this model.

The first issue that was addressed as a part of pre-processing was the missing data in the training set. These missing values make it unfeasible to train a classifier, yet replacing these values also brings some questions of which value should replace them. Initially we may think to replace these values with zero since they are not present however this would heavily skew the distribution of values for this

feature, thus it is more suitable to replace the missing values with the mean of that feature. This preserves the mean value of that feature whilst filling in the missing values. There are still potential disadvantages with this implementation as the likelihood of any given sample having the mean value increases, however this is a sufficient solution to the problem for the purposes of this model.

3.1.1 Standardisation

Out of the multiple thousands of features that are provided as a part of the training data, it is inevitable that some are going to have smaller variance than others. If these low variance features are useful in classifying the data it is likely that they will get overshadowed by the features with higher variation. As such, all features are standardised to have a mean of zero and a variance of one. Note that this does not remove any covariance between features.

3.1.2 Feature Extraction & Dimensionality Reduction

With the multiple thousands of features in each sample it is a matter of necessity to reduce the dimension of the data in order to allow for the limited processing resources available to train the classifier. In order to extract the features with highest variation and reduce the dimensionality of the data, principal component analysis (PCA) is applied, reducing the number of features eight fold down to 512 features. PCA was chosen as the method of dimensionality reduction as it preserves the features with highest variation, meaning that the minimum amount of information about the data is lost which optimises the distinguishability of the data[5].

3.1.3 Prior Knowledge

At this point the data is in a manageable format with standardised features. The training data set is plotted alongside the testing data to observe any differences that may occur due to the difference in the domain.

As displayed in Figure 1, the London domain clearly has a large region of data that is not present in the training data that was observed in Brighton. In order to account for the difference in domain, some assumptions are to be made about the data. From the brief, it is a reasonable assumption to say that there is a higher proportion of non-memorable images in London than there is in Brighton. As such the first inference is that the data within this cluster are images that don't resemble any of the Brighton images as such these images are assumed to be non-memorable. In order to separate the non-memorable cluster from the rest of the testing set, a k-means algorithm with 2 clusters was used. Then, in order to adapt the training data to a comparable domain to the testing data, random selections of the non-memorable cluster were added into the training set. The addition of the

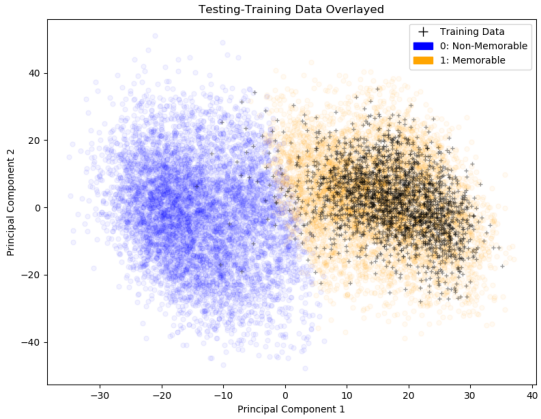

Figure 1. A representation of the testing and training data overlayed on top of one another. Here we see that there exists an entire separate lobe of the testing data that is not represented in the training data. This is assumed to be non-memorable images as they do not resemble the images found in Brighton

cluster had a number of effects, firstly this adapted the shape of training set to resemble the testing set. Secondly, the ratio of memorable to non-memorable images had changed. The number of additions to the testing data are such that the memorable and non-memorable images are equally represented. Which avoids potential feedback effects or tyranny of the majority from taking hold when the training data is skewed too far in favour of one of the classifications.

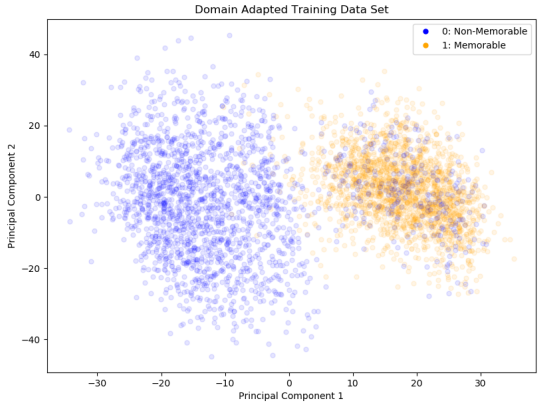

Figure 2. The composite training data with the additional non-memorable samples sourced from the London data, this addition balances the number of memorable and non-memorable in the training data.

3.2. Classification

Now with a set of training data that is more representative of the testing criteria, we begin to build the structure of the binary classifier. For this classification, a MLP is applied to the composite data set. The purpose of the MLP is to define a decision boundary based off of the composite data, this decision boundary can then be exported and applied to the entire testing data set, thus this boundary becomes the model for which we decide whether or not an image is memorable.

3.2.1 Architecture

The structure of the MLP used has three hidden layers in addition to the input and output layers that are set by the data restrictions. The hidden layers have a structure described in Figure 3

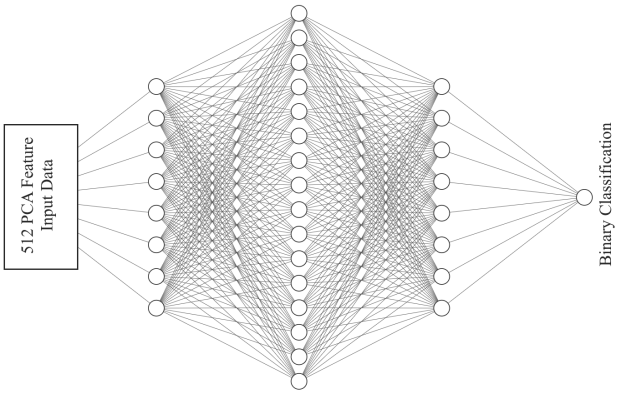

Figure 3. A diagram depicting the architecture of the MLP classifier trained in this analysis, each circle represents a node of the perceptron, every line represents a connection between the layers, all of the layers are fully connected (note that the connections between the input layer and the first hidden layer do not accurately represent the number of connections present)

This structure provides a total of 4360 different parameters that can be tuned to define a decision boundary for the training data. This number of parameters is greater than the number of samples provided so there exists the potential risk of over-fitting the data. However a structure like this with a significant amount of connections and three layers of depth allow for non-linearity in the decision boundary. Specifically, as we would not expect a linear decision boundary between these two classes, it allows us to define a decision boundary with the necessary complexity to account for the difference between memorable and non-memorable images.

4. Results

After the classifier is trained, it's important to evaluate whether or not this model has been over-fitted or whether it is still applicable to generalised data. As such we look to the validation set which contains all of the features as well as the associated label, this gives us an indication of the bias-variance trade off present in our model. By inputting the features into the classifier and comparing the MLP expectation values to the target labels we can assign an accuracy to the algorithm.

Figure 4. The accuracy measurements of the training and validation sets through each epoch of the MLP training cycle. Notice that the validation set has not begun to drop in accuracy, indicating that the MLP has not yet over-fitted to the training data.

Keep in mind that the validation set has not undergone any of the domain adaptation measures that the training data had, thus this may be an underestimate of the efficiency. The validation set showed a consistently high classification efficiency, where through all epochs, the accuracy was higher than 85%. This indicates that the trained MLP still correctly assesses data that it has not yet seen and as a result should continue to be applicable to the testing data. This indicates that the training was complete when the variance was high enough to accommodate unseen data points yet the bias was low enough to accurately classify both them and the training data effectively.

The classification algorithm employed here found to be 80.3% accurate when considering a random sample of 25% of the testing data. Although this does not consider the entirety of the testing accuracy, this is comprised of 3000 samples so should give a sufficiently detailed assessment of the performance. The classifier's predictions of the London images are depicted in Figure 5. This same classifier was found to have significantly lower accuracy when the domain considerations were not taken into account (Efficiency = 47.9%). As such it appears that the difference be-

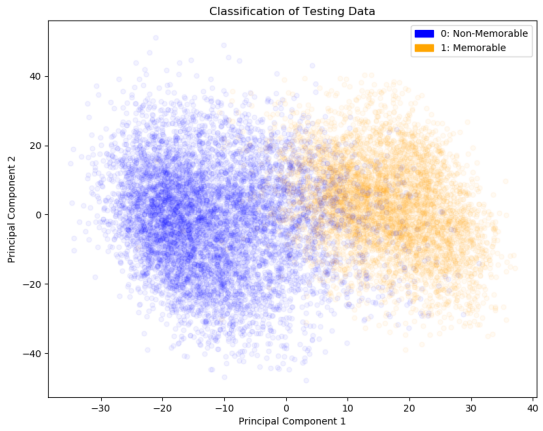

Figure 5. This data set depicts the testing set with labels assigned by the trained MLP classifier.

tween training and testing data is a significant source of the error in the assessment of the algorithm.

5. Discussion

The most notable consideration that has an impact on the efficacy of this algorithm is the issue of domain adaptation. As shown in Figure 1, the training and testing data sets have very different shapes and will have significantly different decision boundaries as such the accuracy of the decision boundary defined for the training data is only as good as the ability to adapt Brighton to London. This is evident from the classification efficiency of a simple k-means clustering of the training data which achieves ≈79.5% on the London sample. The method employed here utilises the k-means clustering algorithm to identify lobes of the testing data that are not present in the training data then applying the classifier to adapt the domain instead of making further assumptions about their target label, obviously this is not the optimum solution as it requires the initial estimation of the non-memorable cluster. An improvement to this method would acquire some sample of London images with associated labels, which would enable a known domain transformation to be created independent of the testing data.

Another notable area for discussion is the choice of the classifier used to predict the labels of the testing data. The classifier has a number of restrictions. It must be able to implement arbitrary decision boundaries meaning a single layer perceptron would not be sufficient, using a support vector machine (SVM) would also be suited to this task[1]. However, the non-linearity of an SVM must be supplied through non-linear activation functions or through composite features that are non-linear in nature. MLPs can achieve this arbitrary decision boundary with a minimum of three total layers[2]. The trade off between using different classifiers is important to consider, the deciding factor to use a MLP was due to its ability to potentially include over-fitting prevention measures such as drop out and further regularisation that cannot be achieved in an SVM.

Briefly mentioned in Section 3.1 was the approach to replacing missing data which was via replacement with the feature's mean value. This has some advantages and disadvantages, one of advantages is that it keeps the replacement within the range of expected values for that feature. A notable disadvantage is that the missing data are replaced with the same number for a given feature independent of which sample is selected. This could introduce some bias into the training data that is detectable by the MLP's model and could therefore be used to identify pieces of data that are missing certain features. Other methods besides mean replacement can be used to differentiate the replacement values between samples, such as multivariate imputers which estimate the value of the missing feature with consideration to the rest of the values in that sample.

The algorithm used to separate the clusters, as mentioned in Section 3.1.3 was k-means clustering which separates data into the two most likely clusters. This algorithm can be adapted via a number of different methods, where these adapted versions of k-means have potential to improve the identification of the cluster of data not present in the data set. A better identification of this cluster would have a significant effect on the way this classifier performs in practice as any overlap between the identified cluster and the training data is likely to significantly bias the classification model to consider these points as not memorable when in fact they should not be included in the non-memorable category simply due to the misidentification of the non-memorable cluster.

6. Conclusion

In conclusion the proposed model is sufficient in the classification of the memorability of images of London (from a combination of CNN output and GIST features). However, the model could be improved with greater access to a representative sample of London scenery that has been human labelled. The approach proposed here requires a minor degree of assumptions about the overall memorability of London and makes use of intuitive human experience of the differences between the cities of Brighton and London. There are numerous areas of improvement that are capable of being taken with the current training data and also with the prospect of forming further data sets to aid the shift in domain. Overall this model is very capable of classifying the testing set despite the complications in producing a model from non-representative training data.

References

[1] N. Cristianini and E. Ricci. *Support Vector Machines*, pages 928–932. Springer US, Boston, MA, 2008. 4

[2] P. A. Flach. Machine learning the art and science of algorithms that make sense of data, 2012. 1, 4

[3] S. S. Haykin. *Neural networks and learning machines*. Pearson Education, Upper Saddle River, NJ, third edition, 2009. 1

[4] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 1

[5] I. Jolliffe. *Principal Component Analysis*, pages 1094–1096. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. 2