Brief for the End of Year Assessment

Summary: You have to participate in the Kaggle competition and have to submit a 2-page report (using the provided template at the end of this description) and an implementation code.

As part of the practical assessment you are required to participate in the Kaggle in Class Competition "Do we just see snow in Brighton?". The assessment grade, which is worth 50% of the total grade, is separated into 2 components: the 2-page report and the source code. The code component will be weighted based on your performance in the Kaggle competition. No submission to the competition means 0.0 weight. You have to make at least one submission to the Kaggle competition to get a non-zero weight!

The competition is a binary classification problem (label 1 for snowy and label 0 for not snowy). You are provided with 456 labelled training data (268 of snowy scenes and 188 of not snowy scenes), and 5040 of test data, which are not labelled. The task is to develop a binary-class classifier that predicts the labels for the test data set. Each data instance is represented as a 4608 dimensional feature vector. This vector is a concatenation of 4096 dimensional deep Convolutional Neural Networks (CNNs) features extracted from the fc7 activation layer of CaffeNet and 512 dimensional GIST features (this representation is given therefore you do not need to perform any feature extraction on images).

Additionally, you are also provided with three types of information that might be useful when building your classifier: a) additional 4104 labelled training data which is incomplete as it has missing feature values, b) confidence of the label annotation for each training data point (456 labelled training data and additional but incomplete 4104 labelled training data), and c) the proportion of positive (snowy) data points and the proportion of not snowy data points in the test set. You can choose ("life is full of choices and consequences") to incorporate or to ignore these additional data.

You can use any of your favourite classifiers. Some of the classifiers that we have discussed are: linear perceptron, multi-layer perceptron, support vector machine, and logistic regression. You are not required to code the classifier from scratch. Feel free to use some of machine learning toolboxes such as Weka (in Java), scikit-learn (in Python; my favourite), shogun (in C++), or stats (in Matlab). I value your creativity in solving the classification problem with the 3 twists. You have to reason which classifier or combination of classifiers you use, how you handle issues specific to competition data set such as high dimensionality of the data (large number of features), how you do model selection (training-validation split or cross validation), and how you do further investigations to take into account the three extra information: additional but incomplete labelled training data, test label proportion and the annotation confidence on labels.

Details of Research Report

You are expected to write a 2-page report detailing your solution to the Kaggle competition problem. Please use the provided latex or word template (see the end part of this description). Your report should include the following components (you are allowed to combine descriptions #2 and #3 but make sure we can easily identify them).

1. APPROACH (Maximum mark: 10)

You should present a high-level description and explanation of the the machine learning approach (e.g. support vector machine, logistic regression, or a combination thereof) you have adopted. Try to cover how the method works and notable assumptions on which the approach depends. Pay a close attention to characteristics of the data set, for example: high dimensionality.

2. METHODOLOGY (Maximum mark: 25)

Describe how you did training and testing of the classifier of your choice. This should include model selection (Did you do model selection? what was it meant for? what were you selecting?) and feature pre-processing or feature selection if you chose to do it. Feature pre-processing could be in the form of:

- Standardisation: to remove the mean and scale the variance for each feature, that is to make each feature having 0 mean and 1 standard deviation.
- Normalisation: to scale individual observation or data point to have a unit norm, be it L1 or L2 norm.
- Binarisation: to threshold numerical feature values to get boolean values.
- Scaling: to scale features to lie between minimum and maximum values, for example to lie in [0,1] or [-1,1].

Feature selection methods are for example: filter methods such as univariate feature selection based on chi-squared statistics, wrapper methods such as recursive feature elimination, and L1 norm penalisation for sparse solutions. You are provided with two types of features: CNNs features and GIST features. Are they equally important?

Describe any of your creative solutions with respect to additional characteristics of the competition data set, such as how to incorporate the extra information about: additional training data with many missing features, test label proportions, and training label confidence.

Reference to appropriate literature may be included.

3. RESULTS AND DISCUSSION (Maximum mark: 25)

The main thing is to present the results sensibly and clearly. Present the results of your model selection. There are different ways this can be done:

Use table or plot to show how the choice of classifier hyper-parameters affect performance of the classifier using validation set (refer to lectures in week 9). Classifier hyper-parameters are for example, regularisation values in support vector machine and in logistic regression.

Use graphs to show changing performance for different training sets (learning curve; refer to lectures in week 9), if you choose to do that.

If any, provide analysis on the usefulness of taking into account the provided additional incomplete training data, test label proportions, and training label confidence.

You should also take the opportunity to discuss any ways you can think of to improve the work you have done. If you think that there are ways of getting better performance, then explain how. If you feel that you could have done a better job of evaluation, then explain how. What lessons, if any have been learnt? Were your goals achieved? Is there anything you now think you should have done differently?

Details of Code (Maximum mark: 40)

You must also submit your implementation codes. Please make sure we will be able to run your code as is. High quality codes with a good structure and comments will be marked favorably. As mentioned earlier, the code component will be weighted based on your performance in the Kaggle competition. No submission to the competition means 0.0 weight.