CSC 411: Introduction to Machine Learning
ASSIGNMENT # 3

Out: March 22, 2016

Due: April 9 (Saturday), 2016 [6pm Toronto time]

Submission

Hand in answers to all the questions below. The answers to your questions should be in pdf form and turned in along with your code. Package your code and a copy of the write-up pdf document into a zip or tar.gz file called A3-*your-student-id*.[zip—tar.gz]. Submit this file on MarkUs. Do not turn in a hard copy of the write-up.

Facial Expression Prediction (100 points)

In this assignment you will use machine learning tools to tackle a challenging problem on a real dataset. We use the Toronto Faces Dataset, where the task is to classify images of faces based on their expression. You will be given access to 2925 labeled images for training and validation. The goal is to write a program that will achieve the highest accuracy that you can manage on an unseen test set. In addition to the 2925 labeled data, you will be given access to 98,058 unlabeled images that you may use with an unsupervised or semi-supervised algorithm to help you with this task.

You are allowed to work in pairs and your solutions will be entered into a competition against your classmates. While you will be awarded marks for doing well in the competition, it is by no means the only way to achieve a high score. Rather, you will be graded on the creativity of your solutions, and the clarity with which you are able to explain them. If your solution does not live up to your expectations, then you should explain why and provide some ideas on how to improve it. You are free to use any third-party ideas or code that you wish as long as it is publicly available. You must provide references to any work that is not your own in the write-up.

We will be using the Kaggle platform (`http://inclass.Kaggle.com/`) in order to allow you to upload your solutions so that your results can be compared with those of your peers. First of all go to the Kaggle in Class website, and create a user account **with your university email address (ending in "mail.utoronto.ca")**. Further instructions will be posted shortly.

Description of the data

The Toronto Faces Dataset (`http://aclab.ca/users/josh/TFD.html`)[1] is a set of 32×32 grayscale images that contain faces that have been extracted from a variety of sources. The faces have been rotated, scaled and aligned to make the task easier. A small subset of the faces have been labeled by experts and research assistants based on their expression. These expressions fall into one of seven categories: **1-Anger, 2-Disgust, 3-Fear, 4-Happy, 5-Sad, 6-Surprise, 7-Neutral**. Examples of some faces with different expressions are shown in Figure 1.

The data has been separated into 4 parts: labeled, public test, hidden test, and unlabeled. You can download the labeled images, public test images, and unlabeled images. The Kaggle website will score your predictions on the public test images and rank your submissions from today until the submission deadline. Several days before the deadline, we will also release a set of hidden test images(so called because it will be hidden from you until a few days before deadline), and you will be asked to submit your final predictions for both public and hidden test images. Your score on the hidden test set is not shown to you, until after the deadline, so that you cannot optimize your models by trying different submissions.

Please note that each labeled face image has an identity, and there are different expressions associated with each person in the labeled set. When you split the labeled set to perform cross-validation on hyper parameter etc., you should make sure that people with the same identity are not put into both training and validation set. Having one person's face in both training and validation introduces a large noise.

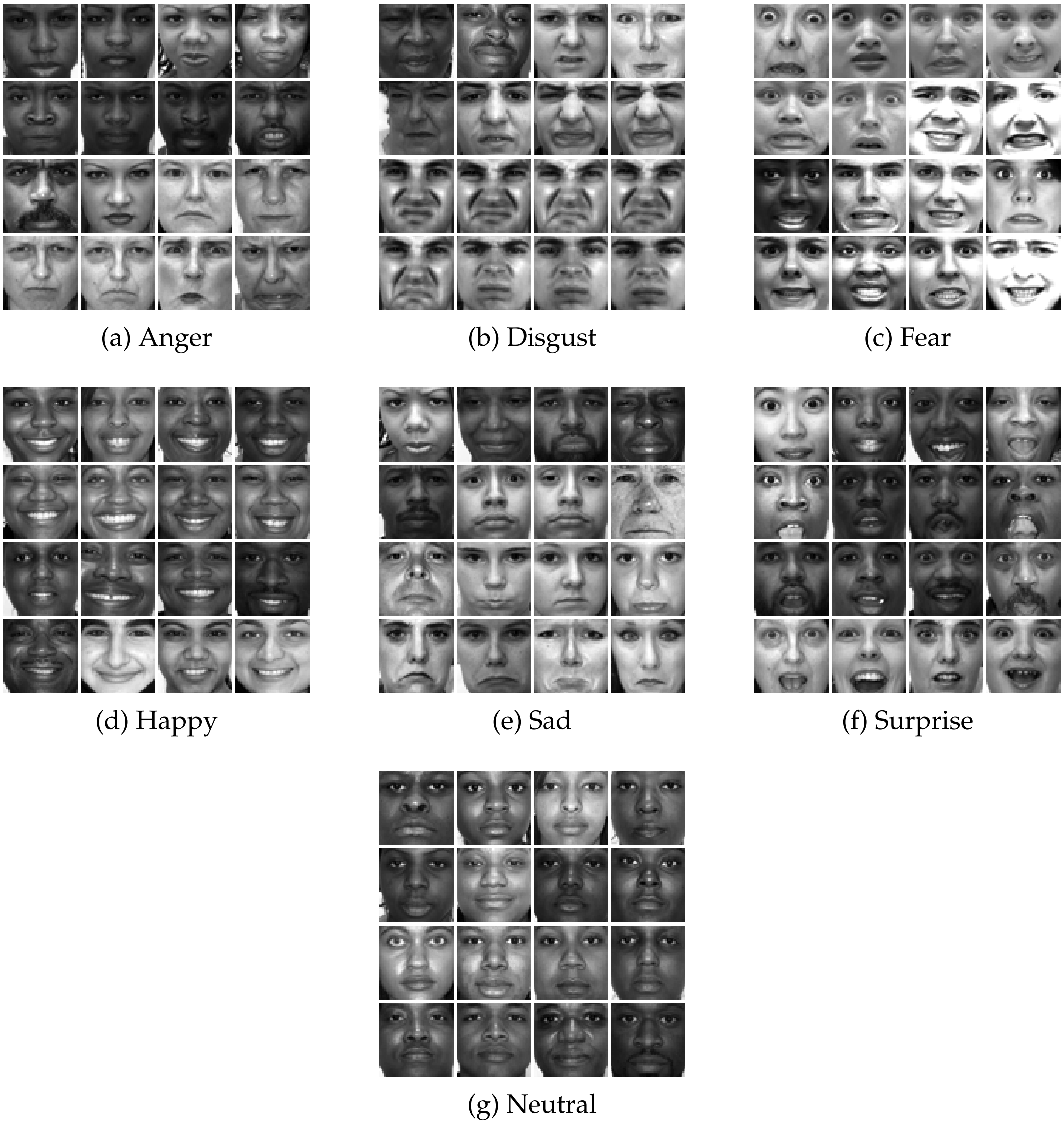
Baseline

In addition to the data, you are also provided with a baseline classifier. This baseline is a k-nearest neighbor classifier that uses only the labeled data to predict the label that appears the most in the Euclidean neighborhood of the public and hidden test images. The baseline achieves just under 60% accuracy on the public test set, which is not bad, but we expect that the winning methods achieve about $75 - 80\%$ accuracy.

Deliverables

Please hand in:

[1] Parts of the Toronto Faces Dataset are derived from sources that are not publicly available and usage of this dataset is therefore restricted to this assignment only. If you wish to use this dataset in the future, please contact Josh Susskind at `josh@aclab.ca`.



(a) Anger (b) Disgust (c) Fear

(d) Happy (e) Sad (f) Surprise

(g) Neutral

Figure 1: Training examples from the Toronto Faces Dataset.

1. Your code which takes the labeled data and unlabeled data, and returns a model that can be used to make predictions for test data. To save time, you should include a mat file with an already trained model called `trainedModel.mat` that you wish to use for the test set evaluation. You must also include a separate function that takes this model and returns a 1 of K matrix of predictions given test data. You should also save the predictions to a file called `test.mat`. Once your model is trained, it should take no more than 10 minutes to run on the test set.

2. A write-up containing no more than 6 pages that is single-spaced with a font no smaller than 12-pt Times New Roman. The write-up should include:

(a) An introduction describing at a high level some approaches that you considered, and why you considered them.

(b) A description of your submitted solution, including any data processing, algorithms used, etc.

(c) A section describing some empirical results from your solution. That is, some experiments that demonstrate that your solution is sensible. **This section must include a comparison of your approach against at least two other approaches that you have tried, with one of them being the kNN classifier included with this assignment**. Additionally, you may choose to include other experimental results such as an evaluation of a regularization technique to prevent overfitting, a comparison of different model parameters (such as tanh vs. sigmoid neurons/the number of neurons in a neural network, or different kernels/kernel parameter settings in an SVM) or the effects of using increasing amounts of unlabeled data in a semi-supervised learning algorithm. The idea is to justify your approach, and to demonstrate the different factors that you considered for your submitted solution.

(d) A conclusion summarizing your work and findings. If your method performed poorly on the public or hidden test data then you may want to include an explanation as to why and suggest how your method may be improved.

(e) References to any code, methods, or ideas that you used that are not your own. Remember, these must be publicly available and free to use.

(f) Any special instructions that are required to run your code.

If you are working for this part of the assignment as a team of two people, you and your partner should submit a single write-up. Please make sure to include both of your names, and Kaggle username.

Your grade will be based on 4 criteria:

1. **Classification performance:** You will get 10 points for beating the baseline and 1 more point for each 1.5% improvement from baseline up to a total of 20 points. Note that your performance will be evaluated on hidden test set, therefore you should not overfit to the labeled and public test set.

2. **Design and analysis of experiments:** You should follow the best practices of experimenting as discussed in the lectures and in your textbook. Using cross validation and performing statistical tests to prove the significance of your results will earn you points. Moreover you should follow a logical way to find and select the best hyperparameters.

3. **Report organization:** Writing a good report is as important as programming a good classifier. You should present your results in a clear and concise fashion. You should

create plots and tables to show important trends and results. Your reports should not exceed 6 pages including figures. It can be less than 6 pages, so don't try to make it too long. Reports should be typed, not hand-written. **Anything not in the 6-page write-up (comments in the source codes, outputs of the program, extra files you've sent with the code etc.) will be ignored.** You should give attention to details such as naming figures correctly, using the right mathematical terms, labeling the axes of plots, etc.

4. **Comments on method and results:** Your method should be motivated and justified from the structure of the dataset. You should explain for which cases your method worked well and for which cases it failed and why. You should comment on your overall performance and explain why is it high/low. You should also explain how you handled unlabeled data (You are free to not use it, but you should at least justify why you didn't). You should also try different variations of your method (regularization, different kernels/neuron types/distance metrics etc.) and comment on what changes when you try different variants.

Hand in answers to all the questions in the parts above. The goal of your write-up is to document the experiments you have done and your main findings. So be sure to explain the results. The answers to your questions should be in pdf form and turned in along with your code. Package your code and a copy of the write-up pdf document into a zip or tar.gz file called A3-*your-student-id*.[zip—tar.gz]. Only include functions and scripts that you modified. Submit this file on MarkUs. Do not turn in a hard copy of the write-up.

Hints

You are free to try any methods or approaches that you wish for this assignment. There are many possible solutions; in fact this dataset is currently being used as a benchmark for active research. There are many approaches that you can try, including discriminative methods (k-nearest neighbors, SVMs, kernel SVMs, neural networks, etc.), generative methods (mixtures of Gaussians, etc.) and dimensionality reduction (PCA, etc.). Your solution does not have to be limited to simply applying machine learning models. In the real world, the best approaches can often be simple ones that rely on clever data processing.

Your solution does not need to use the unlabeled data, in fact you can probably get very good results using only labeled data and you should start with this approach. Once you have a fully supervised solution, then you should start looking at semi-supervised extensions.

The facial expression prediction task is not intended to be a stressful exercise; instead it is a chance for you to experiment, to play and to hopefully have fun! Start with simple methods that work more or less "out of the box" and go from there.