Homework 1 Classifier Agent (CSC8850 Spring 2024, Georgia State University)

Instructor: Dr. Yi Ding

Logistics and timeline

- Due at 11:59 pm on February 13, 2024 (Tuesday)
- Be sure to read ”Academic Honesty” on the course syllabus
- You may discuss with your peers on the high-level but **each student must write his/her own codes and report**. You need to declare your collaborators. We will use software to automatically detect any plagiarisms.
- TA in charge of this coding homework: Nigar Khasayeva (nkhasayeva1@student.gsu.edu)

Introduction

In this homework, you will design an agent to read movie reviews and decide whether the review is positive or negative.

- Two examples of positive reviews

  ```
  This film took me by surprise. I make it a habit of finding
  out as little as possible about films before attending because
  trailers and reviews provide spoiler after spoiler. All I knew
  upon entering the theater is that it was a documentary about a
  long married couple. Filmmaker Doug Block decided to record
  his parents "for posterity" and at the beginning of the film
  we are treated to the requisite interviews with his parents,
  outspoken mother Mina, and less than forthcoming dad, Mike. I
  immediately found this couple interesting and had no idea
  where the filmmaker (Mike & Mina's son Doug) was going to take
  us. As a matter of fact, I doubt that Doug himself knew where
  he was going with this! Life takes unexpected twists and turns
  and this beautifully expressive film follows the journey. It
  is difficult to verbalize just how moved I was with this story
  and the unique way in which it was told. Absolutely riveting
  from beginning to end and it really is a must-see even if you
  aren't a fan of the documentary genre. This film will make you
  think of your own life and might even evoke memories that you
  thought were long forgotten. "51 Birch Street" is one of those
  rare filmgoing experiences that makes a deep impression and
  never leaves you.  BRAVO!!!!!!!!
  ```

Although I was hoping that I'd like it a little more, this was still certainly an impressive film. There were great performances by all the leads, and the story, while not what I'd call chilling, was still effective and it kept me interested. For me, the best part of this film was the look of the picture, for it always looked cold and damp and it just really seemed to suit the film well. I also thought that the low budget suited this movie, for I don't think that a crisp picture and clear sound would have worked as well in a film this grim. All things considered, it fell a little short of my expectations, but I'm still very glad that I finally sat down to watch this movie.

- Two examples of negative reviews

  While Star Trek the Motion Picture was mostly boring, Star Trek The Final Frontier is plain bad. In this terrible sequel, the crew is on shore leave when they get a distress signal from the Federation that ambassadors representing Earth, Romulus and Kronos (the Klingon home world) have been kidnapped by a renegade Vulcan bent on his quest to attain a starship to venture into the great barrier. There, he hopes to find God. This one is so bad it is hard to figure out where to begin. At the core is a good idea that is never really developed. The plot goes nowhere instead of where no man has gone before. It is almost like the writers had no idea how to end this fiasco. The action scenes don't have the suspense of Wrath of Kahn, the philosophy is boring, and the humor is stale. Now I will focus most of my anger on William Shatner. When he takes the director's chair, the ego gets bigger. Most of the focus is on him, Spock, and McCoy, but does not give the others enough to do. In any case, this is the worst of the Star Trek franchise. I should have given it three out of ten instead of five.

  I have now seen quite a few films by Pedro Almodóvar, but this would have to be the most disappointing so far. This film seemed to lack the zaniness that is usually everywhere in his films, and the story just never got me interested. Many Almodóvar regulars appear in this film, so it's not like there was a lack of on-screen talent, but this film just seemed more serious than his other films. If there was a comedic edge to this movie, I certainly couldn't find it, and it made for one surprisingly weak movie.

What do you need to do?

1. Basic coding requirments

The basic part of the homework requires you to complete the implemention of two python classes: (a) a "feature_extractor" class, (b) a "classifier_agent" class.

The "feature_extractor" class will be used to process a paragraph of text like the above into a **Bag of Words** feature vector.

The "classifier_agent" class will involve multiple functionalities of a **binary linear classifier** agent. These include functions for making predictions, learning from labeled training data and evaluating agent performance. For the learning part of it, you will implement the **gradient descent** and **stochastic gradient descent** algorithms that we learned from the lectures.

Through the exercise, you will understand the elements of an agent program and the Modeling / Inference / Learning paradigm of an AI agent design.

The implementation of the learning algorithm will require you to use a *slightly different* loss function to the version of the logistic loss that we learned in the lecture with $\mathcal{Y} = \{-1, 1\}$. We will be using what we called a Cross-Entropy loss that works with $\mathcal{Y} = \{0, 1\}$ instead. The cross-entropy loss for the linear classifier is defined as

$$\ell(w, (x, y)) = -(\log \hat{p}_w(x) y + \log(1 - \hat{p}_w(x))(1 - y)),$$

where

$$\hat{p}_w(x) = \frac{\exp(w^T x)}{1 + \exp(w^T x)}$$

is the probabilistic prediction of the classifier. Here $x \in \mathbb{R}^d$ is the feature vector and the weights $w \in \mathbb{R}^d$.

The training loss function will be the average of the cross-entropy loss over the training data, i.e.,

$$L(w) = \frac{1}{n} \sum_{i=1}^{n} \ell(w, (x_i, y_i)).$$

Make sure you check that your gradient calculations are correct before you implement it.

You should be able to get 85% test accuracy with this baseline classifier using the vanilla Bag of Words features.

2. (Optional) advanced coding requirements

The advanced version of the homework (bonus question) requires you to come up with better features than Bag-of-Words so as to improve the agent's classification accuracy.

A few suggestions are:

- a. tf-idf feature (see, e.g., https://en.wikipedia.org/wiki/Tf%E2%80%93idf)
- b. n-gram (see, e.g., https://en.wikipedia.org/wiki/N-gram)
- c. remove stopwords (Basically words to discard: https://en.wikipedia.org/wiki/Stop_word)

I would encourage you to try at least tf-idf feature, which will bump up your accuracy by quite a bit already;

Otherwise, the task is completely open-ended. You can use anything you like to improve the accuracy of this classifier's performance, such as using BERT features.

3. Report

You need to write a short report (with Jupyter notebook).

If you have hand-written parts, e.g., for the gradient derivation, you may scan them include them in the Jupyter notebook. The report (both an *.ipynb file, and a pdf file you print out) should be submitted to iCollege.

Setting up the Python platform

- Any platforms are fine, please use Python3.7 or above and make sure that the corresponding `numpy, scipy` packages are up to date.
- To install these standard python packages, I would suggest you to use package manage such as `pip` or `conda` .
- For debugging I suggest either using jupyter notebook or Python IDE such as PyCharm.
- There are many different platforms and it is hard for us to test exhaustively. Please try the code early and make sure that things works. If not, seek help from TA early. Do not wait until the last minute because then it will be devastating to be unable to work on the homework due to technical issues.
- For further Questions regarding setting up python environment / jupyter notebook and so on should be addressed to TA and peer help are encouraged.

The list of functions you need to complete

The main tasks for you is to design and train this classifier agent by implementing two different types of the feature extractors and two different types of the learning algortihms.

1. To complete *Basic Coding Requirements*, you need to implement the following functions in `classifier.py`
   - `bag_of_word_feature`
   - `score_function`
   - `predict error`
   - `loss_function`
   - `gradient`
   - `train_gd`
   - `train_sgd`

1. To complete *Advanced Coding Requirements*, you need to implement the following functions in `classifier.py`

   - `tfidf_extractor`
   - `compute_word_idf`

   For other feature extractors you should complete an implementation of

   - `custom_feature_extractor`

What to submit to iCollege

You need to submit two things:

1. The completed python module `classifier.py` with each functions implemented. Please make sure there are no syntax errors. You will be graded for each function you completed.

2. The parameters of your best trained model in a file `best_model.npy` Note that we will not run training for you. Your `custom_feature_extractor` class and this `parameter_vector` from `best_model.npy` will be used to instantiate a classifer that we will evaluate on new examples.

Grading rules

- Basic coding requirements: 70%
- Optional coding: 10% bonus
- Report: 30%