Computer Science 3113: Artificial Intelligence
University of Arkansas – Fort Smith
Spring 2023

Problem Set 3: Gradient Descent

50 points

Version 1.0

due Thursday, 9 March 2023, by 11:00 PM CT

Review the following assignment in its entirety prior to beginning. All submissions will be managed from within the course website.

Application Development

For this problem set, you will implement the *gradient descent* algorithm. Create a class named `PS3`. From the command line, your program should accept the following arguments:

1. Input file path that contains both $\{x_1, x_2, ..., x_p, y\}$
2. An integer representing the index location of the target variable y
3. Output file path to store the weights $\underset{(p+1)\times 1}{\mathbf{w}} = \{w_0, w_1, ..., w_p\}$
4. α – a floating point number representing the **learning rate**
5. ϵ – a floating point number representing the maximum **error** to tolerate.

You will train a model to learn the weights of a linear regression model from the input data $\underset{n\times(p+1)}{\mathbf{X}}$. Do not forget to add a column vector of 1's at the beginning of your design matrix. The data will need to be split into two groups: **training** and **testing sets**. You will need to use 80% of the data for training the weights and reserve 20% of the data for testing and calculating the final loss function error. Use the first $n = 48$ records for training and the last $n = 12$ records for testing data.

Data Normalization

Before you begin, you will notice that the input file will contain features that are several orders of a magnitude larger than other features. This can make convergence take a long time. As a result, you will need to scale the $\mathbf{X}$ and $\mathbf{y}$ feature values such that one vector of data is relative to another vector of data. For example, if you consider salary and GPA, these two vectors are clearly of different scale. The scaling of data is referred to as the **normalization** of data. There are many options available for this. We will use z-scores as previously discussed in lectures.

Loss Function

For the loss function $\mathcal{L}\big(\mathbf{y}, h_{\mathbf{w}}(\mathbf{X})\big)$, we will slightly modify it from our definition in class and define it as the average of the following quadratic function:

$$\mathcal{L}\big(\,\mathbf{y}, h_{\mathbf{w}}(\mathbf{X})\,\big) = \frac{1}{2n}\sum_{j=1}^{n}\Big[\; y^{(j)} - h_w\big(\mathbf{x}^{(j)}\big)\;\Big]^2 \qquad \text{where } h_w(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$$

Gradient Descent Algorithm

The pseudocode that you will need to implement is provided below.

Algorithm 1 Batch Gradient Descent

procedure GRADIENT-DESCENT($\mathbf{X}$, $\mathbf{y}$, α)
  $\mathbf{w} \leftarrow$ initialize to be any point in the parameter space ▷ This is a new vector
  $i = 0$ ▷ Use this to keep track of the current epoch
  while not converged **do**
    for $k = 0$ to p **do** ▷ loops through $p+1$ features (plus bias node)
      $w_k = w_k - \alpha \dfrac{\partial}{\partial\, w_k}\Big[\mathcal{L}\big(\,\mathbf{y}, h_{\mathbf{w}}(\mathbf{X})\,\big)\Big]$
    $i \leftarrow i + 1$
    Update convergence values

For the convergence criteria, you will need to use the constraint $\Delta_{\%\,cost} < \epsilon$ where ϵ is a parameter in your application we will select:

$$\Delta_{\%\,cost} = \frac{\mid \mathcal{L}_{i-1}\big(\,\mathbf{y}, h_{\mathbf{w}}(\mathbf{X})\,\big) - \mathcal{L}_i\big(\,\mathbf{y}, h_{\mathbf{w}}(\mathbf{X})\,\big) \mid \times 100}{\mathcal{L}_{i-1}\big(\,\mathbf{y}, h_{\mathbf{w}}(\mathbf{X})\,\big)}$$

We will define $\mathcal{L}_0\big(\mathbf{y}, h_{\mathbf{w}}(\mathbf{X})\big)$ using a vector of weights $\mathbf{w}$ such that:

$$\mathbf{w} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}_{(p+1)\times 1}$$

Report and Program Output

Once your program has finished training and the convergence criteria has been met, you will output the final weights (line-delimited) to the file specified as a runtime argument. In addition, you should have the following information included in your report and printed to the console (formatted properly using `printf()`):

1. Number of Input Records (n)
2. Number of Features (p)
3. Y-intercept (w_0)
4. Iterations Required for Convergence in Training Data
5. Loss Function output for the Testing Data
6. Loop through the records (the last 12 records in the dataset for simplicity) in your testing data and output the true y value and your model's predicted value $h_{\mathbf{w}}(\mathbf{x})$ (output the true values rather than the scaled values)

Finally, output the iteration number and the loss function output to a separate file for you to create a chart using R or Excel. The chart should plot the error as a function of the iteration number. Include this in your report.

Your program should have the following output.

```
************************************************************
Problem Set:  Problem Set 4:  Gradient Descent
Name:         Andrew Mackey
Synax:        java PS3  arg1  arg2  arg3  arg4  arg5
************************************************************

Training Phase:  /path/to/training/file  <-- use real filename
------------------------------------------------------------
    => Number of Entries (n):              10000
    => Number of Features (p):             5


Starting Gradient Descent:
------------------------------------------------------------

Epoch 1:    Loss of 123.45     N/A              N/A
Epoch 2:    Loss of 111.45     Delta = 10%      Epsilon = 3%)
Epoch 3:    Loss of 101.45     Delta = 8%       Epsilon = 3%)
Epoch 4:    Loss of 99.45      Delta = 3%       Epsilon = 3%)

Epochs Required:   4

Resulting Weights:
W0 (y-intercept):   123.45
W1:                 45.78
W2:                 45.78
W3:                 45.78
...

Testing Phase:
------------------------------------------------------------
     Loss of Testing Data:           12345.456

     Test Record 1:  True: 120.22    Prediction:  123.45    Error:   3.12
     Test Record 2:  True: 120.22    Prediction:  123.45    Error:   3.12
     ...

    => Number of Test Entries (n):              12
```

Deliverables

You will be responsible for delivering the following items:

1. Latex Documents – your latex code should be used to generate a PDF which will then be submitted. Be sure that all documents submitted list your name, problem set information, date and class. A chart should be included in your report.

2. Application Code – be sure to submit your source code as indicated above to the `code` server and the course website. A copy of this should also be uploaded to the course website. All files should be contained in a directory named `ai/ps#` in your home directory (all lower case). All submitted code must have your 1) name 2) username (code server) 3) problem set number and 4) due date as a comment at the top of each class.

   ```
   /*********************************
   Name:           Andrew Mackey
   Username:       ua12345
   Problem Set:    PS1
   Due Date:       Month day, YEAR
   *********************************/
   ```

   Be sure to remove any extraneous code that is unnecessary prior to submission.