**Comparing Gradient Descent Methods in Predicting Crime Rate**

**MAS 3105 Final Project**

Andrea Llinas

**Introduction**:

In this report, I will be comparing the effectiveness of different Gradient Descent variants in predicting crime rates across multiple Florida counties. The objective is to model the relationship between crime rates (the dependent variable) and two independent variables: income and high school graduation rate (HS). The three algorithms under investigation are:

1. **Gradient Descent with Fixed Step Size**: A standard iterative optimization technique used for minimizing the cost function, where the step size is fixed throughout.
2. **Stochastic Gradient Descent (SGD)**: An enhancement to Gradient Descent that updates parameters after processing each training example, leading to faster convergence in many cases.
3. **Gradient Descent with Momentum**: A variant of Gradient Descent that helps accelerate convergence by considering the previous gradients (direction and magnitude of the past updates), smoothing out updates and potentially avoiding local minima.

By understanding the strengths and weaknesses of each algorithm in modeling crime rates, we can identify the most effective optimization technique for this type of regression problem, particularly in scenarios involving large datasets or non-convex loss surfaces.

**Discussion**:

The problem this project addresses is understanding the efficiency and performance of these algorithms when applied to predicting crime rates using socio-economic data. Each method will be evaluated based on convergence time, number of iterations, and predictive accuracy.

To solve this problem, I will apply the three gradient descent methods to a linear regression model that predicts crime rates using income and high school graduation rate as predictors.

**Gradient Descent with Fixed Step Size**:

* + This method finds the minimum of the cost function iteratively by taking steps in the direction of the negative gradient.
  + Uses the entire dataset for each update.
  + **Advantages**: Provides a stable convergence rate when the step size is well chosen.
  + **Disadvantages**: Can be computationally expensive and slower in cases where finding the optimal step size is difficult or when the problem's dimensionality increases.

1. **Stochastic Gradient Descent (SGD)**:
   * Unlike standard Gradient Descent, SGD updates the parameters after processing each data point (or small batch).
   * **Advantages**: Faster convergence due to frequent updates, and it can escape local minima more effectively.
   * **Disadvantages**: More variance in the updates, which can result in oscillations around the minimum. It requires careful choosing of the step size.
2. **Gradient Descent with Momentum**:
   * This method incorporates the previous gradients into the current update, using a momentum term that accelerates the search for the minimum by smoothing out the steps.
   * **Advantages**: Can converge faster and avoids the oscillation issues faced by SGD. It is especially effective when there is sharp curvature in the cost function.
   * **Disadvantages**: Requires careful choosing of an additional momentum parameter and can still have some instability if not carefully managed.

**Implementation and Results**:

To compare the effectiveness of Gradient Descent with Fixed Step Size, Stochastic Gradient Descent (SGD), and Gradient Descent with Momentum, I implemented each algorithm in Python and applied them to a dataset (Florida.dat) containing crime rates across Florida counties. As mentioned in the introduction, Crime is the dependent variable (Y) and Income and HS (high school graduation rate) are the independent variables (X).

Each method was evaluated based on the number of iterations required to converge, the final values for Intercept, Income, and HS, and Runtime.

The table below summarizes the parameter estimates (Intercept, Income, HS) and the number of iterations required for each method:

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Gradient Descent with Fixed Step Size | Stochastic Gradient Descent | Gradient Descent with Momentum |
| Intercept | 52.40061786 | 52.30504008 | 52.40298507 |
| Income | 5.07344906 | 4.78878591 | 4.79226272 |
| HS | 8.98569416 | 9.26698609 | 9.26688073 |
| Iterations | 10000 | 10000 | 6339 |
| Runtime (in Seconds) | 0.12371635437011719 | 10.848258018493652 | 0.07975316047668457 |

From the results we can see that the methods had similar Intercept, Income, and HS values with Gradient Descent with Fixed Step Size diverging the most. We can see that Gradient Descent with Momentum requires significantly less iterations than Gradient Descent with Fixed Step Size and Stochastic Gradient Descent. Stochastic Gradient Descent takes the longest amount of time to converge while Gradient Descent with Momentum takes the least amount of time.

Note: The runtime varies slightly when the code is rerun.

**Conclusion:**

Through implementing and comparing Gradient Descent with Fixed Step Size, Stochastic Gradient Descent (SGD), and Gradient Descent with Momentum, I gained valuable insights into the efficiency and performance of these optimization methods in predicting crime rates across Florida counties.

Gradient Descent with Fixed Step Size, while reliable, required the max number of iterations to converge, highlighting its computational inefficiency. Similarly, Stochastic Gradient Descent also required the max number of iterations to converge and the variability in its updates led to more oscillations and potentially less stability as well as significantly longer to converge. However, it does have the ability to escape local minima. Gradient Descent with Momentum proved to be the most efficient method, requiring significantly fewer iterations to converge while converging the fastest. Its ability to smooth updates and accelerate convergence demonstrates its strength in handling optimization problems with sharp cost function curvatures.

Overall, this analysis has shown me the importance of selecting the appropriate Gradient Descent variant based on the problem context. While each method has its strengths and limitations, Gradient Descent with Momentum was the most effective for this dataset, balancing computational efficiency and accuracy. These findings provide a strong foundation for applying advanced optimization techniques in predictive modeling and similar data-driven tasks.

Sources:   
Dataset: <https://stat4ds.rwth-aachen.de/data/Florida.dat>

The python code was written using notes and resources from another class.