CS 229 Machine Learning, spring 2020

Homework 1: Regression

Due Tuesday February 11, 11:59pm

Submit by the **blackboard system**

***Requirements:***

1. Discussion is encouraged, but independent write-up and submission in your own hand is required for credit.
2. The answers copied from textbook or internet will be graded as a score of 0 point.
3. The work should be written in a \*clear\* way.
   1. The process of your solution must be unambiguously outlined or explicitly shown.
   2. The explanation and discussion must be well-founded by examples, tables, figures, and literatures and so on.
   3. The work should be submitted electronically in pdf/doc format. The scan of hand-written answers is acceptable.
4. Work submitted after the due date will be graded for correction, but not credited.

The goal of this homework is to become familiar with the **gradient descent algorithms and maximum likelihood algorithm**.

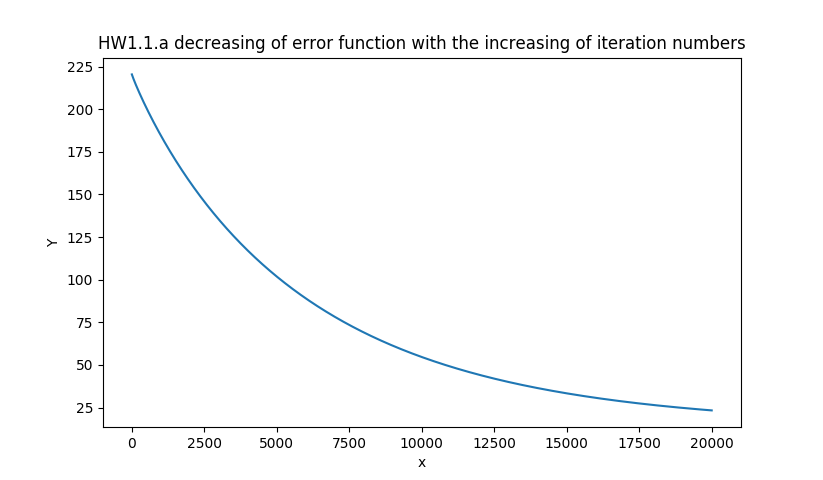
**Data**: Please download the data regression\_x\_t.txt. It includes two columns, one is the variable ***x***, and the other is the target ***t***. Randomly choose 80% of the data for learning the regression function. Use the remaining 20% for evaluating the learned function.

**Code:** Write your code by any programming languages and submit your results together with the programs.

**Regression basis function:** Choose any kind of basis function, e.g., polynomial function, or Gaussian basis function. Try an appropriate number of basis functions.

**Task:**

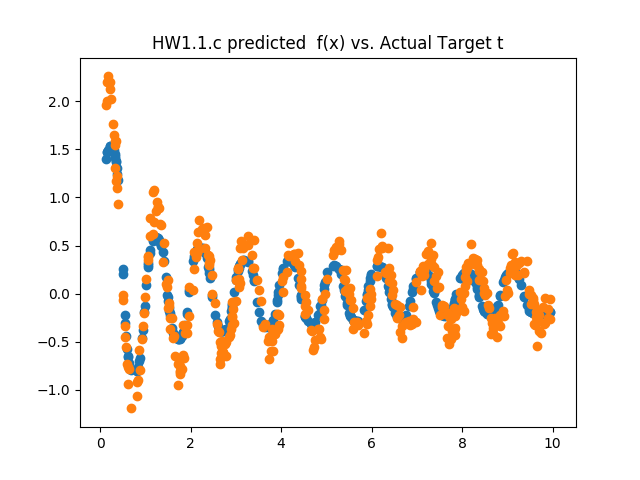
1. (3 points) implement the ***batch gradient descent*** algorithm.
2. Show the decreasing of error function with the increasing of iteration numbers.



1. Give the results of obtained coefficient, **w**

|  |
| --- |
| HW1.1.b BGD Sigmoid obtained coefficient W  [[ 1.59219231]  [-2.49392843]  [ 1.52674377]  [-1.1315709 ]  [ 1.02080263]  [-0.92879687]  [ 0.79150522]  [-0.74858997]  [ 0.7379945 ]  [-0.69892411]  [ 0.64494808]  [-0.61462389]  [ 0.61220291]  [-0.58042568]  [ 0.52007124]  [-0.49519312]  [ 0.48173485]  [-0.46727258]  [ 0.47673656]  [-0.45719155]  [ 0.01825092]] |

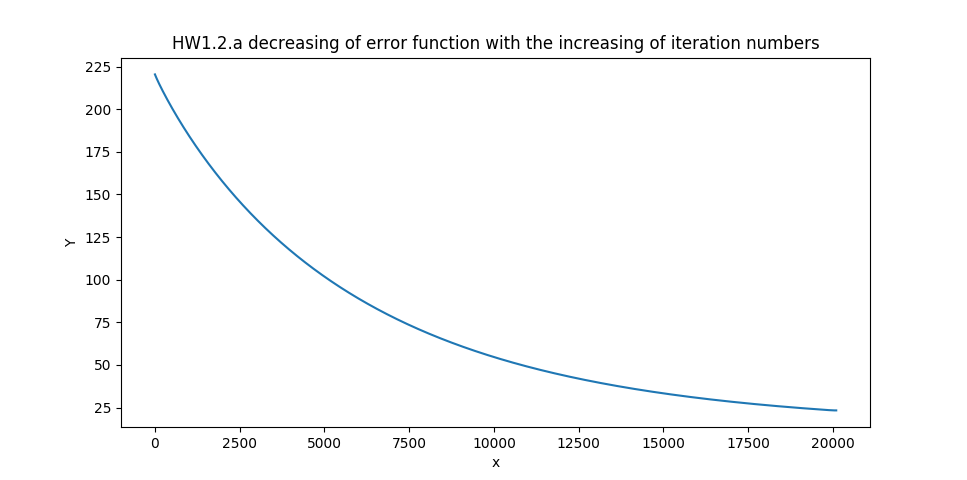
1. Show the predicted ***f*(*x*)** when applying the learned regression function on testing data ***x***, and compare it with the corresponding actual target ***t*** on the same figure;



1. What is the Root-Mean-Square Error on test set?

HW1.1.d BGD Sigmoid Test Set RMS: 13.267

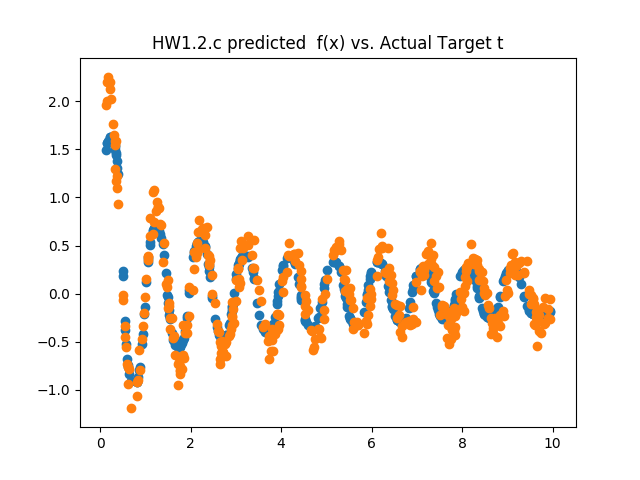
1. (3 points) implement the ***stochastic gradient descent*** algorithm.
2. Show the decreasing of error function with the increasing number of used data points.



1. Give the results of obtained coefficient, **w**

|  |
| --- |
| HW1.2.b Stochastic obtained coefficient W  [[ 1.72279856]  [-2.79087074]  [ 1.84342549]  [-1.40192314]  [ 1.23523401]  [-1.11404155]  [ 0.95501705]  [-0.89373512]  [ 0.86987483]  [-0.82261356]  [ 0.76368231]  [-0.72594517]  [ 0.71960648]  [-0.68468477]  [ 0.62327678]  [-0.60126736]  [ 0.58663021]  [-0.56675627]  [ 0.56068145]  [-0.51100395]  [ 0.05356434]] |

1. Show the predicted ***f*(*x*)** when applying the learned regression function on testing data ***x***, and compare it with the corresponding actual target ***t*** on the same figure;



1. What is the Root-Mean-Square Error on test set?

Stochastic RMS: 13.96

1. (4 points) implement the ***maximum likelihood*** algorithm.
2. Give the results of obtained coefficient, **w**

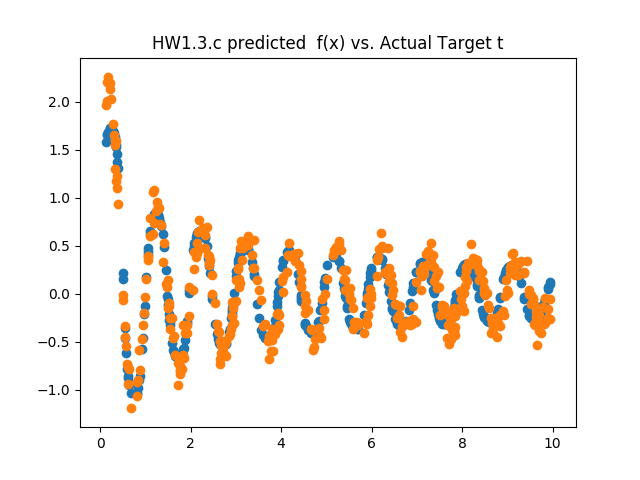
|  |
| --- |
| HW1.3.a Maximum Likelihood W  [[ 1.6943919 ]  [-2.72439831]  [ 1.76732402]  [-1.33382776]  [ 1.1797114 ]  [-1.06794118]  [ 0.91504973]  [-0.85754215]  [ 0.83835814]  [-0.79324981]  [ 0.73588923]  [-0.70006903]  [ 0.69411051]  [-0.65944309]  [ 0.59787229]  [-0.57264802]  [ 0.55987376]  [-0.5407762 ]  [ 0.54051462]  [-0.49779842]  [ 0.04355269]] |

1. Compare the value of **w** obtained at (1) and (2) by gradient descent algorithm, and (3) by maximum likelihood

as we can observe from the following table, the values are very close to each others. High level of similarity with a small difference.

|  |  |  |
| --- | --- | --- |
| BGD | Stochastic | Maximum likelihood |
| [[ 1.6943919 ]  [-2.72439831]  [ 1.76732402]  [-1.33382776]  [ 1.1797114 ]  [-1.06794118]  [ 0.91504973]  [-0.85754215]  [ 0.83835814]  [-0.79324981]  [ 0.73588923]  [-0.70006903]  [ 0.69411051]  [-0.65944309]  [ 0.59787229]  [-0.57264802]  [ 0.55987376]  [-0.5407762 ]  [ 0.54051462]  [-0.49779842]  [ 0.04355269]] | [[ 1.72279856]  [-2.79087074]  [ 1.84342549]  [-1.40192314]  [ 1.23523401]  [-1.11404155]  [ 0.95501705]  [-0.89373512]  [ 0.86987483]  [-0.82261356]  [ 0.76368231]  [-0.72594517]  [ 0.71960648]  [-0.68468477]  [ 0.62327678]  [-0.60126736]  [ 0.58663021]  [-0.56675627]  [ 0.56068145]  [-0.51100395]  [ 0.05356434]] | [[ 1.6943919 ]  [-2.72439831]  [ 1.76732402]  [-1.33382776]  [ 1.1797114 ]  [-1.06794118]  [ 0.91504973]  [-0.85754215]  [ 0.83835814]  [-0.79324981]  [ 0.73588923]  [-0.70006903]  [ 0.69411051]  [-0.65944309]  [ 0.59787229]  [-0.57264802]  [ 0.55987376]  [-0.5407762 ]  [ 0.54051462]  [-0.49779842]  [ 0.04355269]] |

1. Show the predicted ***f*(*x*)** when applying the learned regression function on testing data ***x***, and compare it with the corresponding actual target ***t*** on the same figure;



1. What is the Root-Mean-Square Error on test set? Which method has the best performance? Batch gradient descent, stochastic gradient descent or maximum likelihood?

HW1.3.d Maximum Likelihood RMS: 14.499

All three methods have achieved similar and very close RMS. The BGD one was the lowest with 13.27. Although, BGD and stochastic perhaps could have achieved a better number with the increase of iterations.