CSE 5523: Machine Learning Fall 2020

Project

Instructor: Raef Bassily **Due on:** Mon, Nov 30

Please, read the following instructions carefully

- The is a mini-project where it is required to implement and test the SGD algorithm for logistic regression in different scenarios. Please, read all parts of this assignment carefully.
- Your submission should be in the form of a report that includes
  - A brief introduction containing a description of the goals and a high-level description of your procedure (you may augment that with a pseudo-code if it helps with clarity).
  - A section on your experiments where you explain clearly and concisely how you devised the experiments (in the light of the guidelines and specifications provided in this assignment), provide a description of each task involved **including the generation of training and test data** stating precisely the specific setting of each parameter to be used in any task (that is, do not leave any parameter unspecified).
  - A section on the results of your experiments, where you state and discuss your results, and include all the relevant figures.
  - A conclusion where you state the main findings.
  - A symbol index listing all the symbols used for the parameters and variables in your code and what they denote.
  - An appendix that contains a *well-documented* copy of your code.
- **Please, follow the template given at the end of this document.**
- You can use any language you prefer, e.g., MATLAB, Python, C, etc.
- If you are going to use a function from an existing package/library, you must clearly describe (preferably in a separate section) the inputs (including all the relevant parameters) and outputs of such a function. Also, whenever this function is invoked in your code, you must clearly and explicitly show the setting of each of the input parameters and the format of the output to be expected. Please, *do not* copy and paste the documentation found in the help folder of the package or any vague description you found online. You need to describe **concisely only the relevant functionality and the relevant parameters settings.**
- Building your own version of SGD is highly recommended.
- All submitted plots must clearly show/describe all the variables on the axes, and you need to add a caption to describe briefly the plots in each figure, and (if applicable) the setting that each plot represents.
- Use different line styles or marks for plots that are on the same figure. It is recommended that you use figure legends. If you will eventually print your final submission in black-and-white, please, do **not** use different colors to distinguish different plots on the same figure. Instead, use different line styles or different marks on your plots.

Stochastic Gradient Descent for Logistic Regression

Recall the logistic loss function we defined in class:

$$\ell_{\mathsf{logist}}(\mathbf{w}, (\mathbf{x}, y)) = \ln(1 + \exp(-y\langle \mathbf{w}, \tilde{\mathbf{x}} \rangle))$$

where $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^{d-1}$, $\tilde{\mathbf{x}} \triangleq (\mathbf{x}, 1)$, $y \in \{-1, +1\}$, and $\mathbf{w} \in \mathcal{C} \subset \mathbb{R}^d$. We will consider two scenarios, each with a different setting for $\mathcal{X}$ and $\mathcal{C}$. **In both scenarios, the dimensionality parameter d is 5.**

Scenario 1

- The domain set $\mathcal{X} = [-1, 1]^{d-1}$, i.e., $\mathcal{X}$ is the 4-dimensional hypercube with edge length 2 and centered around the origin.
- The parameter set $\mathcal{C} = [-1, 1]^d$, i.e., $\mathcal{C}$ is the 5-dimensional hypercube with edge length 2 and centered around the origin.

Scenario 2

- The domain set $\mathcal{X} = \left\{\mathbf{x} \in \mathbb{R}^{d-1} : \ \|\mathbf{x}\| \leq 1\right\}$, i.e., $\mathcal{X}$ is the 4-dimensional unit ball centered around the origin.
- The parameter set $\mathcal{C} = \left\{\mathbf{w} \in \mathbb{R}^d : \ \|\mathbf{w}\| \leq 1\right\}$, i.e., $\mathcal{C}$ is the 5-dimensional unit ball centered around the origin.

For each scenario, show that there is a constant ρ such that for all $z \in \mathcal{X} \times \{-1, +1\}$, $\ell_{\mathsf{logist}}(\cdot, z)$ is convex and ρ-Lipschitz over $\mathcal{C}$ and that $\mathcal{C}$ is M-bounded for some $M > 0$. Specify ρ and M for each of the two scenarios. (Note that the values of ρ and M may not be the same for the two scenarios.)

Data Distribution D : In practice, the data distribution is usually unknown. However, since you will be asked to generate training and test examples for the sake of running your experiments, we will describe a data distribution from which examples will be generated for each scenario. (Nevertheless, note that the SGD learner should remain oblivious to the distribution). Each example $(\mathbf{x}, y)$ is generated as follows.

- with probability $1/2$, set $y = -1$ and generate a $d-1$-dimensional Gaussian vector $\mathbf{u} \sim \mathcal{N}(\boldsymbol{\mu}_0, \ \sigma^2 \mathbb{I}_{d-1})$ where $\boldsymbol{\mu}_0 = (-1/4, -1/4, -1/4, -1/4)$ and $\mathbb{I}_{d-1}$ is the identity matrix of rank $d-1$, that is, $\mathbf{u}$ is composed of 4 i.i.d. Gaussian components, each of mean $-1/4$ and variance σ^2 (σ will be specified later).
- with the remaining probability, set $y = 1$ and generate $\mathbf{u} \sim \mathcal{N}(\boldsymbol{\mu}_1, \sigma^2 \mathbb{I}_{d-1})$ where $\boldsymbol{\mu}_1 = (1/4, \ 1/4, \ 1/4, \ 1/4)$.

Then, set $\mathbf{x} = \Pi_{\mathcal{X}}(\mathbf{u})$ where $\Pi_{\mathcal{X}}$ is the Euclidean projection onto $\mathcal{X}$, that is, $\mathbf{u}$ generated above is projected onto $\mathcal{X}$ (in case it lies outside $\mathcal{X}$) and the resulting vector is $\mathbf{x}$.

Note that the procedure above will be used in both scenarios to generate examples for training and testing, however, since $\mathcal{X}$ is different in the two scenarios, the projection step described above will be different.

Let n denote the number of training examples (that will be used by the SGD learner to output a predictor), and let N denote the number of test examples that will be used to evaluate the performance of the output predictor on fresh examples. **The number of test examples, N, will be fixed in all experiments to 400 examples.**

Experiments

Let $L(\mathbf{w}; D) \triangleq \underset{(\mathbf{x},y)\sim D}{\mathbb{E}}\left[\ell_{\mathsf{logist}}(\mathbf{w}, (\mathbf{x}, y))\right]$ denote the risk incurred by a predictor $\mathbf{w} \in \mathcal{C}$ under the logistic loss model w.r.t. the distribution D. Let $\mathbf{err}(\mathbf{w}; D) \triangleq \underset{(\mathbf{x},y)\sim D}{\mathbb{E}}\left[\mathbf{1}\left(\text{sign}\left(\langle \mathbf{w}, (\mathbf{x}, 1)\rangle\right) \neq y\right)\right]$ denote the binary classification error (the risk under '0-1' loss) incurred by $\mathbf{w}$ w.r.t. the distribution D.

For each scenario above, it is required to conduct a set of experiments on the performance of the SGD learner, each experiment represents a different setting of the parameters σ, n. **Namely, for each of the two scenarios, for each $\sigma \in \{0.1,\ 0.35\}$, it is required to plot**

- **an estimate of the expected excess risk** of the SGD learner, namely, $\underset{\widehat{\mathbf{w}}}{\mathbb{E}}\left[L(\widehat{\mathbf{w}}; D)\right] - \min_{\mathbf{w}\in\mathcal{C}} L(\mathbf{w}; D)$ where $\widehat{\mathbf{w}}$ is the output predictor of the SGD given n training examples,
- **an estimate of the expected classification error** of the SGD learner, namely, $\underset{\widehat{\mathbf{w}}}{\mathbb{E}}\left[\mathbf{err}(\widehat{\mathbf{w}}; D)\right]$

versus the number of training examples, n, (which is equal to the number of iterations of the SGD), for $n = 50,\ 100,\ 500,\ 1000$. On your plots, using error bars, it is also required to show the standard deviation of your estimates. That is, for each estimate for the expected excess risk (and each estimate for the expected classification error), you need to provide an estimate for $\sqrt{\mathbf{Var}_{\widehat{\mathbf{w}}}\left[L(\widehat{\mathbf{w}}; D)\right]}$ (and $\sqrt{\mathbf{Var}_{\widehat{\mathbf{w}}}\left[\mathbf{err}(\widehat{\mathbf{w}}; D)\right]}$) shown as an error bar on your plots for the respective expected quantities. Refer to the procedure outlined below for obtaining these estimates.

Obtaining estimates of the expected performance of SGD:

- For each setting of n and σ, in order to obtain an estimate for the **expected** performance of the output predictor $\widehat{\mathbf{w}}$, you need to run the SGD several times (say, **30 times**).
- Each time the SGD is run on a **fresh** set of n training examples (that is, in total you need to generate $30n$ training examples).
- In each run, the SGD outputs a (possibly different) predictor vector $\widehat{\mathbf{w}}$. For each output predictor $\widehat{\mathbf{w}}$, you need to evaluate the risk and classification error incurred by that predictor using the test set of $N = 400$ examples that is held out separately from the training set, that is, compute the average logistic loss and the average binary classification error incurred by $\widehat{\mathbf{w}}$ on those N test examples. (You do **not** need to generate a new test set every time you run the SGD. There should be only **one** test set for **each** (scenario, σ) pair; i.e., you will need 4 test sets for all your experiments in this project.)
- Hence, for each value of n, σ, you end up with a set of 30 estimates for the risk (and another set of 30 estimates for the binary classification error) corresponding to 30 (possibly different) output predictors.
- For the 30 **risk** estimates: compute their **mean** (i.e., average), their **minimum**, and their **standard deviation**. Here, the standard deviation is the average deviation of the 30 estimates around their mean. Calculate the difference between the **mean** and the **minimum**. That should be your estimate for the **expected excess risk** for the particular setting of n, σ being considered (i.e., this is a single point on the expected excess risk plot corresponding to a particular setting of σ). Use your estimate for the **standard deviation** to add an error bar on your plot.
- For the 30 **binary classification error** estimates: Compute their **mean** and their **standard deviation**. The mean represents your estimate for the expected binary classification error for the considered values of n, σ. Show your estimate for the **standard deviation** as an error bar on your plot.

Comment on your results. Explain whether or not they agree with the theoretical results we derived in class. Compare your results in Scenarios 1 and 2. Is there any difference? If so, can you justify it? For each scenario, compare between your results for each setting of σ (the standard deviation of the Gaussian distribution). Do you spot any difference? If so, can you justify it?

Project: Stochastic Gradient Descent

TODO: Add author(s)

1 Introduction

TODO: Describe the overall goals of this report
TODO: Outline your algorithm for stochastic gradient descent

2 Experiments

TODO: Describe the experiments that you ran, including the combinations of parameters that you chose for SGD. Also, describe how you generated your training and test datasets in each scenario **including a clear description of how the projection step is done in each of the two scenarios.**

3 Analysis of ρ-Lipschitz properties

TODO: Answer the question about ρ-Lipschitz properties for each scenario
TODO: If you use non-constant learning rates, describe how you selected them (for each scenario)

4 Results

TODO: Fill in the following table of results. (Note: here, n is the training set size, and N is the test set size. Excess risk should be calculated as "mean − min".)

					Logistic loss				Classification error	
Scenario	σ	n	N	# trials	Mean	Std Dev	Min	Excess Risk	Mean	Std Dev
1	0.1	50	400	30	TODO	TODO	TODO	TODO	TODO	TODO
1	0.1	100	400	30	TODO	TODO	TODO	TODO	TODO	TODO
1	0.1	500	400	30	TODO	TODO	TODO	TODO	TODO	TODO
1	0.1	1000	400	30	TODO	TODO	TODO	TODO	TODO	TODO
1	0.35	50	400	30	TODO	TODO	TODO	TODO	TODO	TODO
1	0.35	100	400	30	TODO	TODO	TODO	TODO	TODO	TODO
1	0.35	500	400	30	TODO	TODO	TODO	TODO	TODO	TODO
1	0.35	1000	400	30	TODO	TODO	TODO	TODO	TODO	TODO
2	0.1	50	400	30	TODO	TODO	TODO	TODO	TODO	TODO
2	0.1	100	400	30	TODO	TODO	TODO	TODO	TODO	TODO
2	0.1	500	400	30	TODO	TODO	TODO	TODO	TODO	TODO
2	0.1	1000	400	30	TODO	TODO	TODO	TODO	TODO	TODO
2	0.35	50	400	30	TODO	TODO	TODO	TODO	TODO	TODO
2	0.35	100	400	30	TODO	TODO	TODO	TODO	TODO	TODO
2	0.35	500	400	30	TODO	TODO	TODO	TODO	TODO	TODO
2	0.35	1000	400	30	TODO	TODO	TODO	TODO	TODO	TODO

TODO: Include figures of results

5 Conclusion

TODO: Comment on your results. Explain whether or not they agree with the theoretical results we derived in class. Compare your results in Scenarios 1 and 2. Is there any difference? If so, can you justify it? For each scenario, compare between your results for each setting of σ (the standard deviation of the Gaussian distribution). Do you spot any difference? If so, can you justify it?

A Appendix: Symbol Listing

TODO: Provide a listing of the symbols used in your analysis, along with a brief description of their meaning. If your code uses a different variable name corresponding to a particular symbol, mention it here (for example, your analysis might use N to refer to the test-set size, while your code uses `testSetSize`.)

B Appendix: Library Routines

TODO: If you used any packages or built-in libraries in your code (for example, to perform linear algebra operations), please briefly describe the functions/subroutines you used here. This is merely to help us understand your code, so you only need to describe library routines which are not trivial and whose functionality is not obvious from the name.

C Appendix: Code

TODO: Include your code below. Please make sure your code is sufficiently well-documented for us to understand what it is doing.

```
#### Insert your code here ####
```