CSE 250: Machine Learning Theory Spring 2016

Homework 3 (Project)

Instructor: Raef Bassily **Due on:** Wed June 1

Please, read the following instructions carefully

- The homework 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 resport 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 symbol index listing all the symbols used for the parameters and variables in your code and what they denote.
  - A section on the design of the experiments where you explain clearly and concisely in your own words 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 resuls of your expreiments, where you state and discuss your results, and include all the relevant figures.
  - A conclusion where you state the main findings and lessons learned.
  - An appendix that contains a well-documented copy of your code.
- **Do not forget to include your answers to the explicit questions in this assignment** either within the sections above that you find relevant, or, separately in another section.
- You can use any language you prefer, e.g., MATLAB, Python, C, etc.
- You are required to submit a well-documented copy of your code. That is, the main steps in your code must be preceded with a brief comment describing the line/segment of code to follow.
- 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 (and state in your comment) the setting of each of the input parameters and the format of the output to be expected. **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 (e.g., circle, diamond, etc.) for plots (i.e., curves) that share the same figure. It is recommended that you use figure legends to describe different plots on the same figure. If you will eventually print your final submission in black-and-white, 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:

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

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 $L(\mathbf{w}; D) \triangleq \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim D}[\ell_{\mathsf{logist}}(\mathbf{w}, (\mathbf{x}, y))]$ denote the risk incurred by a predictor $\mathbf{w} \in \mathcal{C}$ under the logistic loss model w.r.t. the distribution is D. Let $\mathbf{err}(\mathbf{w}; D) \triangleq \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim D}[\mathbf{1}(\text{sign}(\langle \mathbf{w}, (\mathbf{x}, 1)\rangle) \neq y)]$ denote the binary classification error (the risk under '0-1' loss) incurred by $\mathbf{w}$ w.r.t. the distribution is 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 $\sigma \in \{0.05,\ 0.25\}$, it is required to plot**

- **an estimate of the expected risk** of the SGD learner, namely, $\mathop{\mathbb{E}}_{\hat{\mathbf{w}}}[L(\hat{\mathbf{w}}; D)]$ where $\hat{\mathbf{w}}$ is the output predictor of the SGD given n training examples,
- **an estimate of the expected classification error** of the SGD learner, namely, $\mathop{\mathbb{E}}_{\hat{\mathbf{w}}}[\mathbf{err}(\hat{\mathbf{w}}; D)]$

against 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 risk (and each estimate for the expected classification error), you need to provide an estimate for $\sqrt{\mathbf{Var}_{\hat{\mathbf{w}}}[L(\hat{\mathbf{w}}; D)]}$ (and $\sqrt{\mathbf{Var}_{\hat{\mathbf{w}}}[\mathbf{err}(\hat{\mathbf{w}}; D)]}$) 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 the SGD:

For each setting of n, in order to obtain an estimate for the **expected** performance of the output predictor $\hat{\mathbf{w}}$, you need to run the SGD several times (say, **20 times**). Each time the SGD is run on a **fresh** set of n training examples (that is, in total you need to generate $20n$ training examples). In each run, the SGD outputs a (possibly different) predictor vector $\hat{\mathbf{w}}$. For each output predictor $\hat{\mathbf{w}}$, 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 $\hat{\mathbf{w}}$ 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 **all** your experiments.)

Hence, you end up with a set of 20 estimates for the risk (and another set of 20 estimates for the binary classification error) corresponding to 20 (possibly different) output predictors. Then, you proceed by computing the **average** and the **standard deviation** for each set. The average of the first set represents your estimate for the expected risk for the particular setting of n, σ being considered (i.e., this is a single point on the expected risk plot corresponding to a particular setting of σ), and the average of the second set represents your estimate for the expected binary classification error for those n, σ.

The standard deviation of each set reflects the average deviation of the 20 estimates around their mean. **In your plots, it is also required that you show the standard deviation for each point on the plot using error bars.**

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?

Hint: In your experiments, you can either set the step size of the SGD (i.e., the learning rate) as discussed in class, or use a varying step size, in particular, $\alpha_t = \frac{M}{\rho\sqrt{t}},\ t = 1, \ldots, T$. But, you must clearly state which setting you are using.