Notes: Léon Bottou's SGD Paper

You are responsible for Sections 1-3 of Bottou's paper *Large-Scale Machine Learning with Stochastic Gradient Descent.*

Section 3.2, specialized for the logistic loss

Recall that in logistic regression, we use the logistic loss

$$Q(z) = \log(1 + \exp(-z)), \tag{1}$$

where $z = \hat{y}y = \mathbf{x}^T\mathbf{w}y$.

1. What is the gradient of the logistic loss (i.e. what is $\nabla_{\mathbf{w}} Q(z)$)? Also, what is the shape of the result and what is the runtime of computing it?

2. What is the hessian of the logistic loss (i.e. what is $\nabla^2_{\mathbf{w}} Q(z)$)? Also, what is the shape of the result and what is the runtime of computing it?

Table 2 from the paper is reproduced below. Recall that all stated values are implicitly in big-O notation, and that the stated run times explicitly ignore dependencies on the number of dimensions d and the cost of computing the Q function. Use the information from the previous page to make the run times below more precise by including these dependencies.

	GD	2GD	SGD	2SGD
Time per iteration:	n	n	1	1
Iterations to accuracy ρ:	$\log \frac{1}{\rho}$	$\log \log \frac{1}{\rho}$	$\frac{1}{\rho}$	$\frac{1}{\rho}$
Time to accuracy ρ:	$n \log \frac{1}{\rho}$	$n \log \log \frac{1}{\rho}$	$\frac{1}{\rho}$	$\frac{1}{\rho}$
Time to excess error $\mathcal{E}$:	$\frac{1}{\epsilon^{1/\alpha}} \log^2 \frac{1}{\epsilon}$	$\frac{1}{\epsilon^{1/\alpha}} \log \frac{1}{\epsilon} \log \log \frac{1}{\epsilon}$	$\frac{1}{\epsilon}$	$\frac{1}{\epsilon}$

Problems

Problem 1. You are training a logistic regression model using the polynomial feature map with a very high degree and a small number of data points.

1. Which optimization algorithm do you choose and why?

2. Does VC theory predict good or bad statistical performance?

Problem 2. You are training a logistic regression model. Your original dataset had a large number of features and few data points, so you applied the PCA feature map to reduce the dimensions.

1. Why does VC theory predict that applying the PCA feature map was a good idea?

2. Which optimization algorithm do you choose and why?

3. How does applying the PCA feature map influence the runtime of the optimization?

4. If the PCA feature map had poor empirical risk, which other feature map(s) might you try?

Problem 3. You are working at a large social media company, and your task is to use logistic regression to predict when a user will click on an ad. Your dataset is very large. The company has billions of users, and each user has thousands of interactions with ads, and so the number of data points you have is $N > 10^{12}$. But the number of feature dimension is relatively small, with $d = 100$.

1. Your boss suggests reducing the size of the dataset and using gradient descent to solve the problem. Use VC theory to explain the resulting effect on the excess error of your problem.

2. Instead of using gradient descent on a sampled dataset, you could use stochastic gradient descent on the original dataset. When would this be a good idea?

3. Your company offers a profit sharing bonus. Whenever an employee discovers an algorithm for increasing ad revenue, the employee receives 10% of the resulting increased revenue over the next quarter. Last quarter's ad revenue was 1 billion dollars. Therefore increasing performance by only 0.1% will result in the company making 1 million dollars more and a personal bonus of $100,000.

   Use VC theory to come up with a strategy to increase revenue. How will your choice of optimization algorithm change for your new strategy?