CSE Ph.D. Qualifying Exam, Spring 2023

Data Analysis

Please answer three of the following four questions. All questions are graded on a scale of 10. If you answer all four, all answers will be graded and the three lowest scores will be used in computing your total. Show all your work and write in a readable way.

1. **Logistic Regression with Sparse Features**

   In many real-world scenarios our data has millions of dimensions, but a given example has only hundreds of non-zero features. For example, in document analysis with word counts for features, our dictionary may have millions of words, but a given document has only hundreds of unique words. In this question we will make l_2 regularized SGD efficient when our input data is sparse. Recall that in l_2 regularized logistic regression, we want to maximize the following objective (in this problem we have excluded w_0 for simplicity):

   $$F(\mathbf{w}) = \frac{1}{N}\sum_{j=1}^{N} l\left(\mathbf{x}^{(j)}, y^{(j)}, \mathbf{w}\right) - \frac{\lambda}{2}\sum_{i=1}^{d} \mathbf{w}_i^2$$

   where $l\left(\mathbf{x}^{(j)}, y^{(j)}, \mathbf{w}\right)$ is the logistic objective function

   $$l\left(\mathbf{x}^{(j)}, y^{(j)}, \mathbf{w}\right) = y^{(j)}\left(\sum_{i=1}^{d} \mathbf{w}_i \mathbf{x}_i^{(j)}\right) - \ln\left(1 + \exp\left(\sum_{i=1}^{d} \mathbf{w}_i \mathbf{x}_i^{(j)}\right)\right)$$

   and the remaining sum is our regularization penalty. When we do stochastic gradient descent on point $\left(\mathbf{x}^{(j)}, y^{(j)}\right)$, we are approximating the objective function as

   $$F(\mathbf{w}) \approx l\left(\mathbf{x}^{(j)}, y^{(j)}, \mathbf{w}\right) - \frac{\lambda}{2}\sum_{i=1}^{d} \mathbf{w}_i^2$$

   Definition of sparsity: Assume that our input data has d features, i.e. $\mathbf{x}^{(j)} \in \mathbb{R}^d$. In this problem, we will consider the scenario where $\mathbf{x}^{(j)}$ is sparse. Formally, let s be average number of nonzero elements in each example. We say the data is sparse when $s << d$. In the following questions, your answer should take the sparsity of $\mathbf{x}^{(j)}$ into consideration when possible. Note: When we use a sparse data structure, we can iterate over the non-zero elements in $O(s)$ time, whereas a dense data structure requires $O(d)$ time.

   a. [1 point] Let us first consider the case when $\lambda = 0$. Write down the SGD update rule for $\mathbf{w}_i$ when $\lambda = 0$, using step size η, given the example $\left(\mathbf{x}^{(j)}, y^{(j)}\right)$.

   b. [2 points] If we use a dense data structure, what is the average time complexity to update $\mathbf{w}_i$ when $\lambda = 0$? What if we use a sparse data structure? Justify your answer in one or two sentences.

c. [1 point] Now let us consider the general case when $\lambda > 0$. Write down the SGD update rule for $\mathbf{w}_i$ when $\lambda > 0$, using step size η, given the example $\left(\mathbf{x}^{(j)}, y^{(j)}\right)$.

d. [1 point] If we use a dense data structure, what is the average time complexity to update $\mathbf{w}_i$ when $\lambda > 0$?

e. [2 points] Let $\mathbf{w}_i^{(t)}$ be the weight vector after t-th update. Now imagine that we perform k SGD updates on w using examples $\left(\mathbf{x}^{(t+1)}, y^{(t+1)}\right), \cdots, \left(\mathbf{x}^{(t+k)}, y^{(t+k)}\right)$, where $\mathbf{x}_i^{(j)} = 0$ for every example in the sequence. (i.e. the i-th feature is zero for all of the examples in the sequence). Express the new weight, $\mathbf{w}_i^{(t+k)}$ in terms of $\mathbf{w}_i^{(t)}, k, \eta$, and λ.

f. [3 points] Using your answer in the previous part, come up with an efficient algorithm for regularized SGD when we use a sparse data structure. What is the average time complexity per example?

2. **Mixture Discriminant** We consider a multi-class classification problem where we predict one out of K classes based on d real-valued features. We use the probabilistic model given by

$$P\left(X|Y=k\right) = \sum_{r=1}^{R_k} \pi_{kr} \phi\left(X; \mu_{kr}, \mathbf{C}\right) \tag{1}$$

Here, the function $\phi(x; \mu, \mathbf{C})$ denotes a Gaussian density with mean μ and covariance matrix $\mathbf{C}$, evaluated in x. Thus, the class conditional for each class k is given by a Gaussian mixture with R_k mixture components, weights given by $\pi_{k:}$, means given by $\mu_{k:}$, and covariance matrix $\mathbf{C}$.

a. [2 pts] Use Bayes rule to derive the class-posterior probabilities as

$$P\left(Y=k|X=x\right) = \frac{\sum_{r=1}^{R_k} \pi_{kr} \phi\left(X; \mu_{kr}, \mathbf{C}\right) \Pi_k}{\sum_{l=1}^{K} \sum_{r=1}^{R_l} \pi_{lr} \phi(X; \mu_{lr}, \mathbf{C}) \Pi_l}, \tag{2}$$

with the $\{\pi_k\}_{1 \le k \le K}$ denoting the prior on the relative abundance of the different classes.

b. [2 pts] Formulate an expectation-maximization algorithm for computing the maximum likelihood estimator of (1), given training data $X, Y \in \mathbb{R}^{N \times d} \times \{1, \ldots K\}^N$.

c. [2 pts] Consider instead the following model, prescribed through it's joint distribution

$$P(X, Y) = \sum_{r=1}^{R} \pi_r P_r(Y) \phi(X; \mu_r, \mathbf{C}). \tag{3}$$

Derive the posterior class distribution as

$$P\left(Y=k|X=x\right) = \frac{\sum_{r=1}^{R} \pi_r P_r\left(Y=k\right) \phi\left(x; \mu_r, \mathbf{C}\right)}{\sum_{r=1}^{R} \pi_r P_r\left(Y=k\right)} \tag{4}$$

d. [2 pts] Derive the class conditional $P(X|Y)$ for (3) and show that the associated model is a generalization of (1).

e. [2 pts] Derive the EM algorithm for (3).

3. **Support Vector Machine**

The soft margin SVM is formulated as

$$\begin{aligned} \min_{w\in\mathbb{R}^n, b\in\mathbb{R}, s\in\mathbb{R}^m} \quad & \frac{1}{2}w^\top w + C\mathbf{1}^\top s \\ s.t. \quad & X^\top w + by + s - \mathbf{1} \geq \mathbf{0}, \\ & s \geq \mathbf{0}. \end{aligned}$$

(1) Let $\{x_i, y_i\}_{i=1}^m$ with $x_i \in \mathbb{R}^n$ and $y_i \in \{\pm 1\}$, $i \in [1:m]$, be a training dataset. For a fixed value of C, let the corresponding SVM classifier have parameters w^*, b^*.

(a) Let $h \in \mathbb{R}^n$ and $Q \in \mathcal{O}_n$ (Q is an $n \times n$ matrix), and form the second training set: $\{Q(x_i - h), y_i\}_{i=1}^m$. Show that the SVM classifier for this second dataset using the same value of C has parameters $Qw^*, {w^*}^\top h + b^*$.

(b) If we first center the training examples, how does this change the SVM classifier?

(2) Suppose that instead of using $C\sum_{i=1}^m s_i$ as the penalty term in the objective of the primal SVM problem we use the quadratic penalty $\frac{1}{2}C\sum_{i=1}^m s_i^2$, while maintaining the constraint $s_i \geq 0$.
(a) Formulate the new primal problem in vector form. When is the primal problem feasible?
(b) Does strong duality hold for this problem? Justify your answer.
(c) Write down the KKT conditions.
(d) Find the dual problem.

4. **KL divergence**

In many machine learning problems, we often need to measure the "distance" between two probability distributions, such as in the E-step of EM algorithms and variational inference. The Kullback-Leibler (KL) divergence is such a statistical distance, and it measures how one probability distribution differs from another. In this problem, we consider discrete probability distributions, i.e.

$$\mathcal{P} = \left\{ (p_1, \ldots, p_n) | \sum_i^n p_i = 1, p_i \geq 0 \right\}.$$

Now for two probability distributions $p, q \in \mathcal{P}$, their KL divergence is defined as

$$KL(p\|q) = -\sum_{i=1}^n p_i \log \frac{q_i}{p_i}.$$

(1) [2pts] Prove that KL divergence is non-negative, i.e., $KL(p\|q) \geq 0$, and $KL(p\|q) = 0$ if and only if $p = q$.

(2) [1pt] Is KL divergence symmetric or asymmetric (i.e., is it true that $KL(p\|q) = KL(q\|p)$)? Justify your answer.

(3) [3pts] Consider two random variables X and Y that follow probability distributions p_X and p_Y, respectively, and joint distribution p_{XY}. If X and Y are independent, we have $p_{X,Y} = p_X p_Y$. If not, we may be interested in quantifying the degree of their independence. One way to measure this is to consider $KL(p_{X,Y}\|p_X p_Y)$, which is also known as the *mutual information* between X and Y, denoted as $I(X, Y)$. Prove that

$$I(X, Y) = H(X) - H(X|Y),$$

where $H(X) := -\sum_x p_X(x) \log p_X(x)$ is the *entropy* of X, measuring the uncertainty of X, and $H(X|Y) := -\sum_{x,y} p_{X,Y}(x, y) \log p_{X|Y}(x|y)$ is the *conditional entropy* of X given Y.

(4) [4pts] Now let us consider a toy machine learning task in generative modeling, where we have a dataset that follows a bimodal distribution $p(X)$, as illustrated in Figure 1. Our goal is to approximate the real distribution $p(X)$ with a model distribution $q_\theta(X)$. For simplicity, we restrict the $q_\theta(X)$ to normal distributions, i.e., $q_\theta(X) = \mathcal{N}(\mu, \sigma^2)$. Since we want to approximate $p(X)$ using $q_\theta(X)$, a natural objective function is to minimize the KL divergence between these two probability distributions. Here, we have two options for the objective, i.e., $\min_\theta KL(p\|q_\theta)$ or $\min_\theta KL(q_\theta\|p)$. For each of the two choices, draw the density curve of q_θ that could be obtained by minimizing the corresponding KL divergence, and explain your answers.

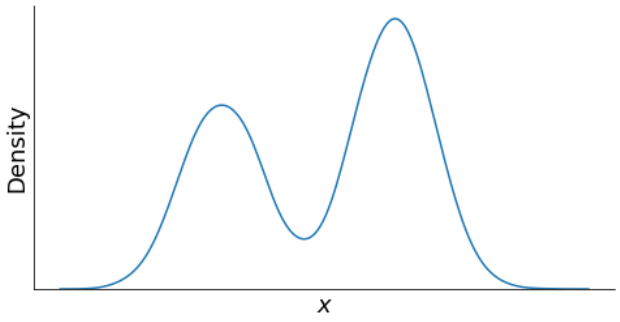

Figure 1: Probability density of $p(X)$.