CAP6610 Machine Learning, Spring 2022

Midterm 2

4/5/2022

1. (10%) *Train vs test datasets.* Suppose you are building a classifier that identifies cats and dogs. You have a dataset of 3,000 images containing cats, dogs, or other objects (neither cat nor dog). You randomly split the data into a 2,500 image training set and a 500 image test set.

   (a) Why is it important to "reserve" some images for the test dataset? (Why shouldn't we use all 3,000 images to train the classifier?)

   (b) After training your classifier for a while, you observe it performs well on the training images, but poorly on the test images. What is one possible explanation?

2. (20%) *Elementary properties of the quadratic regularized logistic classification.* Consider

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad \frac{1}{n}\sum_{i=1}^{n}\log(1+\exp(-y_i\boldsymbol{\phi}_i^\top\boldsymbol{\theta}))+\lambda\|\boldsymbol{\theta}\|^2, \tag{1}$$

   for $y_i = \pm 1$. Answer the following true/false questions:

   (a) Problem (1) has multiple locally optimal solutions.

   (b) Let $\boldsymbol{\theta}^\star$ be an optimal solution for (1), $\boldsymbol{\theta}^\star$ is sparse (has many zero entries).

   (c) If the training data is linearly separable, then some coefficients θ_j might become infinite if $\lambda = 0$.

   (d) At optimum, the empirical risk always increases as we increase λ.

   (e) On a test set, the prediction accuracy always increases as we increase λ.

3. (30%) *Learning algorithms for L_1 regularized Huber regression.* Consider the L_1-norm regularized Huber regression problem

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad \frac{1}{n}\sum_{i=1}^{n}\ell(\boldsymbol{\phi}_i^\top\boldsymbol{\theta}-y_i)+\lambda\|\boldsymbol{\theta}\|_1, \tag{2}$$

   where the loss function is the Huber loss

$$\ell(\varepsilon)=\begin{cases}\varepsilon^2/2 & |\varepsilon|\le a\\ a(|\varepsilon|-a) & |\varepsilon|>a\end{cases}$$

   with some constant a. We assume the empirical risk part $(1/n)\sum_{i=1}^{n}\ell(\boldsymbol{\phi}_i^\top\boldsymbol{\theta}-y_i)$ is *strongly convex*.

   (a) Is the Huber loss differentiable? If yes, write down the gradient of the loss of one sample $\ell(\boldsymbol{\phi}_i^\top\boldsymbol{\theta}-y_i)$; if not, find a subgradient of it at any $\boldsymbol{\theta}$;

   (b) Give the pseudo-code of the proximal (sub)gradient method with constant step size γ for solving (2). Your answer should not contain any abstract operation such as "the proximal operator of the L_1 norm". What is the convergence rate of this algorithm?

(c) Give the pseudo-code of the stochastic proximal (sub)gradient method with constant step size γ for solving (2). Again, your answer should not contain any abstract operation such as "the proximal operator of the L_1 norm". What is the expected convergence rate of this algorithm?

Hint. The convergence rate of either algorithm is one of the following: superlinear, linear, $(1/t)$-sublinear, or $(1/\sqrt{t})$-sublinear.

4. (20%) *An important inequality to prove convergence of the proximal gradient descent algorithm.* Consider a regularized empirical risk minimization problem

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad L(\boldsymbol{\theta}) + \lambda r(\boldsymbol{\theta}).$$

We try to solve this optimization problem using proximal gradient descent (PGD): assuming $L(\boldsymbol{\theta})$ is differentiable, PGD iteratively updates the variable as

$$\boldsymbol{\theta}^{(t+1)} = \text{Prox}_{\gamma^{(t)}\lambda r}(\boldsymbol{\theta}^{(t)} - \gamma^{(t)}\nabla L(\boldsymbol{\theta}^{(t)})).$$

To show that it converges to global minimum, a key inequality given in `lec6.pdf` page 38 is

$$L(\boldsymbol{\theta}^{(t+1)}) + \lambda r(\boldsymbol{\theta}^{(t+1)}) \leq L(\boldsymbol{\theta}) + \lambda r(\boldsymbol{\theta}) + \frac{1}{\gamma^{(t)}}(\boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}^{(t+1)})^\top(\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}) + \frac{M}{2}\|\boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}^{(t+1)}\|^2$$

for any $\boldsymbol{\theta}$. In this question you should prove this inequality using the following assumptions:

- Both L and r are convex. Since L is assumed to be differentiable, we have

  $$L(\boldsymbol{\theta}) \geq L(\boldsymbol{\theta}^{(t)}) + \nabla L(\boldsymbol{\theta}^{(t)})^\top(\boldsymbol{\theta} - \boldsymbol{\theta}^{(t)})$$

  for all $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^{(t)}$; r is not necessarily differentiable, but there exists a similar inequality using a subgradient.
- The gradients of L are Lipschitz continuous with parameter M, which means

  $$L(\boldsymbol{\theta}) \leq L(\boldsymbol{\theta}^{(t)}) + \nabla L(\boldsymbol{\theta}^{(t)})^\top(\boldsymbol{\theta} - \boldsymbol{\theta}^{(t)}) + \frac{M}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}^{(t)}\|^2$$

  for all $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^{(t)}$.

5. (10%) *Clustering with pre-assigned vectors.* Suppose that some of the vectors in $\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_n$ are assigned to specific groups. For example, we might insist that $\boldsymbol{\phi}_{27}$ be assigned to cluster 5. Suggest a simple modification of the k-means algorithm that respects this requirement. Describe a practical example where this might arise, when each vector represents m features of a medical patient.

6. (10%) *Implementation of PCA.* In MATLAB you are given two functions to compute the singular value decomposition of a matrix, `svd` and `svds`. Explain which one is the preferred function to use in the following scenarios:

   (a) The data matrix $\boldsymbol{\Phi}$ is 100×300 and dense, and we want to embed each column as a 10-dimensional vector;

   (b) The data matrix $\boldsymbol{\Phi}$ is $10^5 \times 10^6$ and sparse (with approximately 10^7 nonzeros), and we want to embed each column as a 100-dimension vector.

   In each case, you are given the data matrix `Phi`. You should describe in detail how to use either functions, possibly with a few additional steps, to obtain the embedding matrix $\boldsymbol{Y}$ and the projection matrix $\boldsymbol{\Theta}$ that satisfies $\boldsymbol{\Theta}^\top\boldsymbol{\Theta} = \boldsymbol{I}$.