CSE 547 - Assignment 1

Philip Pham

May 16, 2018

Problem 0

List of collaborators: I have not collaborated with anyone.

List of acknowledgements: None.

Certify that you have read the instructions: I have read and understood these policies.

Problem 1: Gaussian Random Projections and Inner Products

Let $\phi(x) = \frac{1}{\sqrt{m}}Ax$ represent our random projection of $x \in \mathbb{R}^d$, with A an $m \times d$ projection matrix with each entry sampled i.i.d from $N(0,1)$. Note that each row of A is a random projection vector, $v^{(i)}$.

The *norm preservation theorem* states that for all $x \in \mathbb{R}^d$, the norm of the random projection $\phi(x)$ approximately maintains the norm of the original x with high probability:

$$\mathbb{P}\left((1-\epsilon)\|x\|^2 \leq \|\phi(x)\|^2 \leq (1+\epsilon)\|x\|^2\right) \geq 1 - 2\exp\left(-\left(\epsilon^2 - \epsilon^3\right)m/4\right), \tag{1}$$

where $\epsilon \in (0, 1/2)$.

Using the norm preservation theorem, prove that for any $u, v \in \mathbb{R}^d$ such that $\|u\| \leq 1$ and $\|v\| \leq 1$,

$$\mathbb{P}\left(|u \cdot v - \phi(u) \cdot \phi(v)| \geq \epsilon\right) \leq 4\exp\left(-\left(\epsilon^2 - \epsilon^3\right)m/4\right) \tag{2}$$

Solution

Proof. First note that

$$(1-\epsilon)\|u+v\|^2 \leq \|\phi(u+v)\|^2 \leq (1+\epsilon)\|u+v\|^2$$

implies that

$$\|u+v\|^2 - 2\epsilon \leq \|\phi(u+v)\|^2 \leq \|u+v\|^2 + 2\epsilon \tag{3}$$

by triangle inequality and the assumption of the norms of u and v.

Thus, the probability of the event in Equation 3 than that of Equation 1.

Using this and taking the additive inverse, we have that

$$\mathbb{P}\left(\|\phi(u+v)\|^2 \not\in \left[\|u+v\|^2 - 2\epsilon, \|u+v\|^2 + 2\epsilon\right]\right) \leq 2\exp\left(-\left(\epsilon^2 - \epsilon^3\right)m/4\right) \tag{4}$$

$$\mathbb{P}\left(\|\phi(u-v)\|^2 \not\in \left[\|u-v\|^2 - 2\epsilon, \|u-v\|^2 + 2\epsilon\right]\right) \leq 2\exp\left(-\left(\epsilon^2 - \epsilon^3\right)m/4\right). \tag{5}$$

By the countable sub-additivity property of probability distributions, we have that the probability of both these events occurring is at most $4\exp\left(-\left(\epsilon^2-\epsilon^3\right)m/4\right)$. Thus, we are done if we can show $\{|u \cdot v - \phi(u) \cdot \phi(v)| \geq \epsilon\}$ subsets these two conditions.

If we have the pair

$$\|\phi(u+v)\|^2 \leq \|u+v\|^2 - 2\epsilon \Rightarrow \|u+v\|^2 - \|\phi(u+v)\|^2 \geq 2\epsilon \tag{6}$$

$$\|\phi(u-v)\|^2 \geq \|u-v\|^2 + 2\epsilon \Rightarrow \|u-v\|^2 - \|\phi(u-v)\|^2 \geq 2\epsilon, \tag{7}$$

we can use the linearity of ϕ and the expansion $\|u \pm v\|^2 = \|u\|^2 + \|v\|^2 \pm 2u \cdot v$, we can add the two inequalities to obtain

$$4\left(u \cdot v - \phi\left(u\right) \cdot \phi\left(v\right)\right) \geq 4\epsilon.$$

Thus, we have that the conditions in Equations 4 and 5 imply $u \cdot v - \phi\left(u\right) \cdot \phi\left(v\right) \geq \epsilon$.

Similarly, we show that the pair

$$\|\phi(u+v)\|^2 \geq \|u+v\|^2 + 2\epsilon \tag{8}$$

$$\|\phi(u-v)\|^2 \leq \|u-v\|^2 - 2\epsilon \tag{9}$$

implies $u \cdot v - \phi\left(u\right) \cdot \phi\left(v\right) \leq -\epsilon$, which gives us $|u \cdot v - \phi\left(u\right) \cdot \phi\left(v\right)| \geq \epsilon$. $\square$

Problem 2: Locality-Sensitive Hashing (LSH) for Angle Similarity

Suppose our set of n points $D = \{p_1, \ldots, p_n\}$ are vectors in d dimensions. Our problem is: given a query point q find a point $p \in D$, which has a small angle with q. Recall that the angle between two vectors a and b is $\cos^{-1}\left(\frac{a \cdot b}{\|a\|\|b\|}\right)$.

As doing this exactly may be computationally expensive, let us try to do this approximately with a fast algorithm. The approximate objective is as follows: suppose there exists a point $p \in D$ which has angle less than θ with p, then our goal is return a point with angle less than $c\theta$, where $c > 1$.

Let us try to do this with LSH. Let us consider the a family of hash functions, where $h(p) = \operatorname{sign}\left(u \cdot p\right)$ where we will sample u uniformly at random from a Gaussian (or from a unit sphere).

1. Provide an exact expression for $\mathbb{P}\left(h\left(p\right) = h\left(p'\right)\right)$ based on some geometric relation between p and p'.

Solution

Define

$$\operatorname{angle}\left(u, v\right) = \cos^{-1}\left(\frac{u \cdot v}{\|u\|\|v\|}\right), \tag{10}$$

which is the angle between two vectors.

Then,

$$\boxed{\mathbb{P}\left(h\left(p\right) = h\left(p'\right)\right) = 1 - \frac{\operatorname{angle}\left(p, p'\right)}{\pi}.} \tag{11}$$

2. Provide an expression for P_1 and P_2 in terms of θ and $c\theta$. Note that since we want a small angle, we should use:
   (a) If $\text{angle}\left(p, p'\right) < \theta$, then $\mathbb{P}\left(h\left(p\right) = h\left(p'\right)\right) \geq P_1$.
   (b) If $\text{angle}\left(p, p'\right) > c\theta$, then $\mathbb{P}\left(h\left(p\right) = h\left(p'\right)\right) \leq P_2$.

   Solution

   If $\text{angle}\left(p, p'\right) < \theta$, then

   $$\mathbb{P}\left(h\left(p\right) = h\left(p'\right)\right) = 1 - \frac{\text{angle}\left(p, p'\right)}{\pi} \geq 1 - \frac{\theta}{\pi},$$

   so $\boxed{P_1 = 1 - \frac{\theta}{\pi}.}$

   If $\text{angle}\left(p, p'\right) > c\theta$, then

   $$\mathbb{P}\left(h\left(p\right) = h\left(p'\right)\right) = 1 - \frac{\text{angle}\left(p, p'\right)}{\pi} \leq 1 - \frac{c\theta}{\pi},$$

   so $\boxed{P_2 = 1 - \frac{c\theta}{\pi}.}$

3. Provide expressions for query time for point q, the space to store the hash tables, and the construction time of our datastructure.

   Solution

   Suppose we have L hash functions. If we use the algorithm discussed in class, to query a point, we need to compute L hashes. Then, up to 3 times, we iterate through the buckets: for each bucket, we choose a point and check how close it is to q; if it is $c\theta$ close, we stop. The worst case is that we decide there exists no point that is θ close to q. In this case, we iterate through the L buckets 3 times, so the time complexity is $O(L)$.

   For the space needed to store the hash tables, we need to store L bits for each point, so the space needed is $O(nL)$

   For construction, we need to compute L hashes for each point, so the computational complexity is $O(nL)$ as well.

Problem 3: Dual Coordinate Ascent

Consider the problem

$$\min_{w} L\left(x\right), \text{ where } L\left(x\right) = \sum_{i=1}^{n} \left(w \cdot x_i - y_i\right)^2 + \lambda \left\|w\right\|^2. \tag{12}$$

1. Show that the solution for Equation 12 is obtained for weights

$$w^* = \left(X^\intercal X + \lambda I\right)^{-1} X^\intercal Y \tag{13}$$

$$= \frac{1}{\lambda} X^\intercal \alpha^*, \tag{14}$$

where $\alpha^* = (I + XX^\intercal/\lambda)^{-1}$.

Solution

Proof. We can take the derivative of L in Equation 12 directly. Note that $D\left(x \mapsto Ax\right)(x) = A$ and $D\left(x \mapsto x^\intercal x\right)(x) = 2x^\intercal$. Therefore by the chain rule,

$$D\left(x \mapsto (Ax)^\intercal (Ax)\right)(x) = 2x^\intercal A^\intercal A. \tag{15}$$

We can reformulate Equation 12 as a function of w

$$\begin{aligned} l_{X,y}(w) &= (Aw - y)^\intercal (Aw - y) + \lambda w^\intercal w \\ &= (Aw)^\intercal (Aw) - 2y^\intercal Aw + y^\intercal y + \lambda w^\intercal w. \end{aligned} \tag{16}$$

Taking the derivative, we have that

$$D\left(l_{X,y}\right)(w) = 2w^\intercal X^\intercal X - 2y^\intercal X + 2\lambda w^\intercal. \tag{17}$$

Setting Equation 17 to 0 and solving for w, we have

$$\begin{aligned} 0 &= 2w^\intercal X^\intercal X - 2y^\intercal X + 2w^\intercal \\ w^\intercal \left(X^\intercal X + \lambda I\right) &= y^\intercal X \\ \left(X^\intercal X + \lambda I\right) w &= X^\intercal y \\ w &= \left(X^\intercal X + \lambda I\right)^{-1} X^\intercal y. \end{aligned}$$

Since Equation 16 is a quadractic form, the problem is convex, and

$$w^* = \left(X^\intercal X + \lambda I\right)^{-1} X^\intercal Y$$

minimizes Equation 12.

Now, note that

$$\left(X^\intercal X + \lambda I\right) X^\intercal = X^\intercal X X^\intercal + \lambda X^\intercal = X^\intercal \left(XX^\intercal + \lambda I\right).$$

Multiplying on the left by $\left(X^\intercal X + \lambda I\right)^{-1}$ and on the right by $\left(XX^\intercal + \lambda I\right)^{-1}$, we have that

$$X^\intercal \left(XX^\intercal + \lambda I\right)^{-1} = \left(X^\intercal X + \lambda I\right)^{-1} X^\intercal.$$

Substituting this into Equation 13, we obtain

$$\begin{aligned} w^* &= X^\intercal \left(XX^\intercal + \lambda I\right)^{-1} y \\ &= X^\intercal \left(\lambda \left(I + \frac{XX^\intercal}{\lambda}\right)\right)^{-1} y \\ &= \frac{1}{\lambda} X^\intercal \left(I + \frac{XX^\intercal}{\lambda}\right)^{-1} y, \end{aligned}$$

which gives us the desired result. $\square$

If $\lambda = 0$, in general, this is not true since $XX^\intercal + \lambda I$ may not be invertable when $n > d$. However, if $d \geq n$, and $\operatorname{rank}(X) \geq n$, Equation 14 may still be well-defined.

2. Define

$$G(\alpha_1, \alpha_2, \ldots, \alpha_n) = \frac{1}{2}\alpha^\intercal \left(I + XX^\intercal/\lambda\right)\alpha - Y^\intercal\alpha. \tag{18}$$

Start with $\alpha = 0$. Choose coordinate i randomly, and update

$$\alpha_i = \arg\min_z G(\alpha_1, \ldots, \alpha_{i-1}, z, \alpha_{i+1}, \ldots, \alpha_n). \tag{19}$$

Show that the solution to the inner optimization problem for α_i is:

$$\alpha_i = \frac{y_i - \frac{1}{\lambda}\left(\sum_{j\neq i}\alpha_j x_j\right)\cdot x_i}{1 + \|x_i\|^2/\lambda}. \tag{20}$$

Solution

Proof. We can take the partial derivative of Equation 18 directly to obtain

$$\begin{aligned}\frac{\partial G}{\partial \alpha_i} &= \alpha_i + \frac{1}{\lambda}\left(\alpha^\intercal XX^\intercal\right)_i - y_i \\ &= \alpha_i + \frac{1}{\lambda}\left(\alpha_i \|x_i\|^2 + \sum_{j\neq i}\alpha_j (x_j \cdot x_i)\right) - y_i. \end{aligned} \tag{21}$$

Setting Equation 21 to 0, solving for α_i, and taking advantage of convexity, we find

$$\alpha_i = \frac{y_i - \left(\sum_{j\neq i}\alpha_j x_j\right)\cdot x_i}{1 + \|x_i\|^2/\lambda} \tag{22}$$

minimizes Equation 18 as a function of α_i, and solves the inner optimization problem. $\square$

3. What is the computational complexity of this update, as it is stated?

Solution

The complexity of updating α_i with Equation 20 is $O(nd)$ since we need to iterate over the n rows of X, and take the d-dimensional dot product of each row with x_i.

4. What is the computational complexity of one stochastic gradient descent update?

Solution

The complexity of one stochastic gradient descent update is $O(d)$. We computed the derivative in Equation 17 for the full matrix X. In stochastic gradient descent we'd replace X by a vector by randomly sampling a row from X. Then, to compute the gradient we have to do some dot products along with scalar operations.

5. Now consider the procedure.
   - Start with $\alpha = 0$, $w = \frac{1}{\lambda} X^\intercal \alpha = 0$.
   - Choose coordinate i randomly and perform the following update:
     - Compute the differences:
$$\Delta\alpha_i = \frac{(y_i - w \cdot x_i) - \alpha_i}{1 + \|x_i\|^2 / \lambda} \tag{23}$$
     - Update the parameters as follows:
$$\begin{aligned} \alpha_i &\leftarrow \alpha_i + \Delta\alpha_i \\ w &\leftarrow w + \frac{\Delta\alpha_i}{\lambda} x_i. \end{aligned} \tag{24}$$

   Prove that the update rule in Equation 24 is valid.

Solution

Proof. Let α' and w' be the result of updating coordinate i of α. Assume that $w = \frac{1}{\lambda} X^\intercal \alpha$. This is true when $\alpha = 0$. We will show that this invariant holds as α is updated.

To see that, the update rule for w is valid, we can rewrite
$$w = \frac{\alpha_1}{\lambda} x_1 + \cdots + \frac{\alpha_i}{\lambda} x_i + \cdots + \frac{\alpha_n}{\lambda} x_n, \tag{25}$$
so
$$\begin{aligned} w' &= w + \frac{\Delta\alpha_i}{\lambda} x_i \\ &= \frac{\alpha_1}{\lambda} x_1 + \cdots + \frac{\alpha_i + \Delta\alpha_i}{\lambda} x_i + \cdots + \frac{\alpha_n}{\lambda} x_n \\ &= \frac{\alpha_1}{\lambda} x_1 + \cdots + \frac{\alpha_i'}{\lambda} x_i + \cdots + \frac{\alpha_n}{\lambda} x_n \\ &= \frac{1}{\lambda} X^\intercal \alpha'. \end{aligned}$$
Thus, the w update is valid.

To see that the α update is valid, we show that Equations 20 and 24 are equivalent. Both algorithms initiate $\alpha = 0$, so they are equivalent at the initial step.

By using the definition $w = \frac{1}{\lambda} X^\intercal \alpha$,
$$\begin{aligned} \alpha_i' &= \alpha_i + \Delta\alpha_i \\ &= \frac{(y_i - w \cdot x_i) - \alpha_i}{1 + \|x_i\|^2 / \lambda} + \frac{\alpha_i + \alpha_i \|x_i\|^2 / \lambda}{1 + \|x_i\|^2 / \lambda} \\ &= \frac{1}{1 + \|x_i\|^2 / \lambda} \left(y_i - \frac{1}{\lambda} \left(\sum_{j \neq i} \alpha_j x_j \right) \cdot x_i - \frac{1}{\lambda} \alpha_i \|x_i\|^2 + \frac{1}{\lambda} \alpha_i \|x_i\|^2 \right) \\ &= \frac{y_i - \frac{1}{\lambda} \left(\sum_{j \neq i} \alpha_j x_j \right) \cdot x_i}{1 + \|x_i\|^2 / \lambda}, \end{aligned}$$
so both update rules are equivalent. $\square$

6. What is the computation complexity of the update defined by Equations 23 and 24?

Solution

The computation complexity is $O(d)$. Computing the dot product when computing $\Delta\alpha_i$ and updating w are both $O(d)$ operations. Everywhere else, we do scalar operations.

This is much faster than the $O(nd)$ update for Equation 20.

Problem 4: Project Milestone

Build a simple object dection model. Our object detector will comprise of a binary classifier per category: given features of an image patch corresponding to a bounding box, does this patch contain an object of the category of interest?

Solution