COMP 441 — Large Scale Machine Learning. 10th Feb, 2016

Assignment 2

Due By: 24nd Feb 2016 *Due By: 24nd Feb 2016*

This assignment is worth total 25 points. Your assignment is due by 12:00pm, 24th Feb, 2016 either by email or in class.

Some Logistics

1. You are encouraged to discuss but you should write your own solutions in detail.
2. Solve either the theory part (Section 1) or the programming part (Section 2).
3. If you do both theory and programming, the maximum of the two will be considered as your final score. You will get 10 bonus points if you solve both the parts perfectly.
4. You will get 10 bonus points if you implement the programming part with parallelization (multi-core, multi-node or GPUs) and show significant speedups.
5. You can only claim at max 10 bonus points

1 Theory Question (25 Point)

1.1 Problem 1 (20 Points): A Variant of Random Projections

Given two vectors $x,\ y \in \mathbb{R}^D$. We will study another variant of random projection. Define k i.i.d random vectors r^i, $i \in \{1,\ 2, ...,\ k\}$ such that each component j, $r^i_j\ \forall i$, is randomly (and independently) chosen as +1 or -1 with each probability $\frac{1}{2}$. We then generate x' and y' with component $i \in \{1,\ 2, ...,\ k\}$, given by $x'_i = {r^i}^T x$ and $y'_i = {r^i}^T y$. (x' and y' is k dimensional projected vector).

Prove the following

1. For any vector x we have $\frac{1}{k}\mathbb{E}(||x'||_2^2) = ||x||_2^2$ (Hint: It should also hold for $k = 1$)
2. $Var(\frac{1}{k}||x'||_2^2) \leq \frac{4}{k}||x||_2^4$ (Hint: $||x||_4 \leq ||x||_2$)
3. Find k such that $(1-\epsilon)||x-y||_2^2 \leq \frac{1}{k}||x'-y'||_2^2 \leq (1+\epsilon)||x-y||_2^2$, with probability$1-\delta$ (Hint: Show $k \geq \frac{2}{\epsilon^2\delta}$). (Even if you cannot prove 2 you can assume it for this part.)

1.2 Problem 2 (5 Points): Proving Convexity

Prove the following:

1. $-ax + x\log x$ is a convex function. (You can use any known property of convex functions to prove it)
2. Given that $f_i(x)$ is convex for all i, prove that the function $f(x) = \max_i f_i(x)$, (pointwise maximum of all functions) is convex. (Hint: A function is convex if and only if the set $\{(x,\mu) : \mu \geq f(x)\}$ is convex. You can also use the fact that intersection of convex sets is convex.)

1.3 Problem 3 (10 Bonus Points): Convergence

Given a convex function. If the function is M-Lipschitz and m-strongly convex then we have the following upper and lower (quadratic) bounds on the function f) for any $y \in Domain(f)$

$$f(x) + \Delta f(x)^T(x-y) + \frac{m}{2}||x-y||_2^2 \leq f(x) \leq f(x) + \Delta f(x)^T(x-y) + \frac{M}{2}||x-y||_2^2$$

Prove that if a function if both M-Lipschitz and m-strongly convex then the gradient descent step $x^{t+1} = x^t - \frac{1}{M}\Delta f(x^t)$ leads to

$$f(x^{t+1}) - f(x^*) = (1 - \frac{m}{M})(f(x^t) - f(x^*)),$$

where x^* is the minimizer of $f(x)$
(Hint: Show $f(x^{t+1}) \leq f(x^t) - \frac{1}{2M}||\Delta f(x^t)||_2^2$ and $f(x^*) \geq f(x^t) - \frac{1}{2m}||\Delta f(x^t)||_2^2$)
What can we say about the number of iterations T such that $f(x^T) - f(x^*) \leq \epsilon$

2 Programming Question (25 Point)

Implement the LSH algorithm and test it on WEBSPAM dataset (You can use other real datasets if you want). You can use the following guidelines, you are free to use other resources.

1. Partition the dataset into Train set (T) and query set (Q). (recommended use 90% for building tables and 10% for querying)
2. Choose parameters K and L. (It should be input to the code)
3. Generate $K \times L$ random vectors of dimension D (same as data dimension), r_i $i \in 1, 2, ..., K \times L$, where each component of r_i is i.i.d standard normal. Store these vectors. (You can be more smart and simply store random seeds)
4. The j^{th} LSH hash function we be (Signed Random Projections) defined as $h_j(x) = sign(r_j^T x)$ (one bit), were r_j is a j^{th} vector. (Make sure the indexing j is consistent).
5. Now initialize L different hash tables (or hashmaps).
6. For every data point x in T, its key in the i^{th} hash table will be given by K-bits $F_i(x) = [h_{(i-1)*K+1}(x); h_{(i-1)*K+2}(x); h_{(i-1)*K+3}(x); ....; h_{i*K}(x)]$, here ; is the concatenations . (Make sure your indexing is consistent across all data vectors including the query.).
7. Take all data points $x \in T$ and insert it into all the L hashmaps (key to list) with the corresponding key. (L hashmap inserts here)
8. For query q, report the union of elements from L hashtables retrieved using $F_i(q)$ for $i \in \{1, 2, ..., L\}$.. (L hashmap queries, with q, followed by union)

Try to make your implementation as memory efficient and light as possible.
Bonus 10 points: Exploit opportunities of parallelism and show significant speedup using any of the following (multi-core, multi-node, GPUs).