COMS W6998-5: Algorithms through Geometric Lens (Fall'17) Oct 2, 2017

Lecture 8 – Fast Linear Regression, NNS / ANN

Instructors: *Alex Andoni, Ilya Razenshteyn* Scribes: *Max Aalto*

1 Introduction

Today's lecture covers the remainder of fast linear regression and an introduction to nearest neighbor search and approximate nearest neighbors. It uses principles developed in earlier classes, particularly the oblivious subspace embedding property and fast Johnson-Lindenstrauss.

2 Fast Linear Regression

2.1 Review from last class

Last class we introduced the problem of linear regression, minimizing x in $||Ax - b||_2^2$. We hope to apply a linear transformation S to this problem such that the solution to our new problem is approximately the solution to the old problem.

From last class, we know that a random matrix $S \in \mathbb{R}^{s \times n}$ is an oblivious subspace embedding if $\forall U \subset \mathbb{R}^n$ with dimension $U = d$, we have the property:

$$Pr(\forall y \in U, ||Sy||_2^2 \in (1 \pm \epsilon)||y||_2^2) \geq 0.9$$

Theorem 1. *For $s = O(\frac{d \log(\frac{1}{\epsilon})}{\epsilon^2})$, a Gaussian, we have $S \in \mathbb{R}^{s \times n}$ is O.S.E. (oblivious subspace embedding).*

We also defined an epsilon net on the unit sphere in last class, a discretization procedure. If $\mathbb{S}^U = \{y \in U : ||y||_2 = 1\}$, an epsilon net on $\mathbb{S}^U$ is a subset N_ϵ whose points have epsilon-neighborhoods that completely cover the sphere.

Claim 2. $\exists N_\epsilon \subset \mathbb{S}^U : |N_\epsilon| \leq (\frac{10}{\epsilon})^d$

This was proved in the last class.

Claim 3. *If $s = \Theta(\frac{d \log(\frac{1}{\epsilon})}{\epsilon^2})$, then $Pr(\forall y \in N_\epsilon, ||Sy||_2^2 \in (1 \pm \epsilon)||y||_2^2) \geq 0.9$*

This was also proved in last class, and is the last statement from the previous class that was reviewed in today's lecture.

2.2 Utilizing the epsilon net

We want to show that the properties of points in the epsilon net, given above, allow us to show that this property holds for any point in our subspace. How can we be sure that $||Sy||_2^2$ does not diverge?

Claim 4. $Pr(\forall y \in N_\epsilon, ||Sy||_2^2 \in (1 \pm \epsilon)||y||_2^2) \geq 0.9 \implies Pr(\forall y \in U, ||Sy||_2^2 \in (1 \pm \epsilon)||y||_2^2) \geq 0.9$

Proof. Using the linearity of U, it suffices to this is true for $U = \mathbb{S}^U$ (ie the unit sphere in dim d). We take v_0, the closest point from N_ϵ to our fixed $y \in \mathbb{S}^U$ and use it to decompose y into an infinite sum of points in N_ϵ, ie:

$$y = v_0 + v_1 + v_2 + \dots$$

Where the following properties are clearly true:

$$\text{(a) } ||v_i||_2 = 1$$

$$\text{(b) } v_i = \alpha_i u_i, u_i \in N_\epsilon$$

$$\text{(c) } |\alpha_i| \leq \epsilon^i$$

To calculate the components of this decomposition (and prove that such a decomposition exists), we proceed as follows: take $y - v_0$ and normalize it, ie $\frac{y-v_0}{||y-v_0||}$. Then take u_1 to be the closest point to this vector from N_ϵ, and let $v_1 = ||y - v_0||u_1$. This process generalizes iteratively to v_n.

Since for each v_i we have $Pr(\forall y \in N_\epsilon, ||Sy||_2^2 \in (1 \pm \epsilon)||y||_2^2) \geq 0.9$, it remains to show that:

$$||Sy||_2^2 \in (1 \pm O(\epsilon))||y||_2^2$$

By the triangle inequality, we can see that $||Sy||_2 \leq \sum_{i=0}^{\infty} ||Sv_i||_2 \leq (1+\epsilon)+\epsilon(1+\epsilon)+\epsilon^2(1+\epsilon)+\dots \leq 1 + O(\epsilon)$, by property (c) above. This serves as an upper bound of $||Sy||_2$. By the triangle inequality we also have $||Sy||_2 \geq ||Sv_0||_2 - \sum_{i=1}^{\infty} ||Sv_i||_2 \geq (1-\epsilon) - \epsilon(1+\epsilon) - \epsilon^2(1+\epsilon) - \dots \geq 1 - O(\epsilon)$, which is our lower bound. Thus:

$$||Sy||_2^2 \in (1 \pm \epsilon)||y||_2^2$$

Which completes the proof.

□

In summary: we take the unit sphere in dimension d, we discretize it into an epsilon-net, then by union bound we have the O.S.E. property for all points in the epsilon-net, and then we decompose arbitrary points on the sphere into a linear combination of points from the epsilon-net.

2.3 The fast linear regression algorithm

Now that we have shown the necessary steps for fast linear regression, we perform the operations algorithmically. We will use the naive method of matrix multiplication and matrix-vector multiplication for $||SAx - Sb||_2^2$ for this case.

Since SA has size $\frac{d}{\epsilon^2}\log(\frac{1}{\epsilon}) \times d$, the time to compute SAx is polynomial in d and $\frac{1}{\epsilon}$, which is good. However, how much time does it take to reduce dimension, ie calculate SA?

The dimensions of these matrices are $\frac{d}{\epsilon^2}\log(\frac{1}{\epsilon} \times n \times d \approx nd^2\frac{\log(\frac{1}{\epsilon})}{\epsilon^2}$, which is very bad (it's worse than without performing dimensionality reduction).

To optimize this, instead of taking S from a Gaussian distribution we use the same method as in fast Johnson-Lindenstrauss. Recall that in fast J-L we have three matrices:

$$\Pi HD$$

Where Π is a subsampling matrix, H is a Hadamard matrix, and D is a random diagonal matrix whose entries are randomly sampled from $\{0, 1\}$. Using this matrix product, we worsen our bound on dimension slightly:

Theorem 5. *In this case,* $s = O(\frac{d^2\log^2(\frac{1}{\epsilon})}{\epsilon^2}) \implies \Pi HD$ *is O.S.E.*

Note the exponents on d and $\log(\frac{1}{\epsilon})$. The new runtime, however, for $S = \Pi HD \cdot A$ is $dn\log n$ (follows from proofs from previous class on fast J-L).

Theorem 6 (Sarlos, 2006)**.** *For the least squares problem* $||Ax - b||_2^2 \to$ *min, we can find* $(1 \pm \epsilon)$*-approximate solution in time* $O(nd\log(n) +$ *poly*$(\frac{d}{\epsilon}))$*, and this bound is almost tight.*

2.4 Fast linear regression where A is sparse

Theorem 7 (Clarkson-Woodruff, 2013)**.** *For a sparse matrix A, we can find an approximation solution in time* $O(n[nz(A)] + poly(\frac{d}{\epsilon}))$*, where* $nz(A)$ *is the number of non-zero entries in A*

Proof. Proving this theorem relies on proving that the hashing tree / count sketch matrix has the OSE property, which was not done in class.

□

3 Nearest Neighbor and Approximate Nearest Neighbor

3.1 Similarity Search

This is a problem that involves a dataset of n objects, a query that returns an object, and a goal that involves finding 1 or more of the most similar data items.

3.2 Nearest Neighbor Search

From the previous problem, we have a similar but different problem involving a dataset of n points in dim d feature space, ie $X \subset \mathbb{R}^d, |X| = n$, a query $q \in \mathbb{R}^d$, and a goal to return the "most similar" $p \in X$ to q.

What does most similar mean? In the context of a matrix space, we can say that it is the point with the smallest distance to q.

3.2.1 Metric Spaces

Generally, a metric space is a set X equipped with a metric $d : X \times X \to \mathbb{R}^+$, where:

$$(1)\ d(x, y) = 0 \text{ iff } x = y$$

$$(2)\ d(x, y) = d(y, x)$$

$$(3)\ d(x, z) \leq d(x, y) + d(y, z)$$

3.2.2 4 cases of this problem

For a proper understanding of the nearest neighbors problem, we will consider four different metric spaces:

$$(1)\ ||x - y||_1, l_1 \text{ norm}$$

$$(2)\ ||\cdot||_1 : X, q \subset 0, 1^d \text{ (ie Hamming distance)}$$

$$(3)\ ||\cdot||_2, l_2 \text{ norm}$$

$$(4)\ ||\cdot||_2 : X, q \in \mathbb{S}^{d-1} \text{ (NNS cosine similarity)}$$

We have three main parameters for the search:

(1) Query time, best case O(d) or sublinear in n

(2) Space, ideally O(nd)

(3) Preprocessing time

However, we must confront an issue known as the "Curse of Dimensionality", which states that (1) all the known data structures with o(n) query time require $2^{\Omega(d)}$ space, and (2) if $\exists$ NNS data structure with (a) $d^{o(n)}$ query time and (b) preprocessing time is polynomial in d and n, then this implies that the strong exponential time hypothesis is false.

3.3 Approximate Nearest Neighbor Search (ANN)

Given these issues, we define a third problem, with the same dataset and query but with the goal of returning a $\hat{p} \in X$ such that $||q - \hat{p}|| \leq C \cdot \ \min_{p^\star \in X} ||q - p^\star||$.

In other words, we have a $C > 1$-approximation. Is this a meaningful approximation?

Note that the curse of dimensionality is less of a problem for nicely distributed data sets it holds for NNS in less interesting cases, but we will confront the issue regardless.

Theorem 8. *$\exists$ ANN data structure for $(\{0, 1\}^d, ||\cdot||_1)$ with space $d^{O(n)}, n^{1+\frac{1}{C}}$ and query time $d^{O(1)}, n^{\frac{1}{c}}$*

3.4 Approximate Near Neighbor Search (ANN)

Note that we are looking for a near neighbor now, not the nearest. This is approximately equivalent in Hamming space. We use the same dataset, but our query is now a $q \in \mathbb{R}^d$ such that $\exists p^\star \in X$ with $d(q, p^\star) \leq r$ (this property is 'hard-wired in the data structure').

We want a point $\hat{p} \in X$ such that $d(q, \hat{p}) \leq Cr$. In this event we use a similar algorithm to binary search in the data structure, where we divide the structure in half and if our result is too low, we take the upper half, and if it is too high, we take the lower half until our goal is reached.

It can be shown that the probability of success is $1 - \frac{1}{10^d}$, which we did not show in class today but will show in the following class.