COMS 4774 Unsupervised Learning Fall 2024
Problem Set #4

Brian Chen - bc2924@columbia.edu

December 19, 2024

Paper 1

Optimal rates for k-NN density and mode estimation. by Dasgupta and Kpotufe.

Definition 1. (kNN density estimate). For every $x \in \mathbb{R}^d$, let $r_k(x)$ be the distance from x to its k-th nearest neighbor in a sample X drawn iid from some continuous distribution F. Then the density estimate is

$$f_k(x) = \frac{k}{n \cdot v_d \cdot r_k(x)^d}$$

where v_d is the volume of the d-dimensional unit sphere.

Theorem 1. *Letting $\delta > 0$, if f has a single mode x_0 and satisfies assumptions 1 and 2, then $\exists N_{x_0,\delta}$ s.t. if k satisfies*

$$\left(\frac{24C_{\delta,n}f(x_0)}{\check{C}_{x_0}r_{x_0}^2}\right)^2 \leq k \leq \left(\frac{1}{2}\sqrt{\frac{C_{\delta,n}}{\hat{C}_{x_0}}}\right)^{4d/(4+d)} f(x_0)^{(2d+4)/(4+d)} \left(\frac{v_d}{4}n\right)^{4/(4+d)}$$

then with probability at least $1 - 2\delta$,

$$\|x - x_0\| \leq 5\sqrt{\frac{C_\delta}{C_{x_0}}f(x_0)} \cdot \frac{1}{k^{1/4}}$$

Theorem 2. *Assuming f satisfies the earlier assumoptions, let $k = \Omega(C_{\delta,n}^2$. Then $\exists N = N(x_0, \epsilon(n))$ s.t. for $n \geq N$, and k satisfying*

$$\left(\frac{24C_{\delta,n}f(x_0)}{\check{C}_{x_0}\min\left\{r_{x_0}^2/4, (r/\alpha)^2\right\}}\right)^2 \leq k \leq \left(\frac{1}{2}\sqrt{\frac{C_{\delta,n}}{\hat{C}_{x_0}}}\right)^{4d/(4+d)} \lambda_{x_0}^{(2d+4)/(4+d)} \left(\frac{v_d}{4}n\right)^{4/(4+d)}$$

then for M_n being the modes from the procedure defined in fig. 3, then with probability at least $1 - 2\delta$ $\exists x \in M_n$ s.t.

$$\|x - x_0\| \leq 5\sqrt{\frac{C_\delta}{C_{x_0}} f(x_0)} \cdot \frac{1}{k^{1/4}}$$

Theorem 3. *Let $\Lambda = \sup f(x)$ and $r(\epsilon) = \sup_x \max r'(\epsilon, x), r''(\epsilon, x)$. If f satisfies assumption 2 and $r(\epsilon) = \Omega(k/n)^{1/d}$ (and in particular if f is Holder continuous, then we let $\epsilon = \Omega(k/n)^{\beta/d}$. Defining*

$$\lambda_0 = \max\left\{2\tilde{\epsilon}, 8\frac{\Lambda}{k}C_{\delta,n}^2, \left(\frac{k}{n} + C_{\delta,n}\frac{\sqrt{k}}{n}\right)\frac{2}{v_d r(\tilde{\epsilon})^d}\right\}$$

then with probability at least $1 - \delta$ we can pick any $\lambda \geq 2\lambda_0$ and let $\lambda_f = \inf f(x)$ s.t. all estimated modes in $M_n \cap X_n^\lambda$ can be assigned to distinct modes in $M \cap X^{\lambda_f}$

Paper 2

Do GANs learn the distribution? Some theory and empirics by Arora, Risteski, Zhang. The paper more formally investigates the question of whether generative adversarial nets can actually learn the underlying target distribution. Earlier work by Goodfellow et. al. indicate that very deep nets with sufficient training data and computation time. Later work indicated that there are limits to this ability in the context of a generated distribution with very low support, e.g. if there is a "mode collapse". The paper proposes a measure of the size of the output distribution's support based on the birthday paradox (e.g. the probability of "collisions") and also provide theoretical results indicating that encoder decoder based GAN architectures are also unable to resolve the "mode collapse" issue.

Theorem 4. *Given a discrete probability distribution P on set Ω, if there exists a subset $S \subseteq \Omega$ of size N such that $\sum_{s \in S} P(s) \geq \rho$, then the probability of encountering at least one collision among M iid samples from P is $\geq 1 - \exp\left\{-\frac{(M^2-M)\rho}{2N}\right\}$*

Proof. Note that in the setting of the birthday paradox, if we draw from a distribution of support N, then we expect that a batch of size $\sqrt{N}$ is likely (with probability $> 50\%$) that there is a collision. This indicates that we can estimate the size of the support of a distribution in the opposite direction, namely that if we find that a sampling of batch images from the GAN of size s has some duplicates with good probability, then we can estimate that the size of the support of the underlying distribution is approximately s^2.

The probability that there is a collision within a set S among M samples is explicitly

$$\Pr[\text{collision}] \geq 1 - \Pr[\text{no collisions}]$$

$$\geq 1 - (1 - \frac{\rho}{N}) \times (1 - \frac{2\rho}{N}) \times \cdots \times (1 - \frac{(M-1)\rho}{N})$$

$$\geq 1 - \exp\left\{-\frac{(M^2 - M)\rho}{2N}\right\}$$

since every additional sample "type" must be dissimilar from all of the previously seen sample types. In the worst case situation where the probability mass is uniformly distributed, we get the exponential bound.

□

Theorem 5. *Given a discrete probability distribution P on a set Ω, if the probability of encountering at least one collision among M iid samples from P is γ, then for $\rho = 1 - o(1)$ there exists a subset $S \subset \Omega$ s.t. $\sum_{s \in S} P(s) \geq \rho$ with size $\leq \frac{2M\rho^2}{(-3+\sqrt{9+24/M \ln \frac{1}{1-\gamma}})-2M(1-\rho)^2}$*

Proof. This implies that if there are consistently collisions seen in some batches, then there is some component of the overall distribution that does have limited support, but its distribution is hard to distinguish from the full distribution with limited samples.

Assuming that X_n are iid samples from an underlying discrete distribution P, let $T = \inf\{t \geq 2, X_t \in \{X_i\}\}$ be the "collision time" and $\beta = \frac{1}{\Pr[T=2]}$ be a measure of the "uniformity" of P. Then we can upper bound $\Pr[T \geq M]$ for large β with

$$\Pr[T \geq M] \geq \exp\left(-\frac{M^2}{2\beta} - \frac{M^3}{6\beta^2}\right)$$

where we can estimate β with the fact that

$$\Pr[T \geq M] = 1 - \gamma \geq \exp\left(-\frac{M^2}{2\beta} - \frac{M^3}{6\beta^2}\right)$$

by definition and therefore

$$\beta \leq \frac{2M}{-3 + \sqrt{9 + \frac{24}{M}\ln\frac{1}{1-\gamma}}} = \beta^*$$

The largest possible size of the distribution that satisfies this inequality is found by letting

$$\frac{1}{(\frac{\rho}{N}})^2 N + (1-\rho)^2 \leq \beta^*$$

from which we can plug this in to the previous expression to obtain a largest possible bound

$$N \leq \frac{2M\rho^2}{(-3 + \sqrt{9 + \frac{24}{M}\ln\frac{1}{1-\gamma}}) - 2M(1-\rho)^2}$$

□

Finally, the paper proves a result of the limitations of encoder decoder GAN models.

Theorem 6. *There exists a generator G of support $\frac{p\Delta^2 \log^2(p\Delta L L_\phi/\epsilon)}{\epsilon^2}$ an an encoder with at most d nonzero weights such that for all L-Lipshitz discriminators D with capacity less than p,*

$$|E_{x\sim\mu}\phi(D(x, E(x))) - E_{z\sim\nu}\phi(D(G(z), z))| \leq \epsilon$$

Proof. Call a set T of samples non colliding if they do not lie in the same block. Let T_{nc} be the distribution over non colliding sets where each set is sampled iid from the distribution corresponding to the i-th block. We aim to show that we can construct an encoder/coder model probabilistically that can succeed with high probability. First, it can be shown that the expected encoder matches the expectation of $\phi(D, (x, E(x)))$ by showing that $E_G E_{z\sim\nu}\phi(G(z), z)$ concentrates around the expectation as a function of the randomness in G. Lemma D.1 shows that we can calculate the expectation of $\phi(D(G(z), z))$ as

$$\mathbb{E}_{z\sim\nu}\phi(D(G(z), z)) = \mathbb{E}_{T\sim T_{nc}}\mathbb{E}_{z\sim T}\phi(D(G(z), z))$$

so we only need to calculate the random variable $\mathbb{E}_{z\sim T}\phi(D(G(z),z))$. We can use Mcdiarmid's inequality to then get the concentration result in terms of T and G. Finally, use Markov's inequality to show that

$$|E_{x\sim\mu}\phi(D(x,E(x)))-E_{z\sim\nu}\phi(D(G(z),z))|\leq\epsilon$$

is small for almost all non-colliding sets, from which the theorem result follows immediately.

□

Paper 3

Approximate Nearest Neighbors Towards Removing the Curse of Dimensionality by Indyk and Motwani.

Given the nearest neighbor problem, where in some metric space X we are given a set of n points such that we can find the nearest neighbor to a query point $q \in X$, we are interested in solving this problem approximately in faster than brute force time. This has obvious applications in, for instance, nearest neighbor based classification algorithms, where querying points and brute force searching can be very costly. The paper presents two algorithmic results that improve known bounds on the approximate nearest neighbor algorithm. First, that preprocessing can be done in poolynomial time in n and d, and second that you can query a point in sublinear time (polynomial in $\log n$ and d). These algorithms are based on the idea of locality-sensitive hashing.

The first set of algorhtms is based on a reduction of the ϵ-NN problem to the problem of point location in equal balls.

Definition 1. Point location in Equal Balls (PLEB). Given n radius r balls centered at $C = \{c_1, \ldots, c_n\}$ in $M = (X, d)$, devise a data structure which any query point $q \in X$ returns YES if there is a $c_i \in C$ such that $q \in B(c_i, r)$, otherwise NO.

Definition 2. ϵ-point location in equal balls (ϵ-PLEB). Defined analogously to the above, except that if for the c_i closest to q, $r \leq d(q, c_i) \leq ((1+\epsilon)r)$ then return either yes or no.

Note that ϵ-PLEB problem reduces trivially to ϵ-NN, since we can just make each of the centers a point in the data set. We need to prove that we can reduce it in the other way as well, e.g. ϵ-NN solves ϵ-PLEB with only small overhead in preprocessing and query cost.

Theorem 1. *For any $P, 0 < \alpha < 1, \beta > 1$, either P has an (α, α, β) ring separator, or P contains a $(1/2\beta, \alpha)$ cluster of size $\geq (1 - 2\alpha)|P|$*

Assume that P does not have an (α, α, β) separator. Define

$$f_p^\infty = |P - B(p, \beta r)|$$

and

$$f_p^0(r) = |P \cap B(p, r)|$$

Note that f_p^∞ is monotonically decreasing, $f_p^0(r)$ is monotonically increasing, and $f_p^\infty(0) = n$, $f_p^\infty(\infty) = 0$, $f_p^0 = 0$, and $f_p^0(\infty) = n$. Then there must be some r such that $f_p^\infty(r_p) = f_p^0(r_p)$ by intermediate value theorem. This value must be some $\alpha < 1/2$ scale of n, e.g. this value of r_p is such that $f_p^\infty(r_p) = f_p^0(r_p) \leq \alpha n$. Let q be some point such that it minimizes r_q. For $S = P \cap R(q, r_q, \beta r_q)$, simple algebra yields $|S| \geq (1 - 2\alpha)n$. This also implies that $\Delta(S) \leq 2\beta r_q$ by triangle inequality. Therefore, for all $s \in S$,

$$|P \cap B(s, r_q)| \leq |P \cap B(s, r_s)| \leq \alpha n$$

Theorem 2. *Let S be a (γ, δ) cluster for P. Then for any B, there is an algorithm, which produces a sequence of sets $A_1, \ldots, A_k \subset P$ constituting a $(b, \delta, \frac{\gamma}{(1+\gamma)\log_b n})$ cover for S.*

Proof. This is accomplished by the following algorithm definition. The correctness is justified afterwards. The algorithm is correct in terms of finding finding the cover for S. The outer loop is guaranteed to terminate, so $S \subset A = \cup_j A_j$. The condition that $\forall j \in \{1, \ldots, k\}$ and any $p \in S$, $\left|P \cap \cup_{q \in A_j} B(p, r)\right| \leq b|A_j|$ must be true since the inner loop terminates. Finally, since the inner loop occurs a max of $\log_b n$ times, and S is a γ, δ-cluster, then $|B(p_j, \gamma\Delta(S)) \cap P| \leq \delta|P|$ and therefore $|A_j| \leq \delta|P|$, indicating that we have indeed found a cover for S from the earlier definitions of a cover. □

Algorithm 1: Cover Algorithm

```
Data: $S = P \cap R(q, r_q, \beta r_q)$
Result: $r \leftarrow \frac{\gamma\Delta(S)}{\log_b n}$, $j \leftarrow 0$
repeat
    $j \leftarrow j + 1$, choose some $p_j \in S$, $B_j^1 \leftarrow \{p_j\}$;
    $i \leftarrow 1$;
    while $|P \cap \bigcup_{q \in B_j^i} B(q, r)| > b|B_j^i|$ do
        $B_j^{i+1} \leftarrow P \cap \bigcup_{q \in B_j^i} B(q, r)$;
        $i \leftarrow i + 1$;
    end
    $A_j \leftarrow B_j^i$, $S \leftarrow S - A_j$, $P \leftarrow P - A_j$;
until;
$S = \phi$, $k \leftarrow j$
```

Theorem 3. *The ring cover tree can be constructed in deterministic $O(n^2)$ time.*

Proof. We can construct the tree as follows. Determine whether a node for each level is a ring or a cover, and then compute either the ring or the cover. We can compute f_p^∞ and f_p^0 for each $p \in P$. Then we can construct this list L_p containing all points in P sorted in order, which takes $O(n^2 \log n)$ time. If we find a ring, then we are done and we have our ring. Otherwise, we find a cluster and apply the earlier algorithm to find the cover. This only takes $O(1)$ additional time, so the overall time is $O(n^2)$ if we are given the lists themselves. □

Then some analysis on ring cover tree construction is given in the context of the algorithm "Procedure Search" given on page 8.

Lemma 1. *Procedure Search(q, P) produces an ϵ-nearest neighbor for q in P.*

Proof. Two cases. Either P is a ring node or a cover node. If it's a ring node, then for any $s \in P - S_1$, then $d(s, p) \leq d(s, q) + d(q, p)$, and since $s \notin S_1$, then $d(s, p) \geq \beta r = 2(1 + 1/\epsilon)r$ and $d(p, q) \leq r(1 + 1/\epsilon)$. Combining the worst case distances yields $d(s, q) \geq (1 + 1/\epsilon)r \geq d(q, p)$ so we've found the approximate nearest neighbor. Similar logic extends if P is a cover node, thus indicating that procedure search produces the epsilon nearest neighbor regardless of P. □

Lemma 2. *A ring cover tree requires at most* $O(knb^{\log_{1/2\alpha} n}(1+2(1-2\alpha))^{\log_n}$ *space, not counting any non data storage.*

Proof. The proof is done by upper bounding the space requirement for rings or cover nodes. This takes the form

$$S(n) \leq \max_l \max_A [\sum_{i=1}^{l} S(b|A_i|] + S(n - |A|) + |A|bk$$

for cover nodes and

$$S \leq 2S(\frac{n}{2}(1+2(1-2\alpha))) + 1$$

for ring nodes. □

Theorem 4. *If there is an* (r_1, r_2, p_1, p_2) *sensitive family* H *for* D*, then there exists an algorithm for* $(r_1, r_2) - PLEB$ *under measure* D *which uses* $O(dn + n^{1+\rho})$ *space and* $O(n^\rho)$ *evaluations of the hash function for each query, for* $\rho = \frac{\ln 1/p_1}{\ln 1/p_2}$

Proof. This proof was done in an earlier paper on LSH for which I wrote proof sketches for. The gits of the proof is to show that there is an expected number of collisions for fixed g for the query point q, and an expected number of collisions for any g with query point q is at most l. Then by Markov inequality it can be shown that the probability that both properties for x-sensitive hashing hold is $> 1/2$. This implies that the probability that the hashing bucket is the same is bounded from below with

$$p_1^k = p_1^{\log_{1/p_2} n} = n^{-\rho}$$

□

Proposition 1. *For* $S = H^d$ *and* $D(p,q)$ *be the Hamming metric for* $p, q \in H$*, then for any* $r, \epsilon > 0$*, the family* $H = \{h_i : h_i((b_1, \ldots, b_d)) = b_i, i = 1, \ldots, n\}$ *is* $(r, r(1+\epsilon), 1 - \frac{r}{d}, 1 - \frac{r(1+\epsilon)}{d})$ *sensitive.*

Proof. □

Problem 4