COMS E6998-9: Algorithms for Massive Data (Fall'25) Oct 13, 2025

Lecture 12: VKDE, Attention, and HNSW

Instructor: *Alex Andoni* Scribes: *Soham Samal*

1 Review of KDE

1.1 Definition

Given a dataset $P \subset \mathbb{R}^d$ and kernel $k(x, y) = e^{-\|x-y\|^2}$, the KDE at a query point q is defined as:

$$\text{KDE}(q) = \sum_{i=1}^{n} k(p_i, q).$$

Each term measures how close p_i is to q, and the sum gives the estimated density around q.

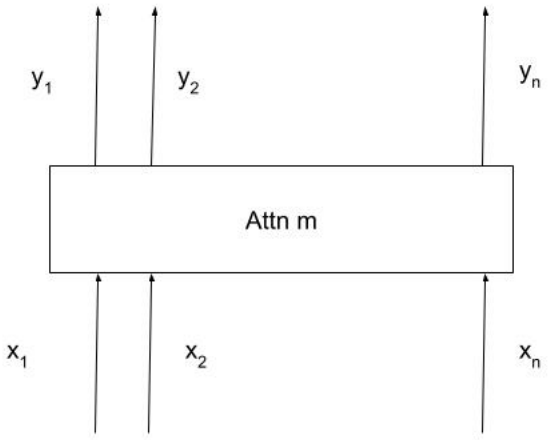
2 Solving Attention via VKDE

Definition 1. *We preprocess the dataset as* $P = \{(p_i, v_i)\} \in \mathbb{R}^d \times \mathbb{R}^{d_v}$, *and define*

$$\text{VKDE}(q) = \sum_{i=1}^{n} k(p_i, q)\, v_i.$$

VKDE averages vectors v_i weighted by their similarity to q, generalizing KDE to vector-valued data.

2.1 Attention Mechanism



$x_1, x_2, ..., x_n$ corresponds to the tokens of size n = context length

Definition 2. *Given pairs* $(q_i, k_i, v_i) \in \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^{d_v}$*, define*

$$A_{ij} = exp[QK^T]_{ij} = e^{q_i \cdot k_j}, \quad D_i = \sum_j A_{ij} \triangleq d_i.$$

Then the attention output is:

$$Y = D^{-1}AV, \quad where\ Y_i = \sum_j \frac{e^{q_i \cdot k_j}}{d_i} v_j.$$

$D^{-1}A$ is row normalized, where each row is essentially a probability distribution. We can then prove this is a normalized value-KDE where the kernel is the exponential dot-product.
Algorithm for attention via VKDE:

$$Y_i = \frac{1}{d_i} \sum_j e^{2q_i k_j} \cdot v_j = \frac{1}{d_i} \sum_j e^{\|q_i\|^2 + \|k_j\|^2 - \|q_i - k_j\|^2} v_j =$$

Suppose $\|q_i\| = \|k_j\| = \alpha$. This is an appropriate assumption because we can always rescale and/or group by weight of q_i's and k_j's.

$$\frac{e^{2\alpha^2}}{d_i} \sum_j e^{-\|q_i - k_j\|^2} v_j = \frac{e^{2\alpha^2}}{d_i} VKDE(q_i)$$

The exponential dot-product kernel is proportional to the Gaussian kernel $e^{-\|q_i - k_j\|^2}$. Thus, attention performs a Gaussian-weighted average over the values, up to a constant normalization factor. The normalization term d_i also corresponds to a VKDE instance over the same kernel but with all values equal to one (scalar).

2.2 Hyper Attention paper

Speedups computation of attention by designing an algorithm to find $S \in \mathbb{R}^{m \times n}$, a $n \times n$ diagonal matrix D, such that

$$\|Y - D^{-1}AS^TSV\|_{op} \leq \varepsilon \|D^{-1}A\|_{op} \|V\|_{op}.$$

The matrix S acts as a sketching operator that reduces computational complexity while preserving the operator norm up to a small error ε. The goal is to approximate the full attention computation efficiently without sacrificing much accuracy.

2.3 Approximating VKDE using LSH

Fact 3. *We can build an LSH hash function* $h : \mathbb{R}^d \to V$ *such that:*

$$\Pr[h(p) = h(q)] \approx e^{-\|p-q\|^2} = k(p, q).$$

VKDE can be approximated *in expectation* via hashing:

- Build hash table H with hash function h.
- For each bucket b: $H[b] = \sum_{j:h(p_j)=b} v_j$.

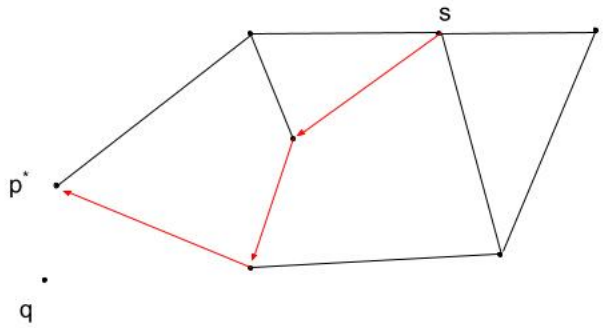
Claim 4. $\mathbb{E}[out] = VKDE(q)$

Proof: $\mathbb{E}[H[h(q)]] = \mathbb{E}[\sum_j \mathbb{1}[h(p_j) = h(q)]v_j] = \sum_j e^{-\|p_j - q\|^2 \cdot v_j}$

The expectation over random hash functions recovers the VKDE result, since the indicator function $\mathbb{1}[h(p_j) = h(q)]$ has expected value equal to the kernel similarity.

Note: If a function f solves KDE, we incur $n^{1+\alpha}$ preprocess time and n^{β} query time for $\alpha, \beta < 1$, which implies that that total time for the attention algorithm would be $n^{1+\alpha} + n^{1+\beta}$.

3 HNSW: Hierarchical Navigable Small World Graphs



HNSW is a class of graph-based approximate nearest neighbor (ANN) data structures. The intuition is to move greedily toward the node whose neighbor is closest to the query. It is efficient in practice but lacks formal guarantees for random or arbitrary graphs. First, we need someway to define "intrinsic dimension." There exist multiple ways to define it, but we will use doubling dimension.

Definition 5. *For a metric space (P, d), the* doubling dimension $\mathrm{dd}(P)$ *is the smallest number such that for any $p \in P, r > 0$, the ball $B(p, r) \cap P$ can be covered by $2^{\mathrm{dd}(P)}$ balls $B(p_i, r/2)$ with $p_i \in P$.*

Fact 6. *If $P = \mathbb{R}^k$, then $\mathrm{dd}(P) = \Theta(k)$.*

With infinite points, the doubling dimension should intuitively be k. Even if $P \subset \mathbb{R}^k$ is random, we have:

$$\mathrm{dd}(P) \leq \min\{\Theta(k), O(\log n)\}.$$

since JL proves that you can appproximately preserve all distances with dimension reduction to $O(\log n)$ space.

Theorem 7. *(Indyk–Xu '23) Disk ANN: We can build a graph G with degree $O((4\alpha)^{\mathrm{dd}} \log \Delta)$ where $\alpha > 1$ is the approximation factor and Δ is the aspect ratio $= \frac{\max distP}{\min_{\neq 0} distP}$.*

For each query:

$$\#\text{steps} \leq O\left(\log_\alpha \frac{\Delta}{(\alpha-1)\varepsilon}\right).$$

Guarantee:

$$C = \frac{\alpha+1}{\alpha-1} + \varepsilon, \quad \forall \varepsilon > 0.$$