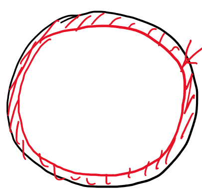
3. Johnson Lindenstrauss

Tuesday, September 14, 2021 11:26 PM

Last-time:
- Hyperballs are weird.
- We can use Gaussians to sample from hyperspheres.

Today:
- Gaussian annulus theorem
- Johnson-Lindenstrauss Lemma and random projections

Recall: Most points in a hyperball B_d are tightly concentrated near the "surface."

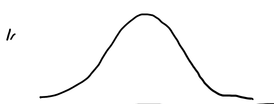


annulus of width $\frac{1}{d}$

Can we say something similar about spherical Gaussians?

Terms: Let $\vec{x} = (x_1, \dots, x_d)$, $x_i \sim \mathcal{N}(0,1)$ i.i.d. Then $\vec{x}$ is a spherical Gaussian r.v. centered at the origin with unit variance in every dir.

Note: Gaussians don't have a "boundary."



But, $$\mathbb{E}(|\vec{x}|^2) = \sum_{i=1}^{d} \mathbb{E}(x_i^2) = d\,\mathbb{E}(x_1^2) = d\Big[\underbrace{\mathbb{E}(x_1^2) - \overset{0}{\mathbb{E}(x_1)^2}}_{Var(x_1)}\Big] = d$$

We call $\sqrt{d}$ the radius of the Gaussian.

So we know that the expected distance from the origin is $\sqrt{d}$.
Let's show that with high probability, we are near the expectation.

Gaussian annulus thm: Let $\vec{x} = (x_1, \dots, x_d)$, $x_i \sim \mathcal{N}(0,1)$ i.i.d. Let $r = |\vec{x}|$.
Then $\forall \beta \le \sqrt{d}$, $\text{Prob}(|r - \sqrt{d}| \ge \beta) \le 3e^{-c\beta^2}$.

i.e. all prob. mass is concentrated near the radius.

proof. If $|r - \sqrt{d}| \ge \beta$, then $|r^2 - d| \ge \beta(r + \sqrt{d}) \ge \beta\sqrt{d}$.

Thus, $\text{Prob}(\underline{|r - \sqrt{d}| \ge \beta}) \le \text{Prob}(|r^2 - d| \ge \beta\sqrt{d})$ ← we can just bound this event!

└ this event is less likely because it implies the other event.

Note, $r^2 - d = (x_1^2 + \dots + x_d^2) - d = (x_1^2 - 1) + \dots + (x_d^2 - 1)$.

Let $y_i = x_i^2 - 1$. Then $\mathbb{E}\, y_i = \mathbb{E}\, x_i^2 - 1 = 0$.

Moments: And $\mathbb{E}(y_i^s) \le \mathbb{E}(|y_i|^s)$ } If $|y_i| \le 1$, $|y_i|^s \le 1$

Moments: And $\mathbb{E}(y_i^s) \le \mathbb{E}(|y_i|^s)$

$\le \mathbb{E}(1 + x_i^{2s})$

If $|x_i| \le 1$, $|y_i|^s \le 1$

If $|x_i| \ge 1$, $|y_i|^s \le |x_i^2|^s = x_i^{2s}$

$= 1 + \underbrace{\mathbb{E}(x_i^{2s})}_{\text{even moments of Gaussian, which we can look up}}$

$= 1 + (2s-1)!!$ (double factorial, prod. of all integers from 1 to $2s-1$ that have the same parity)

$\le 2^s s!$

$5!! = 5 \cdot 3 \cdot 1 = 15$

$7!! = 7 \cdot 5 \cdot 3 \cdot 1 = 105$

Then $\mathrm{Var}(y_i) = \mathbb{E}(y_i^2) - \underbrace{\mathbb{E}(y_i)^2}_{0} \le 2^2 \cdot 2 = 8.$

(Recall: Master Tail Bounds Thm: Let $x = x_1 + \dots + x_n$, where x_i's have 0 mean and variance at most σ^2. Let $0 \le a \le \sqrt{2}\, n\sigma^2$. If $|\mathbb{E}(x_i^s)| \le \sigma^2 s!$ for $s = 3, 4, \dots, \lfloor \frac{a^2}{4n\sigma^2} \rfloor$,

Then $\mathrm{Prob}(|x| \ge a) \le 3e^{\frac{-a^2}{12n\sigma^2}}$)

doesn't quite work because $\mathbb{E}(y_i^s) \le 2^s s! \not\le 8s!$ for $s \ge 3$.

Let $w_i = \frac{y_i}{2}$. Then $\mathrm{Var}(w_i) \le 2$ and $|\mathbb{E}(w_i^s)| \le 2s!$, so things work.

Then apply MTBT with $a = \frac{\beta\sqrt{d}}{2}$, $\sigma^2 = 2$, $n = d$

$$\Rightarrow \mathrm{Prob}(|r - \sqrt{d}| \ge \beta) \le \mathrm{Prob}\left(|w_1 + \dots + w_d| \ge \frac{\beta\sqrt{d}}{2}\right) \le 3e^{\frac{-\beta^2 d}{48 d \cdot 2}} = \underbrace{3e^{\frac{-\beta^2}{96}}}_{\text{not tight, but good enough.}}$$

Random Projection

Let $\vec{x}_1, \dots, \vec{x}_n \in \mathbb{R}^d$. For $\vec{v} \in \mathbb{R}^d$, want to find $\overbrace{\operatorname*{argmin}_i |\vec{x}_i - \vec{v}|}^{\text{nearest neighbor}}$.

Naive approach: Compare against each $\vec{x}_i$: $O(n)$ comparisons.

Takes $O(d)$ operations to compute $|\vec{x}_i - \vec{v}|$ for a given i.

Total runtime: $O(dn)$.

Can we do better if we are allowed to preprocess a database of $\vec{x}_i$'s?

Thm 2.10: (Random Projection)

Consider a matrix $A = \begin{bmatrix} u_{11} & \cdots & u_{1d} \\ \vdots & \ddots & \vdots \\ u_{k1} & \cdots & u_{kd} \end{bmatrix}$, where $u_{i,j} \sim \mathcal{N}(0,1)$ i.i.d.

(Want $k \ll d$)

Let $\vec{v} \in \mathbb{R}^d$. Then $\exists\, c > 0$ s.t. for $\varepsilon \in (0,1)$

$$\text{Prob}\left(\Big| \, |A\vec{v}| - \sqrt{k}\,|\vec{v}| \, \Big| \geq \varepsilon \sqrt{k}\, |\vec{v}| \right) \leq 3 e^{-ck\varepsilon^2}.$$

proof. WLOG, assume $|\vec{v}| = 1$, Say $\vec{v} = \begin{bmatrix} v_1 \\ \vdots \\ v_d \end{bmatrix}$.

$$A\vec{v} = \begin{bmatrix} u_{11}v_1 + \cdots + u_{1d}v_d \\ \vdots \\ u_{k1}v_1 + \cdots + u_{kd}v_d \end{bmatrix} = \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix},$$ letting $w_i = u_{i1}v_1 + \cdots + u_{id}v_d$ random variables

Result from probability: Sum of independents Gaussians is Gaussian

Since u_{ij} are $\mathcal{N}(0,1)$, $\mathbb{E}\, w_i = 0$.

$$\text{Var}(w_i) = \sum_{j=1}^{d} v_j^2 \, \text{Var}(u_{ij}) = \sum_{j=1}^{d} v_j^2 = 1.$$

Thus, $w_i \sim \mathcal{N}(0,1)$, so $\vec{w}$ is a k-dim spherical Gaussian.

By applying the Gaussian annulus theorem with d replaced by k, we complete the proof. $\square$

Thm J-L

For any $0 < \varepsilon < 1$, $n \in \mathbb{N}$, let $k \geq \frac{3}{c\varepsilon^2} \ln n$, with c as in the Gaussian annulus thm.

For any set of n pts in $\mathbb{R}^d$, the random projection $f(\vec{v}): \mathbb{R}^d \to \mathbb{R}^k$ defined by $A\vec{v}$ has the property that for all pairs of points $\vec{v}_i$ and $\vec{v}_j$, with prob. at least $1 - \frac{3}{2n}$,

$$(1-\varepsilon)\sqrt{k}\,|\vec{v}_i - \vec{v}_j| \leq |f(\vec{v}_i) - f(\vec{v}_j)| \leq (1+\varepsilon)\sqrt{k}\,|\vec{v}_i - \vec{v}_j|.$$

proof. Apply a union bound for every pair of points, after using random proj. thm.

If $k \geq \frac{3 \ln n}{c\varepsilon^2}$, then $3e^{-ck\varepsilon^2} \leq \frac{3}{n^3}$

There are $\binom{n}{2} < \frac{n^2}{2}$ pairs of points, so the prob. that any pair of points has large distortion is $< \frac{3}{2n}$. $\square$

Time-complexity: $k = O(\frac{1}{\varepsilon^2}\log n)$, so if $k<d$, we reduce dimension.

Comparing 2 pts in $\mathbb{R}^k$ takes only $O(k)$ time (instead of $O(d)$)

Comparing against all n pts: $O(dn) \to O(kn) = O(\frac{1}{\varepsilon^2} n \log n)$.

Projecting down from $\mathbb{R}^d \to \mathbb{R}^k$ takes $O(kd)$ time per point,

so overall query takes $O(kd + kn) = O(\frac{1}{\varepsilon^2}(\log n)(n+d))$ time.

However, preprocessing takes $O(kdn)$ time to build the database,

because A is a dense matrix with kd entries.

Can we do better?

[Kane, Nelson, 2010], [Dasgupta-Kumar-Sarlós, 2010], ...

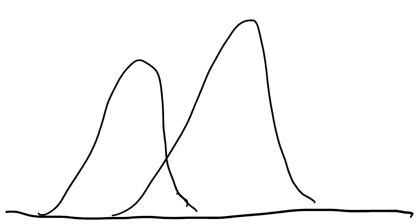
prove that we don't even need Gaussians and can use alternate A, such as sparse Bernoulli $\{-\frac{1}{2}, \frac{1}{2}\}$ matrices.

Clustering mixture of spherical Gaussians (simple Gaussian mixture model)

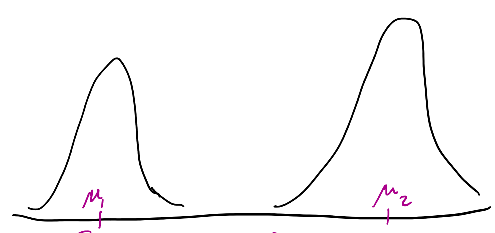
Task: partition a set of points into 2 subsets or clusters where each consists of points from a spherical Gaussian, and determine the parameters of the Gaussians.

Proposed naive solution:
1. Cluster our data by grouping nearby points.
2. Fit a Gaussian to each cluster

easy because it turns out that the best-fit spherical Gaussian is just the one with the empirical mean + variance.



difficult to cluster when there is a lot of overlap



Easier when far apart

e.g. $|\mu_1 - \mu_2| > 6\max(\sigma_1, \sigma_2)$

$\Rightarrow < 0.03$ overlap.

In high dimensions: If $\vec{x}, \vec{y} \sim \mathcal{N}(\mu, \sigma)$, $\vec{x}, \vec{y} \in \mathbb{R}^d$,

then $|\vec{x} - \vec{y}|^2 \approx 2(\sqrt{d} \pm O(1))^2 \sigma^2$

radius of Gaussian

then $|\vec{x}-\vec{y}|^2 \approx 2(\sqrt{d} \pm O(1))^2\sigma^2$

radius of Gaussian

orthogonality of points

$$\text{If } \vec{x} \sim \mathcal{N}(\mu_1, \sigma), \quad \vec{y} \sim \mathcal{N}(\mu_2, \sigma), \quad |\mu_1 - \mu_2| = \Delta,$$

$$\text{then } |\vec{x}-\vec{y}|^2 \approx \underbrace{2(\sqrt{d} \pm O(1))^2 \sigma^2 + \Delta^2}_{\text{all directions "orthogonal"}}$$

So we need $2(\sqrt{d} \pm O(1))^2\sigma^2 + \Delta^2 > 2(\sqrt{d} \pm O(1))^2\sigma^2$

$$(2d \pm O(1)\sqrt{d})\sigma^2 + \Delta^2 > (2d \pm O(1)\sqrt{d})\sigma^2$$

$$\Delta^2 > O(1)\sqrt{d}\,\sigma^2$$

$$\Delta > O(1)\,\sigma\, d^{\frac{1}{4}} = c\,\sigma\, d^{\frac{1}{4}}, \text{ for some constant } c.$$

unfortunately, separation needed is dependent on dimension, instead of just standard deviation.

More sophisticated: Use a clever projection based on the "Singular-Value-Decomposition" to remove dependency and get better separation.