K-nearest Neighbor

Announcements:

1. HW1 will be out today / early tomorrow and Due Sep 12

2. P1 will be out later this week

Recap on ML basics

T/F: A hypothesis that achieves zero training error is always good

T/F: zero-one loss is a good loss function for regression

T/F: We can use validation dataset to check if our model overfits

Objective

Understand KNN — our first ML algorithm that can do both regression and classification

Outline for Today

1. The K-NN Algorithm

2. Why/When does K-NN work

3. Curse of dimensionality (i.e., when it can fail)

The K-NN Algorithm

Input: classification training dataset $\{x_i, y_i\}_{i=1}^n$, and parameter $K \in \mathbb{N}^+$, and a distance metric $d(x, x')$ (e.g., $\|x - x'\|_2$ euclidean distance)

K-NN Algorithm:

Store all training data

For any test point x :

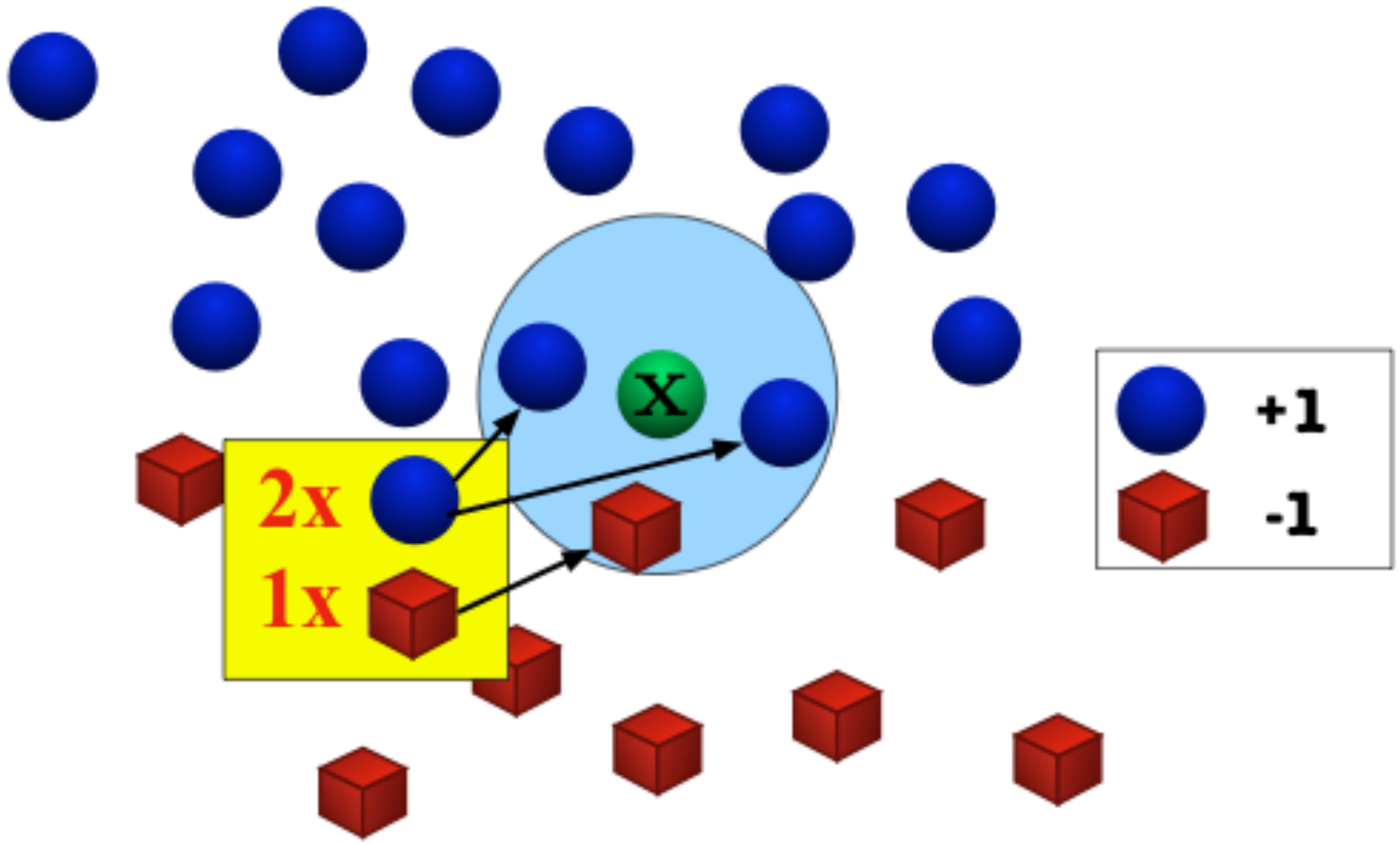
Find its top K nearest neighbors (under metric d)

Return the most common label among these K neighbors

(If for regression, return the average value of the K neighbors)

The K-NN Algorithm

Example: 3-NN for binary classification using Euclidean distance



The choice of metric

1. We assume our metric d captures similarities between examples:

Examples that are close to each other under distance d share similar labels

Another example: Manhattan distance (ℓ_1)

$$d(x, x') = \sum_{j=1}^{d} |x[j] - x'[j]|$$

The choice of K

1. What if we set K very large?

Top K-neighbors will include examples that are very far away…

2. What if we set K very small (K=1)?

label has noise (easily **overfit** to the noise)

(What about the training error when K = 1?)

Outline for Today

1. The K-NN Algorithm ✓

2. Why/When does K-NN work

3. Curse of dimensionality (i.e., why it can fail in high-dimension data)

Bayes Optimal Predictor

Assume our data is collected in an i.i.d fashion, i.e., $(x, y) \sim P$ (say $y \in \{-1,1\}$)

Assume we know $P(y \mid x)$ for now

Q: what label you would predict?

A: we will simply predict the most-likely label,

$$h_{opt}(x) = \arg \max_{y \in \{-1,1\}} P(y \mid x)$$

Bayes optimal predictor

Bayes Optimal Predictor

Assume our data is collected in an i.i.d fashion, i.e., $(x, y) \sim P$ (say $y \in \{-1, 1\}$)

Bayes optimal predictor: $$h_{opt}(x) = \arg\max_{y \in \{-1,1\}} P(y \mid x)$$

Example:

$$\begin{cases} P(1 \mid x) = 0.8 \\ P(-1 \mid x) = 0.2 \end{cases}$$

$$y_b := h_{opt}(x) = 1$$

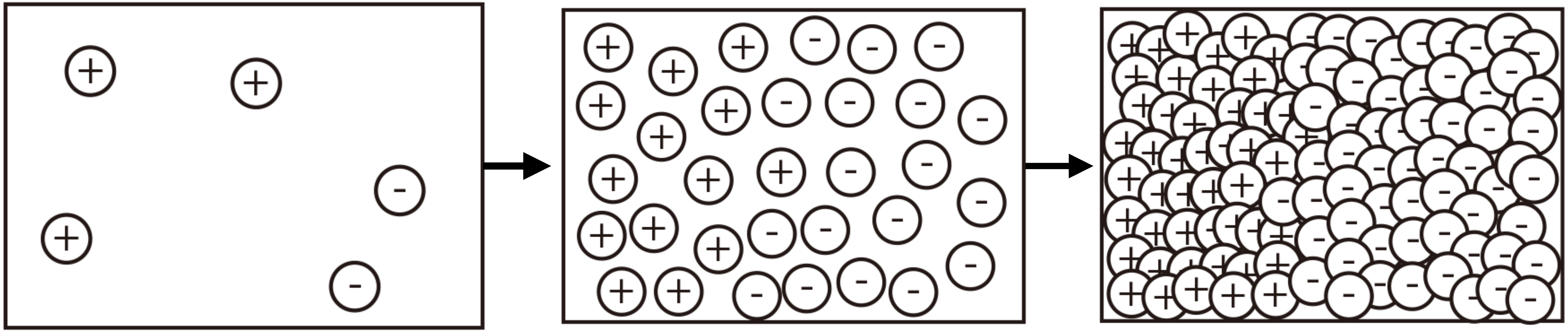
Q: What's the probability of h_{opt} making a mistake on x?

$$\epsilon_{opt} = 1 - P(y_b \mid x) = 0.2$$

Guarantee of KNN when $K = 1$ and $n \to \infty$

Assume $x \in [-1,1]^2$, $P(x)$ has support everywhere $P(x) > 0, \forall x \in [-1,1]^2$

What does it look when $n \to \infty$?



Given test x, as $n \to \infty$, its nearest neighbor x_{NN} is super close, i.e., $d(x, x_{NN}) \to 0$!

Guarantee of KNN when $K = 1$ and $n \to \infty$

Theorem: as $n \to \infty$, 1-NN prediction error is **no more than twice** of the error of the Bayes optimal classifier

Proof:

1. Fix a test example x, denote its NN as x_{NN}. When $n \to \infty$, we have $x_{NN} \to x$
2. WLOG assume for x, the Bayes optimal predicts $y_b = h_{opt}(x) = 1$
3. Calculate the 1-NN's prediction error:

Case 1 when $y_{NN} = 1$ (it happens w/ prob $P(1 \mid x_{NN}) = P(1 \mid x)$):

The probability of making a mistake: $\epsilon = P(y \neq 1 \mid x) = P(y = -1 \mid x)$

$$= 1 - P(y_b \mid x)$$

Guarantee of KNN when $K = 1$ and $n \to \infty$

Theorem: as $n \to \infty$, 1-NN prediction error is **no more than twice** of the error of the Bayes optimal classifier

✔ **Case 1** when $y_{NN} = 1$ (it happens w/ prob $P(1 \mid x_{NN}) = P(1 \mid x)$):

The probability of making a mistake: $\epsilon = 1 - P(y_b \mid x)$

Case 2 when $y_{NN} = -1$ (it happens w/ prob $P(-1 \mid x_{NN}) = P(-1 \mid x)$):

The probability of making a mistake: $\epsilon = P(y \neq -1 \mid x) = P(y = 1 \mid x) = P(y_b \mid x)$

Final prediction error at x:

$$P(1 \mid x)(1 - P(y_b \mid x)) + P(-1 \mid x)P(y_b \mid x) = P(1 \mid x)(1 - P(y_b \mid x)) + (1 - P(y_b \mid x))P(y_b \mid x)$$

$$\leq (1 - P(y_b \mid x)) + (1 - P(y_b \mid x)) = 2\epsilon_{opt}$$

What happens if K is large?
(e.g., $K = 1e6, n \to \infty$)

A: Given any x, the K-NN should return the y_b — the solution of the Bayes optimal

Outline for Today

1. The K-NN Algorithm ✓

2. Why/When does K-NN work ✓

3. Curse of dimensionality (i.e., why it can fail in high-dimension data)

Finite sample error rate of 1-NN in high-dimension setting

(Informal result and no proof)

Fix $n \in \mathbb{N}^+$, assume $x \in [0,1]^d$, assume $P(y \mid x)$ is Lipschitz continuous with respect to x, i.e., $\mid P(y \mid x) - P(y \mid x') \mid \leq d(x, x')$

Then, we have:

$$\mathbb{E}_{x,y \sim P}\left[\mathbf{1}(y \neq \text{1NN}(x))\right] \leq 2\mathbb{E}_{x,y \sim P}\left[\mathbf{1}(y \neq h_{opt}(x))\right] + O\left(\left(\frac{1}{n}\right)^{1/d}\right)$$

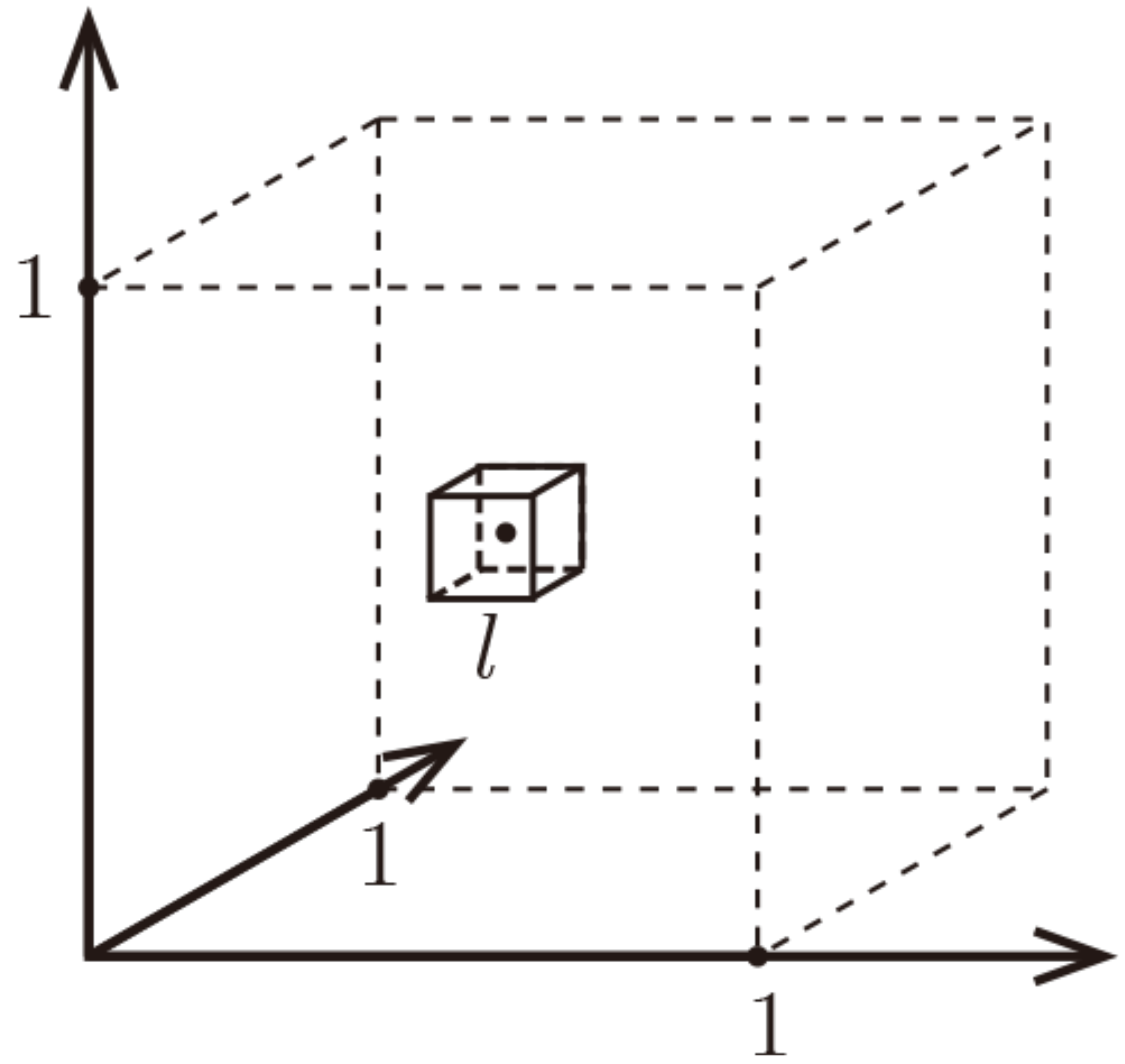
Curse of dimensionality!

The bound is meaningless when $d \to \infty$, while n is some finite number!

Curse of Dimensionality Explanation

Key problem: in high dimensional space, points that are draw from a distribution tends to be far away from each other!

Example: let us consider uniform distribution over a cube $[0,1]^d$

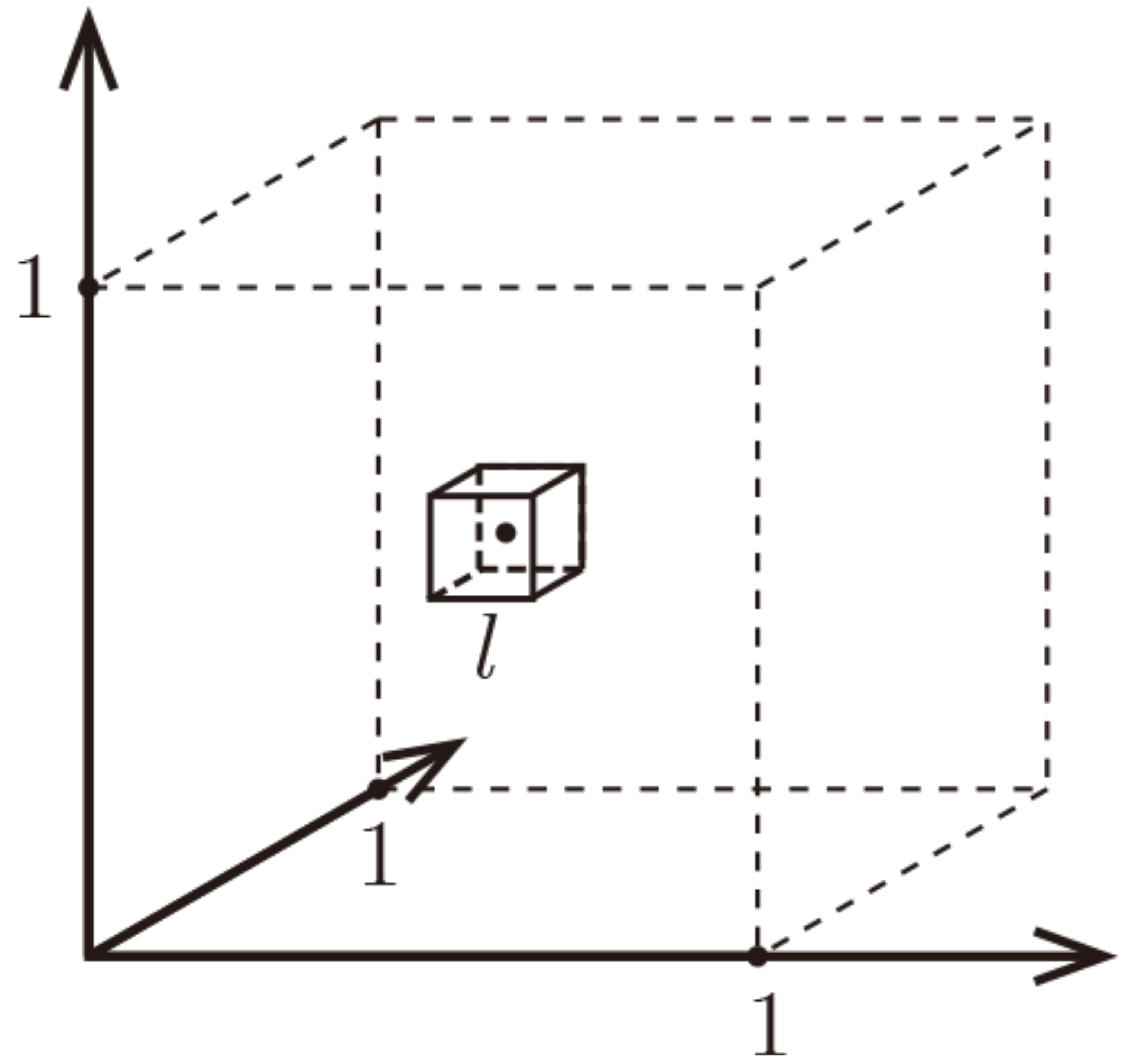


Q: sample x uniformly, what is the probability that x is inside the small cube?

A: Volume(small cube)/volume($[0,1]^d$) $= l^d$

Curse of Dimensionality Explanation

Example: let us consider uniform distribution over a cube $[0,1]^d$



Now assume we sampled n points uniform randomly, and we observed K points fall inside the small cube

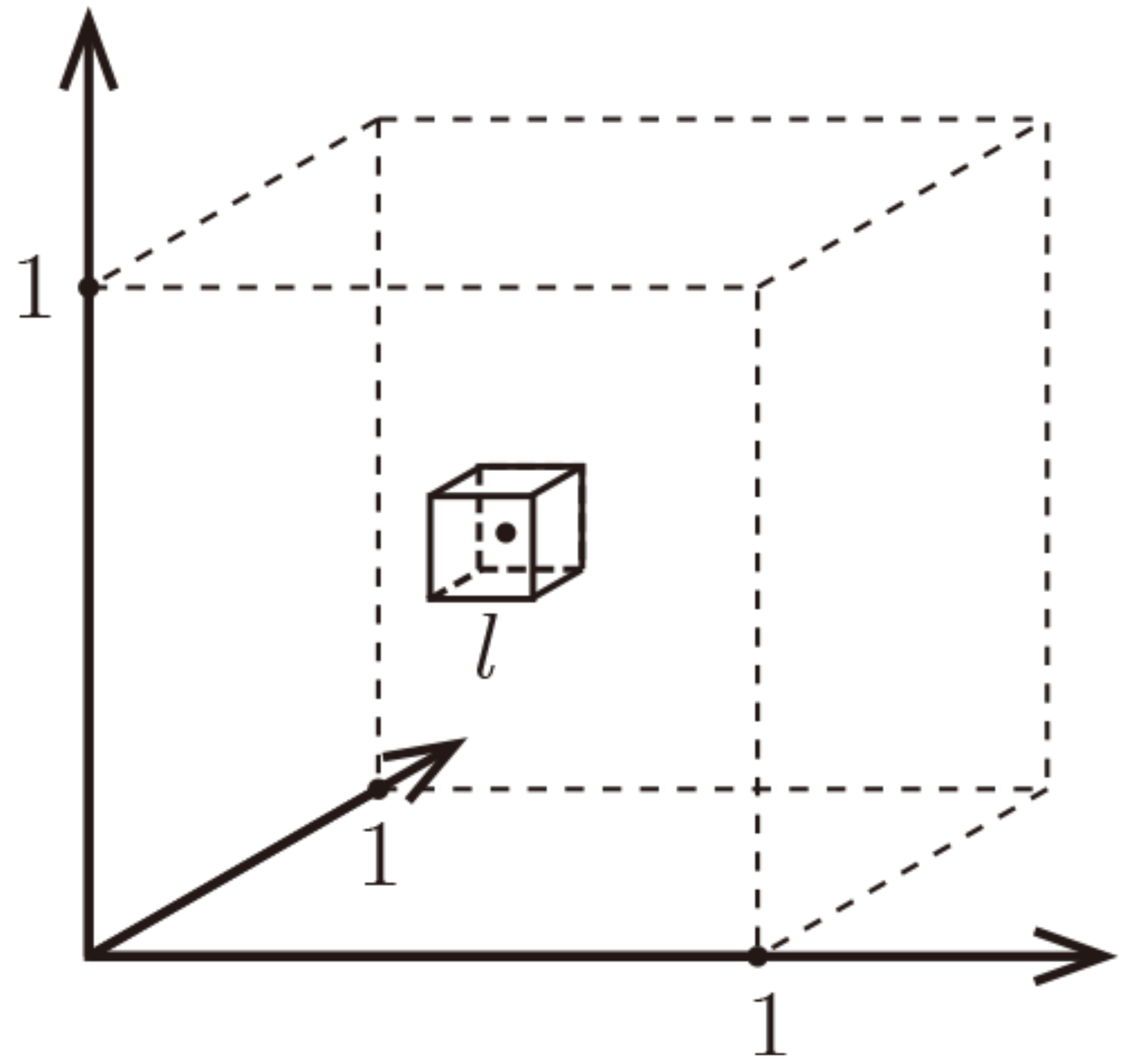
So empirically, the probability of sampling a point inside the small cube is roughly K/n

Thus, we have $l^d \approx \frac{K}{n}$

Curse of Dimensionality Explanation

Example: let us consider uniform distribution over a cube $[0,1]^d$

We have $l^d \approx \dfrac{K}{n}$



Q: how large we should set l, s.t., we will have K examples (out of n) fall inside the small cube?

$$l \approx (K/n)^{1/d} \to 1, \text{ as } d \to \infty$$

Bad news: when $d \to \infty$, the K nearest neighbors will be all over the place! (Cannot trust them, as they are not nearby points anymore!)

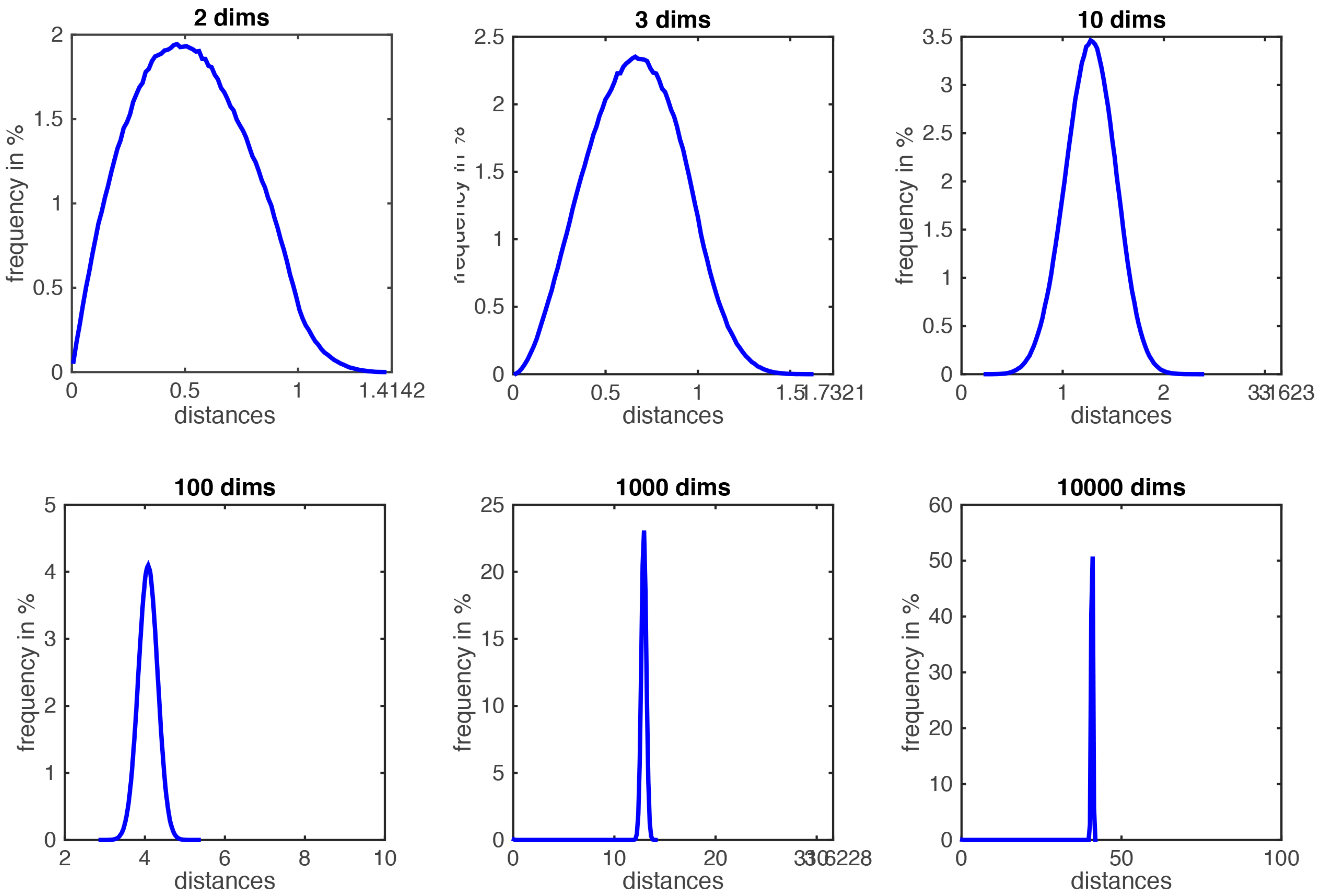
The distance between two sampled points increases as d grows

In $[0,1]^d$, we uniformly sample two points x, x', calculate

$$d(x, x') = \|x - x'\|_2$$

Let's plot the distribution of such distance:

Q: can you compute $\mathbb{E}_{x,x'}\|x - x'\|_2^2$?

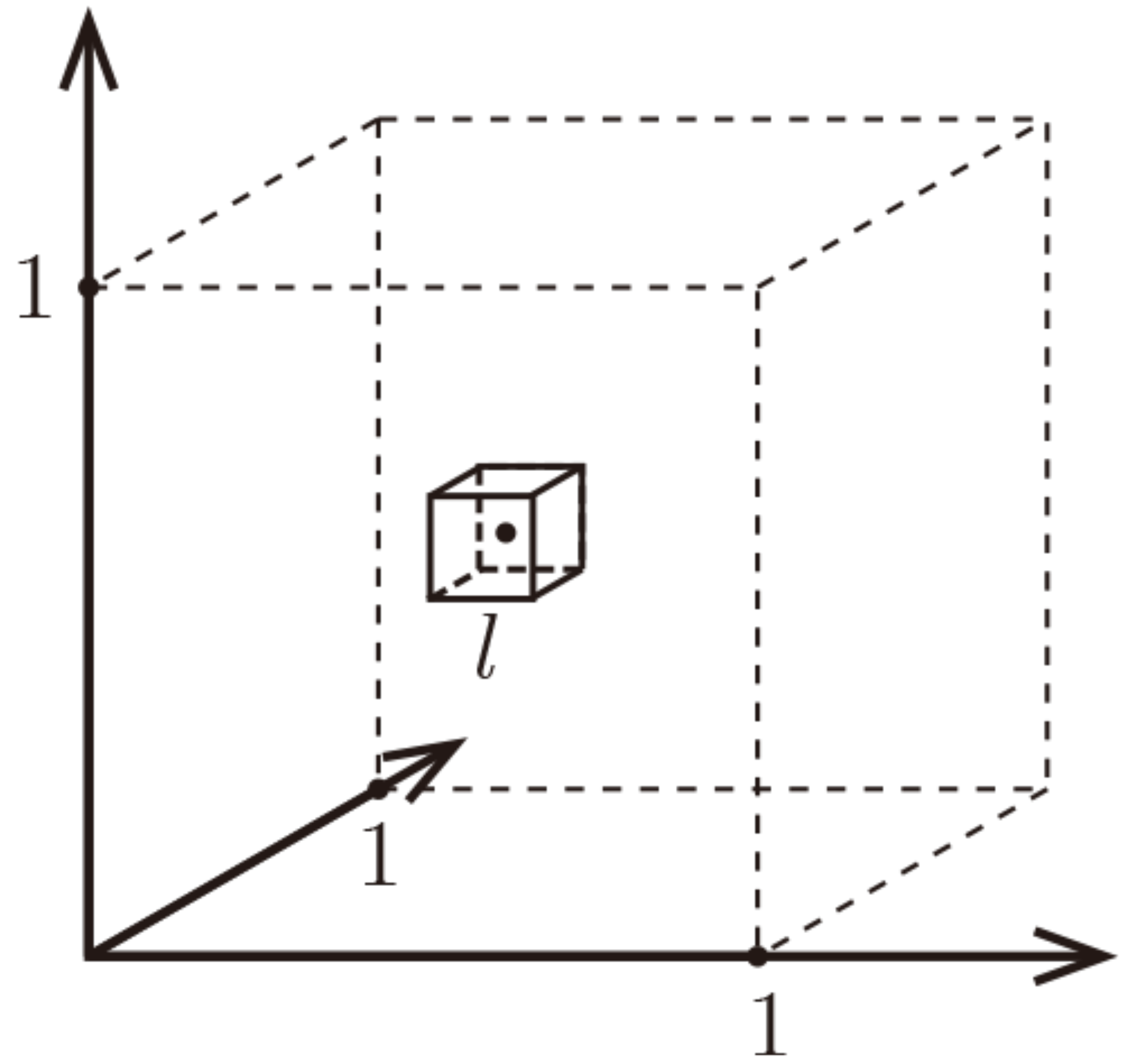


Distance increases as $d \to \infty$

Well, can we just increase n to avoid this?

Example: let us consider uniform distribution over a cube $[0,1]^d$

We have $l^d \approx \frac{K}{n}$

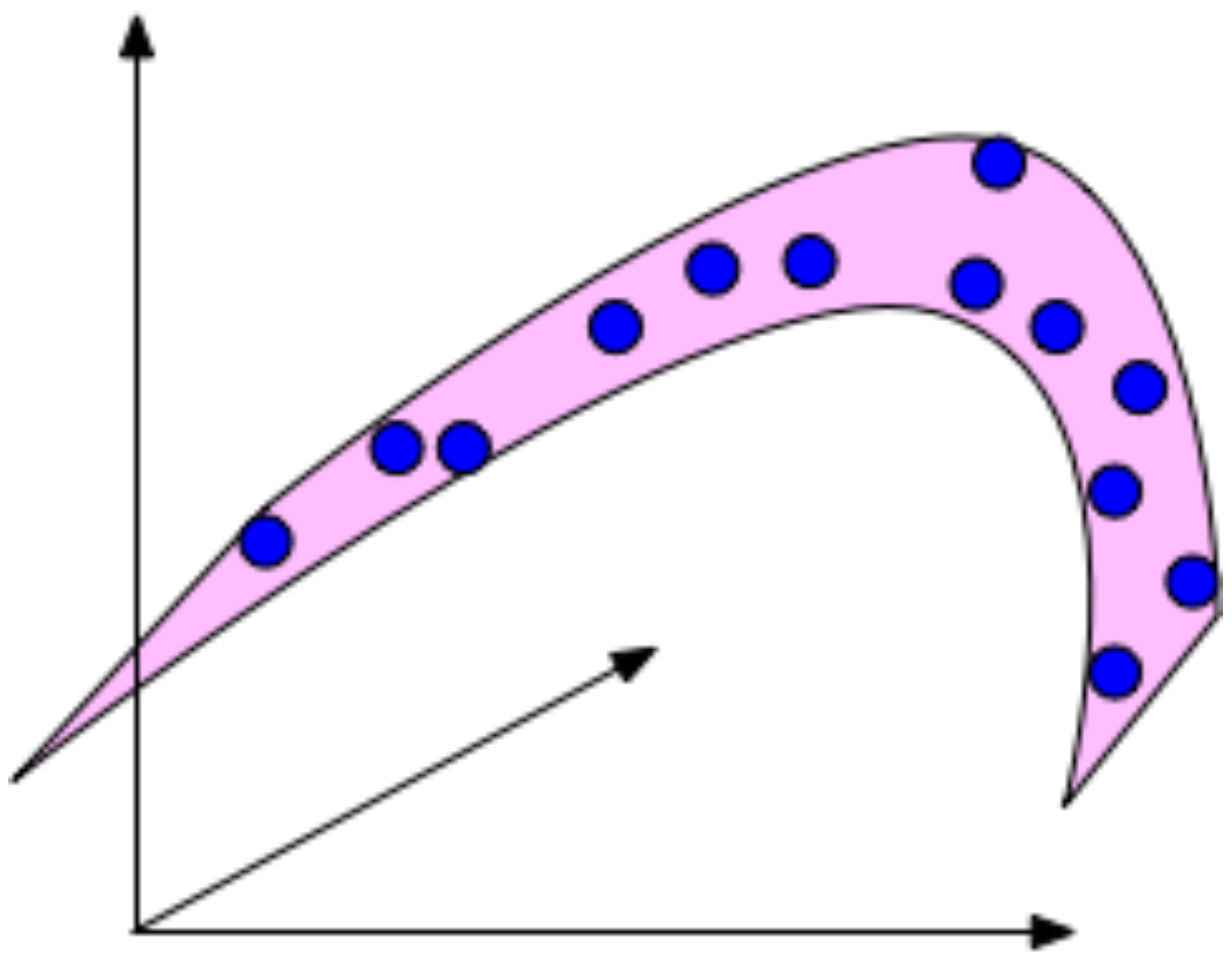


Q: to make sure that we have one sample inside a small cube, how large n needs to be?

Set $\ell = 0.1, K = 1$, then $n = 1/(0.1)^d = 10^d$

Bad news: when $d \geq 100$, # of samples needs to be larger than total # of atoms in the universe!

Luckily, real world data often has low-dimensional structure!



Data lives in 2-d manifold

Example: face images

Original image: $\mathbb{R}^{64^2}$

Next week: we will see that these faces approximately live in 100-d space!

Summary for Today

1. K-NN: the simplest ML algorithm (very good baseline, should always try!)

2. Works well when data is low-dimensional (e.g., can compare against the Bayes optimal)

3. Suffer when data is high-dimensional, due to the fact that in high-dimension space, data tends to spread far away from each other