K-nearest Neighbor

Announcement:

1. HW1 will be out today / early tomorrow and Due Sep 13

Recap

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:

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

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 :

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)

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

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 :

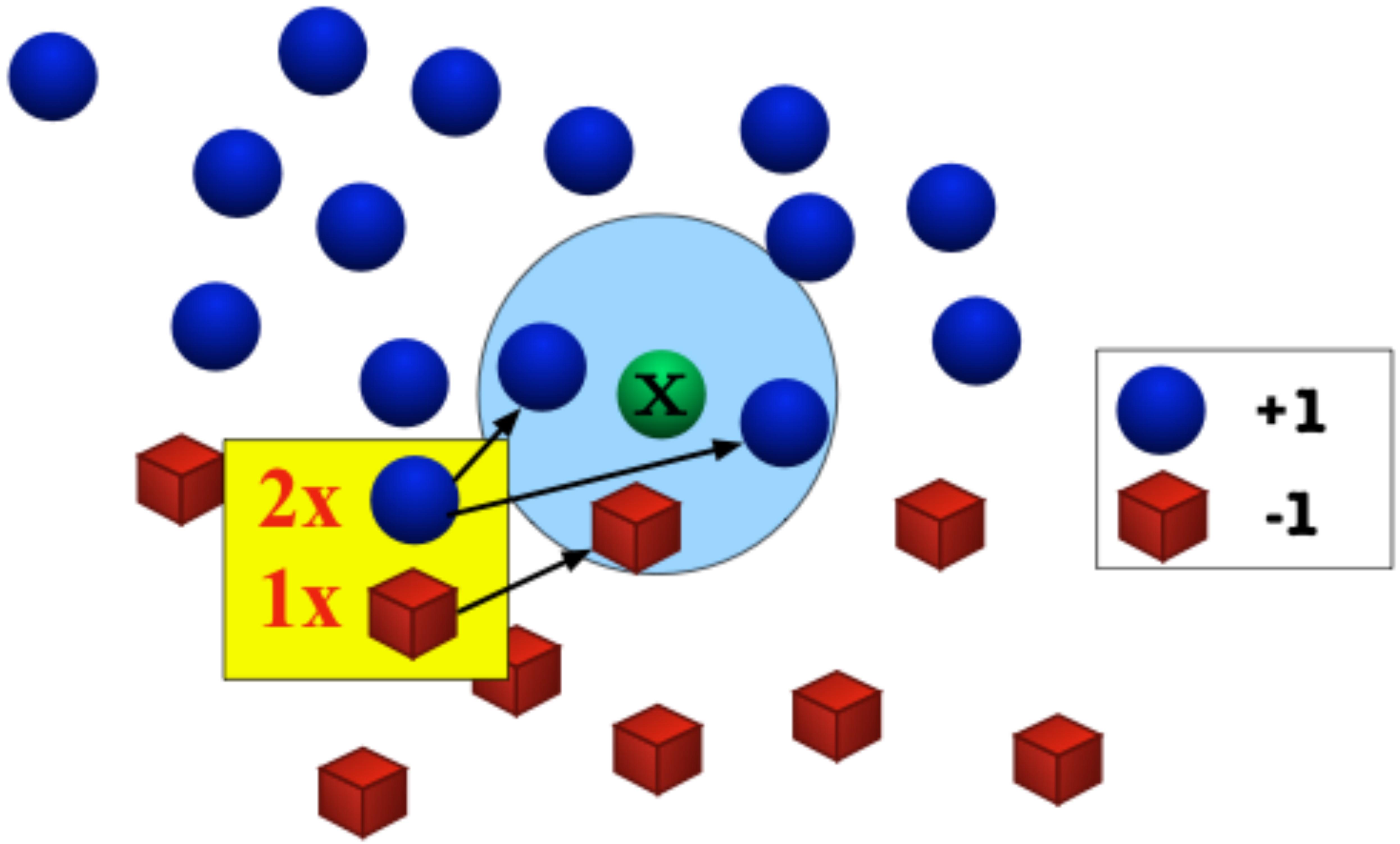
Find its top K nearest neighbors (under metric d)

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 believe our metric d captures similarities between examples:

Examples that are close to each other share similar labels

The choice of metric

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

Examples that are close to each other 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?

The choice of K

1. What if we set K very large?

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

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)?

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)

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 $y \in \{-1, 1\}$)

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?

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

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:

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

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:

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

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

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

Example:

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

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

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

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

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

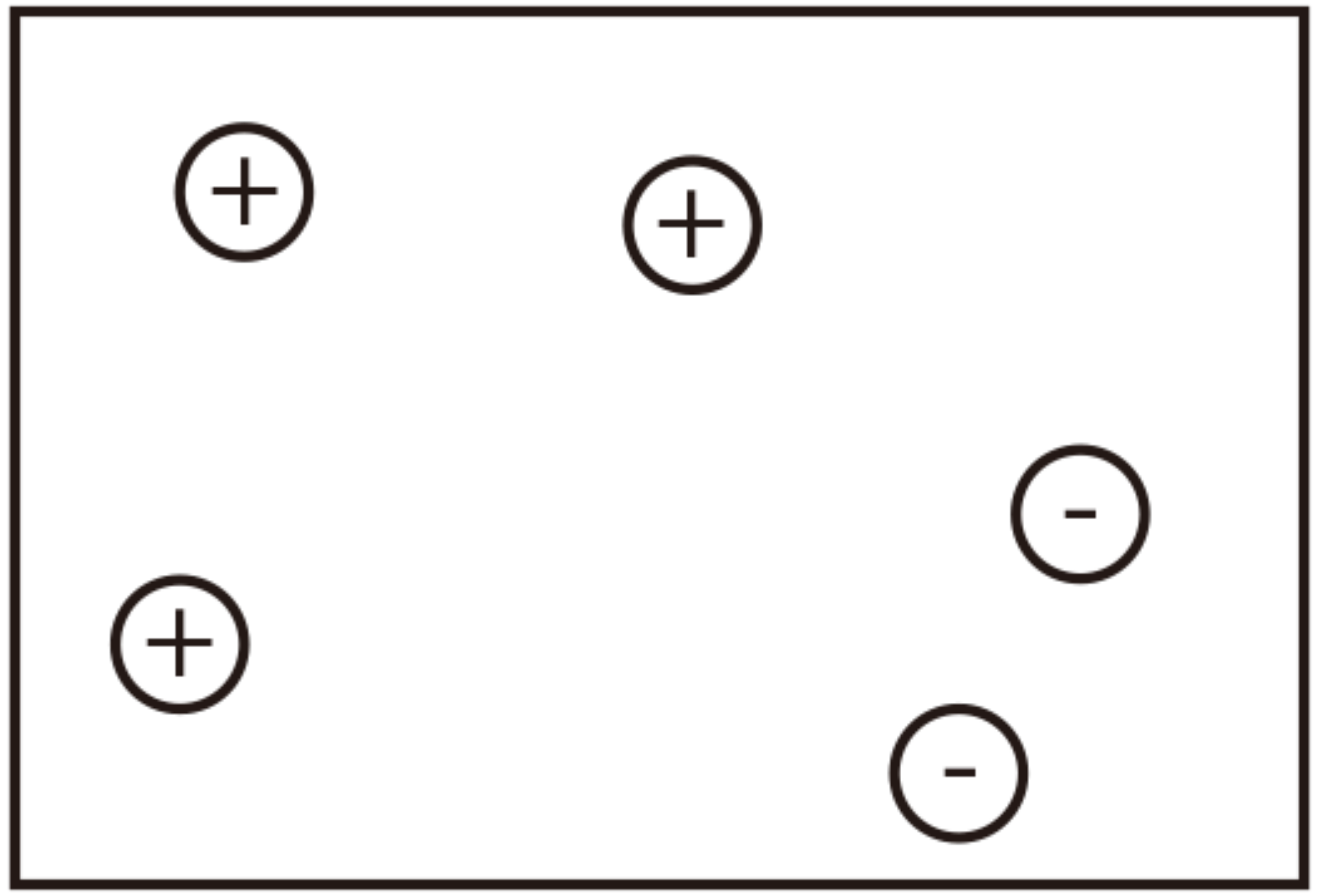
Assume $x \in [-1,1]^2$, $P(x)$ has support everywhere $P(x) > 0, \forall x \in [-1,1]^2$

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

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

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

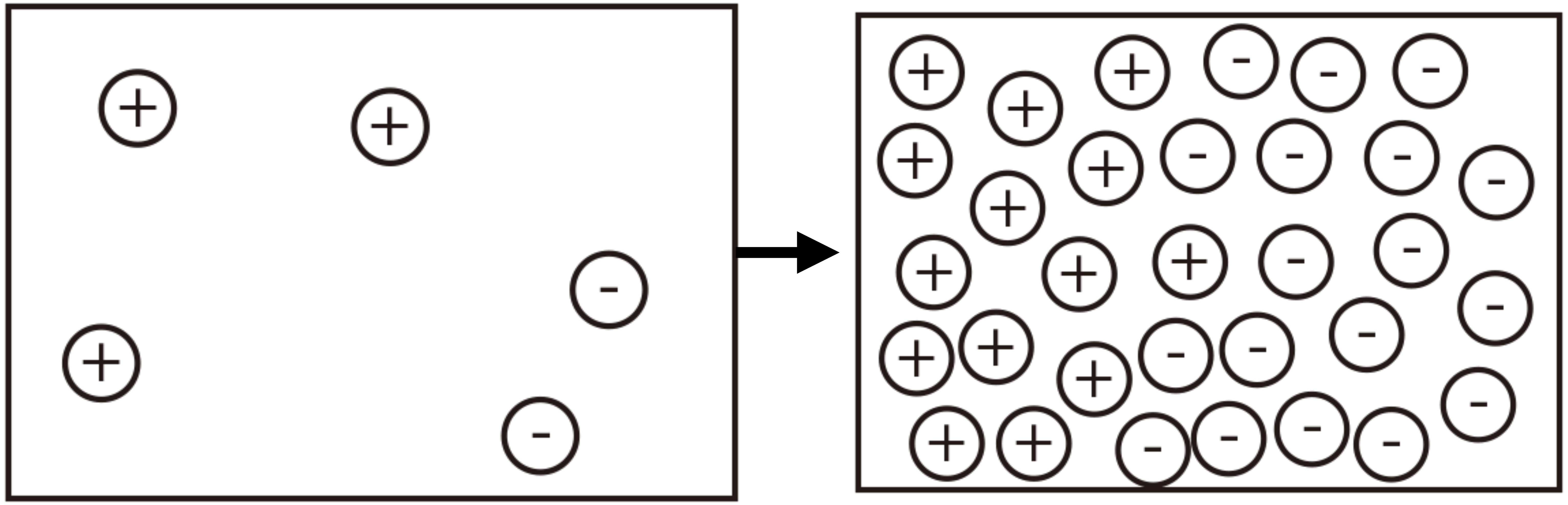
What does it look when $n \to \infty$?



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

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

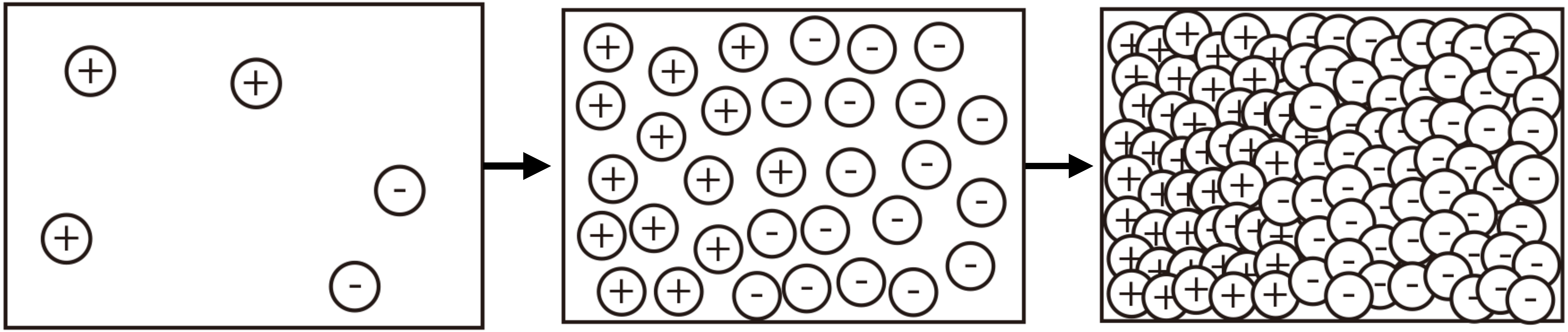
What does it look when $n \to \infty$?



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

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

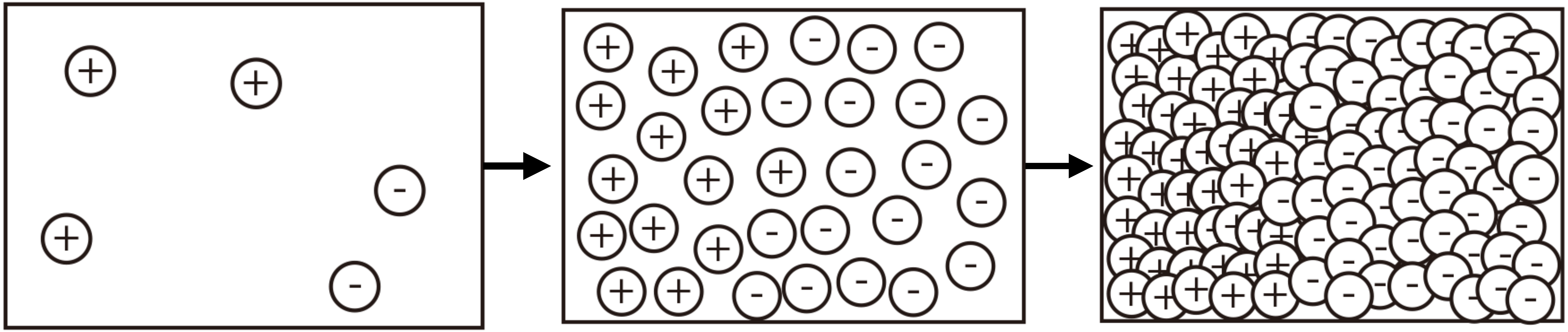
What does it look when $n \to \infty$?



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:

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 $x_{NN} \to x$

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

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:

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 $P(1 \mid x_{NN}) = P(1 \mid x)$):

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: $\epsilon = P(y \neq 1 \mid x) = P(y = -1 \mid x)$

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

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 $P(-1 \mid x_{NN}) = P(-1 \mid x)$):

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: $\epsilon = P(y \neq -1 \mid x) = P(y = 1 \mid x)$

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: $\epsilon = P(y \neq -1 \mid x) = P(y = 1 \mid x) = P(y_b \mid x)$

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: $\epsilon = P(y \neq -1 \mid x) = P(y = 1 \mid x) = P(y_b \mid x)$

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: $\epsilon = P(y \neq -1 \mid x) = P(y = 1 \mid x) = P(y_b \mid x)$

Our prediction error at 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: $\epsilon = P(y \neq -1 \mid x) = P(y = 1 \mid x) = P(y_b \mid x)$

Our prediction error at x:

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

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: $\epsilon = P(y \neq -1 \mid x) = P(y = 1 \mid x) = P(y_b \mid x)$

Our prediction error at x:

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

What happens if K is large?

(e.g., $K = 1e6, n \to \infty$)

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)

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., $|P(y \mid x) - P(y \mid x')| \leq d(x, x')$

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., $|P(y \mid x) - P(y \mid x')| \leq d(x, x')$

Then, we have:

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 $P(y|x)$ is Lipschitz continuous with respect to x, i.e., $|P(y|x) - P(y|x')| \leq d(x, x')$

Then, we have:

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

<u>Finite sample</u> error rate of 1-NN in <u>high-dimension</u> 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., $|P(y \mid x) - P(y \mid x')| \leq d(x, x')$

Then, we have:

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

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

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., $|P(y \mid x) - P(y \mid x')| \leq d(x, x')$

Then, we have:

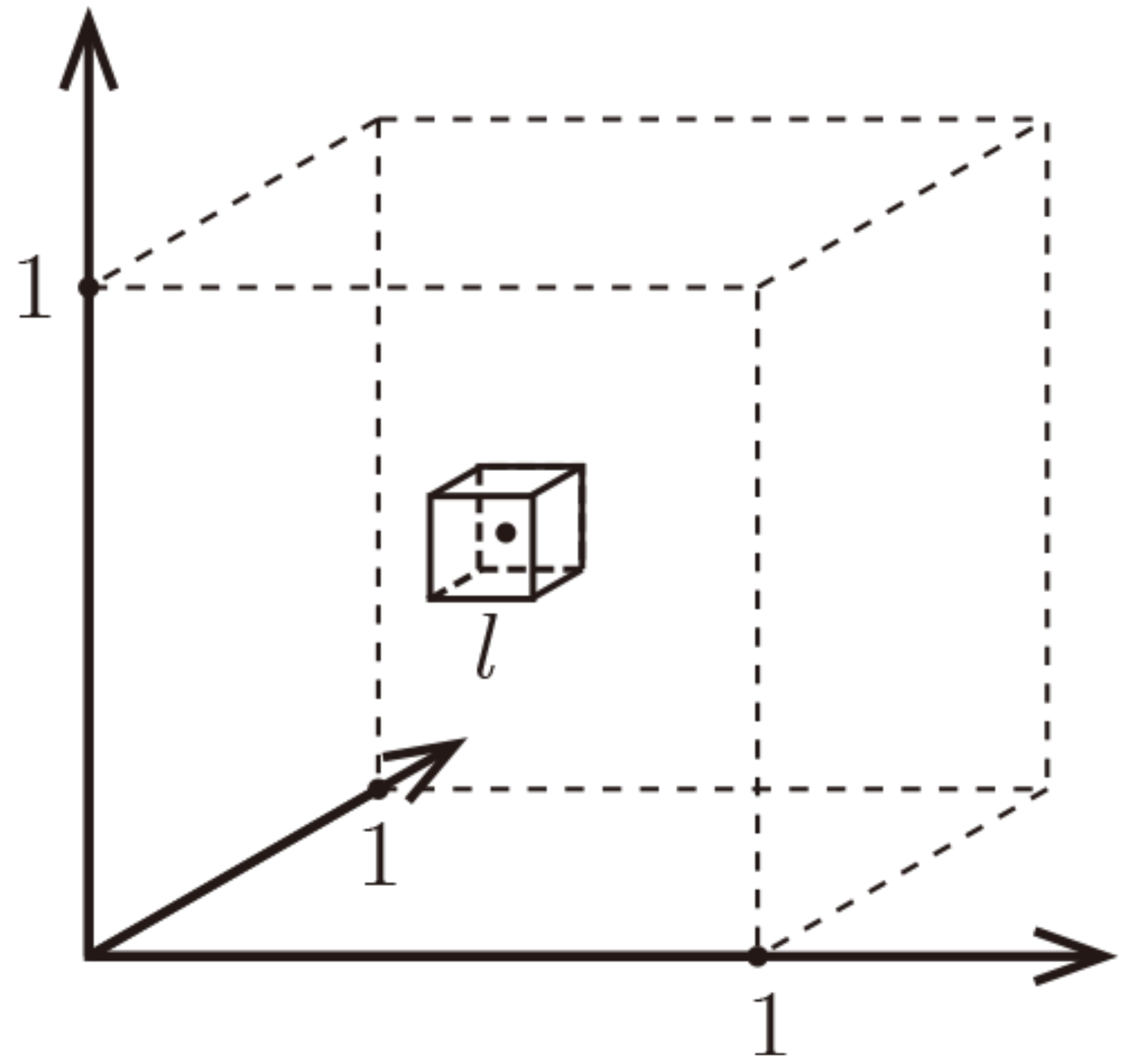
$$\mathbb{E}_{x,y \sim P}\left[\mathbf{1}(y \neq \text{1NN}(x))\right] \leq 2\mathbb{E}_{x,y \sim P}\left[\mathbf{1}(y \neq h_{opt}(x))\right] + O\left(\left(\frac{1}{n}\right)^{1/d}\right)$$

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

Curse of dimensionality!

Curse of Dimensionality Explanation

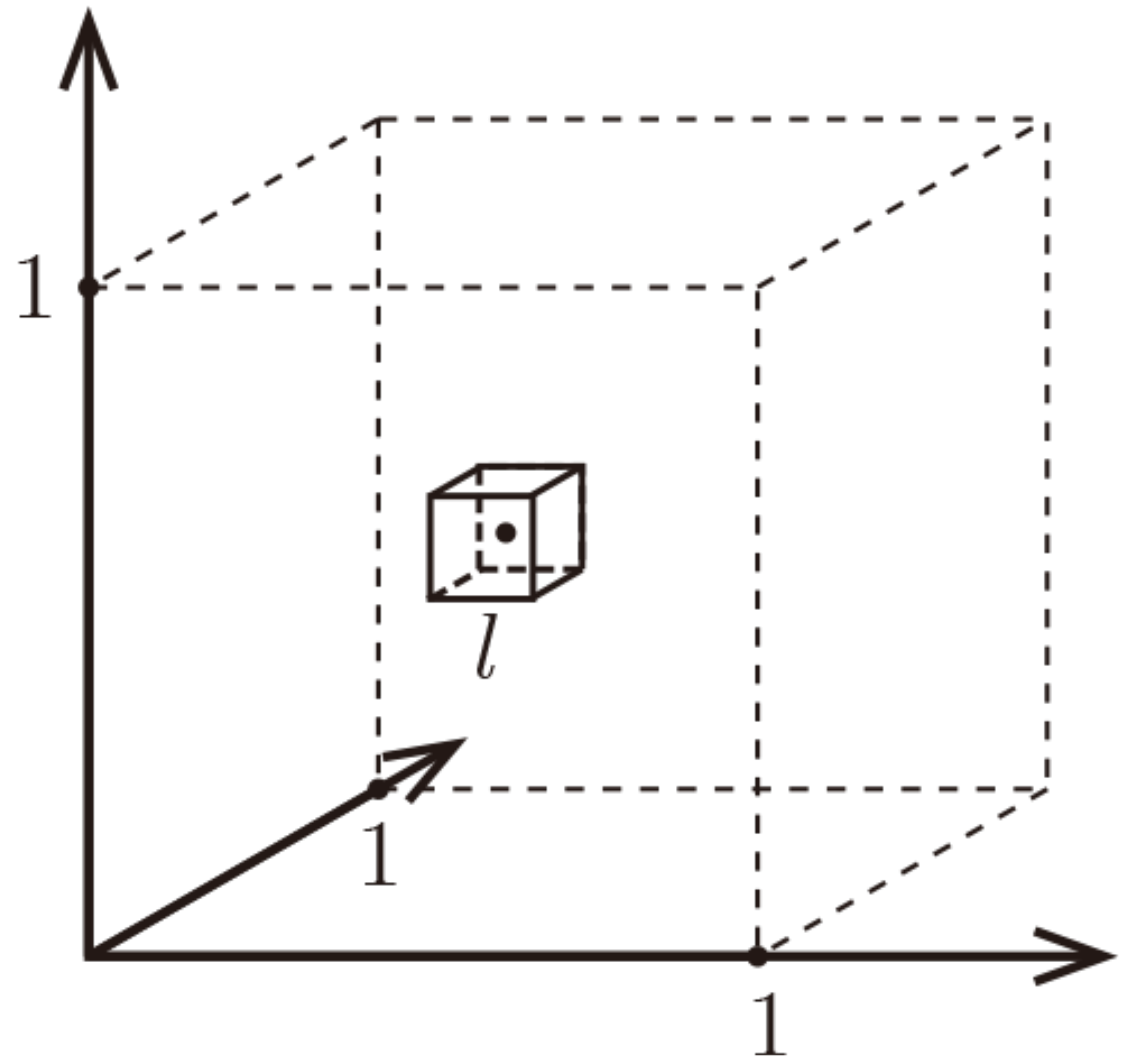
Key problem: in high dimensional space, points that are draw from a distribution tends to be far away from each other!



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!

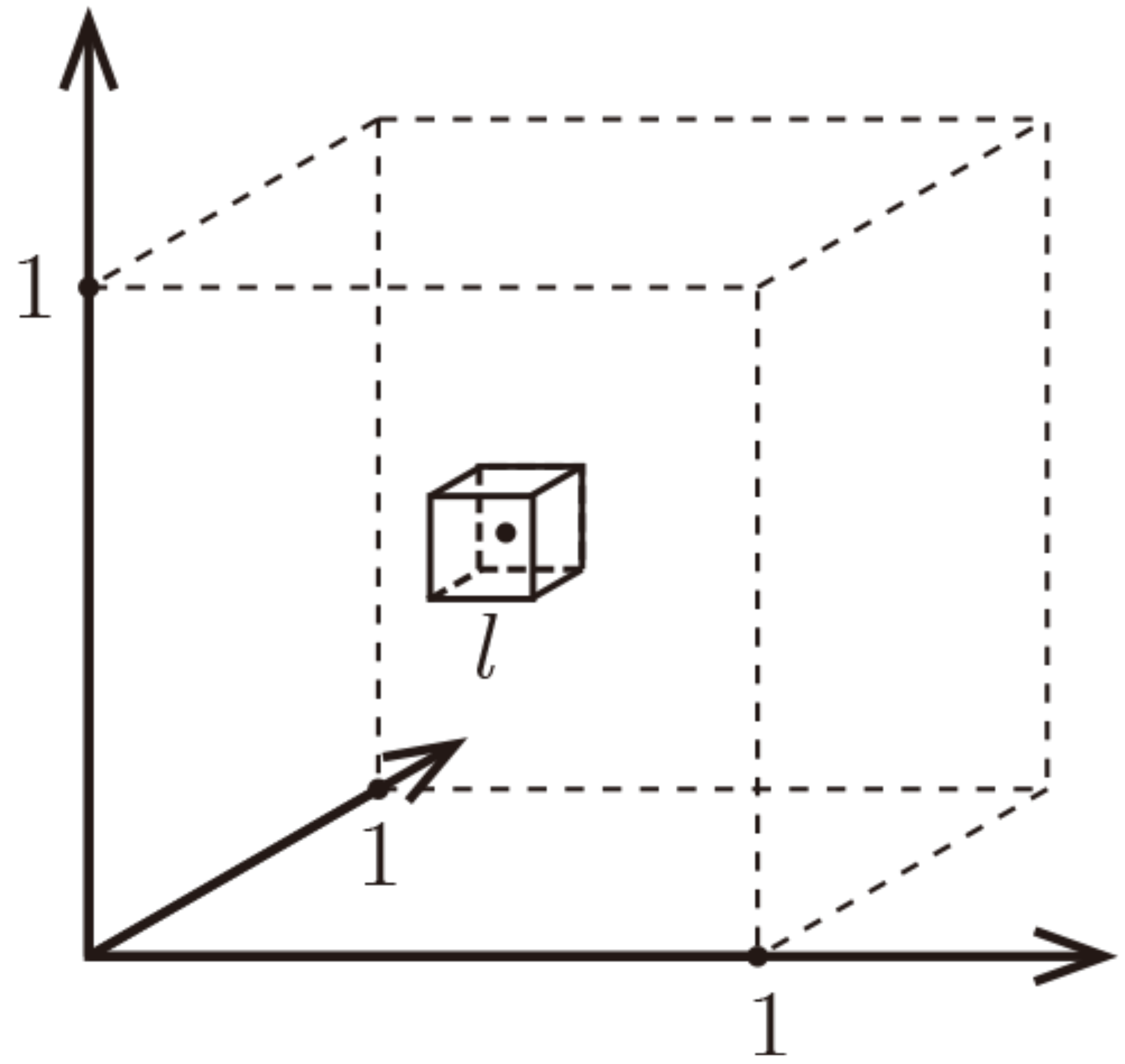
Example: let us consider uniform distribution over a cube $[0,1]^d$



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 $[0,1]^d$

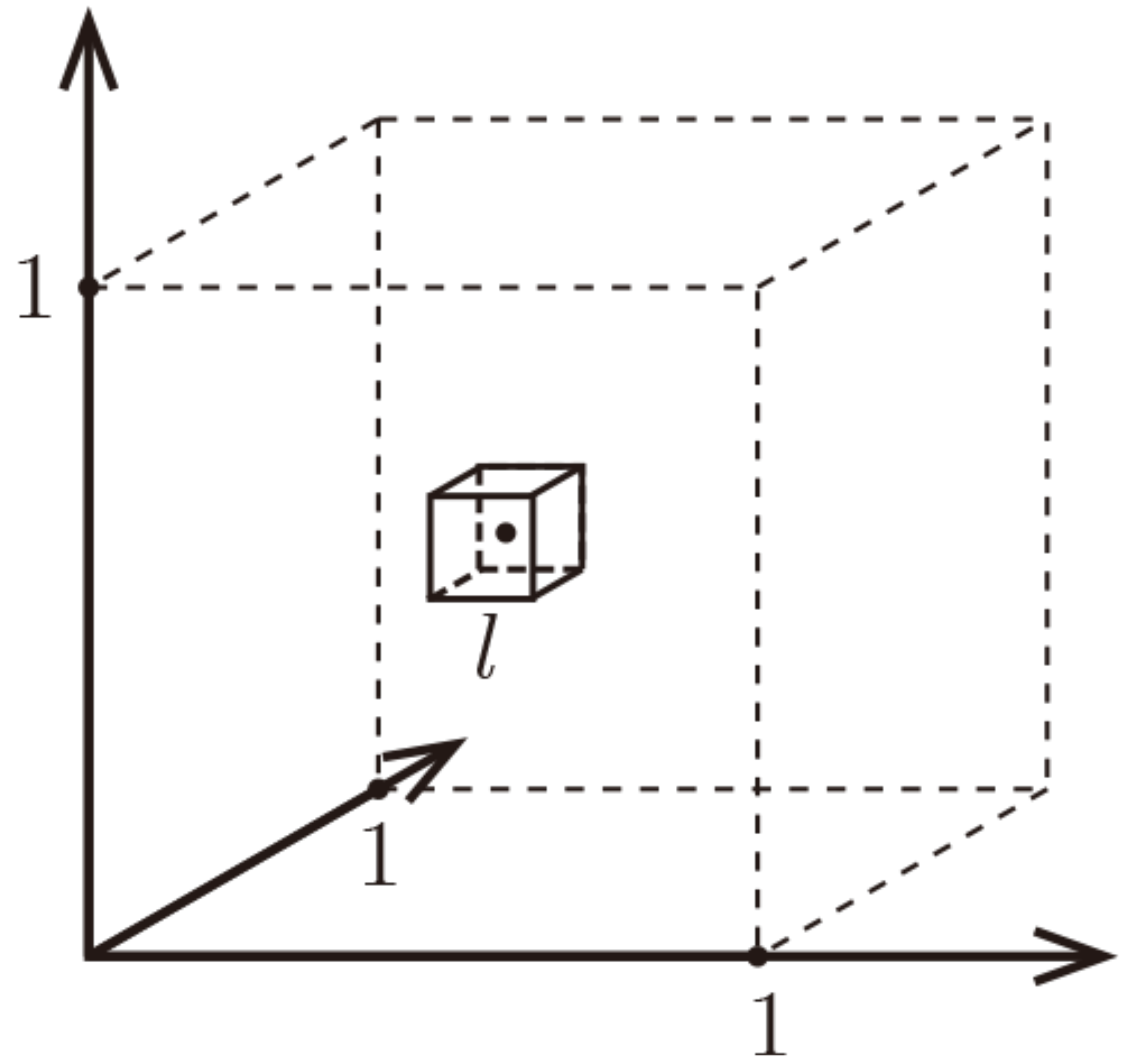


Q: sample x uniformly, what is the probability that x is inside the small cube?

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 $[0,1]^d$



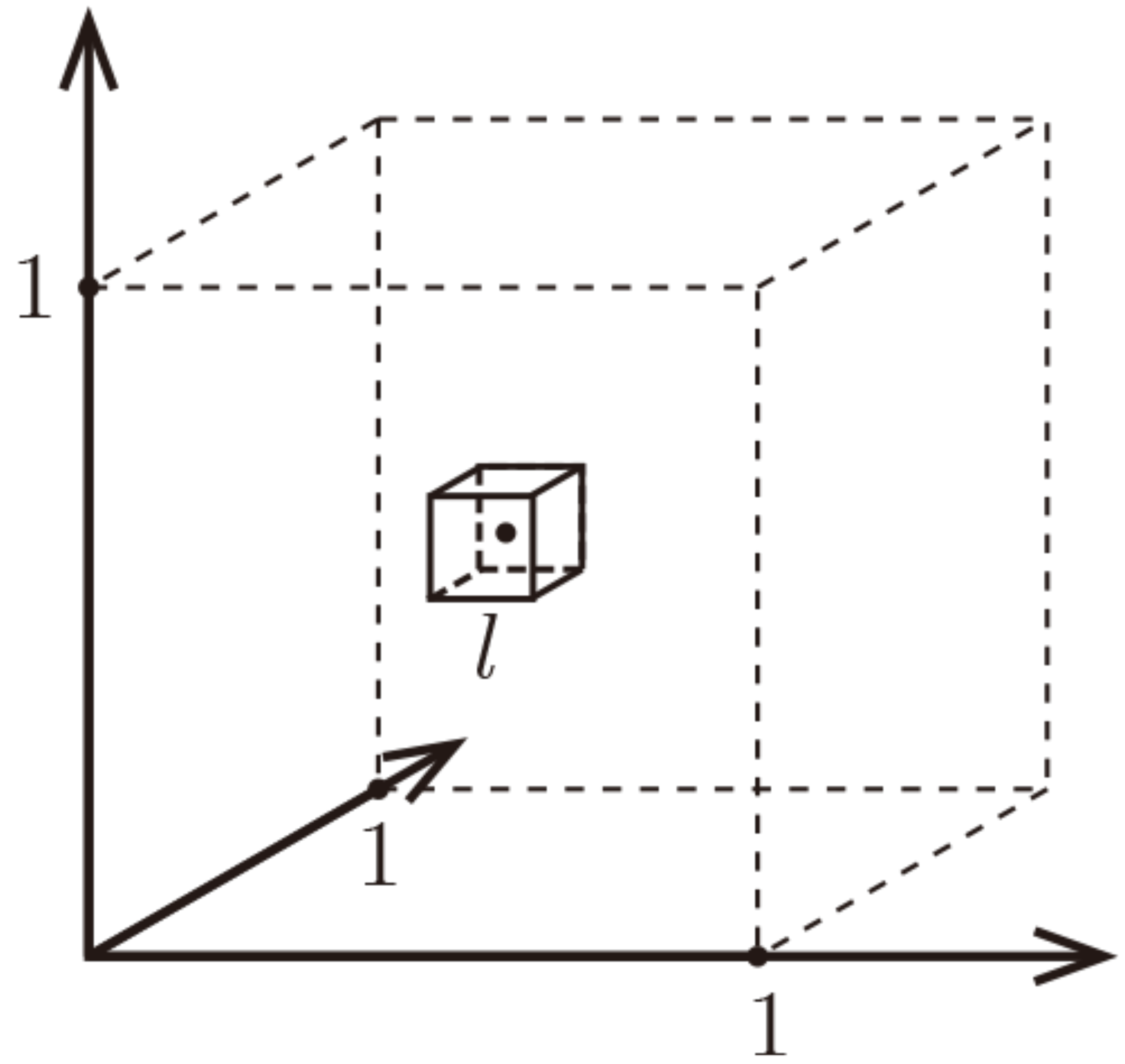
Q: sample x uniformly, what is the probability that x is inside the small cube?

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

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 $[0,1]^d$

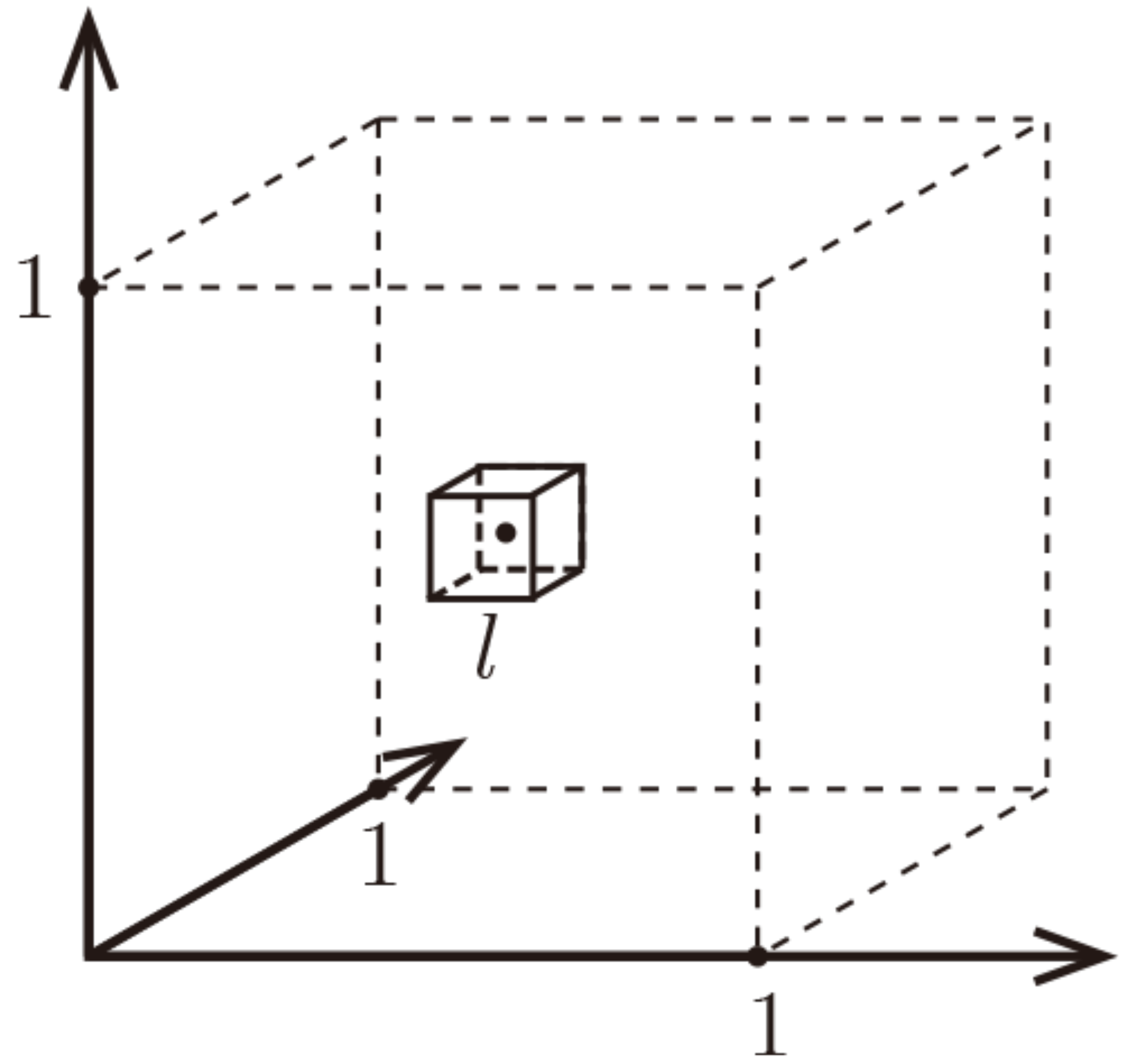


Q: sample x uniformly, what is the probability that x is inside the small cube?

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

Curse of Dimensionality Explanation

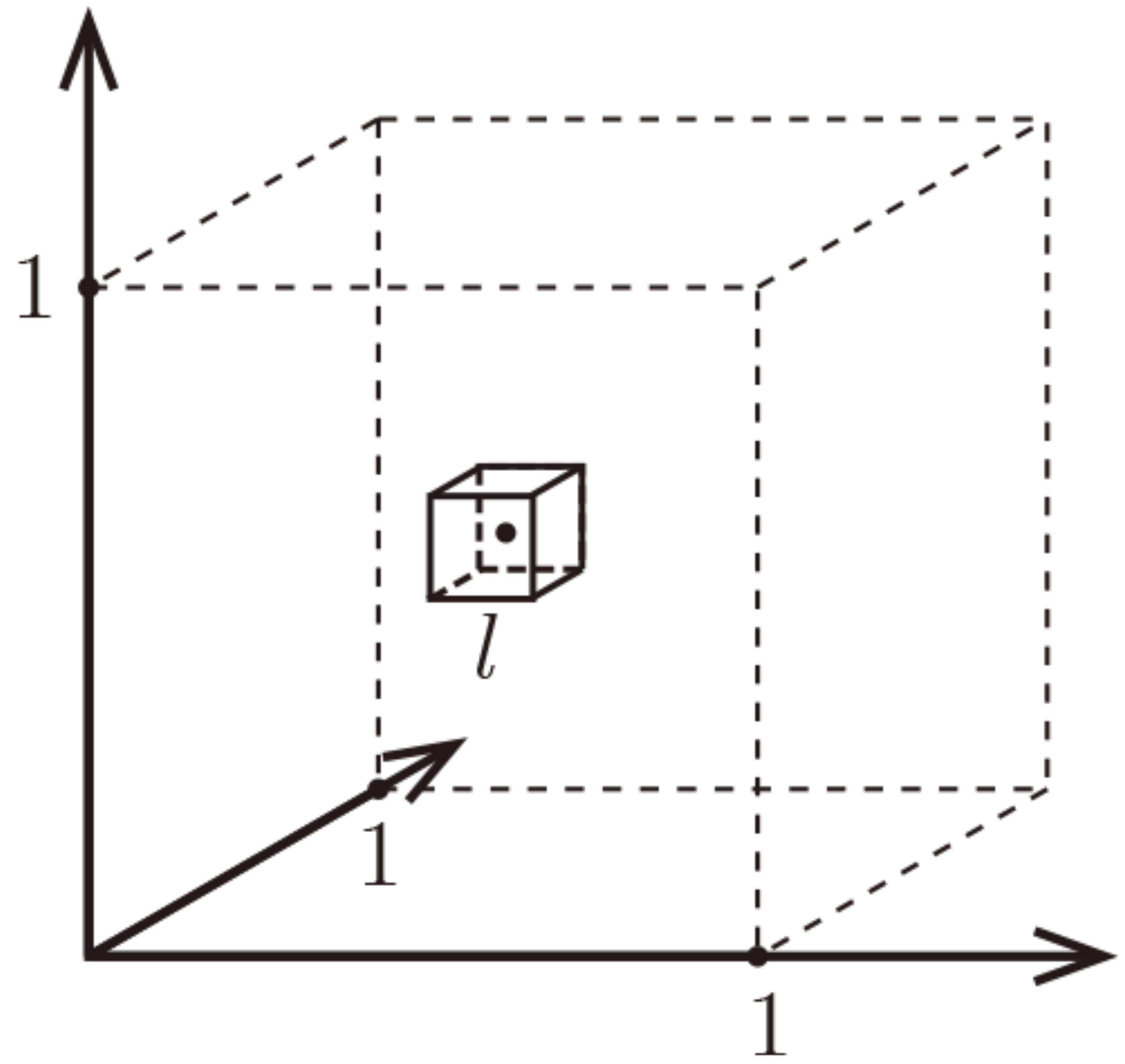
Example: let us consider uniform distribution over a cube $[0,1]^d$



Now assume we sample n points uniform randomly, and we observe K points fall inside the small cube

Curse of Dimensionality Explanation

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

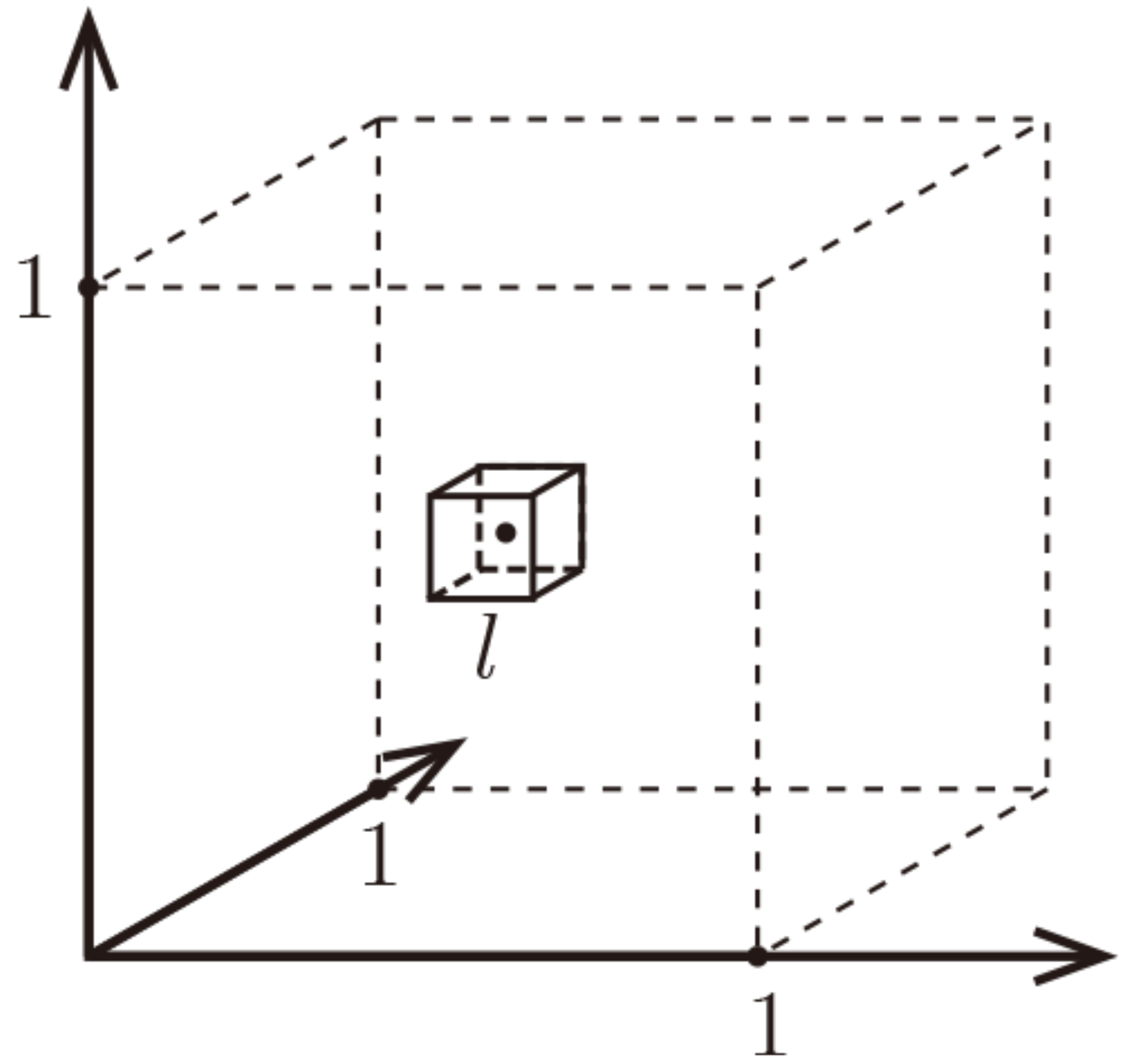


Now assume we sample n points uniform randomly, and we observe K points fall inside the small cube

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

Curse of Dimensionality Explanation

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



Now assume we sample n points uniform randomly, and we observe K points fall inside the small cube

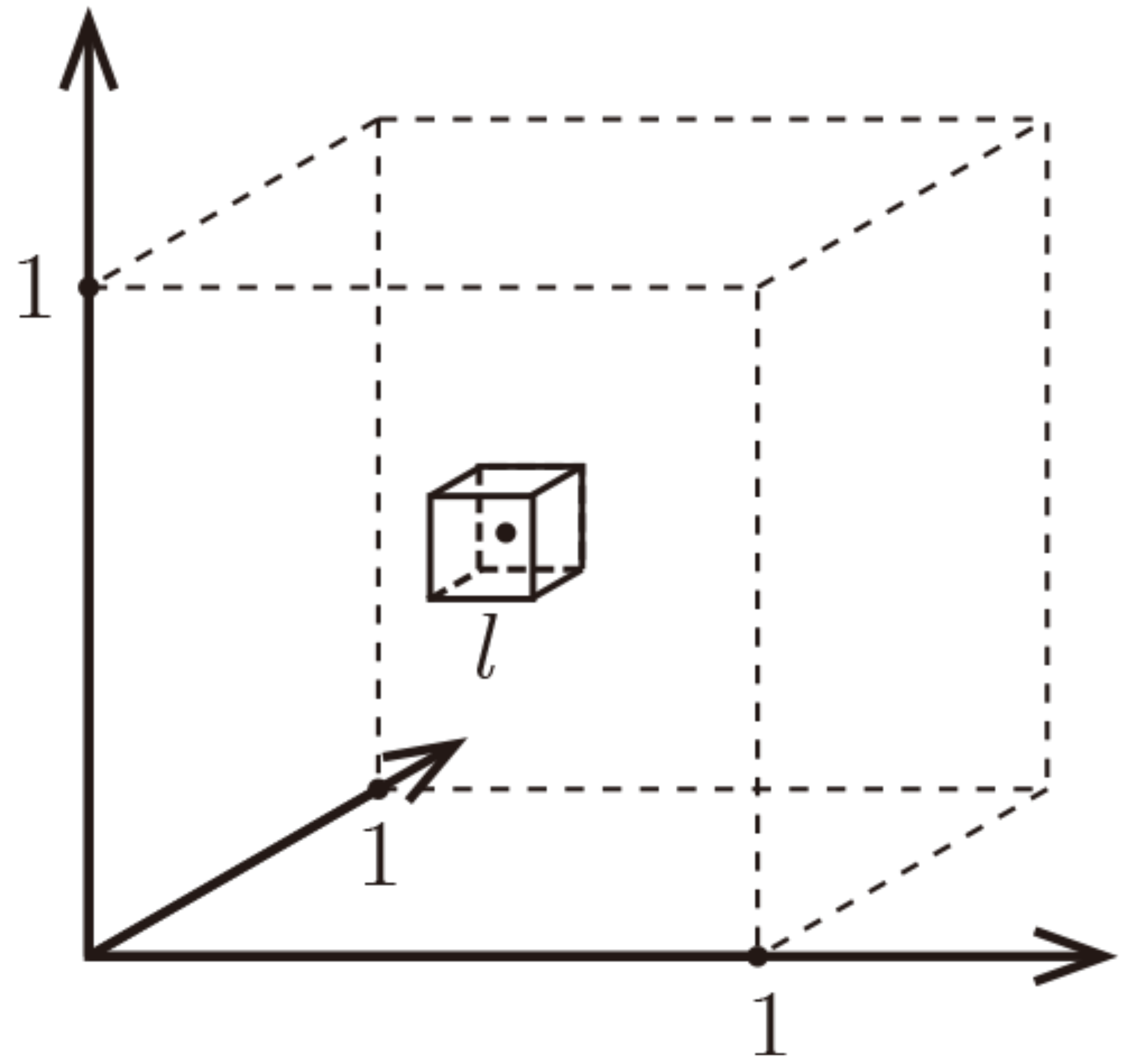
So empirically, the probability of sampling a point inside the small cube is roughly K/n

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

Curse of Dimensionality Explanation

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

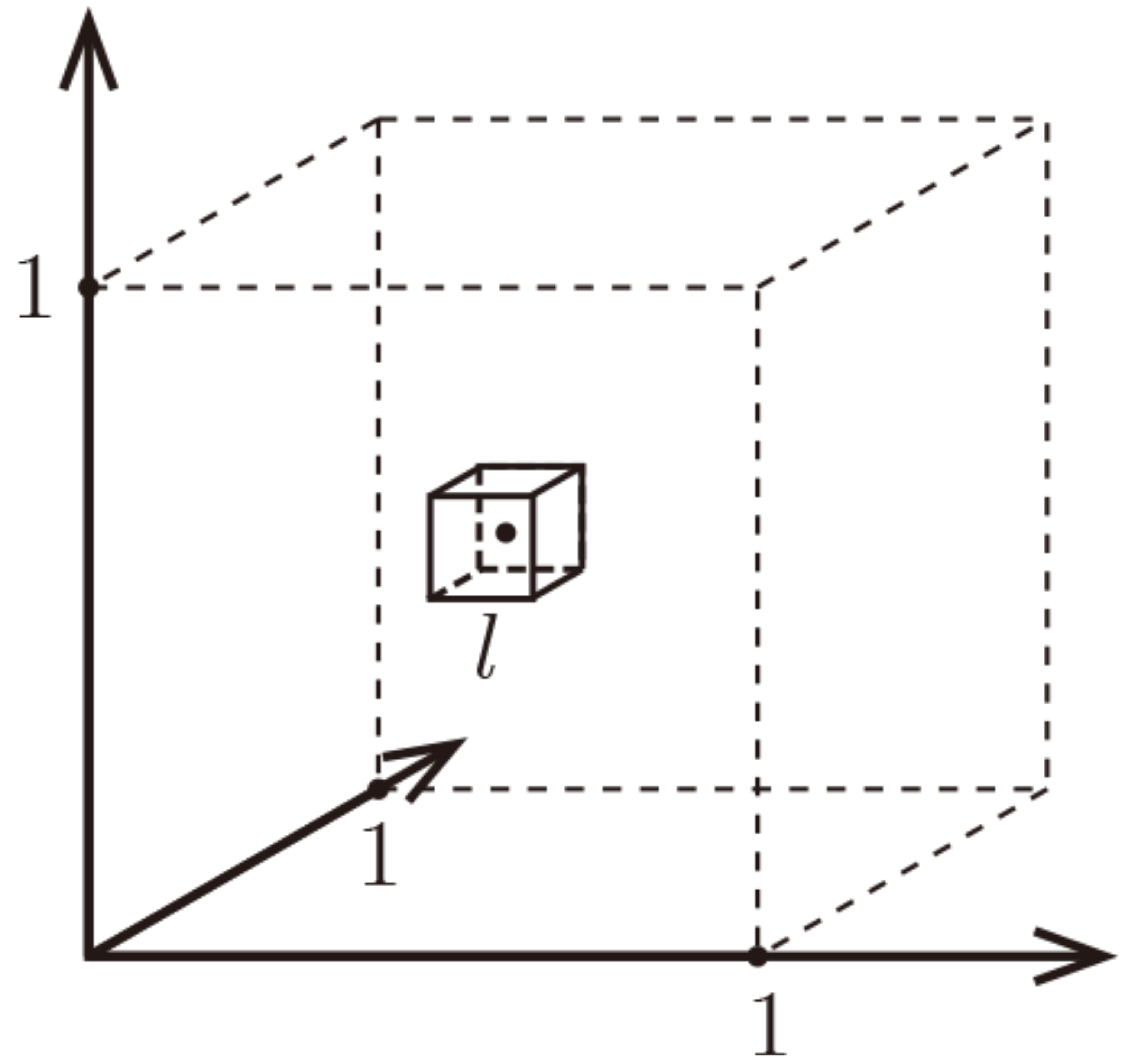
We have $l^d \approx \frac{K}{n}$



Curse of Dimensionality Explanation

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

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

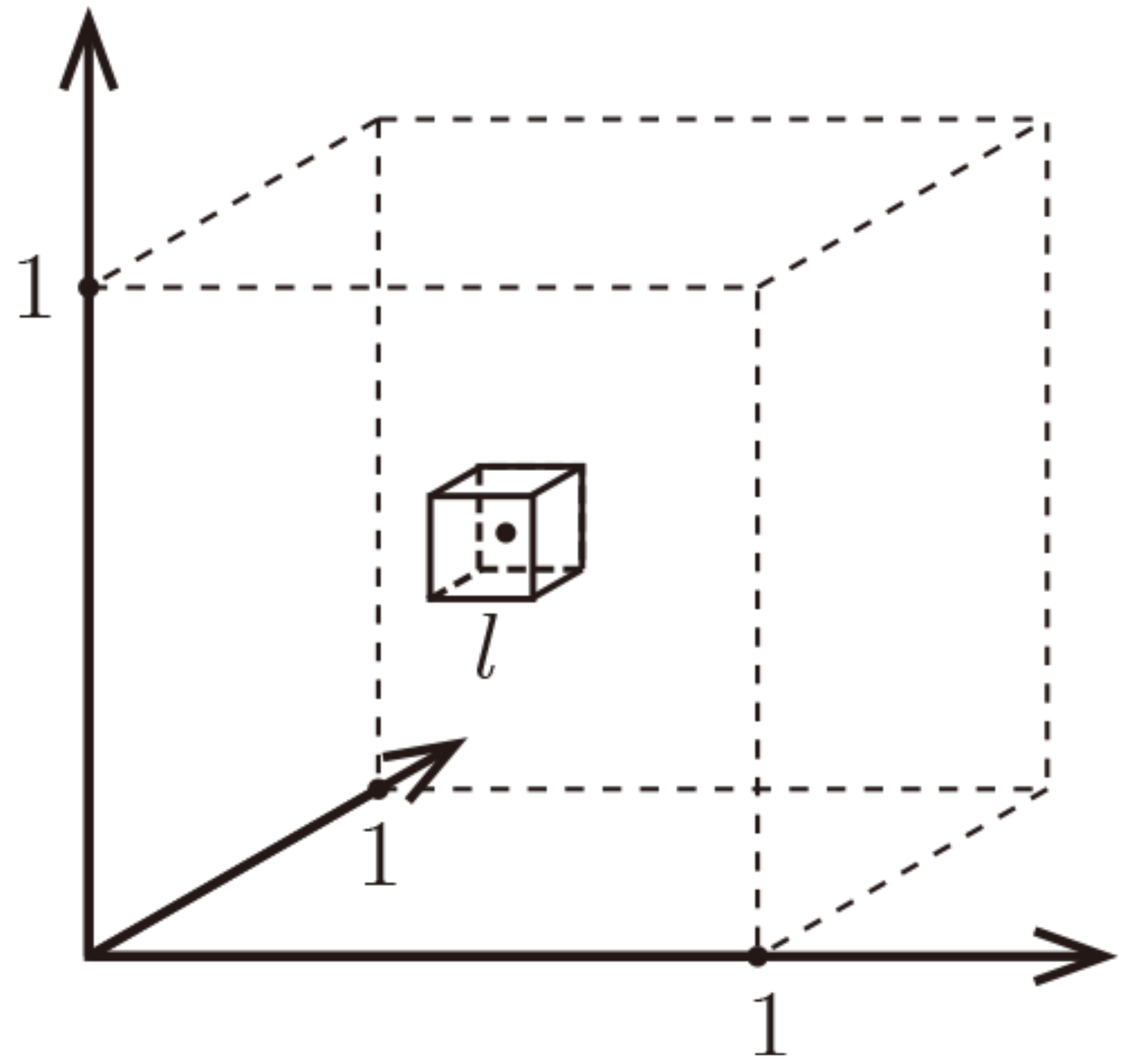


Q: how large we should set l, s.t., we will have K examples (out of n) fall inside the small cube?

Curse of Dimensionality Explanation

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

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



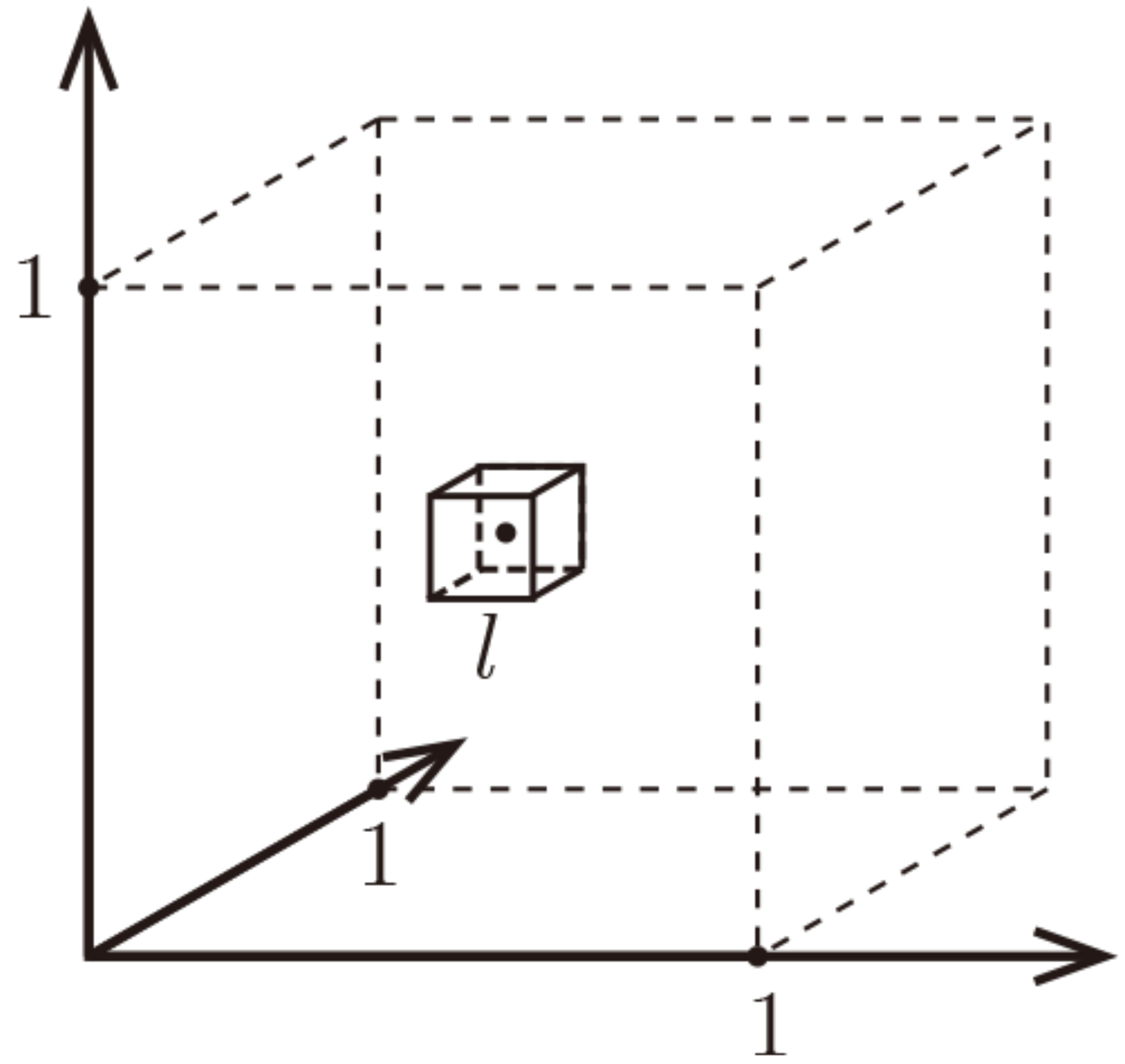
Q: how large we should set l, s.t., we will have K examples (out of n) fall inside the small cube?

$$l \approx (K/n)^{1/d}$$

Curse of Dimensionality Explanation

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

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



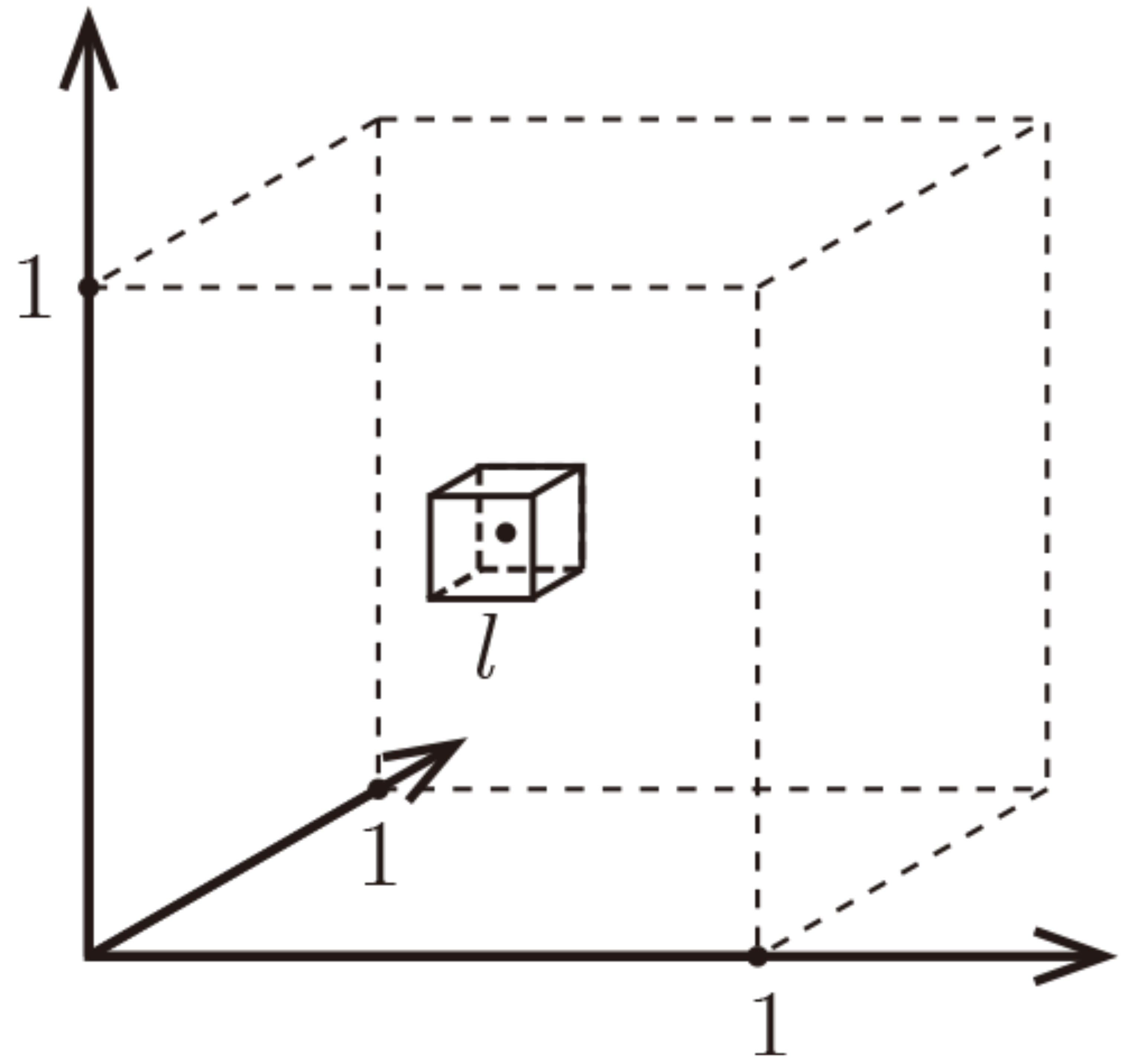
Q: how large we should set l, s.t., we will have K examples (out of n) fall inside the small cube?

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

Curse of Dimensionality Explanation

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

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



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

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

Bad news: when $d \rightarrow \infty$, 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

The distance between two sampled points increases as d grows

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

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

The distance between two sampled points increases as d grows

In $[0,1]^d$, we uniformly sample two points x, x', calculate
$$d(x, x') = \|x - x'\|_2$$

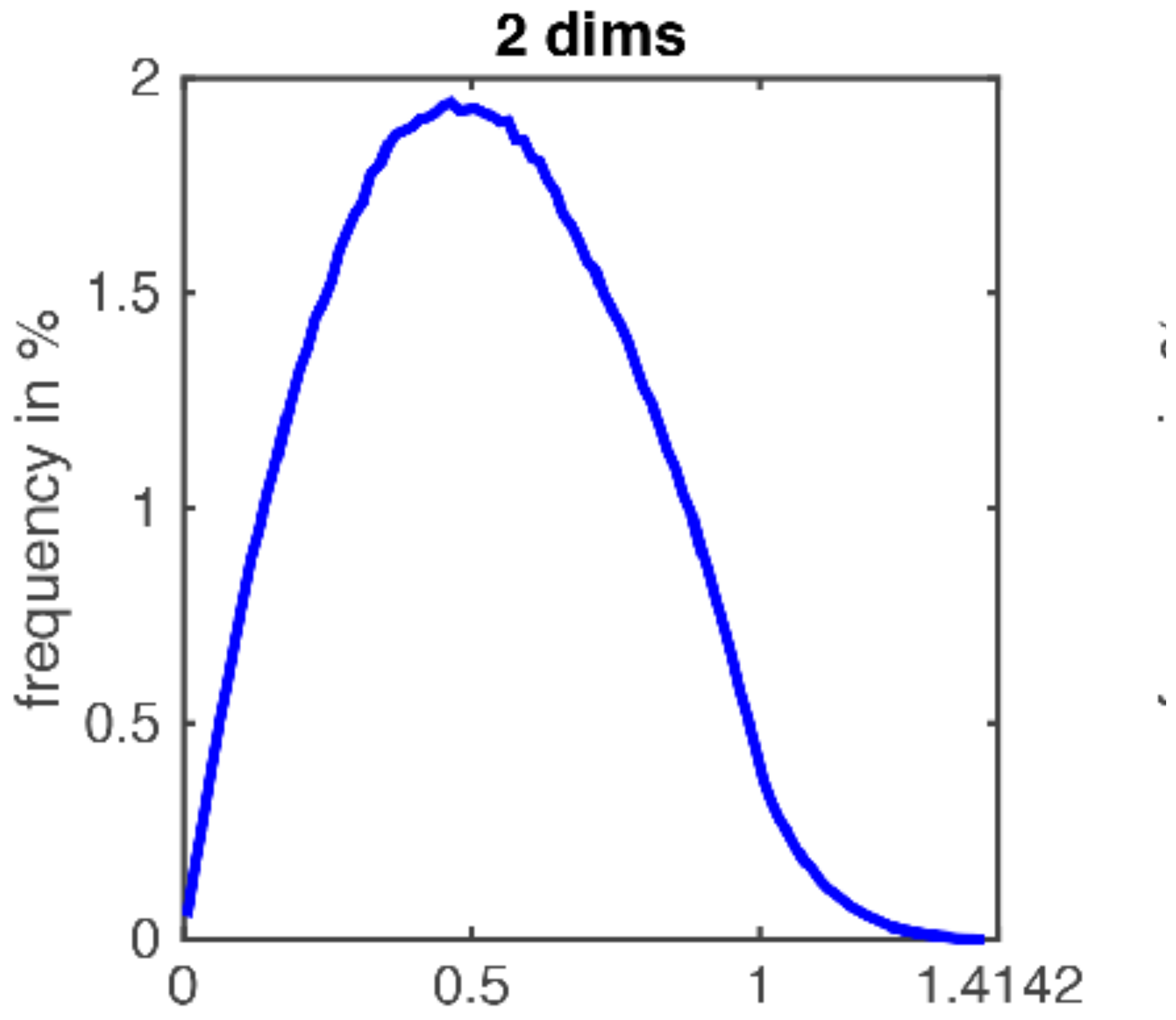
Let's plot the distribution of such distance:

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:

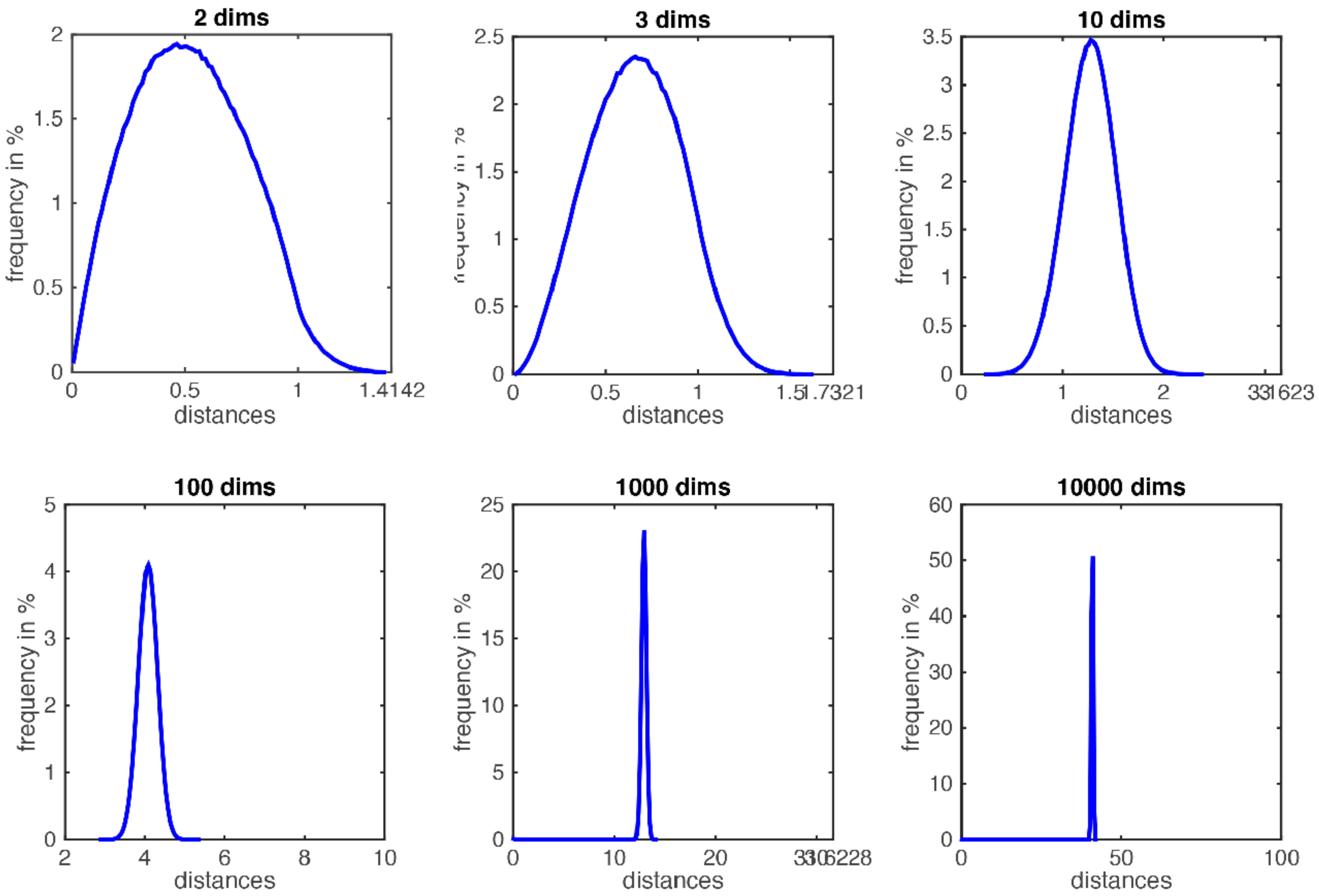


The distance between two sampled points increases as d grows

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:

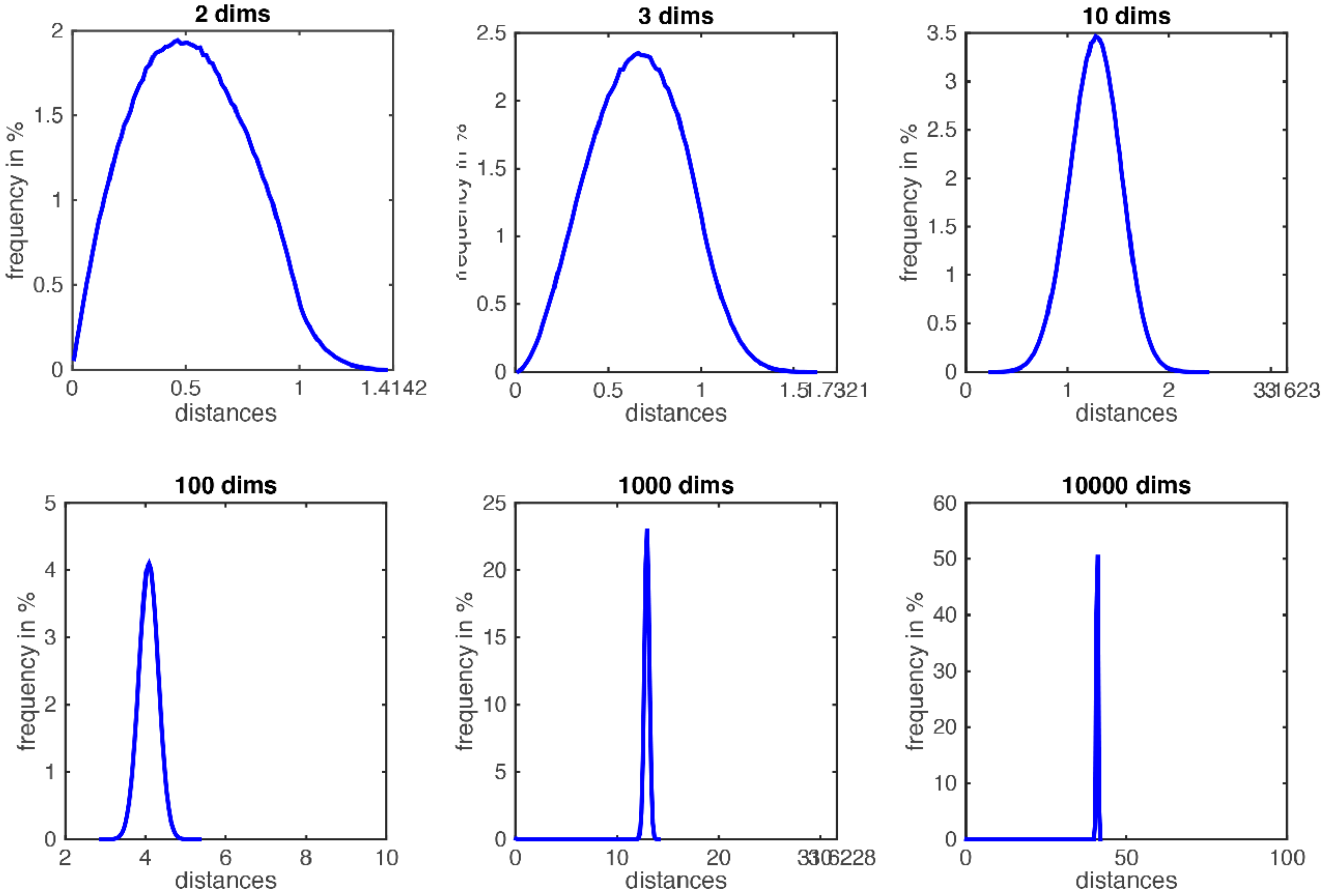


The distance between two sampled points increases as d grows

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

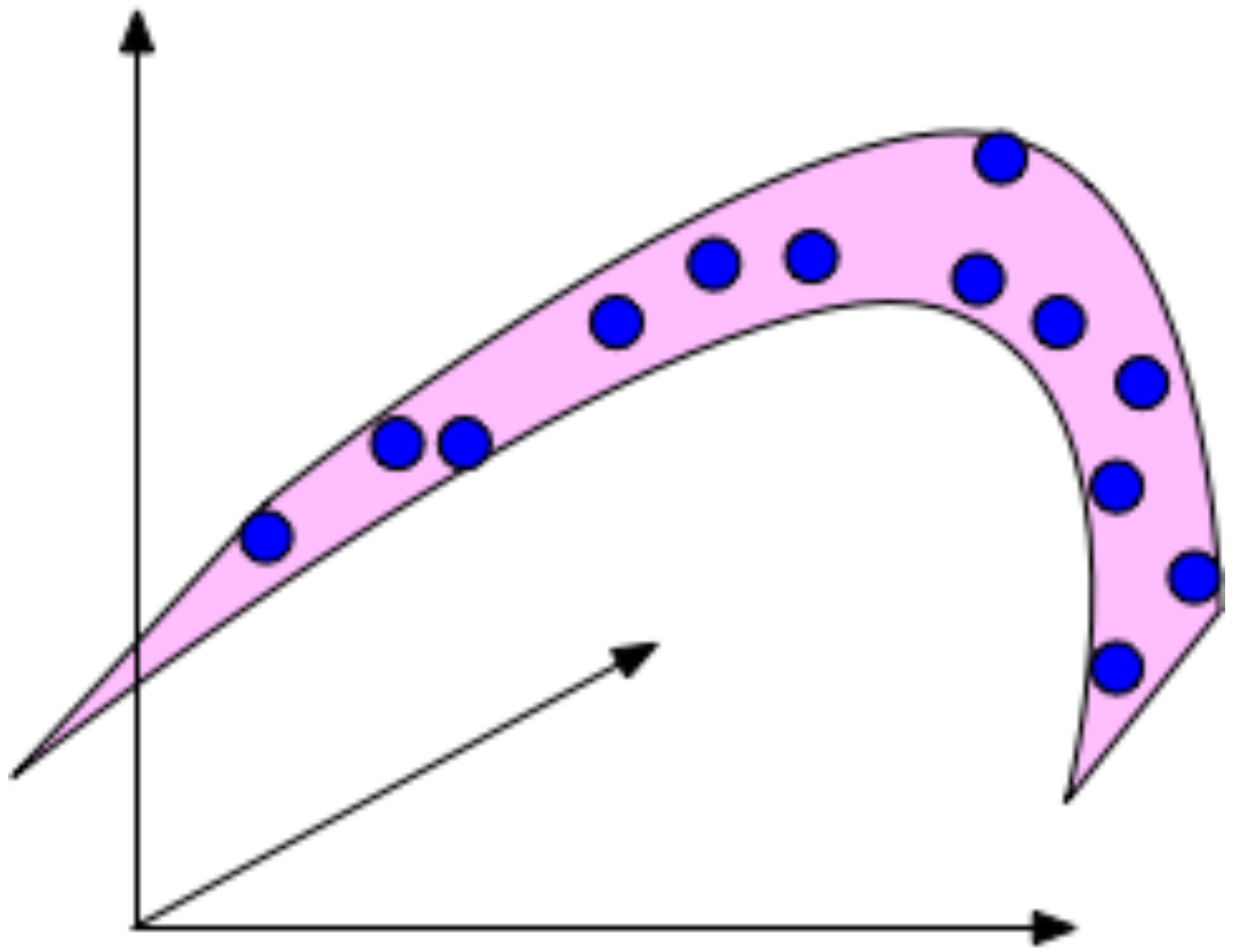
$$d(x, x') = \|x - x'\|_2$$

Let's plot the distribution of such distance:



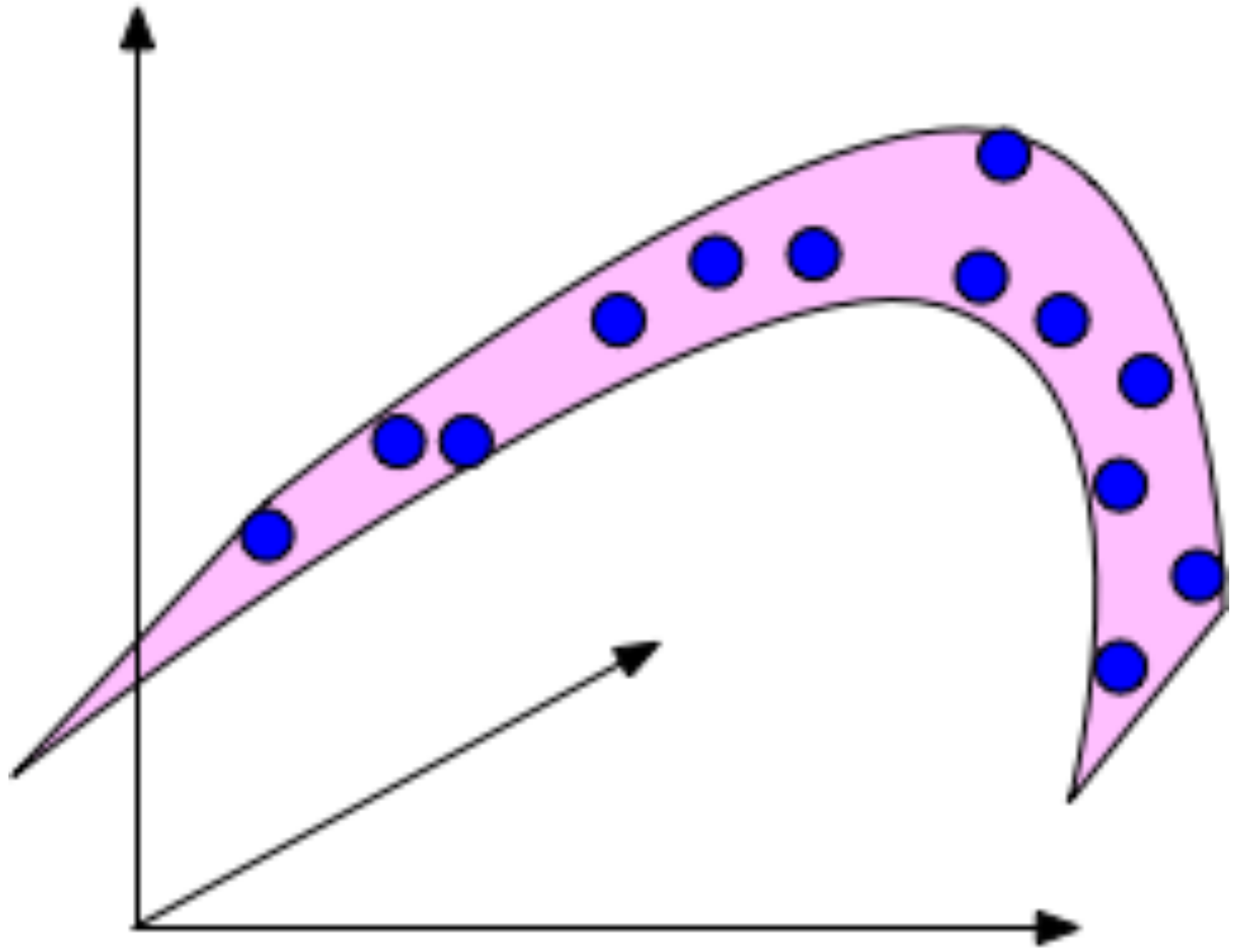
Distance increases as $d \to \infty$

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



Data lives in 2-d manifold

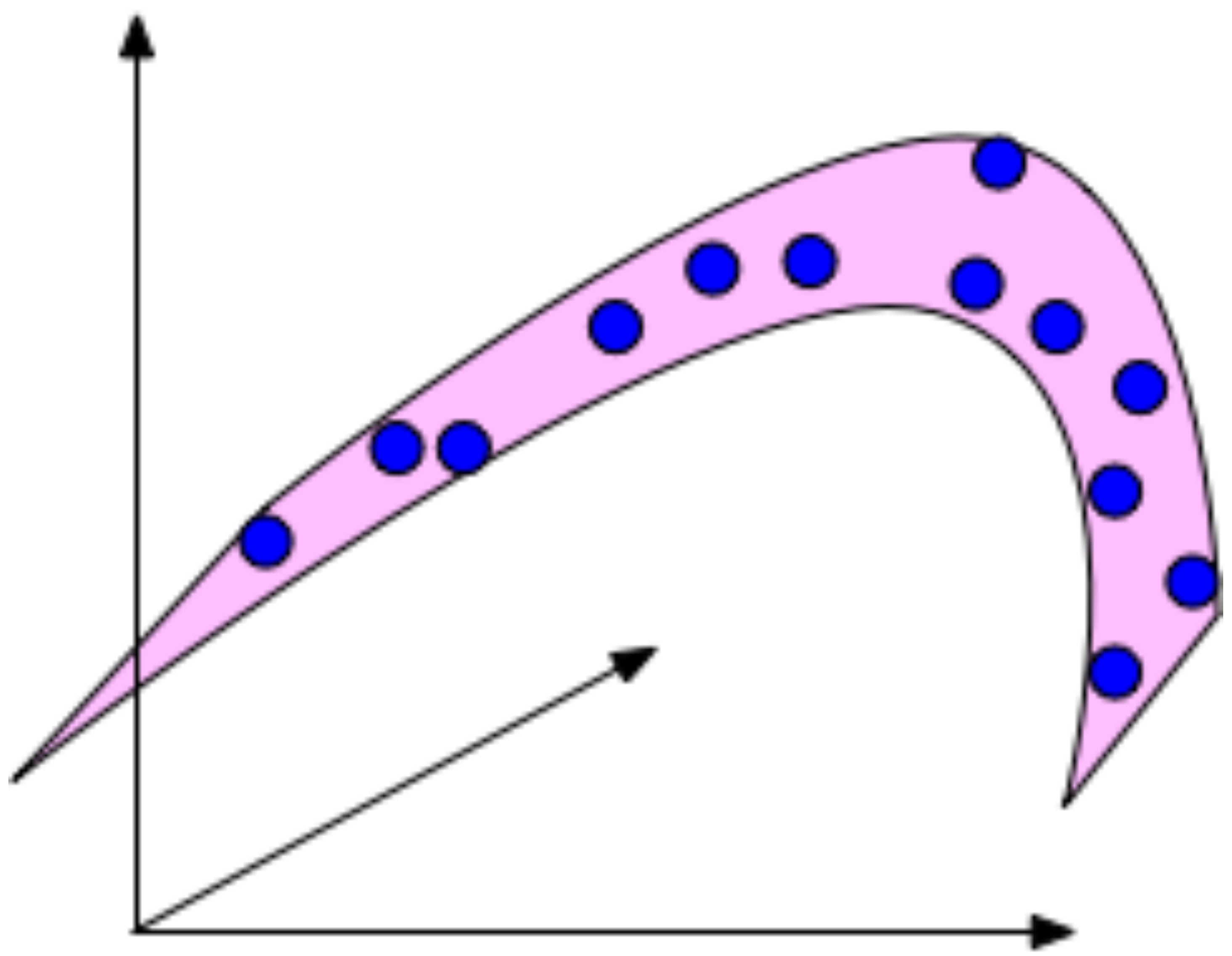
Luckily, real world data often has low-dimensional structure!



Data lives in 2-d manifold

Example: face images

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

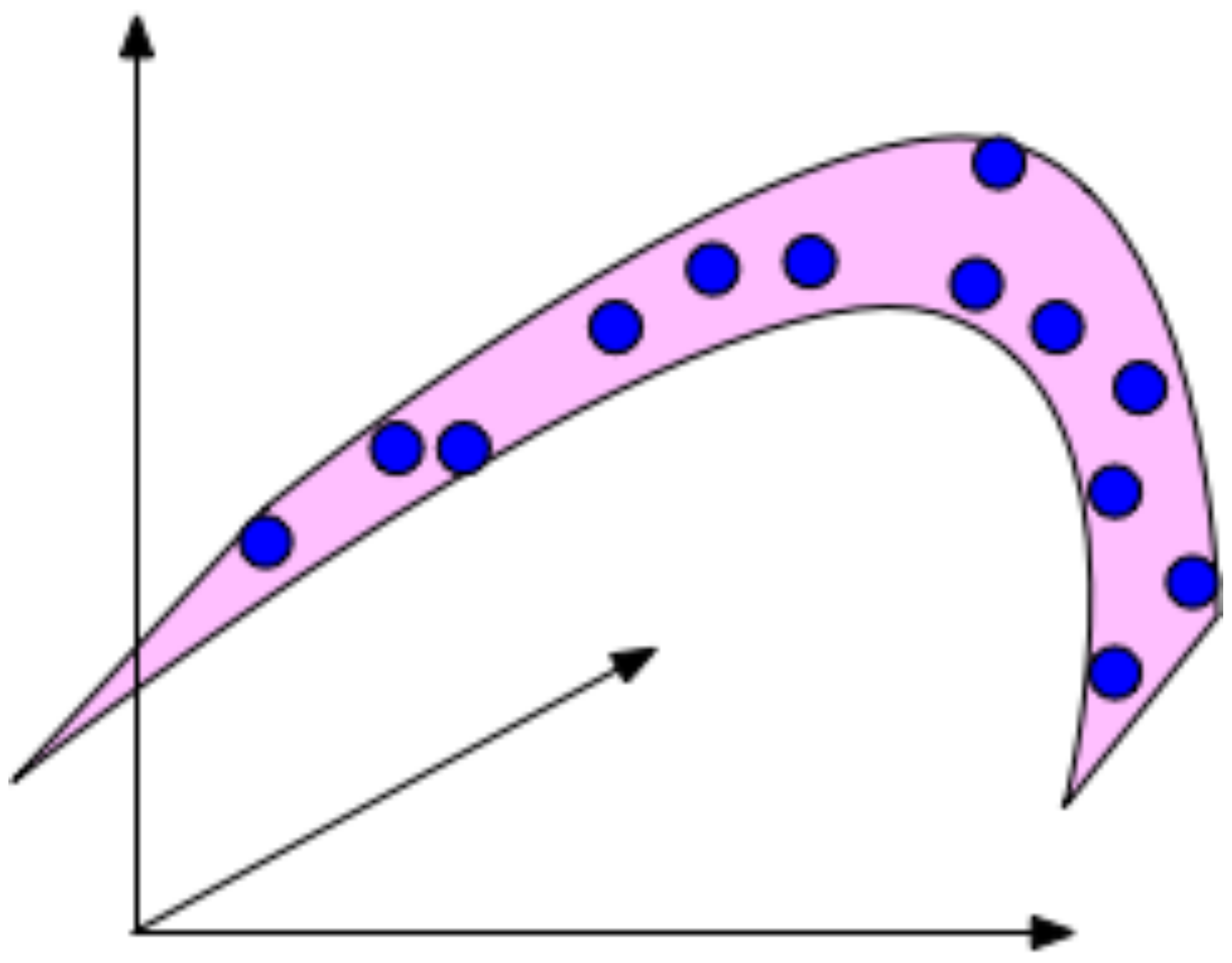


Data lives in 2-d manifold

Example: face images

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

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!)

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)

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