High dimensional computing - the upside of the curse of dimensionality

Peer Neubert
Stefan Schubert
Kenny Schlegel

KI 2019

September 23-26th, 2019, Kassel, Germany

TECHNISCHE UNIVERSITÄT
CHEMNITZ

Topic: (Symbolic) Computation with large vectors

Roughly synonyms:

- High dimensional Computing
- Hyperdimensional Computing
- Hypervectors
- Vector Symbolic Architectures
- Computing with large random vectors
- ...

Topic: (Symbolic) Computation with large vectors

Roughly synonyms:

- High dimensional Computing
- Hyperdimensional Computing
- Hypervectors
- Vector Symbolic Architectures
- Computing with large random vectors
- ...

2D

$$\begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

3D

$$\begin{pmatrix} -3 \\ 2.3 \\ -0.1 \end{pmatrix}$$

$\longrightarrow$

thousands of dimensions

$$\begin{pmatrix} 1.0 \\ 3.9 \\ -0.5 \\ . \\ . \\ . \\ 2.9 \\ -6.0 \\ 9.8 \end{pmatrix}$$

Topic: (Symbolic) Computation with large vectors

Roughly synonyms:

- High dimensional Computing
- Hyperdimensional Computing
- Hypervectors
- Vector Symbolic Architectures
- Computing with large random vectors
- ...

2D

3D

thousands of dimensions

$$\begin{pmatrix} 1 \\ 2 \end{pmatrix} \quad \begin{pmatrix} -3 \\ 2.3 \\ -0.1 \end{pmatrix} \longrightarrow \begin{pmatrix} 1.0 \\ 3.9 \\ -0.5 \\ \cdot \\ \cdot \\ \cdot \\ 2.9 \\ -6.0 \\ 9.8 \end{pmatrix}$$

Pentti Kanerva. 2009. *Hyperdimensional Computing: An Introduction to Computing in Distributed Representation with High-Dimensional Random Vectors*. Cognitive Computation 1, 2 (2009), 139–159. https://doi.org/10.1007/s12559-009-9009-8

Neubert, P., Schubert, S., Protzel, P. 2019. An Introduction to Hyperdimensional Computing for Robotics. KI - Künstliche Intelligenz. https://doi.org/10.1007/s13218-019-00623-z

Reasons to attend

Interest in

- Exploiting the “curse of dimensionality”
- Extending (deep) ANNs with symbolic processing
- Noise robustness (and power efficiency)

Reasons to attend

Interest in

- Exploiting the “curse of dimensionality”
- Extending (deep) ANNs with symbolic processing
- Noise robustness (and power efficiency)

Related Fields

- Information retrieval
- Vector models for NLP
- Robotics
- Quantum cognition/probability/logic
- ...

Reasons to attend

Interest in

- Exploiting the “curse of dimensionality”
- Extending (deep) ANNs with symbolic processing
- Noise robustness (and power efficiency)

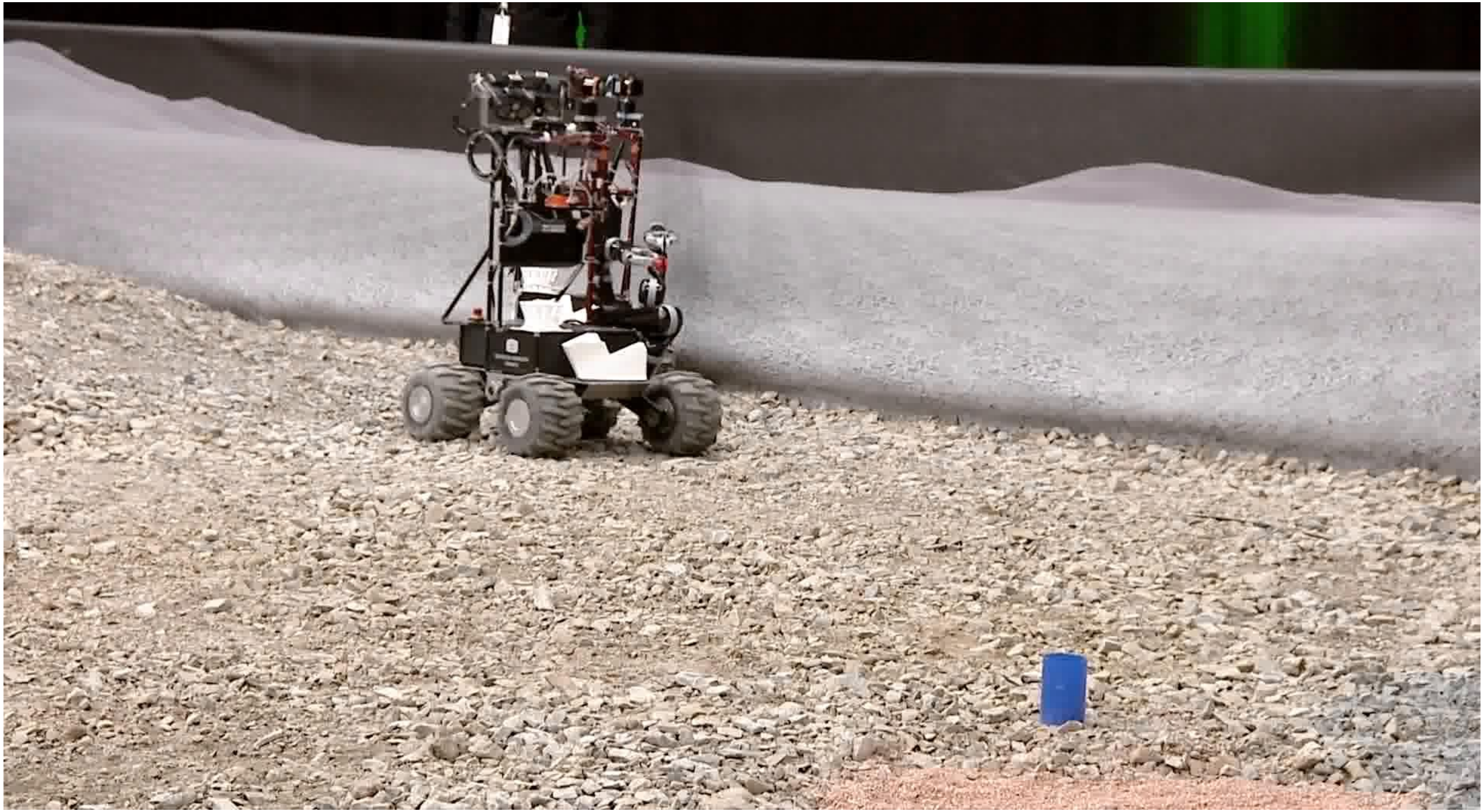
Related Fields

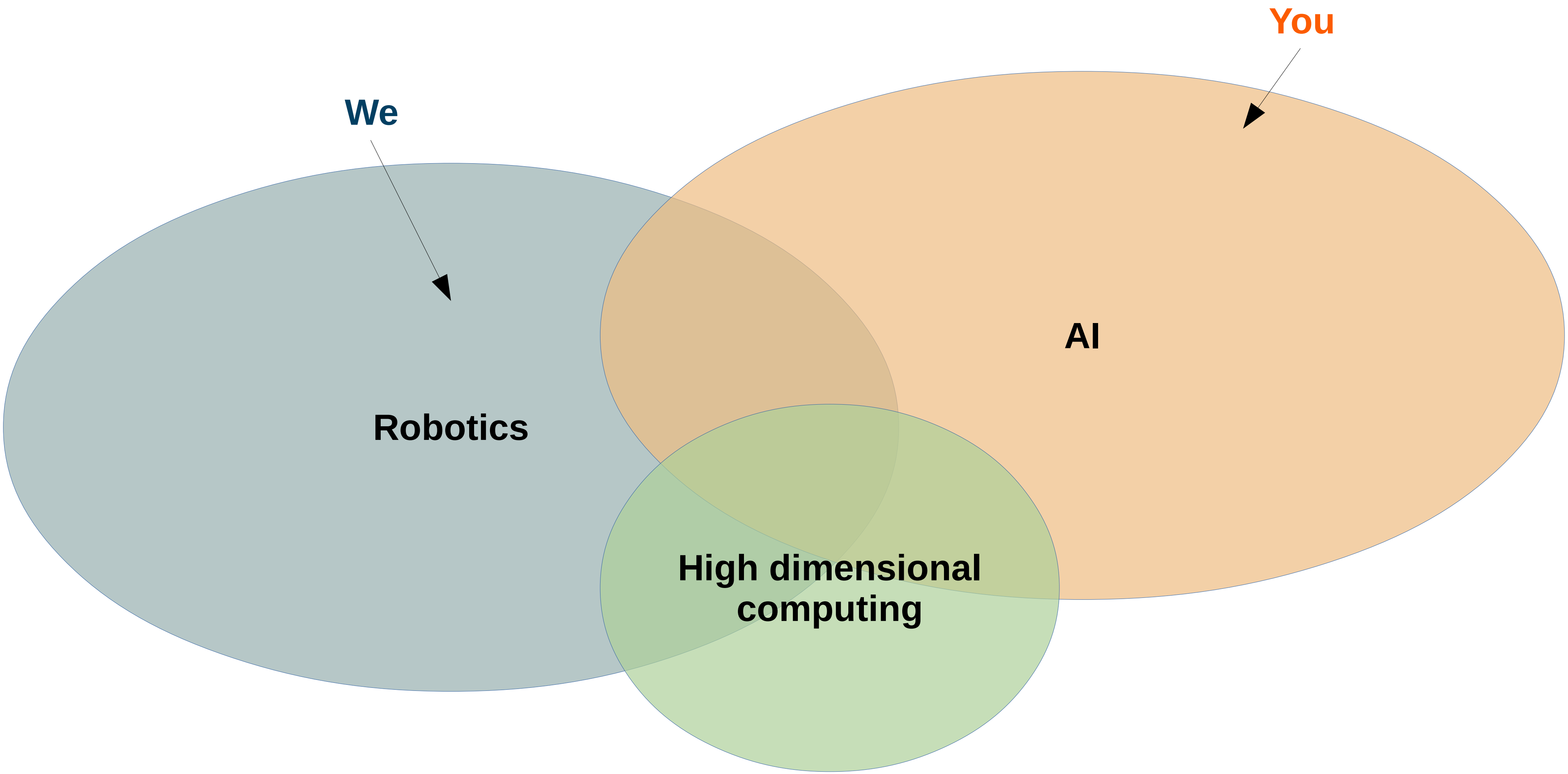
- Information retrieval
- Vector models for NLP
- Robotics
- Quantum cognition/probability/logic
- ...

Goals

- Introduction to the topic
- Intuition towards underlying mathematical properties
- Link to available approaches and implementations
- Outline potential applications
- Provide some first hands-on experience

What we are doing





- Our background is neither classic AI nor mathematics
- We are very much interested in any thoughts and feedback!

Outline

14:00 Welcome

14:05 Introduction to high dimensional computing

15:05 Implementations in form of Vector Symbolic Architectures

15:30 Coffee break

16:00 Vector encodings of real world data

16:30 Applications

17:15 Discussion and conclusion

Teaser application 1: “What is the Dollar of Mexico?”

Credits: Pentti Kanerva

Teaser application 1: “What is the Dollar of Mexico?”

Credits: Pentti Kanerva

Given are 2 records:

United States of America

Name:	USA
Capital City:	Washington DC
Currency:	Dollar

Mexico

Name:	Mexico
Capital City:	Mexico City
Currency:	Peso

Teaser application 1: “What is the Dollar of Mexico?”

Credits: Pentti Kanerva

Given are 2 records:

United States of America

Name: USA
Capital City: Washington DC
Currency: Dollar

Mexico

Name: Mexico
Capital City: Mexico City
Currency: Peso

Question: What is the Dollar of Mexico?

Teaser application 1: “What is the Dollar of Mexico?”

Credits: Pentti Kanerva

Given are 2 records:

United States of America

Name:	USA
Capital City:	Washington DC
Currency:	Dollar

Mexico

Name:	Mexico
Capital City:	Mexico City
Currency:	Peso

Question: What is the Dollar of Mexico?

Hyperdimensional computing approach:

1. Assign a **random** high-dimensional vector to each entity
   ”Name” is a random vector NAM
   ”USA” is a random vector USA
   ”Capital city” is a random vector CAP
   …

Teaser application 1: “What is the Dollar of Mexico?”

Credits: Pentti Kanerva

Given are 2 records:

United States of America

Name:	USA
Capital City:	Washington DC
Currency:	Dollar

Mexico

Name:	Mexico
Capital City:	Mexico City
Currency:	Peso

Question: What is the Dollar of Mexico?

Hyperdimensional computing approach:

1. Assign a **random** high-dimensional vector to each entity
   ”Name” is a random vector NAM
   ”USA” is a random vector USA
   ”Capital city” is a random vector CAP
   …

2. Calculate a **single** high-dimensional vector that contains all information
   F = (NAM*USA+CAP*WDC+CUR*DOL)*(NAM*MEX+CAP*MCX+CUR*PES)

Teaser application 1: "What is the Dollar of Mexico?"

Credits: Pentti Kanerva

Given are 2 records:

United States of America

Name:	USA
Capital City:	Washington DC
Currency:	Dollar

Mexico

Name:	Mexico
Capital City:	Mexico City
Currency:	Peso

Question: What is the Dollar of Mexico?

Hyperdimensional computing approach:

1. Assign a **random** high-dimensional vector to each entity
   "Name" is a random vector NAM
   "USA" is a random vector USA
   "Capital city" is a random vector CAP
   …

2. Calculate a **single** high-dimensional vector that contains all information
   F = (NAM*USA+CAP*WDC+CUR*DOL)*(NAM*MEX+CAP*MCX+CUR*PES)

3. **Calculate** the query answer: F*DOL ~ PES

Teaser application 2: Place recognition in changing environments

Problem: Visual place recognition

Image credits: M. Milford and G. F. Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2012.

Teaser application 2: Place recognition in changing environments

Problem: Visual place recognition in changing environments

Image credits: M. Milford and G. F. Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2012.

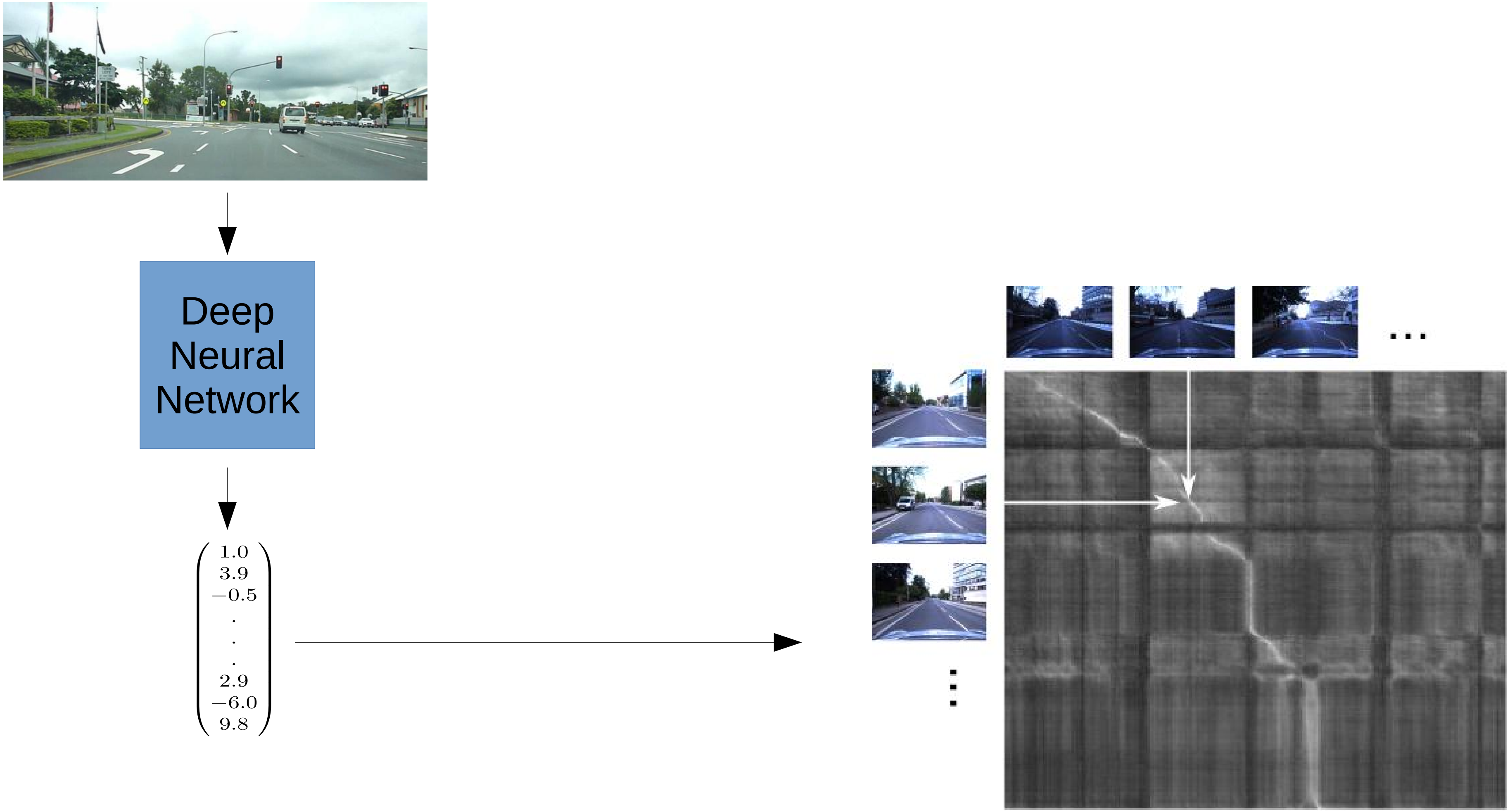
Teaser application 2: Place recognition in changing environments

Teaser application 2: Place recognition in changing environments

Deep Neural Network

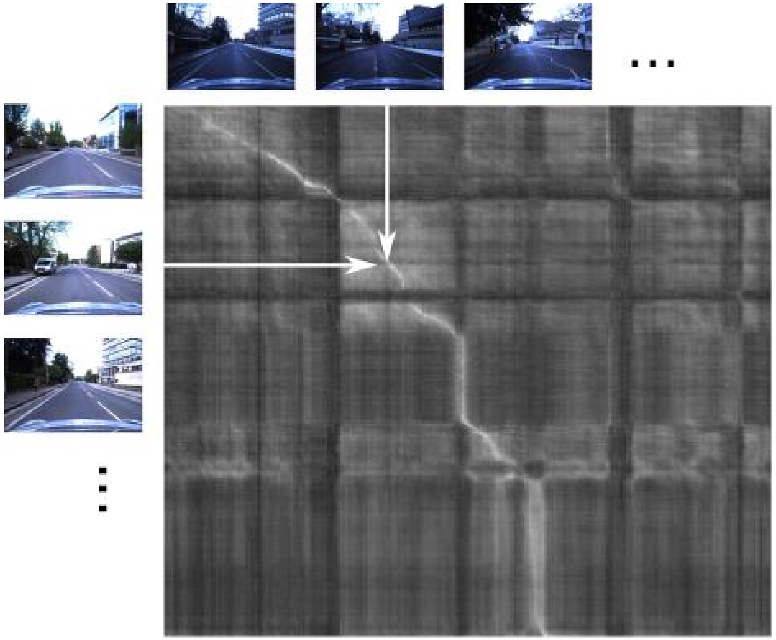
$$\begin{pmatrix} 1.0 \\ 3.9 \\ -0.5 \\ . \\ . \\ . \\ 2.9 \\ -6.0 \\ 9.8 \end{pmatrix}$$

Teaser application 2: Place recognition in changing environments

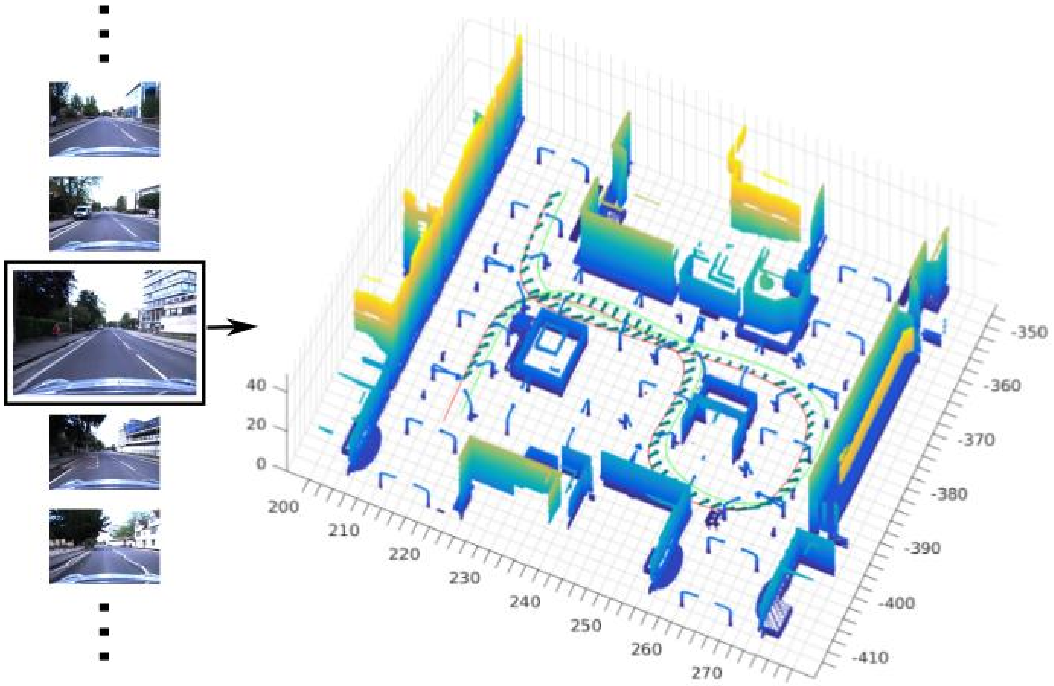


Approaches to place recognition

Pairwise image comparison

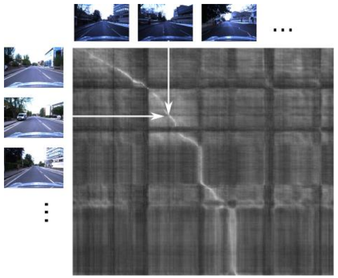


Metric SLAM

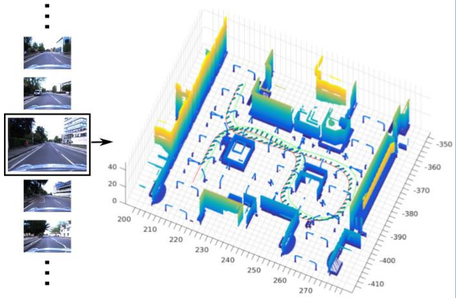


Approaches to place recognition

Pairwise comparison

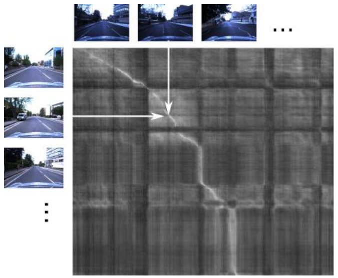


Metric SLAM

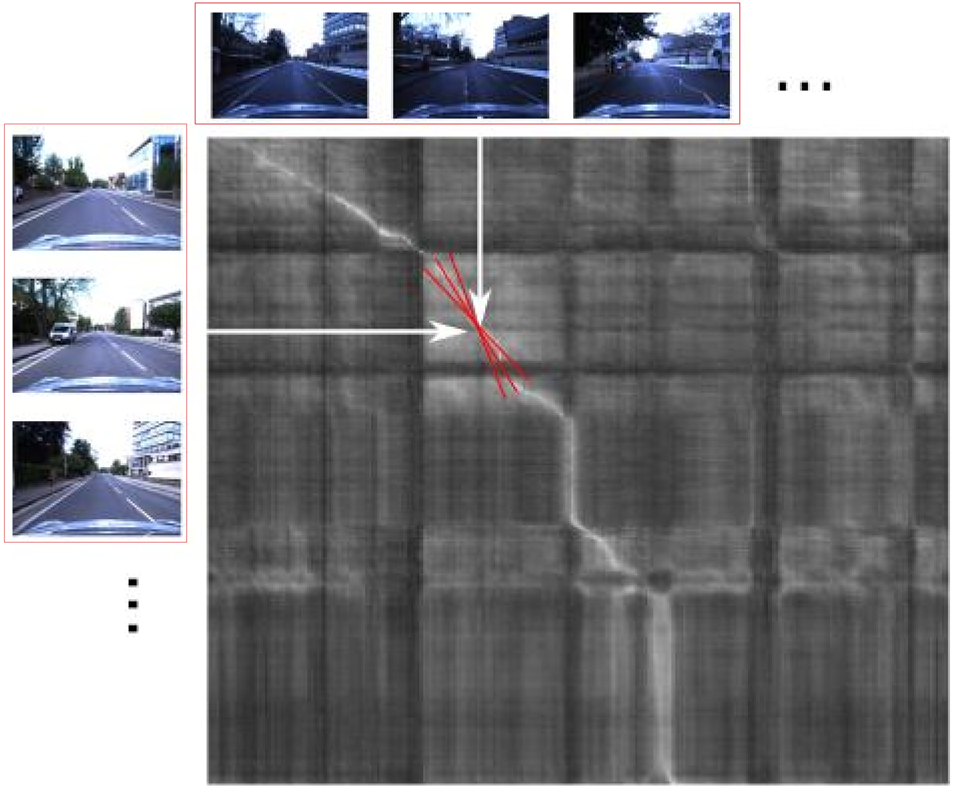


Approaches to place recognition

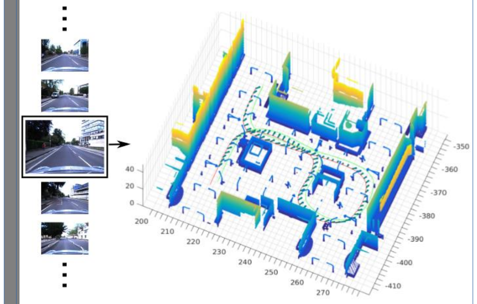
Pairwise independently



In between, e.g. **SeqSLAM**



Metric SLAM



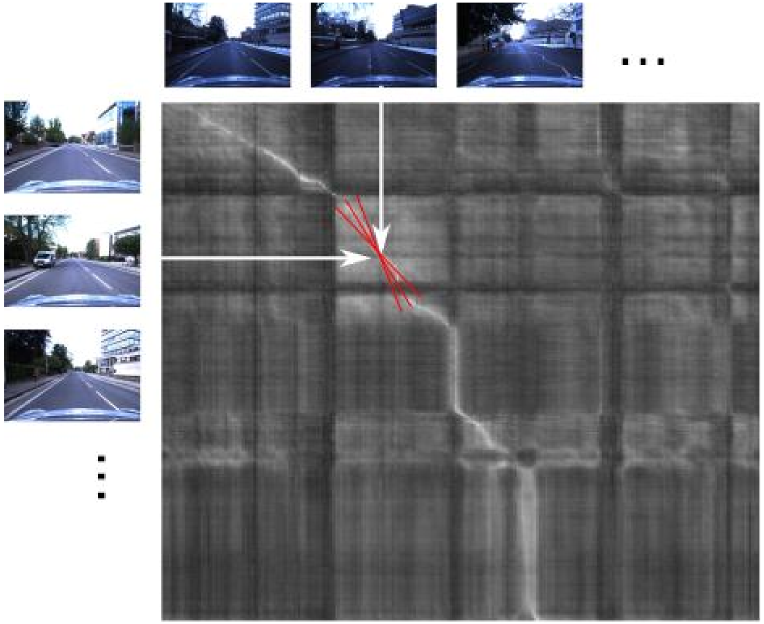
Teaser application 2: Place recognition in changing environments

Deep Neural Network

$$\begin{pmatrix} 1.0 \\ 3.9 \\ -0.5 \\ \cdot \\ \cdot \\ \cdot \\ 2.9 \\ -6.0 \\ 9.8 \end{pmatrix}$$

This is where the hyperdimensional magic happens

$$Y_i = \mathop{\LARGE +}_{k=0}^{d} (X_{i-k} \otimes P_{-k})$$



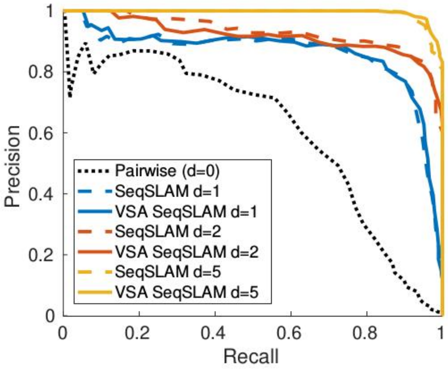
Teaser application 2: Place recognition in changing environments

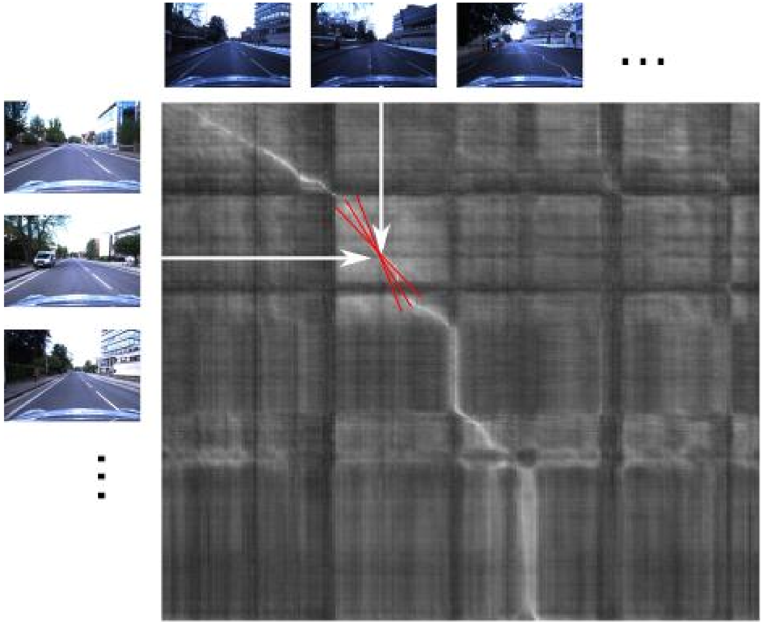
Deep Neural Network

$$\begin{pmatrix} 1.0 \\ 3.9 \\ -0.5 \\ \cdot \\ \cdot \\ \cdot \\ 2.9 \\ -6.0 \\ 9.8 \end{pmatrix}$$

This is where the hyperdimensional magic happens

$$Y_i = \mathop{\boldsymbol{+}}_{k=0}^{d} (X_{i-k} \otimes P_{-k})$$





Outline: Introduction to high dimensional computing

1) Historical note

2) High dimensional vector spaces and where they are used

3) Mathematical properties of high dimensional vector spaces

4) Vector Symbolic Architectures or “How to do symbolic computations using vectors spaces”

    including “What is the Dollar of Mexico?”

Historical note

See: “Geometry and Meaning” by Dominic Widdows 2004, CSLI Publications, Stanford, ISBN 9781575864488

- Ancient Greeks: Roots of geometry
  - Plato: geometric theory of creation and elements
  - Journey and work of Aristotle
  - Euclid: “Elements of geometry”

Historical note

See: “Geometry and Meaning” by Dominic Widdows 2004, CSLI Publications, Stanford, ISBN 9781575864488

- Ancient Greeks: Roots of geometry
  - Plato: geometric theory of creation and elements
  - Journey and work of Aristotle
  - Euclid: “Elements of geometry”
- Modern scientific progress: Geometry and vectors
  - 1637 Descartes “Analytic Geometry”
  - 1844 Graßmann and 1853 Hamilton introduce vectors
  - 1936 Birkhoff and von Neumann introduce quantum logic

Historical note

See: “Geometry and Meaning” by Dominic Widdows 2004, CSLI Publications, Stanford, ISBN 9781575864488

- Ancient Greeks: Roots of geometry
  - Plato: geometric theory of creation and elements
  - Journey and work of Aristotle
  - Euclid: “Elements of geometry”

- Modern scientific progress: Geometry and vectors
  - 1637 Descartes “Analytic Geometry”
  - 1844 Graßmann and 1853 Hamilton introduce vectors
  - 1936 Birkhoff and von Neumann introduce quantum logic

- More recently: Hyperdimensional Computing
  - Kanerva: Sparse Distributed Memory, Computing with large random vectors
  - Smolensky, Plate, Gaylor: Vector Symbolic Architectures
  - Fields: Vector models for NLP, Quantum cognition, ...

Vector space

- e.g., n-dimensional real valued vectors

$$x \in \mathbb{R}^n$$

- Intuitive meaning in 1 to 3 dimensional Euclidean spaces

  e.g., position of a book in a rack

$$(x_1)$$

Vector space

- e.g., n-dimensional real valued vectors

$$x \in \mathbb{R}^n$$

- Intuitive meaning in 1 to 3 dimensional Euclidean spaces
  - e.g., position of a book in a rack or **bookshelf**

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

Vector space

- e.g., n-dimensional real valued vectors

$$x \in \mathbb{R}^n$$

- Intuitive meaning in 1 to 3 dimensional Euclidean spaces
  - e.g., position of a book in a rack, bookshelf or **library**

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

Image: Ralf Roletschek / Roletschek.at. Science library of Upper Lusatia in Görlitz, Germany.

Vector space

- e.g., n-dimensional real valued vectors

$$x \in \mathbb{R}^n$$

- Intuitive meaning in 1 to 3 dimensional Euclidean spaces
  - e.g., position of a book in a rack, bookshelf or library

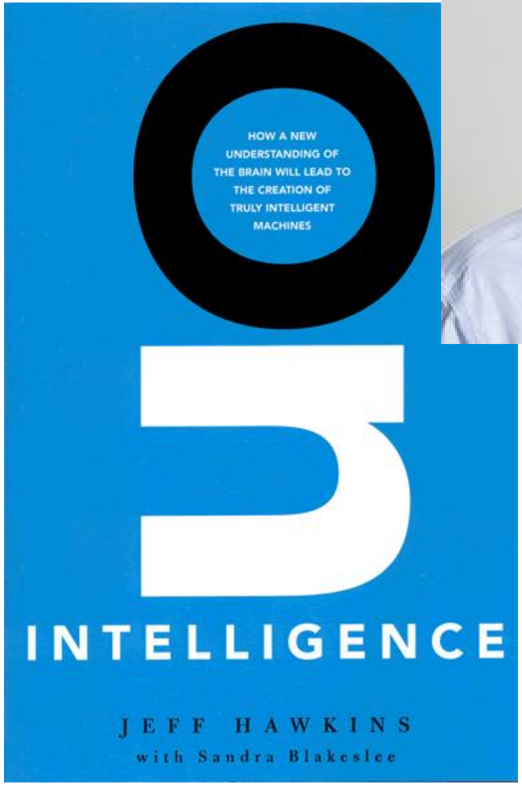
?

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_n \end{pmatrix}$$

Where are such vectors used?

- Feature vectors, e.g., in computer vision or information retrieval
- (Intermediate) representations in deep ANN
- Vector models for natural language processing
- Memory and storage models, e.g., Pentti Kanerva’s Sparse Distributed Memory or Deepmind’s long-short term memory
- Computational brain models, e.g. Jeff Hawkins’ HTM or Chris Eliasmith’s SPAUN
- Quantum cognition approaches
- ...

Hierarchical Temporal Memory

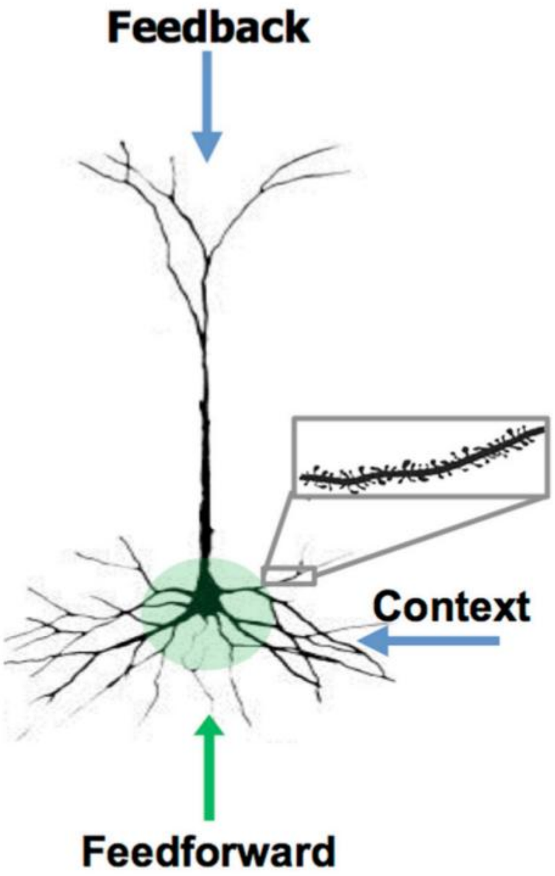


Jeff
Hawkins

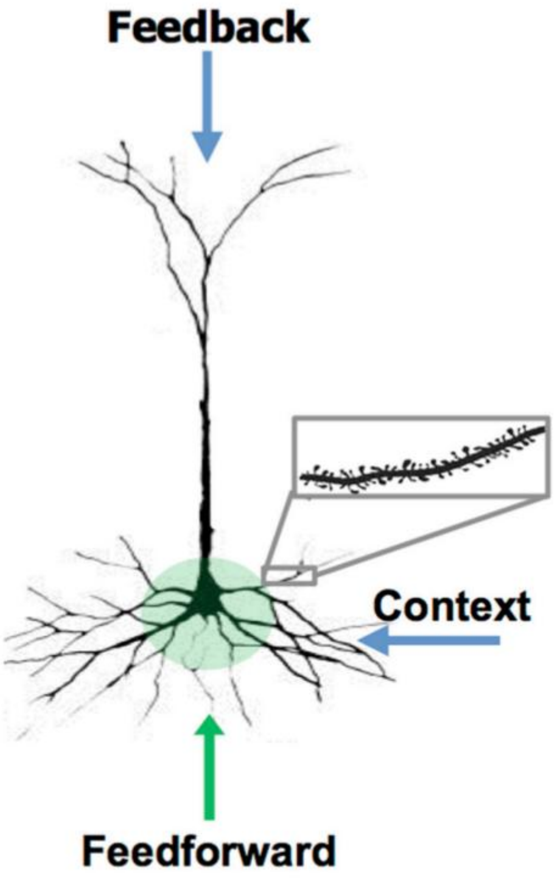
Numenta

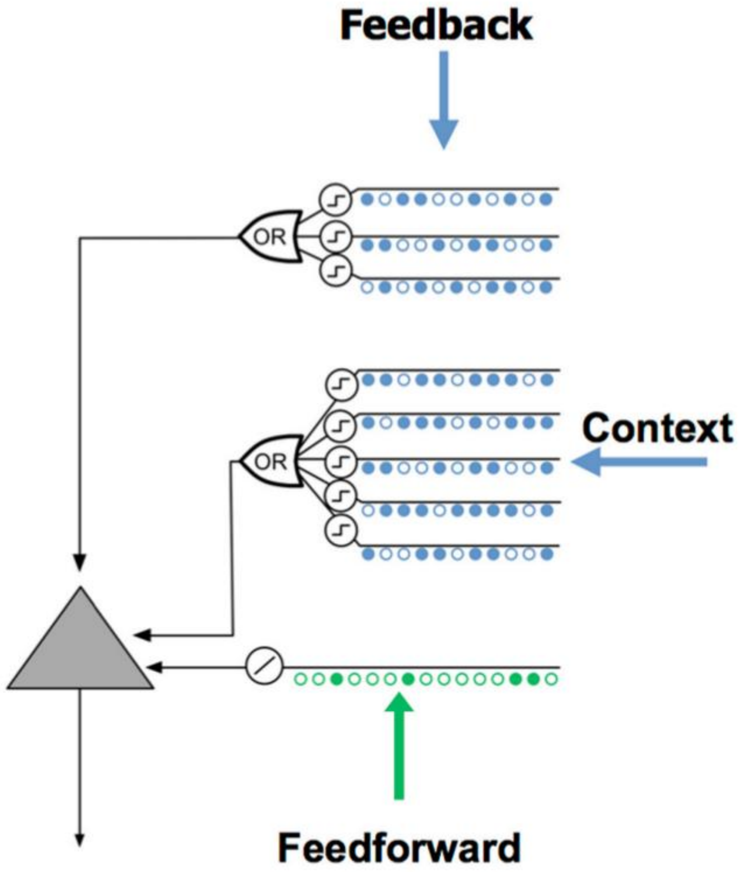
Image source: https://numenta.com/

Hierarchical Temporal Memory: Neuron model

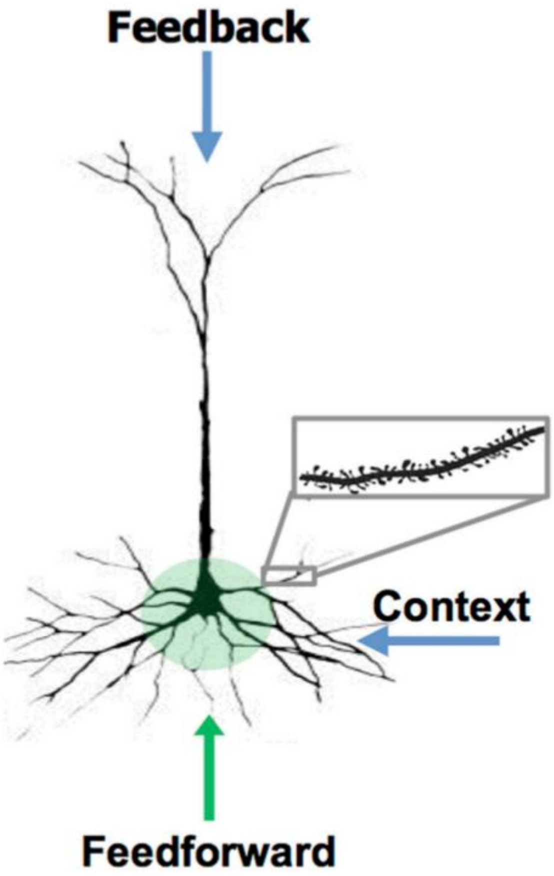


Hierarchical Temporal Memory: Neuron model

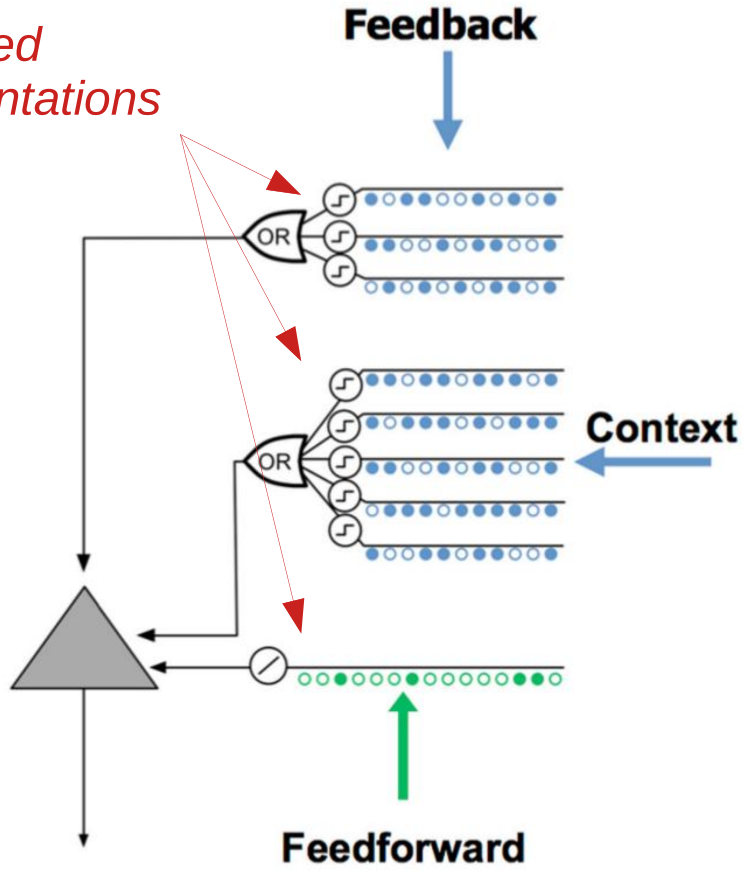




Hierarchical Temporal Memory: Neuron model



Sparse Distributed Representations



Quantum Cognition

- **Not** quantum mind (=“the brain works by micro-physical quantum mechanics”)
- A theory that models cognition by the same math that is used to describe quantum mechanics

Quantum Cognition

- **Not** quantum mind (=“the brain works by micro-physical quantum mechanics”)
- A theory that models cognition by the same math that is used to describe quantum mechanics
- Important tool: representation using vector spaces and operators (e.g. sums and projections)
- Motivation: Some paradox or irrational judgements of humans can’t be explained using classical probability theory and logic, e.g. conjunction and disjunction errors or order effects

Busemeyer, J., & Bruza, P. (2012). Quantum Models of Cognition and Decision. Cambridge: Cambridge University Press. doi:10.1017/CBO9780511997716

Outline: Introduction to high dimensional computing

1) Historical overview

2) High dimensional vector spaces and where they are used

3) **Mathematical properties of high dimensional vector spaces**

4) Vector Symbolic Architectures or “How to do symbolic computations using vectors spaces”

    including “What is the Dollar of Mexico?”

Four properties of high-dimensional vector spaces

“The good, the bad, and the ugly”

The Trivial
The Bad
The Surprising
The Good

Properties 1/4: High-dimensional vector spaces have huge capacity

The Trivial
The Bad
The Surprising
The Good

Properties 1/4: High-dimensional vector spaces have huge capacity

The Trivial
The Bad
The Surprising
The Good

- Capacity grows exponentially
- Here: “high-dimensional” means thousands of dimensions

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_{4000} \end{pmatrix}$$

Properties 1/4: High-dimensional vector spaces have huge capacity

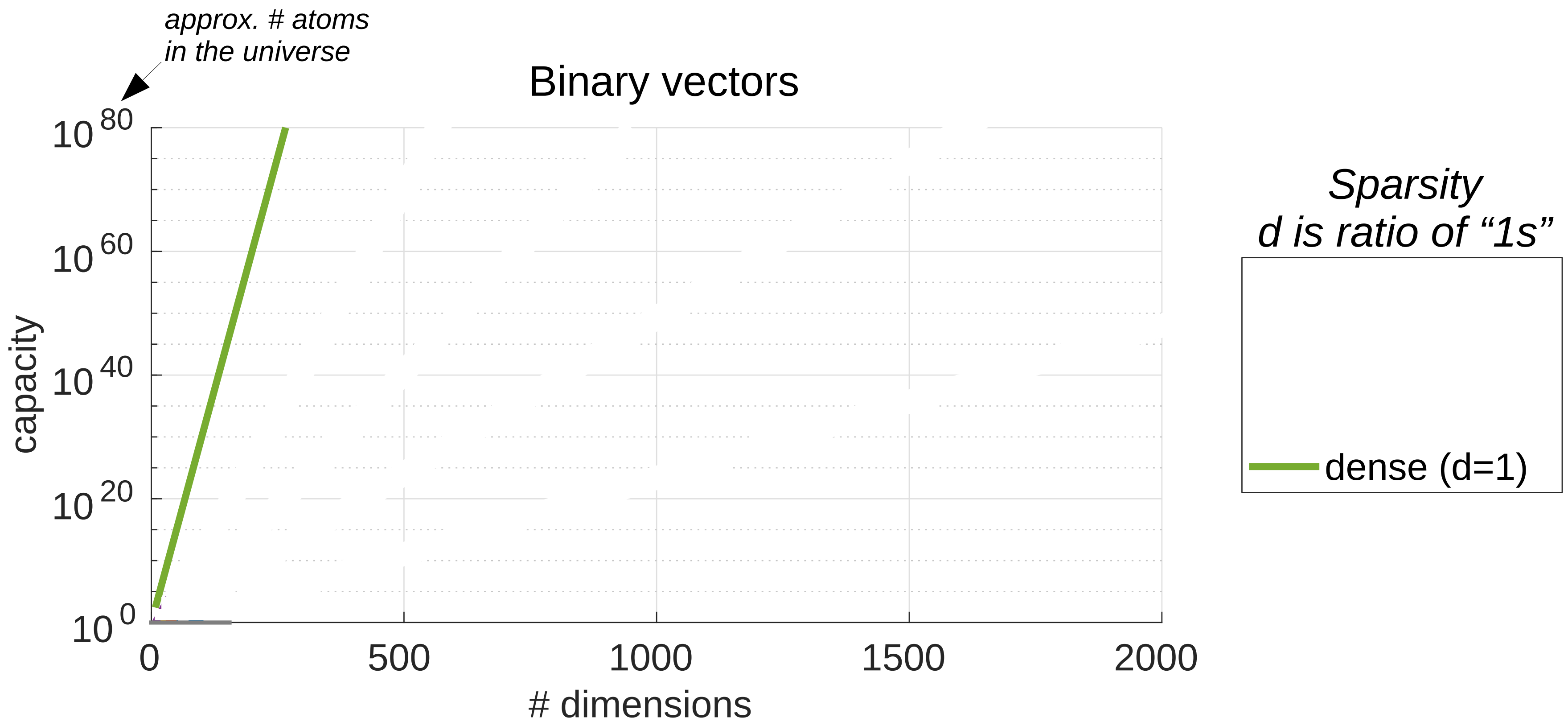
The Trivial
The Bad
The Surprising
The Good

- Capacity grows exponentially
- Here: “high-dimensional” means thousands of dimensions

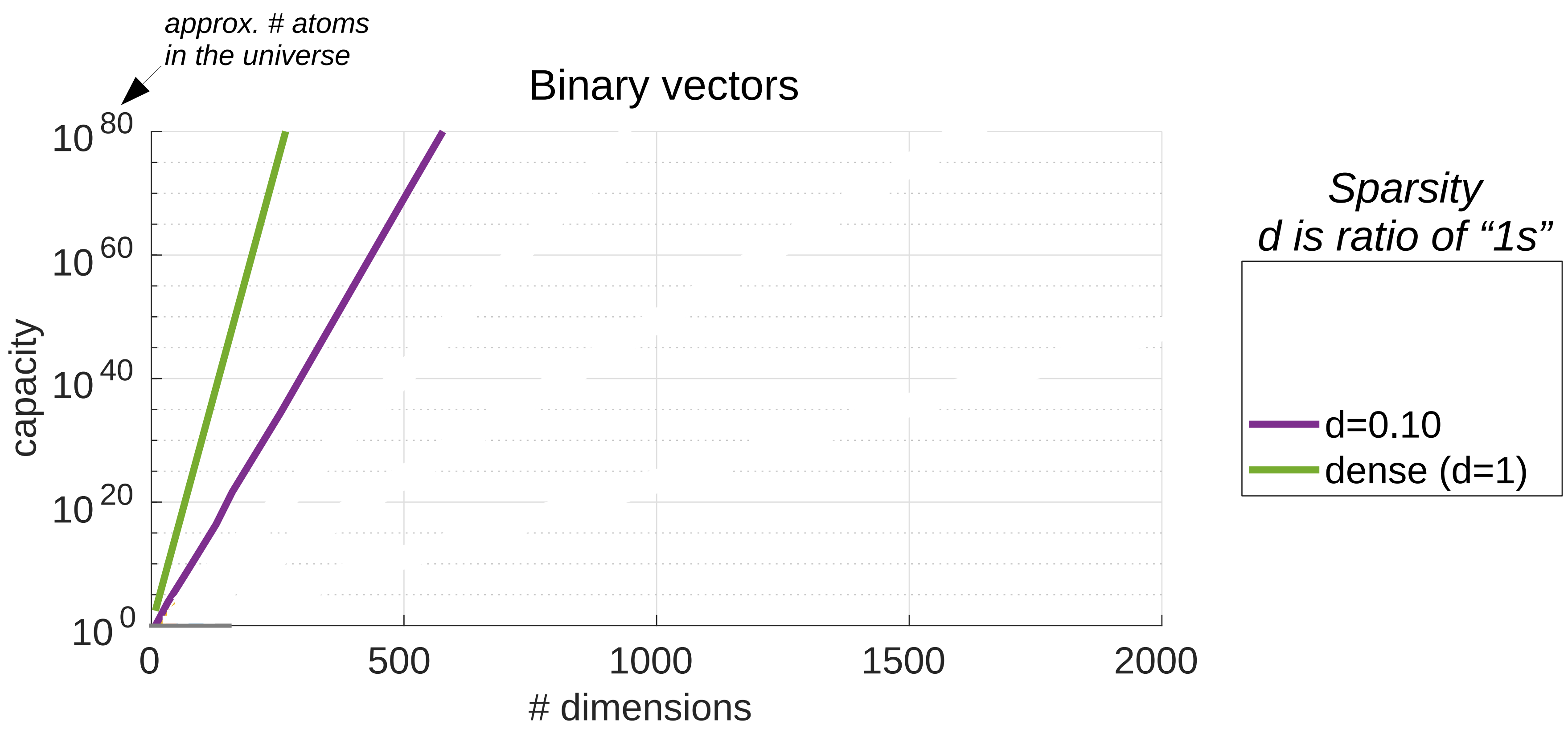
$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_{4000} \end{pmatrix}$$

- This property also holds for other vector spaces than $\mathbb{R}^n$
  - Binary, e.g. $\{0, 1\}^n$, $\{-1, 1\}^n$
  - Ternary, e.g. $\{-1, 0, 1\}^n$
  - Real, e.g. $[-1, 1]^n$
  - Sparse or Dense

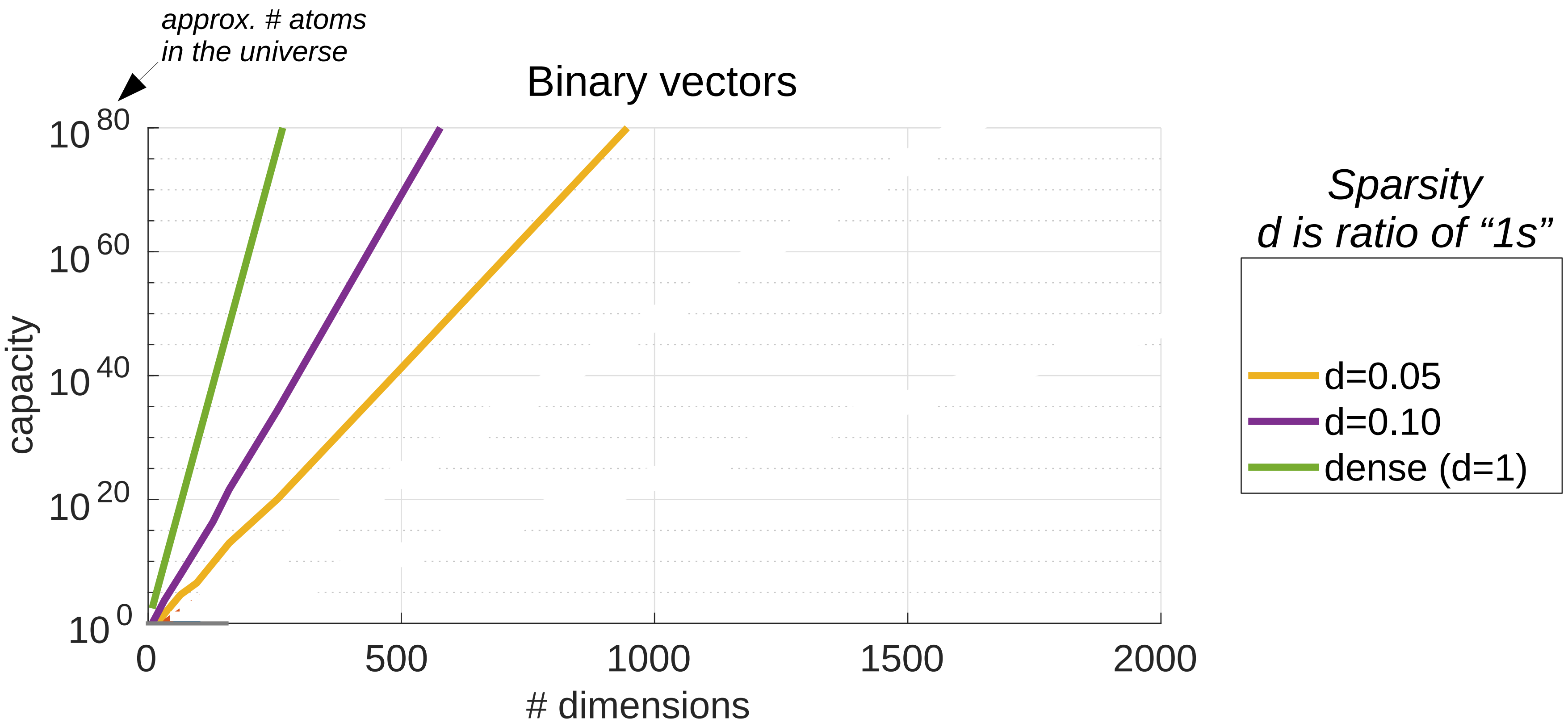
Properties 1/4: High capacity



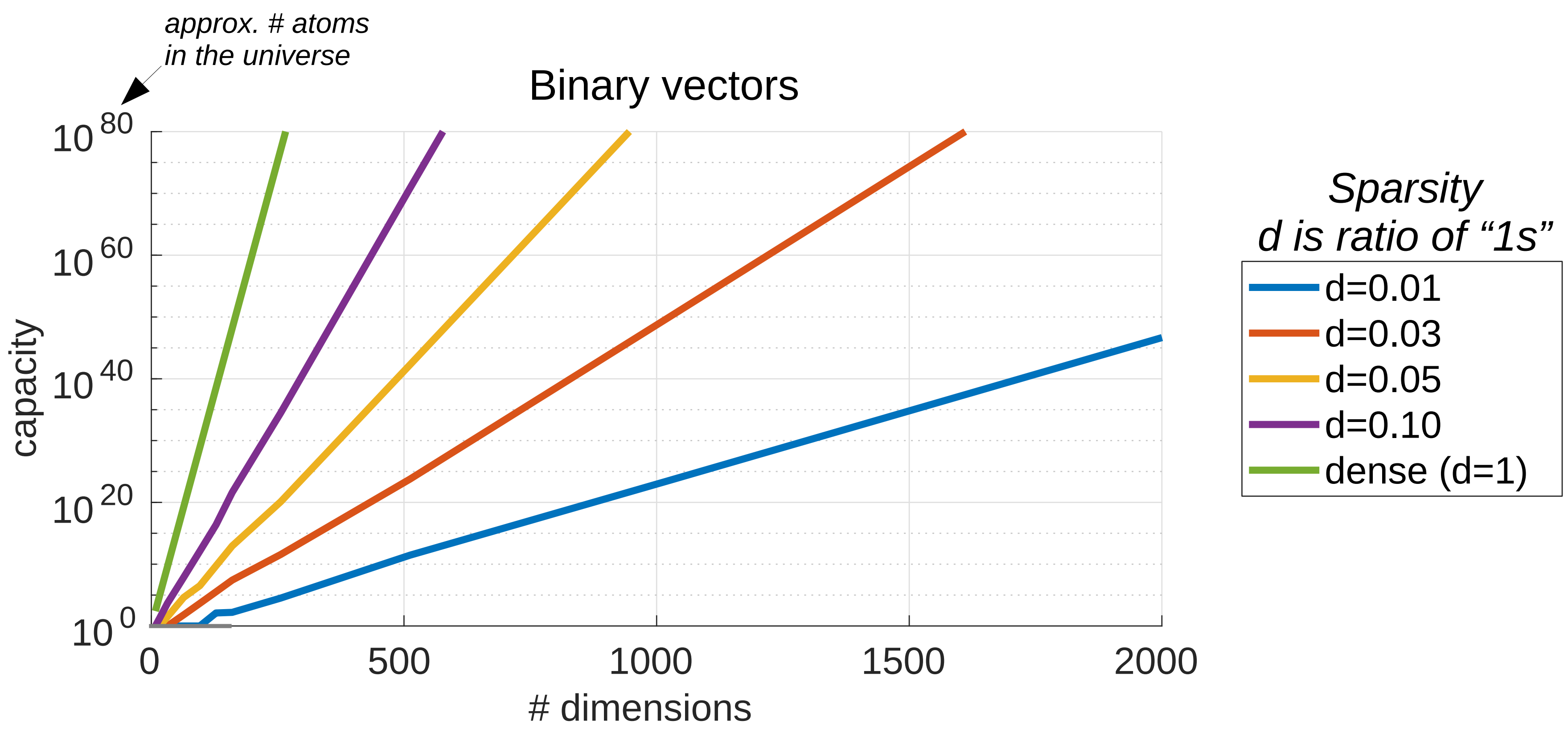
Properties 1/4: High capacity



Properties 1/4: High capacity



Properties 1/4: High capacity



Properties 2/4: Nearest neighbor becomes unstable or meaningless

The Trivial
The Bad
The Surprising
The Good

Properties 2/4: Nearest neighbor becomes unstable or meaningless

The Trivial
The Bad
The Surprising
The Good

Downside of so much space:

Bellman, 1961: “**Curse of dimensionality**”

- “Algorithms that work in low dimensional space fail in higher dimensional spaces”
- We require exponential amounts of samples to represent space with statistical significance (e.g., Hastie et al. 2009)

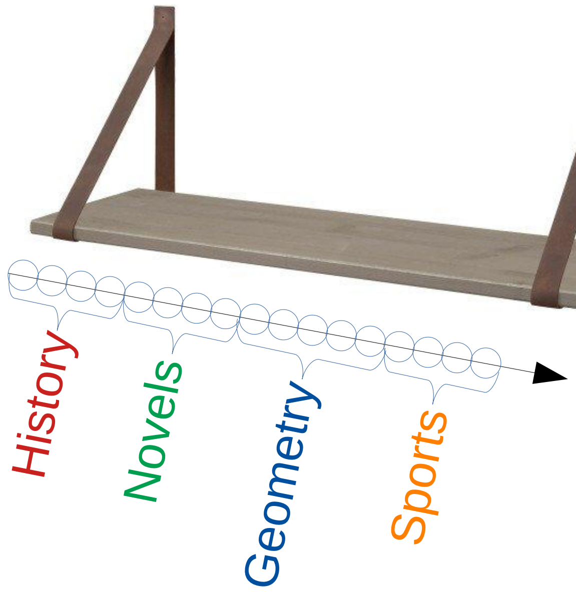
Bellman, R. E. (1961) Adaptive Control Processes: A Guided Tour. MIT Press, Cambridge

Hastie, Tibshirani and Friedman (2009). The Elements of Statistical Learning (2nd edition)Springer-Verlag

Example: Sorted library

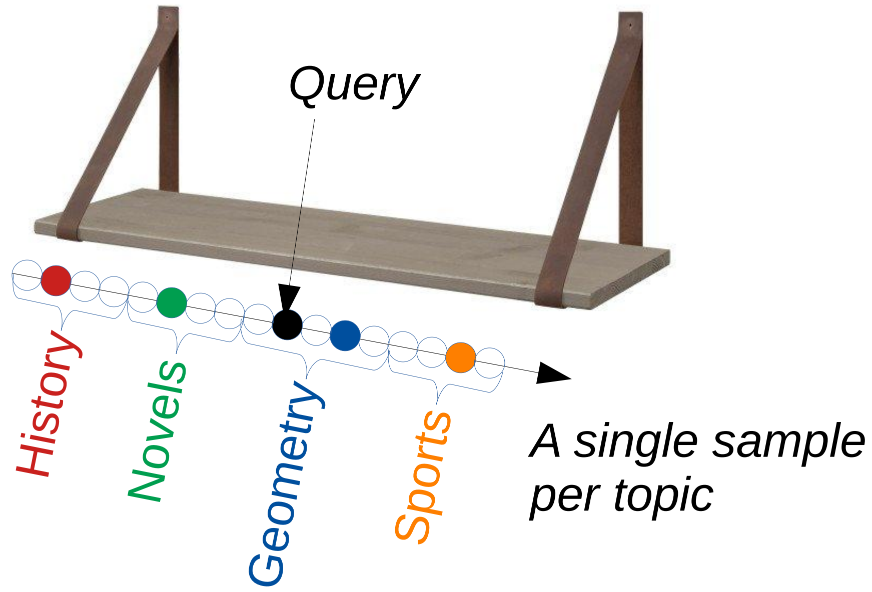
The Trivial
The Bad
The Surprising
The Good

Example: Sorted library



- Library contains books about 4 topics

Example: Sorted library

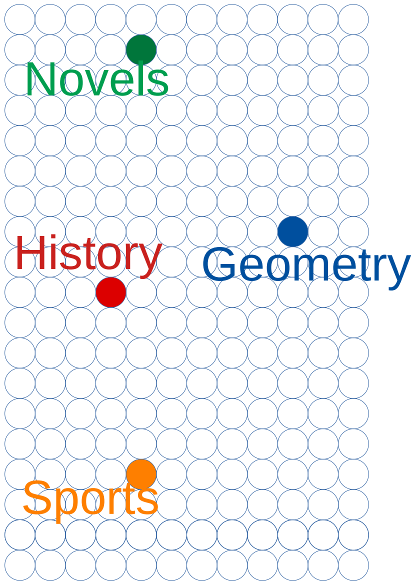


- Library contains books about 4 topics
- We can’t infer the topic from the pose directly, only by nearby samples.

Example: Sorted library

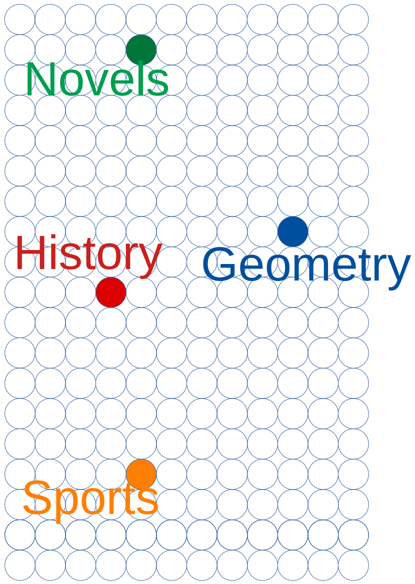
The Trivial
The Bad
The Surprising
The Good

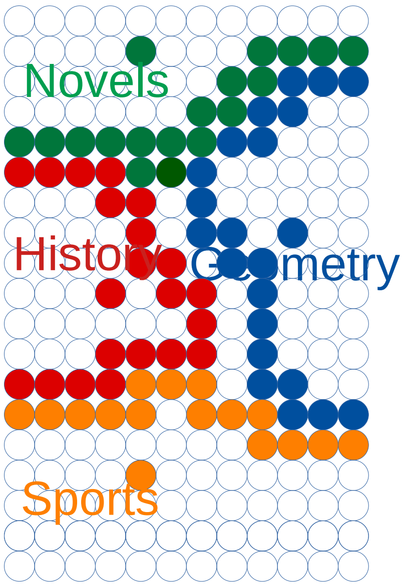
Example: Sorted library



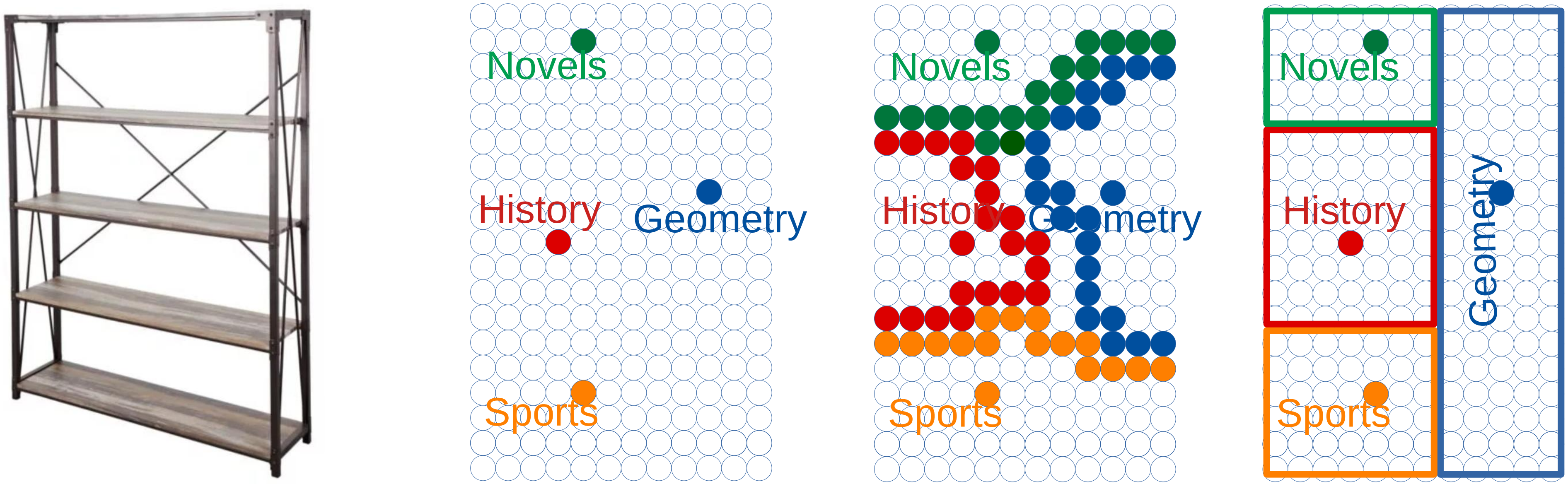
The Trivial
The Bad
The Surprising
The Good

Example: Sorted library





Example: Sorted library



The more dimensions, the more samples are required to represent the shape of the clusters.

Exponential growth!

Properties 2/4: Nearest neighbor becomes unstable or meaningless

- Beyer K, Goldstein J, Ramakrishnan R, Shaft U (1999) **When Is nearest neighbor meaningful?** In: Database theory—ICDT'99. Springer, Berlin, Heidelberg, pp 217–235

The Trivial
The Bad
The Surprising
The Good

Properties 2/4: Nearest neighbor becomes unstable or meaningless

- Beyer K, Goldstein J, Ramakrishnan R, Shaft U (1999) **When Is nearest neighbor meaningful?** In: Database theory—ICDT'99. Springer, Berlin, Heidelberg, pp 217–235

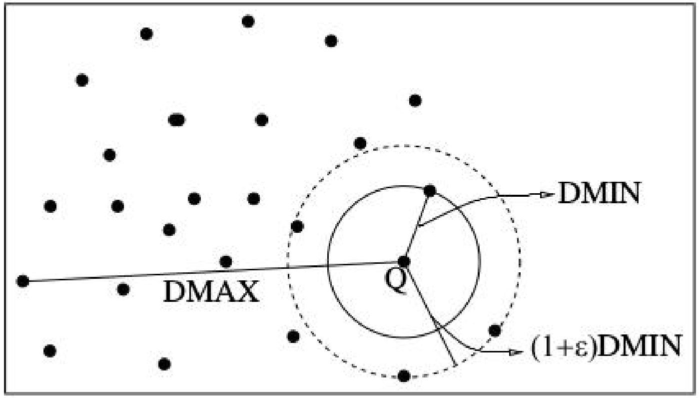


Fig. 4. Illustration of query region and enlarged region. (DMIN is the distance to the nearest neighbor, and DMAX to the farthest data point.)

Properties 2/4: Nearest neighbor becomes unstable or meaningless

- Beyer K, Goldstein J, Ramakrishnan R, Shaft U (1999) **When Is nearest neighbor meaningful?** In: Database theory—ICDT’99. Springer, Berlin, Heidelberg, pp 217–235

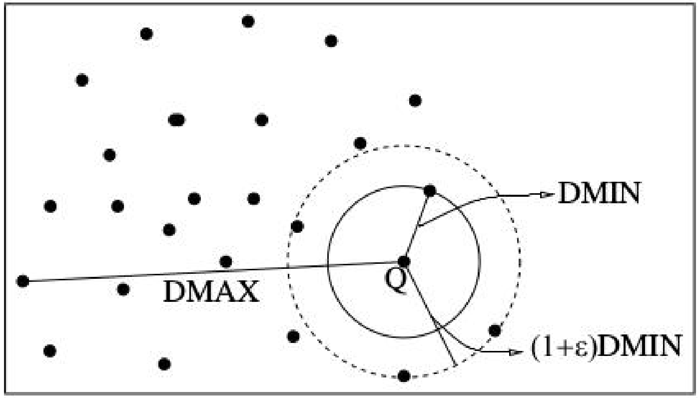


Fig. 4. Illustration of query region and enlarged region. (DMIN is the distance to the nearest neighbor, and DMAX to the farthest data point.)

The Trivial
The Bad
The Surprising
The Good

“under a broad set of conditions (much broader than independent and identically distributed dimensions)”

Then for every $\varepsilon > 0$

$$\lim_{m \to \infty} P\left[DMAX_m \leq (1+\varepsilon) DMIN_m\right] = 1$$

Increasing #dimensions

Properties 2/4: Nearest neighbor becomes unstable or meaningless

- Beyer K, Goldstein J, Ramakrishnan R, Shaft U (1999) **When Is nearest neighbor meaningful?** In: Database theory—ICDT'99. Springer, Berlin, Heidelberg, pp 217–235

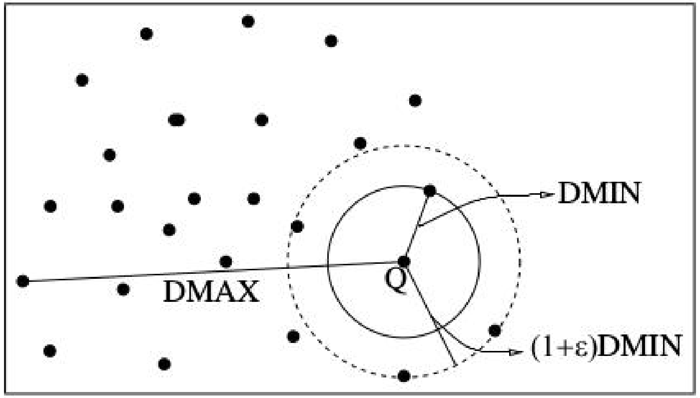


Fig. 4. Illustration of query region and enlarged region. (DMIN is the distance to the nearest neighbor, and DMAX to the farthest data point.)

The Trivial
The Bad
The Surprising
The Good

"under a broad set of conditions (much broader than independent and identically distributed dimensions)"

→

Then for every $\varepsilon > 0$

$$\lim_{m \to \infty} P\left[DMAX_m \leq (1+\varepsilon) DMIN_m\right] = 1$$

Increasing #dimensions

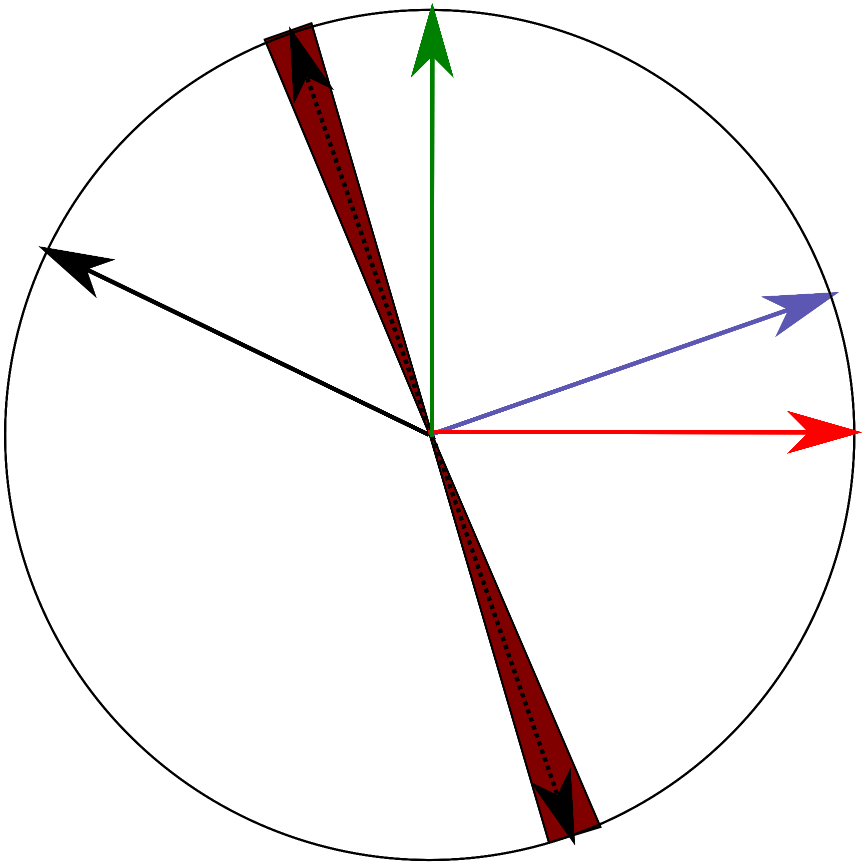
- Aggarwal CC, Hinneburg A, Keim DA (2001) On the surprising behavior of **distance metrics** in high dimensional space. In: Database theory—ICDT 2001. Springer, Berlin Heidelberg, pp 420–434

Properties 3/4: Time to gamble!

The Trivial
The Bad
The Surprising
The Good

Properties 3/4: Experiment

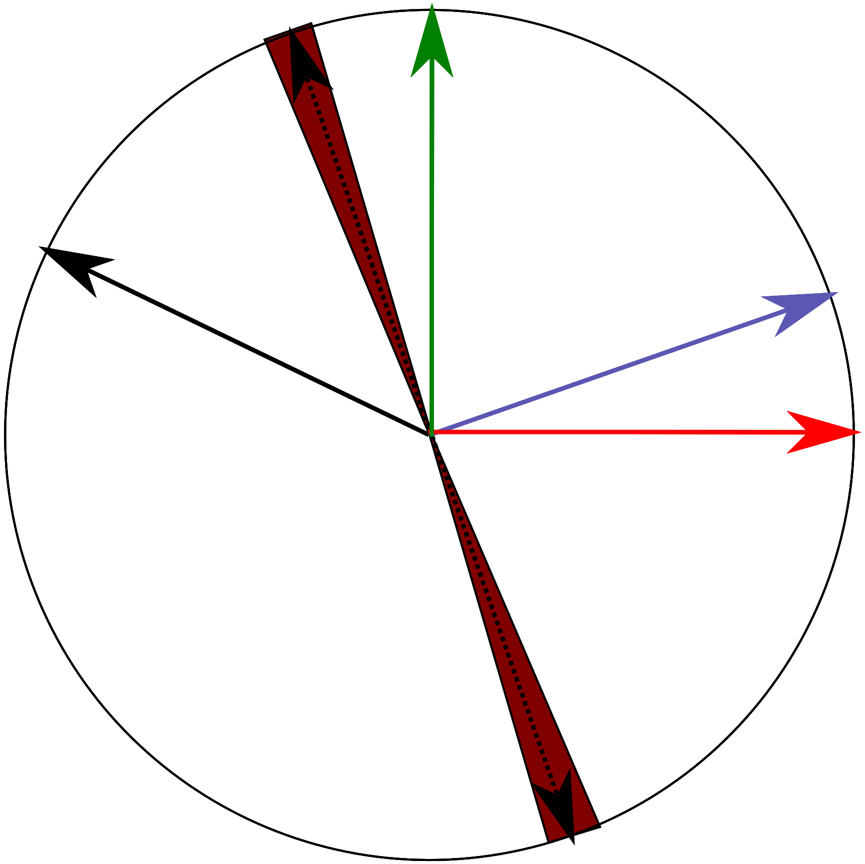
The Trivial
The Bad
The Surprising
The Good



- Random vectors:
  - uniformly distributed angles
  - obtained by sampling each dimension iid. ~N(0,1)

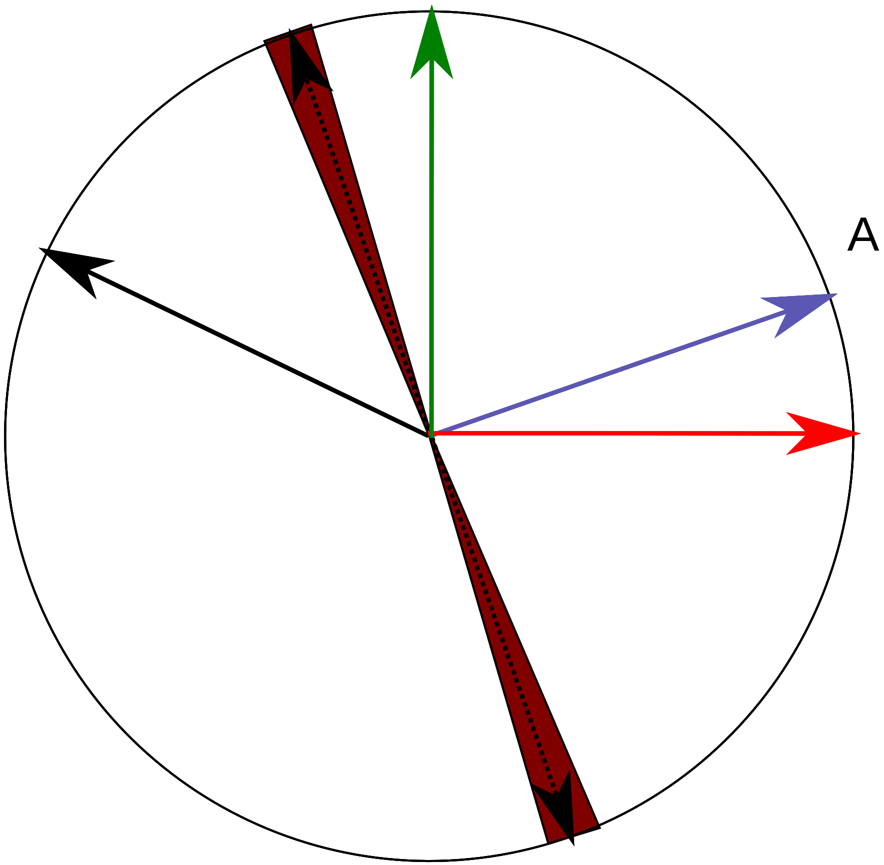
Properties 3/4: Experiment

The Trivial
The Bad
The Surprising
The Good



- Random vectors:
  - uniformly distributed angles
  - obtained by sampling each dimension iid. ~N(0,1)

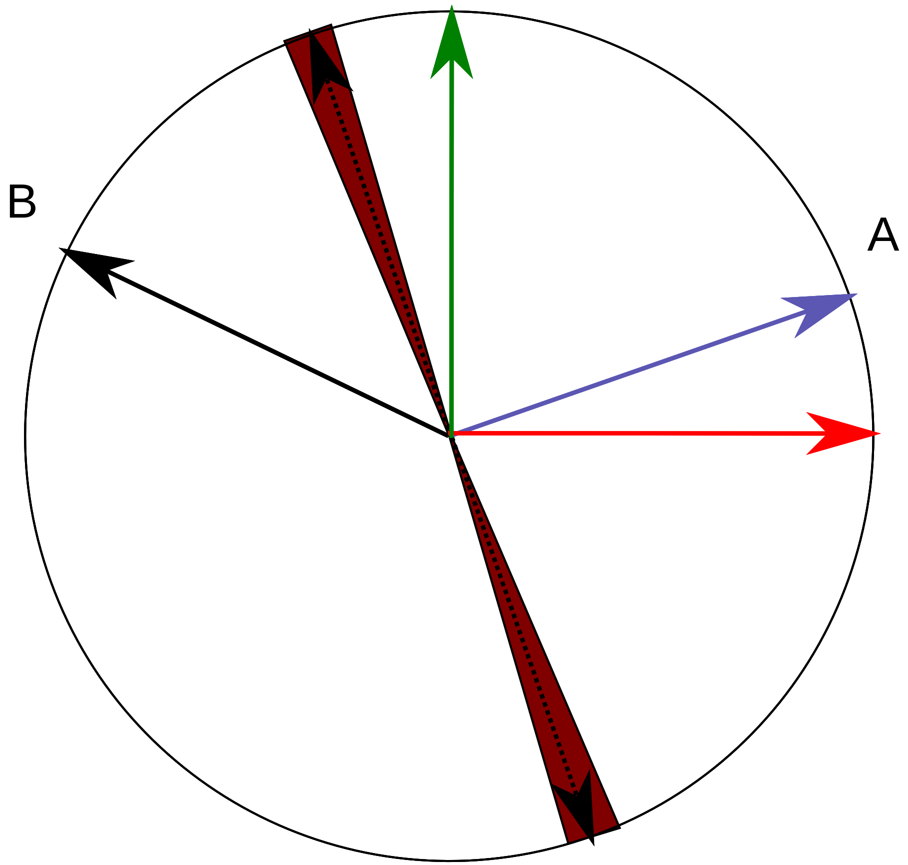
Properties 3/4: Experiment



- Random vectors:
  - uniformly distributed angles
  - obtained by sampling each dimension iid. ~N(0,1)
- I bet:
  - Given a random vector A, we can independently sample a second random vector B and it will be almost orthogonal (+/- 5°) ...

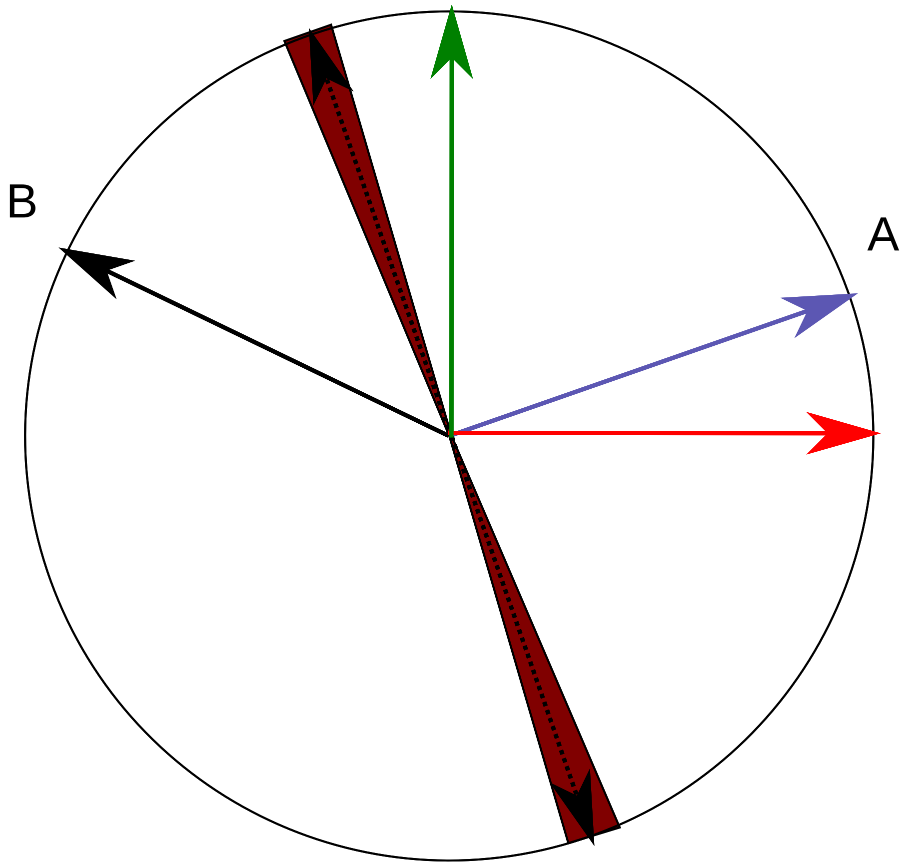
Properties 3/4: Experiment

The Trivial
The Bad
The Surprising
The Good



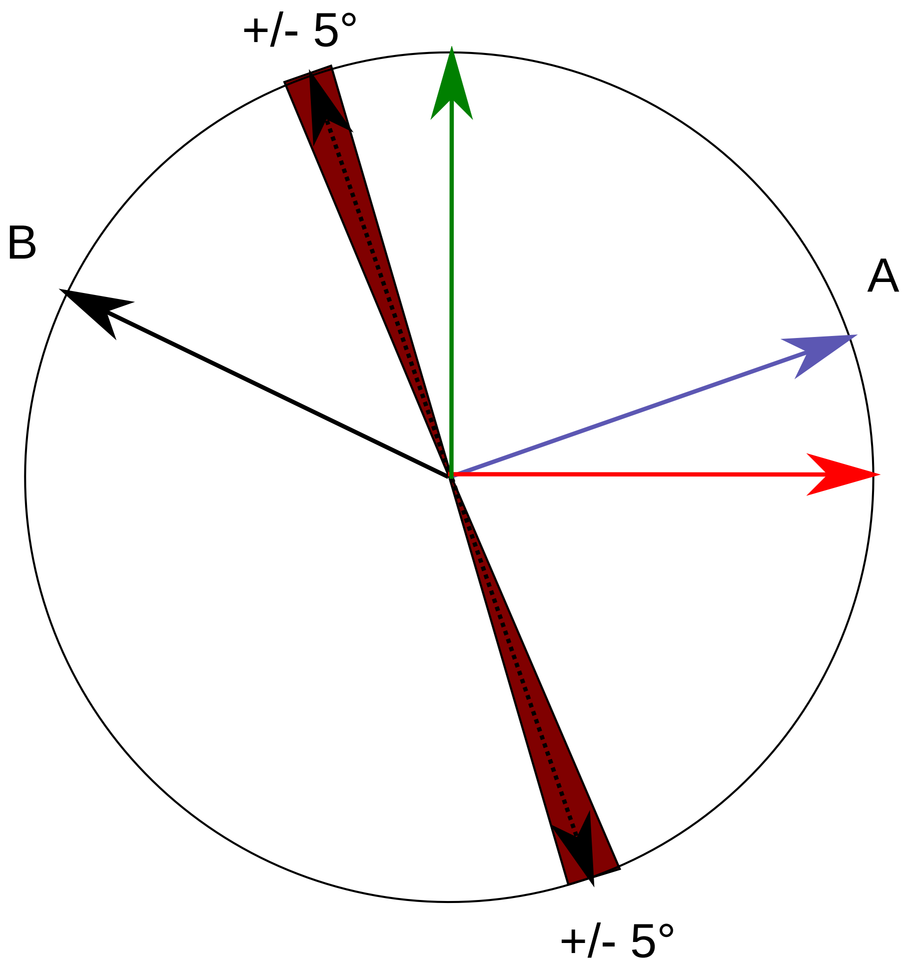
- Random vectors:
  - uniformly distributed angles
  - obtained by sampling each dimension iid. ~N(0,1)
- I bet:
  - Given a random vector A, we can independently sample a second random vector B and it will be almost orthogonal (+/- 5°) ...

Properties 3/4: Experiment



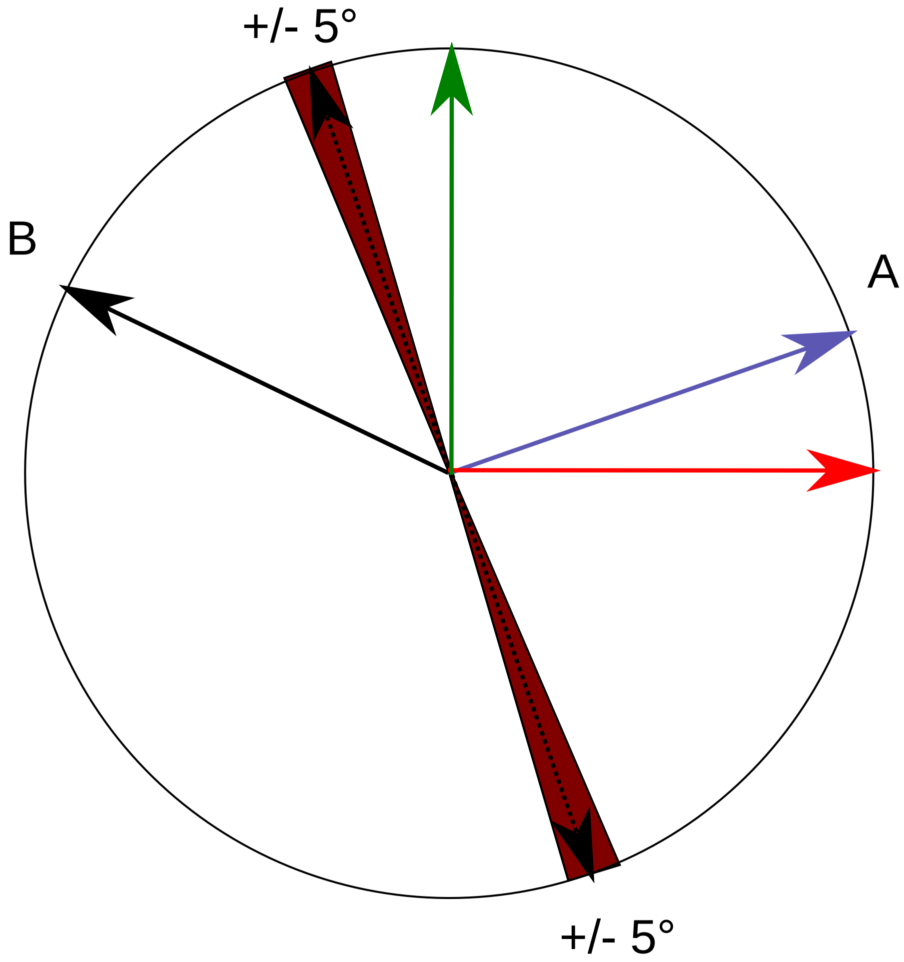
- Random vectors:
  - uniformly distributed angles
  - obtained by sampling each dimension iid. ~N(0,1)
- I bet:
  - Given a random vector A, we can independently sample a second random vector B and it will be almost orthogonal (+/- 5°) ...

Properties 3/4: Experiment



- Random vectors:
  - uniformly distributed angles
  - obtained by sampling each dimension iid. ~N(0,1)
- I bet:
  - Given a random vector A, we can independently sample a second random vector B and it will be almost orthogonal (+/- 5°) ...

Properties 3/4: Experiment

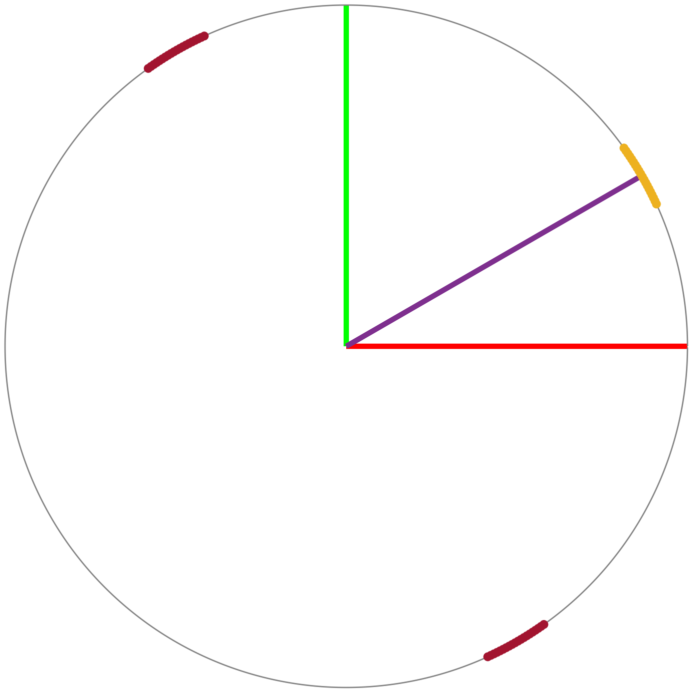


- Random vectors:
  - uniformly distributed angles
  - obtained by sampling each dimension iid. ~N(0,1)
- I bet:
  - Given a random vector A, we can independently sample a second random vector B and it will be almost orthogonal (+/- 5°) ...
  - … if we are in a 4,000 dimensional vector space.

Properties 3/4: Random vectors are very likely almost orthogonal

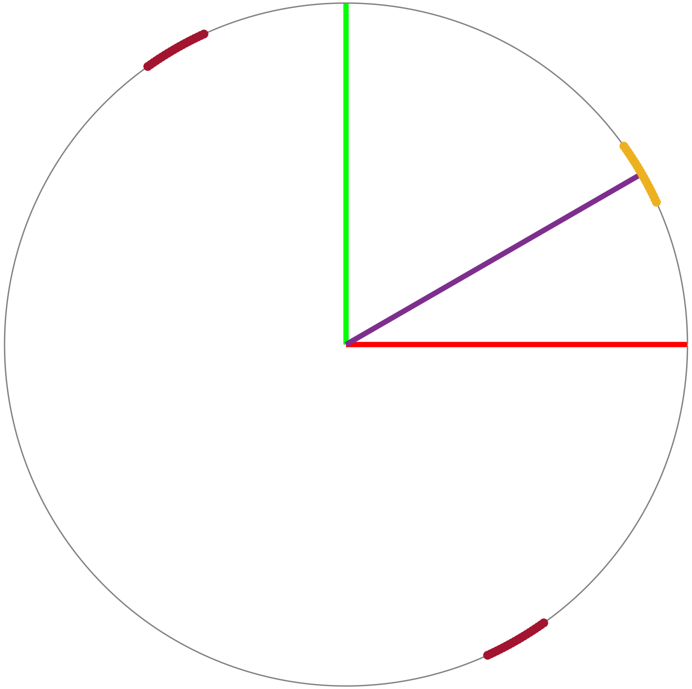
The Trivial
The Bad
The Surprising
The Good

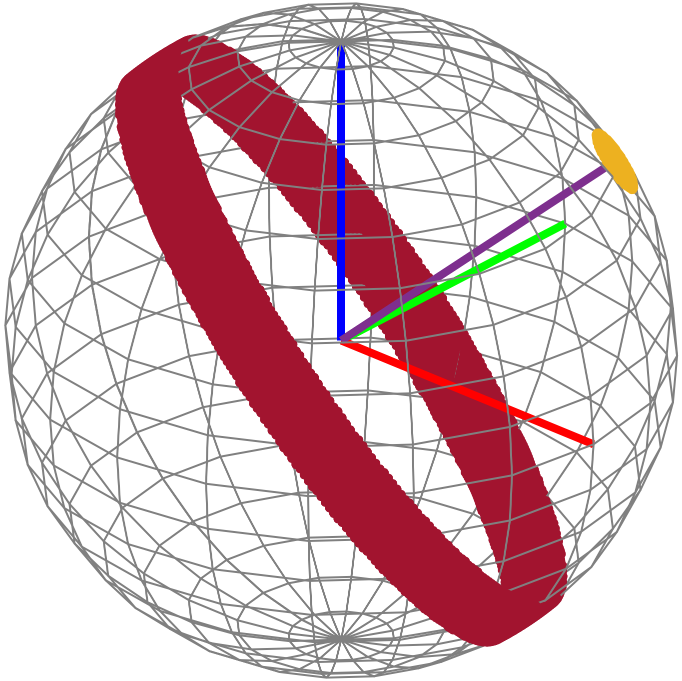
- Random vectors: iid, uniform



Properties 3/4: Random vectors are very likely almost orthogonal

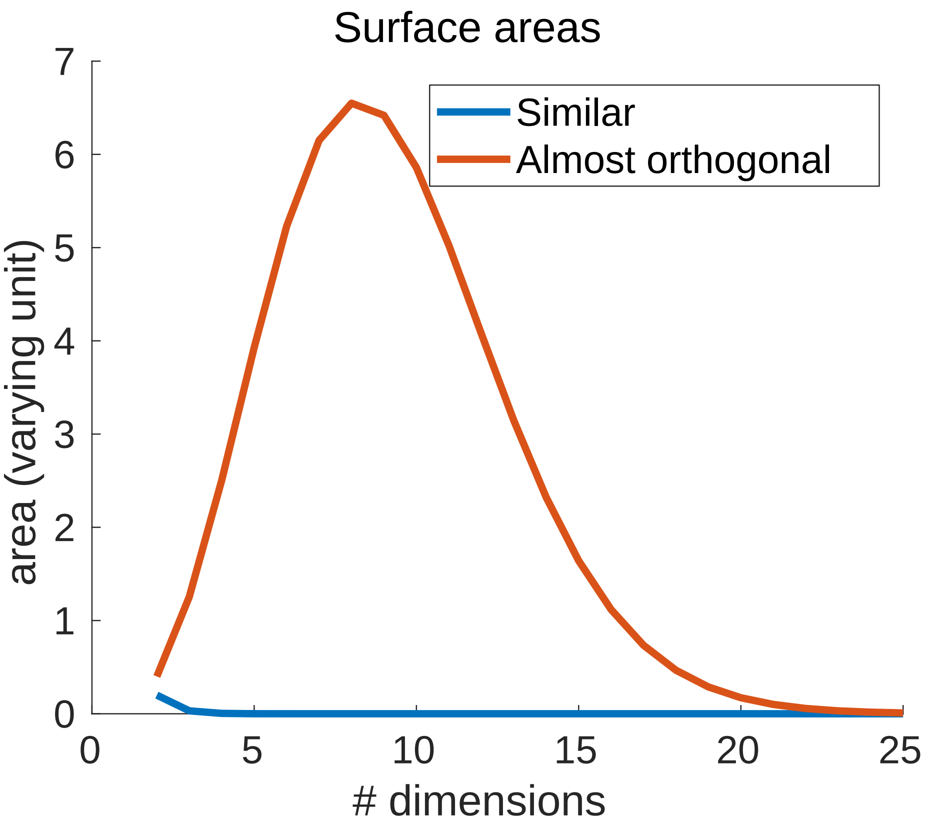
- Random vectors: iid, uniform





Properties 3/4: Random vectors are very likely almost orthogonal

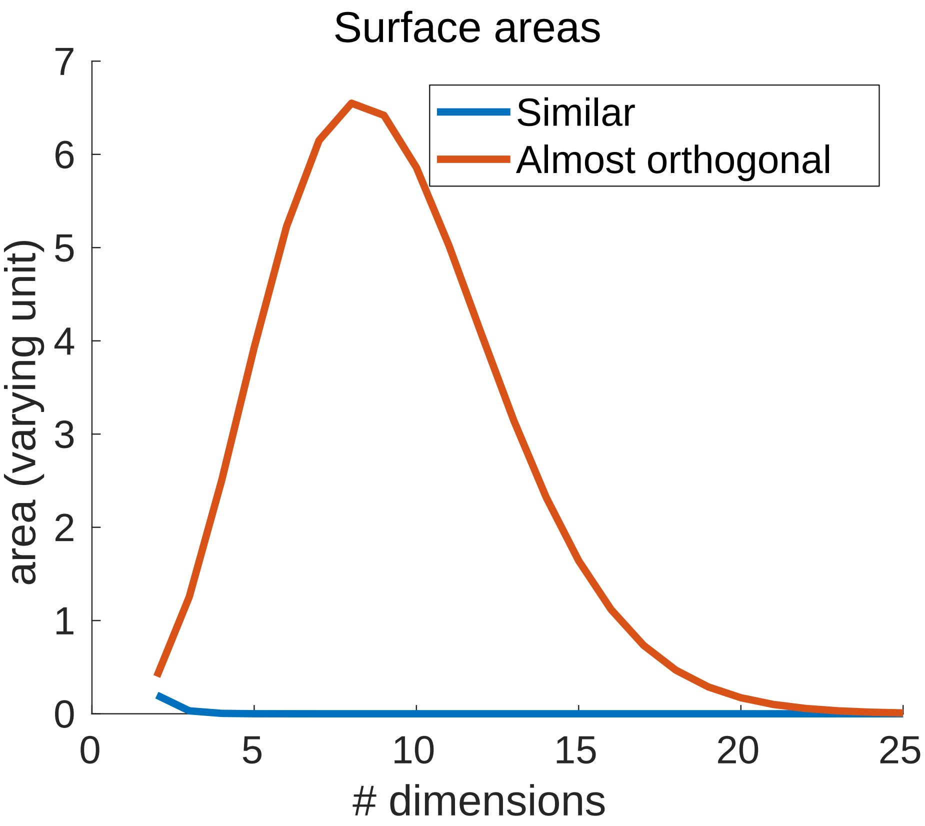
- Random vectors: iid, uniform

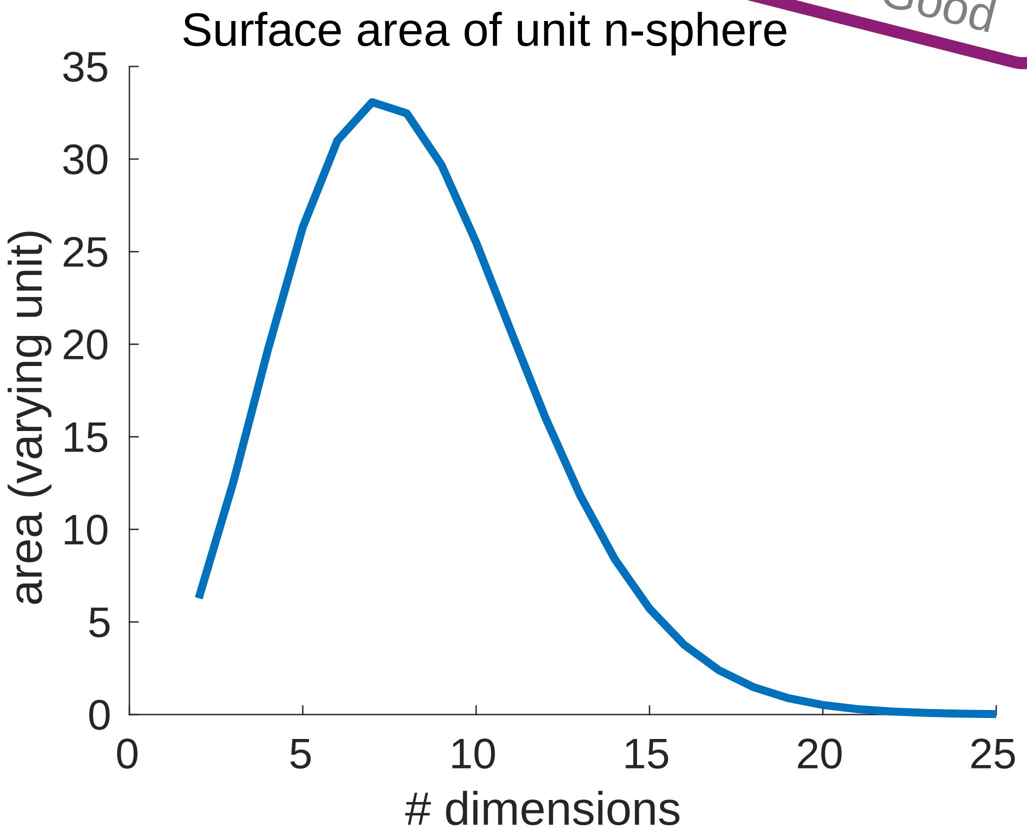


Properties 3/4: Random vectors are very likely almost orthogonal

The Trivial
The Bad
The Surprising
The Good

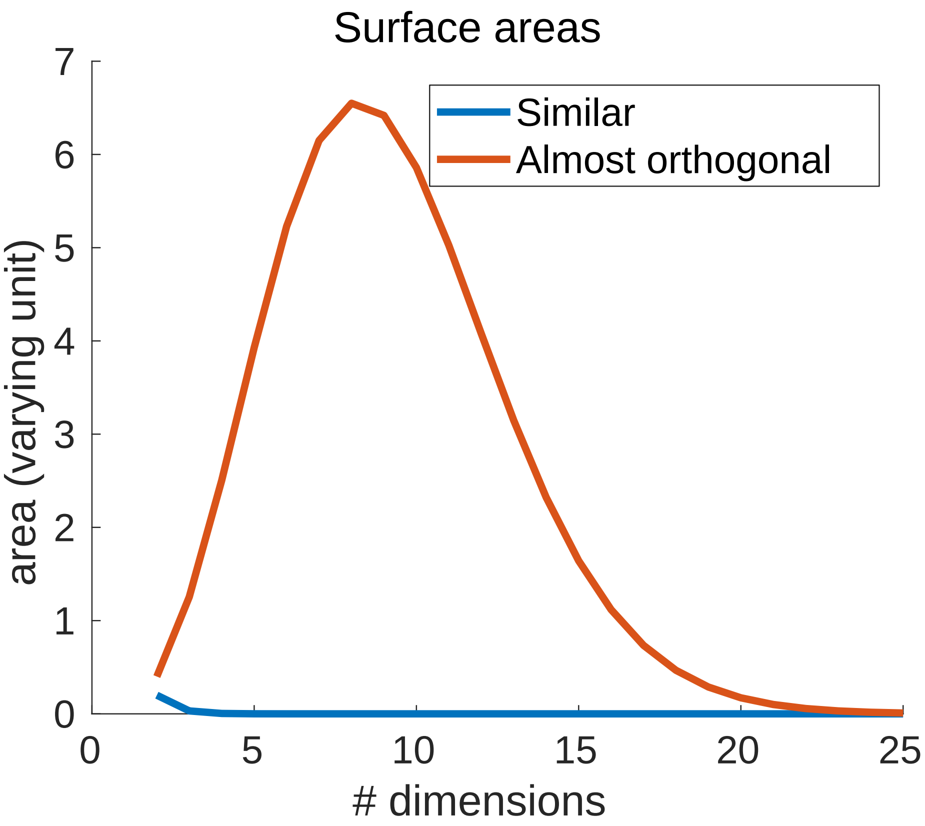
- Random vectors: iid, uniform

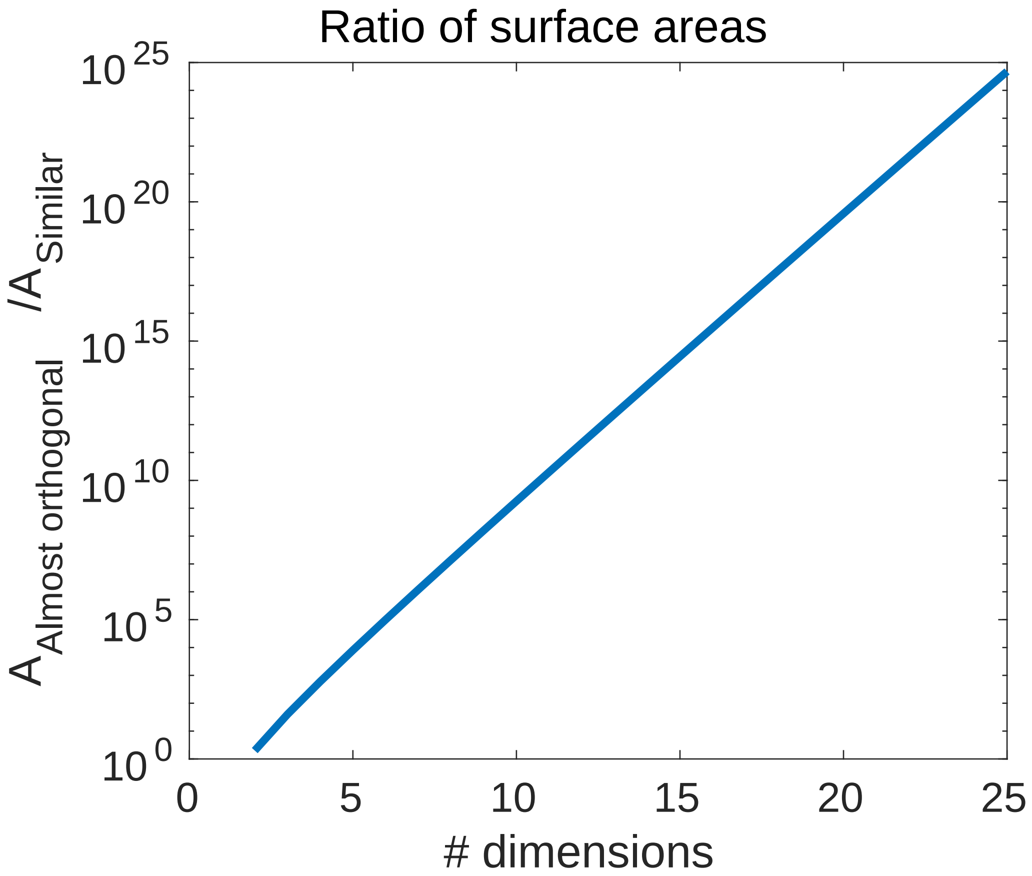




Properties 3/4: Random vectors are very likely almost orthogonal

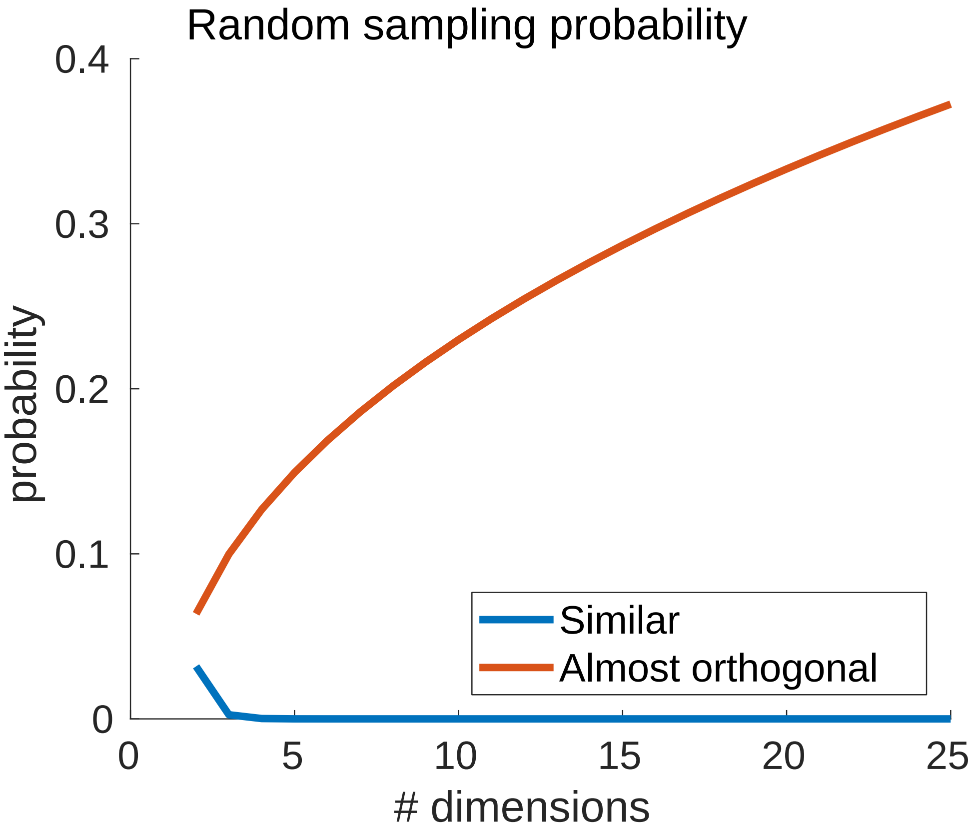
- Random vectors: iid, uniform

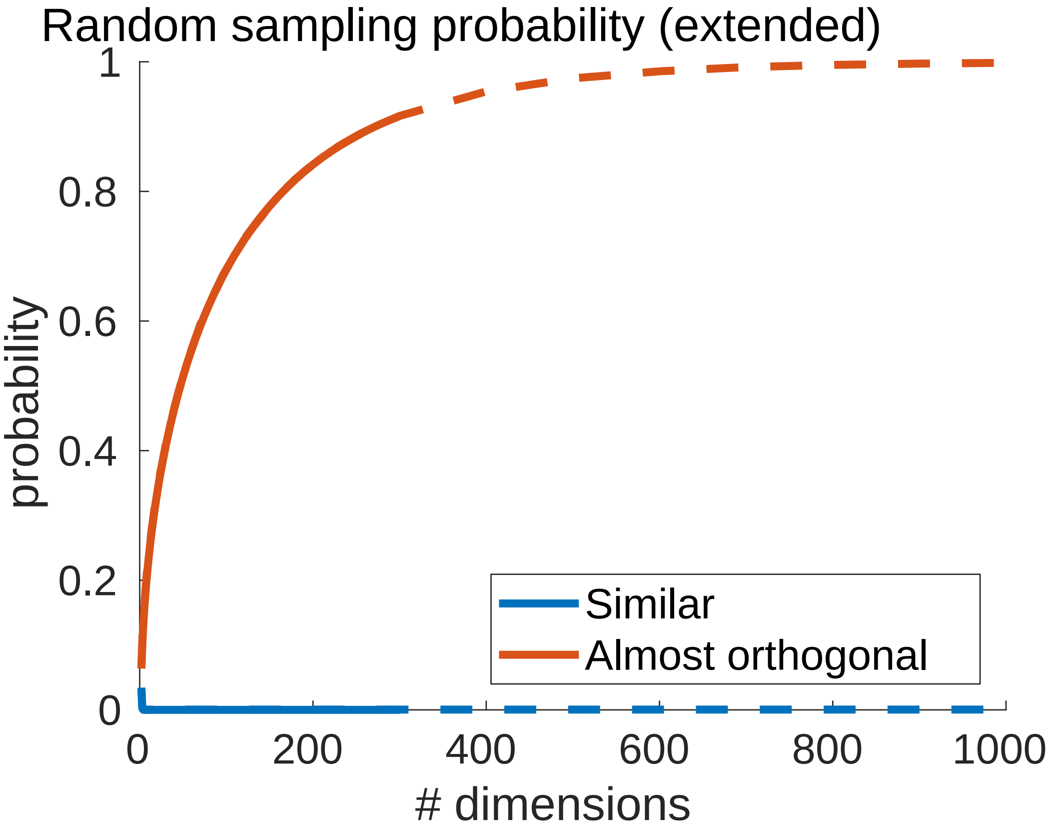




Properties 3/4: Random vectors are very likely almost orthogonal

- Random vectors: iid, uniform



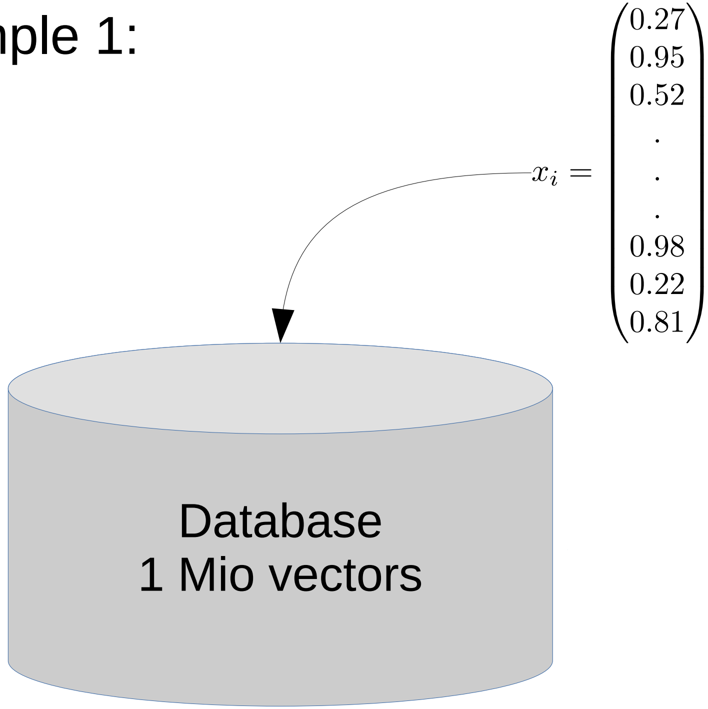


Properties 4/4: Noise has low influence on nearest neighbor queries with random vectors

The Trivial
The Bad
The Surprising
The Good

Properties 4/4: Noise has low influence on nearest neighbor queries with random vectors

- Example 1:



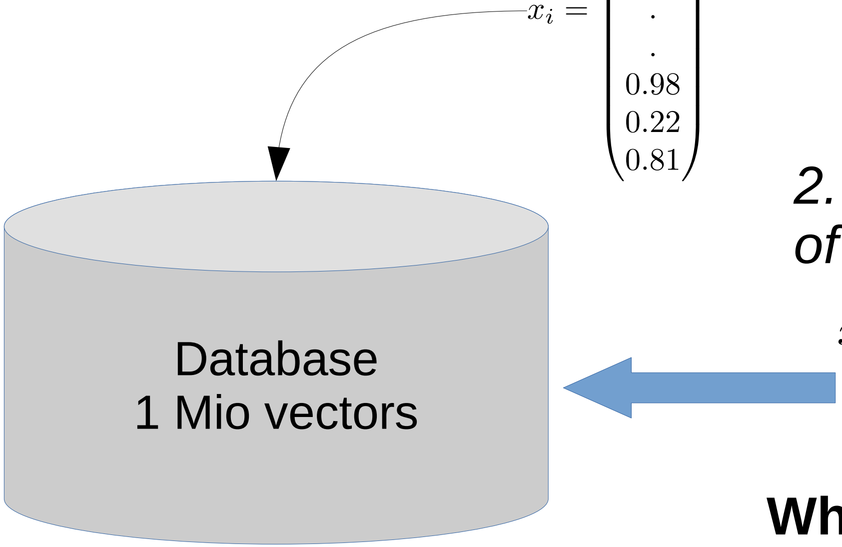
1. One million random feature vectors $[0,1]^d$

Properties 4/4: Noise has low influence on nearest neighbor queries with random vectors

The Trivial
The Bad
The Surprising
The Good

- Example 1:

$$x_i = \begin{pmatrix} 0.27 \\ 0.95 \\ 0.52 \\ . \\ . \\ . \\ 0.98 \\ 0.22 \\ 0.81 \end{pmatrix}$$



1. One million random feature vectors $[0,1]^d$

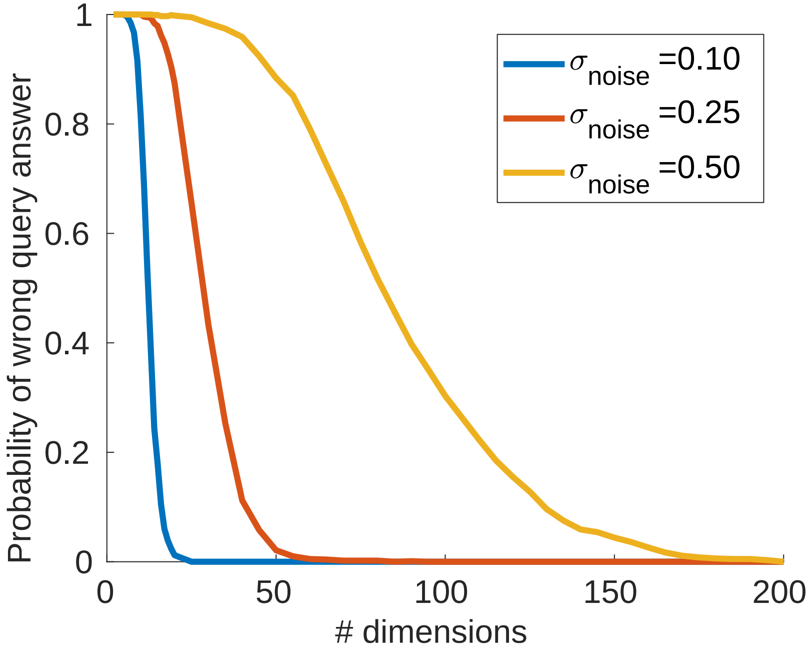
2. query: noisy measurements of feature vectors

$$x_i + \epsilon \text{ with } \epsilon \sim \mathcal{N}(0, \sigma)$$

What is the probability to get the wrong query answer?

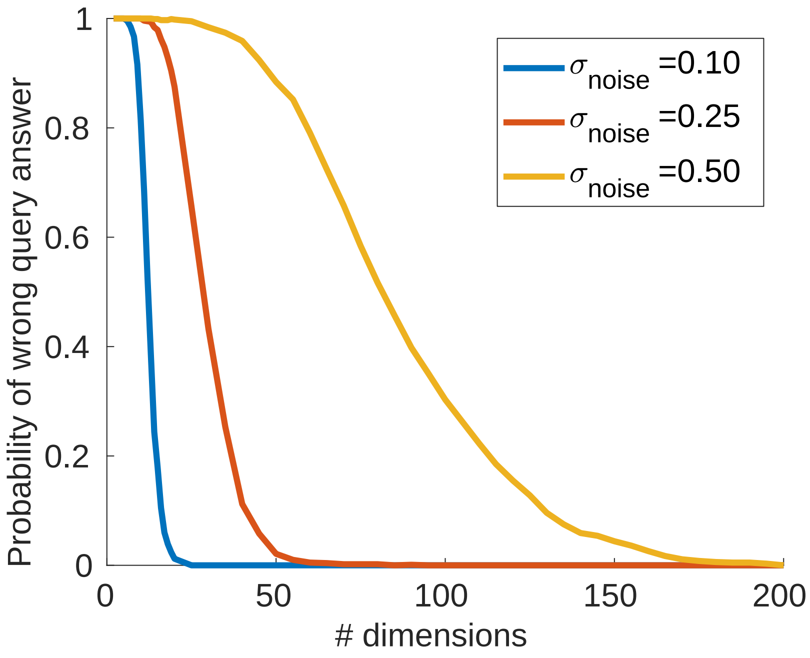
Properties 4/4: Noise has low influence on nearest neighbor queries with random vectors

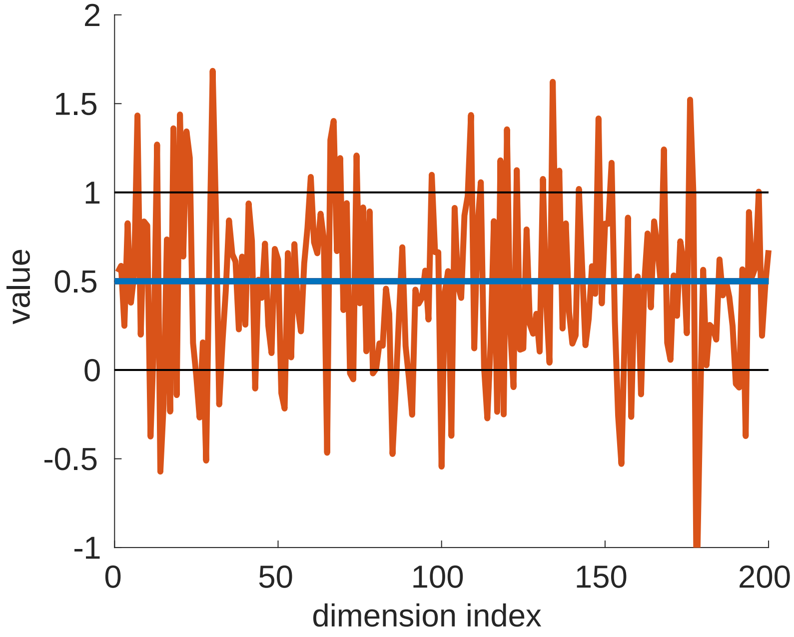
- Example 1:



Properties 4/4: Noise has low influence on nearest neighbor queries with random vectors

- Example 1:



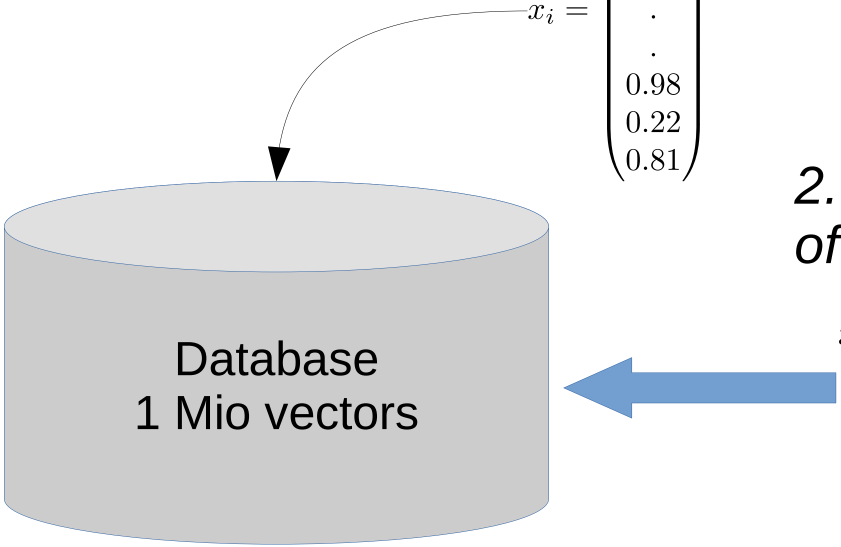


Properties 4/4: Noise has low influence on nearest neighbor queries with random vectors

- Example 2:

$$x_i = \begin{pmatrix} 0.27 \\ 0.95 \\ 0.52 \\ . \\ . \\ . \\ 0.98 \\ 0.22 \\ 0.81 \end{pmatrix}$$

1. One million random feature vectors [0,1]^d



2. query: noisy measurements of feature vectors

$$x_i + \epsilon \text{ with } \epsilon \sim \mathcal{N}(0, \sigma)$$

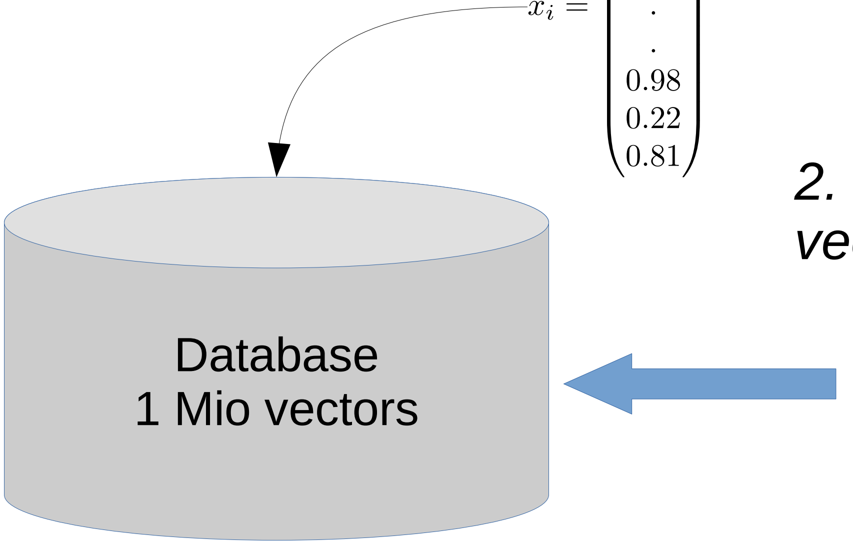
What if the noise-vector is again a database vector?

Properties 4/4: Noise has low influence on nearest neighbor queries with random vectors

- Example 2:

$$x_i = \begin{pmatrix} 0.27 \\ 0.95 \\ 0.52 \\ . \\ . \\ . \\ 0.98 \\ 0.22 \\ 0.81 \end{pmatrix}$$

1. One million random feature vectors [0,1]^d



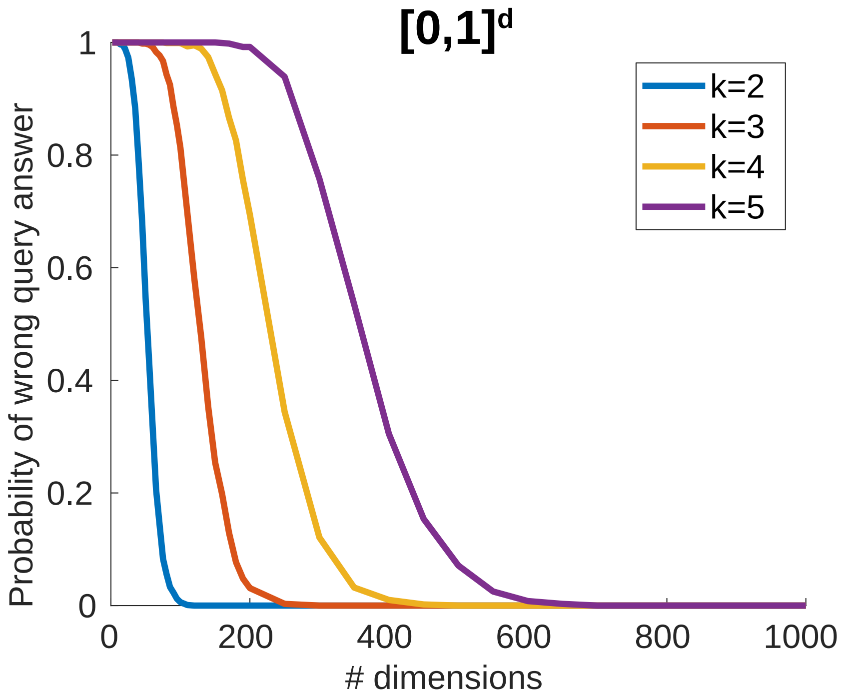
2. query: sum of feature vectors

$$\sum_{i=1}^{k} x_i$$

How many database vectors can we add (=bundle) and still get exactly *all* the added vectors as answer?

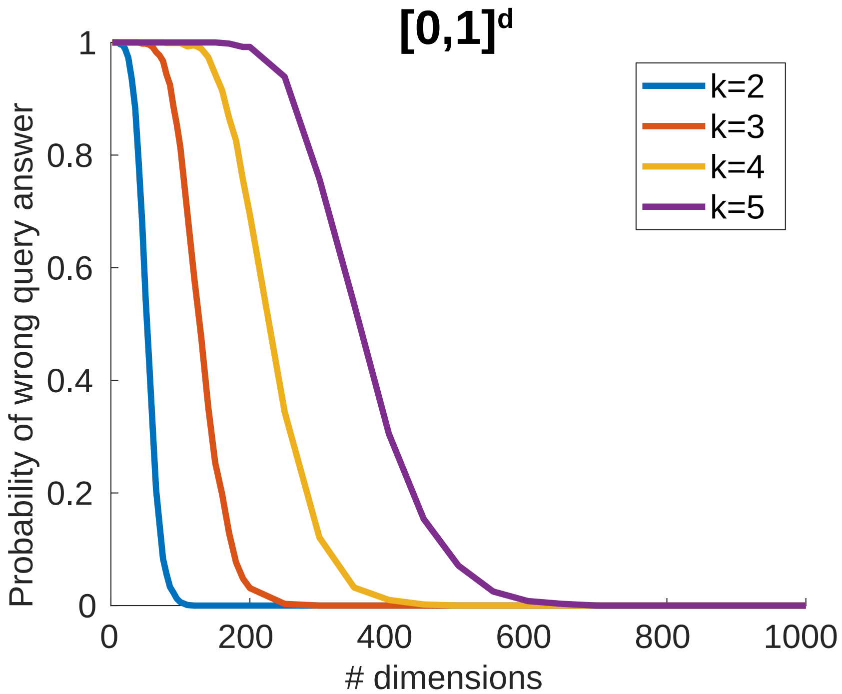
Properties 4/4: Noise has low influence on nearest neighbor queries with random vectors

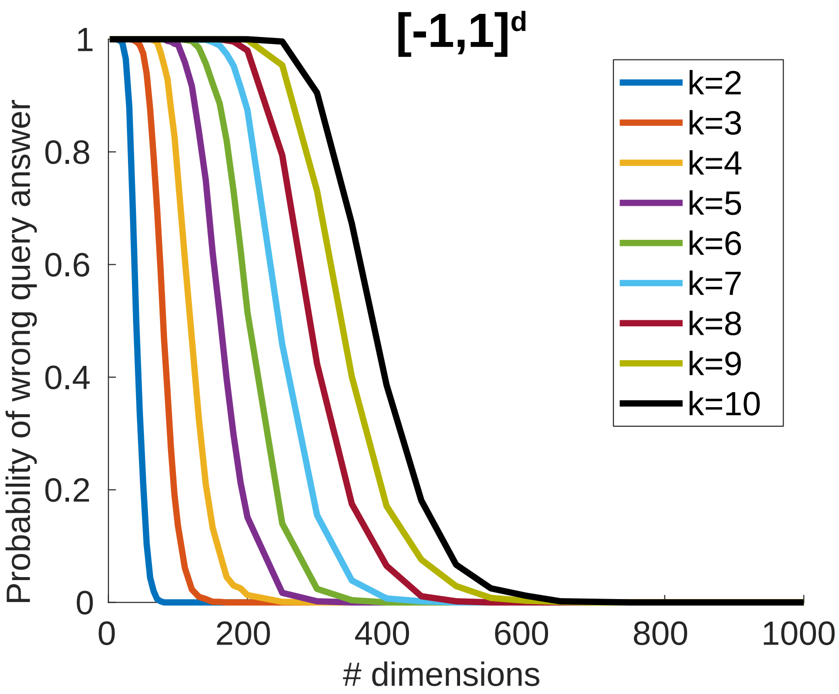
- Example 2:
  - How many database vectors can we add (=bundle) and still get exactly *all* the added vectors as answer?



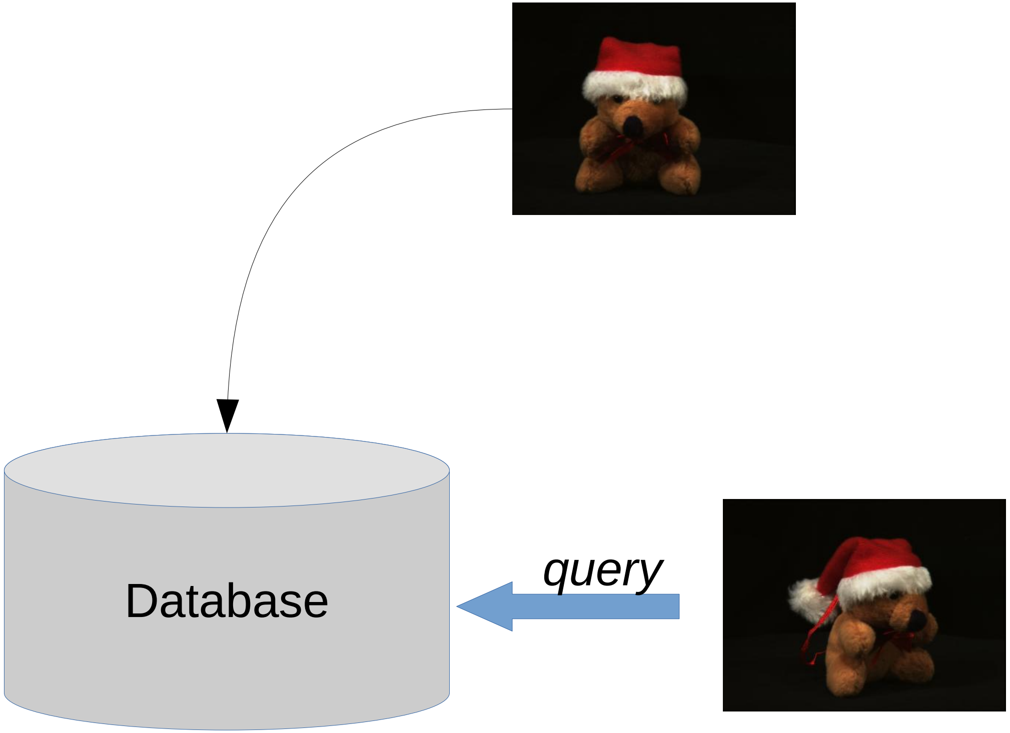
Properties 4/4: Noise has low influence on nearest neighbor queries with random vectors

- Example 2:
  - How many database vectors can we add (=bundle) and still get exactly *all* the added vectors as answer?

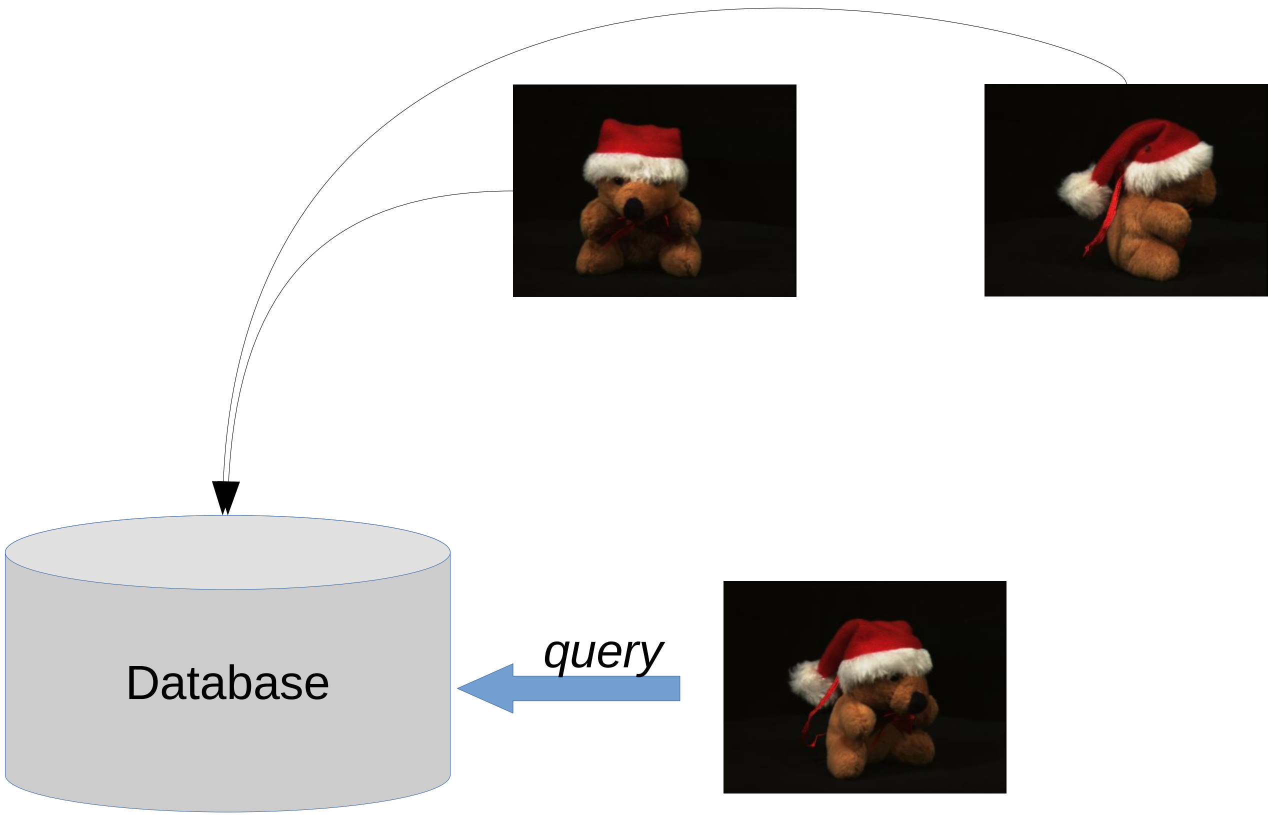




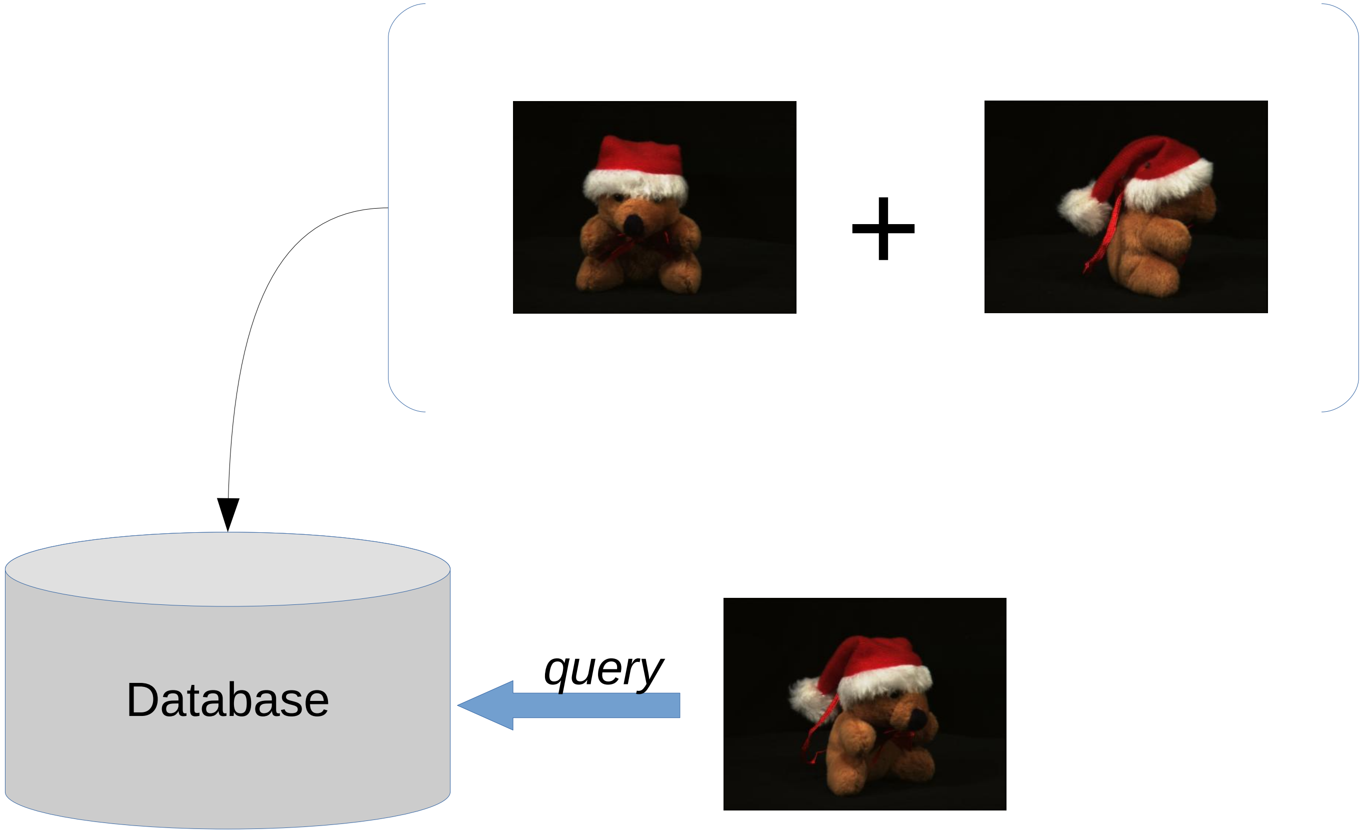
Example application: Object recognition



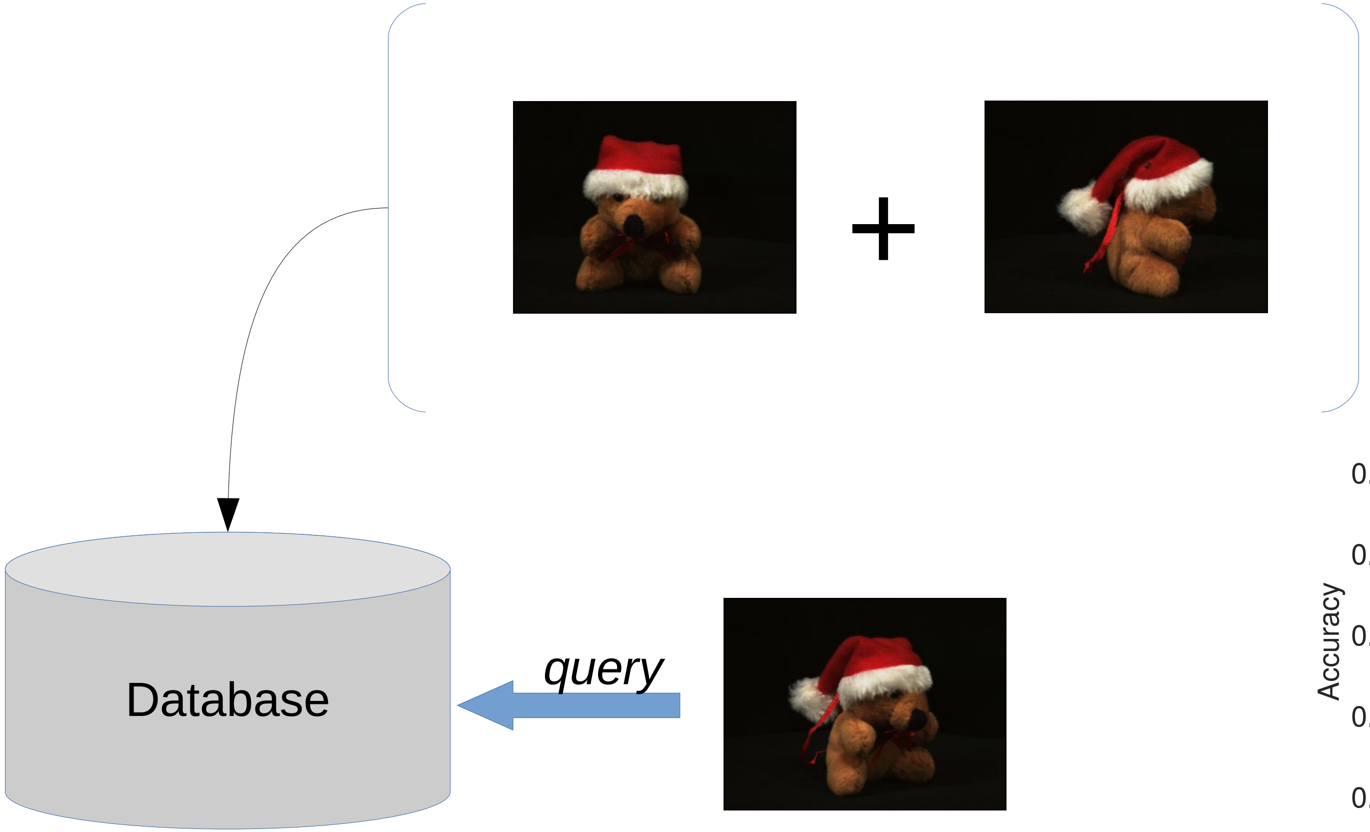
Example application: Object recognition

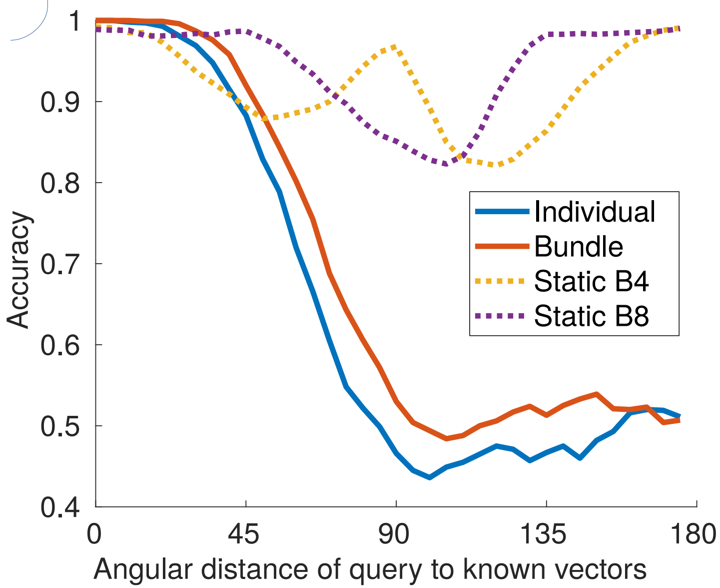


Example application: Object recognition



Example application: Object recognition





Credits: Pentti Kanerva

How to store structured data?

Given are 2 records:

United States of America

Name:	USA
Capital City:	Washington DC
Currency:	Dollar

Mexico

Name:	Mexico
Capital City:	Mexico City
Currency:	Peso

roles *fillers*

Question: What is the Dollar of Mexico?

How to store structured data?

Credits: Pentti Kanerva

Given are 2 records:

United States of America

Name: USA
Capital City: Washington DC
Currency: Dollar

Mexico

Name: Mexico
Capital City: Mexico City
Currency: Peso

roles *fillers*

Question: What is the Dollar of Mexico?

Hyperdimensional computing approach:

1. Assign a **random** high-dimensional vector to each entity
   ”Name” is a random vector NAM
   ”USA” is a random vector USA
   ”Capital city” is a random vector CAP
   …

2. Calculate a **single** high-dimensional vector that contains all information
   F = (NAM*USA+CAP*WDC+CUR*DOL)*(NAM*MEX+CAP*MCX+CUR*PES)

3. **Calculate** the query answer: F*DOL ~ PES

Vector Symbolic Architectures (VSA)

- VSA = high dimensional vector space + operations

Vector Symbolic Architectures (VSA)

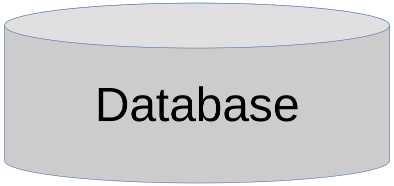
- VSA = high dimensional vector space + operations
- Operations in a VSA:
  - Bind()
  - Bundle()
  - Permute()/Protect()

Pentti Kanerva. 2009. *Hyperdimensional Computing: An Introduction to Computing in Distributed Representation with High-Dimensional Random Vectors*. Cognitive Computation 1, 2 (2009), 139–159. https://doi.org/10.1007/s12559-009-9009-8

Term: Gayler RW (2003) Vector symbolic architectures answer Jackendoff's challenges for cognitive neuroscience. In: Proc. of ICCS/ASCS Int. Conf. on cognitive science, pp 133–138. Sydney, Australia

Vector Symbolic Architectures (VSA)

- VSA = high dimensional vector space + operations
- Operations in a VSA:
  - Bind()
  - Bundle()
  - Permute()/Protect()
- Additionally
  - Encoding/decoding
  - Clean-up memory



Pentti Kanerva. 2009. *Hyperdimensional Computing: An Introduction to Computing in Distributed Representation with High-Dimensional Random Vectors*. Cognitive Computation 1, 2 (2009), 139–159. https://doi.org/10.1007/s12559-009-9009-8

Term: Gayler RW (2003) Vector symbolic architectures answer Jackendoff's challenges for cognitive neuroscience. In: Proc. of ICCS/ASCS Int. Conf. on cognitive science, pp 133–138. Sydney, Australia

Vector Symbolic Architectures (VSA)

- VSA = high dimensional vector space + operations
- Operations in a VSA:
  - Bind()
  - Bundle()
  - Permute()/Protect()
- Additionally
  - Encoding/decoding
  - Clean-up memory

Pentti Kanerva. 2009. *Hyperdimensional Computing: An Introduction to Computing in Distributed Representation with High-Dimensional Random Vectors*. Cognitive Computation 1, 2 (2009), 139–159. https://doi.org/10.1007/s12559-009-9009-8

Term: Gayler RW (2003) Vector symbolic architectures answer Jackendoff's challenges for cognitive neuroscience. In: Proc. of ICCS/ASCS Int. Conf. on cognitive science, pp 133–138. Sydney, Australia

	Smolensky [28]	Plate[26]	Kanerva[13]	Gayler[7]
space $\mathbb{V}$	tensors of real numbers	real and complex vectors	$\{0,1\}^n$	$[-1,1]^n$ (or $\{-1,1\}^n$)
bundle $\oplus$	elementwise sum	elementwise sum	thresholded elementwise sum	limited elementwise sum
bind $\otimes$	tensor product	circular convolution	elementwise XOR	elementwise product
braid $\odot$	(not considered)	(not considered)	permutations	permutations

VSA operations

- Bundling +
  - Goal: combine two vectors, such that
    - the result is **similar** to both vectors
  - Application: superpose information

VSA operations

- Bundling +
  - Goal: combine two vectors, such that
    - the result is **similar** to both vectors
  - Application: superpose information

- Binding⊗
  - Goal: combine two vectors, such that
    - the result is **nonsimilar** to both vectors
    - one can be recreated from the result using the other
  - Application: bind value “a” to variable “x” (or a “filler” to a “role” or ...)

VSA operations

- Binding⊗
  - Properties
    - Associative, commutative
    - Self-inverse: X⊗X=I (or additional *unbind* operator)
    - Result nonsimilar to input

VSA operations

- Binding ⊗
  - Properties
    - Associative, commutative
    - Self-inverse: $X \otimes X = I$ (or additional *unbind* operator)
    - Result nonsimilar to input
  - Example:
    - Hypervector space $\{-1,1\}^D$
    - binding can be elementwise multiplication

VSA operations

- Binding $\otimes$
  - Properties
    - Associative, commutative
    - Self-inverse: $X \otimes X = I$ (or additional *unbind* operator)
    - Result nonsimilar to input
  - Example:
    - Hypervector space $\{-1,1\}^D$
    - binding can be elementwise multiplication
  - Application: bind value “a” to variable “x” (or a “filler” to a “role” or ...)
    - Bind:
      - x = a → $H = X \otimes A$
    - Unbind:
      - x = ? → $X \otimes H$
        $= X \otimes (X \otimes A)$
        $= A$

VSA operations

- Bundling +
  - properties
    - Associative, commutative, binding distributes over bundling
    - Result is similar to both inputs

VSA operations

- Bundling +
  - properties
    - Associative, commutative, binding distributes over bundling
    - Result is similar to both inputs
  - Example
    - Hypervector space $[-1,1]^D$
    - bundling can be elementwise sum, elementwise normalized to [-1,1]

VSA operations

- Bundling +
  - properties
    - Associative, commutative, binding distributes over bundling
    - Result is similar to both inputs
  - Example
    - Hypervector space $[-1,1]^D$
    - bundling can be elementwise sum, elementwise normalized to [-1,1]
  - Application:

    $\{x = a, y = b\} \rightarrow H = X \otimes A + Y \otimes B$

VSA operations

- Bundling +
  - properties
    - Associative, commutative, binding distributes over bundling
    - Result is similar to both inputs
  - Example
    - Hypervector space $[-1,1]^D$
    - bundling can be elementwise sum, elementwise normalized to [-1,1]
  - Application:

    $\{x = a, y = b\} \rightarrow H = X \otimes A + Y \otimes B$

    - Unbind a bundle

    $\{x = a, y = b\} \rightarrow H = X \otimes A + Y \otimes B$

    $$x = ? \rightarrow X \otimes H = X \otimes (X \otimes A + Y \otimes B)$$

    $$= (X \otimes X \otimes A) + (X \otimes Y \otimes B)$$

    $$= A + \text{noise}$$

Teaser application 1: “What is the Dollar of Mexico?”

Credits: Pentti Kanerva

Given are 2 records:

United States of America

Name:	USA
Capital City:	Washington DC
Currency:	Dollar

Mexico

Name:	Mexico
Capital City:	Mexico City
Currency:	Peso

Question: What is the Dollar of Mexico?

Hyperdimensional computing approach:

1. Assign a **random** high-dimensional vector to each entity
   ”Name” is a random vector NAM
   ”USA” is a random vector USA
   ”Capital city” is a random vector CAP
   …

2. Calculate a **single** high-dimensional vector that contains all information
   F = (NAM*USA+CAP*WDC+CUR*DOL)*(NAM*MEX+CAP*MCX+CUR*PES)

3. **Calculate** the query answer: F*DOL ~ PES

Teaser application 1: “What is the Dollar of Mexico?”

Credits: Pentti Kanerva

$$\text{USTATES} = \left[(\text{NAM} * \text{USA}) + (\text{CAP} * \text{WDC}) + (\text{MON} * \text{DOL})\right]$$

$$\text{MEXICO} = \left[(\text{NAM} * \text{MEX}) + (\text{CAP} * \text{MXC}) + (\text{MON} * \text{PES})\right]$$

Teaser application 1: “What is the Dollar of Mexico?”

Credits: Pentti Kanerva

$$\text{USTATES} = [(\text{NAM} * \text{USA}) + (\text{CAP} * \text{WDC}) + (\text{MON} * \text{DOL})]$$

$$\text{MEXICO} = [(\text{NAM} * \text{MEX}) + (\text{CAP} * \text{MXC}) + (\text{MON} * \text{PES})]$$

$$\begin{aligned} F_{UM} &= \text{USTATES} * \text{MEXICO} \\ &= [(\text{USA} * \text{MEX}) + (\text{WDC} * \text{MXC}) \\ &\quad + (\text{DOL} * \text{PES}) + \text{noise}] \end{aligned}$$

Teaser application 1: “What is the Dollar of Mexico?”

Credits: Pentti Kanerva

$$\text{USTATES} = [(\text{NAM} * \text{USA}) + (\text{CAP} * \text{WDC}) + (\text{MON} * \text{DOL})]$$
$$\text{MEXICO} = [(\text{NAM} * \text{MEX}) + (\text{CAP} * \text{MXC}) + (\text{MON} * \text{PES})]$$

$$\begin{aligned} F_{UM} &= \text{USTATES} * \text{MEXICO} \\ &= [(\text{USA} * \text{MEX}) + (\text{WDC} * \text{MXC}) \\ &\quad +(\text{DOL} * \text{PES}) + \text{noise}] \end{aligned}$$

$$\begin{aligned} \text{DOL} * F_{UM} &= \text{DOL} * [(\text{USA} * \text{MEX}) + (\text{WDC} * \text{MXC}) \\ &\quad +(\text{DOL} * \text{PES}) + \text{noise}] \\ &= [(\text{DOL} * \text{USA} * \text{MEX}) \\ &\quad +(\text{DOL} * \text{WDC} * \text{MXC}) \\ &\quad +(\text{DOL} * \text{DOL} * \text{PES}) + (\text{DOL} * \text{noise})] \\ &= [\text{noise}_1 + \text{noise}_2 + \text{PES} + \text{noise}_3] \\ &= [\text{PES} + \text{noise}_4] \\ &\approx \text{PES} \end{aligned}$$