COMP309/AIML421 — ML Tools and Techniques

Week 1

Machine Learning Overview

Dr Qi Chen

School of Engineering and Computer Science

Victoria University of Wellington

Qi.Chen@vuw.ac.nz

Week Overview

★ AI and Machine Learning

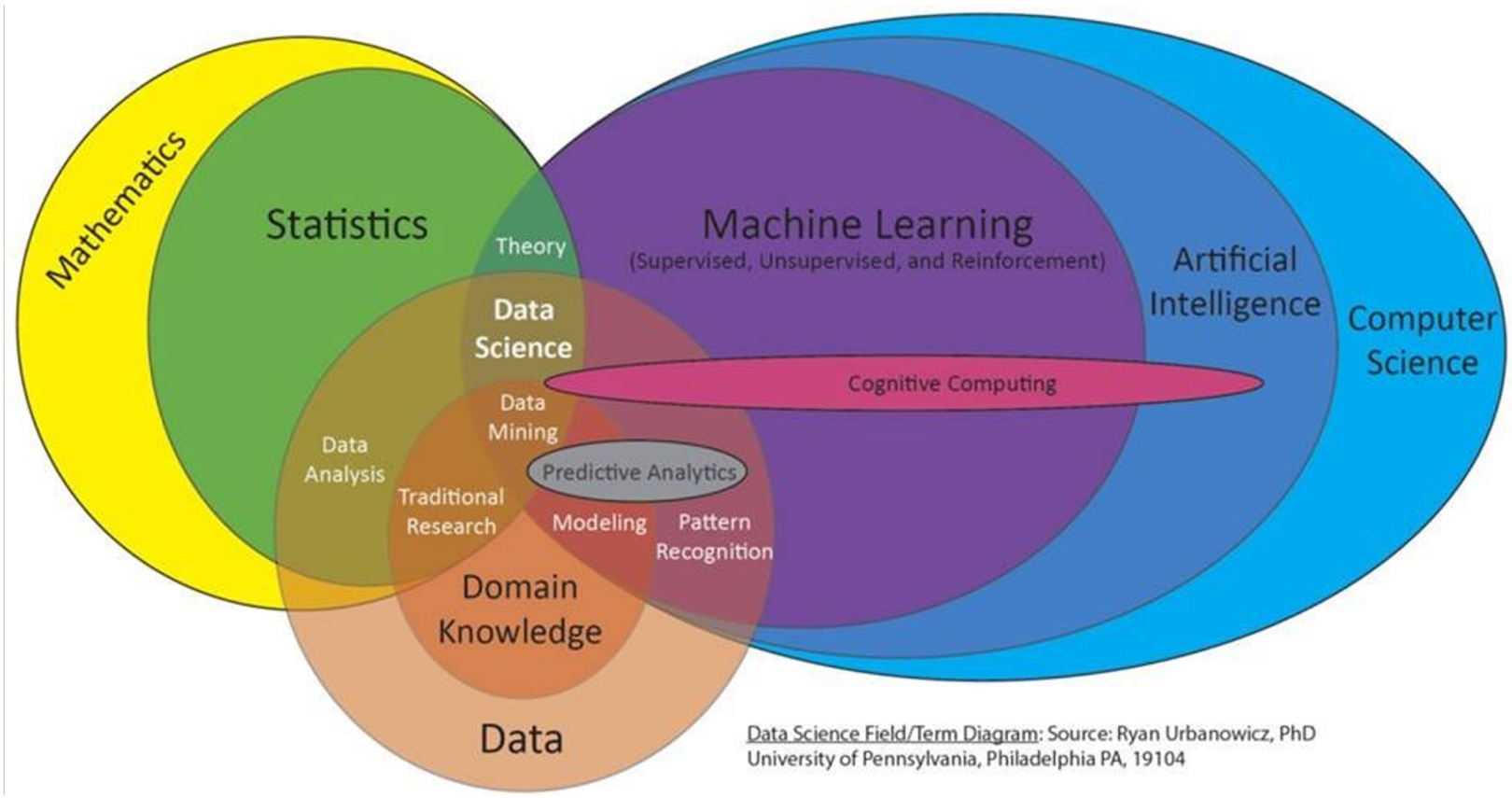
★ Machine Learning Scope: Data, Task, Model, and Algorithm

★ Data in Machine Learning

★ Machine Learning Tasks

Artificial Intelligence and Machine Learning

- Artificial Intelligence is the broader concept of machines being able to carry out tasks in a way that we would consider “smart”.
- Machine Learning is a current application of AI based around the idea that giving machines access to data and let them learn for themselves.



Data Science Field/Term Diagram: Source: Ryan Urbanowicz, PhD
University of Pennsylvania, Philadelphia PA, 19104

Machine Learning

- science of getting computers to act without being explicitly programmed.

- a branch of artificial intelligence focuses on the development of algorithms and statistical models

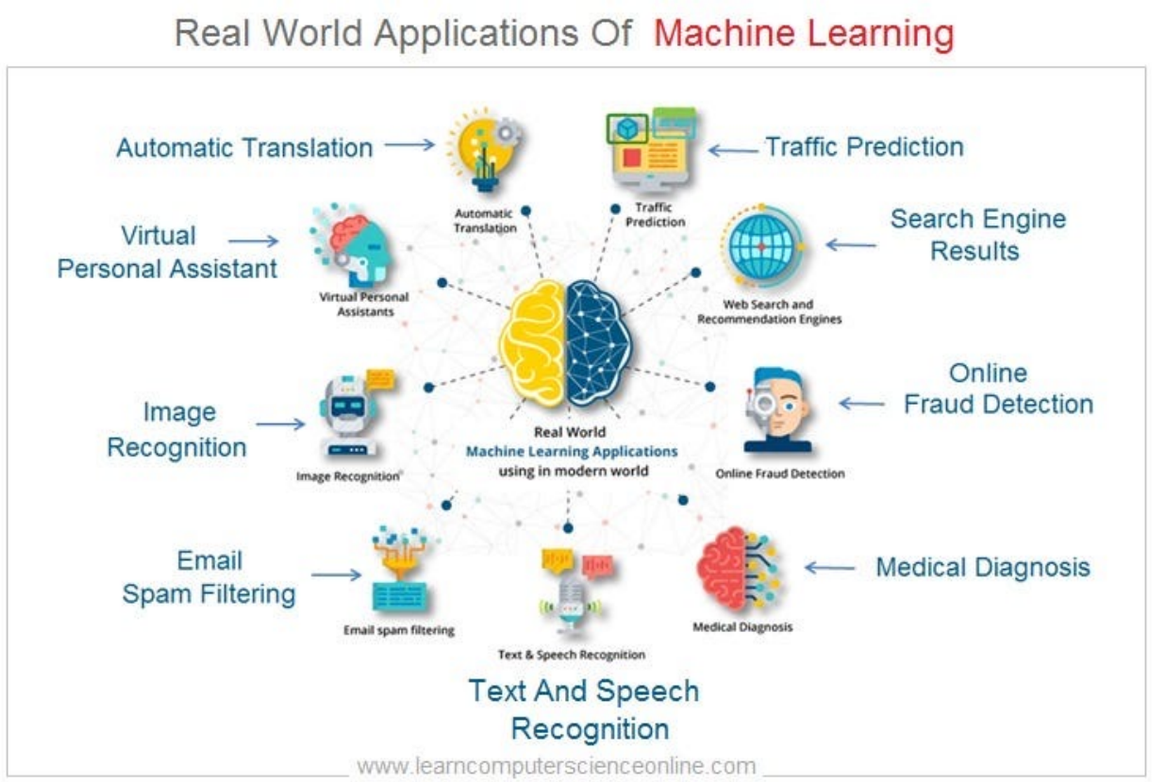
- based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention.

- a method of data analysis that automates analytical model building.

Machine Learning Applications

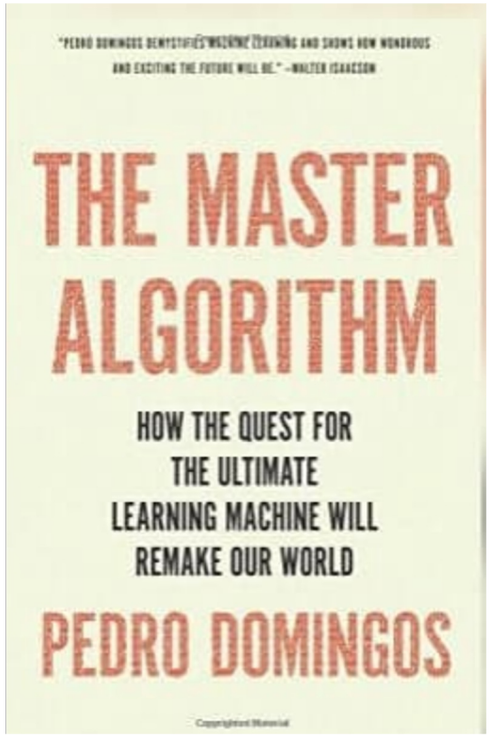
- In the past decade, machine learning has given us self-driving cars, speech recognition, effective web search, and a vastly improved understanding of the human genome

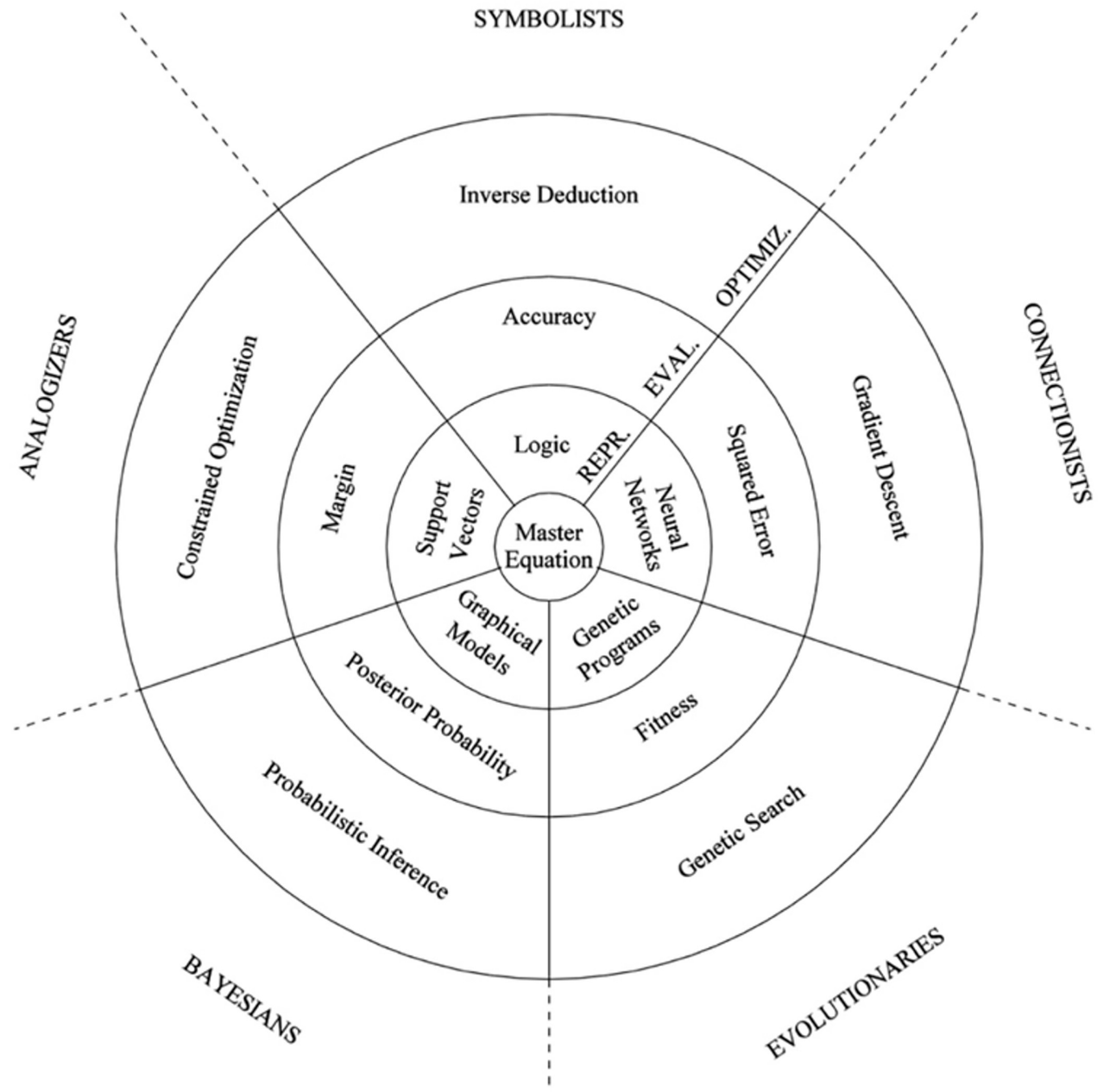
- Machine learning is so pervasive today that you probably use it dozens of times a day without knowing it.



https://tharalla-anil555.medium.com/applications-of-machine-learning-24507c72d47d

Five Tribes of Machine Learning





Domingos, Pedro. "A few useful things to know about machine learning." *Communications of the ACM* 55, no. 10 (2012): 78-87.

The Scope of Machine Learning

ML involves a *wide variety* of each of these:

- Data: foundational input of machine learning algorithms
- Task: specific problems that machine learning try to solve
- Model: mathematical frameworks/structures that interpret the data and perform the tasks
- Algorithm: the step-by-step procedures or rules followed by the models to learn from the data

The first 3 weeks --

- Today we'll address a couple of aspects of data.
- Next lecture: the tasks in Machine Learning
- Then: on to the most common models and algorithms (in overview only)

Data

- datahub.io
- openml.org

e.g.

- iris -- https://www.openml.org/d/61
- penguins -- https://www.openml.org/d/42585
- diabetes -- https://www.openml.org/d/37
- banknotes -- https://www.openml.org/d/1462
- …

Data

- eg: banknotes -- https://www.openml.org/d/1462
- someone having a go at clustering with this data: https://towardsdatascience.com/k-means-clustering-project-banknote-authentication-289cfe773873
- there are many blogs / tutorials / notebooks out there doing similar
- there are lots of new tools all the time, but near-generic tools at the moment:
  - python
  - numpy
  - sklearn
  - pandas
  - matplotlib
  - jupyter notebooks

Data

- Consider banknotes.csv:
- V1-V4 are values of 4 “features”
- the Class is 1 (legit) or 2 (forged)
- Common to talk separately about X and Y:

V1	V2	V3	V4	Class
3.6216	8.6661	-2.8073	-0.44699	1
4.5459	8.1674	-2.4586	-1.4621	1
3.866	-2.6383	1.9242	0.10645	1
3.4566	9.5228	4.0112	-3.5944	1
0.32924	-4.4552	4.5718	-0.9888	1
4.3684	9.6718	3.9606	-3.1625	1
3.5912	3.0129	0.72888	0.56421	1
2.0922	-6.81	8.4636	-0.60216	1
3.2032	5.7588	-0.75345	-0.61251	1
1.5356	9.1772	-2.2718	-0.73535	1
1.2247	8.7779	-2.2135	-0.80647	1

X

Y

~1300 rows in this case. The “Class=2” ones are further down.

Data as “Vectors” in a “Space”

- each row is one data item, here consisting of a pairing:

V1	V2	V3	V4	Class
3.6216	8.6661	-2.8073	-0.44699	1

$$X \rightarrow y$$

- we might talk about **X** = (x_0, x_1, x_2, x_3) instead of the names v_1, etc specific to this dataset.
- ***X*** is a 4-dimensional *vector*
- x_i is the value for the i^{th} *dimension*

Data as points in a space

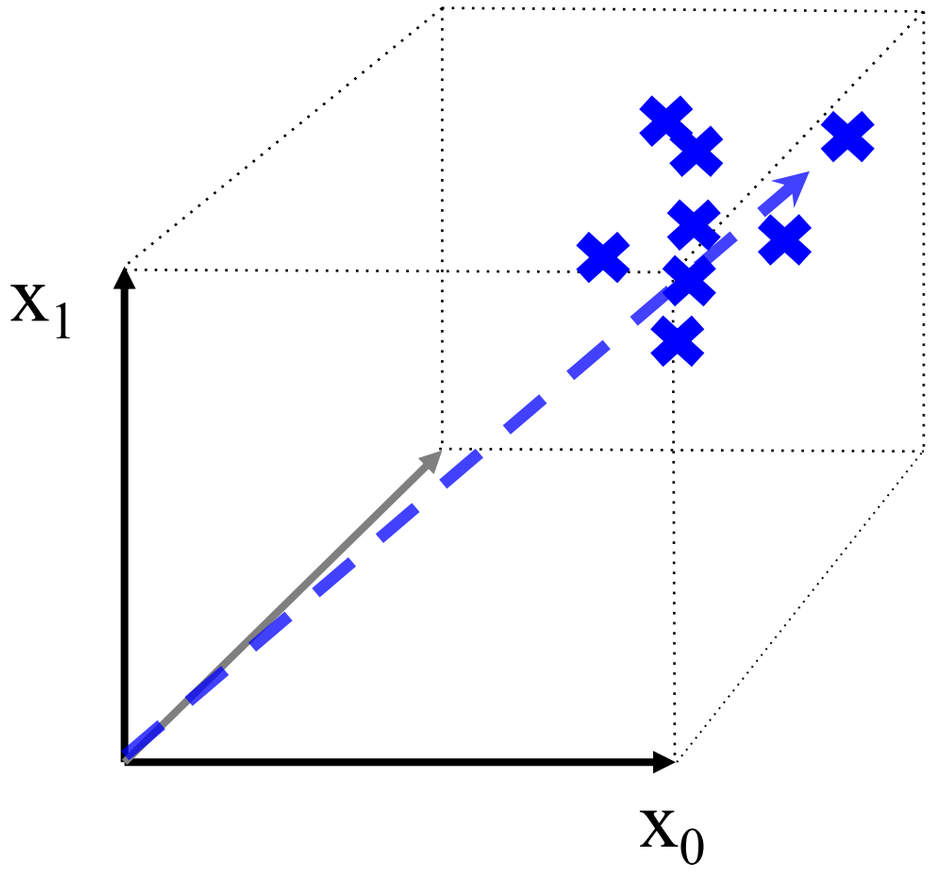
A “row” can be thought of as a “point” in a “space” of data

It’s easy to visualise when dimensionality is low:

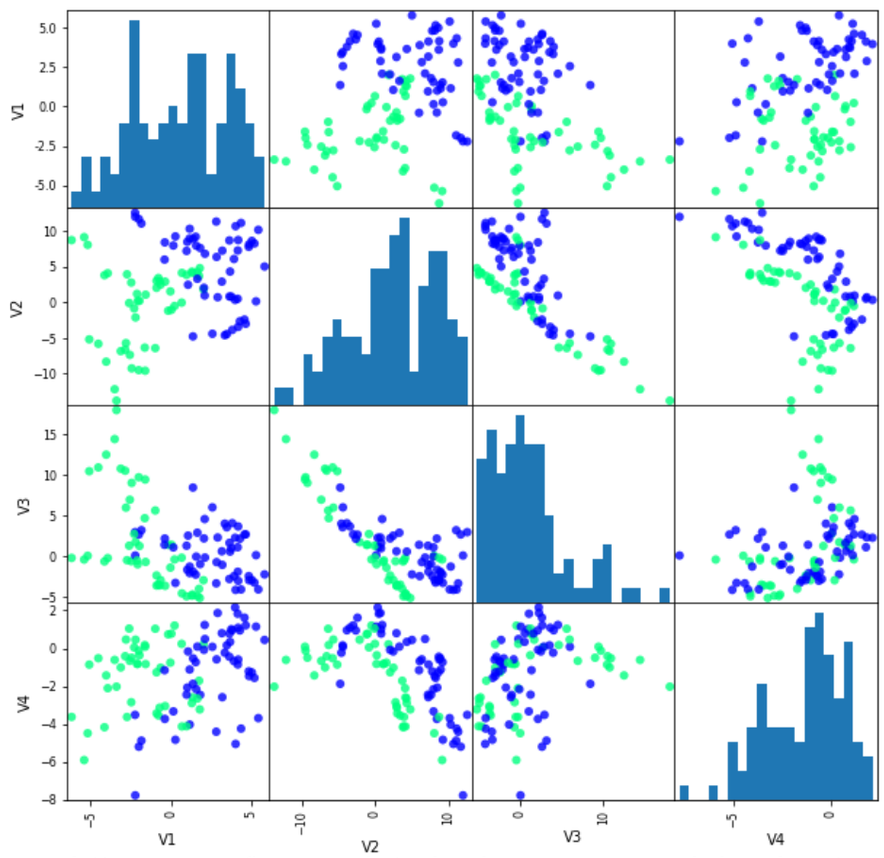
- 1 dimension, e.g. just V1
- 2 dimensions, easy
- 3 dimensions, easy

V1	V2	V3	V4	Class
3.6216	8.6661	-2.8073	-0.44699	1
4.5459	8.1674	-2.4586	-1.4621	1
3.866	-2.6383	1.9242	0.10645	1
3.4566	9.5228	-4.0112	-3.5944	1
0.32924	-4.4552	4.5718	-0.9888	1
4.3684	9.6718	-3.9606	-3.1625	1
3.5912	3.0129	0.72888	0.56421	1
2.0922	-6.81	8.4636	-0.60216	1
3.2032	5.7588	-0.75345	-0.61251	1
1.5356	9.1772	-2.2718	-0.73535	1
1.2247	8.7779	-2.2135	-0.80647	1

Data as Points in a Vector Space



V1	V2	V3	V4	Class
3.6216	8.6661	-2.8073	-0.44699	1
4.5459	8.1674	-2.4586	-1.4621	1
3.866	-2.6383	1.9242	0.10645	1
3.4566	9.5228	-4.0112	-3.5944	1
0.32924	-4.4552	4.5718	-0.9888	1
4.3684	9.6718	-3.9606	-3.1625	1
3.5912	3.0129	0.72888	0.56421	1
2.0922	-6.81	8.4636	-0.60216	1
3.2032	5.7588	-0.75345	-0.61251	1
1.5356	9.1772	-2.2718	-0.73535	1
1.2247	8.7779	-2.2135	-0.80647	1



https://sporf.neurodata.io/demos/openml_banknote

Humans can’t see in more than 3d

Some of our intuitions hold, some fail

Data in High dimensions

roughly speaking,

it's all "in the corners"

it's all "at right-angles"

it's all "equally far apart"

```
1 for dims in range(1,21):
2     print(dims, countInside(1000000,dims))
```

```
1 1000000
2 785776
3 523488
4 307377
5 164873
6 80816
7 37145
8 15879
9 6370
10 2581
11 915
12 324
13 113
14 41
15 11
16 3
17 1
18 0
19 0
20 0
```

none

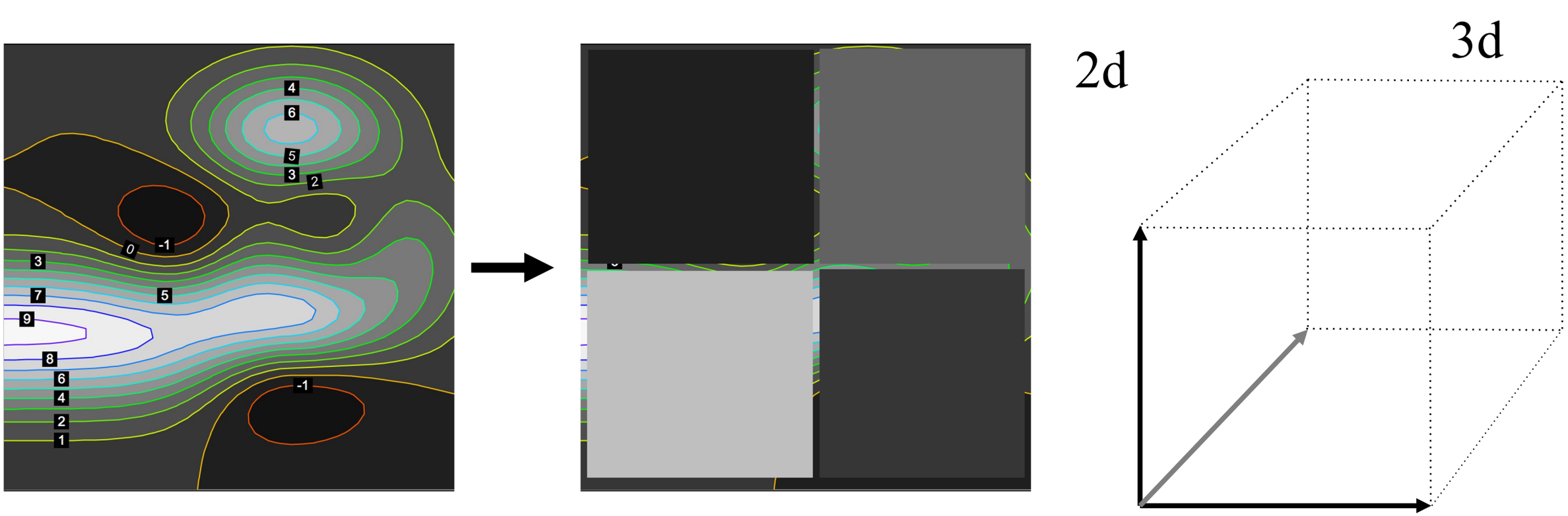
Eg: I made a million data points that were *d*-dimensional, with each dimension being randomly chosen in range -1 to +1. i.e. in a "box" of side 2.

A ball with diameter 2 fits snugly inside this box. Out of a million random points, how many land inside the ball?

The Curse of Dimensionality

consider a vector space that is bounded within 0 to 1 for each dimension.

I decide to just brutally summarise / compress the function, by giving just *one value per "quadrant". Eg. in 2d, I need 4 numbers.*



how many numbers do I need, in d-dimensions?