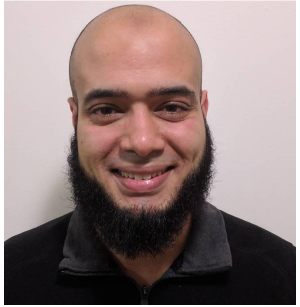
Machine Learning

Common ML Jargons

Mostafa S. Ibrahim
Teaching, Training and Coaching for more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / MSc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)



Occam’s Razor

- A principle by a philosopher. In machine learning context what matters is:
- Given two models with the same generalization error, the **simpler** one should be preferred because **simplicity is desirable** in itself
    - Simpler models are supposed to have **less assumptions**
        - This implies the simpler model might have better real generalization
        - They are also faster / use less memory / has better explainability
    - With more assumptions, we are subjective to more constraints / limitations
- This raises up a question: How can we measure model’s complexity?
    - Relevant concept: VC dimension: a model capacity measurement
- In practice, we use many complex deep learning and ensemble models

No Free Lunch (NFL) Theorem

- No **single** machine learning algorithm is **universally** the best-performing algorithm for **ALL** problems
- Why?
  - Every supervised algorithm makes **prior assumptions** about the input/output relationships
  - From a problem to another, the *assumptions will be wrong*
- **In practice**: we may need to try different algorithms that could work for this specific problem
  - Some people might think tools like Deep Learning or XGboost can work for everything
  - Although deep learning has several success scenarios, you don't hear about failures
  - XGboost for example fails in extrapolation tasks like stock prediction

Curse of Dimensionality (CoD)

- Assume our input has 2 integer features of range [1, 100]
- How many examples can cover all possible cases?
- What about 10 features? 500 features?

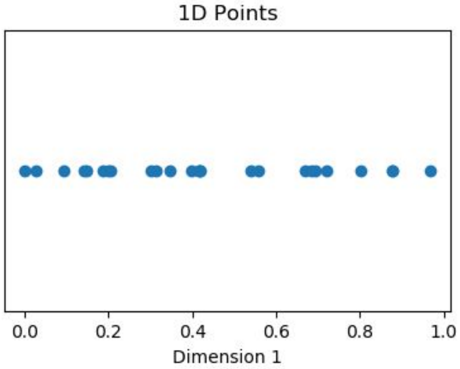
- 100^D where D is the number of features
- This is an exponential growth
- With high D, even collecting millions of examples, our dataset will be a tiny fraction of all the possible combinations!
- Every added dimension adds a big challenge (a curse)
  - We need more data to cover more scenarios!
  - We need algorithms that works well with high dimensionality data!

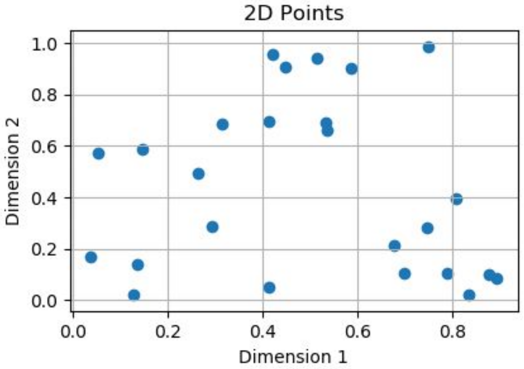
Curse of Dimensionality (CoD)

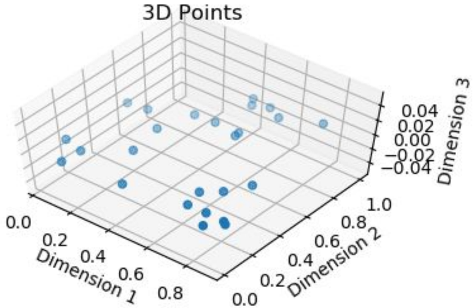
- Imagine having 10 examples with 2 features each of range [1-10]
  - Assume we computed euclidean distance between every pair of examples
- Imagine we extended them with other 2 features
- Imagine we extended them with other 100 features
- With more added features for the examples, what do you expect will happen for the computed distance? The same? Increasing rapidly?

Curse of Dimensionality (CoD)

- The distance will keep increasing
- This means our examples that can be close in small dimensionality will be too far from each in high dimensionality! The geometric scene will look **sparse**!
  - More intensive data is required to make the overall less sparse
- But our ML should find patterns and such setup make it so far
  - Bunch of very separated points: No smoothness or continuity







Img src

Curse of Dimensionality (CoD)

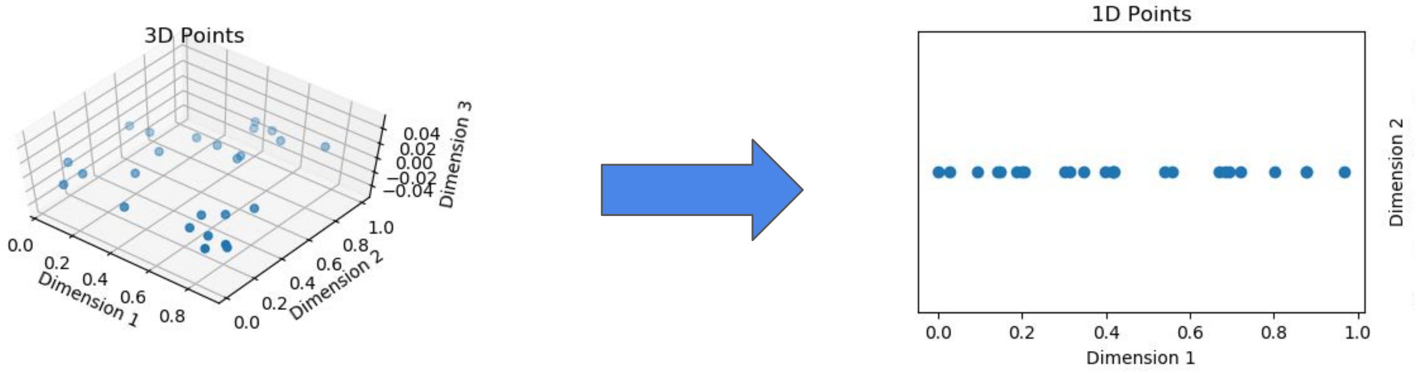
- Definition: It is a **data property**: when the **dimensionality** of the data **increases**, the **sparsity** of the data **increases**
  - As a result, we need to collect exponential number of training samples to cover this huge space, but this is **impossible**!
  - Also, the distance between points is increasing
  - It is about the data NOT the model
- Every machine learning algorithm suffers from the curse to some extent
  - Some in a severe way: e.g. K-NN, K-Means (*neighbor-driven* algorithms)
  - Some in a medium way: e.g. Decision Trees
  - Some in a limited way such as Deep Learning
- Still, how can we tackle this challenge?

Manifold Hypothesis

- A **hypothesis** is an **assumption** that is made based on *some evidence*.
  - In research, we start from a hypothesis and do further investigations
- The **Manifold Hypothesis** states that real-world high-dimensional data **lie** on low-dimensional manifolds embedded within the high-dimensional space
- Let’s simplify for now with a concrete example:
  - Imagine we have feature vector of e.g. 10,000 features
    - E.g. 100x100 binary image that a digit from 0 to 9
  - There is a corresponding representative vector of e.g. 64 features (let’s call it **embedding**)
  - If we have this representative vector, we can use it smoothly with ML algorithms!
  - The question how **can we transform** from this input vector to the transformed one!

CoD and Dimensionality Reduction

- Dimensionality Reduction: transformation of data from a **high-dimensional** space into a **low-dimensional** space so that the low-dimensional representation retains some meaningful properties of the original data
- Now, use this reduced vector with different algorithms (e.g. K-means)



CoD and Deep Learning

- Deep Learning Networks, with proper design, are working pretty well with several high dimensional data (In images, speech, text, etc)
- In deep network, we do many consecutive and complex transformations of the input data
- Although the transformed data might still in high dimensions, it seems **semantically similar points are grouped closer together**
  - E.g. Cat images are close from each others
  - So still sparse, but not an issue
- It is still an active research to reveal the reasons behind the performance. Seems the **locality** in processing is the key

Relevant resources

- Occam's Razor: Article, Article
- Theory vs Theorem: Link
- No Free Lunch: Article
- Curse of Dimensionality: Article
- Manifold Hypothesis: Article, Slides
- CoD and Deep learning: Paper, Paper, Article

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."