

What Is the Curse of Dimensionality?

Machine learning excels at analyzing data with many dimensions, but it becomes more challenging to create meaningful models as the number of dimensions increase.



Written by [Badreesh Shetty](#)



Image: Shutterstock

ILLUSTRATED BY

FOR EMPLOYERS

In machine learning, we often have high-dimensional data. If we're recording 60 different metrics for each of our shoppers, we're working in a space with 60 dimensions. If we're analyzing grayscale images sized 50x50, we're working in a space with 2,500 dimensions. If the images are RGB-colored, the dimensionality increases to 7,500 dimensions (one dimension for each color channel in each pixel in the image).

WHAT IS THE CURSE OF DIMENSIONALITY?

Coined by mathematician Richard E. Bellman, the curse of dimensionality references increasing data dimensions and its explosive tendencies. This phenomenon typically results in an increase in computational efforts required for its processing and analysis.

Curse of
DIMENSIONALITY

CURSE OF DIMENSIONALITY

As the dimensionality of the features space increases, the number of configurations can grow exponentially, and thus the number of configurations covered by an observation decreases.

Chris Albon

Regarding the curse of dimensionality — also known as the Hughes Phenomenon — there are two things to consider. On the one hand, ML excels at analyzing data with many dimensions. Humans are not good at finding patterns that may be spread out across so many dimensions, especially if those dimensions are interrelated in counter-intuitive ways. On the other hand, as we add more dimensions we also increase the processing power we need to analyze the data, and we also increase the amount of training data required to make meaningful data models.

HIGH DIMENSIONAL DATA

High dimensional data is when a dataset a number of features (p) that is bigger than the number of observations (N). High dimensional data is the problem that leads to the curse of dimensionality. The equation for high dimensional data is usually written like $p \gg N$.

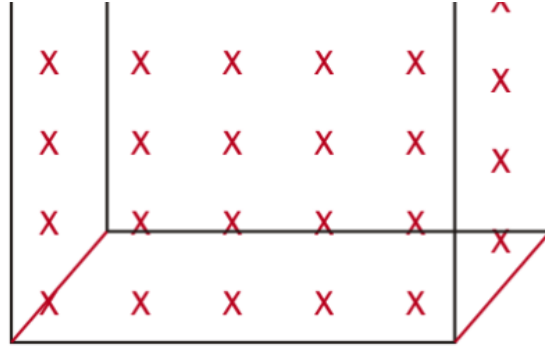
X X X X X

A one-dimensional features space with five data points



A two-dimensional features space with 25 data points

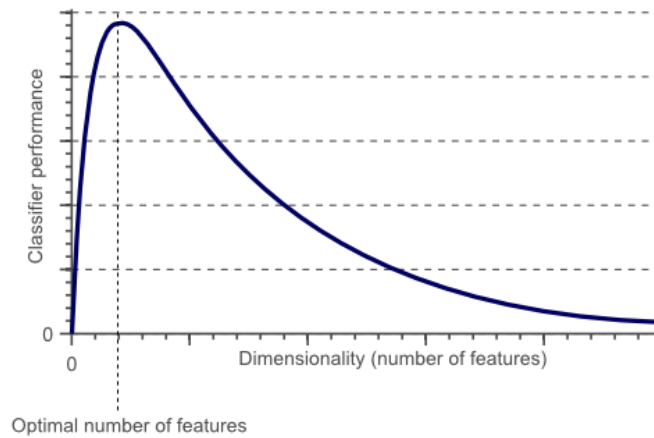




A three-dimensional feature space with 125 data points

Hughes Phenomenon

The Hughes Phenomenon shows that as the number of features increases, the classifier's performance increases as well until we reach the optimal number of features. Adding more features based on the same size as the training set will then degrade the classifier's performance.



Curse of Dimensionality in Distance Function

An increase in the number of dimensions of a [dataset](#) means there are more entries in the vector of features that represents each observation in the corresponding Euclidean space. We measure the distance in a vector space using Euclidean distance.

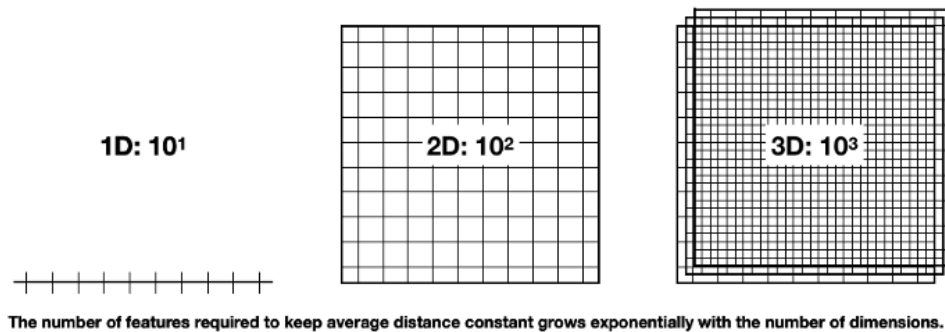
The Euclidean distance between 2-dimensional vectors with Cartesian coordinates $p = (p_1, p_2, \dots, p_n)$ and $q = (q_1, q_2, \dots, q_n)$ is computed using the familiar formula:

$$d(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

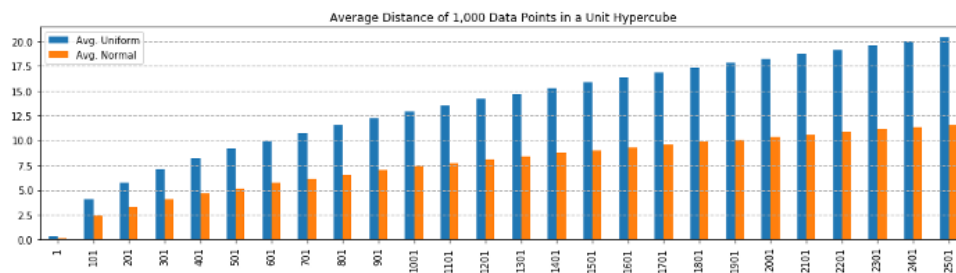
Hence, each new dimension adds a non-negative term to the sum, so the distance increases with the number of dimensions for distinct vectors. In other words, as the number of features grows for a given number of observations, the feature space becomes increasingly sparse; that is, less dense or emptier. On the flip side, the lower data density requires more observations to

keep the average distance between data points the same.

Similar to the above figures, below shows how many data points we need to maintain the average distance of 10 observations uniformly distributed on a line. It increases exponentially from 10^1 in a single dimension to 10^2 in two and 10^3 in three dimensions, as the data needs to expand by a factor of 10 each time we add a new dimension:



The figure below simulates how the average and minimum distances between data points increase as the number of dimensions grows:



From the above figure, it is proven that with the increase in dimensions, mean distance increases rapidly. Hence the higher the dimensions, the more data is needed to overcome the curse of dimensionality!

When the distance between observations grows, supervised machine learning becomes more difficult because predictions for new samples are less likely to be based on learning from similar training features. The number of possible unique rows grows exponentially as the number of features increases, which makes it so much harder to efficiently generalize. The variance increases as they get more opportunity to overfit to noise in more dimensions, resulting in poor generalization performance.

In practice, features are correlated or do not exhibit much variation. For these reasons, dimensionality reduction helps compress the data without losing much of the signal, and combat the curse while also economizing on memory.

Overfitting and Underfitting

KNN is very susceptible to overfitting due to the curse of dimensionality. Curse of dimensionality also describes the phenomenon where the feature space becomes increasingly sparse for an increasing number of dimensions of a fixed-size training dataset. Intuitively, we can think of even the closest neighbors being too far away in a high-dimensional space to give a good estimate. Regularization is one way to avoid overfitting. However, in models where regularization is not applicable, such as decision trees and KNN, we can use feature selection

and dimensionality reduction techniques to help us avoid the curse of dimensionality.

Curse of Dimensionality Overfitting

Overfitting occurs when a model starts to memorize the aspects of the training set and in turn loses the ability to generalize. Image: Chris Albon

This notion is closely related to the problem of overfitting. As our training data is not enough, we risk producing a model that could be very good at predicting the target class on the training dataset but fail miserably when faced with new data. That is, our model does not have the generalization power. That is why it's so important to evaluate our methods on previously unseen data.

The general rule to avoid overfitting is to preference simple (i.e., less parameters) methods, something that could be seen as an instantiation of the philosophical principle of Occam's Razor, which states that among competing hypotheses, the hypothesis with the fewest assumptions should be selected.

Curse of Dimensionality Occam's Razor

When multiple hypotheses explain something equally well, we should prefer the simplest one. Image: Chris Albon

However, we should also take into account Einstein’s words: *“Everything should be made as simple as possible, but not simpler.”*

The idem curse of dimensionality may suggest that we keep our models simple, but on the other hand, if our model is too simple we run the risk of suffering from underfitting. Underfitting problems arise when our model has such a low representation power that it cannot model the data even if we had all the training data we want. We clearly have underfitting when our algorithm cannot achieve good performance measures even when measuring on the training set.

Curse of Dimensionality Underfitting

Underfitting is when a model fails to capture the pattern in the data. Image: Chris Albon

Curse of Dimensionality overfit vs underfit

A graph visualization of the differences between overfit and underfit. Image: Chris Albon

As a result, we will have to achieve a balance between overfitting and underfitting. This is one of the most important problems to address when designing our machine learning models.

Curse of Dimensionality Case:

<https://nbviewer.jupyter.org/gist/BadreeshShetty/bf9cb1dced8263ef997bcb2c3926569b>

OR

<https://gist.github.com/BadreeshShetty/bf9cb1dced8263ef997bcb2c3926569b>

From looking at the above case, it is proven that with increase in dimensions, mean distance increases logarithmically. Hence higher the dimensions, more data is needed to overcome the curse of dimensionality!

Read more content from Badreesh Shetty:

[An In-Depth Guide to Supervised Machine Learning Classification](#)

[An In-Depth Guide to How Recommender Systems Work](#)

RELATED

[Read More About Data Science](#)

Subscribe to Built In to get tech articles + jobs in your inbox.

Your Expertise



SUBSCRIBE

RECENT DATA SCIENCE ARTICLES

Should You Build or Buy AI?

Here's How to Take Your FinOps Game to the Next Level

90 Data Science Interview Questions to Know

Data Science

Expert Contributors



Expert Contributors

Built In's expert contributor network publishes thoughtful, solutions-oriented stories written by innovative tech professionals. It is the tech industry's definitive destination for sharing compelling, first-person accounts of problem-solving on the road to innovation.

LEARN MORE

Great Companies Need Great People. **That's Where We Come In.**

RECRUIT WITH US

BuiltIn

Built In is the online community for startups and tech companies. Find startup jobs, tech news and events.

About

Our Story

Careers

Our Staff Writers

Content Descriptions

Company News

Get Involved

Recruit With Built In

Become an Expert Contributor

Send Us a News Tip

Resources

Customer Support

Share Feedback

Report a Bug

Tech A-Z

Browse Jobs

Tech Hubs

Built In Austin

Built In Boston

Built In Chicago

Built In Colorado

Built In LA

Built In NYC

Built In San Francisco

Built In Seattle

See All Tech Hubs

