Table of Contents

[**1.** **Supervised Learning Algorithms** 1](#_Toc449026465)

[1. Regression 2](#_Toc449026466)

[2. Classification 2](#_Toc449026467)

[1. K-nearest 2](#_Toc449026468)

[2. Support Vector Machine 3](#_Toc449026469)

[**3.** **Unsupervised Learning Algorithms** 4](#_Toc449026470)

[**4.** **Statistics Concepts** 4](#_Toc449026471)

[**5.** **Machine Learning Concepts** 4](#_Toc449026472)

[**1.** **Sample Study** 4](#_Toc449026473)

[**1.** **Curse of dimensionality** 4](#_Toc449026474)

[2. Data Standardization 5](#_Toc449026475)

# **Supervised Learning Algorithms**

## Regression

## Classification

### K-nearest

**Sample Study**

The **curse of dimensionality** refers to various phenomena that arise when analyzing and organizing data in [high-dimensional spaces](https://en.wikipedia.org/wiki/High-dimensional_space) (often with hundreds or thousands of dimensions) that do not occur in low-dimensional settings such as the [three-dimensional](https://en.wikipedia.org/wiki/Three-dimensional_space) [physical space](https://en.wikipedia.org/wiki/Physical_space) of everyday experience

There are multiple phenomena referred to by this name in domains such as [numerical analysis](https://en.wikipedia.org/wiki/Numerical_analysis), [sampling](https://en.wikipedia.org/wiki/Sampling_(statistics)), [combinatorics](https://en.wikipedia.org/wiki/Combinatorics), [machine learning](https://en.wikipedia.org/wiki/Machine_learning), [data mining](https://en.wikipedia.org/wiki/Data_mining), and [databases](https://en.wikipedia.org/wiki/Database). The common theme of these problems is that when the dimensionality increases, the [volume](https://en.wikipedia.org/wiki/Volume) of the space increases so fast that the available data become sparse. This sparsity is problematic for any method that requires statistical significance. In order to obtain a statistically sound and reliable result, the amount of data needed to support the result often grows exponentially with the dimensionality.

The "curse of dimensionality" is not a problem of high-dimensional data, but a joint problem of the data and the algorithm being applied[*[citation needed](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed" \o "Wikipedia:Citation needed)*]. It arises when the algorithm does not scale well to high-dimensional data, typically due to needing an amount of time or memory that is [exponential](https://en.wikipedia.org/wiki/Exponential_growth) in the number of dimensions of the data.

When facing the curse of dimensionality, a good solution can often be found by changing the algorithm, or by pre-processing the data into a lower-dimensional form. For example, the notion of [intrinsic dimension](https://en.wikipedia.org/wiki/Intrinsic_dimension) refers to the fact that any low-dimensional data space can trivially be turned into a higher-dimensional space by adding redundant (e.g. duplicate) or randomized dimensions, and in turn many high-dimensional data sets can be reduced to lower-dimensional data without significant information loss. This is also reflected by the effectiveness of [dimension reduction](https://en.wikipedia.org/wiki/Dimension_reduction) methods such as [principal component analysis](https://en.wikipedia.org/wiki/Principal_component_analysis) in many situations. Algorithms that are based on distance functions or nearest neighbor search can also work robustly on data having many spurious dimensions[*[citation needed](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed" \o "Wikipedia:Citation needed)*], depending on the statistics of those dimensions.

### Support Vector Machine

Support vector machines (SVMs), a method for the classiﬁcation of both linear and nonlinear data. In a nutshell, an SVM is an algorithm that works as follows. It uses a nonlinear mapping to transform the original training data into a higher dimension. Within this new dimension, it searches for the linear optimalseparatinghyperplane(i.e.,a“decisionboundary”separatingthetuplesofoneclass from another). With an appropriate nonlinear mapping to a sufﬁciently high dimension, data from two classes can always be separated by a hyperplane. The SVM ﬁnds this hyperplane using support vectors (“essential” training tuples) and margins (deﬁned by the support vectors).

# **Unsupervised Learning Algorithms**

# **Statistics Concepts**

# **Machine Learning Concepts**

## **Sample Study**

### **Curse of dimensionality**

The **Curse of dimensionality** refers to various phenomena that arise when analyzing and organizing data in [high-dimensional spaces](https://en.wikipedia.org/wiki/High-dimensional_space) (often with hundreds or thousands of dimensions) that do not occur in low-dimensional settings such as the [three-dimensional](https://en.wikipedia.org/wiki/Three-dimensional_space) [physical space](https://en.wikipedia.org/wiki/Physical_space) of everyday experience

There are multiple phenomena referred to by this name in domains such as [numerical analysis](https://en.wikipedia.org/wiki/Numerical_analysis), [sampling](https://en.wikipedia.org/wiki/Sampling_(statistics)), [combinatorics](https://en.wikipedia.org/wiki/Combinatorics), [machine learning](https://en.wikipedia.org/wiki/Machine_learning), [data mining](https://en.wikipedia.org/wiki/Data_mining), and [databases](https://en.wikipedia.org/wiki/Database). The common theme of these problems is that when the dimensionality increases, the [volume](https://en.wikipedia.org/wiki/Volume) of the space increases so fast that the available data become sparse. This sparsity is problematic for any method that requires statistical significance. In order to obtain a statistically sound and reliable result, the amount of data needed to support the result often grows exponentially with the dimensionality.

The "curse of dimensionality" is not a problem of high-dimensional data, but a joint problem of the data and the algorithm being applied[*[citation needed](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed" \o "Wikipedia:Citation needed)*]. It arises when the algorithm does not scale well to high-dimensional data, typically due to needing an amount of time or memory that is [exponential](https://en.wikipedia.org/wiki/Exponential_growth) in the number of dimensions of the data.

When facing the curse of dimensionality, a good solution can often be found by changing the algorithm, or by pre-processing the data into a lower-dimensional form. For example, the notion of [intrinsic dimension](https://en.wikipedia.org/wiki/Intrinsic_dimension) refers to the fact that any low-dimensional data space can trivially be turned into a higher-dimensional space by adding redundant (e.g. duplicate) or randomized dimensions, and in turn many high-dimensional data sets can be reduced to lower-dimensional data without significant information loss. This is also reflected by the effectiveness of [dimension reduction](https://en.wikipedia.org/wiki/Dimension_reduction) methods such as [principal component analysis](https://en.wikipedia.org/wiki/Principal_component_analysis) in many situations. Algorithms that are based on distance functions or nearest neighbor search can also work robustly on data having many spurious dimensions[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)], depending on the statistics of those dimensions.

### Data Standardization

A **standardized variable** (sometimes called a z-score or a standard score) is a **variable** that has been rescaled to have a **mean** of zero and a standard deviation of one

In [statistics](https://en.wikipedia.org/wiki/Statistics), the **standard score** is the signed number of [standard deviations](https://en.wikipedia.org/wiki/Standard_deviation) an observation or [datum](https://en.wikipedia.org/wiki/Data) is *above* the[mean](https://en.wikipedia.org/wiki/Mean). A positive standard score indicates a datum above the mean, while a negative standard score indicates a datum below the mean. It is a [dimensionless quantity](https://en.wikipedia.org/wiki/Dimensionless_number) obtained by subtracting the [population mean](https://en.wikipedia.org/wiki/Population_mean) from an individual[raw score](https://en.wikipedia.org/wiki/Raw_score) and then dividing the difference by the [population](https://en.wikipedia.org/wiki/Statistical_population) [standard deviation](https://en.wikipedia.org/wiki/Standard_deviation). This conversion process is called **standardizing**

Other explanation

If you knew the mean and standard deviation of a population for a particular variable, than you can compute the probabilities associated with a particular value of that variable within that population. The problem is, to do so, you have to use a really long equation that involves math and stuff, and if you’re reading this, chances are you’re not a big fan of math. I know I’m not. What we need, then, is a simpler way to get those probabilities. And it turns out there is just such a way: a **standardized normal distribution**. What’s the standard normal distribution, you ask? Well, it’s quite simple: it’s a normal distribution with a mean of 0 and a standard deviation of 1. And because we know the properties of the normal distribution, we can convert any particular normal distribution into a standardized normal distribution with a simple equation:

*Z = (X – μ)/σ*

That is, the z-score for any particular value of the variable X is X minus the mean (μ) for X, divided by the standard deviation (σ) of X. If you do this for all of the scores, you’ll get the standardized normal distribution. So if you choose a value of X equal to the mean, then you will subtract the mean from itself, which yeilds 0. Thus, the mean in the standard normal distribution is z = 0. If you choose a score that is one standard deviation above the mean and subtract the mean from it, then what you’ll have left is the standard deviation. When you then divide the standard deviation from that (thus, from itself), you’ll have z = 1. If you pick a score that’s 1.5 standard deviations from above mean, subtract it from the mean, and then divide by the standard deviation, you’ll get z = 1.5. And so on. You can do the same with negative scores. If you have a score that is one standard deviation below the mean, and subtract that from the mean, you’ll get -σ. Divide σ from that, and you’ll get -1.

The standardized normal distribution is associated with standard probabilities, so you don’t have to use the equation, you can just look them up! That’s what makes it so great. For example, we already know that the mean is associated with a probability of .5 (that is, there’s a .5 chance that you’ll get a score less than the mean). But now we can look up the probability associated with 1 standard deviation (z = 1). If you want to play around with these, you can check out the calculator [here](http://psych.colorado.edu/~mcclella/java/normal/normz.html). For our purposes, though, it will be good to memorize a few z-score probabilities. Here they are:

* The probability of getting a score between z = -1 and z = 1 is about **0.68**. That is, there’s a 68% chance that if you take a random instance of X from the population, its value will be between -1 and 1 standard distribution from the mean. If you want cumulative probabilities, the chances of getting a score up to z = -1 is .16, the probability of getting a score up to z = 1 is .84. The chances of getting a score outside of the -1 to 1 range is 1 – 0.68, or 0.32.
* The probability of getting a score between z = -2 and z = 2 is about **0.954**, or there’s a 95.4% chance of getting a score between -2 and 2 standard deviations from the mean.
* The range associated with a 95% probability (which will be important later on) is z = -1.96 to z = 1.96.