Methods and applications of DIMENSIONALITY REDUCTION

Cédric Uden

cedric@uden.de

Computer Science and Business Information Systems
Hochschule Karlsruhe, University of Applied Sciences

December 3rd, 2021

Hello World

Table of Contents

1 Introduction

2 Theory

With data aggregation rapidly growing in volume over the last few decades, it becomes increasingly more difficult to process gathered data into insightful information. We are now exploring first cognisances in the field of dimensionality reduction and are then going to summarise underlying theories to be able to better grasp and tackle this problem.

2.1 The curse of dimensionality

The concept of the *curse of dimensionality* was first introduced by Richard Bellman in 1957. [1] His book, *Dynamic Programming*, explains a method he had developed to explore more efficient solutions to counteract the increasing complexity in problems facing our day-to-day lives. The range of domains where applicable is vast and even covers problems that were impossible to foresee for Richard Bellman. Most notable to us, this includes data science in the 21st century and falls right into the realm of dimensionality reduction.

Bellman observed that, when considering a larger set of variables, even simple and well-understood problems such as determining the maximum of a given function becomes worrisome. They face a variety of difficulties of both gross and subtle nature.

The gross issues primarily consist of a finite amount of computational resources available, especially back in the 1950's. Having access to the computer systems we have today, they could certainly be considered a computational nirvana for any scientists and mathematicians 65 years ago. While Bellman phantisised about such possibilities, he proactively considered them and accurately thought of more subtle problems that could potentially arise. His observations were spot on and are an important factor in why his theories are still relevant as ever nowadays.

The problem is not to be considered solved in the mathematical sense until the structure of the optimal policy is understood. [1] This quote from Richard Bellman eloquently summarises the problem at hand. While it appears plausible, that more quantitative measurements would yield more accurate predictions, our extremely complicated world often misleads us. This results in ourselves neither being able to rigorously understand the problem, nor to improve our predictions of challenges we are unable to analyse. We need to walk a narrow path between the *Pitfalls of Oversimplification and the Morass of Overcomplication.* [1]

With the premise being described by Bellman in 1957, we can conclude that a large amount of features does not only heavily impede model training, but can additionally result in worse solutions. [2] Both from a performance and traceability point of view.

2.2 Necessity

The necessity of reducing dimensions in a large data set serves different purposes in the problems we face in data science. Now follows a brief summary of the different applications where we can utilise the methods of dimensionality reduction. Finally, we will conclude how despite their disparities, the various application methods come together in our ultimate goal of finding tractable and resource-conscious solutions.

Computational impact The most intuitive benefit is certainly its impact on computational performance. Having fewer features reduces our required storage space and allows our learning algorithms to run much faster. [3]

Feature engineering Our models will only be capable of producing relevant results if the features we supply it with are also relevant. [4] *Feature extraction* combines many semantically related features into few, which are found through dimensionality reduction. This helps us to reduce the number of features while retaining as much information as possible from the originating data set. This is not to be confused with *feature selection*, which does not benefit from dimensionality reduction and is therefore not considered in this work.

Data visualisation Reducing dimensions can also help us to visually represent data in an intuitive fashion. Even picturing a relatively basic 4-dimensional hypercube is incredibly difficult to make sense of. Actively recalling this information is not only important for our own understanding of data. It is even more important when sharing our observations and ideas with other data scientists and essential when communicating our conclusions to people in an interdisciplinary environment. [5]

Traceability Having discussed the importance of understanding the problem and its solution earlier in section 2.1, it is often important to be able to comprehend the prediction a model makes instead of trusting a black box model. Reducing the dimensions and understanding the given data helps us to alleviate this issue.

2.2.1 Conclusion

Considering the computational impact helps us to narrow down a complex task into a feasible scope to find resource-conscious solutions. Feature extraction helps us to remove dimensions coupled with an increase in predictive performance right off the bat. This helps us to visualise the data and to be able to understand and trace its predictions. Combined, this helps us to build a tractable solution to our problem.

2.3 Mathematical background

To round up the theoretical premises required for this topic, we will become aware of the boundaries of intuitive mathematical concepts which result in highly counter-intuitive behaviours in high dimensional space, as well as their practical solution approaches.

2.3.1 Euclidean distance & sparse matrices

An important aspect which is frequently utilised in various machine learning methods is to evaluate the euclidean distance between two points in a high-dimensional space. While the concept is simple to understand and illustrate in two or three dimensions, its behaviour in a high-dimensional space changes dramatically and it becomes heavily counter-intuitive to get a hold off.

To discover its behavioural implications, we are considering an n-dimensional space and observing the impact of increasing dimensions between seemingly contrastable points. For this ....

finish this section & consider the sparse matrices section

2.3.2 Eigenvalues and eigenvectors

Quickly recapitulate eigenvectors and explain why they are relevant

3 Techniques

Being acquainted with the premise, we now possess a coarse understanding of the relevant mathematical scaffold required to understand dimensionality reduction. Additionally, we became aware of the potential pitfalls that hold true to the subject in general. Therefore, we are now ready to take a closer look on how to categorise and associate various techniques.

3.1 Linear vs. non-linear problems

The utmost category, used to differentiate between various techniques is fortunately mutually exclusive to any method. Thanks to this characteristic, we can distinctively classify and associate a given technique. This allocation, contrary to the following classifications, is the one that considers the domain of the problem. Thus, it requires special attention by its users since it can easily be used erroneously. We need to understand the gross patterns in the data set we need to operate on.

The below figure 1 illustrates and contrasts the general patterns that we need to identify.

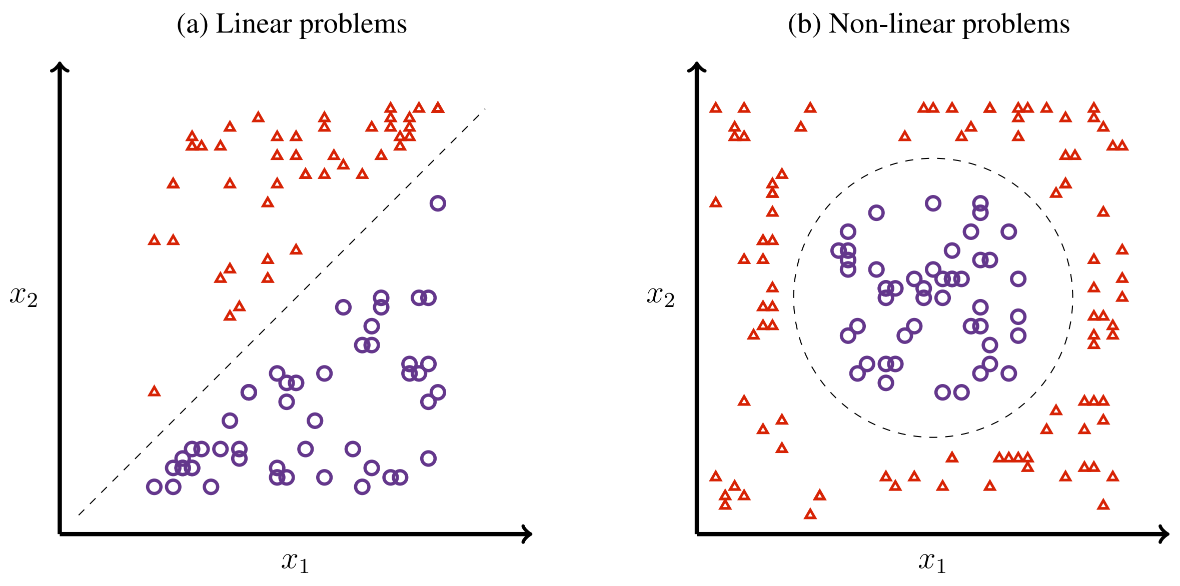

Figure 1: Linear vs. non-linear problems.

As we can see in figure 1a, a linear problem is given when we are able to identify a straight line, or any hyperplane, to split the data into unambiguous subsets. Intuitively and, as we will later examine, factually, these types of problem are comparably easy to solve.

A non-linear problem, as pictured in figure 1b, already looks more complex on the first glance. And indeed, we will illuminate the solution approaches to this scenario and elaborate on its comportment.

3.2 Projection vs. manifold learning

In this section, we will compare the general solution approaches available which can be utilised to solve both linear as well as non-linear problems. [2]

Projection In contrast, this is the trivial concept of the two. The idea is to project the data points onto a hyperplane which summarises the data with as little information loss as possible.

Figure 2 illustrates this in a simple example. As we can observe, when we pick the right hyperplane, such as the x axis in the example, we lose far fewer information than if we would have picked the y axis.

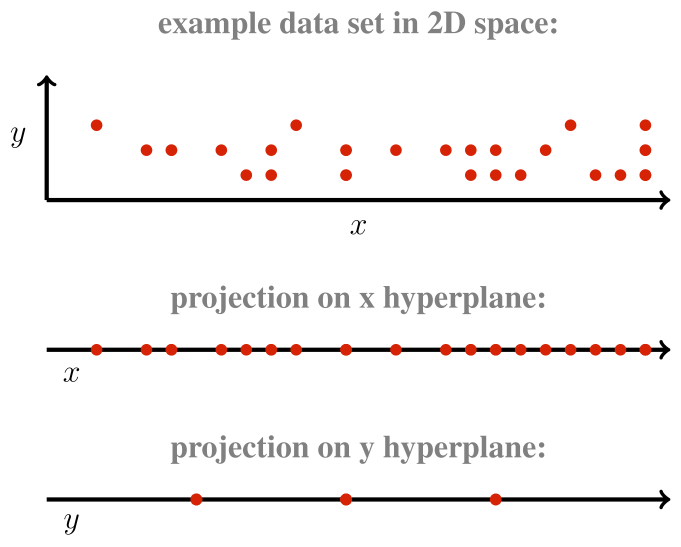
example data set in 2D space:
y
x
projection on x hyperplane:
x
projection on y hyperplane:
y

Figure 2: Simple example of a projection

Manifold learning This concept is significantly more difficult to get a hold of. Significant breakthroughs [6] in this field were accomplished in the year 2000 in the significant and commonly cited paper *A global geometric framework for nonlinear dimensionality reduction*. [7] To understand the basic idea, we will demonstrate its behaviour and briefly dive into the mathematical details using the popular swiss roll data set pictured in figure 3.

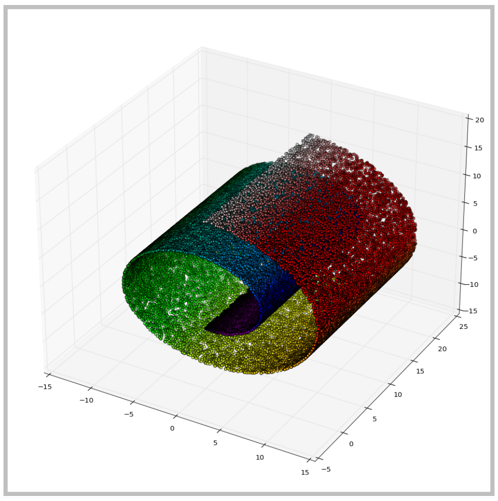

Figure 3: Swiss Roll generated from scikit-learn [8]

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Finish this section

placeholder for swiss roll demo

3.3 Linear techniques

3.3.1 Principal Component Analysis

3.3.2 Singular Value Decomposition

3.3.3 Linear Discriminant Analysis

3.4 Non-linear techniques

3.4.1 Locally Linear Embedding

3.4.2 Isomap Embedding

4 Experiments

5 Evaluation

6 Conclusion

A Bibliography

[1] Richard Bellman. *Dynamic Programming*. Princeton University Press, New Jersey, 1957. ISBN: 0486428095.

[2] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 2nd ed. O'Reilly, 2017. Chap. 8. ISBN: 1492032646.

[3] Sebastian Raschka and Vahid Mirjalili. *Python Machine Learning*. 3rd ed. Packt Publishing, 2019. Chap. 1. ISBN: 9781789955750.

[4] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 2nd ed. O'Reilly, 2017. Chap. 1. ISBN: 1492032646.

[5] Sebastian Raschka and Vahid Mirjalili. *Python Machine Learning*. 3rd ed. Packt Publishing, 2019. Chap. 8. ISBN: 9781789955750.

[6] Yunqian Ma and Yun Fu. *Manifold learning theory and applications*. Vol. 434. CRC press Boca Raton, FL, 2012.

[7] Joshua B Tenenbaum, Vin De Silva, and John C Langford. "A global geometric framework for nonlinear dimensionality reduction". In: *science* 290.5500 (2000), pp. 2319–2323.

[8] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

B Glossary

hyperplane A hyperplane is a subspace whose dimension is one less of the space it is currently represented in. e.g. a one-dimensional line in a two-dimensional graph. 5, 6

C Acronyms

D List of Figures