

Dimensionality reduction

Matouš Dzivjak

17. 11. 2019

Introduction

The goal of this tutorial is to get familiar with some basic methods for dimensionality reduction, complete your own implementation of the **Isomap algorithm** (in cooperation with *Multidimensional scaling*), experiment with its parameters and compare with other techniques of dimensionality reduction (**PCA**, **t-SNE**).

Background

The data you will be working with are vector representations of words in a latent (unknown) high-dimensional space. This representation of words, also known as word embedding, differs from standard bag-of-words (BoW, TFIDF, etc.) representations in that the meaning of the words is distributed across all the dimensions. Generally speaking, the word embedding algorithms seek to learn a mapping projecting the original BoW representation (simple word index into a given vocabulary) into a lower-dimensional (but still too high for our cause) continuous vector-space, based on their distributional properties observed in some raw text corpus. This distributional semantics approach to word representations is based on the basic idea that linguistic items with similar distributions typically have similar meanings, i.e. words that often appear in a similar context (words that surround them) tend to have similar (vector) representations.

Specifically, the data you are presented with are vector representations coming from the most popular algorithm for word embedding known as word2vec¹ by Tomas Mikolov (VUT-Brno alumni). *Word2vec* is a (shallow) neural model learning the projection of BoW word representations into a latent space by the means of gradient descent. Your task is to further reduce the dimensionality of the word representations to get a visual insight into what has been learned.

Data

You are given 300-dimensional word2vec vector embeddings in the file *data.csv* with corresponding word labels in *labels.txt* for each line. Each of these words comes from one of 10 selected classes of synonyms, which can be recognized (and depicted) w.r.t. labels denoted in the file *colors.csv*

Tasks

1. **Load the dataset of 165 words**, each represented as a 300-dimensional vector. Each word is assigned to one of 10 clusters.

```
#Load the dataset of 165 words
mydata <- read.csv('data.csv', header = FALSE)
mylabels <- read.csv('labels.txt', header = FALSE)
mycolors <- read.csv('colors.csv', header = FALSE)
```

The data is in the matrix `mydata`, cluster assignment in `mycolors` and the actual words (useful for visualization) in `mylabels`. You can plot the data by using only the first 2 dimensions.

```
PlotPoints(mydata[,c(1,2)], mylabels, mycolors)
```

```
## deldir 0.1-23
```

¹Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111-3119, 2013.


```
#PLOT results  
print(paste("ISOMAP ACC:", mean(accISOMAP)))  
print(paste("PCA ACC:", mean(accPCA)))  
print(paste("t-SNE ACC:", mean(accTSNE)))
```

Conclusion

The ISOMAP method achieved best accuracy (if correctly implemented) which can be seen from the above plots as well. From the plots can be also seen that the points are more separated compared to other methods where the points tend to be more centered and closer together.