Visualize the dataset of malware using t-SNE

Project Part 1

Abdulrahman Alshammari

Monday 10th April, 2017

1 TSNE

T-Distributed Stochastic Neighbor Embedding (t-SNE) is a technique that is used for reducing the dimensions of a particular data that has too many dimensions and visualize the data in a convenience way. In this project, we applied clustering algorithms on the data and generate many sub-clusters. In each sub-cluster, we would like to highlight where the instances located when we visualize it. To do that, we apply T-SNE on each sub-cluster and we try visualize only the data belongs to the selected one as one color, and the rest of other sub-clusters as a different color. We will do the same process among all other sub-clusters. To run that, the T-SNE script is written in python that takes two parameters, the whole clustered data and the labels (sub-cluster). The expected result will be two dimensional plot with many points where represent instances from clustered data. If the instance belongs to the selected sub-clustered, it will have a certain color, otherwise it will have the second color. Note that the implementation is available in many different languages, but we pick python for simplicity. For explanation, figure (1) is an example of one of sub-clusters TSNE. If a sample belongs to the selected sub-cluster, it will be represented as a green point. Otherwise, the sample will be represented as a dark red point as shown in.

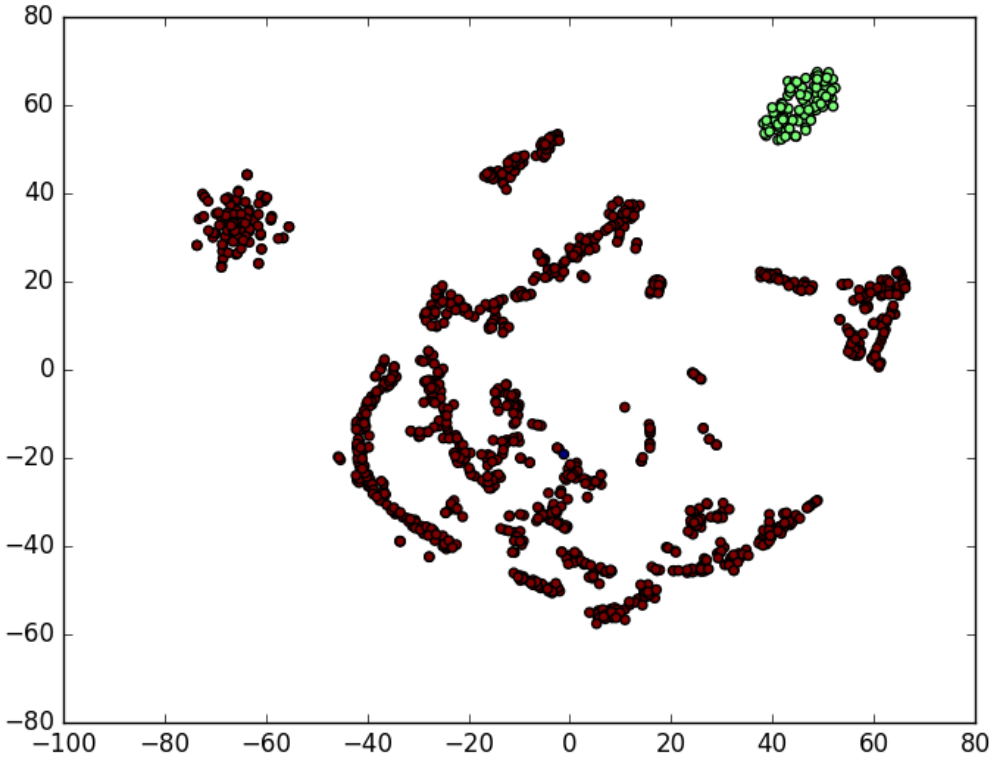

Figure 1.1: A result example of TSNE

2 Related work

For high dimensions reduction, there are some tools beside T-SNE used for visualization like PCA which stands for Principal Components Analysis. This technique is used to emphasize variation and make data easy to visualize. It obviously reduces the number of dimensions of a complex data set. PCA works better with linear data. Compared with T-SNE, PCA is a free parameter where T-SNE relies on many different parameters such as learning rate and number of iterations.

3 Limitations and Challenges

These few points summarize the difficulties I faced in this part:

- I start learning Python when I realized that it is needed to write T-SNE script in Python. The workflow of this part would be faster if we depend on a language such as Java.
- The result of each time we run the T-SNE script has different shapes despite the fact that have the same patterns of design. This is not related to the script itself, but this is one of TSNE features.
- To ensure that we get the right result design, I decided to run the data on TSNE java versions and compare the two results. This expand time of getting the final result.
- The script, which have been edited, with great assistance from Tristan, consider the first row as the name of the features while the data I got from my team doesnt include the features names. After I realized that, I have discussed with the team to match the output.

4 Future plan

In term of T-SNE, I plan to work hard to see if I can generate the same result, for example the same shape, and try to use different colors that we can easily indicate instances belong to a selected subclass. I plan to write a script that get two names; the dataset and data labels and generate the TSNE. Beside that, I will work with my team to generate other components of the Class Signature such as confusion Matrix.

References

[1] G. H. Laurens van der Maaten. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9, Nov 2008.

[2] L. van der Maaten. t-SNE Laurens van der Maaten. `https://lvdmaaten.github.io/tsne/`, 2017.