Mr. Shakespeare, Meet Mr. Tucker, Part II

Charles Nicholas

CSEE Department, UMBC

May 4, 2020

Objectives

To investigate the use of tensor decomposition in static malware analysis - on a large scale

- Malware analysis is often done "in the small", that is, on one specimen at a time [1]
- We need to do malware analysis "in the large"
- Can we use tensor decomposition to gain insight into large collections of malware?

Building the Tensor

We selected a specific malware family, the well-known Zeus Trojans [2], as test subjects.

The tensor X is constructed so that: for each Zeus file i, entry $x_{i,j,k}$ is how many times 4-gram j occurs in decile k of the file. That is,

- $1 <= i <= 8020$, the number of Zeus specimens available to us
- $1 <= j <= 2^{32}$, the upper bound on the number of distinct 4-grams. The actual number of distinct 4-grams of course varies from file to file.
- $1 <= k <= 10$, since we chose to represent the approximate location in each specimen by dividing each specimen into ten parts of equal length.

Tucker Decomposition from Kolda [3]

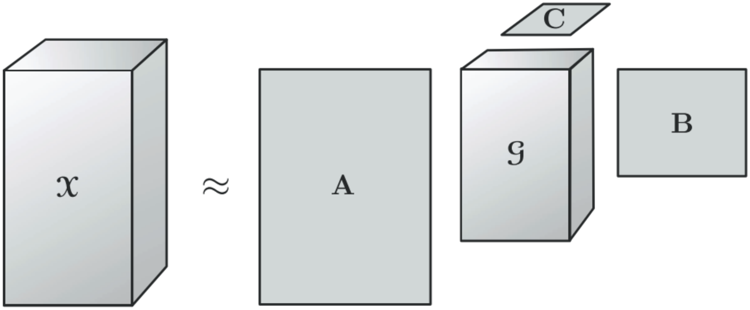
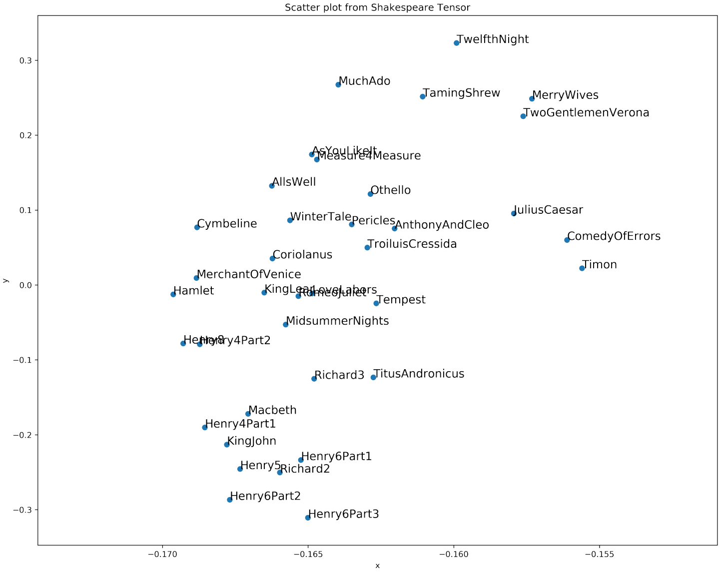


Fig. 4.1 *Tucker decomposition of a three-way array.*

Sanity Check - Shakespeare

Before trying the Zeus data, we wanted to try a smaller corpus - the Shakespearean plays. [4]
Using Python packages `sklearn` (to parse the text data) and `tensorD`[5] and `tensorflow` (to do the tensor calculations), in a Jupyter Notebook, we built the tensor X as described earlier, and ran both HOSVD and HOOI versions of Tucker.

Plot of First Two Factors from Tucker Decomposition



Tensor and Results

- In the Shakespearean tensor X, entry $x_{i,j,k}$ is the number of times word j occurs in Act k of play i. The value of i ranges from 1 to 37, j ranges from 1 to about 30,000, and k ranges from 1 to 5. The tensor is quite sparse.
- Plotting the first two factors produced by HOOI, HOSVD gave similar results
- We are pleased with the (unsupervised!) clustering of the history plays at the bottom of the plot.

Research Continues: Update

- Malware binaries will have *many* more terms than Shakespeare does, so we must be selective.
- Only some of the Zeus binaries are unpacked, so focus on those first.

Acknowledgements

- This work was supported by the Laboratory for Physical Sciences
- An earlier version of this talk was presented as a poster at the High Performance Computing and Data Analytics Workshop, September 10-11, 2019.

- Email: nicholas@umbc.edu

References

- M. Sikorski and A. Honig, *Practical Malware Analysis*. no starch press, 2012.
- A. Mohaisen, O. Alrawi, and O. Mohaisen Abedelaziz Alrawi, "Unveiling Zeus: automated classification of malware samples," in *Proceedings of the 22nd international conference on World Wide Web companion*, 2013, pp. 829–832, ISBN: 978-1-4503-2038-2.
- T. G. Kolda and B. W. Bader, "Tensor Decompositions and Applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009. DOI: 10.1137/07070111X.
- P. T. Kesha, "Detection of Malware using Tensor Decomposition," UMBC M.S. Writing Project, Tech. Rep., 2019.
- L. Hao, S. Liang, J. Ye, and Z. Xu, "TensorD: A tensor decomposition library in TensorFlow," *Neurocomputing*, vol. 318, pp. 196–200, Nov. 2018. DOI: 10.1016/j.neucom.2018.08.055.