CS 6962 Decomposition Techniques for Computational Data-Enabled Science and Engineering
Spring Semester 2017

Instructor: Chris Johnson, Ph.D.

Time: T,Th 12:25 - 1:45 p.m.

Place: WEB L105

Office: 3850 WEB

Phone: 801-581-7705

Email: crj@sci.utah.edu

Class web page: www.eng.utah.edu/~cs6962

Co-Instructor: Tushar Athawale, Ph.D.

Email: tushar.athawale@sci.utah.edu

Course Overview:

“We are drowning in data and starving for knowledge.” - John Naisbitt

Researchers in a variety of fields collect measurements, observe data, perform simulations and use a wide range of techniques to describe, classify, analyze and draw conclusions from these data. Selecting appropriate techniques and understanding their advantages and disadvantages is an important component of data analysis. In particular, in this age of big data, large data sets provide distinct challenges given that many of the general techniques for small data sets do not scale to larger problems and are often prohibitively expensive from a computational perspective.

In this class, we will survey several data decomposition techniques For computational data-enabled science and engineering applications including:

- Principle Component Analysis
- Independent Component Analysis
- Singular Value Decomposition
- Generalized Singular Value Decomposition
- Signal Fraction Analysis
- Non-Negative Matrix Factorization
- Dynamic Mode Decomposition
- Singular Spectrum Decomposition

- Karhunen-Loeve Decomposition
- Sparse Multivariate Methods

and evaluate their strengths and weaknesses for a variety of applications.

Prerequisite: Linear Algebra

Course Goals: Upon completion of this course, the student should

- Know a wide variety of algorithms and techniques for data decomposition,
- Have seen and discussed examples of the use of data decomposition techniques applied to problems in a variety of fields,
- Know where to locate data decomposition software resources and references,
- Have completed a data decomposition project.

Assignments: There are two main types of assingments for this course. One is in the form of in between class homework that will primarily consist of using open source software to analyze data sets. The second will be in the form of a data decomposition project. This project can involve data you have collected from a simulation and/or experiment or development of a new software tool(s). My goal is to supply the student with as close to *real life* data decomposition research applications as possible within the confines of a semester long class.

Languages: For this course we will primarily use Matlab along with additional open source software tools.

Grades: Final course grades will be computed according to 80% Homework and Labs and 20% Final Project.

Incompletes: As the project is due by the end of the semester, in past similar project-based courses, it has turned out that some people do not wisely schedule their time and do not finish their projects. They then want to take an incomplete and finish the project sometime in the summer. I only give incompletes very rarely and only for truely unusual circumstances (death in the family, etc.), so **please** work to finish your final project on time.

Syllabus for CS 6962

Week 1. Class mechanics, motivation and overview of application problems

Week 2. Linear algebra background

Week 3. Singular value decomposition and principal component analysis

Week 4. Independent component analysis and generalized SVD

Week 5. Guest lectures

Week 6. Signal fraction analysis

Week 7. Nonnegative matrix factorization

Week 8. Guest lectures

Week 9. Case studies

Week 10. Spring break

Week 11. Tensor decompositions

Week 12. Guest lectures

Week 13. Project presentations

Week 14. Project presentations

Week 15. Final project due

Note: During the semester, we will have several guest lectures on selected topics in decomposition methods.

Decomposition Techniques for Computational Data-Enabled Science and Engineering Project

Due dates: Project description **due March 7**. Project presentations will be on April 20 and April 25. The final project write up is **due May 1**.

The decomposition project can be (1) from simulation and/or experimental data you have or (2) development of new software tools or extending existing software tools.

It is your responsibility to pitch your project at the appropriate level. Challenge, but do not exhaust, yourself. Please ensure that even if you underestimate the difficulty of your project, you will have something to hand in by the due date (choosing too difficult a project is not a valid reason for an incomplete).

Group projects are allowed, however, the size and difficulty of the project should reflect the number of people involved in a single project.

On **March 7** your project design report is due. This should be a well thought out, well-written one page description of your proposed project. It should outline any necessary background, specifically what goals you plan on accomplishing, and what you will need to do in order to accomplish your goals. You will also need to include what software/hardware you plan to use, and what you intend to hand in (i.e. what are the "deliverables").

In grading the projects, I will be looking for a well-designed, substantial, interesting project. Furthermore, your implementation, content and style of the final results should be of high quality. A final criteria for grading is that the progress report and final report are handed in on time.

You will present your final projects on Tuesday, **April 20** or Thursday, **April 25**. A final project sign-up sheet will be handed out in class for you to schedule a time. Presentations should typically take approximately 15 minutes.

Project Design Report

Please hand in your Project Design Report by **March 7** (or sooner). It should contain the following information.

Student Name(s):

Project title:

- Give an overview of the project.
- Why is this project important and/or interesting?
- If you are doing a programming project, list the hardware and software you will be using:
- What have you done thus far and what will you have to do to complete this project?
- When the project is completed, how *specifically* can we evaluate how successful it is?
- Any other useful information:

Project Final Report

You will be required to hand in your Project Final Report on **May 1.** Your final report should contain the following information.

Student Name(s):

Project title:

- Provide a brief description of your project and how to run it if it is not self-explanatory.
- Outline what you learned from doing this project.
- If you have not accomplished all the goals of your project, or if you have exceeded them, describe how the finished project differs form the description in your project design.
- Evaluate your project: how successful do you think it was? What are the strengths and weaknesses of your project?
- Any other comments useful for me in evaluating your project:

References

1. R. Baraniuk, Compressive sensing, IEEE Signal Processing Magazine, pp. 118-124, July 2007.
2. E. Cands, L1-magic toolbox , http://www-stat.stanford.edu/~candes/l1magic.
3. A. Cichocki, N. Lee, I.V. Oseledets, A-H. Phan, Q. Zhao, D. Mandic. Low-Rank Tensor Networks for Dimensionality Reduction and Large-Scale Optimization Problems: Perspectives and Challenges PART 1. arXiv:1609.00893, https://arxiv.org/abs/1609.00893, 2016.
4. Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan, Shun-ichi Amari. Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation, Wiley, 2009.
5. J. Demmel. Applied Numerical Linear Algebra, SIAM Press, 1997.
6. D. Donoho, Compressed sensing, IEEE Transactions on Information Theory 52 , 1289-1306 (2006).
7. L. Elden. Matrix Methods in Data Mining and Pattern Recognition, SIAM Press, 2007.
8. F. Emdad. High Dimensional Data Analysis: Overview, Analysis, and Applications, VDM Verlag, 2008.
9. G. Golub and C. Van Loan. Matrix Computations. Johns Hopkins University Press, 2012.
10. by Trevor Hastie, Robert Tibshirani, Martin Wainwright. Statistical Learning with Sparsity: The Lasso and Generalizations, CRC Press, 2015.
11. A. Hyvrinen and E. Oja, Independent component analysis: Algorithms and applications, http://mlsp.cs.cmu.edu/courses/fall2012/lectures/ICA_Hyvarinen.pdf.
12. M. Kirby and L. Sirovich, Application of the KarhunenLoeve procedure for the characterization of human faces, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 , 103–108 (1990).
13. J. Nathan Kutz, Data-Driven Modeling and Scientific Computation: Methods for Complex Systems and Big Data", Oxford University Press, 2013.
14. by J. Nathan Kutz, Steven L. Brunton, Bingni W. Brunton, Joshua L. Proctor. Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems, SIAM Press, 2016.
15. M. Turk and A. Pentland, Eigenfaces for recognition, Journal of Cognitive Neuroscience 3 , 71–86 (1991).

16. J. Shlens, A tutorial on principal component analysis,
www.sci.utah.edu/~chris/PCA-Tutorial-Shlens-2014.pdf.

17. D. Skillicorn. Understanding Complex Datasets: Data Mining with Matrix Decompositions, Chapman and Hall/CRC, 2007.

18. L. N. Trefethen and D. Bau, Numerical Linear Algebra (SIAM, 1997).

19. X. Yuan and J. Yang, Sparse and low-rank matrix decomposition via alternating direction methods, www.optimization-online.org, (2009).