FARIBORZ MASEEH
DEPARTMENT OF
MATHEMATICS + STATISTICS

MTH 371 Winter 2021: Large-Scale Data Algorithms

Instructor: *Panayot Vassilevski, FMH Q460, panayot@pdx.edu, http://web.pdx.edu/~panayot/*
Office Hours: TBD via zoom.

Textbook: not available yet; will use material based on instructor's lectures; three books suggested for reference:

Mark Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong, *Mathematics for Machine Learning*, Cambridge University Press, 2020.

M.E.J. Newman, *Networks. An Introduction*, Oxford University Press, 2010.

G. Strang, *Linear Algebra and Learning from Data*, Wellesley – Cambridge Press, 2019 (distributed by SIAM)

Synopsis

The course emphasizes the need for more specialized linear algebra methods when dealing with large scale data, numerical or by relations, namely ones represented by sparse matrices and/or by graphs. It covers graph Laplacian, modularity matrix and related functional; their use for clustering/community detection and partitioning. These tools are employed to build two and multilevel algorithms for solving large systems of equations and possibly for training neural networks. Programming homework assignments and a final project using publicly available data repositories will complement the studied methods.

Prerequisites: MTH 343, or by instructor consent.

Required or be able to learn during class skills: knowledge of programming language (preferably python, or C/C++)

Disclaimer This course is for *advanced* undergraduate students that are *very comfortable* with linear algebra (matrix-vector) notation and knowledge of fundamental linear algebra facts. The students are expected to be familiar with the Σ notation and the use of subscripts (indices). A working knowledge of programming language (python (preferably) or C/C++) is required. A brief review of the linear algebra facts studied in MTH 343 (and before that in MTH 261) will be provided (in particular all matrix decompositions for square and rectangular matrices). The emphasis will be on the cost for memory and operations which impose serious requirements for their use on large data (stored as matrices).

Grading: Based on 5 homework assignments and a final project.

Student Learning Outcomes: Upon completion of this course students will have the ability to:

- Formulate a problem in terms of graphs, possibly with edge weights.
- Apply several clustering techniques based on graph modularity, spectral methods, and multilevel weighted matching methods and compare the obtained results in terms of accuracy and timings.
- Set up and solve large scale problems by running programs implementing gradient type methods in a hierarchical (two- and multilevel) fashion for better performance.
- Set up commonly used (deep) neural networks and understand the cost associated with their training in terms of both time and accuracy.

Using existing publicly available libraries will be encouraged (and helped with by the instructor) for better and faster implementation of the studied algorithms.

Access and Inclusion for Students with Disabilities: PSU values diversity and inclusion; we are committed to fostering mutual respect and full participation for all students. My goal is to create a learning environment that is equitable, useable, inclusive, and welcoming. If any aspects of instruction or course design result in barriers to your inclusion or learning, please notify me. The Disability Resource Center (DRC) provides reasonable accommodations for students who encounter barriers in the learning environment.

If you have, or think you may have, a disability that may affect your work in this class and feel you need accommodations, contact the Disability Resource Center to schedule an appointment and initiate a conversation about reasonable accommodations. The DRC is located in 116 Smith Memorial Student Union, 503-725-4150, drc@pdx.edu, https://www.pdx.edu/drc.

- If you already have accommodations, please contact me to make sure that I have received a faculty notification letter and discuss your accommodations.

- Students who need accommodations for tests and quizzes are expected to schedule their tests to overlap with the time the class is taking the test.

- Please be aware that the accessible tables or chairs in the room should remain available for students who find that standard classroom seating is not useable.

- For information about emergency preparedness, please go to the Fire and Life Safety webpage (https://www.pdx.edu/environmental-health-safety/fire-and-life-safety) for information.

Title IX Reporting:
As an instructor, one of my responsibilities is to help create a safe learning environment for my students and for the campus as a whole. As a member of the university community, I have the responsibility to report any instances of sexual harassment, sexual violence and/or other forms of prohibited discrimination. If you would rather share information about sexual harassment, sexual violence or

discrimination to a confidential employee who does not have this reporting responsibility, you can find a list of those individuals.

Academic Integrity

Academic integrity is a vital part of the educational experience at PSU. Please see the PSU Student Code of Conduct for the university's policy on academic dishonesty. A confirmed violation of that Code in this course will result in failure of the course.

Topics and tentative schedule of lectures

Jan. 3-12: (Review)

(a) Fundamentals: Matrix-matrix products; multiplication by blocks. Entry-free matrices (relations). Composition of relations. CSR matrix format.
(b) Linear dependence, basis, orthogonality and standard vector norm. Column space and nullspace of matrices. Matrix inverses.
(c) Systems of equations: elimination via LU factorization.
(d) QR factorization and least-squares (modified Gram Schmidt)
(e) Eigenvalues – using orthogonal transformation to reduce the matrix to tridiagonal, Hessenberg, bidiagonal forms. Spectral (Q ΛQ^T) and singular value (U Σ V^T) decompositions.
(f) Jan. 12 "**pass/no pass"** midterm/quiz on fundamentals of LA (review of facts (theory) with true/false answers).

(i) Jan. 17 (MLK) –Jan. 19: Matrices and vectors and operations between them – cost in terms of memory and operations –dense versus sparse. Matrix Factorization – the Schur complement form – algorithm and estimation of the cost. Algorithms for solving systems of equations with factorized matrix (forward and backward elimination) – estimating cost in terms of memory and operations. **Homework #1** (due Jan. 25) – on implementation of these algorithms.

(ii) Jan. 24-26: Vector and matrix norms; inner products. Symmetric positive definite matrices. Iterative methods for solving systems of equations. Concept of convergence. Jacobi, Gauss-Seidel (forward, backward and symmetric Gauss-Seidel). Convergence of iterative methods in terms of matrix condition number. **Homework # 2** (due Feb 1) – write down the GS algorithm(s) and estimate the cost for one iteration in the case of dense and sparse matrices, respectively.

(iii) Jan. 31-Feb. 2: Sparse matrices and graphs; equivalence between them. Two major sources of (very) large sparse matrices: graphs (relations) from social networks and matrices coming from scientific computing (solving numerically partial differential equations (PDEs)). The second example (numerical solution of PDEs) involves assigning unknown values to mesh entities (e.g., associated with the vertices of a mesh) with the property the finer the mesh the better accuracy expected for the numerical solution. Some popular graph matrices: incidence matrix, graph Laplacian, modularity matrix. Probabilistic interpretation of modularity. **Homework # 3** (due Feb. 8) –given the edge-vertex relation as a sparse matrix, write algorithms for computing graph incidence matrix, the graph Laplacian and the modularity matrix.

(iv) Feb. 7-9: Matching in graphs. Utilizing edge weights in graph matching algorithms – the parallel matching algorithm of Luby. Using recursive matching for graph clustering, i.e., in creating communities/aggregates. Representing communities (clusters, aggregates) with a sparse matrix P. Creating coarse graph from aggregates and its P^TAP matrix representation. **Homework # 4** (due Feb. 15): Given a graph G represented by its weighted adjacency matrix, write a weighted matching algorithm and estimate its cost. For a specific graph, write the respective P matrix representing the matchings (as pairwise aggregates and possible singletons), and then compute the adjacency matrix of the coarse graph corresponding to the aggregates.

(v) Feb. 14-16: Inefficiency of stationary iterative method (Jacobi, Gauss-Seidel) for ill-conditioned matrices. Variational iterative methods- the conjugate gradient (CG) method. Improving conditioning by preconditioners. The preconditioned CG method. The two-level preconditioner as an algorithm. Using graph coarsening to define TL preconditioners. Recursive use of TL methods – multilevel methods. **Homework #5** (due Feb. 22): Estimate the cost of one iteration of the CG and PCG algorithms in terms of the sparsity of the matrix A and the cost of implementing the inverse action of the preconditioner. Estimate the number of iterations required to achieve relative tolerance ε (epsilon) based on the condition number of the preconditioned matrix.

(vi) Feb. 21-23: Graph embedding: assigning coordinates to graph vertices. Commute distance as a distance metric. Using data with coordinates in the K-means algorithms for clustering. Fisher linear discriminant analysis (LDA) – separate two clouds of data points (not always possible with a plane) – the resulting eigenvalue problem and its equivalent linear system. Start discussions with instructor about **final project**.

(vii) Feb. 28 -Mar. 2: Iterative methods and propagation of information over networks; error components – local (oscillatory) and global (smooth) components – geometric interpretation (illustration) and its algebraic counterpart – algebraically smooth (or near null) components of the error. Using near-null components of the error for edge weights in coarsening (via matching). The *algebraic* two- and multilevel method. Composition of two preconditioners. The adaptive algebraic multilevel method.

(viii) Mar 7-9: From linear and nonlinear least squares to deep neural networks (DNNs). The linear algebra of DNNs. An idea for the approximation ability of DNNs as *universal* approximation tool. Training DNNs using stochastic gradient descent – the main idea.

(ix) Mar 14: Final project (negotiated with instructor well in advance) due.