CS369M: Algorithms for Modern Massive Data Set Analysis

Instructor: Michael Mahoney (mmahoney@cs.stanford.edu)

Teaching Assistant: Alex Shkolnik (ads2@stanford.edu)

Major research project: An important component of the class will be a major research project. The goal is to drill down in much more detail on some topic related to what was covered in the lectures.

Important dates: Please email a ps or pdf of the following reports to the TA by 5PM on the date specified—*do not be late*.

- **Fri, Oct 9, 2009:** A brief statement stating (1) who you will be working with and (2) the project you plan to address. (One or two sentences as text is fine—we want to make sure people are working on a range of projects and will get back very soon if there is any problem.)
- **Fri, Oct 16, 2009:** Initial proposal, consisting of not more than a one page summary of the proposed project. (This will be mostly to make sure you are on the right track—we will get back to you within a few days if there are any issues, and we can also discuss any concerns you might have.) Note: this may be turned in with the first homework which will be due on 10/19/09 in class.
- **Fri, Nov 13, 2009:** Mid-term report, consisting of approximately three to four pages with a brief summary of relevant literature, summary of proposed directions, and any questions or problems encountered.
- **Fri, Dec 11, 2009:** Final report, consisting of an eight to ten page report prepared in a format appropriate for publication.

Also, note that writing well (say, so that another person like the instructor or TA or a fellow student can read your paper and understand your ideas) is very hard, even after you have figured out the main ideas in the papers you will be summarizing or have the research results you want to present mostly completed. Thus, it would be wise to start the writing early and make a few iterations, perhaps running it by a fellow student.

More details: I suggest working in teams of two, in which case you should make clear who did what as a footnote in the final report. You may work individually or in a group of three, with an associated adjustment in expectations, but you should coordinate with the instructor first. The project will have two equally-important components:

- **Paper-reading component:** The goal will be to synthesize, summarize, and provide a detailed evaluation of some number of papers, most like roughly 2 to 4 papers, on some topic related to what we discussed in class. The final report should include a high-quality critique of these papers, much like but perhaps more detailed than a good review paper. It should place the papers in a broader context, and it should include a discussion of methods/results of the papers, of the strengths and weaknesses of the particular papers, as well as of their relationships with other related work.

- **Research component:** The goal is for you to perform new research, extending in a novel direction the papers you have read and reported upon. Ideally, you will obtain some interesting original theoretical or empirical results related to the topics we discussed in class and in doing so make substantial progress toward a nice conference paper. Since research sometimes does not succeed, success will certainly not be required to obtain a good grade on the final project. The final report should provide a detailed description of your methods and what novel theoretical or empirical results were obtained, your interpretation of the work, including why it succeeded or failed, what it reveals about the problem or the data or the techniques you used, and what its broader implications are.

The final report should be written in the form of a conference paper submission. Thus, it should include an introduction, a description of previous related work, a description of novel theoretical or empirical results that have been obtained, and a conclusion summarizing the results and further directions to follow.

The level of exposition of your report should be for one of your classmates, i.e., someone who has a good understanding of the area and of the lectures but who has not gone into detail on the particular topic you chose to address in detail. Two examples of a good final report (by Nikola Milosavljev in Tim Roughgarden's CS364A class in Fall 2004 and by Dan Gillick, Arlo Faria, John DeNero in Berkeley's CS262, respectively) may be found here:

- `http://theory.stanford.edu/(TILDE)tim/f06/nikola.pdf`
- `http://www.icsi.berkeley.edu/(TILDE)arlo/publications/gillick_cs262a_proj.pdf`

Clearly, your report will differ, depending among other things whether you are doing a more theoretical project or a more applied project, etc. Your report should, however, be at a similar level of depth, detail, and clarity.

Additional resources: In addition to the references on the class web page and those listed below for particular projects, two very good resources for ideas for the project are the web pages for the 2006 and 2008 MMDS meetings:

- MMDS 2006: `http://www.stanford.edu/group/mmds/mmds2006.html`
- MMDS 2008: `http://www.stanford.edu/group/mmds/`

In particular, you might look at the slides from the speakers. Many of the topics in class were discussed at these meetings, and many of the slides have a wealth of information about both the theory and applications.

Potential project topics: Here are a few suggested topics. (Of course, you are free to suggest another.) Note that some of these topics might be more easily-addressable than others, depending on your background and interests. In addition to the references on the class web page, which you should use as a resource to get started finding relevant papers, a few additional pointers for some of the suggested topics are given. These should just get you started—it would also be good to look at other references.

- Implement and apply randomized matrix algorithms:

- Use random projection or sampling primitives to do other computations faster and/or better, e.g., approximate eigenvectors, underconstrained regression, graph sparsification, solve SDPs, etc. (Start with [46, 20, 30].)
- Combine random projection and random sampling. For example, do a random projection to compute leverage scores to identify nonuniformity in low-dimensional space to then sample randomly or greedily. (Start with [23, 10, 19, 37].)

- Implement and apply spectral and flow graph algorithms:
  - Use spectral or flow-based partitioning methods to understand the structure of data sets such as images or social and information networks or for other tasks such as semi-supervised learning or manifold learning. (Start with [44, 2] or with [29, 49, 42], respectively.)
  - Use local spectral methods to identify local structure in large data sets such as as images or social and information networks, or for locally performing other tasks of interest. (Start with [1, 2, 14, 15].)

- Use graph algorithms to probe the structure of large networks:
  - Find locally linear structure in social and information networks. Use local spectral methods to identify local patches in large graphs that are meaningfully low dimensional, and compare this with more combinatorial methods. (Start with [43, 34, 1].)
  - Test the manifold hypothesis. Define statistics that data generated according to a manifold would satisfy and see if real data satisfies it. Relatedly, test whether locally low-dimensional structure in a dataset "propagates" into meaningful global structure. (Start with [43, 34, 1].)
  - Visualization of large social and information networks. Combine existing work on using eigenvectors to visualize low-dimensional graphs and other methods to visualize expander-like hyperbolic graphs to visualize real social and information networks that have both properties at different size scales. (Start with [40, 32, 34].)

- Kernels, data-motivated matrix factorization methods, and other data applications:
  - Explore the Nyström method, which has been widely used to speed up kernel-based learning machines, provide low-rank approximations, provide coordinates for new data points, etc. (Start with [48, 6, 33].)
  - Explore diffusion-based kernels and their relevance for data in light of spectral and flow-based graph partitioning methods. (Start with [31, 7, 18].)
  - Implement and apply data-motivated matrix factorizations (e.g., sparse PCA, maximum margin methods, matrix rank minimization, Bregmann divergences, CUR and related decompositions, Nyström-based methods, etc.) with and without ensemble-based methods. (Start with [47, 22], which do it for one such factorization.)
  - Random graph models have been widely-studied and are of interest in a wide range of applications and give rise to numerous theoretical and applied questions. (Start with [16, 41, 11, 9].) Think of something new—one possibility is to explore the connections between recent work like [34, 8, 39] and how this work begins to address some of the failings of popular models.

- Heavy-tailed data, e.g., social or information graphs with heavy-tailed degree sequences, are ubiquitous in applications. Sometimes people think of them as low dimensional, e.g., when they truncate the SVD or PCA, whereas sometimes people think of them as high dimensional. Which are they? Explore the extent to which low-dimensional versus high-dimensional tools we have discussed are relevant for this class of data; the most interesting observations will come from considering networked data. (Start with [38, 28, 17, 34].)
- Violations of Wigner's semicircle law in sparse and heavy-tailed data. Explore why the law is violated in these classes of data sets, including similarities and differences between these two classes of graphs and the relationship of them with expander graphs. (Start with [24, 25].)
- Multiplicative weights update method, boosting, games, and online learning. Describe connections between the methods and evaluate how they perform with respect to worst-case analysis and in typical applications. (Start with [35, 26, 3].)
- DNA SNP data provide a fertile ground for the methods we have discussed, although state-of-the art methods sometimes use heavier-duty and more expensive statistical modeling techniques. Apply spectral methods or matrix decomposition methods, with or without kernels, to structure extraction from HapMap data. (Talk with the instructor if interested.)
- There are a lot of other extensions of what we talked about to data analysis and kernel-based learning, and the most interesting of these often come from a particular application-dependent problem. Suggest and explore one.

- Implementation in very large-scale systems:
  - Describe large-scale computational issues by doing a project on how to implement the ideas we have discussed in a large-scale system, e.g., in a MapReduce-like system. (Start with [21, 5, 13, 45, 12].)
  - Describe large-scale computational issues by doing a project on how to implement the ideas we have discussed in a large-scale system, e.g., in a high performance computing system. (Start with with [4, 27, 36].)

- Approximation algorithms and implicit regularization:
  - Approximate eigenvector computation and statistical regularization. Show that doing a randomized or approximate computation of an eigenvector leads to nicer or smoother or more compact eigenvectors. (Talk with the instructor if interested.)
  - Approximate graph computation and implicit regularization. Do the same with approximation algorithms with graphs—for example, show a flow-based graph partitioning algorithm lead to smoother or more local cuts in expanders. (Talk with the instructor if interested.)

- Any other ideas?

References

[1] R. Andersen, F.R.K. Chung, and K. Lang. Local graph partitioning using PageRank vectors. In *FOCS '06: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 475–486, 2006.

[2] R. Andersen and K. Lang. Communities from seed sets. In *WWW '06: Proceedings of the 15th International Conference on World Wide Web*, pages 223–232, 2006.

[3] S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: a meta algorithm and applications. *Manuscript.*, 2005.

[4] D. A. Bader. Petascale computing for large-scale graph problems. In *Proceedings of the 7th International Conference on Parallel Processing and Applied Mathematics (PPAM 2007)*, pages 166–169, 2008.

[5] J. Becla and D. L. Wang. Lessons learned from managing a petabyte. In *CIDR*, pages 70–83, 2005.

[6] M.-A. Belabbas and P. J. Wolfe. Spectral methods in machine learning and new strategies for very large datasets. *Proc. Natl. Acad. Sci. USA*, 106:369–374, 2009.

[7] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[8] M. Boguñá, D. Krioukov, and K.C. Claffy. Navigability of complex networks. *Nature Physics*, 5:74–80, 2009.

[9] B. Bollobás and O. M. Riordan. Mathematical results on scale-free random graphs. In S. Bornholdt and H.G. Schuster, editors, *Handbook of Graphs and Networks*, pages 1–34. Wiley, 2004.

[10] C. Boutsidis, M.W. Mahoney, and P. Drineas. An improved approximation algorithm for the column subset selection problem. Technical report. Preprint: arXiv:0812.4293 (2008).

[11] D. Chakrabarti and C. Faloutsos. Graph mining: Laws, generators, and algorithms. *ACM Computing Surveys*, 38(1):2, 2006.

[12] E. Y. Chang, K. Zhu, H. Wang, H. Bai, J. Li, Z. Qiu, and H. Cui. PSVM: Parallelizing support vector machines on distributed computers. In *Annual Advances in Neural Information Processing Systems 20: Proceedings of the 2007 Conference*, pages 000–000, 2008.

[13] C.-T. Chu, S. K. Kim, Y.-A. Lin, Y. Y. Yu, G. Bradski, A. Y. Ng, and K. Olukotun. Map-reduce for machine learning on multicore. In *Annual Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference*, pages 000–000, 2007.

[14] F.R.K Chung. Four proofs of Cheeger inequality and graph partition algorithms. In *Proceedings of ICCM*, 2007.

[15] F.R.K. Chung. The heat kernel as the pagerank of a graph. *Proceedings of the National Academy of Sciences of the United States of America*, 104(50):19735–19740, 2007.

[16] F.R.K. Chung and L. Lu. *Complex Graphs and Networks*, volume 107 of *CBMS Regional Conference Series in Mathematics*. American Mathematical Society, 2006.

[17] A. Clauset, C.R. Shalizi, and M.E.J. Newman. Power-law distributions in empirical data. arXiv:0706.1062, June 2007.

[18] R.R. Coifman, S. Lafon, A.B. Lee, M. Maggioni, B. Nadler, F. Warner, and S.W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition in data: Diffusion maps. *Proc. Natl. Acad. Sci. USA*, 102(21):7426–7431, 2005.

[19] S. Dasgupta and A. Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures and Algorithms*, 22(1):60–65, 2003.

[20] A. d'Aspremont. Subsampling algorithms for semidefinite programming. Technical report. Preprint: arXiv:0803.1990 (2008).

[21] J. Dean and S. Ghemawat. MapReduce: Simplified data processing on large clusters. In *Proceedings of the 6th conference on Symposium on Opearting Systems Design and Implementation*, pages 10–10, 2004.

[22] D. DeCoste. Collaborative prediction using ensembles of maximum margin matrix factorizations. In *Proceedings of the 23rd international conference on Machine learning*, pages 249–256, 2006.

[23] P. Drineas, M.W. Mahoney, and S. Muthukrishnan. Relative-error CUR matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, 30:844–881, 2008.

[24] I. J. Farkas, I. Derényi, A.-L. Barabási, and T. Vicsek. Spectra of "real-world" graphs: Beyond the semicircle law. *Physical Review E*, 64:026704, 2001.

[25] U. Feige and E. Ofek. Spectral techniques applied to sparse random graphs. *Random Structures and Algorithms*, 27(2):251–275, 2005.

[26] Y. Freund and R.E. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1-2):79–103, 1999.

[27] J. R. Gilbert, S. Reinhardt, and V. B. Shah. A unified framework for numerical and combinatorial computing. *Computing in Sciences and Engineering*, Mar/Apr 2008.

[28] C. Gkantsidis, M. Mihail, and A. Saberi. Conductance and congestion in power law graphs. In *Proceedings of the 2003 ACM SIGMETRICS International Conference on Measurement and modeling of computer systems*, pages 148–159, 2003.

[29] T. Joachims. Transductive learning via spectral graph partitioning. In *Proceedings of the 20th International Conference on Machine Learning*, pages 290–297, 2003.

[30] N. El Karoui and A. d'Aspremont. Approximating eigenvectors by subsampling. Technical report. Preprint: arXiv:0908.0137 (2009).

[31] R. I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete structures. In *Proceedings of the 19th International Conference on Machine Learning*, pages 315–322, 2002.

[32] Y. Koren. Drawing graphs by eigenvectors: theory and practice. *Computers and Mathematics with Applications*, 49(11-12):1867–1888, 2005.

[33] S. Kumar, M. Mohri, and A. Talwalkar. On sampling-based approximate spectral decomposition. In *Proceedings of the 26th International Conference on Machine Learning*, pages 553–560, 2009.

[34] J. Leskovec, K.J. Lang, A. Dasgupta, and M.W. Mahoney. Statistical properties of community structure in large social and information networks. In *WWW '08: Proceedings of the 17th International Conference on World Wide Web*, pages 695–704, 2008.

[35] N. Littlestone and M.K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

[36] A. Lumsdaine, D. Gregor, J. Berry, and B. Hendrickson. Challenges in parallel graph processing. *Manuscript*, 2009.

[37] J. Matoušek. On variants of the Johnson–Lindenstrauss lemma. *Manuscript.*

[38] M. Mihail and C.H. Papadimitriou. On the eigenvalue power law. In *RANDOM '02: Proceedings of the 6th International Workshop on Randomization and Approximation Techniques*, pages 254–262, 2002.

[39] A. Montanari and A. Saberi. Convergence to equilibrium in local interaction games. *Manuscript. To appear in FOCS 2009.*

[40] T. Munzner and P. Burchard. Visualizing the structure of the World Wide Web in 3D hyperbolic space. In *Proceedings of the first symposium on Virtual reality modeling language*, pages 33–38, 1995.

[41] M.E.J. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.

[42] P. Niyogi. Manifold regularization and semi-supervised learning: Some theoretical analyses. Technical Report TR-2008-01, University of Chicago, Computer Science Dept., 2008.

[43] L. K. Saul, K. Q. Weinberger, J. H. Ham, F. Sha, and D. D. Lee. Spectral methods for dimensionality reduction. In O. Chapelle, B. Schoelkopf, and A. Zien, editors, *Semisupervised Learning*, pages 293–308. MIT Press, 2006.

[44] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transcations of Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[45] Y. Song, W.-Y. Chen, H. Bai, C.-J. Lin, and E. Chang. Parallel spectral clustering. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*, pages 374–389, 2008.

[46] D.A. Spielman and N. Srivastava. Graph sparsification by effective resistances. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 563–568, 2008.

[47] N. Srebro, J. Rennie, and T. Jaakkola. Maximum margin matrix factorizations. In *Annual Advances in Neural Information Processing Systems 17: Proceedings of the 2004 Conference*, pages 682–688, 2005.

[48] C.K.I. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In *Annual Advances in Neural Information Processing Systems 13: Proceedings of the 2000 Conference*, pages 682–688, 2001.

[49] X. Zhu. Semi-supervised learning literature survey. *Manuscript.*, July 19, 2008.