You may work on the projects by yourself or with a group of 2 (need more? need justification & permission). More is expected from a two-student paper: it is a good idea to team up with someone else if the project you have in mind requires a big effort (e.g. substantial coding involved). In this situation, please clarify who did what in a seperate section [call it e.g. "Tasks"]

You will be asked to present a plan in a couple of weeks – so it is time to start thinking about projects. A separate discussion on presentations will start taking place some time in mid to late March.

This is a preliminary version!

1 Project themes.

You essentially have three options.

1. You can do a project on a topic related to sparse matrix computations in your own research. [for example : test / development of a solver for fluid flow problems, or sparse matrix methods for information retrieval]
2. Write a theoretical paper or a survey paper related to sparse matrix computation [for example a survey 'on the use of sparse matrix techniques in genomics'].
3. You can select among a few topics suggested in the next section. These are just a few examples.

2 A few suggestions

The following is a list of possible projects. This list is by no means exhaustive. The orderering is not significant. The list may be updated a couple of times during the semester. Also note that the references are given here just to get you started.

Project#1 Sparse matrix computations on GPUs – either an overview of what people have done or an implementation /comparison of your own of some techniques [in the latter case you will need to have access to a GPU - e.g. at MSI] – see, e.g., [26].

Project#2 Label propagation and applications [39, 41, 22]. This is an important and very interesting topic in semi-supervised learning. More references to be added.

Project#3 Graph Neural Networks. This is an area that is generating quite a bit of papers currently. It is about how to adapt Convolutional Networks in Deep Learning to data that consist of graphs. Here are some papers: [14, 24, 21, 20, 4, 42, 25, 7, 40, 43, 44]

Project#4 Hypergraphs and their use [47, 46, 9, 15, 31, 13, 8]

Project#5 Explore the use of sparse matrices in signal processing, e.g., the problem of Dictionary learning, see for example [30].

Project#6 Comparison of reordering techniques for direct solvers. Nested Dissection ordering, Minimal Degree algorithm, etc. You can obtain the 'meshpart' matlab toolbox from

http://www.cerfacs.fr/algor/Softs/MESHPART/

Then test various ordering strategies available (see the parameter 'method' in ndpart). Compare with any other reorderings you can get from matlab (those available from matlab are are colamd and symrcm). If you prefer you can do this project in C or Fortran and download a few reordering techniques available.

Project#7 Generate linear systems from IFISS CDF code and integrate different solvers into ifiss. Ifiss is a matlab code for finite element discretization and solution of various fluid dynamics problems. The code can be downloaded from here:

http://www.maths.manchester.ac.uk/ djs/ifiss/

Project#8 Graph coarsening is an important ingredient in multilevel iterative methods such as Algebraic MultiGrid (AMG), see, e.g.,[3], It has also appeared in work related to machine learning [1, 12, 11, 16, 17, 19, 27, 29]. A project along these lines would explore specific techniques with implementations, or present a good survey of coarsening for data-related applications – by being as broad as possible in the applications covered.

Project#9 There are very interesting applications of sparse matrix techniques in computer graphics and image processing. One such application (or a class of applications actually) is *Poisson Image Editing*. This consists of modifying images in a seamless way. Modifications may include importing a piece from another image ('cloning'), illumunation changes, background color modifications, seamless tiling, etc.. This leads to solving sparse linear systems, specifically after discretizing a Poisson equation. The main reference is [35], see also [10].

Project#10 Present an overview of methods used in image segmentation. A few good starting points: [45, 34].

Project#11 Sparse tensors and applications. Tensors have become very important due to their use in image/video processing. More recently, *Sparse* tensors have appeared as an important tool in text processing for example. Here are a couple of references: [38, 2, 37].

Project#12 Explore sparsity in Deep Neural networks.. It is known that the types of nonlinear functions used in neural networks tend to introduce sparsity. It is not clear how/whether this has been effectively exploited by practitioners so far. This could be am exciting study on a subject of great current interest. A few recent references are [28, 32, 33, 18].

Project#13 An important area of research centers around "matrix completion", i.e., the problem of finding missing entries in a given matrix. The best illustration of this is in recommender system: you are given a matrix of ratings of (e.g.) movies by individuals and you would like to guess some of those entries which are not filled up for the purpose or recommending movies (or books to by as is done in Amazon).. There are many references here - a good starting point for recommender systems is [36]. To study the problem in-depth see for example [23, 5, 6]. An idea that is key in this topic is the judicious use of norms to control sparsity and rank.

References

[1] Taha Atahan Akyildiz, Amro Alabsi Aljundi, and Kamer Kaya. Understanding coarsening for embedding large-scale graphs. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 2937–2946, 2020.

[2] M. Baskaran, B. Meister, N. Vasilache, and R. Lethin. Efficient and scalable computations with sparse tensors. In *2012 IEEE Conference on High Performance Extreme Computing*, pages 1–6, Sep. 2012.

[3] W. L. Briggs, V. E. Henson, and S. F. Mc Cormick. *A multigrid tutorial*. SIAM, Philadelphia, PA, 2000. Second edition.

[4] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, July 2017.

[5] Emmanuel Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9:717–772, 2009. 10.1007/s10208-009-9045-5.

[6] Emmanuel J. Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

[7] Catalina Cangea, Petar Velickovic, Nikola Jovanovic, Thomas Kipf, and Pietro Liò. Towards sparse hierarchical graph classifiers. *ArXiv*, abs/1811.01287, 2018.

[8] Timoteo Carletti, Federico Battiston, Giulia Cencetti, and Duccio Fanelli. Random walks on hypergraphs. *Phys. Rev. E*, 101:022308, Feb 2020.

[9] U. V. Catalyurek and C. Aykanat. Hypergraph-partitioning based decomposition for parallel sparse-matrix vector multiplication. *IEEE Transaction on Parallel and Distributed Systems*, 10(7):673–693, 1999.

[10] Tony F Chan and Jianhong Shen. *Image processing and analysis: variational, PDE, wavelet, and stochastic methods*. SIAM, 2005.

[11] Haochen Chen, Bryan Perozzi, Yifan Hu, and Steven Skiena. HARP: Hierarchical representation learning for networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018.

[12] Jie Chen, Yousef Saad, and Zechen Zhang. Graph coarsening: from scientific computing to machine learning. *SeMA Journal*, 79(1):187–223, 2022.

[13] Guilherme Ferraz de Arruda, Giovanni Petri, and Yamir Moreno. Social contagion models on hypergraphs. *Phys. Rev. Research*, 2:023032, Apr 2020.

[14] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 3844–3852, Red Hook, NY, USA, 2016. Curran Associates Inc.

[15] K.D. Devine, E.G. Boman, R.T. Heaphy, R.H. Bisseling, and U.V. Catalyurek. Parallel hypergraph partitioning for scientific computing. In *Proc. of 20th International Parallel and Distributed Processing Symposium (IPDPS'06)*. IEEE, 2006.

[16] I. S. Dhillon, Y. Guan, and B. Kulis. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(11):1944–1957, Nov 2007.

[17] Florian Dorfler and Francesco Bullo. Kron reduction of graphs with applications to electrical networks. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 60(1):150–163, 2013.

[18] Benjamin Graham. Spatially-sparse convolutional neural networks. *CoRR*, abs/1409.6070, 2014. available from arXiv:https://arxiv.org/abs/1409.6070v1.

[19] Aditya Grover and Jure Leskovec. Node2vec: Scalable feature learning for networks. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 855–864, New York, NY, USA, 2016. ACM.

[20] William L. Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications, 2017.

[21] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data, 2015.

[22] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[23] R.H. Keshavan, A. Montanari, and Sewoong Oh. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6):2980 –2998, june 2010.

[24] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2016.

[25] Junhyun Lee, Inyeop Lee, and Jaewoo Kang. Self-attention graph pooling. In *ICML*, 2019.

[26] Ruipeng Li and Yousef Saad. GPU-accelerated preconditioned iterative linear solvers. *The Journal of Supercomputing*, 63(2):443–466, 2013.

[27] Jiongqian Liang, Saket Gurukar, and Srinivasan Parthasarathy. MILE: a multi-level framework for scalable graph embedding. *ArXiv preprint arXiv:1802.09612 [cs.AI]*, 2018.

[28] Baoyuan Liu, Min Wang, Hassan Foroosh, Marshall Tappen, and Marianna Pensky. Sparse convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[29] Andreas Loukas. Graph reduction with spectral and cut guarantees. *Journal of Machine Learning Research*, 20(116):1–42, 2019.

[30] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *JMLR*, 11:19–60, 2010.

[31] M. M. Ozdal and C. Aykanat. Hypergraph models and algorithms for data-pattern-based clustering. *Data Mining and Knowledge Discovery*, 9:29–57, 2004.

[32] A. Parashar, M. Rhu, A. Mukkara, A. Puglielli, R. Venkatesan, B. Khailany, J. Emer, S. W. Keckler, and W. J. Dally. Scnn: An accelerator for compressed-sparse convolutional neural networks. In *2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA)*, pages 27–40, June 2017.

[33] JongSoo Park, Sheng R. Li, Wei Wen, Ping Tak Peter Tang, Hai Li, Yiran Chen, and Pradeep Dubey. Faster cnns with direct sparse convolutions and guided pruning. In *ICLR 2017*, 2017. Available from arXiv: https://arxiv.org/abs/1608.01409.

[34] Bo Peng, Lei Zhang, and David Zhang. A survey of graph theoretical approaches to image segmentation. *Pattern Recognition*, 46(3):1020 – 1038, 2013.

[35] P. Pérez, P. Gangnet, and A. Blake. Poisson image editing. *ACM Trans. Graphics*, 22(3):313–318, 2003.

[36] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Application of dimensionality reduction in recommender systems – a case study. In *ACM WebKDD, Web Mining for E-Commerce Workshop*, 2000.

[37] S. Smith and G. Karypis. A medium-grained algorithm for sparse tensor factorization. In *2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 902–911, May 2016.

[38] S. Smith, N. Ravindran, N. D. Sidiropoulos, and G. Karypis. Splatt: Efficient and parallel sparse tensor-matrix multiplication. In *2015 IEEE International Parallel and Distributed Processing Symposium*, pages 61–70, May 2015.

[39] Ze Tian and Rui Kuang. Global linear neighborhood for efficient label propagation. In *Proc. of SIAM International Conference on Data Mining (SDM)*, 2012. In Press.

[40] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2017.

[41] Fei Wang and Changshui Zhang. Label propagation through linear neighborhoods. *Knowledge and Data Engineering, IEEE Transactions on*, 20(1):55 –67, jan. 2008.

[42] Minjie Wang, Lingfan Yu, Da Zheng, Quan Gan, Yu Gai, Zihao Ye, Mufei Li, Jinjing Zhou, Qi Huang, Chao Ma, Ziyue Huang, Qipeng Guo, Hao Zhang, Haibin Lin, Junbo Zhao, Jinyang Li, Alexander Smola, and Zheng Zhang. Deep graph library: Towards efficient and scalable deep learning on graphs, 2019.

[43] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks?, 2018.

[44] Rex Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L. Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. In *Proceedings of the 32Nd International Conference on Neural Information Processing Systems*, NIPS'18, pages 4805–4815, USA, 2018. Curran Associates Inc.

[45] Nida M. Zaitoun and Musbah J. Aqel. Survey on image segmentation techniques. *Procedia Computer Science*, 65:797 – 806, 2015. International Conference on Communications, management, and Information technology (ICCMIT'2015).

[46] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In S. Thrun, L. Saul, and B. Schl̈kopf, editors, *Advances in Neural Information Processing Systems*, volume 16, pages 321–328, Cambridge, MA, 2004. MIT Press.

[47] D. Zhou, J. Huang, and B. Scholkopf. Learning with hypergraphs: Clustering, classification, and embedding. *Advances in Neural Information Processing Systems*, 19:1601, 2007.