Chapter 10

In the previous lectures, we studied non-convex optimization in the context of sparse feature selection and low rank recovery, where non-convexity is introduced by the constraints. We considered low-rank model selection in data science application and went beyond hard thresholding methods to discuss the non-convex path. We will now discuss the landscape of non-convex optimization problems in general, including the types of stationary points including saddle points and conditions that would allow escaping from these saddle points.

Saddle points, Matrix sensing

Non-convex optimization problems are NP-hard in general, although some specific cases can be solved in polynomial time. Neural networks, the classical example of non-convex optimization, cannot be even solved to global optimality without a fine grid search over the space of initial points. To further illustrate the difficulty of non-convex optimization, consider the example of homogeneous quartics.

Homogeneous Quartics. Homogeneous quartics are functions of the form

$$f(x) = \sum_{i,j=1}^{p} Q_{ij} x_i^2 x_j^2$$

If $Q \succeq 0$, then $f(x) \geq 0$, and $x = 0$ is the global minimum. However, if Q is arbitrary, $\nabla f(x)$ at zero is zero but zero can be a minimum, a maximum, or a saddle point. Checking if 0 is a global minimizer is equivalent to checking if there is a point that leads to a negative objective. Using a change of variable $u_i = x_i^2$, we transform the original objective function into $f(u) = u^\top Q u$. Looking for a non-negative u such that $u^\top Q u < 0$ is equivalent to checking if Q is co-positive, which is an NP-hard problem.

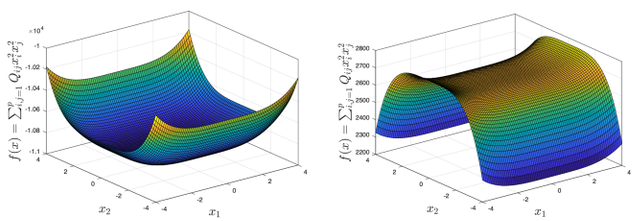

Fig. 52. For $Q \succeq 0$, 0 is the global minimum, but for any arbitrary Q, 0 can be a minimum, a maximum, or a saddle point

Previously when we studied the Newton's method, we have turned to the Hessian for information about the local curvature. However, for homogeneous quartics, $\nabla^2 f(0) = 0$, the Hessian provides no useful insights. We could use even higher-order information such as third or fourth-order derivatives, but that would propel the problem into the realm of NP-hardness. Hence, we see that in non-convex optimization, determining the identity of a stationary point is a difficult task in and of itself. Even if we were at the global minimum, proving that our solution is indeed globally minimum is NP-hard. This challenge is not only found in homogeneous quartics but in many other non-convex problems: quadratic combinatorial optimization (QCOP), matrix completion and sensing, tensor decomposition, etc.

Local minima: the next best thing to global minimum. Recall that a critical or stationary point, $x^\star$ where $\nabla f(x^\star) = 0$, can be one of the following

- Global minima: all directions go upwards and $f(x^\star) \leq f(x), \forall x$ these are desirable but not easily attainable
- Local minima: all directions go upwards and maybe $f(x^\star) \geq f(x), \exists x$; these are the next best thing
- Saddle points: there are upwards, downwards, and/or flat directions

Having seen convex optimization algorithms and studied their convergence to globally optimal solutions, we may be averse to accepting local minima as solutions to non-convex problems. However, for larger models like neural networks, local minima tend to yield similar loss values as the global minimum. While poor local minima exist, it has been shown that the probability of convergence to a poor local minimum is near zero for some models. This is consistent with the fact that, in practice, training a neural network with different random seeds often leads to models that perform similarly well.

Motivation for escaping saddle points. With the knowledge that good local minima exist for some non-convex optimization problems, convergence to local minima can still be a challenging process. Saddle points can stall the convergence to a good quality local minimum. In the optimization landscape, saddle points can be large plateaus or flat regions where the slope is very slow. Saddle points can dramatically slow down learning, giving the illusion that we have reached a local minimum. Recall that for a generic smooth function, the update according to gradient descent is given by $x_{t+1} = x_t - \frac{1}{L}\nabla f(x_t)$. As t increases, gradient descent converges to the points where the gradient has zero energy.

Saddle points can be a particularly ubiquitous issue for optimization in high dimensions, as saddle points emerge and their numbers may even increase exponentially with increasing dimensionality. To illustrate this, consider the example $f(x) = (x^2 - 2)^2$, which has two local/global minima, one local maximum, and no saddle points.

Extending the same function from 1D to 2D, $f(x, y) = (x^2 - 2)^2 + (y^2 - 2)^2 + 8$, saddle points emerge. The 2D function's landscape resembles an egg holder, and there are 4 saddle points, one between each "slot". From 1D to 2D, the number of saddle points has increased from 0 to 4. In fact, the number of saddle points will continue to increase with higher dimensions. For the same function in 3D, we will get 8 saddle points.

Escaping saddle points: Second-order derivative test. Consider the Hessian, $\nabla^2 f(x) \in \mathbb{R}^{1\times 1}$, at a critical point x. The Hessian is square and symmetric, which means that we can compute its eigendecomposition and characterize the critical

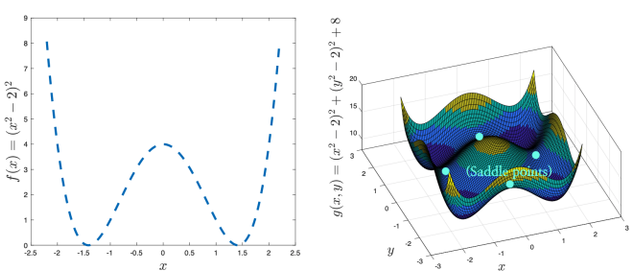

Fig. 53. Saddle points emerge and increase in number with higher dimensionality

point based on the signs of its eigenvalues. If $\nabla^2 f(x)$ has only positive eigenvalues, then the critical point x is a local minimum. To prove this, consider the second-order Taylor's expansion and the fact that $\nabla f(x) = 0$ at the critical point.

$$f(x + \eta u) = f(x) + \eta \langle \nabla f(x), u \rangle + \frac{\eta^2}{2} \langle \nabla^2 f(x) u, u \rangle$$
$$= f(x) + \frac{\eta^2}{2} \langle \nabla^2 f(x) u, u \rangle > f(x)$$

Hence, when $\nabla^2 f(x)$ has only positive eigenvalues, all directions go upwards, and the critical point is a local minimum.

Based on similar reasoning, we can devise the following rules for characterizing a critical point.

- Only positive eigenvalues: local minimum
- Only negative eigenvalues: local maximum
- Only positive and negative eigenvalues: strict saddle point
- Positive, negative, and zero eigenvalues: general saddle point

At a saddle point, strict or general, the objective function decreases in the direction of the eigenvector that corresponds to a negative eigenvalue. By following the direction of this eigenvector, we can escape a saddle point.

Strict saddle property. A function $f(x)$ satisfies the strict saddle property, if all points x in its domain satisfies the at least one of the following:

- The gradient is large, i.e. $\|\nabla f(x)\|_2 \geq \alpha$
- The Hessian has at least one negative eigenvalue, bounded away from zero, i.e. $\lambda_{\min}(\nabla^2 f(x)) \leq -\gamma$
- x is near a local minimum

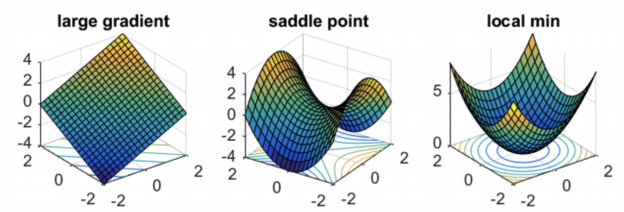

Fig. 54. Strict saddle property

For a function that satisfies this property, the minimum eigenvalue of $\nabla^2 f(x)$ is bounded by a negative value. Therefore, there is always an escape route from a saddle point of a function that satisfies this property. However, finding the minimum eigenvalue requires computing the eigendecomposition of the Hessian, which has $\mathcal{O}(p^3)$ complexity. Some existing methods such as cubic regularization and trust-region methods do not compute the full eigendecomposition but are nonetheless time consuming in practice, as they require second-order information from the Hessian.

Noisy gradient descent. The good news is that it is possible to escape from saddle points using first-order methods such as gradient descent. Although gradient at saddle points is null, strict saddle points are quite unstable. At a strict saddle point where the Hessian has no zero eigenvalue, if we perturb the our location even by just a little bit, we will fall and escape from the saddle point. We can incorporate this perturbation in the form of noise into the gradient descent step.

$$x_{t+1} = x_t - \eta \nabla f(x_t) + \epsilon, \epsilon \sim \eta \cdot \mathcal{S}^{p-1}$$

Alternatively, an even easier approach is simply using the stochastic gradient descent, which naturally has noise incorporated into each step.

$$x_{t+1} = x_t - \eta \nabla f_{i_t}(x_t) = x_t - \eta \nabla f(x_t) + \epsilon, \epsilon = \eta(\nabla f(x_t) - \nabla f_{i_t}(x_t))$$

It has been proven that noisy gradient descent finds a local minimum of an objective function that satisfies the strict saddle property in polynomial time, up to $\mathcal{O}(\frac{1}{\epsilon^4})$ iterations. To put this result in perspective, convergence of gradient descent to a critical point (not necessarily a local minimum) has a running time of $\mathcal{O}(\frac{1}{\epsilon^2})$, the noisy gradient descent converges to a local minimum but at the cost of more iterations.

A different perspective on saddle points. We have seen that strict saddle points are highly unstable, and we can escape from them with a little perturbation. Another important perspective to consider is that convergence to saddle points depends strongly on initialization. Consider the 2D example $f(x, y) = \frac{1}{2}x^2 + \frac{1}{4}y^4 - \frac{1}{2}y^2$. The only saddle point is $(0, 0)$, and to converge to this saddle point, initialization has to be of the form $(x, 0)$, which in the case of random initialization, occurs with a probability of 0. In practice, if you pick any random initial point, you are safe not to converge to a saddle point.

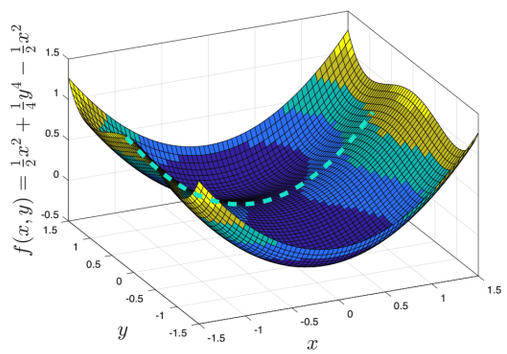

Fig. 55. Initialization has to be along the dotted line to converge to the saddle point $(0, 0)$

Matrix sensing using RIP and PSD matrix factorization. In the previous section, methods to escape saddle points were discussed. Related questions are how to escape local minima and whether we can infer that local minima are just as good as global minima in the non-convex setting. To explore this question, the following will cover matrix sensing using RIP and PSD matrix factorization.

The setting is

$$y = A(x^*)$$

where x is rank r, $x^* \in \mathbb{R}^{n \times n}$, and $x^* \geq 0$. $A(\cdot) : \mathbb{R}^{n \times n} \to \mathbb{R}^m$ where $m << n^2$. $y \in \mathbb{R}^m$.

Because x is PSD, $x^* = UU^T$. Hence, we are interested in

$$\min_{U \in \mathbb{R}^{n \times r}} \{f(u) := \|y - A(UU^T)\|_2^2\}$$

and we solve this problem using the recursion

$$U_{t+1} = U_t - \eta \nabla f(U_t U_t^T) \cdot U_t$$

The following assumption is made

$$\text{RIP} : (1-\delta)\|x\|_F^2 \leq \frac{1}{m} \cdot \|A(x)\|_2^2 \leq (1+\delta)\|x\|_F^2$$

which then leads to

$$(1-\delta)\|x\|_F^2 \leq \frac{1}{m}\sum_{i=1}^{m}\langle A_i, x\rangle^2 \leq (1+\delta)\|x\|_F^2$$

A corollary of RIP is

$$|\frac{1}{m}\sum_{i=1}^{m}\langle A_i, x\rangle\langle A_i, y\rangle - \langle x, y\rangle| \leq \delta\|x\|_F\|y\|_F$$

U is a stationary point if $\nabla f(U) = 0$ which then means that $-2A^+(y - A(UU^T))U = 0$ and that $\sum_{i=1}^{m}\langle A_i, UU^T - U^*U^{*T}\rangle A_i U = 0$. This means we can relate the stationary point to the global point. Note that $\sum_{i=1}^{m}\langle A_i, UU^T - U^*U^{*T}\rangle A_i U \in \mathbb{R}^{n\times r}$. This last point means that we can also find

$$\sum_{i=1}^{m}\langle A_i, UU^T - U^*U^{*T}\rangle\langle A_i U, V\rangle = 0$$

where we have the freedom to choose V, UU^T is the local and U^*U^{*T} optimum. U can further be decomposed as $U = QR$ which implies $V = ZQR^{-1}$ such that

$$|\langle UU^T - U^*U^{*T}, QQ^TZ^T\rangle|_{\leq}\delta\|UU^T - U^*U^{*T}\|_F\|QQ^TZ^T\|_F$$

which then further implies

$$\|(UU^T - U^*U^{*T})QQ^T\|_F \leq \delta\|UU^T - U^*U^{*T}\|_F$$

Note that $U \in \mathbb{R}^{n\times r}$ which means that $\nabla^2 f(u) \in \mathbb{R}^{nr\times nr}$. If U is a local minimum, then

$$vec(z)^T\nabla^2 f(u)vect(z) \succcurlyeq 0 \forall z \in \mathbb{R}^{n\times r}$$

Using

$$\lim_{t\to 0}[\nabla f(u+tz) - \nabla f(u)]$$

and second-order optimality, the following holds

1. $\|U(U - U^*R)\|_F^2 \geq \frac{1-\delta}{2(1+\delta)}\|UU^T - U^*U^{*T}\|_f^2$
2. $\|U(U - U^*R)\|_F^2 \leq \frac{1}{8}\|UU^T - U^*U^{*T}\|_F^2 + \frac{34}{8}\|(UU^T - U^*U^{*T}QQ^T\|_F^2$

Combining inequalities we then have,

$$\left(\frac{1-\delta}{2(1+\delta)} - \frac{1}{8}\right)\|UU^T - U^*U^{*T}\|_F^2 \leq \frac{34\delta^2}{8}\|UU^T - U^*U^{*T}\|_F^2$$

$$\left(\frac{1-\delta}{2(1+\delta)} - \frac{1}{8} - \frac{34\delta^2}{8}\right)\|UU^T - U^*U^{*T}\|_F^2 \leq 0$$

If the first term $\left(\frac{1-\delta}{2(1+\delta} - \frac{1}{8} - \frac{34\delta^2}{8}\right)$ is forced to be greater than 0, because δ is a hyperparameter that cannot be controlled due to the RIP assumption, the local minimum should be identical to the global minimum. More specifically if $\delta \leq 1/5$, all local minimum are equal to global minimum.

Lastly, for saddle points if $\lambda_{\min}(\frac{1}{m}\nabla^2 f(u)) \leq \frac{-4}{5}\sigma_r(x^*)$, there is always a direction of escape. If the r-th singular value is very small, it is not easy to escape from a saddle point and the matrix is very close to being rank $r+1$.

Appendix

1. J. Nocedal and S. Wright. Numerical optimization. Springer Science & Business Media, 2006.
2. Y. Nesterov. Introductory lectures on convex optimization: A basic course, volume 87. Springer Science & Business Media, 2013.
3. S. Boyd and L. Vandenberghe. Convex optimization. Cambridge university press, 2004.
4. D. Bertsekas. Convex optimization algorithms. Athena Scientific Belmont, 2015.
5. Sébastien Bubeck. Convex optimization: Algorithms and complexity. Foundations and Trends® in Machine Learning, 8(3-4):231–357, 2015.
6. S. Weisberg. Applied linear regression, volume 528. John Wiley & Sons, 2005.
7. T. Hastie, R. Tibshirani, and M. Wainwright. Statistical learning with sparsity: the lasso and generalizations. CRC press, 2015.
8. J. Friedman, T. Hastie, and R. Tibshirani. The elements of statistical learning, volume 1. Springer series in statistics New York, 2001.
9. M. Paris and J. Rehacek. Quantum state estimation, volume 649. Springer Science & Business Media, 2004.
10. M. Daskin. A maximum expected covering location model: formulation, properties and heuristic solution. Transportation science, 17(1):48–70, 1983.
11. I. Goodfellow, Y. Bengio, and A. Courville. Deep learning. MIT press, 2016.
12. L. Trefethen and D. Bau III. Numerical linear algebra, volume 50. Siam, 1997.
13. G. Strang. Introduction to linear algebra, volume 3. Wellesley-Cambridge Press Wellesley, MA, 1993.
14. G. Golub. Cmatrix computations. The Johns Hopkins, 1996.
15. A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.
16. K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.
17. S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99, 2015.
18. T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, pages 3111–3119, 2013.
19. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.
20. Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pages 1243–1252. JMLR. org, 2017.
21. Haşim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In Fifteenth annual conference of the international speech communication association, 2014.
22. Tom Sercu, Christian Puhrsch, Brian Kingsbury, and Yann LeCun. Very deep multilingual convolutional neural networks for LVCSR. In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 4955–4959. IEEE, 2016.
23. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. page arXiv:1706.03762, 2017.
24. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. page arXiv:1810.04805, 2018.
25. Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and VQA. In AAAI, pages 13041–13049, 2020.
26. Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. arXiv preprint arXiv:2005.14165, 2020.
27. Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using gpu model parallelism. arXiv preprint arXiv:1909.08053, 2019.
28. Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:1910.10683, 2019.
29. Gary Marcus, Ernest Davis, and Scott Aaronson. A very preliminary analysis of DALL-E 2. arXiv preprint arXiv:2204.13807, 2022.
30. John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with AlphaFold. Nature, 596(7873):583–589, 2021.
31. Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.
32. Or Sharir, Barak Peleg, and Yoav Shoham. The cost of training nlp models: A concise overview. arXiv preprint arXiv:2004.08900, 2020.
33. H. Karimi, J. Nutini, and M. Schmidt. Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pages 795–811. Springer, 2016.
34. Philip Wolfe. Convergence conditions for ascent methods. SIAM review, 11(2):226–235, 1969.
35. Larry Armijo. Minimization of functions having Lipschitz continuous first partial derivatives. Pacific Journal of mathematics, 16(1):1–3, 1966.
36. Stephen Wright and Jorge Nocedal. Numerical optimization. Springer Science, 35(67-68):7, 1999.
37. B. Polyak. Introduction to optimization. Inc., Publications Division, New York, 1, 1987.
38. Stephen Boyd, Lin Xiao, and Almir Mutapcic. Subgradient methods. lecture notes of EE392o, Stanford University, Autumn Quarter, 2004:2004–2005, 2003.
39. Marguerite Frank, Philip Wolfe, et al. An algorithm for quadratic programming. Naval research logistics quarterly, 3(1-2):95–110, 1956.
40. M. Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In Proceedings of the 30th international conference on machine learning, number CONF, pages 427–435, 2013.
41. J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the ℓ_1-ball for learning in high dimensions. In Proceedings of the 25th international conference on Machine learning, pages 272–279, 2008.
42. Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. Computer, (8):30–37, 2009.
43. A. Mnih and R. Salakhutdinov. Probabilistic matrix factorization. In Advances in neural information processing systems, pages 1257–1264, 2008.
44. T. Booth and J. Gubernatis. Improved criticality convergence via a modified Monte Carlo power iteration method. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.
45. S. Zavriev and F. Kostyuk. Heavy-ball method in nonconvex optimization problems. Computational Mathematics and Modeling, 4(4):336–341, 1993.
46. E. Ghadimi, H. Feyzmahdavian, and M. Johansson. Global convergence of the heavy-ball method for convex optimization. In 2015 European control conference (ECC), pages 310–315. IEEE, 2015.
47. Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(\frac{1}{k^2})$. In Soviet Mathematics Doklady, volume 27, pages 372–376, 1983.
48. B. O'Donoghue and E. Candes. Adaptive restart for accelerated gradient schemes. Foundations of computational mathematics, 15(3):715–732, 2015.
49. O. Devolder, F. Glineur, and Y. Nesterov. First-order methods of smooth convex optimization with inexact oracle. Mathematical Programming, 146(1-2):37–75, 2014.
50. L. Bottou, F. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. Siam Review, 60(2):223–311, 2018.
51. S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. SIAM review, 43(1):129–159, 2001.
52. R. Tibshirani. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1):267–288, 1996.
53. P. Hoff. Lasso, fractional norm and structured sparse estimation using a Hadamard product parametrization. Computational Statistics & Data Analysis, 115:186–198, 2017.
54. S. Becker, J. Bobin, and E. Candès. NESTA: A fast and accurate first-order method for sparse recovery. SIAM Journal on Imaging Sciences, 4(1):1–39, 2011.
55. T. Blumensath and M. Davies. Iterative hard thresholding for compressed sensing. Applied and computational harmonic analysis, 27(3):265–274, 2009.
56. D. Needell and J. Tropp. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. Applied and computational harmonic analysis, 26(3):301–321, 2009.
57. S. Foucart. Hard thresholding pursuit: an algorithm for compressive sensing. SIAM Journal on Numerical Analysis, 49(6):2543–2563, 2011.
58. J. Tanner and K. Wei. Normalized iterative hard thresholding for matrix completion. SIAM Journal on Scientific Computing, 35(5):S104–S125, 2013.
59. K. Wei. Fast iterative hard thresholding for compressed sensing. IEEE Signal processing letters, 22(5):593–597, 2014.
60. Rajiv Khanna and Anastasios Kyrillidis. Iht dies hard: Provable accelerated iterative hard thresholding. In International Conference on Artificial Intelligence and Statistics, pages 188–198. PMLR, 2018.
61. Jeffrey D Blanchard and Jared Tanner. GPU accelerated greedy algorithms for compressed sensing. Mathematical Programming Computation, 5(3):267–304, 2013.
62. A. Kyrillidis, G. Puy, and V. Cevher. Hard thresholding with norm constraints. In 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 3645–3648. Ieee, 2012.
63. A. Kyrillidis and V. Cevher. Recipes on hard thresholding methods. In Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2011 4th IEEE International Workshop on, pages 353–356. IEEE, 2011.
64. X. Zhang, Y. Yu, L. Wang, and Q. Gu. Learning one-hidden-layer ReLU networks via gradient descent. In The 22nd International Conference on Artificial Intelligence and Statistics, pages 1524–1534, 2019.

65. Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. IEEE Transactions on information theory, 52(2):489–509, 2006.
66. Joseph B Altepeter, Daniel FV James, and Paul G Kwiat. 4 qubit quantum state tomography. In Quantum state estimation, pages 113–145. Springer, 2004.
67. Jens Eisert, Dominik Hangleiter, Nathan Walk, Ingo Roth, Damian Markham, Rhea Parekh, Ulysse Chabaud, and Elham Kashefi. Quantum certification and benchmarking. arXiv preprint arXiv:1910.06343, 2019.
68. Masoud Mohseni, AT Rezakhani, and DA Lidar. Quantum-process tomography: Resource analysis of different strategies. Physical Review A, 77(3):032322, 2008.
69. D. Gross, Y.-K. Liu, S. Flammia, S. Becker, and J. Eisert. Quantum state tomography via compressed sensing. Physical review letters, 105(15):150401, 2010.
70. Y.-K. Liu. Universal low-rank matrix recovery from Pauli measurements. In Advances in Neural Information Processing Systems, pages 1638–1646, 2011.
71. K Vogel and H Risken. Determination of quasiprobability distributions in terms of probability distributions for the rotated quadrature phase. Physical Review A, 40(5):2847, 1989.
72. Miroslav Ježek, Jaromír Fiurášek, and Zdeněk Hradil. Quantum inference of states and processes. Physical Review A, 68(1):012305, 2003.
73. Konrad Banaszek, Marcus Cramer, and David Gross. Focus on quantum tomography. New Journal of Physics, 15(12):125020, 2013.
74. A. Kalev, R. Kosut, and I. Deutsch. Quantum tomography protocols with positivity are compressed sensing protocols. Nature partner journals (npj) Quantum Information, 1:15018, 2015.
75. Giacomo Torlai, Guglielmo Mazzola, Juan Carrasquilla, Matthias Troyer, Roger Melko, and Giuseppe Carleo. Neural-network quantum state tomography. Nat. Phys., 14:447–450, May 2018.
76. Matthew JS Beach, Isaac De Vlugt, Anna Golubeva, Patrick Huembeli, Bohdan Kulchytskyy, Xiuzhe Luo, Roger G Melko, Ejaaz Merali, and Giacomo Torlai. Qucumber: wavefunction reconstruction with neural networks. SciPost Physics, 7(1):009, 2019.
77. Giacomo Torlai and Roger Melko. Machine-learning quantum states in the NISQ era. Annual Review of Condensed Matter Physics, 11, 2019.
78. M. Cramer, M. B. Plenio, S. T. Flammia, R. Somma, D. Gross, S. D. Bartlett, O. Landon-Cardinal, D. Poulin, and Y.-K. Liu. Efficient quantum state tomography. Nat. Comm., 1:149, 2010.
79. BP Lanyon, C Maier, Milan Holzäpfel, Tillmann Baumgratz, C Hempel, P Jurcevic, Ish Dhand, AS Buyskikh, AJ Daley, Marcus Cramer, et al. Efficient tomography of a quantum many-body system. Nature Physics, 13(12):1158–1162, 2017.
80. D. Gonçalves, M. Gomes-Ruggiero, and C. Lavor. A projected gradient method for optimization over density matrices. Optimization Methods and Software, 31(2):328–341, 2016.
81. E. Bolduc, G. Knee, E. Gauger, and J. Leach. Projected gradient descent algorithms for quantum state tomography. npj Quantum Information, 3(1):44, 2017.
82. Jiangwei Shang, Zhengyun Zhang, and Hui Khoon Ng. Superfast maximum-likelihood reconstruction for quantum tomography. Phys. Rev. A, 95:062336, Jun 2017.
83. Zhilin Hu, Kezhi Li, Shuang Cong, and Yaru Tang. Reconstructing pure 14-qubit quantum states in three hours using compressive sensing. IFAC-PapersOnLine, 52(11):188 – 193, 2019. 5th IFAC Conference on Intelligent Control and Automation Sciences ICONS 2019.
84. Zhibo Hou, Han-Sen Zhong, Ye Tian, Daoyi Dong, Bo Qi, Li Li, Yuanlong Wang, Franco Nori, Guo-Yong Xiang, Chuan-Feng Li, et al. Full reconstruction of a 14-qubit state within four hours. New Journal of Physics, 18(8):083036, 2016.
85. C. Riofrío, D. Gross, S.T. Flammia, T. Monz, D. Nigg, R. Blatt, and J. Eisert. Experimental quantum compressed sensing for a seven-qubit system. Nature Communications, 8, 2017.
86. Martin Kliesch, Richard Kueng, Jens Eisert, and David Gross. Guaranteed recovery of quantum processes from few measurements. Quantum, 3:171, 2019.
87. S. Flammia, D. Gross, Y.-K. Liu, and J. Eisert. Quantum tomography via compressed sensing: Error bounds, sample complexity and efficient estimators. New Journal of Physics, 14(9):095022, 2012.
88. A. Kyrillidis, A. Kalev, D. Park, S. Bhojanapalli, C. Caramanis, and S. Sanghavi. Provable quantum state tomography via non-convex methods. npj Quantum Information, 4(36), 2018.
89. B. Recht, M. Fazel, and P. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. SIAM review, 52(3):471–501, 2010.
90. N. Srebro, J. Rennie, and T. Jaakkola. Maximum-margin matrix factorization. In Advances in neural information processing systems, pages 1329–1336, 2004.
91. J. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In Proceedings of the 22nd international conference on Machine learning, pages 713–719. ACM, 2005.
92. D. DeCoste. Collaborative prediction using ensembles of maximum margin matrix factorizations. In Proceedings of the 23rd international conference on Machine learning, pages 249–256. ACM, 2006.
93. J. Bennett and S. Lanning. The Netflix prize. In Proceedings of KDD cup and workshop, volume 2007, page 35, 2007.
94. M. Jaggi and M. Sulovsk. A simple algorithm for nuclear norm regularized problems. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), pages 471–478, 2010.
95. R. Keshavan. Efficient algorithms for collaborative filtering. PhD thesis, Stanford University, 2012.
96. R. Agrawal, A. Gupta, Y. Prabhu, and M. Varma. Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In Proceedings of the 22nd international conference on World Wide Web, pages 13–24. International World Wide Web Conferences Steering Committee, 2013.
97. K. Bhatia, H. Jain, P. Kar, M. Varma, and P. Jain. Sparse local embeddings for extreme multi-label classification. In Advances in Neural Information Processing Systems, pages 730–738, 2015.
98. G. Carneiro, A. Chan, P. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 29(3):394–410, 2007.
99. A. Makadia, V. Pavlovic, and S. Kumar. A new baseline for image annotation. In Computer Vision–ECCV 2008, pages 316–329. Springer, 2008.
100. C. Wang, S. Yan, L. Zhang, and H.-J. Zhang. Multi-label sparse coding for automatic image annotation. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pages 1643–1650. IEEE, 2009.
101. J. Weston, S. Bengio, and N. Usunier. WSABIE: Scaling up to large vocabulary image annotation. In IJCAI, volume 11, pages 2764–2770, 2011.
102. Andrew I. Schein, Lawrence K. Saul, and Lyle H. Ungar. A generalized linear model for principal component analysis of binary data. In AISTATS, 2003.
103. K.-Y. Chiang, C.-J. Hsieh, N. Natarajan, I. Dhillon, and A. Tewari. Prediction and clustering in signed networks: A local to global perspective. The Journal of Machine Learning Research, 15(1):1177–1213, 2014.
104. C. Johnson. Logistic matrix factorization for implicit feedback data. Advances in Neural Information Processing Systems, 27, 2014.
105. Koen Verstrepen. Collaborative Filtering with Binary, Positive-only Data. PhD thesis, University of Antwerpen, 2015.
106. N. Gupta and S. Singh. Collectively embedding multi-relational data for predicting user preferences. arXiv preprint arXiv:1504.06165, 2015.
107. Y. Liu, M. Wu, C. Miao, P. Zhao, and X.-L. Li. Neighborhood regularized logistic matrix factorization for drug-target interaction prediction. PLoS Computational Biology, 12(2):e1004760, 2016.
108. S. Aaronson. The learnability of quantum states. In Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, volume 463, pages 3089–3114. The Royal Society, 2007.
109. E. Candes, Y. Eldar, T. Strohmer, and V. Voroninski. Phase retrieval via matrix completion. SIAM Review, 57(2):225–251, 2015.
110. I. Waldspurger, A. d'Aspremont, and S. Mallat. Phase recovery, MaxCut and complex semidefinite programming. Mathematical Programming, 149(1-2):47–81, 2015.
111. P. Biswas, T.-C. Liang, K.-C. Toh, Y. Ye, and T.-C. Wang. Semidefinite programming approaches for sensor network localization with noisy distance measurements. IEEE transactions on automation science and engineering, 3(4):360, 2006.
112. K. Weinberger, F. Sha, Q. Zhu, and L. Saul. Graph Laplacian regularization for large-scale semidefinite programming. In Advances in Neural Information Processing Systems, pages 1489–1496, 2007.
113. F. Lu, S. Keles, S. Wright, and G. Wahba. Framework for kernel regularization with application to protein clustering. Proceedings of the National Academy of Sciences of the United States of America, 102(35):12332–12337, 2005.
114. H. Andrews and C. Patterson III. Singular value decomposition (SVD) image coding. Communications, IEEE Transactions on, 24(4):425–432, 1976.
115. M. Fazel, H. Hindi, and S. Boyd. Rank minimization and applications in system theory. In American Control Conference, 2004. Proceedings of the 2004, volume 4, pages 3273–3278. IEEE, 2004.
116. E. Candès and B. Recht. Exact matrix completion via convex optimization. Foundations of Computational mathematics, 9(6):717–772, 2009.
117. P. Jain, R. Meka, and I. Dhillon. Guaranteed rank minimization via singular value projection. In Advances in Neural Information Processing Systems, pages 937–945, 2010.
118. S. Becker, V. Cevher, and A. Kyrillidis. Randomized low-memory singular value projection. In 10th International Conference on Sampling Theory and Applications (Sampta), 2013.
119. L. Balzano, R. Nowak, and B. Recht. Online identification and tracking of subspaces from highly incomplete information. In Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on, pages 704–711. IEEE, 2010.
120. K. Lee and Y. Bresler. ADMiRA: Atomic decomposition for minimum rank approximation. Information Theory, IEEE Transactions on, 56(9):4402–4416, 2010.
121. A. Kyrillidis and V. Cevher. Matrix recipes for hard thresholding methods. Journal of mathematical imaging and vision, 48(2):235–265, 2014.
122. Z. Lin, M. Chen, and Y. Ma. The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. arXiv preprint arXiv:1009.5055, 2010.
123. S. Becker, E. Candès, and M. Grant. Templates for convex cone problems with applications to sparse signal recovery. Mathematical Programming Computation, 3(3):165–218, 2011.
124. J. Cai, E. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. SIAM Journal on Optimization, 20(4):1956–1982, 2010.
125. Y. Chen, S. Bhojanapalli, S. Sanghavi, and R. Ward. Coherent matrix completion. In Proceedings of The 31st International Conference on Machine Learning, pages 674–682, 2014.
126. A. Yurtsever, Q. Tran-Dinh, and V. Cevher. A universal primal-dual convex optimization framework. In Advances in Neural Information Processing Systems 28, pages 3132–3140. 2015.