Chapter 8

In this chapter, we discuss the problem of sparse model selection, i.e., how to perform optimization when the desired/unknown model is constrained by sparsity. While the problem has natural convex translations, we study the iterative hard thresholding (IHT) algorithm and prove its performance results.

Sparse model selection | Iterative hard thresholding

Over the run of this course, we have mostly been discussing problems of the form:

$$\min_{x\in\mathcal{C}\subseteq\mathbb{R}^p} f(x),$$

where $f(\cdot)$ represents a convex objective, and $x \in \mathcal{C}$ represents some constraint. In this lecture, we will be discussing a case in which $\mathcal{C} \subseteq \mathbb{R}^p$ is *not* a convex constraint, namely when it is the requirement that x be k-sparse. To think about this problem, let us introduce its simplest non-trivial version of this problem: the *sparse linear regression* problem, defined as follows:

$$\begin{aligned} \underset{x\in\mathbb{R}^p}{\text{minimize}} \quad & \tfrac{1}{2}\|y - Ax\|_2^2 \\ \text{subject to} \quad & \|x\|_0 \leq k. \end{aligned}$$

Here, $A \in \mathbb{R}^{n\times p}$ and $y \in \mathbb{R}^n$. Linear regression problems can have a "teacher" generative model assumption where $y = Ax^\star$ + noise, where $x^\star$ is the unknown k-sparse signal we look for. This problem is interesting when we restrict that $n \ll p$; i.e., the problem is ill-posed, and classical linear algebra solvers on sets of linear equations do not necessarily recover $x^\star$.

Let us first discuss some procedures that deal with this problem.

- The above problem is non-convex: the inclusion of the ℓ_0-pseudonorm makes the problem non-convex. Classical approaches include convexification: the tightest convex relaxation of the ℓ_0-pseudonorm is that of the ℓ_1-norm (assuming bounded energy on the initial non-convex set). This leads to the re-definition of the problem as:

  $$\begin{aligned} \underset{x\in\mathbb{R}^p}{\text{minimize}} \quad & \tfrac{1}{2}\|y - Ax\|_2^2 \\ \text{subject to} \quad & \|x\|_1 \leq \lambda. \end{aligned}$$

  There is long-listed literature on this subject [54–57]; e.g., look into the Rice DSP list of compressed sensing papers (`https://dsp.rice.edu/cs/`). One caveat of this approach is that the λ hyperparameter/regularization parameter is not intuitive to be set up correctly (while sparsity k is easier to set up).
- An alternative formulation uses the notion of proximal operators and proximal gradient descent:

  $$\underset{x\in\mathbb{R}^p}{\text{minimize}} \quad \tfrac{1}{2}\|y - Ax\|_2^2 + \rho\|x\|_1.$$

  This formulation "moves" the convex ℓ_1-norm constraint into the objective and uses the following update rule

  $$x_{t+1} = \text{Prox}_{\rho\|\cdot\|_1}(x_t - \eta\nabla f(x_t)).$$

  The ℓ_1-norm in the objective "biases" the solution towards sparsity (could be seen as an approximation to the exact ℓ_1-norm projection). Like the case above, selecting the ρ value to achieve good performance is unclear.
- Finally, one could keep the non-convexity and use non-convex projected gradient descent. This leads to

  $$x_{t+1} = \mathcal{H}_k(x_t - \eta\nabla f(x_t)).$$

  This is perhaps somewhat like sorting the input concerning magnitude and selecting the k largest ones. Similarly to the above cases, it is not trivial to set up k; yet, in many cases, choosing k is more intuitive (remember, this is an integer value) than selecting a continuous-valued regularization parameter like λ and ρ.

The ℓ_0-pseudonorm generally introduces hardness in the problem definition since it suggests we solve the problem in a *combinatorial* way. We are looking for the active support set with k elements and the values for the corresponding active entries. If we try to select the k size subsets from p, we experience a combinatorial explosion. However, the key to focus on here is the word "in general": This chapter will focus on problem cases where randomness is enough to lead to an exception to this rule and admit polynomial complexity.

For this chapter, we will mostly focus on the *iterative hard thresholding algorithm* [58–67] or, IHT for short. In IHT, we have:

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

where

$$H_k(z) = \underset{\|x_0\|\leq k}{\text{argmin}}\|x - z\|_2^2.$$

Before continuing the discussion on this algorithm, let us first get a sense of what the hyperparameters we are dealing with are:

- Starting point x_0;
- Step size selection η;
- Sparsity level choice k.

For a second, let's imagine that we were dealing with the simple case of $A = I$, i.e., A is the identity matrix in $\mathbb{R}^{p\times p}$. Note that this is an oversimplification of the problem: in this case, $n = p$ by definition of the identity matrix. Then, we would end up with a new problem formulation:

$$\begin{aligned} \underset{x\in\mathbb{R}^p}{\text{minimize}} \quad & f(x) = \|y - x\|_2^2 \\ \text{subject to} \quad & \|x\|_0 \leq k. \end{aligned}$$

We have seen this problem before in an earlier homework assignment—. This is not difficult to solve. In this scenario, the problem is the simple projection step $H_k(\cdot)$ as defined above. What this problem reformulation tells is the following: given *enough* data y (in this particular case, also non-perturbed data since we do not observe $y = Ax^\star$, but $y = x^\star$), the problem is easy to solve in closed form solution, *even if the problem involves a combinatorially-hard operation; that of a sparse projection.* I.e., we know that $H_k(\cdot)$ introduces some complexity to the overall problem, but there are cases where this does not always create issues.

Isometry and restricted isometries.. Where we should direct our attention is when one deviates from $A = I$ and starts *i*) perturbing the measurements as in $y = Ax^\star$, and *ii*) even more importantly, what happens when $n \ll p$, i.e., we do not have enough measurements to solve the problem with a matrix inversion.

Focus on the following expression - it always holds for $x_1, x_2 \in \mathbb{R}^p$ and for all $\delta \in [0, 1)$:

$$(1 - \delta)\|x_1 - x_2\|_2^2 \leq \|I(x_1 - x_2)\|_2^2 \leq (1 + \delta)\|x_1 - x_2\|_2^2.$$

The inequalities above hold with equality for $\delta = 0$. What is the purpose of these inequalities? They show *how much the geometry of the vector $x_1 - x_2$ changes when someone applies the operator I on the vector $x_1 - x_2$*. To see this clearly, the left and right-hand sides of the above expressions indicate by how much "energy" we deviate from the true image $x_1 - x_2$ (when $\delta > 0$) when we apply $x_1 - x_2$ on I. For this toy example, of course, as we mentioned above, we do not lose anything: the above expressions hold with equality for $\delta = 0$.

The above lead to the notion of *isometry*: Intuitively, this means that the matrix I does not perturb the distance between x_1 and x_2 too much, in the sense that the resulting image $I(x_1 - x_2)$ is identical to that of $x_1 - x_2$. The question becomes interesting when one deviates from I; e.g., under which conditions do the above expressions hold for some A matrix and some δ? Also, does this hold for any vectors x_1, x_2, or should they satisfy some constraints?

The above leads us to the definition of the *restricted isometry property* for sparse vectors.

Definition 31. (Restricted Isometry Property (RIP) [68]) *A matrix $A \in \mathbb{R}^{n\times p}$ where $n \leq p$ satisfies the RIP with constant $\delta_k \in (0,1)$ if and only if:*

$$(1-\delta_k)\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1+\delta_k)\|x\|_2^2,$$

$\forall x \in \mathbb{R}^p$ such that $\|x\|_0 \leq k$.

In the literature, other properties like null space and eigenvalues are considered. Still, for this lecture, we will focus on the restricted isometry property (henceforth RIP) and the analysis we can do based on this. Note that verifying if a matrix satisfies the RIP is NP-hard. Therefore, let us take for granted that we have such a matrix and now attempt to prove convergence given this restriction; later in the chapter, we will provide proof that this holds a high probability for general classes of random matrices.

The geometric interpretation of RIP matrices lies in the following two key observations: *i*) one difficulty for a matrix A to satisfy RIP is the fact that A might be adversarially picked such that there is no small constant δ that satisfies these two inequalities; *ii*) More importantly, even if A is "nice" enough, it might be the case that the rows of A, n are so much smaller than the dimension p. In other words, A "squeezes" the information/"energy" in x when one applies Ax, making it hard to guarantee that the energy $\|Ax\|_2$ will be comparable to that of the original $\|x\|_2$ for a small δ. What RIP guarantees is that there might exist matrices A that preserve the "energy" (i.e., distances) of high dimensional vectors $x \in \mathbb{R}^p$, when "projected" onto lower-dimensional subspaces $\mathbb{R}^n$, such that $n \ll p$, when x satisfy some exciting properties (here, sparsity).

Convergence proof of non-convex IHT algorithm. We will assume that $A \in \mathbb{R}^{n\times p}$ satisfies the RIP for some $n \ll p$. To set up the background, we remind that we consider the following problem:

$$\begin{aligned} \underset{x\in\mathbb{R}^p}{\text{minimize}} \quad & f(x) := \tfrac{1}{2}\|y - Ax\|_2^2 \\ \text{subject to} \quad & \|x\|_0 \leq k. \end{aligned}$$

The IHT algorithm solves this problem with the following gradient-based recursion:

$$x_{t+1} = H_k(x_t - \eta \nabla f(x_t)).$$

This is nothing else but *projected gradient descent*, but the projection step is *non-convex*. Thus, any arguments originating from convex analysis breaks (see Chapter 3). Consider the following example: we will try to prove whether the fact

$$\|H_k(x_1) - H_k(x_2)\|_2 \leq \|x_1 - x_2\|_2$$

holds, which is one of the fundamental properties of projections onto convex sets. Here, we prove that this is not true anymore. Consider the following two vectors:

$$x_1 = \begin{bmatrix} 1 \\ 10 \end{bmatrix}, \quad x_2 = \begin{bmatrix} 10 \\ 1 \end{bmatrix}$$

Consider the case of $k = 1$. We could use the analysis of convex projected gradient descent if we could have:

$$\begin{aligned} \|H_1(x_1) - H_1(x_2)\|_2 \leq \|x_1 - x_2\|_2 \Rightarrow \\ \left\| H_1\left(\begin{bmatrix} 1 \\ 10 \end{bmatrix}\right) - H_1\left(\begin{bmatrix} 10 \\ 1 \end{bmatrix}\right) \right\|_2 \leq \left\| \begin{bmatrix} 1 \\ 10 \end{bmatrix} - \begin{bmatrix} 10 \\ 1 \end{bmatrix} \right\|_2 \Rightarrow \\ \left\| \begin{bmatrix} 0 \\ 10 \end{bmatrix} - \begin{bmatrix} 10 \\ 0 \end{bmatrix} \right\|_2 \leq \left\| \begin{bmatrix} 1 \\ 10 \end{bmatrix} - \begin{bmatrix} 10 \\ 1 \end{bmatrix} \right\|_2 \Rightarrow \\ 10\sqrt{2} \leq 9\sqrt{2}, \end{aligned}$$

which is not true; thus, we cannot use this property.

We will start by recalling some relevant details to the proof. For the linear regression problem, the gradient of the function satisfies:

$$\nabla f(x_t) = -A^\top (y - Ax_t).$$

Therefore, the IHT recursion for this particular problem can be simplified into:

$$x_{t+1} = H_k(x_t + \eta A^\top (y - Ax_t))$$

We will assume that we know $k = \|x^\star\|_0$. Also, for the moment, assume $\eta = 1$; this assumption will be broken in other variants of IHT.[13]

But, even if the projection is non-convex, what can we say about our projection? Denote $\tilde{x}_t = x_t + A^\top (y - Ax_t)$. Also, we know that $x_{t+1} = H_k(\tilde{x}_t)$, i.e., x_{t+1} is the best k-sparse projection of $\tilde{x}_t$, based on the ℓ_2-norm distance. With this notation, this implies:

$$\begin{aligned} \|x_{t+1} - \tilde{x}_t\|_2^2 \leq \|x^\star - \tilde{x}_t\|_2^2 \Rightarrow \\ \|(x_{t+1} - x^\star) + (x^\star - \tilde{x}_t)_2^2\| \leq \|x^\star - \tilde{x}_t\|_2^2 \Rightarrow \\ \|x_{t+1} - x^\star\|_2^2 + \|x^\star - \tilde{x}_t\|_2^2 + 2\langle x_{t+1} - x^\star, x^\star - \tilde{x}_t\rangle \leq \|x^\star - \tilde{x}_t\|_2^2 \Rightarrow \\ \|x_{t+1} - x^\star\|_2^2 \leq 2\langle x_{t+1} - x^\star, x^\star - \tilde{x}_t\rangle \end{aligned}$$

Now, we have an expression that includes $\|x_{t+1} - x^\star\|_2^2$ on the left-hand side and an inner product that involves (as we will see) x_t and $x^\star$ on the right-hand side. To proceed, we will define $\mathcal{U} := \text{supp}(x_t) \cup \text{supp}(x_{t+1}) \cup \text{supp}(x^\star)$, where $\text{supp}(\cdot)$ is the support function that, given an argument vector, returns the index set of non-zero elements. In words, the set $\mathcal{U}$ contains the union of the support set of the vectors x_t, x_{t+1}, as well as the optimal set $x^\star$ (we will not use any information of the index set of $x^\star$ in the proof, just the fact that it is a k-sparse set).

Since by definition $y = Ax^\star$ and the fact that $\tilde{x}_t = x_t + A^\top (y - Ax_t)$, we have:

$$\begin{aligned} \tilde{x}_t = x_t + A^\top (y - Ax_t) &= x_t + A^\top (Ax^\star - Ax_t) \\ &= x_t + A^\top A(x^\star - x_t). \end{aligned}$$

[13] As we will see, this step size is valid based on the strict assumption that A satisfies the RIP *with symmetry*. I.e., the RIP inequalities $(1-\delta_k)\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1+\delta_k)\|x\|_2^2$ are centered in the interval $[(1-\delta_k)\|x\|_2^2, (1+\delta_k)\|x\|_2^2]$. However, this symmetry breaks in reality, so step size selection should be completed more carefully.

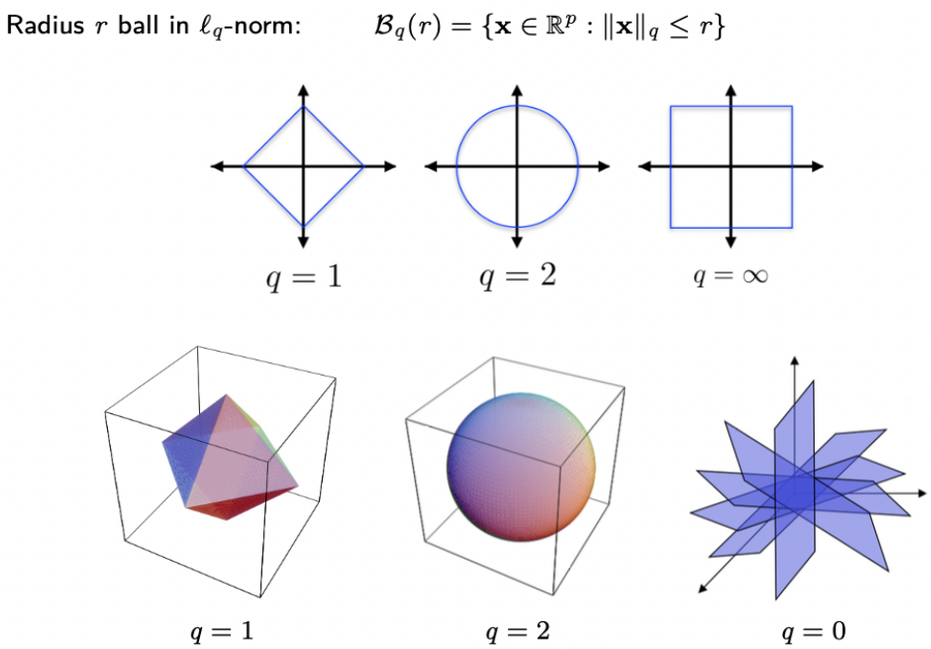


Fig. 47. 2D and 3D representations of some unit norms, both convex and non-convex. The ℓ_0-pseudonorm represents the hyperplanes that span the coordinate system, based on the level of sparsity k.

We will use this definition of $\tilde{x}_t$ in the inequality above. In particular, we have:

$$\begin{aligned}\|x_{t+1} - x^\star\|_2^2 \\ \leq 2\langle x_{t+1} - x^\star, x_t + A^\top A(x^\star - x_t) - x^\star\rangle,\end{aligned}$$

where the RHS equals to:

$$2\langle x_{t+1} - x^\star, (I - A_{\mathcal{U}}^\top A_{\mathcal{U}}) \cdot (x_t - x^\star)\rangle.$$

Here, $A_{\mathcal{U}}$ indicates the matrix A with only columns restricted and indexed by the set $\mathcal{U}$. This selection is based on a key product of the inner product operator to note:

$$\langle x, A^\top y\rangle = x^\top A^\top y = (Ax)^\top y = \langle Ax, y\rangle.$$

I.e., in the quadratic form, the matrix could be "moved" to be applied both on the left and right-hand side of the operator $\langle\cdot,\cdot\rangle$. This means that we can safely restrict the active columns of A on the union of the support set of the vectors $x_{t+1} - x^\star$ and $x_t - x^\star$, which are subsets of the superset $\mathcal{U}$.

For the main term in our recursion, we have:

$$\begin{aligned}\langle x_{t+1} - x^\star, &(I - A_{\mathcal{U}}^\top A_{\mathcal{U}})(x_t - x^\star)\rangle \\ &\leq \|x_{t+1} - x^\star\|_2 \cdot \|(I - A_{\mathcal{U}}^\top A_{\mathcal{U}})(x_t - x^\star)\|_2 \\ &\leq \|x_{t+1} - x^\star\|_2 \cdot \|I - A_{\mathcal{U}}^\top A_{\mathcal{U}}\|_2 \cdot \|x_t - x^\star\|_2\end{aligned}$$

where, again, we use Cauchy-Schwartz inequality, and by using the RIP bounds, we can show that:

$$\|I - A_{\mathcal{U}}^\top A_{\mathcal{U}}\|_2 \leq \max\{(1+\delta_k) - 1, 1 - (1-\delta_k)\} = \delta_k.$$

Using the above in our main expression, we obtain:

$$\begin{aligned}&\|x_{t+1} - x^\star\|_2^2 \leq 2\delta_k \|x_{t+1} - x^\star\|_2 \cdot \|x_t - x^\star\|_2 \implies \\ &\|x_{t+1} - x^\star\|_2 \leq 2\delta_k \|x_t - x^\star\|\end{aligned}$$

Let us define $\rho := 2\delta_k$. One logical expectation for convergence is to assume/require $\rho < 1$, which further assumes $\delta_k \leq \frac{1}{2}$. (*Later on, we will see how the δ_k requirements affect the number of measurements n the matrix A should have to guarantee this convergence, thus the $x^\star$ recovery*).

In what follows, we will unroll our main recursion over t iterations to obtain the following:

$$\begin{aligned}\|x_{t+1} - x^\star\|_2 &\leq \rho \cdot \|x_t - x^\star\|_2 \\ &\leq \rho^t \cdot \|x_0 - x^\star\|_2,\end{aligned}$$

based on $\rho < 1$. To conclude, this implies that we can obtain $\|x_{t+1} - x^\star\|_2 \leq \varepsilon$ by running IHT for $O(\log \frac{\|x_0 - x^\star\|_2}{\varepsilon})$ iterations.

Step size based on convex optimization analysis. In the analysis above, we have used the fact that $\eta = 1$. Yet, this selection does not work well in practice (as we can see in the Demo file of this chapter). This behavior is because, often in practice, the *symmetry* in the RIP condition is not always satisfied. I.e., bounds:

$$(1 - \delta_k)\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1 + \delta_k)\|x\|_2^2,$$

are not "centered". More specifically, there might be some lower and upper bound constants, μ_k and L_k[14], such that we still have:

$$\mu_k\|x\|_2^2 \leq \|Ax\|_2^2 \leq L_k\|x\|_2^2,$$

Can we pick a new step size based on the RIP property?

It is not hard to show that these lower and upper bound constants are the *minimum and maximum* eigenvalues of the Hessian matrix when one is restricted to sparse signals. I.e., one can use (μ_k, L_k), if known, to apply step size selection techniques, like the ones we used in convex optimization. E.g.,

[14] The selection for this notation is on purpose.

- In Convex Optimization, $\eta = \frac{1}{L}$ works well (where L is the Lipschitz constant of the objective function). L is also the upper bound on the eigenvalues of the Hessian of the function.
- In our case, we have L_k (but assumed known for now). In the symmetric version of RIP, $L_k = (1 + \delta_k)$.

Let us drive a deeper connection between the above notions. By definition of $f(\cdot)$, for x_1, x_2 that are k-sparse, and using the definition of the L-Lipschitzness, we have:

$$\begin{aligned}\|\nabla f(x_1) - \nabla f(x_2)\|_2 &= \|-A^\top(y - Ax_1) + A^\top(y - Ax_2)\|_2 \\ &= \|A^\top A(x_1 - x_2)\|_2 \\ &\leq \max_{\mathcal{S}:|\mathcal{S}|\leq 2k} \|(A^\top A)_{\mathcal{S}}\|_2 \cdot \|x_1 - x_2\|_2 \\ &\leq (1 + \delta_{2k})\|x_1 - x_2\|_2\end{aligned}$$

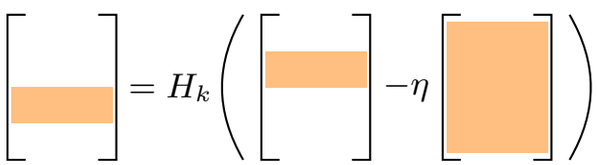
by definition of RIP on $2k$-sparse vectors. This drives the connection that, similarly to convex optimization that one uses $\eta = \frac{1}{L}$, one could potentially use $\eta = \frac{1}{1+\delta_{2k}}$ as a step size. Yet, the difficulty of this choice is that δ-values are NP-hard to know a priori. So, a better strategy should be devised.

Adaptive Step Sizes. To close the IHT section, we will consider adaptive step sizes. We want to consider whether there are efficient adaptive step size selection formulas η_t in $x_{t+1} = H_k(x_t - \eta_t \cdot \nabla f(x_t))$.

To do so, let us start with some observations:

- x_t is k-sparse;
- x_{t+1} is k-sparse;
- x_{t+1} could potentially have intersection with the support set of x_t, as well as the set $H_k(-\nabla f(x_t))$ (outside of $\text{supp}(x_t)$).

Schematically, the above observations lead to the following picture for the IHT recursion:



We will present the idea of *line search*. This is the case where choosing step size is the result of an optimization problem, as in:

$$\eta := \underset{\eta}{\text{argmin}} \|y - A(x_t - \eta \nabla f(x_t)\|_2$$

As we will show in the Demo, such approaches perform better in practice than any constant step size selection that theory might suggest. The key attribute for line search approaches is for η to be efficiently computable.

To complete the above, let us define first:

$$\begin{aligned}\mathcal{S}_t &= \text{supp}(x_t) \\ \mathcal{Q}_t &= \mathcal{S}_t \cup \text{supp}(H_k(\nabla_{\mathcal{S}_t^c} f(x_t))) \\ \mathcal{S}_{t+1} &= \text{supp}(x_{t+1}) \subseteq \mathcal{Q}_t,\end{aligned}$$

where $\mathcal{S}^c$ represent the complement of a set. Then, based on the scheme above, observe that:

$$H_k(x_t - \eta \nabla f(x_t)) = H_k(x_t - \eta \cdot \nabla_{\mathcal{Q}_t} f(x_t));$$

i.e., what matters in the gradient $\nabla f(x_t)$ is indexed by the set $\mathcal{Q}_t$. This observation changes the line search problem above:

$$\eta = \underset{\eta}{\text{argmin}} \|y - A(x_t - \eta \cdot \nabla_{\mathcal{Q}_t} f(x_t))\|_2^2.$$

But what is the solution to this 1D problem with respect to η? Define the auxiliary objective $g(\eta) := \|y - A(x_t - \eta \cdot \nabla_{\mathcal{Q}_t} f(x_t))\|_2^2$. Taking the derivative and setting it equal to zero:

$$\begin{aligned}0 &= \nabla g(\eta) \\ &= 2\langle A\nabla_{\mathcal{Q}_t} f(x_t), y - Ax_t\rangle + 2\eta\|A\nabla_{\mathcal{Q}_t} f(x_t)\|_2^2 \\ \implies \eta &= \frac{-\langle A\nabla_{\mathcal{Q}_t} f(x_t), y - Ax_t\rangle}{\|A\nabla_{\mathcal{Q}_t} f(x_t)\|_2^2} = \frac{\|\nabla_{\mathcal{Q}_t} f(x_t)\|_2^2}{\|A\nabla_{\mathcal{Q}_t} f(x_t)\|_2^2}\end{aligned}$$

Can we relate η to the RIP? We know that in the original definition, the following holds:

$$1 - \delta \leq \frac{\|Ax\|_2^2}{\|x\|_2^2} \leq 1 + \delta,$$

for all sparse vectors x. In our case above, $\nabla_{\mathcal{Q}_t} f(x_t)$ is still a sparse vector. How much sparse? $2k$-sparse! Thus, the term $\frac{\|\nabla_{\mathcal{Q}_t} f(x_t)\|_2^2}{\|A\nabla_{\mathcal{Q}_t} f(x_t)\|_2^2}$ has A applying on the sparse gradient vector, which further leads to (based on the RIP bounds:

$$1 + \delta \leq \eta \leq \frac{1}{1 - \delta}.$$

But is this computed efficiently? Well, it turns out that $\eta_t = \frac{\|\nabla_{\mathcal{Q}_t} f(x_t)\|_2^2}{\|A\nabla_{\mathcal{Q}_t} f(x_t)\|_2^2}$. Here, the gradient vector is already computed per iteration; what we only need to compute is the set $\mathcal{Q}_t$, which depends on sorting the elements of the dense gradient vector and selecting the k-sparse best subset out of the $\mathcal{S}_t$ set. Finally, applying the operation $A\nabla_{\mathcal{Q}_t} f(x_t)$ does not add much to the overall complexity of the algorithm. Thus, computing η_t is efficient! And it comes with nice theoretical properties that we can use!

Proof of Adaptive Step Sizes in IHT. Following the same procedure as in $\eta = 1$, we have[15]

$$\|x_{t+1} - x^\star\|_2 \leq 2\|I - \eta A_{\mathcal{U}}^\top A_{\mathcal{U}}\|_2 \cdot \|x_t - x^\star\|_2$$

By RIP, along with η inclusion in the equations, we get:

$$\begin{aligned}\|I - \eta A_{\mathcal{U}}^\top A_{\mathcal{U}}\|_2 &\leq \max\{\eta(1 + \delta) - 1, 1 - \eta(1 - \delta)\} \\ &\leq \max\left\{\tfrac{1+\delta}{1-\delta} - 1, 1 - \tfrac{1-\delta}{1+\delta}\right\} \\ &\leq \tfrac{2\delta}{1-\delta},\end{aligned}$$

where the last inequality we use the property $1 + \delta \leq \eta \leq \frac{1}{1-\delta}$ of the step size. Then, going back to the original expression:

$$\begin{aligned}\|x_{t+1} - x^\star\|_2 &\leq 2\tfrac{2\delta}{1-\delta} \cdot \|x_t - x^\star\|_2 \\ &= \tfrac{4\delta}{1-\delta}\|x_t - x^\star\|_2.\end{aligned}$$

Assuming:

$$\delta < \tfrac{1}{5},$$

we get

$$\tfrac{4\delta}{1-\delta} =: \rho < 1.$$

We get convergence as shown in the proof of convergence of regular IHT (in which $\rho < 1$).

———— ∞ ————

[15] We drop the dependence on k in δ_k for ease of exposition.

Graphical Model Selection.

The Story

Graphs are increasingly becoming a common problem in massive datasets. The simplest example is a network of friends: You are friends with some people, and they have friends until you have an interconnected web of individuals and relationships. There is a dependent relationship between the connected nodes. Therefore, it may be helpful to determine who is connected vs who isn't to provide recommendations or create new connections.

To bring things back to our realm, there exists the problem of *covariance selection*: In a graph representation of the random variable X, nodes are the components of X (i.e. X_i's), and edges exist when there any two X_i and X_j are conditionally dependent; This is also known as a *normal (Gaussian) graphical model* of the random variable [69]. Further reading can be found in [69]. For now, we will look at the concept of conditional independence.

Conditional Independence [69]

Let x, y, z be random variables with continuous distributions. We say *x and y are conditionally independent given z* if...

$$f(x|y,z) = f(x|z)$$

This means y has nothing to provide information about x given that we know about z. For graphical selection, one will work with random *vectors*, so determining the independence of components within a random vector becomes necessary. Find whether two coefficients are independent given all the other coefficients (notice the absence of x_j in the RHS)...

$$f(x_i|x_1, x_2, ..., x_{i-1}, x_{i+1}, ..., x_n) = f(x_i|x_1, x_2, ..., x_{i-1}, x_{i+1}, ..., x_{j-1}, x_{j+1}, ..., x_n)$$

For Normal variables specifically, this becomes trivial with the covariance matrix. Suppose $X \sim N(\mu, \epsilon)$. We know the conditional distribution of (x_i, x_j) given the other components of X is Gaussian, and has covariance matrix

$$\begin{bmatrix} (\Sigma^{-}1)_{ii} & (\Sigma^{-}1)_{ij} \\ (\Sigma^{-}1)_{ji} & (\Sigma^{-}1)_{jj} \end{bmatrix}.$$

We can claim conditional independence for x_i and x_j if their covariance matrix is diagonal if $(\Sigma^{-}1)_{ij} = 0$.

Deep Dive

Now let us go deeper into normal distributions

Let $x \sim \mathcal{N}(\mu, \Sigma)$. Then its probability density satisfies:

$$f(x) = \frac{1}{(2\pi)^{p/2}\det(\Sigma)^{1/2}} \cdot \exp\left\{-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right\}.$$

Define $\Theta = \Sigma^{-1}$ as the inverse covariance matrix or precision matrix. Then:

$$f(x) = \frac{\det(\Theta)^{1/2}}{(2\pi)^{p/2}} \cdot \exp\left\{-\frac{1}{2}(x-\mu)^\top \cdot \Theta \cdot (x-\mu)\right\}.$$

We now introduce the problem definition: assume we do not know (μ, Σ), but we have samples $\{x_i\}_{i=1}^n$, $x_i \sim \mathcal{N}(\mu, \Sigma)$. Let's see what we can do with these samples. Assume independence between the x_i's. The log-likelihood function is:

$$\begin{aligned} l(\mu, \theta) &= \sum_{i=1}^n \log f(x_i) \\ &\propto \sum_{i=1}^n \log\det(\Theta)^{1/2} - \sum_{i=1}^n \frac{1}{2}(x_i-\mu)^\top \Theta (x_i-\mu) \\ &= \frac{n}{2}\log\det(\Theta) - \frac{1}{2}\sum_{i=1}^n (x_i-\mu)^\top \cdot \Theta \cdot (x_i-\mu) \end{aligned}$$

Observe that:

$$\begin{aligned} &-\operatorname{tr}(\Theta \cdot \hat{\Sigma}) - (\mu-\hat{\mu})^\top \Theta(\mu-\hat{\mu}) \\ &= -\operatorname{tr}\left(\Theta \cdot \frac{1}{n}\sum_{i=1}^n \left(x_i - \frac{1}{n}\sum_{i=1}^n x_i\right)\left(x_i - \frac{1}{n}\sum_{i=1}^n x_i\right)^\top\right) \\ &\quad - \left(\mu - \frac{1}{n}\sum_{i=1}^n x_i\right)^\top \Theta \left(\mu - \frac{1}{n}\sum_{i=1}^n x_i\right), \end{aligned}$$

where we used $\hat{\mu} = \frac{1}{n}\sum_{i=1}^n x_i$ and $\hat{\Sigma} = \frac{1}{n}\sum_{i=1}^n (x_i-\hat{\mu})(x_i-\hat{\mu})^\top$. Working further on this expression, we get the following:

$$\begin{aligned} &-\operatorname{tr}(\Theta \cdot \hat{\Sigma}) - (\mu-\hat{\mu})^\top \Theta(\mu-\hat{\mu}) \\ &\quad = -\frac{1}{n}\sum_{i=1}^n (x_i - \frac{1}{n}\sum_{i=1}^n x_i)^\top \Theta (x_i - \frac{1}{n}\sum_{i=1}^n x_i) \\ &\qquad - (\mu - \frac{1}{n}\sum_{i=1}^n x_i)^\top \Theta (\mu - \frac{1}{n}\sum_{i=1}^n x_i) \\ &\quad = -\frac{1}{n}\sum_{i=1}^n (x_i-\mu)^\top \Theta (x_i-\mu). \end{aligned}$$

Thus our $l(\cdot,\cdot)$ transforms into:

$$l(\mu, \Theta) = \frac{n}{2}\left(\log\det(\Theta) - \operatorname{tr}(\Theta \cdot \hat{\Sigma}) - (\mu-\hat{\mu})^\top \Theta(\mu-\hat{\mu})\right)$$

Maximum likelihood estimation of (μ, Σ) leads to:

$$\min_{\mu, \theta \succ 0} -\log\det(\Theta) + \operatorname{tr}(\Theta \cdot \hat{\Sigma}) + (\mu-\hat{\mu})^\top \Theta(\mu-\hat{\mu}).$$

Only the last term in the above expression contains μ; and since $\Theta \succ 0$, $\mu^* = \hat{\mu}$. So letting $\mu^* = \hat{\mu}$, we find:

$$\min_{\substack{\Theta \succ 0 \\ \Theta \in \mathbb{R}^{p\times p}}} -\log\det(\Theta) + \operatorname{tr}(\Theta \cdot \hat{\Sigma}) = -\log\det(\Theta) + \langle \Theta, \hat{\Sigma} \rangle$$

As a side note, the determinant of a squared matrix is (relatively) a challenging object/operation to describe. The geometric way of thinking of it is if we had a unit cube in p dimensions, then $\det(\Theta)$ measures the volume of the cube after applying the rows/columns of Θ on that cube. Another way to see it is:

$$\det(\Theta) = \prod_{i=1}^p \lambda_i(\Theta),$$

where $\lambda_i(\Theta)$ is the i-th eigenvalue of Θ.

Why do we care about all this? A very nice theory connects undirected graphs under Gaussian assumptions and covariance selection. This theory assumes that variables $x(i), x(j)$ from $x \sim \mathcal{N}(\mu, \Sigma)$ are conditionally independent if and only if $\Theta^*_{ij} = 0$. You can see the example drawn out in the slides.

Concretely, we can ask the question: given samples $\{x_i\}_{i=1}^n$, can we infer the underlying undirected graph structure?

Answer #1: We can take many samples and use them to compute $\hat{\mu}$, $\hat{\Sigma}$. Then we can derive $\hat{\Sigma}^{-1}$. But if p is on the order of 10^5 to 10^6, this is often impossible.

Answer #2: We find the most important part of the graph. Assuming sparsity in Σ^{-1}, we find $\Theta = \Sigma^{-1}$ satisfying:

$$\begin{aligned} \underset{\Theta \succ 0}{\text{minimize}} \quad & -\log\det(\Theta) - \text{tr}(\Theta \cdot \hat{\Sigma}) \\ \text{subject to} \quad & \|\Theta\|_0 \leq k. \end{aligned}$$

Note that $-\log\det(\Theta) + \text{tr}(\Theta \cdot \hat{\Sigma})$ is locally Lipschitz gradient. (More to be added in future versions of this note).

RIP proof for sub-Gaussian matrices. Matrices that satisfy:

$$\mathbb{P}_{A \sim D^{n \times p}}[\|\|A_x\|_2^2 - \|x\|_2^2] > \epsilon \cdot \|x\|_2^2 (\leq 2e^{-\Omega(n)}),$$

will also satisfy the RIP property with probability $1 - 2e^{-\Omega(n)}$, whenever $n \geq \Omega(\frac{k}{\delta^2} \log \frac{p}{k})$. So, this hints at a way to get RIP matrices (which, as we mentioned before, were computationally expensive to verify). Gaussian and Bernoulli matrices $A \in \mathbb{R}^{n \times p}$ will satisfy the above property, making good candidates.

Below, we use the following definitions: a random variable x is called Sub-Gaussian if there exists $\beta, k > 0$ such that:

$$\mathbb{P}(|x| \geq t) \leq \beta e^{-kt^2}, \forall t > 0$$

In general, x is called Sub-Exponential if there exist $\beta, k > 0$ such that

$$\mathbb{P}(|x| \geq t) \leq \beta \cdot e^{-kt}, \forall t > 0$$

Finally, a vector $y \in \mathbb{R}^p$ is called isotropic if $\mathbb{E}[|\langle y, x\rangle|^2] = \|x\|_2^2, \forall x \in \mathbb{R}^p$.

Step 1: *Let $A \in \mathbb{R}^{n \times p}$ with independent, isotropic, and Sub-Gaussian rows. Then, $\forall x \in \mathbb{R}^p$ and $\forall t \in (0,1)$:*

$$\mathbb{P}\left(\left|\frac{1}{n}\|AX\|_2^2 - \|x\|_2^2\right| \geq t \cdot \|x\|_2^2\right) \leq 2e^{-ct^2 n}, c \text{ constant}$$

Proof: W.L.O.G., $\|x\|_2 = 1$. Let $\alpha_1, \alpha_2, \ldots, \alpha_n \in \mathbb{R}^p$ be the rows of A. Define $z_i = |\langle \alpha_i, x\rangle|^2 - \|x\|_2^2$. Since α_i is isotropic, $\mathbb{E}[z_i] = 0$. Further, z_i is Sub-Exponential, since $\langle \alpha_i, x\rangle$ is Sub-Gaussian; this means

$$\mathbb{P}(|z_i| \geq r) \leq \beta e^{-kr}, \forall r > 0$$

Observe:

$$\frac{1}{n}\|Ax\|_2^2 - \|x\|_2^2 = \frac{1}{n}\sum_{i=1}^n (|\langle \alpha_i, x\rangle|^2 - \|x\|_2^2) = \frac{1}{2}\sum_{i=1}^n z_i$$

Since the α_i's are independent, the z_i's are independent. We will now use Berstein inequality: Let $x_1, \ldots, X_M$ be independent, zero-mean, Sub-Exponential random variables, with constants β, k. Then:

$$\mathbb{P}\left(\left|\sum_{i=1}^M x_i\right| \geq t\right) \leq 2e^{-\frac{(kt)^2/2}{2\beta M + kt}}$$

In our case, this translates into

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^n z_i\right| \geq t\right) = \mathbb{P}\left(\left|\sum_{i=1}^n z_i\right| \geq tn\right) \leq 2e^{-\frac{k^2 n^2 t^2/2}{2\beta n + knt}}$$

$$\leq 2e^{-\frac{k^2}{4\beta + 2k} \cdot nt^2} \quad \text{for } t \in (0,1)$$

□

Step 2: *Assume Step 1 holds. Fix a set $S \subset [p]$ with $|S| = k$ and δ, $\xi \in (0,1)$. If*

$$n \geq \frac{c}{\delta^2}\left(7k + 2\ln\left(\frac{2}{\xi}\right)\right), c \text{ constant}$$

Then W.P. at $1 - \xi$:

$$\|A_s^\top A_s - I\|_2 < \delta$$

Proof: We will use the construction of ϵ-nets over unit balls. Let $B = \{x \in \mathbb{R}^p, \|x\|_2 \leq 1\}$. An ϵ-net over B is a set such that, for every point in B, there is a point in the ϵ-net that is ϵ-close by some distance function (e.g., $\|x - y\|_2 \leq \epsilon$). The number of points in such an ϵ-net can be bounded by:

$$\mathcal{N}(B, \|\cdot\|_2, \epsilon) \leq \left(1 + \frac{2}{\epsilon}\right)^p$$

In our case, we generate an ϵ-net on $B = \{x \in \mathbb{R}^p, \text{supp}(x) \subset S, \|x\|_2 \leq 1\}$. In this case:

$$\mathcal{N}(B, \|\cdot\|_2, \epsilon) \leq \left(1 + \frac{2}{\epsilon}\right)^k$$

Then, from Step 1:

$$\begin{aligned} &\mathbb{P}\left(|\|Au\|_2^2 - \|u\|_2^2| \geq t \cdot \|u\|_2^2, \text{for some } u \text{ in } \epsilon\text{-net}\right) \\ &\leq \sum_{u \text{ in } \epsilon\text{-net}} \mathbb{P}\left(|\|Au\|_2^2 - \|u\|_2^2| \geq t \cdot \|u\|_2^2\right) \\ &\leq 2 \cdot \left(1 + \frac{2}{\epsilon}\right)^k e^{-ct^2 n} \end{aligned}$$

Define $D = A_s^\top A_s - I$. Then:

$$\begin{aligned} \left|\|Au\|_2^2 - \|u\|_2\right| &= \left|\langle A_s^\top A_s u, u\rangle - \langle u, u\rangle\right| \\ &= \left|\langle (A_s^\top A_s - I)u, u\rangle\right| \\ &= |\langle Du, u\rangle| \end{aligned}$$

Then, our goal is to prove $|\langle Dx, x\rangle| < t$ (for $x \in B$, and proper t) via $|\langle Du, u\rangle| < t$ where u is in the ϵ-net.

Assume $|\langle Du, u\rangle| < t$. This occurs W.P. $1 - 2\left(1 + \frac{2}{\epsilon}\right)^k e^{-ct^2 n}$. Then, for some $x \in B$, and some u in ϵ-net such that $\|x - u\|_2 \leq \epsilon < \frac{1}{2}$, we get:

$$\begin{aligned} |\langle Du, u\rangle| &= |\langle Du, u\rangle + \langle D(x+u), x-u\rangle| \\ &\leq |\langle Du, u\rangle| + |\langle D(x+u), x-u\rangle| \\ &\leq t + \|D\|_2 \cdot \|x+u\|_2 \cdot \|x-u\|_2 \leq t + 2 \cdot \|D\|_2 \cdot \epsilon \end{aligned}$$

Taking the maximum over $x \in B$:

$$\|D\|_2 < t + 2\|D\|_2 \cdot \epsilon \implies \|D\|_2 \leq \frac{t}{1 - 2\epsilon}$$

Choose $t = (1 - 2\epsilon) \cdot \delta \to \|D\|_2 < \delta$. This means:

$$\mathbb{P}\left(\|A_s^\top A_s - I\|_2 \geq \delta\right) \leq 2\left(1 + \frac{2}{\epsilon}\right)^k e^{-c(1-2\epsilon)^2 \delta^2 n}$$

Choosing $\epsilon = \frac{2}{e^{7/2} - 1}$, we get that $\|A_s^\top A_s - I\|_2 \leq \delta$ with probability $1 - \xi$ provided

$$n \geq \frac{c}{\delta^2}\left(7k + 2\ln\left(\frac{2}{\xi}\right)\right)$$

□

Step 3: *We proved that $\|A_s^\top A_s - I\|_2 < \delta$ for a single s. Taking all $\binom{p}{k}$ subsets $S \subset [p]$ with $|S| = k$, we get:*

$$\begin{aligned}\mathbb{P}\left(\sup_S \|A_s^\top A_s - I\|_2 \geq \delta\right) &\leq \sum_s \mathbb{P}(\|A_s^\top A_s - I\|_2 \geq \delta) \\ &\leq 2\binom{p}{k}\left(1+\frac{2}{\epsilon}\right)^k \cdot e^{-c(1-2\epsilon)^2\delta^2 n} \\ &\leq 2\binom{ep}{k}^k\left(1+\frac{2}{\epsilon}\right)^k e^{-c(1-2\epsilon)^2\delta^2 n}\end{aligned}$$

Forcing this probability to be less than ξ, we get

$$n \geq O\left(k\ln\left(\frac{ep}{k}\right) + \frac{14}{3}k + \frac{4}{3}\ln\left(\frac{2}{\xi}\right)\right)$$

□

———— ∞ ————

Practical Applications: Signal Recovery

One of the most common use cases is extracting information from a signal, where the data in the signal is embedded in noise. Typically, this is done by directly reading the raw data and attempting to remove the noise from it after the fact, but this can be very inefficient and may lead to data loss. From this, we derive the idea of *compressive sampling* - "for certain types of signals, a small number of nonadaptive samples carries sufficient information to approximate the signal well" [59].

To solve this problem, one might represent the sparse input as a convex optimization problem, where the result of minimizing against some program results in the approximation of the target signal, but this can be computationally intense. This is where solutions such as *CoSaMP* algorithm, proposed by Needell and Tropp [59], can be used, which takes inspiration from restricted isometry in k-sparse signals to estimate the target signal.

The Basics. Consider the signal reconstruction problem. As with any sparse model problem, utilizing the most significant components from the target signal has the best chance of getting the best estimate of the underlying data. Suppose the sampling matrix Φ has a significantly small isometry constant. We can then define $\mathbf{y} = \Phi * \Phi\mathbf{x}$ where x is our k-sparse signal, and y is a "proxy" that provides a mapping to our information. The k most significant entries in y map to the k most significant entries in x, so we can obtain this "proxy" by applying $\Phi*$ to x. Doing this iteratively, we're able to approximate the target signal. Formally, CoSAMP is defined as follows:

Definition 32. (CoSAMP Algorithm [59]) *Suppose that Φ is an $m \times N$ sampling matrix with restricted isometry constant $\delta_{2s} \leq c$. Let $\boldsymbol{u} = \Phi\boldsymbol{x} + \boldsymbol{e}$ be a vector of samples of an arbitrary signal contaminated with arbitrary noise. For a given precision parameter η, The* CoSAMP *algorithm produces a $2k$-sparse approximation $\boldsymbol{a}$ that satisfies*

$$\|\boldsymbol{x} - \boldsymbol{a}\|_2 \leq C \cdot \max\left\{\eta\,\frac{1}{\sqrt{k}}\|\boldsymbol{x} - \boldsymbol{x}_k\|_1 + \|\boldsymbol{e}\|_2\right\}$$

where $\boldsymbol{x}_k$ is a best k-sparse approximation to $\boldsymbol{x}$. The running time is $O\left(\mathcal{L}\cdot\log\left(\|x\|_2/\eta\right)\right)$, where $\mathcal{L}$ bounds the cost of a matrix-vector multiply with Φ or $\Phi*$. The working storage use is $O(N)$.

Algorithm [59]. CoSAMP relies on access to the following: we must have access to a vector with the noisy samples that contain the underlying signal x and the sampling operator Φ, information on the sparsity of the approximation k, and some halting criteria.

With this information, The algorithm first creates the proxy $\mathbf{y} = \Phi * \Phi\mathbf{x}$ and, based on residuals from current sample data, obtains the most significant components of the proxy. These components are then combined with the elements of the current approximation of the signal, where least squares are used to estimate the target signal. We keep only the largest (k) entries from this estimation and loop back around with the remaining residual data to continue estimation. This is done repeatedly until the halting criteria are reached.

For this to work, there are a few assumptions which need to be made that are typical in compressive sampling:

- The sparsity k is fixed
- The sampling operator Φ is $m \times N$ and has restricted isometry $\delta \leq 0.1$
- The sample vector is $\mathbf{u} = \Phi\mathbf{x} + \mathbf{e}$
- The input signal is arbitrary $\mathbf{x} \in \mathbb{C}^N$
- The noise vector is arbitrary $e \in \mathbb{C}^m$

In addition, there is certain *unrecoverable energy* (denoted as ν), essentially an error resulting from the sample input being non-sparse or containing significant noise. CoSAMP works best when this value is high, as seen below:

Theorem 9. (CoSAMP Iteration [59]) *For each iteration $t \geq 0$, the signal approximation a^t is k-sparse and...*

$$\left\|x - a^{t+1}\right\|_2 \leq 0.5\left\|x - a^t\right\|_2 + 10\nu$$

In particular...

$$\left\|x - a^t\right\|_2 \leq 2^{-t}\|x\|_2 + 20\nu$$

Proof can be found in [59].

Regarding measuring the quality of the reconstructed signal, the *signal-to-noise (SNR)* is typically used in communications. An analogous definition for the SNR of a sparse reconstruction can be established as follows...

$$\text{R-SNR} = 10\log_{10}\left(\frac{\|x\|_2}{\nu}\right)\ \text{dB}$$

...where the SNR is the ratio between the currently reconstructed signal and the unrecoverable energy...

$$\text{SNR} = 10\log_{10}\left(\frac{\|x\|_2}{\nu}\right)$$

Using the iteration definition, we can establish the following SNR ceiling on the k-th iteration:

$$\text{R-SNR} \lesssim 3 - \min\{3k, \text{SNR} - 13\}$$

In other words, CoSAMP can reduce the SNR by 3 dB until it reaches the noise floor. Many iterations may be needed to get a high enough SNR (or the SNR starts extremely small). As it turns out, this depends on how precise the arithmetic behind CoSAMP is: Should high precision be available, CoSAMP can return a high SNR result in a fixed number of iterations:

Theorem 10. (CoSAMP Iteration Count [59]) *If CoSAMP is implemented with high arithmetic precision, then after at most*

$6(t+1)$ *iterations, CoSAMP produces a k-sparse approximation a that satisfies*

$$\|x - a\|_2 \leq 20\nu$$

Proof can be found in Appendix B of [59]

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. Yann LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In Neural networks: Tricks of the trade, pages 9–50. Springer, 2002.
16. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE international conference on computer vision, pages 1026–1034, 2015.
17. Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. Advances in neural information processing systems, 31, 2018.
18. 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.
19. 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.
20. 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.
21. 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.
22. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.
23. 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.
24. 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.
25. 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.
26. 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.
27. 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.
28. 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.
29. 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.
30. 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.
31. 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.
32. Gary Marcus, Ernest Davis, and Scott Aaronson. A very preliminary analysis of DALL-E 2. arXiv preprint arXiv:2204.13807, 2022.
33. 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.
34. 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.
35. Or Sharir, Barak Peleg, and Yoav Shoham. The cost of training nlp models: A concise overview. arXiv preprint arXiv:2004.08900, 2020.
36. 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.
37. Philip Wolfe. Convergence conditions for ascent methods. SIAM review, 11(2):226–235, 1969.
38. Larry Armijo. Minimization of functions having Lipschitz continuous first partial derivatives. Pacific Journal of mathematics, 16(1):1–3, 1966.
39. Stephen Wright and Jorge Nocedal. Numerical optimization. Springer Science, 35(67-68):7, 1999.
40. B. Polyak. Introduction to optimization. Inc., Publications Division, New York, 1, 1987.
41. Stephen Boyd, Lin Xiao, and Almir Mutapcic. Subgradient methods. lecture notes of EE392o, Stanford University, Autumn Quarter, 2004:2004–2005, 2003.
42. Marguerite Frank, Philip Wolfe, et al. An algorithm for quadratic programming. Naval research logistics quarterly, 3(1-2):95–110, 1956.
43. 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.
44. 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.
45. Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. Computer, (8):30–37, 2009.
46. A. Mnih and R. Salakhutdinov. Probabilistic matrix factorization. In Advances in neural information processing systems, pages 1257–1264, 2008.
47. 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.
48. S. Zavriev and F. Kostyuk. Heavy-ball method in nonconvex optimization problems. Computational Mathematics and Modeling, 4(4):336–341, 1993.
49. 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.
50. 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.
51. B. O'Donoghue and E. Candes. Adaptive restart for accelerated gradient schemes. Foundations of computational mathematics, 15(3):715–732, 2015.
52. 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.
53. L. Bottou, F. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. Siam Review, 60(2):223–311, 2018.
54. S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. SIAM review, 43(1):129–159, 2001.
55. R. Tibshirani. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1):267–288, 1996.
56. P. Hoff. Lasso, fractional norm and structured sparse estimation using a Hadamard product parametrization. Computational Statistics & Data Analysis, 115:186–198, 2017.
57. 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.
58. T. Blumensath and M. Davies. Iterative hard thresholding for compressed sensing. Applied and computational harmonic analysis, 27(3):265–274, 2009.
59. D. Needell and J. Tropp. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. Applied and computational harmonic analysis, 26(3):301–321, 2009.
60. S. Foucart. Hard thresholding pursuit: an algorithm for compressive sensing. SIAM Journal on Numerical Analysis, 49(6):2543–2563, 2011.
61. J. Tanner and K. Wei. Normalized iterative hard thresholding for matrix completion. SIAM Journal on Scientific Computing, 35(5):S104–S125, 2013.
62. K. Wei. Fast iterative hard thresholding for compressed sensing. IEEE Signal processing letters, 22(5):593–597, 2014.
63. 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.
64. Jeffrey D Blanchard and Jared Tanner. GPU accelerated greedy algorithms for compressed sensing. Mathematical Programming Computation, 5(3):267–304, 2013.
65. 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.
66. 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.

67. 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.
68. 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.
69. Joachim Dahl, Lieven Vandenberghe, and Vwani Roychowdhury. Covariance selection for nonchordal graphs via chordal embedding. Optimization Methods & Software, 23(4):501–520, 2008.
70. Joseph B Altepeter, Daniel FV James, and Paul G Kwiat. 4 qubit quantum state tomography. In Quantum state estimation, pages 113–145. Springer, 2004.
71. 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.
72. Masoud Mohseni, AT Rezakhani, and DA Lidar. Quantum-process tomography: Resource analysis of different strategies. Physical Review A, 77(3):032322, 2008.
73. 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.
74. Y.-K. Liu. Universal low-rank matrix recovery from Pauli measurements. In Advances in Neural Information Processing Systems, pages 1638–1646, 2011.
75. 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.
76. Miroslav Ježek, Jaromír Fiurášek, and Zdeněk Hradil. Quantum inference of states and processes. Physical Review A, 68(1):012305, 2003.
77. Konrad Banaszek, Marcus Cramer, and David Gross. Focus on quantum tomography. New Journal of Physics, 15(12):125020, 2013.
78. 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.
79. 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.
80. 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.
81. Giacomo Torlai and Roger Melko. Machine-learning quantum states in the NISQ era. Annual Review of Condensed Matter Physics, 11, 2019.
82. 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.
83. 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.
84. 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.
85. E. Bolduc, G. Knee, E. Gauger, and J. Leach. Projected gradient descent algorithms for quantum state tomography. npj Quantum Information, 3(1):44, 2017.
86. Jiangwei Shang, Zhengyun Zhang, and Hui Khoon Ng. Superfast maximum-likelihood reconstruction for quantum tomography. Phys. Rev. A, 95:062336, Jun 2017.
87. 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.
88. 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.
89. 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.
90. Martin Kliesch, Richard Kueng, Jens Eisert, and David Gross. Guaranteed recovery of quantum processes from few measurements. Quantum, 3:171, 2019.
91. 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.
92. 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.
93. 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.
94. N. Srebro, J. Rennie, and T. Jaakkola. Maximum-margin matrix factorization. In Advances in neural information processing systems, pages 1329–1336, 2004.
95. 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.
96. 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.
97. J. Bennett and S. Lanning. The Netflix prize. In Proceedings of KDD cup and workshop, volume 2007, page 35, 2007.
98. 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.
99. R. Keshavan. Efficient algorithms for collaborative filtering. PhD thesis, Stanford University, 2012.
100. 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.
101. 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.
102. 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.
103. A. Makadia, V. Pavlovic, and S. Kumar. A new baseline for image annotation. In Computer Vision–ECCV 2008, pages 316–329. Springer, 2008.
104. 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.
105. J. Weston, S. Bengio, and N. Usunier. WSABIE: Scaling up to large vocabulary image annotation. In IJCAI, volume 11, pages 2764–2770, 2011.
106. Andrew I. Schein, Lawrence K. Saul, and Lyle H. Ungar. A generalized linear model for principal component analysis of binary data. In AISTATS, 2003.
107. 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.
108. C. Johnson. Logistic matrix factorization for implicit feedback data. Advances in Neural Information Processing Systems, 27, 2014.
109. Koen Verstrepen. Collaborative Filtering with Binary, Positive-only Data. PhD thesis, University of Antwerpen, 2015.
110. N. Gupta and S. Singh. Collectively embedding multi-relational data for predicting user preferences. arXiv preprint arXiv:1504.06165, 2015.
111. 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.
112. 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.
113. E. Candes, Y. Eldar, T. Strohmer, and V. Voroninski. Phase retrieval via matrix completion. SIAM Review, 57(2):225–251, 2015.
114. I. Waldspurger, A. d'Aspremont, and S. Mallat. Phase recovery, MaxCut and complex semidefinite programming. Mathematical Programming, 149(1-2):47–81, 2015.
115. 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.
116. 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.
117. 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.
118. H. Andrews and C. Patterson III. Singular value decomposition (SVD) image coding. Communications, IEEE Transactions on, 24(4):425–432, 1976.
119. 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.
120. E. Candès and B. Recht. Exact matrix completion via convex optimization. Foundations of Computational mathematics, 9(6):717–772, 2009.
121. 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.
122. S. Becker, V. Cevher, and A. Kyrillidis. Randomized low-memory singular value projection. In 10th International Conference on Sampling Theory and Applications (Sampta), 2013.
123. 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.
124. K. Lee and Y. Bresler. ADMiRA: Atomic decomposition for minimum rank approximation. Information Theory, IEEE Transactions on, 56(9):4402–4416, 2010.
125. A. Kyrillidis and V. Cevher. Matrix recipes for hard thresholding methods. Journal of mathematical imaging and vision, 48(2):235–265, 2014.
126. 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.
127. 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.
128. J. Cai, E. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. SIAM Journal on Optimization, 20(4):1956–1982, 2010.
129. 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.

130. 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.
131. F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65(6):386–408, 1958.
132. Robin M. Schmidt, Frank Schneider, and Philipp Hennig. Descending through a crowded valley - benchmarking deep learning optimizers. CoRR, abs/2007.01547, 2020.
133. John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res., 12(null):2121–2159, jul 2011.
134. Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Marc' aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, Quoc Le, and Andrew Ng. Large scale distributed deep networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems, volume 25. Curran Associates, Inc., 2012.
135. Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.