Fast spectral algorithms from sum-of-squares analyses: the planted sparse vector problem

In this lecture, we discuss how to design fast and practical spectral algorithms by building on the analysis of sum-of-squares programs. Specifically, we will give a quasi-linear algorithm for the the *planted sparse vector problem.*

1 Planted sparse vector problem

We consider the task of finding the sparsest nonzero vector in a linear subspace.

Problem 1.1 (Sparsest vector in a subspace)**.** Let $W \subset \mathbb{R}^n$ be a linear subspace of dimension d. Given an arbitrary orthonormal basis $a_0, \dots, a_{d-1}$ for W, and letting $A \in \mathbb{R}^{n \times d}$ be the matrix whose columns are given by the a_i, the *sparsest vector* problem asks us to compute

$$v^* = \arg\min_{\|z\|=1} \|Az\|_0, \tag{1}$$

where $\|\cdot\|_0$ is the ℓ_0 norm, which simply counts the number of nonzero entries.[1]

The problem is known to be NP-hard for arbitrary subspaces [McC83, CP86]. However, we may still hope to solve it under favorable conditions. Inspired by an application to the sparse dictionary learning problem in a work by Spielman, Wang, and Wright [SWW12], as well as applications to sparse principal components analysis, Demanet and Hand [DH14] introduced the following distributional variant:

Problem 1.2 (Planted sparse vector problem)**.** Given an arbitrary orthonormal basis of a subspace spanned by vectors $v_0, v_1, \dots, v_{d-1} \in \mathbb{R}^n$, where $\|v_0\|_0 \leqslant \varepsilon n$ and $v_1, \dots, v_{d-1} \sim \mathcal{N}(0, \mathbb{1}_n)$, output a vector $v \in \mathbb{R}^n$ which is at least $\sqrt{1-\eta}$-correlated with the sparse vector v_0, so that $\langle \frac{v}{\|v\|}, \frac{v_0}{\|v_0\|} \rangle^2 > 1 - \eta$.

This distributional problem can be thought of as an average-case real (as opposed to finite field) version of the "shortest codeword" or "lattice shortest vector" problem.

This problem is well-defined (in the sense that it is information theoretically solveable) for η a small constant (say $\frac{1}{100}$) when $d = o(n)$ and $\varepsilon < 0.9$.[2] We wish to solve the problem placing as few restrictions as possible on d, ε within this information-theoretically tractable regime. The most general algorithm for the $\varepsilon = \Omega(1)$ regime is due to Barak et al. [BKS14], who give a polynomial-time (something like $O((nd)^{24})$) algorithm based on a sum-of-squares program which can handle $\varepsilon = \Omega(1)$ so long as $d < n^{1/2}$.[3] Building on this work, [HSSS15] obtain an algorithm which runs in *near-linear* time in the input size, $\tilde{O}(nd)$, with comparable performance. Most interestingly for us, this faster algorithm is obtained via a *general strategy for taking an SoS analysis and extracting from it a fast spectral algorithm.*

[1] The constraint $\|z\| = 1$ is chosen arbitrarily for scaling; all that matters is that $z \neq 0$.

[2] In actuality, the information-theoretically solveable regime is larger, and can be precisely characterized; but for our purposes today this will not matter.

[3] We discuss the algorithmic guarantees of prior works in the bibliographic remarks at the end of these notes.

Theorem 1.3. *There is a spectral algorithm for Problem 1.2 running in time $\tilde{O}(nd)$ which for $d \ll \tilde{O}(n^{1/2})$, with high probability over the choice of $v_1, \dots, v_{d-1}$ and the randomness of the algorithm, recovers a vector v^* such that $\langle \frac{v^*}{\|v^*\|}, \frac{v_0}{\|v_0\|} \rangle \geqslant 1 - O(\varepsilon^{1/4}) - o(1)$.*

We will prove the above theorem. We'll start first by describing the SoS algorithm of Barak et al. upon which we build.

2 Analysis of the SoS Algorithm

Rather than trying to directly trying to find a sparse vector, we will optimize a polynomial objective function which is a proxy for sparsity. Then we will prove that (under the conditions $\varepsilon < \frac{1}{100}$ and $d < \sqrt{n}$) the optimizer is v_0, and there is a degree-4 SoS proof of this fact.

The p-to-q norm ratio as a proxy for sparsity. Recall the definition of the ℓ_p norm:

Definition 2.1 (ℓ_p norm)**.** For $p \geqslant 1$, the ℓ_p norm of $u \in \mathbb{R}^n$ is defined as $\|u\|_p = (\sum_{i=1}^n u_i^p)^{1/p}$.

From Hölder's inequality, we have that for $p < q$ and $u \in \mathbb{R}^n$, the ratio $\|u\|_q / \|u\|_p$ is larger for sparser u:

Lemma 2.2 (q-to-p norm for sparse vectors)**.** *For any vector $u \in \mathbb{R}^n$ with $\|u\|_0 = k > 0$ and any $q \geqslant p \geqslant 0$,*

$$\frac{\|u\|_q}{\|u\|_p} \geqslant \frac{1}{k^{1/p-1/q}}. \tag{2}$$

Proof. Without loss of generality, assume the nonzero entries of u are $u_1, \dots, u_k$. Now by Hölder's inequality,

$$(u_1^q + \cdots + u_k^q)^p \underbrace{(1 + \cdots + 1)}_{k \text{ times}}^{q-p} \geqslant (u_1^p + \cdots + u_k^p)^q.$$

Taking the pqth root and rearranging gives us the result. □

Lemma 2.2 establishes that a sparse vector "looks sparse" in the ratio-of-norms sense. The converse is unfortunately not true (see Example 2.3 below).

Example 2.3. The vector $v \in \mathbb{R}^n$ with $v_1 = 1$ and $v_i = \frac{1}{n(n-1)}$ for all $i \neq 1$ is maximally dense i.e. $\|v\|_0 = n$, but $\|v\|_\infty / \|v\|_1 = 1/(1 + 1/n) = 1 - O(\frac{1}{n})$.

However, for q, p suitably close together, a random subspace is unlikely to contain a non-sparse vector with low ℓ_q to ℓ_p ratio, thus the ℓ_q to ℓ_p ratio is a good proxy for sparsity for the *planted sparse vector* problem, where we are identifying a sparse vector planted within a random subspace.

Different pairs of p, q are possible. [DH14] and [QSW16] use the ℓ_∞ to ℓ_1 ratio, but a random subspace of dimension $\omega(n^{1/4})$ will contain a vector which is 0.01-sparse in an ∞-to-1 sense with high probability. Some works suggest using ℓ_2 to ℓ_1 ratio, which can handle large range of d and ε, but this ratio is hard to compute even for random subspaces. [BKS14, HSSS15] consider ℓ_4 to ℓ_2 ratio, which turns out to be easy to approximate for random and near random subspaces, using a degree 4 SOS system.

Formally, given an orthonormal basis $A \in \mathbb{R}^{n \times d}$ for a d-dimensional subspace $U \subseteq \mathbb{R}^n$, we aim to understand the value

$$\max_{x \in \mathbb{R}^n, \|x\|_2^2 = 1} f(x), \quad \text{for } f(x) = \|Ax\|_4^4 = \sum_{i=1}^d (Ax)_i^4; \tag{3}$$

This is equivalent to maximizing $\|v\|_4^4$ over $v \in U$ with $\|v\|_2^2 = 1$, since the columns of A are an orthonormal basis and thus $\|Ax\| = \|x\| = 1$. The polynomial f has a convenient matrix representation:

$$f(x) = (x \otimes x)^\top M (x \otimes x), \quad \text{for } M = \sum_{i=1}^{n} (a_i a_i^\intercal)^{\otimes 2} \text{ where } a_i \text{ is the } i^{th} \text{ row of } A \tag{4}$$

We'll show that the spectrum of M can give us an algorithm for distinguishing between the case when U contains a planted sparse vector and U is a completely random subspace. First, we will see that M has a large eigenvalue in the case when U contains a planted sparse (unit) vector v_0. Let x_0 be the coefficients of v_0 in the basis A. Then by definition,

$$f(x_0) = \langle x_0^{\otimes 2}, M x_0^{\otimes 2} \rangle = \|v_0\|_4^4 \geqslant \frac{1}{\epsilon n}.$$

On the other hand, one can show that orthogonal to $x_0^{\otimes 2}$, M has small eigenvalues (with the exception of a spurious eigenvalue that one can project away):

Lemma 2.4. *Suppose $M = \sum_{i=1}^n (a_i a_i^\top)^{\otimes 2}$ where $a_1, \dots, a_n$ are the rows of a matrix $A \in \mathbb{R}^{n \times d}$ whose columns form an orthonormal basis for a d-dimensional random subspace of $\mathbb{R}^n$, sampled by taking the span of $v_1, \dots, v_d \sim \mathcal{N}(0, \mathbb{1})$. Let $\Pi = \mathbb{1} - \phi\phi^\top$ for $\phi = \frac{1}{\sqrt{d}} \sum_{i=1}^d e_i^{\otimes 2}$. There exist universal constants C, c such that for n sufficiently large, with high probability over the choice of the subspace, so long as $d^2 > Cn$,*

$$\|\Pi M \Pi\|_{op} = \frac{c}{n},$$

and further, $\phi^\top M \phi = O(\frac{d}{n})$ with high probability.

The proof consists of matrix concentration arguments, which we will not give here. Intuitively, the condition $d^2 = \Omega(n)$ is required in order to ensure that the matrix M is full-rank (which we need to ensure that it does not have any eigenvalue too large compared to the trace).

This lemma establishes that there is a degree-4 SoS proof that $f(x) = O\left(\frac{1}{n}\right)$ when U is completely random (the "null" case): for any vector $x \in \mathbb{R}^d$, $\langle x, \phi \rangle = \frac{1}{\sqrt{d}} \|x\|^2$ is a polynomial equality. Hence, expanding Π and then applying Cauchy-Schwarz,

$$\begin{aligned}
\langle x^{\otimes 2}, \Pi M \Pi x^{\otimes 2} \rangle &= \langle x^{\otimes 2}, M x^{\otimes 2} \rangle - 2\langle x^{\otimes 2}, \phi \rangle \langle \phi, M x^{\otimes 2} \rangle + \langle x^{\otimes 2}, \phi \rangle^2 \cdot \phi^\top M \phi \\
&= \langle x^{\otimes 2}, M x^{\otimes 2} \rangle \pm 2\langle x^{\otimes 2}, \phi \rangle \sqrt{\langle \phi, M\phi \rangle \langle x^{\otimes 2}, M x^{\otimes 2} \rangle} + \langle x^{\otimes 2}, \phi \rangle^2 \cdot \phi^\top M \phi \\
&= f(x) \pm 2\tfrac{1}{\sqrt{d}} \|x\|^2 \cdot \sqrt{O(\tfrac{d}{n}) f(x)} + \tfrac{1}{d} \|x\|_2^4 \cdot O(\tfrac{d}{n}) \geqslant \left(\sqrt{f(x)} \pm O(\tfrac{1}{\sqrt{n}}) \right)^2,
\end{aligned} \tag{5}$$

Which gives us our conclusion. Further, since the program is a relaxation, we know that the SoS value is at least $\Omega(\frac{1}{\epsilon n})$ when there is a planted sparse vector, as witnessed by x_0.

3 Spectral Algorithm using the SoS certificate

Our first runtime improvement over the SoS algorithm above is simple: we observe that rather than solving an SoS relaxation of the program in (3), *we can directly construct the matrix $\tilde{M} = \Pi M \Pi$ from our input A*, and check if its maximum eigenvalue is $\Omega(\frac{1}{\epsilon n})$. The sequence of polynomial inequalities ending in (5) along with the test vector $x = x_0$ can be used to show that the maximum eigenvalue of $\tilde{M}$ is at least $\Omega(\frac{1}{\epsilon n})$ in the planted case, and at most $O(\frac{1}{n})$ in the null case.

This immediately gives us a $\tilde{O}(d^4)$-time algorithm for distinguishing the planted case from the "null" case where U is completely random: using power iteration, we find the maximum eigenvalue of $\tilde{M}$, and if it is at least $\frac{C'}{\varepsilon n}$ for a well-chosen constant $C' > c$, we declare that we are in the planted case; otherwise we declare that we are in the null case. With some additional effort, one can also show that $x_0^{\otimes 2}$ is close to the top eigenvector of $\Pi M \Pi$, which also gives a spectral algorithm for estimating v_0. In this way, we have bypassed the cost of solving the semidefinite program (which naively requires runtime $\Omega(d^{12})$).

Still, $\tilde{O}(d^4)$ time is fairly slow compared to the size of the input, dn, when $d \approx \sqrt{n}$. We will see below how to take this spectral certificate and "compress" it into a lower-dimensional matrix whose spectrum still contains information about v_0.

4 Near-linear time algorithms via partial trace

Ultimately, the SoS analysis is using some spectral properties of a large matrix whose entries are some function of the input i.e. the matrix M in (4). Above, we saw how to get a faster algorithm by constructing M "by hand" instead of solving the SoS system. Here, we'll see that we can use an operation called the *partial trace* to compress this large matrix to a smaller matrix while preserving its spectral properties, specifically the eigenvalue gap i.e. the gap between the largest and second largest eigenvalues.

Definition 4.1 (Partial Trace)**.** The partial trace $\mathrm{Tr}_{\mathbb{R}^d} : \mathbb{R}^{d^2 \times d^2} \to \mathbb{R}^{d \times d}$ is the unique linear operator that satisfies $\mathrm{Tr}_{\mathbb{R}^d} A \otimes B = (\mathrm{Tr}\, A) \cdot B$ for all $A, B \in \mathbb{R}^{d \times d}$.

The Partial trace operation is natural in quantum information theory, where it corresponds to a marginal quantum density on a subset of qubits (the quantum analog of the classical notion of a marginal distribution over a subset of variables). Partial trace operations have also been applied in rounding SoS relaxations.[4]

To see how the partial trace can be used to compress large matrices to smaller ones while preserving the eigenvalue gap, consider the following simplified version of the planted vector problem.

Example 4.2. Given a matrix $M \in \mathbb{R}^{d^2 \times d^2}$ of the form $M = \tau \cdot (x \otimes x)(x \otimes x)^\intercal + A \otimes B$ for some unit vector $x \in \mathbb{R}^d$ and matrices $A, B \in \mathbb{R}^{d \times d}$, we wish to recover the vector x. Think of $\tau \cdot (x \otimes x)(x \otimes x)^\intercal$ as the signal and $A \otimes B$ as noise.

The matrix $A \otimes B$ is a tensor product and thus has spectral norm $\|A \otimes B\| = \|A\| \cdot \|B\|$, and so when $\tau \gg \|A\| \cdot \|B\|$, the matrix M has a noticeable spectral gap, and the top eigenvector of M will be close to $v \otimes v$. If it happens to be the case that $|\mathrm{Tr}\, A| \approx \|A\|$, the partial trace matrix $\mathrm{Tr}_{\mathbb{R}^d} M = \tau \cdot \|x\|^2 \cdot xx^\intercal + \mathrm{Tr}(A) \cdot B$ has a spectral gap matching that of M. So, we can still recover x, but now we only need to compute the top eigenvector of a $d \times d$ (as opposed to $d^2 \times d^2$) matrix. If A is a Wigner matrix (e.g. a symmetric matrix with i.i.d. ± 1 entries), then both $|\mathrm{Tr}(A)|, \|A\| \approx \sqrt{n}$, and the above condition is indeed met.

In our setting the "noise" component is not as simple as $A \otimes B$ with A a Wigner matrix. Nonetheless, with some additional care, we are able to ensure that the noise displays a similar behavior under partial trace operations.

Partial trace, first attempt. To improve upon the $\tilde{O}(d^4)$ runtime in Section 2, we consider the matrix

$$N = \mathrm{Tr}_{\mathbb{R}^d} M = \sum_{i=1}^{n} \|a_i\|_2^2 a_i a_i^\intercal.$$

[4]Specifically, the operations of reweighing and conditioning, used in rounding algorithms for sum-of-squares such as [BKS14, BKS15], correspond to applying a partial trace operation to the moments matrix returned by the sum-of-squares relaxation.

For a cleaner exposition, let us consider the "decision" version of the problem, where we are trying to distinguish the YES case (U contains a planted sparse vector) from the NO case (U is the span of d vectors sampled iid from $\mathcal{N}(0, \mathbb{1}_n)$). In the YES case, the largest eigenvalue of N is

$$\lambda_{\text{YES}} \geqslant \langle x_0, N x_0 \rangle = \sum_{i=1}^{n} \langle a_i, x_0 \rangle^4 + a_i^\top (\mathbb{1} - x_0 x_0^\top) a_i \cdot \langle x_0, a_i \rangle^2 \approx \|v_0\|_4^4 + \frac{d}{n} \geqslant \frac{1}{\varepsilon n} + \frac{d}{n},$$

where in our approximations we use that a_i are distributed roughly like random vectors from $\mathcal{N}(0, \frac{d}{n}\mathbb{1}_d)$ (this can be formalized). In the NO case, we want to bound the largest eigenvalue of N. In order to simplify the bound on the spectral norm of N in the NO case, suppose that the columns of A are iid samples from the Gaussian distribution $\mathcal{N}(0, \frac{1}{n} I)$ (rather than an orthogonal basis for the random subspace – this can again be formalized). In this simplified setup, the matrix N in the NO case is the sum of n i.i.d. matrices $\{\|a_i\|_2^2 a_i a_i^\top\}$ and using matrix concentration inequalities (such as Matrix Bernstein) we can upper bound its spectral norm λ_{NO} by $d/n + O((\frac{d}{n})^{3/2})$. Hence, using the spectral norm of N, we will be able to distinguish between the YES case and the NO case as long as $\frac{1}{\epsilon n} \gg (\frac{d}{n})^{3/2}$, which is equivalent to $(n/\epsilon^2)^{1/3} \gg d$. For $\epsilon = 0.01$, this implies $d \ll O(n^{1/3})$, which is which is somewhat worse than the bound $\sqrt{n}$ bound on the dimension d that the SoS algorithm achieves. We will make a correction to account for this.

Re-centered Partial Trace. Recall that $\operatorname{Tr} B = \sum_i \lambda_i(B)$ for a symmetric matrix B. As discussed above, the partial trace approach works best when the noise behaves as the tensor of two Wigner matrices, in that there are cancellations when the eigenvalues of the noise are summed. In our case, the noise terms $(a_i a_i^\top) \otimes (a_i a_i^\top)$ do not have this property, as in fact $\operatorname{Tr} a_i a_i^\top = \|a_i\|_2^2 \approx d/n$. Thus, in order to improve the dimension bound, we will center the eigenvalue distribution of the noise part of the matrix. This will cause it to behave more like a Wigner matrix, in that the spectral norm of the noise will not increase after a partial trace. Consider the partial trace of a matrix $\tilde{M}$ of the form

$$M - \alpha \cdot I \otimes \sum_i a_i a_i^\top$$

for some constant $\alpha > 0$. Its partial trace is

$$\tilde{N} = \sum_{i=1}^{n} (\|a_i\|_2^2 - \alpha) a_i a_i^\top$$

We set $\alpha = d/n$ so that $\tilde{N}$ has expectation 0 in the NO case, and spectral norm $\lambda_{\text{NO}} \leqslant O(d/n^{3/2})$. In the YES case, we still have $\lambda_{\text{YES}} \geqslant \langle x_0, \tilde{N} x_0 \rangle \approx \|v_0\|_4^4 \geqslant \frac{1}{\epsilon n}$. Thus, we can distinguish between the two cases when $\frac{1}{\epsilon n} \gg d/n^{3/2}$, which is equivalent to $\sqrt{n/\epsilon^2} \gg d$. This is exactly the bound we were shooting for.

Summary: fast spectral algorithm for planted sparse vector. We formally state the algorithm for planted sparse vector, which at a high level computes the top eigenvector of $\tilde{N}$. The proof of correctness follows from the discussion above (modulo the details of working with the orthogonal basis A vs. random vectors, and the difference between decision vs. estimation). We'll give a runtime analysis below.

Algorithm 4.3 (Planted sparse vector)**.** Input: orthogonal basis $A \in \mathbb{R}^{d \times n}$ of $U = \operatorname{span}(v_0, \dots, v_{d-1})$ as in Problem 1.2.

- Compute the "leverage scores" $\|a_1\|_2^2, \dots, \|a_n\|_2^2$ where a_i is the i^{th} row of A.

- Compute the top eigenvector x_0' of the matrix

$$\tilde{N} = \sum_{i=1}^{n} (\|a_i\|_2^2 - d/n) a_i a_i^\top$$

- Output Ax_0' as an approximation to the sparse vector v_0.

Runtime analysis: Computing the leverage scores clearly takes $O(dn)$ time. By a similar argument to the argument to distinguish the YES/NO case, the matrix $\tilde{N}$ has a constant spectral gap, so we can use power iteration to find the largest eigenvalue and corresponding eigenvector of $\tilde{N}$ in $O(\log n)$ iterations. Each step of power iteration can be computed in $O(dn)$ time, since $\tilde{N} = A^\top DA$ for A the $n \times d$ basis and D the diagonal matrix whose ith entry is given by $\|a_i\|^2 - \frac{d}{n}$.

5 Conclusion

Bibliographic remarks. These notes present the algorithm of [HSSS15] for speeding up sum-of-squares algorithms as executed for the planted sparse vector problem; in that paper, the authors also use the same strategy for tensor Principal Component Analysis and overcomplete tensor decomposition.

We discuss the guarantees of some prior works for the planted sparse vector problem. Demanet and Hand [DH14] gave a polynomial-time algorithm based on linear programming which is able to solve the problem so long as $\varepsilon < \frac{1}{\sqrt{d}}$, but for downstream applications, it is more interesting to consider the case when ε is a small constant, say, $\varepsilon \approx \frac{1}{100}$. Qing et al. [QSW16] gave an algorithm running in time $\tilde{O}(n^2 d^5)$ algorithm which works up to $\varepsilon = \Omega(1)$ whenever $d < n^{1/4}$.

Contact. Comments are welcome at tselil@stanford.edu.

References

[BKS14] Boaz Barak, Jonathan A Kelner, and David Steurer. Rounding sum-of-squares relaxations. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 31–40, 2014. 1, 2, 4

[BKS15] Boaz Barak, Jonathan A Kelner, and David Steurer. Dictionary learning and tensor decomposition via the sum-of-squares method. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 143–151, 2015. 4

[CP86] T. F. Coleman and A. Pothen. The null space problem i. complexity, 1986. 1

[DH14] Laurent Demanet and Paul Hand. Scaling law for recovering the sparsest element in a subspace, 2014. 1, 2, 6

[HSSS15] Samuel B. Hopkins, Tselil Schramm, Jonathan Shi, and David Steurer. Speeding up sum-of-squares for tensor decomposition and planted sparse vectors. *CoRR*, abs/1512.02337, 2015. 1, 2, 6

[McC83] S. T. McCormick. A combinatorial approach to some sparse matrix problems, 1983. 1

[QSW16] Q. Qu, J. Sun, and J. Wright. Finding a sparse vector in a subspace: Linear sparsity using alternating directions. *IEEE Transactions on Information Theory*, 62(10):5855–5880, 2016. 2, 6

[SWW12] Daniel A. Spielman, Huan Wang, and John Wright. Exact recovery of sparsely-used dictionaries, 2012. 1