Review of Van Vu's result

Elena Khusainova

28 April 2017

Abstract

Here is a review of Van Vu's 2005 paper on the upper bound of a symmetric random matrix with independent bounded entries. His coding scheme was cleaned and slightly improved.

1 Bounds for a spectral norm of a symmetric matrix

In his paper ([1]) Van Vu improves the upper bound (the result of Furedi and Kolmos [2]) for the spectral norm of symmetric random matrices with independent entries.

Let A be a symmetric matrix, with independent (but not necessarily identical) entries, such that $\mathbb{E}a_{ij} = 0$, $|a_{ij}| \leq K$, $Var(a_{ij}) = \sigma^2$ for all $1 \leq i \leq j \leq n$.

The spectral norm of a matrix A is defined as:

$$\lambda(A) = \sup_{v \in \mathbb{R}^n, \|v\|_2 = 1} |v' A v|$$

The previous result obtained by Furedi and Kolmos is the following:

Theorem 1. For a random matrix A as above there is a positive constant $c = c(\sigma, K)$ such that

$$2\sigma\sqrt{n} - cn^{1/3}\ln n \leq \lambda(A) \leq 2\sigma\sqrt{n} + cn^{1/3}\ln n,$$

hold almost surely.

In his paper Van Vu points out the mistake in Furedi and Komlos proof and proves the better bound:

Theorem 2. For a random matrix A as above there is a positive constant $c = c(\sigma, K)$ such that

$$\lambda(A) \leq 2\sigma\sqrt{n} + cn^{1/4}\ln n,$$

hold almost surely.

1.1 Van Vu's coding scheme

First let us introduce some notations for this section. Let A be an $n \times n$ symmetric matrix with independent bounded entries $A[i,j]$ with bounded variance:

$$A[i,j] < K, \text{ for all } i,j, \ \mathbb{E}\big(A[i,j]^2\big) \leq \sigma^2.$$

Without loss of generality we can assume that $K = 1$ and that $A[i,i] = 0$ for all i. To see the later define D to be the diagonal matrix which diagonal entries are the same as the diagonal entries of A, and let $\widetilde{A} := A - D$. Then

$$\lambda(A) - \lambda(\widetilde{A}) = \sup_{\|v\|_2=1} |v'(\widetilde{A} + D)v| - \sup_{\|u\|_2=1} |u'\widetilde{A}u| \leq$$

$$\leq \sup_{\|v\|_2=1} |v'Dv| = \sup_{\|v\|_2=1} \sum_i |A[i,i]|v_i^2 \leq K \sup_{\|v\|_2=1} \|v\|_2 = 1.$$

For his proof Vu uses the usual Wigner's trace method. That is for any even k we can use the following inequality to bound the spectral norm:

$$\begin{aligned}\lambda(A)^k = \lambda(A^k) \leq \sum_{i=1}^{n} \lambda_i(A^k) = trace(A^k) = \\ = \sum_{i_0=1}^{n}\sum_{i_1=1}^{n}...\sum_{i_{k-1}=1}^{n} A[i_0,i_1]A[i_1,i_2]...A[i_{k-1},i_0] = \\ = \sum_{\mathbb{i}} A[\mathbb{i}]\end{aligned} \tag{1}$$

where $\lambda_i(A^k)$ are the eigenvalues of the matrix A^k and $\mathbb{i} = (i_0, i_1, ..., i_{k-1}, i_0)$ is a multi-index. Note that from the fact that all entries of the matrix A have expectation zero it follows that for each summand $A[\mathbb{i}]$ its expectation is also zero if there is at least one factor of multiplicity 1. Define $\xi_1, ..., \xi_p$ to be distinct factors in the summand. Then

$$\mathbb{E}A[\mathbb{i}] = \mathbb{E}\xi_1^{\alpha_1}...\xi_p^{\alpha_p} = \prod_{i=1}^{p} \mathbb{E}\xi_i^{\alpha_i} \leq \prod_{i=1}^{p} \mathbb{E}\xi_i^2 K^{\alpha_i-2} \overset{\text{as } K=1}{=} \prod_{i=1}^{p} \mathbb{E}\xi_i^2 \leq \sigma^{2p-2},$$

where $\alpha_i \geq 2$ for all i and $\sum_i \alpha_i = k$. Thus we have an upper bound for the expectation of each summand and we will bound the number of summands next.

To proceed we will introduce a complete graph $G = G(V, E)$ with n vertices: $V = \{1, ..., n\}$, and closed (i.e the first and the last node are the same) walks of length k without self-loops on it:

$$\mathcal{W} := [i_0, i_1, ..., i_{k-1}, i_0],$$

where $\{i_0, ..., i_{k-1}\} \subseteq V$. Note that there exists a trivial one-to-one map between multi-indices for the summands in (1) and such walks. We will say from now on that a walk $\mathcal{W}_{\mathbb{i}} = [\mathbb{i}]$ corresponds to a summand $A[\mathbb{i}]$. And therefore we turned the problem of bounding the number of summands with a non-zero expectation into the problem of bounding the number of closed walks on a complete graph with n vertices of length k with each edge used at least two times. The last condition comes from the fact that each edge of the walk $\mathcal{W}_{\mathbb{i}}$ corresponds to a factor in the summand $A[\mathbb{i}]$ and as we mentioned above in order for $A[\mathbb{i}]$ to have a non-zero expectation all its factors have to have multiplicity greater or equal to two. For the same reason we did not allow the walk to have self-loops: a self-loop would mean that there is a factor $A[i_j, i_j]$, for some $j \in \mathbb{i}$, in the summand, but then the whole summand would be zero. Denote $W(n, k, p)$ to be a number of such walks, where p is the number of distinct vertices visited by it.

To bound $W(n, k, p)$ Vu created a scheme for mapping each walk into a sequence of $\oplus$ and $\ominus$ with some *hints* that we will explain below. We will call such sequences *codewords*. Then we will bound the number of walks by the number of the codewords.

Consider a walk

$$\mathcal{W} = [i_0, i_1, ..., i_{k-1}, i_0]$$

with p distinct vertices. Denote these p vertices as $v_1, ..., v_p$ in order of their appearance in the walk. Also denote the time we first visit the vertex v_t as:

$$T(v_t) = \min_j \{j : \ i_j = v_t\}$$

E.g. $T(v_1) = 0$, $T(v_2) = 1$. Obviously edges

$$E(v_2) := [i_{T(v_1)}, i_{T(v_2)}],$$

$$E(v_3) := [i_{T(v_3-1)}, i_{T(v_3)}],$$

$$\dots$$

$$E(v_p) := [i_{T(v_p-1)}, i_{T(v_p)}]$$

form a tree with root at i_0, that we denote $\mathcal{T}(\mathcal{W})$, and therefore

$$2(p-1) \leq k. \tag{2}$$

Now, we build a map in the following 2-stage way:

1. The edges $E(v_2), ..., E(v_p)$ of the walk are coded with $\oplus$, their second appearances are coded with $\ominus$. Any other edge $\mathfrak{e} = [e_1, e_2]$ is coded with its second vertex: e_2 and called *neutral*.

   Example: Consider a walk

   $$\mathcal{W} = [6, 2, 5, 3, 2, 7, 2, 5, 6, 2, 3, 5, 6, 2, 6].$$

   Then $(v_1, ..., v_5) = (6, 2, 5, 3, 7)$ and the code at this stage, denoted as $\mathcal{C}_1(\mathcal{W})$, would be:

   $$\mathcal{C}_1(\mathcal{W}) = \oplus \oplus \oplus v_2 \oplus \ominus \ominus \, v_1 \ominus v_4 \ominus v_1 v_2 v_1$$

2. Note that knowing the sequence $(v_1, ..., v_p)$ and the code $\mathcal{C}_1(\mathcal{W})$ in the example above is not enough to decode it back to the original walk: one can easily decode first 5 symbols of the codeword (or equivalently first 6 vertices of the walk) but then the walk will end up at the vertex $v_2 = 2$ and the decoded part of the walk and its corresponding tree would look like:

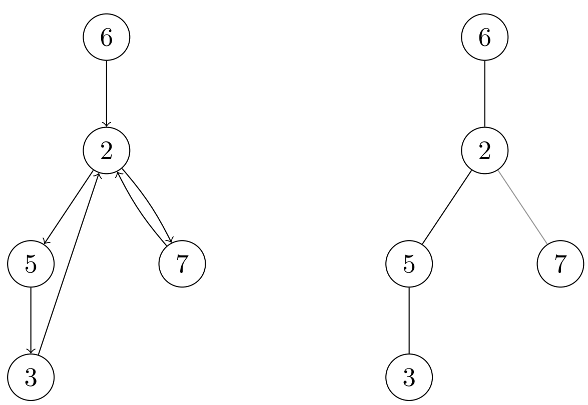


Figure 1: *the decoded part of the walk $\mathcal{W}$ (left) and the decoded part of the tree $\mathcal{T}(\mathcal{W})$ (right). We colored edge $[2, 7]$ of the tree $\mathcal{T}(\mathcal{W})$ gray because by this time it has already been traversed twice.*

Thus we know that the next step of the walk (coded with $\ominus$) should use either the edge $[2, 6]$ or $[2, 5]$, but we don't know which one. We will call such steps of the walk *ambiguous*, that is step six of the walk in the example is ambiguous. This is the time to introduce the hints. For now we will put a destination node (hint) at the end of each series of consecutive $\ominus$, but later we will see that not all of them need hints. The codeword for the example above will be:

$$\mathcal{C}_2'(\mathcal{W}) = \oplus \oplus \oplus v_2 \oplus \ominus \ominus_{v_3} \, v_1 \ominus_{v_2} v_4 \ominus_{v_3} v_1 v_2 v_1$$

We will call edges with hints *critical*. Note that because we keep the tree in mind it is enough to give such 'destination' hints as there is only one path between two nodes in a tree, thus knowing where we are and where we will end up is enough to know how to get there. Also this path has to be a subpath of a path that starts with the ambiguous step and ends at one of the leaves of the tree $\mathcal{T}(\mathcal{W})$, thus we can give this leaf as a hint and the final code will look like:

$$\mathcal{C}_2(\mathcal{W}) = \oplus \oplus \oplus v_2 \oplus \ominus \ominus_{v_4} v_1 \ominus_{v_4} v_4 \ominus_{v_4} v_1 v_2 v_1.$$

Now, before continuing we need a piece of theory regarding Dyck paths.

Definition: A *Dyck path* of length n on an integer lattice is a path form $(0, 0)$ to (n, n) that does not cross (but are allowed to touch) the diagonal $y = x$.

Properties:

- A Dyck path can be written as sequence of $\oplus$ and $\ominus$ where $\oplus$ is a step *right* and $\ominus$ is a step *up*. Then the condition that the path cannot cross the diagonal can be interpreted as that at each time the number of steps *up* (or $\ominus$) taken is less or equal to the taken number of steps *right* (or $\oplus$).

- The Catalan's number defined as

  $$C_n = \frac{1}{n+1}\binom{2n}{n}$$

  is the total number of Dyck paths between $(0, 0)$ and (n, n).

- A Dyck path can also be viewed as a way to code a rooted tree: consider each step *right* as going to a child of the current node that has not been visited before (we visit them from left to right) and step *up* as going to the parent of the current node. Then the condition that Dyck paths do not cross the diagonal $y = x$ is equivalent to that we will never go to the parent of the root.

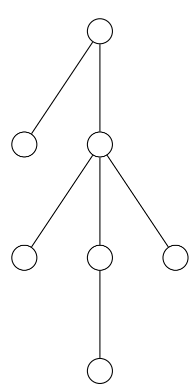
  Example: Consider the following Dyck path:

  $$right, up, right, right, up, right, right, up, up, right, up, up$$

  or equivalently:

  $$\oplus \ominus \oplus \oplus \ominus \oplus \oplus \ominus \ominus \oplus \ominus\ominus$$

  Then it would correspond to the following rooted tree:



Note that because of the above properties the parts of a codeword that are Dyck paths themselves do not cause any problems while decoding. Van Vu called such parts *redundant*. For the example we used we color the redundant part gray:

$$\oplus \oplus \oplus v_2 \oplus\ominus \ominus_{v_4} v_1 \ominus v_4 \ominus_{v_4} v_1 v_2 v_1.$$

Thus only those stretches of $\ominus$ that would remain in the codeword if we deleted all redundant parts need hints.

So, now we have a map that produces a unique codeword for each of the closed walks on a complete graph with n vertices. The bound for the number of possible codewords will be the desired bound for $W(n,k,p)$.

We need just a couple more things before we can get this bound. Let N_p be the number of neutral edges in the walk. Clearly,

$$N_p = k - 2p + 2.$$

1. Note that if we deleted all redundant parts from a codeword there would not be any $\oplus$ before a stretch of $\ominus$, so there must be a neutral edge between two critical ones, plus there must be a neutral edge before the first critical one as the codeword always starts with $\oplus$, thus the total number of critical edges is less or equal to N_p.

2. For a critical edge the possible number of hints is less than $N_p + 2$.

   Proof: Let us draw a tree $\mathcal{T}(\mathcal{W})$ together with the decoding in the following way: at the times $T(v_2), ..., T(v_p)$ we draw the corresponding edges $E(v_2) = [i_{T(v_1)}, i_{T(v_2)}]$, $\ldots$, $E(v_p) = [i_{T(v_p)-1}, i_{T(v_p)}]$ for the tree, then if the walk traverses one of the tree edges for the second time we erase it. Thus by the time we hit an ambiguous edge we will have a forest (part of the tree $\mathcal{T}(\mathcal{W})$). Consider its component the walk is currently in. To proceed it is enough to give one of the leaves of that component as a hint. Let us illustrate our argument with an example.

   Example: Let us assume that we successfully decoded first t steps of the walk with $i_t = v_7$ and the current component of the forest looks like:

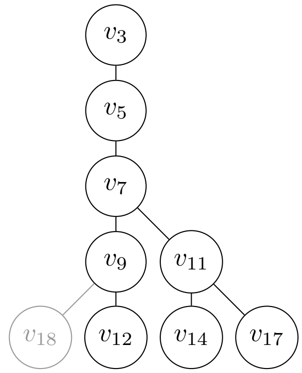


Figure 2: *The current component (black) of the tree $\mathcal{T}(\mathcal{W})$ with a redundant part (gray) that yet to be discovered.*

   with the root v_3. Note that the indices of the children have to be bigger than those of the parents. Let's consider a more difficult case that the stretch ahead that needs decoding has redundant parts: $\ominus \oplus \ominus\ominus$. But recall that as redundant parts are Dyck paths in their nature their have to end exactly where they started and they cannot cause any ambiguity, i.e no vertex of a redundant part can be used as a hint as they all are doomed to be erased by the end of this part. So adding a hint 'v_{12}' to the last $\ominus$, i.e. $\ominus\oplus\ominus\ominus_{v_{12}}$, we therefore make this stretch uniquely decoded into $[v_7 v_9 v_{18} v_9 v_{12}]$.

   Now let us take a closer look at any of the leaves, say v_{12}. From the tree we know that $[i_{T(v_{12})-1}, i_{T(v_{12})}] = [v_9, v_{12}]$. But how did the walk $\mathcal{W}$ get from the node v_{12} to the next leaf of the current component, node v_{14}? Let the part of the walk $[i_{T(v_{12})}, ..., i_{T(v_{14})}]$ be coded as $[c_1, ..., c_{T(v_{14})-T(v_{12})}]$, where $c_j \in \{\oplus, \ominus, v_1, v_2, ...v_{13}\}$. Our claim is that at least one of the c_j must be in $\{v_1, ..., v_{13}\}$, i.e. neutral. Assume not, that is $c_j \in \{\oplus, \ominus\}$. The whole stretch $[c_1, ..., c_{T(v_{14})-T(v_{12})}]$ cannot be a Dyck path as it would mean that we would end up at v_{12} and not v_{14}. Thus if we removed all the redundant parts from it there would be some $\ominus$'s left. But that would mean using the edge $[v_{12}, v_9]$ for the second time, while we know that it has not been erased yet, therefore there must be a neutral edge in the stretch, meaning that there must be a neutral edge between any two leaves of the current component of the tree. So, if L is the number of leaves in the current component:

$$L \leq N_p + 1$$

adding another possible hint, the root of the component, we get the desired bound. $\square$

That is there are at most $\binom{k}{2p-2}$ ways to choose places for $\oplus$'s and $\ominus$'s in the codeword, at most 2^{2p-2} ways to put them in those places, there are at most p^{N_p} ways to fill the rest places with neutral edges and at most $(k-2p+3)^{k-2p+2}$ to put hints where they are needed, so

$$W(n,k,p) \leq \binom{k}{2p-2} 2^{2p-2} p^{k-2p+2} (k-2p+3)^{k-2p+2}$$

and

$$\mathbb{E}\ trace(A^k) \leq \sum_p \sigma^{2p-2} \frac{n!}{(n-p)!} W(n,k,p) \leq$$

$$\leq \sum_p \sigma^{2p-2} \frac{n!}{(n-p)!} \binom{k}{2p-2} 2^{2p-2} p^{k-2p+2} (k-2p+3)^{k-2p+2} =: \sum_p S(n,k,p).$$

Note that

$$\frac{S(n,k,p-1)}{S(n,k,p)} = \frac{\sigma^{2p-4} \frac{n!}{(n-p+1)!} \binom{k}{2p-4} 2^{2p-4} (p-1)^{k-2p+4} (k-2p+5)^{k-2p+4}}{\sigma^{2p-2} \frac{n!}{(n-p)!} \binom{k}{2p-2} 2^{2p-2} p^{k-2p+2} (k-2p+3)^{k-2p+2}} \leq$$

$$\leq \frac{1}{4\sigma^2(n-p+1)} \frac{k!(2p-2)!(k-2p+2)!}{(2p-4)!(k-2p+4)!k!} \frac{(p-1)^2(k-2p+5)^{k-2p+4}}{(k-2p+3)^{k-2p+2}} =$$

$$= \frac{(p-1)^2}{4\sigma^2(n-p+1)} \frac{(2p-2)(2p-3)}{(k-2p+4)(k-2p+3)} \frac{(k-2p+5)^{k-2p+4}}{(k-2p+3)^{k-2p+2}} =$$

$$= \frac{(p-1)^3(2p-3)}{2\sigma^2(n-p+1)(k-2p+4)} \frac{(k-2p+5)^{k-2p+4}}{(k-2p+3)^{k-2p+3}} \leq \frac{p^4}{2\sigma^2(n-p+1)} \left(\frac{k-2p+5}{k-2p+3}\right)^{k-2p+4}.$$

Note that as $2(p-1) \leq k$ then $k-2p+3 \geq 1$ and:

$$\left(\frac{k-2p+5}{k-2p+3}\right)^{k-2p+4} = \left(1 + \frac{2}{k-2p+3}\right)^{k-2p+4} \leq Const,$$

then as $n-p+1 \geq n-k/2$ if $k < n$:

$$S(n,k,p-1) \leq Const \cdot \frac{1}{\sigma^2} \frac{k^4}{n} S(n,k,p),$$

where the constant does not depend on σ and K.

Then for

$$k = a\sigma^{1/2} n^{1/4},$$

for some properly chosen constant a, we will have:

$$S(n,k,p-1) \leq \frac{1}{2} S(n,k,p)$$

and

$$\mathbb{E}\lambda(A)^k \leq \mathbb{E}trace(A^k) \leq \sum_p S(n,k,p) \leq 2S(n,k,k/2+1) \leq$$

$$\leq 2\sigma^k n(n-1)...(n-k/2)2^k \leq 2n(2\sigma\sqrt{n})^k.$$

From the above using Markov inequality and the inequality:

$$\forall x > -1: \; 1 - \frac{x}{1+x} = \frac{1}{1+x} < \exp(-x)$$

we can get:

$$\mathbb{P}(\lambda(A) \geq 2\sigma\sqrt{n} + t) \leq \frac{\mathbb{E}(\lambda(A)^k)}{(2\sigma\sqrt{n}+t)^k} \leq \frac{2n(2\sigma\sqrt{n})^k}{(2\sigma\sqrt{n}+t)^k} = 2n\left(1 - \frac{t}{2\sigma\sqrt{n}+t}\right)^k \leq$$

$$\leq 2n\exp(-tk/2\sigma\sqrt{n}),$$

then if $t := cn^{1/4}\ln n$ and as $k = c_1 n^{1/4}$ for some constant c_1:

$$\mathbb{P}(\lambda(A) \geq 2\sigma\sqrt{n} + cn^{1/4}\ln n) \leq 2n\exp(-cc_1\ln n/2\sigma) = o(1).$$

Thus the theorem 2 is proved.

References

[1] Van H. Vu. Spectral norm of random matrices. *Combinatorica*, 27(6):721–736, 2007.

[2] Zoltán Füredi and János Komlós. The eigenvalues of random symmetric matrices. *Combinatorica*, 1(3):233–241, 1981.