Recent Advances in Random Matrix Theory for Modern Machine Learning

Zhenyu Liao and Romain Couillet

CentraleSupélec, Université Paris-Saclay, France.
GSTATS IDEX DataScience Chair, GIPSA-lab, Université Grenoble-Alpes, France.

May 2, 2019, Kraków, Poland
Outline

1. Motivation
2. Sample covariance matrix for large dimensional data
3. RMT for machine learning: kernel spectral clustering
4. RMT for machine learning: random neural networks
5. From theory to practice
Motivation: the pitfalls of large dimensional statistics

- The big data era: both large dimensional and massive amount of data, the number of instances \( n \) and their dimension \( p \) are both large,
  - large size high resolution images, more involved machine learning systems.

- Counterintuitive phenomenon in the large \( n, p \) regime, e.g.,
  - The “curse of dimensionality” phenomenon:
    little difference between Euclidean distance \( \| x_i - x_j \| \) from the same or different clusters (classes), \( x_i, x_j \in \mathbb{R}^p \) for \( p \) large.
  - Classical machine learning algos (e.g., kernel spectral clustering) still work for large dimensional data, although we do not understand why . . .

- In need of refinement to understand and improve modern machine learning methods for large dimensional problems, made possible with RMT.

- From a RMT viewpoint: with nonlinearity involved and of implicit solution (from an optimization problem)
Sample covariance matrix in the large \( n, p \) regime

- For \( x_i \sim \mathcal{N}(0, C) \), estimate the **covariance matrix** from \( n \) data samples 
  \[ X = [x_1, \ldots, x_n] \in \mathbb{R}^{p \times n}. \]

- Classical maximum likelihood sample covariance matrix:
  \[
  \hat{C} = \frac{1}{n} \sum_{i=1}^{n} x_i x_i^T = \frac{1}{n} XX^T \in \mathbb{R}^{p \times p}
  \]
  of rank at most \( n \).

- In the regime where \( n \sim p \), conventional wisdom breaks down, for \( C = I_p \) with \( n < p \), SCM will **never** be correct:
  \[
  \|C - \hat{C}\| \not\to 0, n, p \to \infty
  \]
  with at least \( p - n \) zero eigenvalues!

- Typically what happens in deep learning: try to fit an **enormous** statistical model (60.2 M of ResNet-152) with **insufficient**, but still **numerous** data (14.2 M images of ImageNet dataset).
When is one under random matrix regime?

For $\mathbf{C} = \mathbf{I}_p$, as $n, p \to \infty$ with $p/n \to c \in (0, \infty)$: the Marčenko–Pastur law

$$
\mu(dx) = (1 + c^{-1})^+ \delta(x) + \frac{1}{2\pi cx} \sqrt{(x-a)^+(b-x)^+}
$$

(1)

where $a = (1 - \sqrt{c})^2$, $b = (1 + \sqrt{c})^2$ and $(x)^+ \equiv \max(x, 0)$.

- eigenvalues span on $[(1 - \sqrt{c})^2, (1 + \sqrt{c})^2]$.
- for $n = 100p$, spread on a range of $4\sqrt{c} = 0.4$ around the true value 1.

Figure: Eigenvalue distribution of $\hat{\mathbf{C}}$ versus Marčenko-Pastur law, $p = 500$, $n = 50000$. 

Z. Liao and R. Couillet (CentraleSupélec) 
Recent Advances in RMT for Modern ML 
May 2, 2019, Kraków, Poland 
7 / 31
Reminder on kernel spectral clustering

Two-step classification of \( n \) data points based on similarity \( S \in \mathbb{R}^{n \times n} \):

\[ \text{Top eigenvectors} \]

\[ \begin{align*}
\text{Eigenv. 1} \\
\text{Eigenv. 2}
\end{align*} \]
Reminder on kernel spectral clustering

$\rightarrow$ $k$-dimensional representation $\rightarrow$

EM or k-means clustering.
Loss of relevance of Euclidean distance

- Simplest binary Gaussian mixture classification setting
  \[ C_1 : x = \mu + z, \quad x \sim \mathcal{N}(\mu, I_p); \]
  \[ C_2 : x = -\mu + (I_p + E)^{1/2} z, \quad x \sim \mathcal{N}(-\mu, I_p + E). \]
  for \( z \sim \mathcal{N}(0, I_p) \).

- Neyman-Pearson test tells us: classification is non-trivial only when
  \[ \| \mu \| \geq O(1), \quad \| E \| \geq O(p^{-1/2}), \quad | \text{tr} E | \geq O(\sqrt{p}), \quad \| E \|_F^2 \geq O(1). \]

- In this non-trivial setting, for \( x_i \in C_a, x_j \in C_b \),
  \[ \frac{1}{p} \| x_i - x_j \|^2 = \begin{cases} \frac{1}{p} \| z_i - z_j \|^2 + Ap^{-1/2}, & \text{for } a = b = 2; \\ \frac{1}{p} \| z_i - z_j \|^2 + Bp^{-1/2}, & \text{for } a = 1, b = 2 \end{cases} \tag{2} \]

- For \( A, B \) both of order \( O(1) \) and \( A > B \) with high probability for \( p \) large, so
  \[ \max_{1 \leq i \neq j \leq n} \left\{ \frac{1}{p} \| x_i - x_j \|^2 - 2 \right\} \rightarrow 0 \tag{3} \]
  almost surely as \( n, p \rightarrow \infty \).
Kernel spectral clustering for large dimensional data

**Objective:** “cluster” data $x_1, \ldots, x_n$ into $K$ similarity classes.
Consider the RBF kernel matrix $K_{ij} = \exp \left( -\frac{1}{2p} \|x_i - x_j\|^2 \right)$.

**Figure:** Kernel matrices $K$ and the second top eigenvectors $v_2$ for small (left, $p = 5, n = 500$) and large (right, $p = 250, n = 500$) dimensional data.
But why kernel spectral clustering works?

The accumulated effect of the small “hidden” statistical information (in $\mu, E$).

$$K = \exp(-1) \left( 1_n 1_n^T + \frac{1}{p} Z^T Z \right) + g(\mu, E) \frac{1}{p} j j^T + * + o_{\|\cdot\|}(1)$$

(4)

with $Z = [z_1, \ldots, z_n] \in \mathbb{R}^{p \times n}$ and $j = [1_{n/2}; -1_{n/2}]$, the class-information vector.

Therefore

- **entry-wise**: for $K_{ij} = \exp \left( -\frac{1}{2p} \| x_i - x_j \|^2 \right)$,

$$K_{ij} = \exp(-1) \left( 1 + \frac{1}{p} z_i^T z_j \right) + \frac{1}{p} g(\mu, E) + *$$

so that $\frac{1}{p} g(\mu, E) \ll \frac{1}{p} z_i^T z_j$;

- **spectrum-wise**: $\| \frac{1}{p} Z^T Z \| = O(1)$ and $\| g(\mu, E) \frac{1}{p} j j^T \| = O(1)$ as well!
Neural networks and deep learning

Figure: Illustration of $L$-hidden-layer nonlinear neural networks

$\mathbf{W}_{L+1} \mathbf{H}_L \quad \mathbf{H}_L \equiv \mathbf{W}_L \sigma(\mathbf{H}_{L-1}) \quad \mathbf{H}_1 \equiv \mathbf{W}_1 \mathbf{X}$

with nonlinear activation function $\sigma(z)$: ReLU($z$) = max($z$, 0), Leaky ReLU max($z$, $az$) ($a > 0$) or sigmoid $\sigma(z) = (1 + e^{-z})^{-1}$, arctan, tanh, etc.
Random neural network with single hidden layer

\[ \beta \in \mathbb{R}^N \quad \text{random features} \]

\[ \sigma \quad \sigma \quad \sigma \quad \sigma \quad \sigma \]

\[ \Sigma \equiv \sigma(WX) \quad \text{random features} \]

\[ X \in \mathbb{R}^{p \times n} \]

- For random \( W \) and \( n, p, N \) large, \( \frac{1}{N} \Sigma^T \Sigma \) is closely related to

\[ K \equiv \frac{1}{N} \mathbb{E}_W[\sigma(WX)^T \sigma(WX)] \]

- For Gaussian \( W_{ij} \sim \mathcal{N}(0, 1) \), \( K \) is explicit for some \( \sigma(\cdot) \) via an integral trick

\[ K_{ij} = \mathbb{E}_W[\sigma(w^T x_i)\sigma(w^T x_j)] = (2\pi)^{-p/2} \int_{\mathbb{R}^p} \sigma(w^T x_i)\sigma(w^T x_j) e^{-\|w\|^2/2} \, dw \]

\[ = \frac{1}{2\pi} \int_{\mathbb{R}^2} \sigma(\tilde{w}^T \tilde{x}_i)\sigma(\tilde{w}^T \tilde{x}_j) e^{-\|\tilde{w}\|^2/2} \, d\tilde{w} \]

with \( \tilde{x}_i = [\|x_i\|; 0] \) and \( \tilde{x}_j = \left[ \frac{x_i^T x_j}{\|x_i\|} ; \sqrt{\|x_j\|^2 - \frac{(x_i^T x_j)^2}{\|x_i\|^2}} \right] \).
Nonlinearity in simple random neural networks

Table: $K_{i,j}$ for commonly used $\sigma(\cdot)$, $\angle \equiv \frac{x_i^\top x_j}{\|x_i\|\|x_j\|}$.

<table>
<thead>
<tr>
<th>$\sigma(t)$</th>
<th>$K_{i,j}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t$</td>
<td>$x_i^\top x_j$</td>
</tr>
<tr>
<td>max$(t, 0)$</td>
<td>$\frac{1}{2\pi} |x_i||x_j| (\angle \arccos (-\angle) + \sqrt{1 - \angle^2})$</td>
</tr>
<tr>
<td>$</td>
<td>t</td>
</tr>
<tr>
<td>$\varsigma_+ \max(t, 0) + \varsigma_- \max(-t, 0)$</td>
<td>$\frac{1}{2} (\varsigma_+^2 + \varsigma_-^2) x_i^\top x_j + \frac{|x_i||x_j|}{2\pi} (\varsigma_+ + \varsigma_-)^2 (\sqrt{1 - \angle^2} - \angle \cdot \arccos(\angle))$</td>
</tr>
<tr>
<td>$1_{t&gt;0}$</td>
<td>$\frac{1}{2} - \frac{1}{2\pi} \arccos (\angle)$</td>
</tr>
<tr>
<td>sign$(t)$</td>
<td>$\frac{2}{\pi} \arcsin (\angle)$</td>
</tr>
<tr>
<td>$\varsigma_2 t^2 + \varsigma_1 t + \varsigma_0$</td>
<td>$\varsigma_2^2 (2 (x_i^\top x_j)^2 + |x_i|^2 |x_j|^2) + \varsigma_1^2 x_i^\top x_j + \varsigma_2 \varsigma_0 (|x_i|^2 + |x_j|^2) + \varsigma_0^2$</td>
</tr>
<tr>
<td>$\cos(t)$</td>
<td>(\exp \left( -\frac{1}{2} \left( |x_i|^2 + |x_j|^2 \right) \right) \cosh(x_i^\top x_j))</td>
</tr>
<tr>
<td>$\sin(t)$</td>
<td>(\exp \left( -\frac{1}{2} \left( |x_i|^2 + |x_j|^2 \right) \right) \sinh(x_i^\top x_j))</td>
</tr>
<tr>
<td>$\text{erf}(t)$</td>
<td>(\frac{2}{\pi} \arcsin \left( \frac{2x_i^\top x_j}{\sqrt{(1+2|x_i|^2)(1+2|x_j|^2)}} \right))</td>
</tr>
<tr>
<td>$\exp(-\frac{t^2}{2})$</td>
<td>(\frac{1}{\sqrt{(1+|x_i|^2)(1+|x_j|^2)-(x_i^\top x_j)^2}})</td>
</tr>
</tbody>
</table>

$\Rightarrow$ (still) highly nonlinear functions of the data $x$!
Dig Deeper into $K$

Data: $K$-class Gaussian mixture model

$x_i \in C_a \iff x_i = \mu_a / \sqrt{p} + z_i$

with $z_i \sim \mathcal{N}(0, C_a/p)$, $a = 1, \ldots, K$ of statistical mean $\mu_a$ and covariance $C_a$.

Non-trivial classification (again)

For $p$ large, $\|\mu_a - \mu_b\| = O(1)$, $\|C_a\| = O(1)$ and $\text{tr}(C_a - C_b) = O(\sqrt{p})$.

As a consequence,

$$\|x_i\|^2 = \|z_i\|^2 + \underbrace{\|\mu_a\|^2/p}_{O(1)} + \underbrace{2\mu_a^T z_i/\sqrt{p}}_{O(p^{-1})}$$

$$= \underbrace{\text{tr} C_a/p}_{O(1)} + \|z_i\|^2 - \underbrace{\text{tr} C_a/p}_{O(p^{-1/2})} + \underbrace{\|\mu_a\|^2/p}_{O(p^{-1})} + \underbrace{2\mu_a^T z_i/\sqrt{p}}_{O(p^{-1})}$$

Then for $C^\circ = \sum_{a=1}^{K} \frac{n_a}{n} C_a$ and $C_a = C^\circ + C^\circ$ for $a = 1, \ldots, K$,

$$\Rightarrow \|x_i\|^2 = \tau + O(p^{-1/2}) \text{ with } \tau \equiv \text{tr}(C^\circ)/p,$$  

$$\|x_i - x_j\|^2 \approx 2\tau!$$
Asymptotic Equivalent of $K$

For all $\sigma(\cdot)$ listed in the table above, we have, as $n \sim p \to \infty$,

$$\|K - \tilde{K}\| \to 0$$

almost surely, with

$$\tilde{K} \equiv d_1 \left( Z + M \frac{J^T}{\sqrt{p}} \right)^T \left( Z + M \frac{J^T}{\sqrt{p}} \right) + d_2 U B U^T + d_0 I_n$$

and

$$U \equiv \left[ \frac{J}{\sqrt{p}}, \phi \right], \quad B \equiv \left[ t t^T + 2S \quad t \right].$$

$J \equiv [j_1, \ldots, j_K]$, $j_a$ canonical vector of $C_a$, weighted by $z$, $\phi$ random fluctuations of data and $M \equiv [\mu_1, \ldots, \mu_K]$, $t \equiv \left\{ \text{tr} \frac{C_a^o}{\sqrt{p}} \right\}_{a=1}^K$, $S \equiv \left\{ \text{tr}(C_a C_b) / p \right\}_{a,b=1}^K$ the statistical information.

---

**Table:** Coefficients $d_i$ in $\tilde{K}$ for different $\sigma(\cdot)$.

<table>
<thead>
<tr>
<th>$\sigma(t)$</th>
<th>$d_1$</th>
<th>$d_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t$</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$\max(t, 0)$</td>
<td>$\frac{1}{4}$</td>
<td>$\frac{1}{8\pi\tau}$</td>
</tr>
<tr>
<td>$</td>
<td>t</td>
<td>$</td>
</tr>
<tr>
<td>$1_{t&gt;0}$</td>
<td>$\frac{1}{2\pi\tau}$</td>
<td>0</td>
</tr>
<tr>
<td>$\text{sign}(t)$</td>
<td>$\frac{2}{\pi\tau}$</td>
<td>0</td>
</tr>
<tr>
<td>$\varsigma_2 t^2 + \varsigma_1 t + \varsigma_0$</td>
<td>$\varsigma_2$</td>
<td>$\varsigma_2$</td>
</tr>
<tr>
<td>$\cos(t)$</td>
<td>0</td>
<td>$e^{-\tau}$</td>
</tr>
<tr>
<td>$\sin(t)$</td>
<td>$0$</td>
<td>0</td>
</tr>
<tr>
<td>$\text{erf}(t)$</td>
<td>$\frac{4}{\pi} \frac{1}{2\tau+1}$</td>
<td>0</td>
</tr>
<tr>
<td>$\exp(-\frac{t^2}{2})$</td>
<td>0</td>
<td>$\frac{1}{4(\tau+1)^3}$</td>
</tr>
</tbody>
</table>
Table: Coefficients $d_i$ in $\tilde{K}$ for different $\sigma(\cdot)$.

<table>
<thead>
<tr>
<th>$\sigma(t)$</th>
<th>$d_1$</th>
<th>$d_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t$</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$\max(t, 0)$</td>
<td>$\frac{1}{4}$</td>
<td>$\frac{1}{8\pi \tau}$</td>
</tr>
<tr>
<td>$</td>
<td>t</td>
<td>$</td>
</tr>
<tr>
<td>$1_{t&gt;0}$</td>
<td>$\frac{1}{2\pi \tau}$</td>
<td>0</td>
</tr>
<tr>
<td>$\text{sign}(t)$</td>
<td>$\frac{2}{\pi \tau}$</td>
<td>0</td>
</tr>
<tr>
<td>$\varsigma_2 t^2 + \varsigma_1 t + \varsigma_0$</td>
<td>$\varsigma_1^2$</td>
<td>$\varsigma_2^2$</td>
</tr>
<tr>
<td>$\cos(t)$</td>
<td>0</td>
<td>$e^{-\tau} - \frac{e^{-\tau}}{4}$</td>
</tr>
<tr>
<td>$\sin(t)$</td>
<td>$e^{-\tau}$</td>
<td>0</td>
</tr>
<tr>
<td>$\text{erf}(t)$</td>
<td>$\frac{4}{\pi} \frac{1}{2\tau + 1}$</td>
<td>0</td>
</tr>
<tr>
<td>$\exp(-\frac{t^2}{2})$</td>
<td>0</td>
<td>$\frac{1}{4(\tau + 1)^3}$</td>
</tr>
</tbody>
</table>

A natural classification of $\sigma(\cdot)$:

- **mean-oriented**, $d_1 \neq 0$, $d_2 = 0$:
  - $t$, $1_{t>0}$, $\text{sign}(t)$, $\sin(t)$ and $\text{erf}(t)$
  - $\Rightarrow$ separate with difference in $M$;

- **cov-oriented**, $d_1 = 0$, $d_2 \neq 0$:
  - $|t|$, $\cos(t)$ and $\exp(-t^2/2)$
  - $\Rightarrow$ track differences in cov $t$, $S$;

- **“balanced”**, both $d_1, d_2 \neq 0$:
  - $\Rightarrow$ make use of both statistics!

$\mathbf{\Rightarrow}$ make use of both statistics!
Numerical Validations: Gaussian Data

**Example**: Gaussian mixture data of four classes: $\mathcal{N}(\mu_1, \mathbf{C}_1)$, $\mathcal{N}(\mu_1, \mathbf{C}_2)$, $\mathcal{N}(\mu_2, \mathbf{C}_1)$ and $\mathcal{N}(\mu_2, \mathbf{C}_2)$ with different $\sigma(\cdot)$ functions.

**Case 1**: linear map $\sigma(t) = t$.

**Case 2**: $\sigma(t) = |t|$.
**Numerical Validations: Gaussian Data**

**Case 3:** the ReLU function $\sigma(t) = \max(t, 0)$.

---

**Eigenvector 1**

---

**Eigenvector 2**

---

**Eigenvector 2**

---

**Eigenvector 1**
Numerical Validations: Real Datasets

Figure: The MNIST image database.

Figure: The epileptic EEG datasets.\(^1\)

Codes available at https://github.com/Zhenyu-LIAO/RMT4RFM.

\(^1\)http://www.meb.unibonn.de/epileptologie/science/physik/eegdata.html.
Numerical Validations: Real Datasets

Table: Empirical estimation of statistical information of the MNIST and EEG datasets.

<table>
<thead>
<tr>
<th></th>
<th>(| M^T M |)</th>
<th>(| t t^T + 2S |)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNIST data</td>
<td>172.4</td>
<td>86.0</td>
</tr>
<tr>
<td>EEG data</td>
<td>1.2</td>
<td>182.7</td>
</tr>
</tbody>
</table>

Table: Clustering accuracies on MNIST.

<table>
<thead>
<tr>
<th>(\sigma(t))</th>
<th>(n = 64)</th>
<th>(n = 128)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(t)</td>
<td>88.94%</td>
<td>87.30%</td>
</tr>
<tr>
<td>(1_{t&gt;0})</td>
<td>82.94%</td>
<td>85.56%</td>
</tr>
<tr>
<td>(\text{sign}(t))</td>
<td>83.34%</td>
<td>85.22%</td>
</tr>
<tr>
<td>(\sin(t))</td>
<td>87.81%</td>
<td><strong>87.50%</strong></td>
</tr>
<tr>
<td>(\cos(t))</td>
<td>59.56%</td>
<td>57.72%</td>
</tr>
<tr>
<td>(\exp(-t^2/2))</td>
<td>60.44%</td>
<td>58.67%</td>
</tr>
<tr>
<td>balanced</td>
<td>85.72%</td>
<td>82.27%</td>
</tr>
</tbody>
</table>

Table: Clustering accuracies on EEG.

<table>
<thead>
<tr>
<th>(\sigma(t))</th>
<th>(n = 64)</th>
<th>(n = 128)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(t)</td>
<td>70.31%</td>
<td>69.58%</td>
</tr>
<tr>
<td>(1_{t&gt;0})</td>
<td>65.87%</td>
<td>63.47%</td>
</tr>
<tr>
<td>(\text{sign}(t))</td>
<td>64.63%</td>
<td>63.03%</td>
</tr>
<tr>
<td>(\sin(t))</td>
<td>70.34%</td>
<td>68.22%</td>
</tr>
<tr>
<td>(\cos(t))</td>
<td>99.38%</td>
<td>99.36%</td>
</tr>
<tr>
<td>(\exp(-t^2/2))</td>
<td><strong>99.81%</strong></td>
<td><strong>99.77%</strong></td>
</tr>
<tr>
<td>balanced</td>
<td><strong>87.91%</strong></td>
<td><strong>90.97%</strong></td>
</tr>
</tbody>
</table>
RMT often assumes $x_i$ are affine maps $A z_i + b$ of $z_i \in \mathbb{R}^p$ with i.i.d. entries.

Concentrated random vectors

For a certain family of functions $f : \mathbb{R}^p \mapsto \mathbb{R}$, there exists deterministic $m_f \in \mathbb{R}$

$$P (|f(x) - m_f| > \epsilon) \leq e^{-g(\epsilon)}, \quad \text{for some strictly increasing function } g. \quad (5)$$

The theory remains valid for concentrated random vectors! But ... so what?
From concentrated random vectors to GANs

Figure: Illustration of a generative adversarial network (GAN).

Figure: Images samples generated by BigGAN (Brock et al., 2018).
Take-away message and perspectives

Take-away messages:
- loss of relevance of Euclidean distance for large dimensional data
- Taylor expansion helps understand kernel spectral clustering and simple random neural nets behavior
- go beyond Gaussian or i.i.d. random vectors with concentrated random vector

Even more question:
- what can we do if Taylor expansion is not possible?
- universality? influence of higher order moments?
- more involved systems, e.g., deep neural nets?

And much more to be done!
- neural nets: loss landscape, gradient descent dynamics
- problems from convex optimization (often of implicit solution)
- more difficult: non-convex optimization problems
- transfer learning, active learning, generative networks (GANs)
- robust statistics in machine learning
- ...
Summary of Results and Perspectives

Kernel Methods: References


Summary of Results and Perspectives

Neural Networks: References


Thank you!

For more information, please visit

- https://zhenyu-liao.github.io;