Can Random Matrices Change the Future of Machine Learning?

MASCOT PhD student 2020 Meeting

Romain COUILLET

CentraleSupélec, L2S, University of Paris-Saclay, France
GSTATS IDEX DataScience Chair, GIPSA-lab, University Grenoble–Alpes, France.

September 15, 2020
A long story short...
A long story short...
A long story short...
A long story short...
A long story short...
A long story short...
A long story short...

- Gaussian mixtures
- Algorithms, heuristics
- Random matrix theory
- Performances
  - Real Data

[Numbers and data points]

[Images of data distributions]
A long story short...
A long story short...

Performances
Synthetic Data: ✔️
Real Data: ☹️

Algorithms, heuristics

Random matrix theory

Concentration of measure
Gaussian mixtures
A long story short...

<table>
<thead>
<tr>
<th>Performances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synthetic Data</td>
</tr>
</tbody>
</table>

- Algorithms, heuristics

- Concentration of measure

- Random matrix theory

\[
\begin{bmatrix}
1 & 8 & 13 & 12 \\
14 & 11 & 2 & 7 \\
4 & 5 & 16 & 9 \\
15 & 10 & 3 & 6
\end{bmatrix}
\]
A long story short...

Performances
Synthetic Data
Real Data

 Algorithms, heuristics

Concentration of measure

Random matrix theory

[1 8 13 12
 14 11 2 7
 4 5 16 9
 15 10 3 6]
Outline

Basics of Random Matrix Theory
  Motivation: Large Sample Covariance Matrices
  Spiked Models

Application to Machine Learning
Outline

Basics of Random Matrix Theory
- Motivation: Large Sample Covariance Matrices
- Spiked Models

Application to Machine Learning
Outline

Basics of Random Matrix Theory
  Motivation: Large Sample Covariance Matrices
    Spiked Models

Application to Machine Learning
**Context**

**Baseline scenario:** \( y_1, \ldots, y_n \in \mathbb{C}^p \) (or \( \mathbb{R}^p \)) i.i.d. with \( E[y_1] = 0 \), \( E[y_1 y_1^*] = C_p \):
Context

**Baseline scenario:** $y_1, \ldots, y_n \in \mathbb{C}^p$ (or $\mathbb{R}^p$) i.i.d. with $E[y_1] = 0$, $E[y_1 y_1^*] = C_p$:

- If $y_1 \sim \mathcal{N}(0, C_p)$, ML estimator for $C_p$ is the sample covariance matrix (SCM)

$$\hat{C}_p = \frac{1}{n} Y_p Y_p^* = \frac{1}{n} \sum_{i=1}^{n} y_i y_i^*$$

($Y_p = [y_1, \ldots, y_n] \in \mathbb{C}^{p \times n}$).
**Context**

**Baseline scenario:** $y_1, \ldots, y_n \in \mathbb{C}^p$ (or $\mathbb{R}^p$) i.i.d. with $E[y_1] = 0$, $E[y_1 y_1^*] = C_p$:

- If $y_1 \sim \mathcal{N}(0, C_p)$, ML estimator for $C_p$ is the sample covariance matrix (SCM)

$$
\hat{C}_p = \frac{1}{n} Y_p Y_p^* = \frac{1}{n} \sum_{i=1}^{n} y_i y_i^*
$$

($Y_p = [y_1, \ldots, y_n] \in \mathbb{C}^{p \times n}$).

- If $n \to \infty$, then, strong law of large numbers

$$
\hat{C}_p \xrightarrow{a.s.} C_p.
$$

or equivalently, in spectral norm

$$
\| \hat{C}_p - C_p \| \xrightarrow{a.s.} 0.
$$
Context

**Baseline scenario:** \( y_1, \ldots, y_n \in \mathbb{C}^p \) (or \( \mathbb{R}^p \)) i.i.d. with \( E[y_1] = 0, E[y_1 y_1^*] = C_p \):

- If \( y_1 \sim \mathcal{N}(0, C_p) \), ML estimator for \( C_p \) is the sample covariance matrix (SCM)
  \[
  \hat{C}_p = \frac{1}{n} Y_p Y_p^* = \frac{1}{n} \sum_{i=1}^{n} y_i y_i^*
  \]
  (\( Y_p = [y_1, \ldots, y_n] \in \mathbb{C}^{p \times n} \)).

- If \( n \to \infty \), then, **strong law of large numbers**
  \[
  \hat{C}_p \xrightarrow{a.s.} C_p.
  \]
  or equivalently, **in spectral norm**
  \[
  \|\hat{C}_p - C_p\| \xrightarrow{a.s.} 0.
  \]

**Random Matrix Regime**

- No longer valid if \( p, n \to \infty \) with \( p/n \to c \in (0, \infty) \),
  \[
  \|\hat{C}_p - C_p\| \not\to 0.
  \]
Context

**Baseline scenario**: \( y_1, \ldots, y_n \in \mathbb{C}^p \) (or \( \mathbb{R}^p \)) i.i.d. with \( E[y_1] = 0, \ E[y_1 y_1^*] = C_p \):

- If \( y_1 \sim \mathcal{N}(0, C_p) \), ML estimator for \( C_p \) is the sample covariance matrix (SCM)

\[
\hat{C}_p = \frac{1}{n} Y_p Y_p^* = \frac{1}{n} \sum_{i=1}^{n} y_i y_i^*
\]

\( (Y_p = [y_1, \ldots, y_n] \in \mathbb{C}^{p \times n}) \).

- If \( n \to \infty \), then, strong law of large numbers

\[
\hat{C}_p \xrightarrow{a.s.} C_p.
\]

or equivalently, in spectral norm

\[
\left\| \hat{C}_p - C_p \right\| \xrightarrow{a.s.} 0.
\]

**Random Matrix Regime**

- No longer valid if \( p, n \to \infty \) with \( p/n \to c \in (0, \infty) \),

\[
\left\| \hat{C}_p - C_p \right\| \not\to 0.
\]

- For practical \( p, n \) with \( p \approx n \), leads to dramatically wrong conclusions
Context

**Baseline scenario:** $y_1, \ldots, y_n \in \mathbb{C}^p$ (or $\mathbb{R}^p$) i.i.d. with $E[y_1] = 0$, $E[y_1y_1^*] = C_p$:

- If $y_1 \sim \mathcal{N}(0, C_p)$, ML estimator for $C_p$ is the sample covariance matrix (SCM)
  \[
  \hat{C}_p = \frac{1}{n} Y_p Y_p^* = \frac{1}{n} \sum_{i=1}^{n} y_i y_i^*
  \]

  ($Y_p = [y_1, \ldots, y_n] \in \mathbb{C}^{p \times n}$).

- If $n \to \infty$, then, strong law of large numbers
  \[
  \hat{C}_p \xrightarrow{a.s.} C_p.
  \]

  or equivalently, in spectral norm
  \[
  \|\hat{C}_p - C_p\| \xrightarrow{a.s.} 0.
  \]

**Random Matrix Regime**

- No longer valid if $p, n \to \infty$ with $p/n \to c \in (0, \infty)$,
  \[
  \|\hat{C}_p - C_p\| \not\to 0.
  \]

- For practical $p, n$ with $p \simeq n$, leads to dramatically wrong conclusions
- **Even for** $p = n/100$.  


The Marčenko–Pastur law

**Figure:** Histogram of the eigenvalues of $\hat{C}_p$ for $c = 1/4$, $C_p = I_p$. 
The Marčenko–Pastur law

Figure: Histogram of the eigenvalues of $\hat{C}_p$ for $c = 1/4$, $C_p = I_p$. 

Density of eigenvalues for $p = 100$, $n = 400$. 

Eigenvalues of $\hat{C}_p$.
The Marčenko–Pastur law

Figure: Histogram of the eigenvalues of $\hat{C}_p$ for $c = 1/4$, $C_p = I_p$. 

Density

Eigenvalues of $\hat{C}_p$
The Marčenko–Pastur law

Figure: Histogram of the eigenvalues of $\hat{C}_p$ for $c = 1/4$, $C_p = I_p$. 
The Marčenko–Pastur law

Figure: Histogram of the eigenvalues of $\hat{C}_p$ for $c = 1/4$, $C_p = I_p$. 
The Marčenko–Pastur law

Figure: Histogram of the eigenvalues of $\hat{C}_p$ for $c = 1/4$, $C_p = I_p$. 
The Marčenko–Pastur law

**Definition (Empirical Spectral Density)**

Empirical spectral density (e.s.d.) $\mu_p$ of Hermitian matrix $A_p \in \mathbb{C}^{p \times p}$ is

$$
\mu_p = \frac{1}{p} \sum_{i=1}^{p} \delta_{\lambda_i(A_p)}.
$$
The Marčenko–Pastur law

Definition (Empirical Spectral Density)
Empirical spectral density (e.s.d.) $\mu_p$ of Hermitian matrix $A_p \in \mathbb{C}^{p \times p}$ is

$$\mu_p = \frac{1}{p} \sum_{i=1}^{p} \delta_{\lambda_i(A_p)}.$$ 

Theorem (Marčenko–Pastur Law [Marčenko, Pastur’67])
$X_p \in \mathbb{C}^{p \times n}$ with i.i.d. zero mean, unit variance entries. As $p, n \to \infty$ with $p/n \to c \in (0, \infty)$, e.s.d. $\mu_p$ of $\frac{1}{n} X_p X_p^*$ satisfies

$$\mu_p \xrightarrow{\text{a.s.}} \mu_c$$

weakly, where

$\mu_c(\{0\}) = \max\{0, 1 - c^{-1}\}$

$\mu_c(\{0\}) = \max\{0, 1 - c^{-1}\}$
The Marčenko–Pastur law

Definition (Empirical Spectral Density)
Empirical spectral density (e.s.d.) $\mu_p$ of Hermitian matrix $A_p \in \mathbb{C}^{p \times p}$ is

$$
\mu_p = \frac{1}{p} \sum_{i=1}^{p} \delta_{\lambda_i(A_p)}.
$$

Theorem (Marčenko–Pastur Law [Marčenko, Pastur’67])
$X_p \in \mathbb{C}^{p \times n}$ with i.i.d. zero mean, unit variance entries.
As $p, n \to \infty$ with $p/n \to c \in (0, \infty)$, e.s.d. $\mu_p$ of $\frac{1}{n} X_p X_p^*$ satisfies

$$
\mu_p \xrightarrow{\text{a.s.}} \mu_c
$$

weakly, where

- $\mu_c(\{0\}) = \max\{0, 1 - c^{-1}\}$
- on $(0, \infty)$, $\mu_c$ has continuous density $f_c$ supported on $[(1 - \sqrt{c})^2, (1 + \sqrt{c})^2]$

$$
f_c(x) = \frac{1}{2\pi c x} \sqrt{(x - (1 - \sqrt{c})^2)((1 + \sqrt{c})^2 - x)}. 
$$
The Marčenko–Pastur law

Figure: Marčenko-Pastur law for different limit ratios $c = \lim_{p \to \infty} p/n$. 
The Marčenko–Pastur law

Figure: Marčenko–Pastur law for different limit ratios $c = \lim_{p \to \infty} p/n$. 
The Marčenko–Pastur law

Figure: Marčenko-Pastur law for different limit ratios $c = \lim_{p \to \infty} p/n$. 
Outline

Basics of Random Matrix Theory
  Motivation: Large Sample Covariance Matrices
  Spiked Models

Application to Machine Learning
**Spiked Models**

**Small rank perturbation:** $C_p = I_p + P$, $P$ of low rank.

**Figure:** Eigenvalues of $\frac{1}{n}Y_p Y_p^T$, $\text{eig}(C_p) = \{1, \ldots, 1, 2, 3, 4, 5\}$. 

$p/n = 1/4 (p = 500)$
Spiked Models

Small rank perturbation: $C_p = I_p + P$, $P$ of low rank.

Figure: Eigenvalues of $\frac{1}{n} Y_p Y_p^T$, $\text{eig}(C_p) = \{1, \ldots, 1, 2, 3, 4, 5\}$.
Spiked Models

**Small rank perturbation:** $C_p = I_p + P$, $P$ of low rank.

Figure: Eigenvalues of $\frac{1}{n} Y_p Y_p^T$, $\text{eig}(C_p) = \{1, \ldots, 1, 2, 3, 4, 5\}$. 

$p/n = 1 (p = 500)$
Small rank perturbation: $C_p = I_p + P$, $P$ of low rank.

Figure: Eigenvalues of $\frac{1}{n} Y_p Y_p^T$, eig($C_p$) = \{1, \ldots, 1, 2, 3, 4, 5\}.

$p/n = 2$ ($p = 500$)
Spiked Models

Theorem (Eigenvalues [Baik, Silverstein’06])

Let $Y_p = C_p^{1/2} X_p$, with

$\triangleright$ $X_p$ with i.i.d. zero mean, unit variance, $E[|X_p|_i^4] < \infty$.

$\triangleright$ $C_p = I_p + P$, $P = U \Omega U^*$, where, for $K$ fixed,

$$\Omega = \text{diag} (\omega_1, \ldots, \omega_K) \in \mathbb{R}^{K \times K}, \text{ with } \omega_1 \geq \ldots \geq \omega_K > 0.$$
Theorem (Eigenvalues [Baik, Silverstein’06])

Let \( Y_p = C_p^{\frac{1}{2}} X_p \), with

\[ X_p \text{ with i.i.d. zero mean, unit variance, } E[|X_p|_4^4] < \infty. \]

\[ C_p = I_p + P, \quad P = U \Omega U^*, \text{ where, for } K \text{ fixed,} \]

\[ \Omega = \text{diag}(\omega_1, \ldots, \omega_K) \in \mathbb{R}^{K \times K}, \text{ with } \omega_1 \geq \ldots \geq \omega_K > 0. \]

Then, as \( p, n \to \infty, p/n \to c \in (0, \infty) \), denoting \( \lambda_m = \lambda_m(\frac{1}{n}Y_pY_p^*) (\lambda_m > \lambda_{m+1}) \),

\[ \lambda_m \xrightarrow{\text{a.s.}} \left\{ \begin{array}{ll} 1 + \omega_m + c \frac{1+\omega_m}{\omega_m} > (1 + \sqrt{c})^2 & , \omega_m > \sqrt{c} \\ (1 + \sqrt{c})^2 & , \omega_m \in (0, \sqrt{c}] \end{array} \right. \]
Spiked Models

Theorem (Eigenvectors [Paul’07])

Let $Y_p = C_p^{1/2} X_p$, with

- $X_p$ with i.i.d. zero mean, unit variance, $E[|X_p|_{ij}^4] < \infty$.
- $C_p = I_p + P$, $P = U\Omega U^* = \sum_{i=1}^K \omega_i u_i u_i^*$, $\omega_1 > \ldots > \omega_M > 0$. 

Then, as $p,n \to \infty$, $p/n \to c \in (0,\infty)$, for $a,b \in \mathbb{C}$ deterministic and $\hat{u}_i$ eigenvector of $\lambda_i(\frac{1}{n} Y_p Y_p^*)$, $a^* \hat{u}_i \hat{u}_i^* b - 1 - c \omega_{i-2} + c \omega_{i-1} \cdot 1 \omega_i > \sqrt{c} \to 0$.

In particular, $|\hat{u}_i^* a|_2 \to 1 - c \omega_{i-1} \cdot 1 \omega_i > \sqrt{c}$. 

s.$ \to 0$
Spiked Models

Theorem (Eigenvectors [Paul’07])

Let $Y_p = C_p^{1/2} X_p$, with

- $X_p$ with i.i.d. zero mean, unit variance, $E[|X_p|^4_{i,j}] < \infty$.
- $C_p = I_p + P$, $P = U\Omega U^* = \sum_{i=1}^K \omega_i u_i u_i^*$, $\omega_1 > \ldots > \omega_M > 0$.

Then, as $p, n \to \infty$, $p/n \to c \in (0, \infty)$, for $a, b \in \mathbb{C}^p$ deterministic and $\hat{u}_i$ eigenvector of $\lambda_i(\frac{1}{n}Y_p Y_p^*)$,

$$a^*\hat{u}_i \hat{u}_i^* b - \frac{1 - c\omega_i^{-2}}{1 + c\omega_i^{-1}} a^* u_i u_i^* b \cdot 1_{\omega_i > \sqrt{c}} \overset{a.s.}{\longrightarrow} 0$$

In particular,

$$|\hat{u}_i^* u_i|^2 \overset{a.s.}{\longrightarrow} \frac{1 - c\omega_i^{-2}}{1 + c\omega_i^{-1}} \cdot 1_{\omega_i > \sqrt{c}}.$$
Figure: Simulated versus limiting $|\hat{u}_1^T u_1|^2$ for $Y_p = C_p^{1/2} X_p$, $C_p = I_p + \omega_1 u_1 u_1^T$, $p/n = 1/3$, varying $\omega_1$. 
Spiked Models

Figure: Simulated versus limiting $|\hat{u}_1^T u_1|^2$ for $Y_p = C_p^{\frac{1}{2}} X_p$, $C_p = I_p + \omega_1 u_1 u_1^T$, $p/n = 1/3$, varying $\omega_1$. 

$\mathbb{E}_\omega (\mathbb{E}_X |\hat{u}_1^T u_1|^2) \approx |\hat{u}_1^T u_1|^2$ for $\omega \to \omega_1$.
Spiked Models

Figure: Simulated versus limiting $|\hat{u}_1^T u_1|^2$ for $Y_p = C_p^{1/2} X_p$, $C_p = I_p + \omega_1 u_1 u_1^T$, $p/n = 1/3$, varying $\omega_1$. 
Figure: Simulated versus limiting $|\hat{u}_1 u_1|^2$ for $Y_p = C_p^{1/2} X_p$, $C_p = I_p + \omega_1 u_1 u_1^T$, $p/n = 1/3$, varying $\omega_1$. 
Similar results for multiple matrix models:

- \( Y_p = \frac{1}{n} (I + P)^{\frac{1}{2}} X_p X_p^* (I + P)^{\frac{1}{2}} \)
- \( Y_p = \frac{1}{n} X_p X_p^* + P \)
- \( Y_p = \frac{1}{n} X_p^* (I + P) X \)
- \( Y_p = \frac{1}{n} (X_p + P)^* (X_p + P) \)
- etc.
Outline

Basics of Random Matrix Theory
  Motivation: Large Sample Covariance Matrices
  Spiked Models

Application to Machine Learning
Takeaway Message 1

“RMT Explains Why Machine Learning Intuitions Collapse in Large Dimensions”
The curse of dimensionality and its consequences

Clustering setting in (not so) large $n, p$:

**GMM setting:**

$x(a) \sim N(\mu(a), C_a), a = 1, \ldots, k$

**Non-trivial task:**

$\|\mu(a) - \mu(b)\| = O(1), \text{tr}(C_a - C_b) = O(\sqrt{p}), \text{tr}[(C_a - C_b)^2] = O(p)$

**Classical method: spectral clustering**

Extract and cluster the dominant eigenvectors of $K = \{\kappa(x_i, x_j)\}_{n \times n}$, $\kappa(x_i, x_j) = f(\frac{1}{p} \|x_i - x_j\|^2)$.

**Why?** Finite-dimensional intuition
The curse of dimensionality and its consequences

Clustering setting in (not so) large \( n, p \):

- GMM setting: \( x_1^{(a)}, \ldots, x_{n_a}^{(a)} \sim \mathcal{N}(\mu_a, C_a), \ a = 1, \ldots, k \)
The curse of dimensionality and its consequences

Clustering setting in (not so) large $n, p$:

- **GMM setting:** $x_1^{(a)}, \ldots, x_n^{(a)} \sim \mathcal{N}(\mu_a, C_a), a = 1, \ldots, k$
- **Non-trivial task:**

\[ \|\mu_a - \mu_b\| = O(1), \quad \text{tr} (C_a - C_b) = O(\sqrt{p}), \quad \text{tr} [(C_a - C_b)^2] = O(p) \]
The curse of dimensionality and its consequences

### Clustering setting in (not so) large $n, p$:

- **GMM setting:** $x_1^{(a)}, \ldots, x_n^{(a)} \sim \mathcal{N}(\mu_a, C_a), \ a = 1, \ldots, k$

- **Non-trivial task:**
  
  $$\|\mu_a - \mu_b\| = O(1), \quad \text{tr} (C_a - C_b) = O(\sqrt{p}), \quad \text{tr} [(C_a - C_b)^2] = O(p)$$

### Classical method: spectral clustering
The curse of dimensionality and its consequences

Clustering setting in (not so) large \( n, p \):

- GMM setting: \( x_1^{(a)}, \ldots, x_{n_a}^{(a)} \sim \mathcal{N}(\mu_a, C_a), \ a = 1, \ldots, k \)
- Non-trivial task:

\[
\|\mu_a - \mu_b\| = O(1), \quad \text{tr} (C_a - C_b) = O(\sqrt{p}), \quad \text{tr} [(C_a - C_b)^2] = O(p)
\]

Classical method: spectral clustering

- Extract and cluster the dominant eigenvectors of

\[
K = \{\kappa(x_i, x_j)\}_{i,j=1}^{n}
\]
The curse of dimensionality and its consequences

Clustering setting in (not so) large $n, p$:

- GMM setting: $x^{(a)}_1, \ldots, x^{(a)}_{n_a} \sim \mathcal{N}(\mu_a, C_a), a = 1, \ldots, k$
- Non-trivial task:

\[
\|\mu_a - \mu_b\| = O(1), \quad \text{tr} (C_a - C_b) = O(\sqrt{p}), \quad \text{tr} [(C_a - C_b)^2] = O(p)
\]

Classical method: spectral clustering

- Extract and cluster the dominant eigenvectors of

\[
K = \{\kappa(x_i, x_j)\}_{i,j=1}^n, \quad \kappa(x_i, x_j) = f \left( \frac{1}{p} \|x_i - x_j\|^2 \right).
\]
The curse of dimensionality and its consequences

Clustering setting in (not so) large $n, p$:

- **GMM setting**: $x^{(a)}_1, \ldots, x^{(a)}_{n_a} \sim \mathcal{N}(\mu_a, C_a), a = 1, \ldots, k$

- **Non-trivial task**:

  $$\|\mu_a - \mu_b\| = O(1), \quad \text{tr}(C_a - C_b) = O(\sqrt{p}), \quad \text{tr}[(C_a - C_b)^2] = O(p)$$

Classical method: spectral clustering

- **Extract and cluster the dominant eigenvectors of**

  $$K = \left\{ \kappa(x_i, x_j) \right\}_{i,j=1}^n, \quad \kappa(x_i, x_j) = f \left( \frac{1}{p} \| x_i - x_j \|^2 \right).$$

- **Why?** Finite-dimensional intuition

  $$K = \begin{pmatrix}
  \kappa(x_i, x_j) & \kappa(x_i, x_j) & \kappa(x_i, x_j) \\
  \ll 1 & \ll 1 & \ll 1 \\
  \kappa(x_i, x_j) & \kappa(x_i, x_j) & \kappa(x_i, x_j) \\
  \ll 1 & \gg 1 & \ll 1 \\
  \kappa(x_i, x_j) & \kappa(x_i, x_j) & \kappa(x_i, x_j) \\
  \ll 1 & \ll 1 & \gg 1 \\
  \end{pmatrix}$$
The curse of dimensionality and its consequences (2)

In reality, here is what happens...

Kernel $K_{ij} = \exp\left(-\frac{1}{2p} \|x_i - x_j\|^2\right)$ and second eigenvector $v_2$

$(x_i \sim \mathcal{N}(\pm \mu, I_p), \mu = (2, 0, \ldots, 0)^T \in \mathbb{R}^p)$. 
In reality, here is what happens...

Kernel $K_{i,j} = \exp\left( -\frac{1}{2p} \|x_i - x_j\|^2 \right)$ and second eigenvector $v_2$

$(x_i \sim \mathcal{N}(\pm\mu, I_p), \mu = (2, 0, \ldots, 0)^T \in \mathbb{R}^p).$
In reality, here is what happens...

Kernel $K_{ij} = \exp\left(-\frac{1}{2p} \| x_i - x_j \|^2 \right)$ and second eigenvector $v_2$

$x_i \sim \mathcal{N}(\pm \mu, I_p)$, $\mu = (2, 0, \ldots, 0)^T \in \mathbb{R}^p$.

Key observation: Under growth rate assumptions, $\max_{1 \leq i \neq j \leq n} \{|x_i^T x_j|\} \rightarrow 0$, $\tau = 2p^k \sum_{i=1}^{n} \text{tr}(C_t)$.

$\Rightarrow$ this suggests $K \approx f(\tau/n)$. 

$p = 4, n = 1000$

$K = \begin{pmatrix} p & n & 1000 \\ & & \end{pmatrix}$

$v_2 = \begin{pmatrix} \vdots \\ \vdots \end{pmatrix}$

$p = 400, n = 1000$

$K = \begin{pmatrix} p & n & 1000 \\ & & \end{pmatrix}$

$v_2 = \begin{pmatrix} \vdots \\ \vdots \end{pmatrix}$
In reality, here is what happens...

Kernel \( K_{ij} = \exp(-\frac{1}{2p} \|x_i - x_j\|^2) \) and second eigenvector \( v_2 \)

\((x_i \sim \mathcal{N}(\pm \mu, I_p), \mu = (2, 0, \ldots, 0)^T \in \mathbb{R}^p)\).

Key observation: Under growth rate assumptions,

\[
\max_{1 \leq i \neq j \leq n} \left\{ \left| \frac{1}{p} \|x_i - x_j\|^2 - \tau \right| \right\} \xrightarrow{a.s.} 0, \quad \tau = \frac{2}{p} \sum_{i=1}^{k} \text{tr} \frac{n_a}{n} C_a.
\]
The curse of dimensionality and its consequences (2)

In reality, here is what happens...

Kernel $K_{ij} = \exp\left(-\frac{1}{2p} \|x_i - x_j\|^2\right)$ and second eigenvector $v_2$

$(x_i \sim \mathcal{N}(\pm \mu, I_p), \mu = (2, 0, \ldots, 0)^T \in \mathbb{R}^p)$.

Key observation: Under growth rate assumptions,

\[
\max_{1 \leq i \neq j \leq n} \left\{ \left| \frac{1}{p} \| x_i - x_j \|^2 - \tau \right| \right\} \xrightarrow{a.s.} 0, \quad \tau = \frac{2}{p} \sum_{i=1}^{k} \text{tr} \frac{n_a}{n} C_a.
\]

- this suggests $K \sim f(\tau)1_n 1_n^T$!
The curse of dimensionality and its consequences (3)

MNIST
- raw
- \( p = 784, \ n = 500 \)

\[ \mathbf{K} = \begin{bmatrix} \text{Images} \end{bmatrix} \]

\[ \mathbf{v}_2 = \begin{bmatrix} 5 & 0 & 4 & 1 \end{bmatrix} \]

(ici, classes “5” et “0”)

ImageNet
- VGG-features
- \( p = 3084, \ n = 500 \)

\[ \mathbf{K} = \begin{bmatrix} \text{Images} \end{bmatrix} \]

\[ \mathbf{v}_2 = \begin{bmatrix} \text{bird} & \text{plane} & \text{car} \end{bmatrix} \]

(ici, classes “bird” et “plane”)

20NewsGroup
- BERT embedding
- \( p = 300, \ n = 500 \)

\[ \mathbf{K} = \begin{bmatrix} \text{Images} \end{bmatrix} \]

(ici, classes “sports” et “sales”)
(Major) consequences:

▶ Most **machine learning intuitions collapse**

Theorem ([C-Benaych'16]

Asymptotic Kernel Behavior)

Under growth rate assumptions, as $p, n \to \infty$,

$$\|K - \hat{K}\| \xrightarrow{a.s.} 0,$$

$$\hat{K} \approx f(\tau)^{1/n} T_n + O(\|\cdot\|(n)) + J A^T + *$$

with $J = [j_1, \ldots, j_k] \in \mathbb{R}^{n \times k}$,

$\mu_a - \mu_b$, $\text{tr}(C_a - C_b)$, $\text{tr}\left((C_a - C_b)^2\right)$, for $a, b \in \{1, \ldots, k\}$.

This is a spiked model! We can study it fully!
The curse of dimensionality and its consequences (4)

(Major) consequences:

- Most **machine learning intuitions collapse**
- **But luckily**, concentration of distances allows for Taylor expansion, linearization…

\[ \| K - \hat{K} \| \to 0, \quad \hat{K} \approx f(\tau) \frac{1}{n} T_n \quad \text{with} \quad J = [j_1, \ldots, j_k] \in \mathbb{R}^{n \times k}, \quad j_a = (0, \frac{1}{n} a, 0)^T \quad (the \ clusters!) \]

\[ \text{function of: } f(\tau), \quad f'(\tau), \quad f''(\tau) \]

\[ \| \mu_a - \mu_b \|, \quad \text{tr}(C_a - C_b), \quad \text{tr}((C_a - C_b)^2), \quad \text{for } a, b \in \{1, \ldots, k\} \]

This is a spiked model! We can study it fully!
(Major) consequences:

- Most **machine learning intuitions collapse**
- **But luckily**, concentration of distances allows for Taylor expansion, linearization...

**Theorem ([C-Benaych’16] Asymptotic Kernel Behavior)**

*Under growth rate assumptions, as \( p, n \to \infty \),*

\[
\| K - \hat{K} \| \xrightarrow{a.s.} 0, \quad \hat{K} \simeq \underbrace{f(\tau)1_n 1_n^T}_{O\|\cdot\|(n)}
\]
The curse of dimensionality and its consequences (4)

(Major) consequences:
▶ Most machine learning intuitions collapse
▶ But luckily, concentration of distances allows for Taylor expansion, linearization...

Theorem ([C-Benaych’16] Asymptotic Kernel Behavior)

*Under growth rate assumptions, as \( p, n \to \infty \),*

\[
\left\| K - \hat{K} \right\| \overset{a.s.}{\longrightarrow} 0, \quad \hat{K} \simeq f(\tau)1_n1_n^T + \frac{1}{p}ZZ^T + JAJ^T + * \quad O\|\cdot\|(n)
\]
The curse of dimensionality and its consequences (4)

(Major) consequences:

- Most **machine learning intuitions collapse**
- But luckily, concentration of distances allows for Taylor expansion, linearization...

Theorem ([C-Benaych’16] Asymptotic Kernel Behavior)

*Under growth rate assumptions, as $p, n \to \infty$,*

\[
\| K - \hat{K} \| \xrightarrow{a.s.} 0, \quad \hat{K} \simeq f(\tau)1_n1_n^T + \frac{1}{p} ZZ^T + JAJ^T + *
\]

\[
\text{with } J = [j_1, \ldots, j_k] \in \mathbb{R}^{n \times k}, j_a = (0, 1_{n_a}, 0)^T \text{ (the clusters!)}
\]
(Major) consequences:

- Most machine learning intuitions collapse
- But luckily, concentration of distances allows for Taylor expansion, linearization...

Theorem ([C-Benaych’16] Asymptotic Kernel Behavior)

Under growth rate assumptions, as \( p, n \to \infty \),

\[
\| K - \hat{K} \| \xrightarrow{a.s.} 0, \quad \hat{K} \sim \left\{ \begin{array}{c}
\frac{1}{p} ZZ^T + JAJ^T + * \\
O \| \cdot \| (n)
\end{array} \right. \\
\| K - \hat{K} \| = f(\tau)1_n1_n^T + \frac{1}{p} ZZ^T + JAJ^T + *
\]

with \( J = [j_1, \ldots, j_k] \in \mathbb{R}^{n \times k} \), \( j_a = (0, 1_{n_a}, 0)^T \) (the clusters!) and \( A \in \mathbb{R}^{k \times k} \)

function of:

- \( f(\tau), f'(\tau), f''(\tau) \)
- \( \| \mu_a - \mu_b \|, \text{tr}(C_a - C_b), \text{tr}((C_a - C_b)^2) \), for \( a, b \in \{1, \ldots, k\} \).
(Major) consequences:
▶ Most machine learning intuitions collapse
▶ But luckily, concentration of distances allows for Taylor expansion, linearization...

**Theorem ([C-Benaych’16] Asymptotic Kernel Behavior)**
*Under growth rate assumptions, as \( p, n \to \infty \),

\[
\left\| K - \hat{K} \right\| \xrightarrow{a.s.} 0, \quad \hat{K} \simeq f(\tau)1_n1_n^T + \frac{1}{p}ZZ^T + JAJ^T + \ast
\]

with \( J = [j_1, \ldots, j_k] \in \mathbb{R}^{n \times k}, j_a = (0, 1_{n_a}, 0)^T \) (the clusters!) and \( A \in \mathbb{R}^{k \times k} \)

function of:
▶ \( f(\tau), f'(\tau), f''(\tau) \)
▶ \( \| \mu_a - \mu_b \|, \text{tr} (C_a - C_b), \text{tr} ((C_a - C_b)^2) \), for \( a, b \in \{1, \ldots, k\} \).

\( \sim \) This is a spiked model! We can study it fully!
Performance prediction: spectral clustering

- Asymptotic analysis of eigenvectors of $K$: (MNIST, $p = 28 \times 28 (= 784)$)

\[
\begin{align*}
\mathbf{x}_1 &= \ldots, & \mathbf{x}_{64} &= \ldots \\
\mathbf{x}_{65} &= \ldots, & \mathbf{x}_{128} &= \ldots \\
\mathbf{x}_{129} &= \ldots, & \mathbf{x}_{192} &= \ldots
\end{align*}
\]

\[
\begin{align*}
\mathbf{v}_1 &= \begin{bmatrix}
\end{align*}
\]

\[
\begin{align*}
\mathbf{v}_2 &= \begin{bmatrix}
\end{align*}
\]

\[
\begin{align*}
\mathbf{v}_3 &= \begin{bmatrix}
\end{align*}
\]
Performance prediction: spectral clustering

- Asymptotic analysis of eigenvectors of $K$: (MNIST, $p = 28 \times 28 (= 784)$)

\[ \begin{align*}
\mathbf{x}_1 &= \ldots \mathbf{x}_{64} = \begin{array}{c}
\vdots \\
\end{array} \\
\mathbf{x}_{65} &= \ldots \mathbf{x}_{128} = \begin{array}{c}
\vdots \\
\end{array} \\
\mathbf{x}_{129} &= \ldots \mathbf{x}_{192} = \begin{array}{c}
\vdots \\
\end{array} \\
\end{align*} \]

\[ \begin{align*}
\mathbf{v}_1 &= \begin{bmatrix}
\vdots \\
\end{bmatrix} \\
\mathbf{v}_2 &= \begin{bmatrix}
\vdots \\
\end{bmatrix} \\
\mathbf{v}_3 &= \begin{bmatrix}
\vdots \\
\end{bmatrix} \\
\end{align*} \]
Performance prediction: spectral clustering

- Asymptotic analysis of eigenvectors of $K$: (MNIST, $p = 28 \times 28 (= 784)$)

$v_1 = \begin{bmatrix} \cdot \cdot \\ \cdot \cdot \end{bmatrix}$

$v_2 = \begin{bmatrix} \cdot \cdot \\ \cdot \cdot \end{bmatrix}$

$v_3 = \begin{bmatrix} \cdot \cdot \\ \cdot \cdot \end{bmatrix}$

Theoretical prediction
Takeaway Message 2

“RMT Reassesses and Improves Data Processing”
Improving Kernel Spectral Clustering

- Going further than ([Kammoun,Couillet’17]),

\[ K \simeq f(\tau)1_n1_n^T + f'(\tau)\frac{1}{p}ZZ^T + JAJ^T, \text{ avec } A = F \left( \begin{array}{c} f(\tau), f'(\tau), f''(\tau) \\ \|\mu_a - \mu_b\|, \text{tr}(C_a - C_b), \ldots \end{array} \right). \]
Improving Kernel Spectral Clustering

• Going further than ([Kammoun, Couillet’17]), if $f'(\tau) = 0$,

$$ K \approx f(\tau)1_n1_n^T + f'(\tau)1_p ZZ^T + JAJ^T, \quad \text{avec} \quad A = F \left( \frac{f(\tau), f'(\tau), f''(\tau)}{\|\mu_a - \mu_b\|, \text{tr}(C_a - C_b), \ldots} \right). $$
Improving Kernel Spectral Clustering

- Going further than ([Kammoun,Couillet’17]), if \( f'(\tau) = 0 \),

\[
K \simeq f(\tau)1_n1_n^T + f'(\tau)\frac{1}{p}ZZ^T + JAJ^T, \quad \text{avec} \quad A = F\left( \frac{f(\tau), f'(\tau), f''(\tau)}{\|\mu_a - \mu_b\|, \text{tr}(C_a - C_b), \ldots} \right).
\]

- **Gaussian case**: \( \mathcal{N}(0, C_1) \) vs. \( \mathcal{N}(0, C_2) \)

Kernel \( K_{ij} = \exp\left(-\frac{1}{2p}\|x_i - x_j\|^2\right) \)

Kernel \( K_{ij} = \left(\frac{1}{p}\|x_i - x_j\|^2 - \tau\right)^2 \)
Improving Kernel Spectral Clustering

- EEG data: sane vs. epileptic patients

Kernel \( K_{ij} = \exp\left(-\frac{1}{2p}\|x_i - x_j\|^2\right) \)

Kernel \( K_{ij} = \left(\frac{1}{p}\|x_i - x_j\|^2 - \tau\right)^2 \)

Remark: highly counter-intuitive kernel!
Improving Kernel Spectral Clustering

- **EEG data**: sane vs. epileptic patients

Kernel $K_{i,j} = \exp\left(-\frac{1}{2p} \|x_i - x_j\|^2\right)$

Kernel $K_{i,j} = \left(\frac{1}{p} \|x_i - x_j\|^2 - \tau\right)^2$

→ **Remark**: highly counter-intuitive kernel!
Another, more striking, example: Semi-supervised Learning

Semi-supervised learning: a great idea that never worked!
Semi-supervised learning: a great idea that never worked!

- **Setting**: assume now
  - \( x_1^{(a)}, \ldots, x_{n_a,[l]}^{(a)} \) already labelled (few),
  - \( x_{n_a,[l]+1}^{(a)}, \ldots, x_{n_a}^{(a)} \) unlabelled (a lot).
Another, more striking, example: Semi-supervised Learning

**Semi-supervised learning**: a great idea that never worked!

- **Setting**: assume now
  - \(x_1^{(a)}, \ldots, x_{n_a,[l]}^{(a)}\) already labelled (few),
  - \(x_{n_a,[l]+1}^{(a)}, \ldots, x_{n_a}^{(a)}\) unlabelled (a lot).

- **Machine Learning original idea**: find “scores” \(F_{ia}\) for \(x_i\) to belong to class \(a\)

\[
F = \arg\min_{F \in \mathbb{R}^{n \times k}} \sum_{a=1}^{k} \sum_{i,j} K_{ij} \left( F_{ia} - F_{ja} \right)^2, \quad F_{ia} = \delta\{x_i \in C_a\}.
\]
Another, more striking, example: Semi-supervised Learning

Semi-supervised learning: a great idea that never worked!

- **Setting**: assume now
  - $x^{(a)}_1, \ldots, x^{(a)}_{n_a,[l]}$ already labelled (few),
  - $x^{(a)}_{n_a,[l]+1}, \ldots, x^{(a)}_{n_a}$ unlabelled (a lot).

- **Machine Learning original idea**: find “scores” $F_{ia}$ for $x_i$ to belong to class $a$

$$F = \arg\min_{F \in \mathbb{R}^{n \times k}} \sum_{a=1}^{k} \sum_{i,j} K_{ij} \left( F_{ia} D_{ii}^\alpha - F_{ja} D_{jj}^\alpha \right)^2, \quad F^{[l]}_{ia} = \delta \{ x_i \in C_a \}. $$
Semi-supervised learning: a great idea that never worked!

- **Setting:** assume now
  - $x_1^{(a)}, \ldots, x_{n_a,[l]}^{(a)}$ already labelled (few),
  - $x_{n_a,[l]+1}^{(a)}, \ldots, x_{n_a}^{(a)}$ unlabelled (a lot).

- **Machine Learning original idea:** find “scores” $F_{ia}$ for $x_i$ to belong to class $a$

$$
F = \arg\min_{F \in \mathbb{R}^{n \times k}} \sum_{a=1}^{k} \sum_{i,j} K_{ij} \left( F_{ia} D_{ii}^\alpha - F_{ja} D_{jj}^\alpha \right)^2,
F_{[l]}^{[u]} = \delta\{x_i \in C_a\}.
$$

- **Explicit solution:**

$$
F^{[u]} = \left( I_{[u]}^{[u]} - D_{[u]}^{[u]} - \alpha K_{[uu]} D_{[uu]}^\alpha \right)^{-1} D_{[u]}^{[u]} - \alpha K_{[ul]} D_{[ul]}^\alpha F^{[l]}
$$

where $D = \text{diag}(K1_n)$ (degree matrix) and $[ul], [uu], \ldots$ blocks of labeled/unlabeled data.
The finite-dimensional case: What we expect

Figure: Outcome $\mathbf{F}$ of Laplacian algorithms ($\alpha = -1$) for $\mathcal{N}(\pm \mu, I_p)$ with $p = 1$. 
The finite-dimensional case: What we expect

![Diagram showing the outcome of Laplacian algorithms for $\mathcal{N}(\pm\mu, I_p)$ with $p = 1$.]

**Figure**: Outcome $\mathbf{F}$ of Laplacian algorithms ($\alpha = -1$) for $\mathcal{N}(\pm\mu, I_p)$ with $p = 1$. 
The reality: What we see!

Figure: Outcome $\mathbf{F}$ of Laplacian algorithms ($\alpha = -1$) for $\mathcal{N}(\pm \mu, I_p)$ with $p = 80$. 
The reality: What we see!

*Figure:* Outcome $\mathbf{F}$ of Laplacian algorithms ($\alpha = -1$) for $\mathcal{N}(\pm \mu, I_p)$ with $p = 80$. 
The reality: What we see! (on MNIST)

Figure: Vectors $[F^{(u)}]_{a}$, $a = 1, 2, 3$, for 3-class MNIST data (zeros, ones, twos), $n = 192$, $p = 784$, $n_l/n = 1/16$, Gaussian kernel.
The reality: What we see! (on MNIST)

Figure: Vectors $[F^{(u)}]_a$, $a = 1, 2, 3$, for 3-class MNIST data (zeros, ones, twos), $n = 192$, $p = 784$, $n_l/n = 1/16$, Gaussian kernel.
The reality: What we see! (on MNIST)

Figure: Vectors $[F(u)]_{a,1}$, $a = 1, 2, 3$, for 3-class MNIST data (zeros, ones, twos), $n = 192$, $p = 784$, $n_l/n = 1/16$, Gaussian kernel.
Consequences of the finite-dimensional “mismatch”
Consequences of the finite-dimensional “mismatch”

- A priori, the algorithm should not work
Consequences of the finite-dimensional “mismatch”

- A priori, the algorithm should not work
- Indeed “in general” it does not!
Consequences of the finite-dimensional “mismatch”

▶ A priori, the algorithm should not work
▶ Indeed “in general” it does not!
▶ But, luckily, after some (not clearly motivated) renormalization (e.g., $\alpha = -1$, $F_i \leftarrow F_i \cdot n_{[l],i}$), it works again...

But it does not use efficiently unlabelled data!


Our concern is this: it is frequently the case that we would be better off just discarding the unlabeled data and employing a supervised method, rather than taking a semi-supervised route. Thus we worry about the embarrassing situation where the addition of unlabeled data degrades the performance of a classifier.
Consequences of the finite-dimensional “mismatch”

- A priori, the algorithm should not work
- Indeed “in general” it does not!
- But, luckily, after some (not clearly motivated) renormalization (e.g., $\alpha = -1$, $F_i \leftarrow F_i / n[l,i]$), it works again...

- BUT it does not use efficiently unlabelled data!
Consequences of the finite-dimensional “mismatch”

▶ A priori, the algorithm should not work
▶ Indeed “in general” it does not!
▶ But, luckily, after some (not clearly motivated) renormalization (e.g., $\alpha = -1$, $F_i \leftarrow F_i / n_{[l],i}$), it works again...

▶ BUT it does not use efficiently unlabelled data!


Our concern is this: it is frequently the case that we would be better off just discarding the unlabeled data and employing a supervised method, rather than taking a semi-supervised route. Thus we worry about the embarrassing situation where the addition of unlabeled data degrades the performance of a classifier.
Theorem (\textbf{[Mai,C’18] Asymptotic Performance of SSL})

For $x_i \in C_b$ unlabelled, score vector $F_{i,.} \in \mathbb{R}^k$ satisfies:

$$F_{i,.} - G_b \to 0, \quad G_b \sim \mathcal{N}(m_b, \Sigma_b)$$

with $m_b \in \mathbb{R}^k$, $\Sigma_b \in \mathbb{R}^{k \times k}$ function of

- $f(\tau), f'(\tau), f''(\tau), \mu_1, \ldots, \mu_k, C_1, \ldots, C_k$
- only $n_l$. 
Asymptotic Performance Analysis

Theorem ([Mai,C’18] Asymptotic Performance of SSL)

For $x_i \in C_b$ unlabelled, score vector $F_{i,.} \in \mathbb{R}^k$ satisfies:

$$F_{i,.} - G_b \to 0, \ G_b \sim \mathcal{N}(m_b, \Sigma_b)$$

with $m_b \in \mathbb{R}^k$, $\Sigma_b \in \mathbb{R}^{k \times k}$ function of

- $f(\tau), f'(\tau), f''(\tau), \mu_1, \ldots, \mu_k, C_1, \ldots, C_k$
- only $n_l$.

Figure: Accuracy as a function of $n_{[u]} / p$ with $n_{[l]} / p = 2$, $c_1 = c_2$, $p = 100$, $-\mu_1 = \mu_2 = [1; 0_{p-1}]$, $\{C\}_{i,j} = .1|\!|i-j|\!|$. Graph constructed with $K_{i,j} = e^{-\|x_i - x_j\|^2 / p}$. 

Asymptotic Performance Analysis

Theorem ([Mai, C’18] Asymptotic Performance of SSL)

For $x_i \in C_b$ unlabelled, score vector $F_i, \cdot \in \mathbb{R}^k$ satisfies:

$$F_i, \cdot - G_b \to 0, \ G_b \sim \mathcal{N}(m_b, \Sigma_b)$$

with $m_b \in \mathbb{R}^k$, $\Sigma_b \in \mathbb{R}^{k \times k}$ function of

- $f(\tau), f'(\tau), f''(\tau), \mu_1, \ldots, \mu_k, C_1, \ldots, C_k$
- only $n_l$.

Figure: Accuracy as a function of $n_{[u]}/p$ with $n_{[l]}/p = 2$, $c_1 = c_2$, $p = 100$, $-\mu_1 = \mu_2 = [1; 0_{p-1}]$, $\{C\}_{i,j} = .1|^{i-j}|$. Graph constructed with $K_{i,j} = e^{-\|x_i - x_j\|^2/p}$. 
**Improved SSL**

**Solution:** From RMT calculus (but not from ML intuition!), solution is to replace $K$ by

$$\tilde{K} \equiv PKP, \quad P = I_n - \frac{1}{n}1_n1_n^T.$$
Improved SSL

**Solution:** From RMT calculus (but not from ML intuition!), solution is to replace $K$ by

$$\tilde{K} \equiv PKP, \quad P = I_n - \frac{1}{n}1_n1_n^T.$$ 

**Theorem ([Mai,C’19] Asymptotic Performance of Improved SSL)**

For $x_i \in C_b$ unlabelled, score vector $\tilde{F}_{i,.} \in \mathbb{R}^k$ satisfies:

$$\tilde{F}_{i,.} - \tilde{G}_b \to 0, \quad \tilde{G}_b \sim \mathcal{N}(\tilde{m}_b, \tilde{\Sigma}_b)$$

with $\tilde{m}_b \in \mathbb{R}^k$, $\tilde{\Sigma}_b \in \mathbb{R}^{k \times k}$ function of

- $f(\tau), f'(\tau), f''(\tau), \mu_1, \ldots, \mu_k, C_1, \ldots, C_k$
- $n_l$ and $n_u$. 

![Figure: Accuracy as a function of $n_u/p$](image-url)
**Improved SSL**

**Solution:** From RMT calculus (but not from ML intuition!), solution is to replace $K$ by

$$\tilde{K} \equiv PKP, \quad P = I_n - \frac{1}{n}1_n1_n^T.$$ 

**Theorem ([Mai,C’19] Asymptotic Performance of Improved SSL)**

For $x_i \in C_b$ unlabelled, score vector $\tilde{F}_{i,.} \in \mathbb{R}^k$ satisfies:

$$\tilde{F}_{i,.} - \tilde{G}_b \to 0, \quad \tilde{G}_b \sim \mathcal{N}(\tilde{m}_b, \tilde{\Sigma}_b)$$

with $\tilde{m}_b \in \mathbb{R}^k$, $\tilde{\Sigma}_b \in \mathbb{R}^{k \times k}$ function of

- $f(\tau), f'(\tau), f''(\tau), \mu_1, \ldots, \mu_k, C_1, \ldots, C_k$
- $n_l$ and $n_u$.

![Accuracy as a function of $n_u/p$](image)
**Improved SSL**

**Solution:** From RMT calculus (but not from ML intuition!), solution is to replace $K$ by

$$\tilde{K} \equiv PKP, \quad P = I_n - \frac{1}{n} 1_n 1_n^T.$$ 

**Theorem ([Mai,C’19] Asymptotic Performance of Improved SSL)**

For $x_i \in C_b$ unlabelled, score vector $\tilde{F}_{i,.} \in \mathbb{R}^k$ satisfies:

$$\tilde{F}_{i,.} - \tilde{G}_b \to 0, \quad \tilde{G}_b \sim \mathcal{N}(\tilde{m}_b, \tilde{\Sigma}_b)$$

with $\tilde{m}_b \in \mathbb{R}^k$, $\tilde{\Sigma}_b \in \mathbb{R}^{k \times k}$ function of

- $f(\tau), f'(\tau), f''(\tau), \mu_1, \ldots, \mu_k, C_1, \ldots, C_k$
- $n_l$ and $n_u$.

---

**Figure:** Accuracy as a function of $n_u/p$ with $n_l/p = 2$, $c_1 = c_2$, $p = 100$.

- Gaussian kernel $K_{ij} = e^{-\|x_i - x_j\|^2/p}$.
What about real data?

Figure: Top: distribution of normalized pairwise distances for noisy MNIST data (8,9). Bottom: average accuracy as a function of $n_u$ with $n_l = 10$, computed over 1000 random realizations.
What about real data?

Figure: Top: distribution of normalized pairwise distances for noisy MNIST data (8,9). Bottom: average accuracy as a function of $n[u]$ with $n[l] = 10$, computed over 1000 random realizations.
What about real data?

\[ \text{SNR} = +\infty \text{dB} \]

\[ \text{SNR} = -5 \text{dB} \]

\[ \text{SNR} = -10 \text{dB} \]

**Figure:** Top: distribution of normalized pairwise distances for noisy MNIST data (8,9). Bottom: average accuracy as a function of \( n[u] \) with \( n[l] = 10 \), computed over 1000 random realizations.
What about real data?

\[ \text{SNR} = +\infty \text{dB} \]

\[ \text{SNR} = -5 \text{dB} \]

\[ \text{SNR} = -10 \text{dB} \]

**Figure:** Top: distribution of normalized pairwise distances for noisy MNIST data (8,9). Bottom: average accuracy as a function of \( n_{[u]} \) with \( n[l] = 10 \), computed over 1000 random realizations.
What about real data?

Figure: Top: distribution of normalized pairwise distances for noisy MNIST data (8,9). Bottom: average accuracy as a function of $n_u$ with $n_l = 10$, computed over 1000 random realizations.
Experimental evidence: MNIST

<table>
<thead>
<tr>
<th>Digits</th>
<th>(0,8)</th>
<th>(2,7)</th>
<th>(6,9)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$n_u = 100$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Centered kernel (RMT)</td>
<td>89.5±3.6</td>
<td>89.5±3.4</td>
<td>85.3±5.9</td>
</tr>
<tr>
<td>Iterated centered kernel (RMT)</td>
<td>89.5±3.6</td>
<td>89.5±3.4</td>
<td>85.3±5.9</td>
</tr>
<tr>
<td>Laplacian</td>
<td>75.5±5.6</td>
<td>74.2±5.8</td>
<td>70.0±5.5</td>
</tr>
<tr>
<td>Iterated Laplacian</td>
<td>87.2±4.7</td>
<td>86.0±5.2</td>
<td>81.4±6.8</td>
</tr>
<tr>
<td>Manifold</td>
<td>88.0±4.7</td>
<td>88.4±3.9</td>
<td>82.8±6.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$n_u = 1000$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Centered kernel (RMT)</td>
<td>92.2±0.9</td>
<td>92.5±0.8</td>
<td>92.6±1.6</td>
</tr>
<tr>
<td>Iterated centered kernel (RMT)</td>
<td>92.3±0.9</td>
<td>92.5±0.8</td>
<td>92.9±1.4</td>
</tr>
<tr>
<td>Laplacian</td>
<td>65.6±4.1</td>
<td>74.4±4.0</td>
<td>69.5±3.7</td>
</tr>
<tr>
<td>Iterated Laplacian</td>
<td>92.2±0.9</td>
<td>92.4±0.9</td>
<td>92.0±1.6</td>
</tr>
<tr>
<td>Manifold</td>
<td>91.1±1.7</td>
<td>91.4±1.9</td>
<td>91.4±2.0</td>
</tr>
</tbody>
</table>

Table: Comparison of classification accuracy (%) on MNIST datasets with $n_I = 10$. Computed over 1000 random iterations for $n_u = 100$ and 100 for $n_u = 1000$. 
Experimental evidence: Traffic signs (HOG features)

Table: Comparison of classification accuracy (%) on German Traffic Sign datasets with $n_l = 10$. Computed over 1000 random iterations for $n_u = 100$ and 100 for $n_u = 1000$. 

<table>
<thead>
<tr>
<th>Class ID</th>
<th>(2,7)</th>
<th>(9,10)</th>
<th>(11,18)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_u = 100$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Centered kernel (RMT)</td>
<td>79.0±10.4</td>
<td>77.5±9.2</td>
<td>78.5±7.1</td>
</tr>
<tr>
<td>Iterated centered kernel (RMT)</td>
<td>85.3±5.9</td>
<td>89.2±5.6</td>
<td>90.1±6.7</td>
</tr>
<tr>
<td>Laplacian</td>
<td>73.8±9.8</td>
<td>77.3±9.5</td>
<td>78.6±7.2</td>
</tr>
<tr>
<td>Iterated Laplacian</td>
<td>83.7±7.2</td>
<td>88.0±6.8</td>
<td>87.1±8.8</td>
</tr>
<tr>
<td>Manifold</td>
<td>77.6±8.9</td>
<td>81.4±10.4</td>
<td>82.3±10.8</td>
</tr>
</tbody>
</table>

| $n_u = 1000$ |                |                |                |
| Centered kernel (RMT) | 83.6±2.4     | 84.6±2.4       | 88.7±9.4       |
| Iterated centered kernel (RMT) | 84.8±3.8     | 88.0±5.5       | 96.4±3.0       |
| Laplacian    | 72.7±4.2       | 88.9±5.7       | 95.8±3.2       |
| Iterated Laplacian | 83.0±5.5     | 88.2±6.0       | 92.7±6.1       |
| Manifold     | 77.7±5.8       | 85.0±9.0       | 90.6±8.1       |
Even more striking: new intuitions and cheap algorithms

- **Computation cost reduction**: \((p, n \gg 1)\)

\[ \varepsilon \text{-subsampling } K \in \mathbb{R}^{n\varepsilon \times n\varepsilon} \]

\[ K = \begin{bmatrix}
\vdots \\
\vdots \\
\end{bmatrix} \quad \rightarrow \quad K = \begin{bmatrix}
\varepsilon = \frac{1}{50} \\
\vdots \\
\vdots \\
\end{bmatrix} \]

\[ \varepsilon \leftarrow K_{\varepsilon} \text{ more sparse} \]

\[ \|\mu\|_2 \text{ (easier task } \rightarrow \text{)} \]

Classification impossible
Even more striking: new intuitions and cheap algorithms

- **Computation cost reduction:** \((p, n \gg 1)\)

  \[ \varepsilon \text{-subsampling } K \in \mathbb{R}^{n\varepsilon \times n\varepsilon} \]

- **Phase transition of spectral clustering:** \((x_i \sim \mathcal{N}(\pm \mu, I_p), n/p = 100)\),

![Graph showing phase transition of spectral clustering](image)
Even more striking: new intuitions and cheap algorithms

- **Computation cost reduction:** $(p, n \gg 1)$
  - $\varepsilon$-subsampling $K \in \mathbb{R}^{n\varepsilon \times n\varepsilon}$
  - $K_\varepsilon \equiv K \odot B$ with $B_{ij} \sim \text{Bern}(\varepsilon)$ i.i.d.

- **Phase transition of spectral clustering:** $(x_i \sim \mathcal{N}(\pm \mu, I_p), \ n/p = 100)$,
Even more striking: new intuitions and cheap algorithms

- Computation cost reduction: \((p, n \gg 1)\)
  - \(\varepsilon\)-subsampling \(K \in \mathbb{R}^{n\varepsilon \times n\varepsilon}\)
  - \(K_\varepsilon \equiv K \odot B\) with \(B_{ij} \sim \text{Bern}(\varepsilon)\) i.i.d.

- Phase transition of spectral clustering: \((x_i \sim \mathcal{N}(\pm \mu, I_p), \frac{n}{p} = 100)\).

\[
\begin{bmatrix}
0 & \varepsilon \\
\varepsilon & 0
\end{bmatrix}
\]
Even more striking: new intuitions and cheap algorithms

- **Computation cost reduction**: \((p, n \gg 1)\)
  
  \(\rightarrow \varepsilon\)-subsampling \(K \in \mathbb{R}^{n\varepsilon \times n\varepsilon}\)
  
  \(\rightarrow K_{\varepsilon} \equiv K \odot B\) with \(B_{ij} \sim \text{Bern}(\varepsilon)\) i.i.d.

- **Phase transition of spectral clustering**: \((x_i \sim \mathcal{N}(\pm \mu, I_p), n/p = 100),\)

\[\parallel \mu \parallel^2\text{ (easier task \(\rightarrow\))} \quad \text{Classification possible}\]
\[\varepsilon \quad \text{\(\leftarrow K_{\varepsilon}\) more sparse}\]
Takeaway Message 3

“RMT Also Grasps ‘Real Data’ Processing”
From i.i.d. to concentrated random vectors

**Beyond Gaussian Mixtures:** results still valid for **concentrated random vectors.**
Beyond Gaussian Mixtures: results still valid for concentrated random vectors.

**Definition (Concentrated Random Vector)**

$x \in \mathbb{R}^p$ is concentrated if, for all Lipschitz $f : \mathbb{R}^p \to \mathbb{R}$, there exists $m_f \in \mathbb{R}$, such that

$$P \left( |f(x) - m_f| > \varepsilon \right) \leq e^{-g(\varepsilon)}, \quad g \text{ increasing function.}$$
From i.i.d. to concentrated random vectors

**Beyond Gaussian Mixtures:** results still valid for **concentrated random vectors.**

**Definition (Concentrated Random Vector)**

$x \in \mathbb{R}^p$ is concentrated if, for all Lipschitz $f : \mathbb{R}^p \rightarrow \mathbb{R}$, there exists $m_f \in \mathbb{R}$, such that

$$P \left( |f(x) - m_f| > \varepsilon \right) \leq e^{-g(\varepsilon)}, \quad g \text{ increasing function.}$$

$$x = (x_1, \ldots, x_p) \sim s_p$$

**Observations**
From i.i.d. to concentrated random vectors

Theorem ([Louart,C’18] [Seddik,C’19] Kernel Universality)

For \( x_i \sim L(\mu_a, C_a) \) concentrated random vector, under the conditions of [C-Benaych’16],

\[
\|K - \hat{K}\| \xrightarrow{a.s.} 0, \quad \hat{K} = f(\tau)1_n1_n^T + \frac{1}{p}ZZ^T + JAJ^T + *
\]

with A only dependent on \( f(\tau), f'(\tau), f''(\tau), \mu_1, \ldots, \mu_k, C_1, \ldots, C_k \).
Theorem ([Louart,C’18] [Seddik,C’19] Kernel Universality)

For $x_i \sim \mathcal{L}(\mu, C)$ concentrated random vector, under the conditions of [C-Benaych’16],

$$\|K - \hat{K}\| \xrightarrow{a.s.} 0, \quad \hat{K} = f(\tau)1_n1_n^T + \frac{1}{p}ZZ^T + JAJ^T + *$$

with $A$ only dependent on $f(\tau), f'(\tau), f''(\tau), \mu_1, \ldots, \mu_k, C_1, \ldots, C_k$.

[right] Same result as [C-Benaych’16]... Universality of first two moments!
Ok... so what?

Key Finding. GAN-generated data are concentrated random vectors!
Key Finding. GAN-generated data are concentrated random vectors!
Ok... so what?

Fake images → Convolutional Neural Net

Lipschitz maps → Concentrated! Feature Vector
Ok... so what?

$\mathcal{X}_1 \quad \cdots \quad \mathcal{X}_n$

Feature Vectors

$K = \left\{ e^{-\frac{1}{2p} \|x_i - x_j\|^2} \right\}_{i,j=1}^n$

Spectral Clustering
Gaussian, GAN, and real data

Results. [Seddik,C’19]
Gaussian, GAN, and real data

![Gaussian Images](image1)

![GAN Images](image2)

![Real Images](image3)
Gaussian, GAN, and real data

resnet50 ($p = 2048$)

vgg16 ($p = 4096$)

densenet201 ($p = 1920$)

GAN Images

Real Images
Gaussian, GAN, and real data

GAN Images

resnet50 (p = 2048)

Eigenvectors

Eigenvector 2

Eigenvector 3

vgg16 (p = 4096)

densnet201 (p = 1920)

Eigenvectors

Eigenvector 2

Eigenvector 3

Real Images

Real images

Gaussian mixture

Eigenvectors

Eigenvector 2

Eigenvector 3
Conclusion

Our Research Activities:

- The road ahead:
  - from theory to practice: exploit theory to improve real-data learning
  - beyond explicit learning: implicit optimizations, non-convex problems.
  - ML = representation + stat-learning (VAE, NN dynamics?)
Our Research Activities:

- **Large-dimensional model analysis**
  - (Laplacians, kernels, non-linear functionals, fixed-point models, ...)

- Random Matrix Theory for Data Processing
Conclusion

Our Research Activities:

Large-dimensional model analysis
(Laplacians, kernels, non-linear functionals, fixed-point models, ...)

Performance study
(support vector machines, graph semi-superv. learning, community detection, spectral methods, ...)

Random Matrix Theory for Data Processing
Our Research Activities:

- **Large-dimensional model analysis**
  - (Laplacians, kernels, non-linear functionals, fixed-point models, ...)

- **Performance study**
  - (support vector machines, graph semi-superv. learning, community detection, spectral methods, ...)

- **Algorithm improvements**
  - (hyperparameter optimization, activation/kernel improvement, new algorithms/insights, ...)

- **Random Matrix Theory for Data Processing**
Conclusion

Our Research Activities:

Large-dimensional model analysis
(Laplacians, kernels, non-linear functionals, fixed-point models, ...)

Performance study
(support vector machines, graph semi-superv. learning, community detection, spectral methods, ...)

Random Matrix Theory for Data Processing

Applications Proof of Concept
(brain signal processing, hyperspectral imaging, statistical finance, ...)

Algorithm improvements
(hyperparameter optimization, activation/kernel improvement, new algorithms/insights, ...)

The road ahead:
▶ from theory to practice: exploit theory to improve real-data learning
▶ beyond explicit learning: implicit optimizations, non-convex problems.

ML = representation + stat-learning (VAE, NN dynamics?)
Conclusion

Our Research Activities:

- **Large-dimensional model analysis**
  - (Laplacians, kernels, non-linear functionals, fixed-point models, ...)

- **Performance study**
  - (support vector machines, graph semi-superv. learning, community detection, spectral methods, ...)

- **Applications Proof of Concept**
  - (brain signal processing, hyperspectral imaging, statistical finance, ...)

- **Algorithm improvements**
  - (hyperparameter optimization, activation/kernel improvement, new algorithms/insights, ...)

**The road ahead:**
- from theory to practice: exploit theory to **improve real-data learning**
Conclusion

Our Research Activities:

The road ahead:

- from theory to practice: exploit theory to improve real-data learning
- beyond explicit learning: implicit optimizations, non-convex problems.
Conclusion

Our Research Activities:

- **Large-dimensional model analysis**
  (Laplacians, kernels, non-linear functionals, fixed-point models, ...)

- **Performance study**
  (support vector machines, graph semi-superv. learning, community detection, spectral methods, ...)

- **Applications Proof of Concept**
  (brain signal processing, hyperspectral imaging, statistical finance, ...)

- **Algorithm improvements**
  (hyperparameter optimization, activation/kernel improvement, new algorithms/insights, ...)

**Random Matrix Theory for Data Processing**

**The road ahead:**

- from theory to practice: exploit theory to **improve real-data learning**
- beyond explicit learning: **implicit optimizations, non-convex problems.**
- **ML = representation + stat-learning** (VAE, NN dynamics?)
Our Team: the MIAI “LargeDATA” chair © University Grenoble-Alpes

- G. Besson
  Institut Fourier
  géométrie

- F. Chatelain
  GIPSA
  statistiques

- P. Comon
  GIPSA
  tenseurs

- E. Gaussier
  LIG
  traitement langage

- N. Le Bihan
  GIPSA
  stats, physique

- N. Tremblay
  GIPSA
  graphes

- S. Zozor
  GIPSA
  théorie de l'info

- O. Michel
  GIPSA
  signal, physique

- M. Seddik
  Apprentissage
  applic vision

- L. Dall’Amico
  Physique
  Stats
  graphes

- C. Louart
  Mathématiques
  concentration

- M. Tiomoko
  Apprentissage
  transfer, SSL

- H. Chakroun
  Mathématiques
  géométrie

- C. Doz
  Apprentissage
  RMT et radar

- T. Zarrouk
  Apprentissage
  RMT structuré

- C. Séjourné
  Apprentissage
  RMT non convexe

- B. Nabet
  Finance
  ML & fi-stats

- H. Goulart
  Trait. signal
  tenseurs
Thank you!


