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 1, 2019, Kraków, Poland

CentraleSupélec

gipsa-lab

Outline

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,

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.

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.,

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 $\|\mathbf{x}_i - \mathbf{x}_j\|$ from the same or different clusters (classes), $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^p$ for p large.

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 $\|\mathbf{x}_i - \mathbf{x}_j\|$ from the same or different clusters (classes), $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^p$ for p large.
  - Classical machine learning algos (e.g., kernel spectral clustering) still work for large dimensional data, although we **do not** understand why ...

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 $\|\mathbf{x}_i - \mathbf{x}_j\|$ from the same or different clusters (classes), $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^p$ 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**.

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 $\|\mathbf{x}_i - \mathbf{x}_j\|$ from the same or different clusters (classes), $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^p$ 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)

Outline

Sample covariance matrix in the large n, p regime

- For $\mathbf{x}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$, estimate the covariance matrix from n data samples $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{p \times n}$.

Sample covariance matrix in the large n, p regime

- For $\mathbf{x}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$, estimate the covariance matrix from n data samples $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{p \times n}$.
- Classical maximum likelihood sample covariance matrix:

$$\hat{\mathbf{C}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^\mathsf{T} = \frac{1}{n} \mathbf{X}\mathbf{X}^\mathsf{T} \in \mathbb{R}^{p \times p}$$

of rank at most n.

Sample covariance matrix in the large n, p regime

- For $\mathbf{x}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$, estimate the covariance matrix from n data samples $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{p \times n}$.
- Classical maximum likelihood sample covariance matrix:

$$\hat{\mathbf{C}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^\mathsf{T} = \frac{1}{n} \mathbf{X}\mathbf{X}^\mathsf{T} \in \mathbb{R}^{p \times p}$$

  of rank at most n.
- In the regime where $n \sim p$, conventional wisdom breaks down, for $\mathbf{C} = \mathbf{I}_p$ with $n < p$, SCM will never be correct:

$$\|\mathbf{C} - \hat{\mathbf{C}}\| \not\to 0, n, p \to \infty$$

  with at least $p - n$ zero eigenvalues!

Sample covariance matrix in the large n, p regime

- For $\mathbf{x}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$, estimate the covariance matrix from n data samples $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{p \times n}$.
- Classical maximum likelihood sample covariance matrix:

$$\hat{\mathbf{C}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^\mathsf{T} = \frac{1}{n} \mathbf{X} \mathbf{X}^\mathsf{T} \in \mathbb{R}^{p \times p}$$

  of rank at most n.
- In the regime where $n \sim p$, conventional wisdom breaks down, for $\mathbf{C} = \mathbf{I}_p$ with $n < p$, SCM will never be correct:

$$\|\mathbf{C} - \hat{\mathbf{C}}\| \not\to 0, n, p \to \infty$$

  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

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

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

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

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

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

- eigenvalues span on $[(1 - \sqrt{c})^2, (1 + \sqrt{c})^2]$.

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

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

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.

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

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

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.

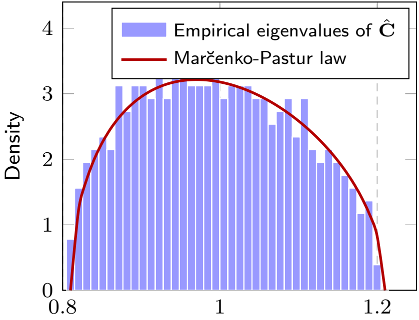
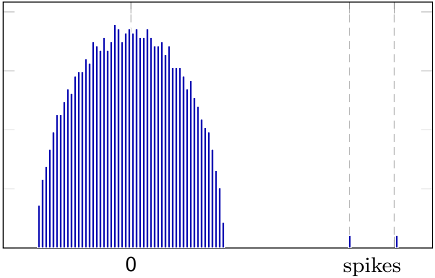


Figure: Eigenvalue distribution of $\hat{\mathbf{C}}$ versus Marčenko-Pastur law, $p = 500$, $n = 50\,000$.

Outline

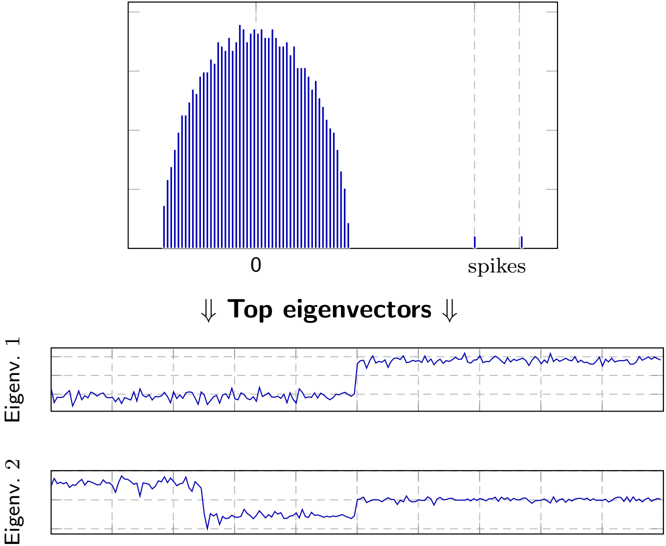
Reminder on kernel spectral clustering

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

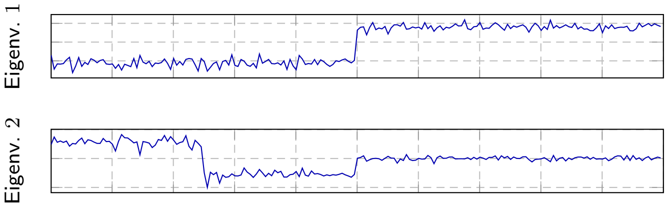


Reminder on kernel spectral clustering

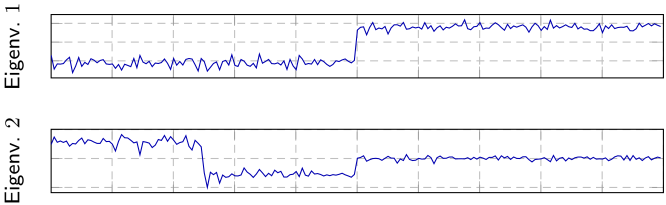
Two-step classification of n data points based on similarity $\mathbf{S} \in \mathbb{R}^{n \times n}$:



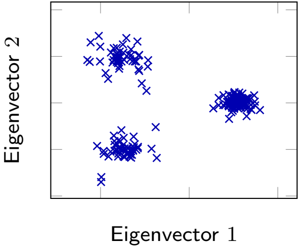
Reminder on kernel spectral clustering



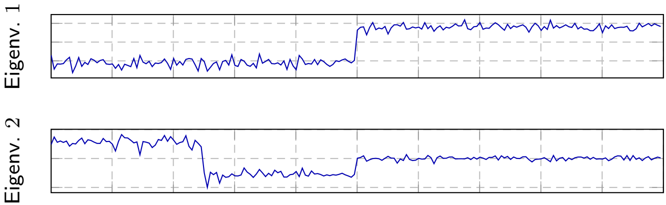
Reminder on kernel spectral clustering



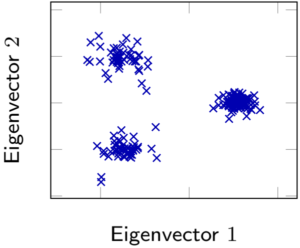
$\Downarrow$ k-**dimensional representation** $\Downarrow$



Reminder on kernel spectral clustering



⇓ k-**dimensional representation** ⇓



⇓

EM or k-means clustering.

Loss of relevance of Euclidean distance

- Simplest binary Gaussian mixture classification setting

$$
\begin{aligned}
\mathcal{C}_1 : & \mathbf{x} = \boldsymbol{\mu} + \mathbf{z}, \quad \mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{I}_p); \\
\mathcal{C}_2 : & \mathbf{x} = -\boldsymbol{\mu} + (\mathbf{I}_p + \mathbf{E})^{\frac{1}{2}} \mathbf{z}, \quad \mathbf{x} \sim \mathcal{N}(-\boldsymbol{\mu}, \mathbf{I}_p + \mathbf{E}).
\end{aligned}
$$

for $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_p)$.

Loss of relevance of Euclidean distance

- Simplest binary Gaussian mixture classification setting

$$\begin{aligned} \mathcal{C}_1 : & \mathbf{x} = \boldsymbol{\mu} + \mathbf{z}, \quad \mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{I}_p); \\ \mathcal{C}_2 : & \mathbf{x} = -\boldsymbol{\mu} + (\mathbf{I}_p + \mathbf{E})^{\frac{1}{2}} \mathbf{z}, \quad \mathbf{x} \sim \mathcal{N}(-\boldsymbol{\mu}, \mathbf{I}_p + \mathbf{E}). \end{aligned}$$

  for $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_p)$.

- Neyman-Pearson test tells us: classification is non-trivial only when

$$\|\boldsymbol{\mu}\| \geq O(1), \quad \|\mathbf{E}\| \geq O(p^{-1/2}), \quad |\operatorname{tr} \mathbf{E}| \geq O(\sqrt{p}), \quad \|\mathbf{E}\|_F^2 \geq O(1).$$

Loss of relevance of Euclidean distance

- Simplest binary Gaussian mixture classification setting

$$\begin{aligned} \mathcal{C}_1 : & \mathbf{x} = \boldsymbol{\mu} + \mathbf{z}, \quad \mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{I}_p); \\ \mathcal{C}_2 : & \mathbf{x} = -\boldsymbol{\mu} + (\mathbf{I}_p + \mathbf{E})^{\frac{1}{2}} \mathbf{z}, \quad \mathbf{x} \sim \mathcal{N}(-\boldsymbol{\mu}, \mathbf{I}_p + \mathbf{E}). \end{aligned}$$

  for $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_p)$.
- Neyman-Pearson test tells us: classification is non-trivial only when

$$\|\boldsymbol{\mu}\| \geq O(1), \quad \|\mathbf{E}\| \geq O(p^{-1/2}), \quad |\operatorname{tr} \mathbf{E}| \geq O(\sqrt{p}), \quad \|\mathbf{E}\|_F^2 \geq O(1).$$

- In this non-trivial setting, for $\mathbf{x}_i \in \mathcal{C}_a, \mathbf{x}_j \in \mathcal{C}_b$,

$$\frac{1}{p}\|\mathbf{x}_i - \mathbf{x}_j\|^2 = \begin{cases} \frac{1}{p}\|\mathbf{z}_i - \mathbf{z}_j\|^2 + Ap^{-1/2}, & \text{for } a = b = 2; \\ \frac{1}{p}\|\mathbf{z}_i - \mathbf{z}_j\|^2 + Bp^{-1/2}, & \text{for } a = 1, b = 2 \end{cases} \tag{2}$$

Loss of relevance of Euclidean distance

- Simplest binary Gaussian mixture classification setting

$$\mathcal{C}_1 : \mathbf{x} = \boldsymbol{\mu} + \mathbf{z}, \quad \mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{I}_p);$$
$$\mathcal{C}_2 : \mathbf{x} = -\boldsymbol{\mu} + (\mathbf{I}_p + \mathbf{E})^{\frac{1}{2}} \mathbf{z}, \quad \mathbf{x} \sim \mathcal{N}(-\boldsymbol{\mu}, \mathbf{I}_p + \mathbf{E}).$$

for $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_p)$.

- Neyman-Pearson test tells us: classification is non-trivial only when

$$\|\boldsymbol{\mu}\| \geq O(1), \quad \|\mathbf{E}\| \geq O(p^{-1/2}), \quad |\operatorname{tr} \mathbf{E}| \geq O(\sqrt{p}), \quad \|\mathbf{E}\|_F^2 \geq O(1).$$

- In this non-trivial setting, for $\mathbf{x}_i \in \mathcal{C}_a, \mathbf{x}_j \in \mathcal{C}_b$,

$$\frac{1}{p}\|\mathbf{x}_i - \mathbf{x}_j\|^2 = \begin{cases} \frac{1}{p}\|\mathbf{z}_i - \mathbf{z}_j\|^2 + Ap^{-1/2}, & \text{for } a = b = 2; \\ \frac{1}{p}\|\mathbf{z}_i - \mathbf{z}_j\|^2 + Bp^{-1/2}, & \text{for } a = 1, b = 2 \end{cases} \tag{2}$$

- For A, B both of order $O(1)$ and $A > B$ with high probability for p large, so

$$\max_{1 \leq i \neq j \leq n} \left\{ \frac{1}{p}\|\mathbf{x}_i - \mathbf{x}_j\|^2 - 2 \right\} \to 0 \tag{3}$$

almost surely as $n, p \to \infty$.

Kernel spectral clustering for large dimensional data

Objective: "cluster" data $\mathbf{x}_1, \ldots, \mathbf{x}_n$ into K similarity classes.

Consider the RBF kernel matrix $\mathbf{K}_{ij} = \exp\left(-\frac{1}{2p}\|\mathbf{x}_i - \mathbf{x}_j\|^2\right)$.

Kernel spectral clustering for large dimensional data

Objective: "cluster" data $\mathbf{x}_1, \ldots, \mathbf{x}_n$ into K similarity classes.
Consider the RBF kernel matrix $\mathbf{K}_{ij} = \exp\left(-\frac{1}{2p}\|\mathbf{x}_i - \mathbf{x}_j\|^2\right)$.

$\mathbf{K} =$

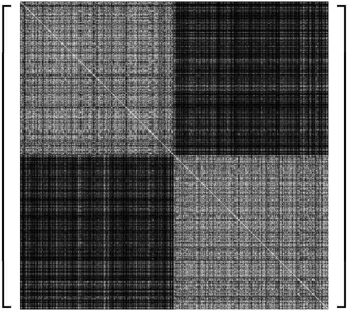
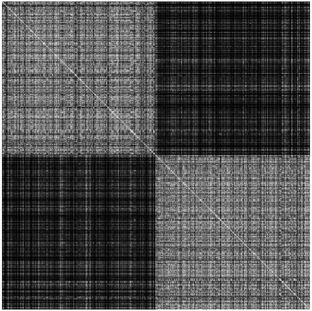


Figure: Kernel matrices $\mathbf{K}$ and the second top eigenvectors $\mathbf{v}_2$ for small (left, $p = 5, n = 500$) and large (right, $p = 250, n = 500$) dimensional data.

Kernel spectral clustering for large dimensional data

Objective: "cluster" data $\mathbf{x}_1, \ldots, \mathbf{x}_n$ into K similarity classes.

Consider the RBF kernel matrix $\mathbf{K}_{ij} = \exp\left(-\frac{1}{2p}\|\mathbf{x}_i - \mathbf{x}_j\|^2\right)$.

$\mathbf{K} =$ []

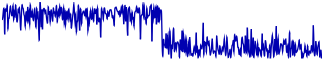
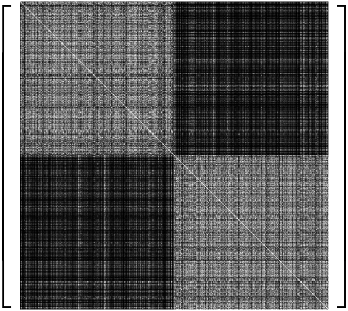
$\mathbf{v}_2 =$ []

Figure: Kernel matrices $\mathbf{K}$ and the second top eigenvectors $\mathbf{v}_2$ for small (left, $p = 5, n = 500$) and large (right, $p = 250, n = 500$) dimensional data.

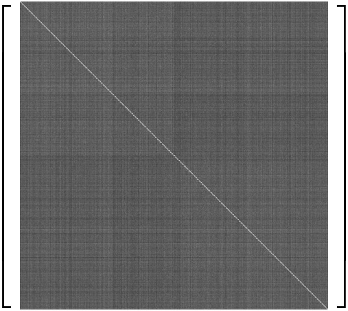
Kernel spectral clustering for large dimensional data

Objective: "cluster" data $\mathbf{x}_1, \ldots, \mathbf{x}_n$ into K similarity classes.
Consider the RBF kernel matrix $\mathbf{K}_{ij} = \exp\left(-\frac{1}{2p}\|\mathbf{x}_i - \mathbf{x}_j\|^2\right)$.

$\mathbf{K} =$



$\mathbf{K} =$



$\mathbf{v}_2 =$

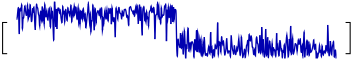


Figure: Kernel matrices $\mathbf{K}$ and the second top eigenvectors $\mathbf{v}_2$ for small (left, $p = 5, n = 500$) and large (right, $p = 250, n = 500$) dimensional data.

Kernel spectral clustering for large dimensional data

Objective: "cluster" data $\mathbf{x}_1, \ldots, \mathbf{x}_n$ into K similarity classes.

Consider the RBF kernel matrix $\mathbf{K}_{ij} = \exp\left(-\frac{1}{2p}\|\mathbf{x}_i - \mathbf{x}_j\|^2\right)$.

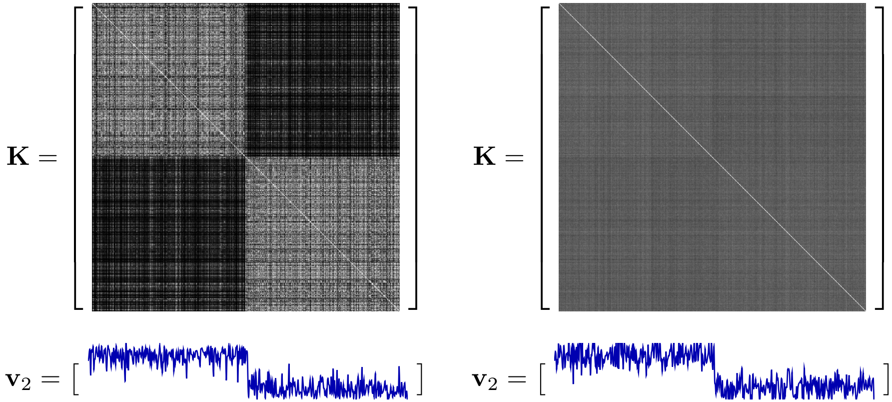


Figure: Kernel matrices $\mathbf{K}$ and the second top eigenvectors $\mathbf{v}_2$ for small (left, $p = 5, n = 500$) and large (right, $p = 250, n = 500$) dimensional data.

But why kernel spectral clustering works?

The accumulated effect of the small "hidden" statistical information (in $\boldsymbol{\mu}, \mathbf{E}$).

But why kernel spectral clustering works?

The accumulated effect of the small "hidden" statistical information (in $\boldsymbol{\mu}, \mathbf{E}$).

$$\mathbf{K} = \exp(-1)\left(\mathbf{1}_n\mathbf{1}_n^\mathsf{T} + \frac{1}{p}\mathbf{Z}^\mathsf{T}\mathbf{Z}\right) + g(\boldsymbol{\mu}, \mathbf{E})\frac{1}{p}\mathbf{j}\mathbf{j}^\mathsf{T} + * + o_{\|\cdot\|}(1) \tag{4}$$

with $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_n] \in \mathbb{R}^{p\times n}$ and $\mathbf{j} = [\mathbf{1}_{n/2}; -\mathbf{1}_{n/2}]$, the class-information vector.

But why kernel spectral clustering works?

The accumulated effect of the small "hidden" statistical information (in $\boldsymbol{\mu}, \mathbf{E}$).

$$\mathbf{K} = \exp(-1)\left(\mathbf{1}_n\mathbf{1}_n^\mathsf{T} + \frac{1}{p}\mathbf{Z}^\mathsf{T}\mathbf{Z}\right) + g(\boldsymbol{\mu}, \mathbf{E})\frac{1}{p}\mathbf{j}\mathbf{j}^\mathsf{T} + * + o_{\|\cdot\|}(1) \tag{4}$$

with $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_n] \in \mathbb{R}^{p\times n}$ and $\mathbf{j} = [\mathbf{1}_{n/2}; -\mathbf{1}_{n/2}]$, the class-information vector.

Therefore

But why kernel spectral clustering works?

The accumulated effect of the small "hidden" statistical information (in $\boldsymbol{\mu}, \mathbf{E}$).

$$\mathbf{K} = \exp(-1)\left(\mathbf{1}_n\mathbf{1}_n^\mathsf{T} + \frac{1}{p}\mathbf{Z}^\mathsf{T}\mathbf{Z}\right) + g(\boldsymbol{\mu}, \mathbf{E})\frac{1}{p}\mathbf{j}\mathbf{j}^\mathsf{T} + * + o_{\|\cdot\|}(1) \tag{4}$$

with $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_n] \in \mathbb{R}^{p\times n}$ and $\mathbf{j} = [\mathbf{1}_{n/2}; -\mathbf{1}_{n/2}]$, the class-information vector.

Therefore

- **entry-wsie**: for $\mathbf{K}_{ij} = \exp\left(-\frac{1}{2p}\|\mathbf{x}_i - \mathbf{x}_j\|^2\right)$,

$$\mathbf{K}_{ij} = \exp(-1)\left(1 + \underbrace{\frac{1}{p}\mathbf{z}_i^\mathsf{T}\mathbf{z}_j}_{O(p^{-1/2})}\right) \pm \underbrace{\frac{1}{p}g(\boldsymbol{\mu}, \mathbf{E})}_{O(p^{-1})} + *$$

  so that $\frac{1}{p}g(\boldsymbol{\mu}, \mathbf{E}) \ll \frac{1}{p}\mathbf{z}_i^\mathsf{T}\mathbf{z}_j$;

But why kernel spectral clustering works?

The accumulated effect of the small "hidden" statistical information (in $\boldsymbol{\mu}, \mathbf{E}$).

$$\mathbf{K} = \exp(-1)\left(\mathbf{1}_n\mathbf{1}_n^\mathsf{T} + \frac{1}{p}\mathbf{Z}^\mathsf{T}\mathbf{Z}\right) + g(\boldsymbol{\mu}, \mathbf{E})\frac{1}{p}\mathbf{j}\mathbf{j}^\mathsf{T} + * + o_{\|\cdot\|}(1) \tag{4}$$

with $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_n] \in \mathbb{R}^{p\times n}$ and $\mathbf{j} = [\mathbf{1}_{n/2}; -\mathbf{1}_{n/2}]$, the class-information vector.

Therefore

- **entry-wsie**: for $\mathbf{K}_{ij} = \exp\left(-\frac{1}{2p}\|\mathbf{x}_i - \mathbf{x}_j\|^2\right)$,

$$\mathbf{K}_{ij} = \exp(-1)\left(1 + \underbrace{\frac{1}{p}\mathbf{z}_i^\mathsf{T}\mathbf{z}_j}_{O(p^{-1/2})}\right) \pm \underbrace{\frac{1}{p}g(\boldsymbol{\mu}, \mathbf{E})}_{O(p^{-1})} + *$$

  so that $\frac{1}{p}g(\boldsymbol{\mu}, \mathbf{E}) \ll \frac{1}{p}\mathbf{z}_i^\mathsf{T}\mathbf{z}_j$;
- **spectrum-wise**: $\|\frac{1}{p}\mathbf{Z}^\mathsf{T}\mathbf{Z}\| = O(1)$ and $\|g(\boldsymbol{\mu}, \mathbf{E})\frac{1}{p}\mathbf{j}\mathbf{j}^\mathsf{T}\| = O(1)$ as well!

Outline

Neural networks and deep learning

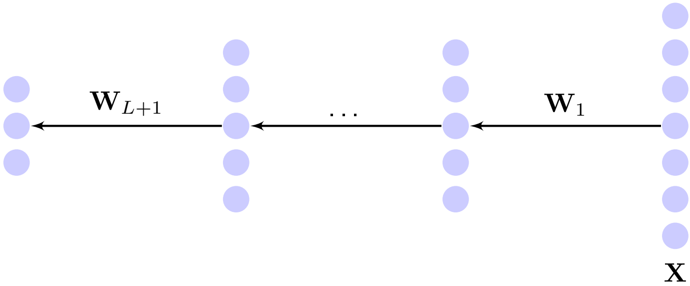


Figure: Illustration of L-hidden-layer nonlinear neural networks

Neural networks and deep learning

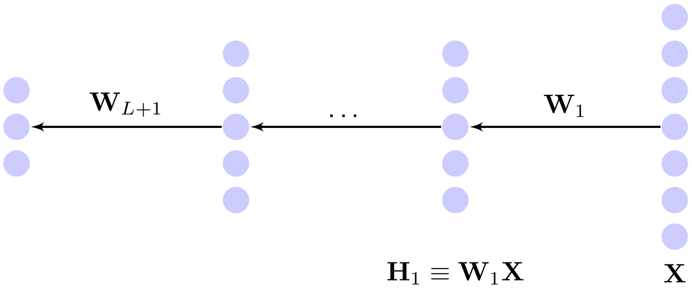


Figure: Illustration of L-hidden-layer nonlinear neural networks

Neural networks and deep learning

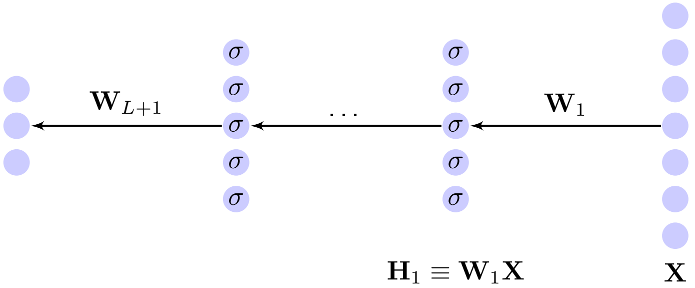


Figure: Illustration of L-hidden-layer nonlinear neural networks

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

Neural networks and deep learning

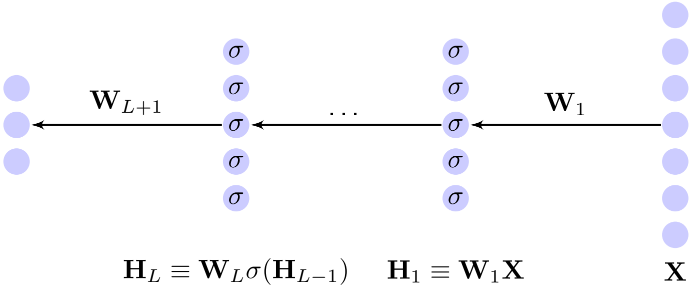


Figure: Illustration of L-hidden-layer nonlinear neural networks

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

Neural networks and deep learning

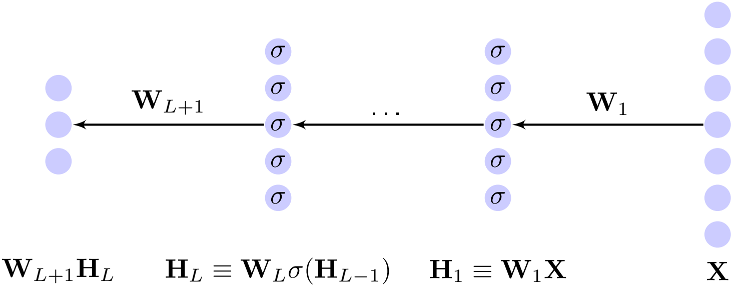
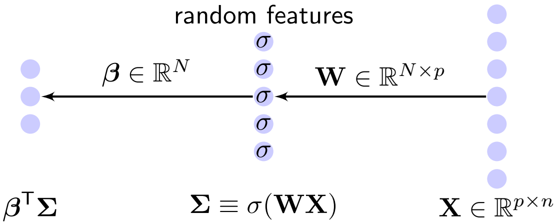


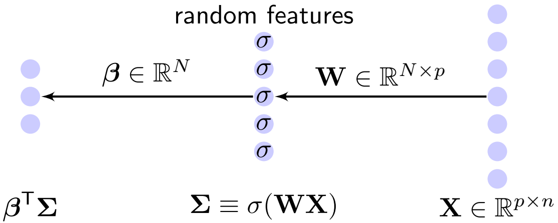
Figure: Illustration of L-hidden-layer nonlinear neural networks

with nonlinear activation function $\sigma(z)$: $\mathrm{ReLU}(z) = \max(z, 0)$, 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



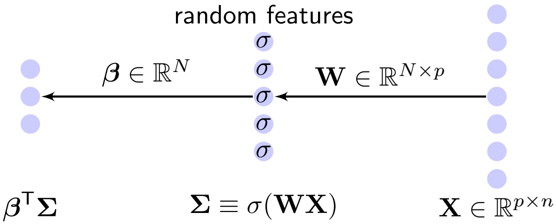
Random neural network with single hidden layer



- For random $\mathbf{W}$ and n, p, N large, $\frac{1}{N}\boldsymbol{\Sigma}^\mathsf{T}\boldsymbol{\Sigma}$ is closely related to

$$\mathbf{K} \equiv \frac{1}{N}\mathbb{E}_{\mathbf{W}}[\sigma(\mathbf{W}\mathbf{X})^\mathsf{T}\sigma(\mathbf{W}\mathbf{X})]$$

Random neural network with single hidden layer



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

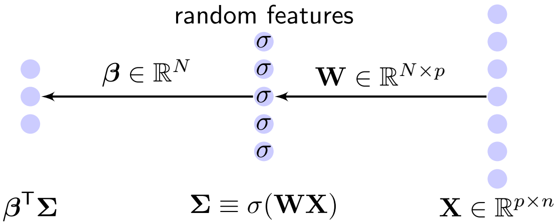
$$\mathbf{K} \equiv \frac{1}{N}\mathbb{E}_{\mathbf{W}}[\sigma(\mathbf{W}\mathbf{X})^\mathsf{T}\sigma(\mathbf{W}\mathbf{X})]$$

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

$$\mathbf{K}_{ij} = \mathbb{E}_{\mathbf{w}}[\sigma(\mathbf{w}^\mathsf{T}\mathbf{x}_i)\sigma(\mathbf{w}^\mathsf{T}\mathbf{x}_j)] = (2\pi)^{-\frac{p}{2}} \int_{\mathbb{R}^p} \sigma(\mathbf{w}^\mathsf{T}\mathbf{x}_i)\sigma(\mathbf{w}^\mathsf{T}\mathbf{x}_j) e^{-\frac{\|\mathbf{w}\|^2}{2}} d\mathbf{w}$$

$$= \frac{1}{2\pi} \int_{\mathbb{R}^2} \sigma(\tilde{\mathbf{w}}^\mathsf{T}\tilde{\mathbf{x}}_i)\sigma(\tilde{\mathbf{w}}^\mathsf{T}\tilde{\mathbf{x}}_j) e^{-\frac{\|\tilde{\mathbf{w}}\|^2}{2}} d\tilde{\mathbf{w}}$$

Random neural network with single hidden layer



- For random $\mathbf{W}$ and n, p, N large, $\frac{1}{N}\boldsymbol{\Sigma}^\mathsf{T}\boldsymbol{\Sigma}$ is closely related to

$$\mathbf{K} \equiv \frac{1}{N}\mathbb{E}_\mathbf{W}[\sigma(\mathbf{WX})^\mathsf{T}\sigma(\mathbf{WX})]$$

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

$$\mathbf{K}_{ij} = \mathbb{E}_\mathbf{w}[\sigma(\mathbf{w}^\mathsf{T}\mathbf{x}_i)\sigma(\mathbf{w}^\mathsf{T}\mathbf{x}_j)] = (2\pi)^{-\frac{p}{2}} \int_{\mathbb{R}^p} \sigma(\mathbf{w}^\mathsf{T}\mathbf{x}_i)\sigma(\mathbf{w}^\mathsf{T}\mathbf{x}_j) e^{-\frac{\|\mathbf{w}\|^2}{2}} d\mathbf{w}$$
$$= \frac{1}{2\pi}\int_{\mathbb{R}^2} \sigma(\tilde{\mathbf{w}}^\mathsf{T}\tilde{\mathbf{x}}_i)\sigma(\tilde{\mathbf{w}}^\mathsf{T}\tilde{\mathbf{x}}_j) e^{-\frac{\|\tilde{\mathbf{w}}\|^2}{2}} d\tilde{\mathbf{w}}$$

with $\tilde{\mathbf{x}}_i = [\|\mathbf{x}_i\|; 0]$ and $\tilde{\mathbf{x}}_j = \left[\frac{\mathbf{x}_i^\mathsf{T}\mathbf{x}_j}{\|\mathbf{x}_i\|}; \sqrt{\|\mathbf{x}_j\|^2 - \frac{(\mathbf{x}_i^\mathsf{T}\mathbf{x}_j)^2}{\|\mathbf{x}_i\|^2}}\right]$.

Nonlinearity in simple random neural networks

Table: $\mathbf{K}_{i,j}$ for commonly used $\sigma(\cdot)$, $\angle \equiv \frac{\mathbf{x}_i^\mathsf{T}\mathbf{x}_j}{\|\mathbf{x}_i\|\|\mathbf{x}_j\|}$.

$\sigma(t)$	$\mathbf{K}_{i,j}$
t	$\mathbf{x}_i^\mathsf{T}\mathbf{x}_j$
$\max(t,0)$	$\frac{1}{2\pi}\|\mathbf{x}_i\|\|\mathbf{x}_j\|\left(\angle \arccos\left(-\angle\right) + \sqrt{1-\angle^2}\right)$
$\|t\|$	$\frac{2}{\pi}\|\mathbf{x}_i\|\|\mathbf{x}_j\|\left(\angle \arcsin\left(\angle\right) + \sqrt{1-\angle^2}\right)$
$\varsigma_+ \max(t,0) +$ $\varsigma_- \max(-t,0)$	$\frac{1}{2}(\varsigma_+^2+\varsigma_-^2)\mathbf{x}_i^\mathsf{T}\mathbf{x}_j + \frac{\|\mathbf{x}_i\|\|\mathbf{x}_j\|}{2\pi}(\varsigma_+ + \varsigma_-)^2\left(\sqrt{1-\angle^2} - \angle\cdot\arccos(\angle)\right)$
$1_{t>0}$	$\frac{1}{2} - \frac{1}{2\pi}\arccos\left(\angle\right)$
$\operatorname{sign}(t)$	$\frac{2}{\pi}\arcsin\left(\angle\right)$
$\varsigma_2 t^2 + \varsigma_1 t + \varsigma_0$	$\varsigma_2^2\left(2\left(\mathbf{x}_i^\mathsf{T}\mathbf{x}_j\right)^2 + \|\mathbf{x}_i\|^2\|\mathbf{x}_j\|^2\right) + \varsigma_1^2\mathbf{x}_i^\mathsf{T}\mathbf{x}_j + \varsigma_2\varsigma_0\left(\|\mathbf{x}_i\|^2+\|\mathbf{x}_j\|^2\right) + \varsigma_0^2$
$\cos(t)$	$\exp\left(-\frac{1}{2}\left(\|\mathbf{x}_i\|^2+\|\mathbf{x}_j\|^2\right)\right)\cosh(\mathbf{x}_i^\mathsf{T}\mathbf{x}_j)$
$\sin(t)$	$\exp\left(-\frac{1}{2}\left(\|\mathbf{x}_i\|^2+\|\mathbf{x}_j\|^2\right)\right)\sinh(\mathbf{x}_i^\mathsf{T}\mathbf{x}_j)$
$\operatorname{erf}(t)$	$\frac{2}{\pi}\arcsin\left(\frac{2\mathbf{x}_i^\mathsf{T}\mathbf{x}_j}{\sqrt{(1+2\|\mathbf{x}_i\|^2)(1+2\|\mathbf{x}_j\|^2)}}\right)$
$\exp(-\frac{t^2}{2})$	$\frac{1}{\sqrt{(1+\|\mathbf{x}_i\|^2)(1+\|\mathbf{x}_j\|^2)-(\mathbf{x}_i^\mathsf{T}\mathbf{x}_j)^2}}$

Nonlinearity in simple random neural networks

Table: $\mathbf{K}_{i,j}$ for commonly used $\sigma(\cdot)$, $\angle \equiv \frac{\mathbf{x}_i^\mathsf{T}\mathbf{x}_j}{\|\mathbf{x}_i\|\|\mathbf{x}_j\|}$.

$\sigma(t)$	$\mathbf{K}_{i,j}$
t	$\mathbf{x}_i^\mathsf{T}\mathbf{x}_j$
$\max(t,0)$	$\frac{1}{2\pi}\|\mathbf{x}_i\|\|\mathbf{x}_j\|\left(\angle \arccos(-\angle) + \sqrt{1-\angle^2}\right)$
$\|t\|$	$\frac{2}{\pi}\|\mathbf{x}_i\|\|\mathbf{x}_j\|\left(\angle \arcsin(\angle) + \sqrt{1-\angle^2}\right)$
$\varsigma_+ \max(t,0)+$ $\varsigma_- \max(-t,0)$	$\frac{1}{2}(\varsigma_+^2+\varsigma_-^2)\mathbf{x}_i^\mathsf{T}\mathbf{x}_j + \frac{\|\mathbf{x}_i\|\|\mathbf{x}_j\|}{2\pi}(\varsigma_+ + \varsigma_-)^2\left(\sqrt{1-\angle^2} - \angle\cdot\arccos(\angle)\right)$
$1_{t>0}$	$\frac{1}{2} - \frac{1}{2\pi}\arccos(\angle)$
$\operatorname{sign}(t)$	$\frac{2}{\pi}\arcsin(\angle)$
$\varsigma_2 t^2 + \varsigma_1 t + \varsigma_0$	$\varsigma_2^2\left(2\left(\mathbf{x}_i^\mathsf{T}\mathbf{x}_j\right)^2 + \|\mathbf{x}_i\|^2\|\mathbf{x}_j\|^2\right) + \varsigma_1^2\mathbf{x}_i^\mathsf{T}\mathbf{x}_j + \varsigma_2\varsigma_0\left(\|\mathbf{x}_i\|^2+\|\mathbf{x}_j\|^2\right) + \varsigma_0^2$
$\cos(t)$	$\exp\left(-\frac{1}{2}\left(\|\mathbf{x}_i\|^2+\|\mathbf{x}_j\|^2\right)\right)\cosh(\mathbf{x}_i^\mathsf{T}\mathbf{x}_j)$
$\sin(t)$	$\exp\left(-\frac{1}{2}\left(\|\mathbf{x}_i\|^2+\|\mathbf{x}_j\|^2\right)\right)\sinh(\mathbf{x}_i^\mathsf{T}\mathbf{x}_j)$
$\operatorname{erf}(t)$	$\frac{2}{\pi}\arcsin\left(\frac{2\mathbf{x}_i^\mathsf{T}\mathbf{x}_j}{\sqrt{(1+2\|\mathbf{x}_i\|^2)(1+2\|\mathbf{x}_j\|^2)}}\right)$
$\exp(-\frac{t^2}{2})$	$\frac{1}{\sqrt{(1+\|\mathbf{x}_i\|^2)(1+\|\mathbf{x}_j\|^2)-(\mathbf{x}_i^\mathsf{T}\mathbf{x}_j)^2}}$

$\Rightarrow$(still) highly nonlinear functions of the data $\mathbf{x}$!

Dig Deeper into $\mathbf{K}$

Data: K-class Gaussian mixture model

$$\mathbf{x}_i \in \mathcal{C}_a \Leftrightarrow \mathbf{x}_i = \boldsymbol{\mu}_a/\sqrt{p} + \mathbf{z}_i$$

with $\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_a/p)$, $a = 1, \ldots, K$ of statistical mean $\boldsymbol{\mu}_a$ and covariance $\mathbf{C}_a$.

Dig Deeper into $\mathbf{K}$

Data: K-class Gaussian mixture model

$$\mathbf{x}_i \in \mathcal{C}_a \Leftrightarrow \mathbf{x}_i = \boldsymbol{\mu}_a/\sqrt{p} + \mathbf{z}_i$$

with $\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_a/p)$, $a = 1, \ldots, K$ of statistical mean $\boldsymbol{\mu}_a$ and covariance $\mathbf{C}_a$.

Non-trivial classification (again)

For p large, $\|\boldsymbol{\mu}_a - \boldsymbol{\mu}_b\| = O(1)$, $\|\mathbf{C}_a\| = O(1)$ and $\operatorname{tr}(\mathbf{C}_a - \mathbf{C}_b) = O(\sqrt{p})$.

Dig Deeper into $\mathbf{K}$

Data: K-class Gaussian mixture model

$$\mathbf{x}_i \in \mathcal{C}_a \Leftrightarrow \mathbf{x}_i = \boldsymbol{\mu}_a/\sqrt{p} + \mathbf{z}_i$$

with $\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_a/p)$, $a = 1, \ldots, K$ of statistical mean $\boldsymbol{\mu}_a$ and covariance $\mathbf{C}_a$.

Non-trivial classification (again)

For p large, $\|\boldsymbol{\mu}_a - \boldsymbol{\mu}_b\| = O(1)$, $\|\mathbf{C}_a\| = O(1)$ and $\operatorname{tr}(\mathbf{C}_a - \mathbf{C}_b) = O(\sqrt{p})$.

As a consequence,

$$\|\mathbf{x}_i\|^2 = \underbrace{\|\mathbf{z}_i\|^2}_{O(1)} + \underbrace{\|\boldsymbol{\mu}_a\|^2/p + 2\boldsymbol{\mu}_a^\mathsf{T}\mathbf{z}_i/\sqrt{p}}_{O(p^{-1})}$$

Dig Deeper into $\mathbf{K}$

Data: K-class Gaussian mixture model

$$\mathbf{x}_i \in \mathcal{C}_a \Leftrightarrow \mathbf{x}_i = \boldsymbol{\mu}_a/\sqrt{p} + \mathbf{z}_i$$

with $\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_a/p)$, $a = 1, \ldots, K$ of statistical mean $\boldsymbol{\mu}_a$ and covariance $\mathbf{C}_a$.

Non-trivial classification (again)

For p large, $\|\boldsymbol{\mu}_a - \boldsymbol{\mu}_b\| = O(1)$, $\|\mathbf{C}_a\| = O(1)$ and $\operatorname{tr}(\mathbf{C}_a - \mathbf{C}_b) = O(\sqrt{p})$.

As a consequence,

$$\begin{aligned} \|\mathbf{x}_i\|^2 &= \underbrace{\|\mathbf{z}_i\|^2}_{O(1)} + \underbrace{\|\boldsymbol{\mu}_a\|^2/p + 2\boldsymbol{\mu}_a^{\mathsf{T}}\mathbf{z}_i/\sqrt{p}}_{O(p^{-1})} \\ &= \underbrace{\operatorname{tr}\mathbf{C}_a/p}_{O(1)} + \underbrace{\|\mathbf{z}_i\|^2 - \operatorname{tr}\mathbf{C}_a/p}_{O(p^{-1/2})} + \underbrace{\|\boldsymbol{\mu}_a\|^2/p + 2\boldsymbol{\mu}_a^{\mathsf{T}}\mathbf{z}_i/\sqrt{p}}_{O(p^{-1})} \end{aligned}$$

Dig Deeper into $\mathbf{K}$

Data: K-class Gaussian mixture model

$$\mathbf{x}_i \in \mathcal{C}_a \Leftrightarrow \mathbf{x}_i = \boldsymbol{\mu}_a/\sqrt{p} + \mathbf{z}_i$$

with $\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_a/p)$, $a = 1, \ldots, K$ of statistical mean $\boldsymbol{\mu}_a$ and covariance $\mathbf{C}_a$.

Non-trivial classification (again)

For p large, $\|\boldsymbol{\mu}_a - \boldsymbol{\mu}_b\| = O(1)$, $\|\mathbf{C}_a\| = O(1)$ and $\operatorname{tr}(\mathbf{C}_a - \mathbf{C}_b) = O(\sqrt{p})$.

As a consequence,

$$\begin{aligned}
\|\mathbf{x}_i\|^2 &= \underbrace{\|\mathbf{z}_i\|^2}_{O(1)} + \underbrace{\|\boldsymbol{\mu}_a\|^2/p + 2\boldsymbol{\mu}_a^\mathsf{T}\mathbf{z}_i/\sqrt{p}}_{O(p^{-1})} \\
&= \underbrace{\operatorname{tr}\mathbf{C}_a/p}_{O(1)} + \underbrace{\|\mathbf{z}_i\|^2 - \operatorname{tr}\mathbf{C}_a/p}_{O(p^{-1/2})} + \underbrace{\|\boldsymbol{\mu}_a\|^2/p + 2\boldsymbol{\mu}_a^\mathsf{T}\mathbf{z}_i/\sqrt{p}}_{O(p^{-1})}
\end{aligned}$$

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

Dig Deeper into $\mathbf{K}$

Data: K-class Gaussian mixture model

$$\mathbf{x}_i \in \mathcal{C}_a \Leftrightarrow \mathbf{x}_i = \boldsymbol{\mu}_a/\sqrt{p} + \mathbf{z}_i$$

with $\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_a/p)$, $a = 1, \ldots, K$ of statistical mean $\boldsymbol{\mu}_a$ and covariance $\mathbf{C}_a$.

Non-trivial classification (again)

For p large, $\|\boldsymbol{\mu}_a - \boldsymbol{\mu}_b\| = O(1)$, $\|\mathbf{C}_a\| = O(1)$ and $\operatorname{tr}(\mathbf{C}_a - \mathbf{C}_b) = O(\sqrt{p})$.

As a consequence,

$$\begin{aligned}\|\mathbf{x}_i\|^2 &= \underbrace{\|\mathbf{z}_i\|^2}_{O(1)} + \underbrace{\|\boldsymbol{\mu}_a\|^2/p + 2\boldsymbol{\mu}_a^\mathsf{T}\mathbf{z}_i/\sqrt{p}}_{O(p^{-1})} \\ &= \underbrace{\operatorname{tr}\mathbf{C}_a/p}_{O(1)} + \underbrace{\|\mathbf{z}_i\|^2 - \operatorname{tr}\mathbf{C}_a/p}_{O(p^{-1/2})} + \underbrace{\|\boldsymbol{\mu}_a\|^2/p + 2\boldsymbol{\mu}_a^\mathsf{T}\mathbf{z}_i/\sqrt{p}}_{O(p^{-1})}\end{aligned}$$

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

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

Dig Deeper into $\mathbf{K}$

Data: K-class Gaussian mixture model

$$\mathbf{x}_i \in \mathcal{C}_a \Leftrightarrow \mathbf{x}_i = \boldsymbol{\mu}_a/\sqrt{p} + \mathbf{z}_i$$

with $\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_a/p)$, $a = 1, \ldots, K$ of statistical mean $\boldsymbol{\mu}_a$ and covariance $\mathbf{C}_a$.

Non-trivial classification (again)

For p large, $\|\boldsymbol{\mu}_a - \boldsymbol{\mu}_b\| = O(1)$, $\|\mathbf{C}_a\| = O(1)$ and $\operatorname{tr}(\mathbf{C}_a - \mathbf{C}_b) = O(\sqrt{p})$.

As a consequence,

$$\begin{aligned}\|\mathbf{x}_i\|^2 &= \underbrace{\|\mathbf{z}_i\|^2}_{O(1)} + \underbrace{\|\boldsymbol{\mu}_a\|^2/p + 2\boldsymbol{\mu}_a^\mathsf{T}\mathbf{z}_i/\sqrt{p}}_{O(p^{-1})} \\ &= \underbrace{\operatorname{tr}\mathbf{C}_a/p}_{O(1)} + \underbrace{\|\mathbf{z}_i\|^2 - \operatorname{tr}\mathbf{C}_a/p}_{O(p^{-1/2})} + \underbrace{\|\boldsymbol{\mu}_a\|^2/p + 2\boldsymbol{\mu}_a^\mathsf{T}\mathbf{z}_i/\sqrt{p}}_{O(p^{-1})}\end{aligned}$$

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

$$\Rightarrow \|\mathbf{x}_i\|^2 = \tau + O(p^{-1/2}) \text{ with } \tau \equiv \operatorname{tr}(\mathbf{C}^\circ)/p, \quad \|\mathbf{x}_i - \mathbf{x}_j\|^2 \approx 2\tau!$$

Understand nonlinearity in random neural networks

Asymptotic Equivalent of $\mathbf{K}$

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

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

almost surely, with

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

and

$$\mathbf{U} \equiv \left[\frac{\mathbf{J}}{\sqrt{p}}, \boldsymbol{\phi}\right], \quad \mathbf{B} \equiv \begin{bmatrix} \mathbf{t}\mathbf{t}^\mathsf{T} + 2\mathbf{S} & \mathbf{t} \\ \mathbf{t}^\mathsf{T} & 1 \end{bmatrix}.$$

Understand nonlinearity in random neural networks

Asymptotic Equivalent of $\mathbf{K}$

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

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

almost surely, with

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

and

$$\mathbf{U} \equiv \left[\frac{\mathbf{J}}{\sqrt{p}}, \boldsymbol{\phi} \right], \quad \mathbf{B} \equiv \begin{bmatrix} \mathbf{t}\mathbf{t}^\mathsf{T} + 2\mathbf{S} & \mathbf{t} \\ \mathbf{t}^\mathsf{T} & 1 \end{bmatrix}.$$

$\mathbf{J} \equiv [\mathbf{j}_1, \ldots, \mathbf{j}_K]$, $\mathbf{j}_a$ canonical vector of $\mathcal{C}_a$, weighted by $\mathbf{z}$, $\boldsymbol{\phi}$ random fluctuations of data and $\mathbf{M} \equiv [\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K]$, $\mathbf{t} \equiv \left\{ \operatorname{tr} \mathbf{C}_a^\circ / \sqrt{p} \right\}_{a=1}^K$, $\mathbf{S} \equiv \{ \operatorname{tr}(\mathbf{C}_a \mathbf{C}_b)/p \}_{a,b=1}^K$ the statistical information.

Understand nonlinearity in random neural networks

Asymptotic Equivalent of $\mathbf{K}$

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

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

almost surely, with

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

and

$$\mathbf{U} \equiv \left[\frac{\mathbf{J}}{\sqrt{p}}, \boldsymbol{\phi} \right], \quad \mathbf{B} \equiv \begin{bmatrix} \mathbf{t}\mathbf{t}^\mathsf{T} + 2\mathbf{S} & \mathbf{t} \\ \mathbf{t}^\mathsf{T} & 1 \end{bmatrix}.$$

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

$\sigma(t)$	d_1	d_2
t	1	0
$\max(t,0)$	$\frac{1}{4}$	$\frac{1}{8\pi\tau}$
$\lvert t\rvert$	0	$\frac{1}{2\pi\tau}$
$1_{t>0}$	$\frac{1}{2\pi\tau}$	0
$\operatorname{sign}(t)$	$\frac{2}{\pi\tau}$	0
$\varsigma_2 t^2 + \varsigma_1 t + \varsigma_0$	ς_1^2	ς_2^2
$\cos(t)$	0	$\frac{e^{-\tau}}{4}$
$\sin(t)$	$e^{-\tau}$	0
$\operatorname{erf}(t)$	$\frac{4}{\pi}\frac{1}{2\tau+1}$	0
$\exp(-\frac{t^2}{2})$	0	$\frac{1}{4(\tau+1)^3}$

$\mathbf{J} \equiv [\mathbf{j}_1, \ldots, \mathbf{j}_K]$, $\mathbf{j}_a$ canonical vector of $\mathcal{C}_a$, weighted by $\mathbf{z}$, $\boldsymbol{\phi}$ random fluctuations of data and $\mathbf{M} \equiv [\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K]$, $\mathbf{t} \equiv \left\{ \operatorname{tr} \mathbf{C}_a^\circ / \sqrt{p} \right\}_{a=1}^K$, $\mathbf{S} \equiv \{\operatorname{tr}(\mathbf{C}_a \mathbf{C}_b)/p\}_{a,b=1}^K$ the statistical information.

Consequence

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

$\sigma(t)$	d_1	d_2
t	1	0
$\max(t,0)$	$\frac{1}{4}$	$\frac{1}{8\pi\tau}$
$\lvert t\rvert$	0	$\frac{1}{2\pi\tau}$
$1_{t>0}$	$\frac{1}{2\pi\tau}$	0
$\operatorname{sign}(t)$	$\frac{2}{\pi\tau}$	0
$\varsigma_2 t^2 + \varsigma_1 t + \varsigma_0$	ς_1^2	ς_2^2
$\cos(t)$	0	$\frac{e^{-\tau}}{4}$
$\sin(t)$	$e^{-\tau}$	0
$\operatorname{erf}(t)$	$\frac{4}{\pi}\frac{1}{2\tau+1}$	0
$\exp(-\frac{t^2}{2})$	0	$\frac{1}{4(\tau+1)^3}$

Consequence

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

$\sigma(t)$	d_1	d_2
t	1	0
$\max(t,0)$	$\frac{1}{4}$	$\frac{1}{8\pi\tau}$
$\|t\|$	0	$\frac{1}{2\pi\tau}$
$1_{t>0}$	$\frac{1}{2\pi\tau}$	0
$\operatorname{sign}(t)$	$\frac{2}{\pi\tau}$	0
$\varsigma_2 t^2 + \varsigma_1 t + \varsigma_0$	ς_1^2	ς_2^2
$\cos(t)$	0	$\frac{e^{-\tau}}{4}$
$\sin(t)$	$e^{-\tau}$	0
$\operatorname{erf}(t)$	$\frac{4}{\pi}\frac{1}{2\tau+1}$	0
$\exp(-\frac{t^2}{2})$	0	$\frac{1}{4(\tau+1)^3}$

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

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

Consequence

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

$\sigma(t)$	d_1	d_2
t	1	0
$\max(t,0)$	$\frac{1}{4}$	$\frac{1}{8\pi\tau}$
$\lvert t\rvert$	0	$\frac{1}{2\pi\tau}$
$1_{t>0}$	$\frac{1}{2\pi\tau}$	0
$\operatorname{sign}(t)$	$\frac{2}{\pi\tau}$	0
$\varsigma_2 t^2 + \varsigma_1 t + \varsigma_0$	ς_1^2	ς_2^2
$\cos(t)$	0	$\frac{e^{-\tau}}{4}$
$\sin(t)$	$e^{-\tau}$	0
$\operatorname{erf}(t)$	$\frac{4}{\pi}\frac{1}{2\tau+1}$	0
$\exp(-\frac{t^2}{2})$	0	$\frac{1}{4(\tau+1)^3}$

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

- *mean-oriented*, $d_1 \neq 0$, $d_2 = 0$: t, $1_{t>0}$, $\operatorname{sign}(t)$, $\sin(t)$ and $\operatorname{erf}(t)$ $\Rightarrow$ separate with difference in $\mathbf{M}$;
- *cov-oriented*, $d_1 = 0$, $d_2 \neq 0$: $|t|$, $\cos(t)$ and $\exp(-t^2/2)$ $\Rightarrow$ track differences in cov $\mathbf{t}$, $\mathbf{S}$;

Consequence

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

$\sigma(t)$	d_1	d_2
t	1	0
$\max(t,0)$	$\frac{1}{4}$	$\frac{1}{8\pi\tau}$
$\lvert t\rvert$	0	$\frac{1}{2\pi\tau}$
$1_{t>0}$	$\frac{1}{2\pi\tau}$	0
$\operatorname{sign}(t)$	$\frac{2}{\pi\tau}$	0
$\varsigma_2 t^2 + \varsigma_1 t + \varsigma_0$	ς_1^2	ς_2^2
$\cos(t)$	0	$\frac{e^{-\tau}}{4}$
$\sin(t)$	$e^{-\tau}$	0
$\operatorname{erf}(t)$	$\frac{4}{\pi}\frac{1}{2\tau+1}$	0
$\exp(-\frac{t^2}{2})$	0	$\frac{1}{4(\tau+1)^3}$

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

- *mean-oriented*, $d_1 \neq 0$, $d_2 = 0$: t, $1_{t>0}$, $\operatorname{sign}(t)$, $\sin(t)$ and $\operatorname{erf}(t)$ $\Rightarrow$ separate with difference in $\mathbf{M}$;
- *cov-oriented*, $d_1 = 0$, $d_2 \neq 0$: $|t|$, $\cos(t)$ and $\exp(-t^2/2)$ $\Rightarrow$ track differences in cov $\mathbf{t}$, $\mathbf{S}$;
- "*balanced*", both $d_1, d_2 \neq 0$:
  - ReLU function $\max(t,0)$,
  - quadratic function $\varsigma_2 t^2 + \varsigma_1 t + \varsigma_0$.

Consequence

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

$\sigma(t)$	d_1	d_2
t	1	0
$\max(t,0)$	$\frac{1}{4}$	$\frac{1}{8\pi\tau}$
$\|t\|$	0	$\frac{1}{2\pi\tau}$
$1_{t>0}$	$\frac{1}{2\pi\tau}$	0
$\operatorname{sign}(t)$	$\frac{2}{\pi\tau}$	0
$\varsigma_2 t^2 + \varsigma_1 t + \varsigma_0$	ς_1^2	ς_2^2
$\cos(t)$	0	$\frac{e^{-\tau}}{4}$
$\sin(t)$	$e^{-\tau}$	0
$\operatorname{erf}(t)$	$\frac{4}{\pi}\frac{1}{2\tau+1}$	0
$\exp(-\frac{t^2}{2})$	0	$\frac{1}{4(\tau+1)^3}$

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

- *mean-oriented*, $d_1 \neq 0$, $d_2 = 0$: t, $1_{t>0}$, $\operatorname{sign}(t)$, $\sin(t)$ and $\operatorname{erf}(t)$ $\Rightarrow$ separate with difference in $\mathbf{M}$;
- *cov-oriented*, $d_1 = 0$, $d_2 \neq 0$: $|t|$, $\cos(t)$ and $\exp(-t^2/2)$ $\Rightarrow$ track differences in cov $\mathbf{t}$, $\mathbf{S}$;
- "*balanced*", both $d_1, d_2 \neq 0$:
  - ReLU function $\max(t,0)$,
  - quadratic function $\varsigma_2 t^2 + \varsigma_1 t + \varsigma_0$.

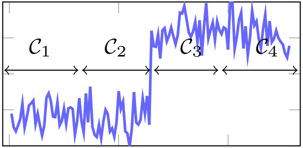
$\Rightarrow$ make use of both statistics!

Numerical Validations: Gaussian Data

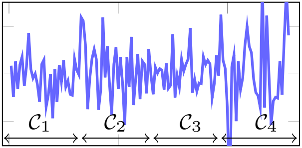
Example: Gaussian mixture data of four classes: $\mathcal{N}(\boldsymbol{\mu}_1, \mathbf{C}_1)$, $\mathcal{N}(\boldsymbol{\mu}_1, \mathbf{C}_2)$, $\mathcal{N}(\boldsymbol{\mu}_2, \mathbf{C}_1)$ and $\mathcal{N}(\boldsymbol{\mu}_2, \mathbf{C}_2)$ with different $\sigma(\cdot)$ functions.

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

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



Eigenvector 1

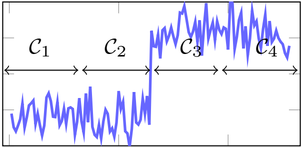


Eigenvector 2

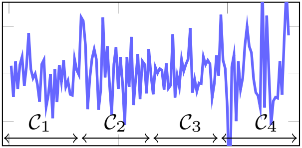
Numerical Validations: Gaussian Data

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

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

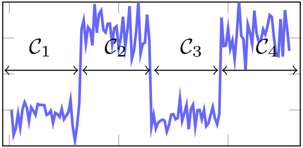


Eigenvector 1

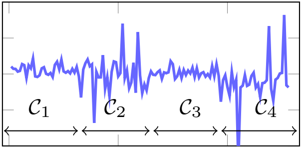


Eigenvector 2

Case 2: $\sigma(t) = |t|$.

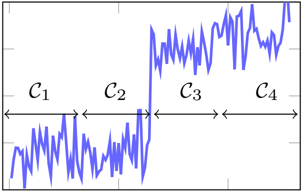


Eigenvector 1

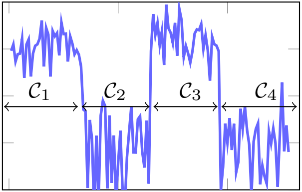


Eigenvector 2

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



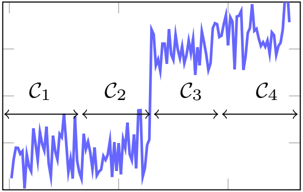
Eigenvector 1



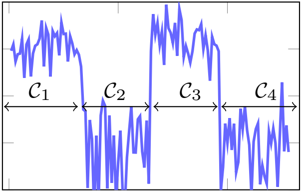
Eigenvector 2

Numerical Validations: Gaussian Data

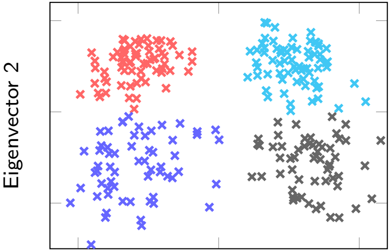
Case 3: the ReLU function $\sigma(t) = \max(t, 0)$.



Eigenvector 1



Eigenvector 2



Eigenvector 1

Numerical Validations: Real Datasets

Figure: The MNIST image database.

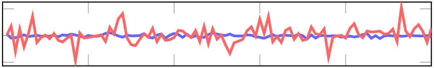


Figure: The epileptic EEG datasets.[1]

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.

	$\|\mathbf{M}^\mathsf{T}\mathbf{M}\|$	$\|\mathbf{t}\mathbf{t}^\mathsf{T} + 2\mathbf{S}\|$
MNIST data	172.4	86.0
EEG data	1.2	182.7

Numerical Validations: Real Datasets

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

	$\|\mathbf{M}^\mathsf{T}\mathbf{M}\|$	$\|\mathbf{t}\mathbf{t}^\mathsf{T} + 2\mathbf{S}\|$
MNIST data	172.4	86.0
EEG data	1.2	182.7

Table: Clustering accuracies on MNIST.

	$\sigma(t)$	$n = 64$	$n = 128$
mean-oriented	t	**88.94%**	87.30%
	$1_{t>0}$	82.94%	85.56%
	$\operatorname{sign}(t)$	83.34%	85.22%
	$\sin(t)$	87.81%	**87.50%**
cov-oriented	$\|t\|$	60.41%	57.81%
	$\cos(t)$	59.56%	57.72%
	$\exp(-t^2/2)$	60.44%	58.67%
balanced	$\operatorname{ReLU}(t)$	85.72%	82.27%

Table: Clustering accuracies on EEG.

	$\sigma(t)$	$n = 64$	$n = 128$
mean-oriented	t	70.31%	69.58%
	$1_{t>0}$	65.87%	63.47%
	$\operatorname{sign}(t)$	64.63%	63.03%
	$\sin(t)$	70.34%	68.22%
cov-oriented	$\|t\|$	99.69%	99.50%
	$\cos(t)$	99.38%	99.36%
	$\exp(-t^2/2)$	**99.81%**	**99.77%**
balanced	$\operatorname{ReLU}(t)$	87.91%	90.97%

Outline

From theory to practice: concentrated random vectors

RMT often assumes $\mathbf{x}_i$ are affine maps $\mathbf{A}\mathbf{z}_i + \mathbf{b}$ of $\mathbf{z}_i \in \mathbb{R}^p$ with i.i.d. entries.

From theory to practice: concentrated random vectors

RMT often assumes $\mathbf{x}_i$ are affine maps $\mathbf{A}\mathbf{z}_i + \mathbf{b}$ of $\mathbf{z}_i \in \mathbb{R}^p$ with i.i.d. entries.

Concentrated random vectors

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

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

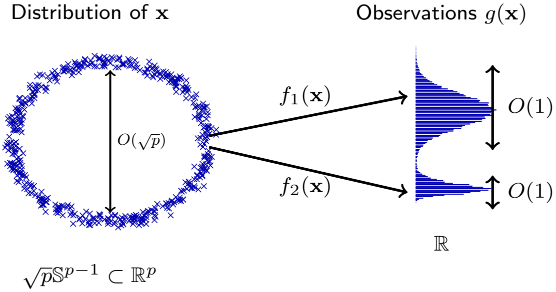
From theory to practice: concentrated random vectors

RMT often assumes $\mathbf{x}_i$ are affine maps $\mathbf{A}\mathbf{z}_i + \mathbf{b}$ of $\mathbf{z}_i \in \mathbb{R}^p$ with i.i.d. entries.

Concentrated random vectors

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

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



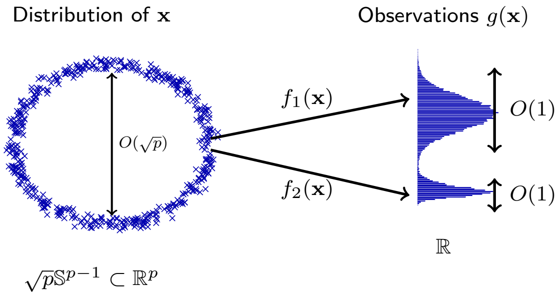
From theory to practice: concentrated random vectors

RMT often assumes $\mathbf{x}_i$ are affine maps $\mathbf{A}\mathbf{z}_i + \mathbf{b}$ of $\mathbf{z}_i \in \mathbb{R}^p$ with i.i.d. entries.

Concentrated random vectors

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

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



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

From concentrated random vectors to GANs

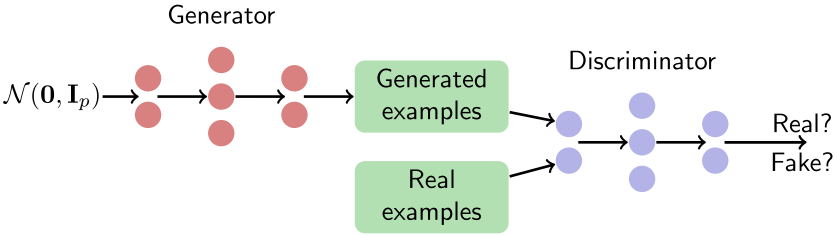


Figure: Illustration of a generative adversarial network (GAN).

From concentrated random vectors to GANs

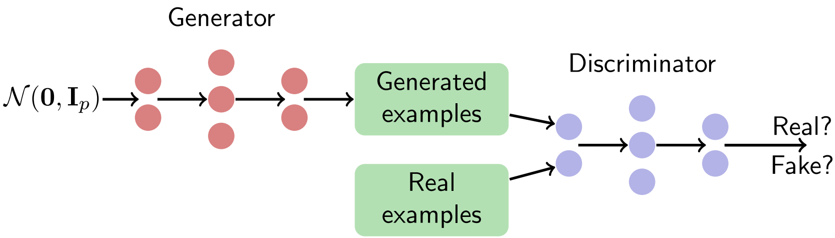


Figure: Illustration of a generative adversarial network (GAN).

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

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

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

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?

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?

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?

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!

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

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*)

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

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)

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

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

R. Couillet, F. Benaych-Georges, "Kernel Spectral Clustering of Large Dimensional Data", Electronic Journal of Statistics, vol. 10, no. 1, pp. 1393-1454, 2016.

Z. Liao, R. Couillet, "Random matrices meet machine learning: a large dimensional analysis of LS-SVM", IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'17), New Orleans, USA, 2017.

X. Mai, R. Couillet, "The counterintuitive mechanism of graph-based semi-supervised learning in the big data regime", IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'17), New Orleans, USA, 2017.

X. Mai, R. Couillet, "A random matrix analysis and improvement of semi-supervised learning for large dimensional data", Journal of Machine Learning Research, 2018.

Z. Liao, R. Couillet, "A Large Dimensional Analysis of Least Squares Support Vector Machines", IEEE Transactions on Signal Processing 67 (4), 1065-1074, 2019.

X. Mai, Z. Liao, R. Couillet, "A Large Scale Analysis of Logistic Regression: Asymptotic Performance and New Insights", IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'19), Brighton, UK, 2019.

Summary of Results and Perspectives

Neural Networks: References

- R. Couillet, G. Wainrib, H. Sevi, H. Tiomoko Ali, "The asymptotic performance of linear echo state neural networks", Journal of Machine Learning Research, vol. 17, no. 178, pp. 1-35, 2016.
- C. Louart, R. Couillet, "Harnessing neural networks: a random matrix approach", IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'17), New Orleans, USA, 2017.
- C. Louart, R. Couillet, "A Random Matrix and Concentration Inequalities Framework for Neural Networks Analysis", IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'18), Calgary, Canada, 2018.
- C. Louart, Z. Liao, R. Couillet, "A Random Matrix Approach to Neural Networks", The Annals of Applied Probability, vol. 28, no. 2, pp. 1190-1248, 2018.
- Z. Liao, R. Couillet, "The Dynamics of Learning: A Random Matrix Approach", International Conference on Machine Learning (ICML), Stockholm, Sweden, 2018.
- Z. Liao, R. Couillet, "On the Spectrum of Random Features Maps of High Dimensional Data", International Conference on Machine Learning (ICML), Stockholm, Sweden, 2018.

Thank you!

For more information, please visit

Thank you!

For more information, please visit

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

Thank you!

For more information, please visit

- https://zhenyu-liao.github.io;
- http://romaincouillet.hebfree.org.