Unsupervised learning:
symmetric low-rank matrix estimation, community detection and triplet loss.

Statistical physics and machine learning back together

Cargèse, August, 2018

Marc Lelarge
Unsupervised learning

**Unsupervised machine learning** is the machine learning task of inferring a function that describes the structure of "unlabeled" data (i.e. data that has not been classified or categorized). Since the examples given to the learning algorithm are unlabeled, there is no straightforward way to evaluate the accuracy of the structure that is produced by the algorithm.

Source:
Overview

- Theoretical approach
- Data driven approach
Overview

- **Theoretical approach**
  - Make assumptions, i.e. take a model for the data, and prove theorems.

- **Data driven approach**
  - Take a dataset and experiment!
Overview

- **Theoretical approach**
  - Make assumptions, i.e. take a model for the data, and prove theorems.

- **Data driven approach**
  - Take a dataset and experiment!

- **Common themes: graphs**
Overview

- **Theoretical approach**
  - Make assumptions, i.e. take a model for the data, and prove theorems.
    - Low-rank matrix estimation
    - Community detection

- **Data driven approach**
  - Take a dataset and experiment!
    - Unsupervised feature learning with deep neural network

- **Common themes: graphs**
Low-rank matrix estimation

“Spiked Wigner” model

\[
\mathbf{Y} = \sqrt{\frac{\lambda}{n}} \mathbf{X} \mathbf{X}^\top + \mathbf{Z}
\]

- \( \mathbf{X} \): vector of dimension \( n \) with entries \( X_i \overset{\text{i.i.d.}}{\sim} P_0 \). \( \mathbb{E}X_1 = 0, \mathbb{E}X_1^2 = 1 \).
- \( Z_{i,j} = Z_{j,i} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1) \).
- \( \lambda \): signal-to-noise ratio.
- \( \lambda \) and \( P_0 \) are known by the statistician.

**Goal:** recover the low-rank matrix \( \mathbf{X} \mathbf{X}^\top \) from \( \mathbf{Y} \).
Principal component analysis (PCA)

B.B.P. phase transition

Spectral estimator:
Estimate $\mathbf{X}$ using the eigenvector $\hat{\mathbf{x}}_n$ associated with the largest eigenvalue $\mu_n$ of $\mathbf{Y}/\sqrt{n}$.

---

Principal component analysis (PCA)

B.B.P. phase transition

**Spectral estimator:**
Estimate $X$ using the eigenvector $\hat{x}_n$ associated with the largest eigenvalue $\mu_n$ of $Y/\sqrt{n}$.

B.B.P. phase transition

- if $\lambda \leq 1$
  \[
  \begin{align*}
  \mu_n & \xrightarrow{a.s.} 2 \\
  X \cdot \hat{x}_n & \xrightarrow{a.s.} 0 \\
  \end{align*}
  \]
- if $\lambda > 1$
  \[
  \begin{align*}
  \mu_n & \xrightarrow{n \to \infty} \sqrt{\lambda} + \frac{1}{\sqrt\lambda} > 2 \\
  |X \cdot \hat{x}_n| & \xrightarrow{n \to \infty} \sqrt{1 - 1/\lambda} > 0 \\
  \end{align*}
  \]

Baik et al., 2005; Benaych-Georges and Nadakuditi, 2011

---

- PCA fails when $\lambda \leq 1$, but is it still possible to recover the signal?
Questions

- PCA fails when $\lambda \leq 1$, but is it still possible to recover the signal?
- When $\lambda > 1$, is PCA optimal?
Questions

- PCA fails when $\lambda \leq 1$, but is it still possible to recover the signal?
- When $\lambda > 1$, is PCA optimal?
- More generally, what is the best achievable estimation performance in both regimes?
MMSE and information-theoretic threshold

Definitions

“MMSE” = Minimal Mean Square Error

\[
\text{MMSE}_n = \min_{\hat{\theta}} \frac{1}{n^2} \mathbb{E} \left\| \mathbf{X}\mathbf{X}^\top - \hat{\theta}(\mathbf{Y}) \right\|^2
\]

\[
= \frac{1}{n^2} \sum_{1 \leq i,j \leq n} (X_i X_j - \mathbb{E}[X_i X_j | \mathbf{Y}])^2 \leq \mathbb{E}_{\mathcal{P}_0}[X_1^2]^2
\]

The information-theoretic threshold is the critical value \( \lambda_c \) such that

- if \( \lambda > \lambda_c \), \( \lim_{n \to \infty} \text{MMSE}_n < \text{Dummy MSE} \)
- if \( \lambda < \lambda_c \), \( \lim_{n \to \infty} \text{MMSE}_n = \text{Dummy MSE} \)
Related work
A short overview

- **Approximate Message Passing (AMP) algorithms:** Rangan and Fletcher, 2012, Deshpande and Montanari, 2014; Lesieur et al., 2015b allows to derive the MMSE when AMP is optimal.

- In presence of a “hard phase”, Barbier et al., 2016 uses AMP and spatial coupling techniques to compute the MMSE under some additional assumptions.

- Banks et al., 2016; Perry et al., 2016 obtained bounds on the information-theoretic threshold by second moment computations and contiguity.
Main result
Limiting formula for the MMSE

Theorem

$$\text{MMSE}_n \xrightarrow{n \to \infty} \underbrace{\mathbb{E}_{P_0}[X^2]^2}_{\text{Dummy MSE}} - q^*(\lambda)^2$$

where $q^*(\lambda)$ is the maximizer of

$$q \geq 0 \mapsto \mathbb{E}_{X_0 \sim P_0, Z_0 \sim \mathcal{N}} \left[ \log \int_{x_0} dP_0(x_0) e^{\sqrt{\lambda qZ_0x_0 + \lambda qX_0x_0 - \frac{\lambda q}{2} x_0^2}} \right] - \frac{\lambda}{4} q^2$$

Lelarge and Miolane, 2016 joint work with Léo Miolane.

This is the same formula as in Jean Barbier’s talk, i.e. Replica Symmetric formula!
Proof ideas

A planted spin system

\[ P(X = x \mid Y) = \frac{1}{Z_n} P_0(x) e^{H_n(x)} \text{ where} \]

\[ H_n(x) = \sum_{i<j} \sqrt{\frac{\lambda}{n}} Y_{i,j} x_i x_j - \frac{\lambda}{2n} x_i^2 x_j^2 \]

\[ = \sum_{i<j} \sqrt{\frac{\lambda}{n}} Z_{i,j} x_i x_j + \frac{\lambda}{n} X_i X_j x_i x_j - \frac{\lambda}{2n} x_i^2 x_j^2 \]

\[ \underbrace{\text{SK}}_{\text{planting}} \]
Proof ideas

A planted spin system

\[ \mathbb{P}(X = x \mid Y) = \frac{1}{Z_n} P_0(x) e^{H_n(x)} \]

where

\[ H_n(x) = \sum_{i<j} \sqrt{\frac{\lambda}{n}} Y_{i,j} x_i x_j - \frac{\lambda}{2n} x_i^2 x_j^2 \]

\[ = \sum_{i<j} \sqrt{\frac{\lambda}{n}} Z_{i,j} x_i x_j + \frac{\lambda}{n} X_i X_j x_i x_j - \frac{\lambda}{2n} x_i^2 x_j^2 \]

\[ \text{SK} \quad \text{planting} \]

Lower bound: Guerra’s interpolation technique. Adapted in Korada and Macris, 2009; Krzakala et al., 2016.

\[
\begin{align*}
\{ & Y = \sqrt{t} \sqrt{\frac{\lambda}{n}} X X^T \quad + \quad Z \\
& Y' = \sqrt{1-t} \sqrt{\lambda} \quad X \quad + \quad Z' 
\end{align*}
\]
Proof ideas

Upper bound: cavity computations and the scalar channel

Cavity computations (Mézard et al., 1987) in physics = in mathematics

Aizenman-Sims-Starr scheme: Aizenman et al., 2003; Talagrand, 2010 to compute the limit of the free energy

$$F_n = \frac{1}{n} \mathbb{E} \log Z_n$$

because

$$\text{Constant} - F_n = \frac{1}{n} I(X; Y) \xrightarrow{\partial \lambda} \text{MMSE}$$

I-MMSE theorem
Proof ideas

Upper bound: cavity computations and the scalar channel

Cavity computations (Mézard et al., 1987) in physics = in mathematics

Aizenman-Sims-Starr scheme: Aizenman et al., 2003; Talagrand, 2010 to compute the limit of the free energy $F_n = \frac{1}{n} \mathbb{E} \log Z_n$ because

$$\text{Constant} - F_n = \frac{1}{n} I(X; Y) \xrightarrow{\text{I-MMSE theorem}} \text{MMSE}$$

Lesieur et al., 2015a conjectured that the problem is characterized par the scalar channel:

$$Y_0 = \sqrt{\gamma} X_0 + Z_0$$

and the scalar free energy: $\mathcal{F}(\gamma) = \mathbb{E} \left[ \log \sum_{x_0} P_0(x_0) e^{\sqrt{\gamma} Y_0 x_0 - \frac{\gamma}{2} x_0^2} \right]$
Proof ideas

Upper bound: cavity computations and the scalar channel

Cavity computations (Mézard et al., 1987) in physics = in mathematics

Aizenman-Sims-Starr scheme: Aizenman et al., 2003; Talagrand, 2010 to compute the limit of the free energy $F_n = \frac{1}{n}E \log Z_n$ because

$$\text{Constant} - F_n = \frac{1}{n} I(X; Y) \xrightarrow{\partial \lambda} \text{MMSE}$$

Lesieur et al., 2015a conjectured that the problem is characterized par the scalar channel:

$$Y_0 = \sqrt{\gamma} X_0 + Z_0$$

and the scalar free energy: $F(\gamma) = E \left[ \log \sum_{x_0} P_0(x_0) e^{\sqrt{\gamma} Y_0 x_0 - \frac{\gamma}{2} x_0^2} \right]$

Replica symmetric formula

$$F_n \xrightarrow{n \to \infty} \sup_{q \geq 0} F(\lambda q) - \frac{\lambda}{4} q^2$$
Recall $Y = \sqrt{\lambda/n}XX^\top + Z$, where $(X_i)_{1 \leq i \leq n} \overset{\text{i.i.d.}}{\sim} P_0$.

- If $P_0 = \mathcal{N}(0, 1)$, PCA is optimal.
Recall $\mathbf{Y} = \sqrt{\lambda/n} \mathbf{X} \mathbf{X}^\top + \mathbf{Z}$, where $(X_i)_{1 \leq i \leq n} \overset{\text{i.i.d.}}{\sim} P_0$.

- If $P_0 = \mathcal{N}(0, 1)$, PCA is optimal.
- Next, we plot the MMSE and $\text{MSE}^{\text{PCA}}$ curves for priors of the form

$$X_i = \begin{cases} \sqrt{\frac{1-p}{p}} & \text{with probability } p \\ -\sqrt{\frac{p}{1-p}} & \text{with probability } 1-p \end{cases}$$

for some $p \in (0, 1)$. 

Plot of MSEs

MMSE, $\text{MSE}^\text{PCA}$ and $\text{MSE}^\text{AMP}$, $p = 0.05$. 
MMSE, MSE_{PCA} and MSE_{AMP}, p = 0.05.
Community detection
From Bernoulli to Gaussian noise

\[ A_{i,j} \sim \text{Ber} \left( \frac{d}{n} + \frac{\sqrt{d}\sqrt{\lambda}}{n} \tilde{X}_i \tilde{X}_j \right) \]  

(1)

where \( \tilde{X}_k = \begin{cases} \sqrt{(1-p)/p} & \text{with probability } p \\ -\sqrt{p/(1-p)} & \text{with probability } 1-p \end{cases} \).

Community detection
From Bernoulli to Gaussian noise

\[ A_{i,j} \sim \text{Ber} \left( \frac{d}{n} + \frac{\sqrt{d} \sqrt{\lambda}}{n} \tilde{X}_i \tilde{X}_j \right) \]  \hspace{1cm} (1)

where \( \tilde{X}_k = \begin{cases} \sqrt{(1 - p)/p} & \text{with probability } p \\ -\sqrt{p/(1 - p)} & \text{with probability } 1 - p \end{cases} \).

The Bernoulli noise model (1) is “equivalent” to the Gaussian noise model (when \( n, d \to \infty \)):

\[ A'_{i,j} = \frac{d}{n} + \frac{\sqrt{d} \sqrt{\lambda}}{n} \tilde{X}_i \tilde{X}_j + \sqrt{\frac{d}{n}} Z_{i,j} \]  \hspace{1cm} (2)

where \( Z_{i,j} \sim \mathcal{N}(0, 1) \),

\[ Yash \text{ Deshpande et al. (2017). “Asymptotic mutual information for the balanced binary stochastic block model”. In: Information and Inference: A Journal of the IMA 6.2, pp. 125–170.} \]
Community detection
From Bernoulli to Gaussian noise

\[ A_{i,j} \sim \text{Ber} \left( \frac{d}{n} + \frac{\sqrt{d\lambda}}{n} \tilde{X}_i \tilde{X}_j \right) \]  

(1)

where

\[ \tilde{X}_k = \begin{cases} \sqrt{(1-p)/p} & \text{with probability } p \\ -\sqrt{p/(1-p)} & \text{with probability } 1-p \end{cases} \]

The Bernoulli noise model (1) is “equivalent” to the Gaussian noise model (when \( n, d \rightarrow \infty \))

\[ A'_{i,j} = \frac{d}{n} + \frac{\sqrt{d\lambda}}{n} \tilde{X}_i \tilde{X}_j + \sqrt{\frac{d}{n}} Z_{i,j} \]  

(2)

where \( Z_{i,j} \overset{i.i.d.}{\sim} \mathcal{N}(0,1) \), and thus to

\[ \sqrt{n} d A'_{i,j} - \sqrt{d} = Y_{i,j} = \sqrt{\frac{\lambda}{n}} \tilde{X}_i \tilde{X}_j + Z_{i,j} \]

Phase diagram for asymmetric community detection

Large degrees asymptotic

\[ p^* = \frac{1}{2} - \frac{1}{2\sqrt{3}} \] as in Guilhem Semerjian’s talk!

Phase diagram from Caltagirone et al., 2017 joint work with Francesco Caltagirone and Léo Miolane.
Community detection (the sparse case)

The Stochastic Block Model (SBM)

$G$ is generated as follows:

- $n$ vertices: $1, \ldots, n$.
- Each vertex $i$ has a label $X_i \in \{1, 2\}$ where $(X_k)_{k \sim i.d.} \sim 1 + \text{Ber}(1 - p)$.
- Two vertices $i, j$ are then connected with probability $M_{X_i, X_j}$.

Important quantity: the signal-to-noise ratio $\frac{d}{1 \neq b^2}$. 

\[ M_{1,2} = M_{2,1} \]

\[ M_{2,2} \]

\[ \approx pn \text{ vertices} \]

\[ \approx (1 - p)n \text{ vertices} \]
Community detection (the sparse case)

The Stochastic Block Model (SBM)

$G$ is generated as follows:

- $n$ vertices: $1, \ldots, n$.
- Each vertex $i$ has a label $X_i \in \{1, 2\}$ where $(X_k)_k \sim \text{i.i.d.} 1 + \text{Ber}(1 - p)$.
- Two vertices $i, j$ are then connected with probability $M_{X_i, X_j}$.

- The connectivity matrix is of the form:

$$M = \frac{d}{n} \begin{pmatrix} a & b \\ b & c \end{pmatrix}$$

$a, c > b$ and $pa + (1 - p)b = pb + (1 - p)c = 1$.

- Important quantity: the signal-to-noise ratio

$$\lambda = d(1 - b)^2$$
The Non-Backtracking Matrix

The problem: if $d \to \infty$, then Wigner’s semi-circle law + BBP phase transition but if $d < \infty$ as $n \to \infty$, then Lifshitz tails.

The solution: the non-backtracking matrix on directed edges of the graph: $B_{u \to v, v \to w} = 1\{\{u, v\} \in E\}1\{\{v, w\} \in E\}1(u \neq w)$ achieves weak reconstruction on the SBM as soon as $\lambda > 1$.

Bordenave et al., 2015 joint work with Charles Bordenave and Laurent Massoulié.

---

$^3$Florent Krzakala et al. (2013). “Spectral redemption in clustering sparse networks”.

Asymmetric communities

The main picture

- if $\lambda > 1$, recovery is possible and easy for all $p < 1/2$
- No proof for the curve $\lambda_c$ for sparse graphs.
The degree-corrected SBM

Impossibility result

The non-backtracking matrix is also working for the symmetric degree-corrected SBM.

\[ P(u \sim v) = \frac{\phi_u \phi_v}{n} \begin{cases} a & \text{if } X_u = X_v \\ b & \text{if } X_u \neq X_v \end{cases} \]

**Theorem**

*Reconstruction is impossible if:*

\[(a - b)^2 \Phi^{(2)} \leq 2(a + b),\]

*with \(\Phi^{(2)}\) the second moment of the weights.*

Gulikers et al., 2015 joint work with Lennart Gulikers and Laurent Massoulié.
Robustness of the non-backtracking matrix

Weak recovery

Let $\rho = \Phi^{(2)} \frac{a+b}{2}$ and $\mu = \Phi^{(2)} \frac{a-b}{2}$.

**Leading eigenvalue** of the non-backtracking matrix $B$ is asymptotic to $\rho$.

**Second eigenvalue** is asymptotic to $\mu$ when $\mu^2 > \rho$, but asymptotically bounded by $\sqrt{\rho}$ when $\mu^2 \leq \rho$.

All the remaining eigenvalues are asymptotically bounded by $\sqrt{\rho}$.

Consequently, a clustering positively-correlated with the true communities can be obtained based on the second eigenvector of $B$ in the regime where $\mu^2 > \rho$ i.e. when $(a - b)^2 \Phi^{(2)} > 2(a + b)$.

Gulikers et al., 2016, joint work with Lennart Gulikers and Laurent Massoulié.
Overview

- **Theoretical approach**
  - Make assumptions, i.e. take a model for the data, and prove theorems.
    - Low-rank matrix estimation
    - Community detection

- **Data driven approach**
  - Take a dataset and experiment!
    - Unsupervised feature learning with deep neural network

- **Common themes: graphs**
Unsupervised learning in practice

- If you have access to an expert.
  a) Ask him to do feature engineering.
  b) Apply your favorite clustering algorithm on the feature vectors.
  c) Ask your expert if your algorithm was correct.
Unsupervised learning in practice

- If you have access to an expert.
  a) Ask him to do feature engineering.
  b) Apply your favorite clustering algorithm on the feature vectors.
  c) Ask your expert if your algorithm was correct.
- In the first part of this talk, we replaced the expert by a model.
Unsupervised learning in practice

- If you have access to an expert.
  a) Ask him to do feature engineering.
  b) Apply your favorite clustering algorithm on the feature vectors.
  c) Ask your expert if your algorithm was correct.

- In the first part of this talk, we replaced the expert by a model.
- Task a) is labor-intensive. Can we leverage the success of deep learning for feature extraction in an unsupervised framework?
Unsupervised learning in practice

- If you have access to an expert.
  - a) Ask him to do feature engineering.
  - b) Apply your favorite clustering algorithm on the feature vectors.
  - c) Ask your expert if your algorithm was correct.

- In the first part of this talk, we replaced the expert by a model.

- Task a) is labor-intensive. Can we leverage the success of deep learning for feature extraction in an unsupervised framework?

- The motivation is not new! Representation learning with Boltzmann Machines, Auto-Encoders, Generative Adversarial Networks...
Unsupervised learning in practice

- If you have access to an expert.
  a) Ask him to do feature engineering.
  b) Apply your favorite clustering algorithm on the feature vectors.
  c) Ask your expert if your algorithm was correct.

- In the first part of this talk, we replaced the expert by a model.

- Task a) is labor-intensive. Can we leverage the success of deep learning for feature extraction in an unsupervised framework?

- The motivation is not new! Representation learning with Boltzmann Machines, Auto-Encoders, Generative Adversarial Networks...

- We propose a new data-driven approach based on the triplet loss on graphs.

ICLR International Conference on Learning Representations
Let’s agree on a dataset!

MNIST

We need to agree on the desired output of the algorithm.
Dimension reduction

**PCA**

Without an expert, it is safe to reduce the dimension before the clustering step, i.e. we replace feature engineering by dimension reduction.

How to improve the feature engineering now?
Random projection

Johnson-Lindenstrauss dimension reduction lemma

The probability that a random projection gives a distance-preserving dimension reduction is high.
Johnson-Lindenstrauss lemma with random LeNet

k-nearest neighbor graph
Side remark

MNIST is too clean!

We are using a noisy version of MNIST.
Learning the neural network

Summary so far:

- We used a neural network with random weights to produce noisy feature vectors $f_i$ from the images.
- We built the k-nearest neighbor graph from the $f_i$.

We need to improve the feature vectors, i.e. learn the weights of the neural network.
Learning the neural network

Summary so far:

- We used a neural network with random weights to produce noisy feature vectors $f_i$ from the images.
- We built the k-nearest neighbor graph from the $f_i$.

We need to improve the feature vectors, i.e. learn the weights of the neural network.

We do not have labels, but there is signal in the graph: for each edge there is $\approx 30\%$ of chances that both end-points belong to the same class, which is much better than the $10\%$ given by a random pairing.
Learning the neural network

Summary so far:
- We used a neural network with random weights to produce noisy feature vectors $f_i$ from the images.
- We built the k-nearest neighbor graph from the $f_i$.

We need to improve the feature vectors, i.e. learn the weights of the neural network.

We do not have labels, but there is signal in the graph: for each edge there is $\approx 30\%$ of chances that both end-points belong to the same class, which is much better than the $10\%$ given by a random pairing.

Idea of the triplet loss: for each edge $i \to j$, pick a random vertex $k$ in the graph and ensure that $\langle f_i, f_j \rangle > \langle f_i, f_k \rangle + \alpha$, by using the hinge loss with margin $\alpha$. 
Back to the Stochastic Block Model

Community detection with the triplet loss

**Online algorithm:** each incoming edge $i \rightarrow j$ induces a loss

$$
\ell_{(i,j)} = (\langle f_i, f_K \rangle + \alpha - \langle f_i, f_j \rangle)^+ \text{ with a random } K;
$$

once a batch arrived, update the embeddings $f_i$ to minimize the loss.
Back to the Stochastic Block Model

Community detection with the triplet loss

**Online algorithm:** each incoming edge $i \rightarrow j$ induces a loss

$$\ell_{(i,j)} = (\langle f_i, f_K \rangle + \alpha - \langle f_i, f_j \rangle)^+$$

with a random $K$;

once a batch arrived, update the embeddings $f_i$ to minimize the loss.

**Comparison of Triplet Loss and Non-Backtracking.**

Our algorithm learns a graph embedding.
Learning the neural network

Empirical results

The accuracy of the clustering increases from \( \approx 25\% \) for the random projection to more than 50\%. 
Learning the neural network

Empirical results

The accuracy of the clustering increases from \(\approx 25\%\) for the random projection to more than 50\%. 
Summary

- **Theoretical approach**
  - Proof of the information theoretic threshold for the low-rank matrix estimation
  - Performance analysis of spectral clustering of the non-backtracking matrix for the community detection problem.

- **Data driven approach**
  - Deep learning algorithm based on the triplet loss and nearest neighbor graph for unsupervised feature extraction.

- **Common themes: graphs**
Thank you for your attention.
References I

References II


References III

References IV

