Mean Field Games: Numerical Methods and Applications in Machine Learning

Part 9: From MFG to ML: Three Examples

Mathieu Laurière

https://mlauriere.github.io/teaching/MFG-PKU-9.pdf

Peking University
Summer School on Applied Mathematics
July 26 – August 6, 2021

RECAP

Outline

1. MF Analysis of SGD for Wide NN

2. MFC Model for Deep Learning

3. MFG Model for Clustering Analysis

In a Nutshell

[Rotskoff, Vanden-Eijnden'18][1]:

"Parameters as interacting particles: long time convergence and asymptotic error scaling of neural networks"

Main points:

- Neural networks with a wide layer
- Mean field of neurons' parameters
- Convex loss function
- SGD: LLN & CLT

Related work:
[Chizat, Bach'18][2], [Mei, Montanari, Nguyen'18][3], [Sirignano, Spiliopoulos'20][4] ...

[1] Rotskoff, G. M., & Vanden-Eijnden, E. (2018). Parameters as interacting particles: long time convergence and asymptotic error scaling of neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (pp. 7146-7155).

[2] Chizat, L., & Bach, F. (2018). On the Global Convergence of Gradient Descent for Over-parameterized Models using Optimal Transport. *Advances in Neural Information Processing Systems*, 31, 3036-3046.

[3] Mei, S., Montanari, A., & Nguyen, P. M. (2018). A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33), E7665-E7671.

[4] Sirignano, J., & Spiliopoulos, K. (2020). Mean field analysis of neural networks: A central limit theorem. *Stochastic Processes and their Applications*, 130(3), 1820-1852.

Wide Neural Network

- Target function $f : \Omega \subset \mathbb{R}^N \to \mathbb{R}$
- **Goal:** minimize mean-squared error over $\tilde{f}$:

$$\ell(f, \tilde{f}) = \frac{1}{2} \int_{\Omega} |f(x) - \tilde{f}(x)|^2 d\mu(x)$$

Wide Neural Network

- Target function $f : \Omega \subset \mathbb{R}^N \to \mathbb{R}$
- **Goal:** minimize mean-squared error over $\tilde{f}$:

$$\ell(f, \tilde{f}) = \frac{1}{2} \int_\Omega |f(x) - \tilde{f}(x)|^2 d\mu(x)$$

- For $\tilde{f}$: take a NN with n neurons:

$$f_n(x) = f_{(c,y)}(x) = \frac{1}{n} \sum_{i=1}^{n} c_i \varphi(x, y_i)$$

where

- $(c, y) = (c_i, y_i)_{i=1}^n \in (\mathbb{R} \times D)^n \subset (\mathbb{R} \times \mathbb{R}^N)^n$ are the parameters
- $\varphi : \Omega \times D \to \mathbb{R}$ is a kernel (activation function, ...)

Wide Neural Network

- Target function $f : \Omega \subset \mathbb{R}^N \to \mathbb{R}$
- **Goal:** minimize mean-squared error over $\tilde{f}$:

$$\ell(f, \tilde{f}) = \frac{1}{2} \int_\Omega |f(x) - \tilde{f}(x)|^2 d\mu(x)$$

- For $\tilde{f}$: take a NN with n neurons:

$$f_n(x) = f_{(c,y)}(x) = \frac{1}{n} \sum_{i=1}^{n} c_i \varphi(x, y_i)$$

  where

  - $(c, y) = (c_i, y_i)_{i=1}^n \in (\mathbb{R} \times D)^n \subset (\mathbb{R} \times \mathbb{R}^N)^n$ are the parameters
  - $\varphi : \Omega \times D \to \mathbb{R}$ is a kernel (activation function, ...)

- NB: $\ell(f, f_n) = \ell(f, f_{(c,y)})$ is not convex w.r.t. (c, y)
- Here shallow NN but enough to have wide final layer

- Rewriting:

$$f_n(x) = \int_D \frac{1}{n} \sum_{i=1}^{n} c_i \varphi(x, y) \delta_{y_i}(y) dy =: \varphi \star G_n(x)$$

- where: Weighted empirical distribution:

$$G_n : D \ni y \mapsto \frac{1}{n} \sum_{i=1}^{n} c_i \delta_{y_i}(y) \in \mathbb{R}$$

Wide Neural Network – Mean Field Limit

- Rewriting:

$$f_n(x) = \int_D \frac{1}{n} \sum_{i=1}^{n} c_i \varphi(x, y) \delta_{y_i}(y) dy =: \varphi \star G_n(x)$$

- where: Weighted empirical distribution:

$$G_n : D \ni y \mapsto \frac{1}{n} \sum_{i=1}^{n} c_i \delta_{y_i}(y) \in \mathbb{R}$$

- Limit $n \to +\infty$:

$$G_n \to G, \qquad \ell(f, f_n) \to \ell(f, \varphi \star G)$$

- Note: $\ell(f, \cdot)$ becomes convex
  $\to$ unique minimal value ℓ^*; possibly multiple minimizers G^*

SGD Convergence

- Minimizing loss $\ell \Leftrightarrow$ Minimizing energy E:

$$E(c_1, y_1, \dots, c_n, y_n) = n(\ell(f, f_n) - C_f) = -\sum_{i=1}^{n} c_i F(y_i) + \frac{1}{2n} \sum_{i,j=1}^{n} c_i c_j K(y_i, y_j)$$

where $F(y) = \int_\Omega f(x)\varphi(x,y)d\mu(x)$, $K(y,z) = \int_\Omega \varphi(x,y)\varphi(x,z)d\mu(x)$

SGD Convergence

- Minimizing loss $\ell \Leftrightarrow$ Minimizing energy E:

$$E(c_1, y_1, \ldots, c_n, y_n) = n(\ell(f, f_n) - C_f) = -\sum_{i=1}^{n} c_i F(y_i) + \frac{1}{2n} \sum_{i,j=1}^{n} c_i c_j K(y_i, y_j)$$

where $F(y) = \int_\Omega f(x)\varphi(x,y)d\mu(x)$, $K(y,z) = \int_\Omega \varphi(x,y)\varphi(x,z)d\mu(x)$

- Gradient Descent dynamics: coupled ODEs for $i = 1, \ldots, n$:

$$\begin{cases} (Y_i(0), C_i(0)) \sim \rho_{in} \text{ i.i.d.} \\ \begin{cases} \dot{Y}_i = C_i \nabla F(Y_i) - \frac{1}{n}\sum_{j=1}^{n} C_i C_j \nabla K(Y_i, Y_j) \\ \dot{C}_i = F(Y_i) - \frac{1}{n}\sum_{j=1}^{n} C_j K(Y_i, Y_j) \end{cases} \end{cases}$$

SGD Convergence – Mean Field Limit

- Particle empirical distribution:

$$\rho_n(t,y,c) = \frac{1}{n}\sum_{i=1}^{n} \delta_{C_i(t)}(c)\delta_{Y_i(t)}(y)$$

- First moment w.r.t. c = $G_n(t,y)$; $f_n(t,x) = (\varphi \star G_n(t))(x)$
- When $n \to \infty$,

$$\rho_n \to \rho$$

- ρ solves the PDE:

$$\begin{cases} \rho_0 = \rho_{in} \\ \partial_t \rho_t = \nabla \cdot (c\,\nabla U([\rho_t],y)\rho_t) + \partial_c(U([\rho_t],y)\rho_t) \end{cases}$$

where

$$U([\rho],y) = -F(y) + \int_{D\times\mathbb{R}} c' K(y,y')\rho(y',c')dy'dc'$$

- Gradient descent in Wasserstein space on convex energy functional

SGD Convergence – Mean Field Limit

- Particle empirical distribution:

$$\rho_n(t,y,c) = \frac{1}{n}\sum_{i=1}^{n} \delta_{C_i(t)}(c)\delta_{Y_i(t)}(y)$$

- First moment w.r.t. c = $G_n(t,y)$; $f_n(t,x) = (\varphi \star G_n(t))(x)$
- When $n \to \infty$,

$$\rho_n \to \rho$$

- ρ solves the PDE:

$$\begin{cases} \rho_0 = \rho_{in} \\ \partial_t \rho_t = \nabla \cdot (c \nabla U([\rho_t], y)\rho_t) + \partial_c (U([\rho_t], y)\rho_t) \end{cases}$$

  where

$$U([\rho], y) = -F(y) + \int_{D\times\mathbb{R}} c' K(y,y')\rho(y',c')dy'dc'$$

- Gradient descent in Wasserstein space on convex energy functional
- Stochastic version (SGD); LLN; CLT

Some Extensions: Neurons Birth and Death

[Rotskoff, Jelassi, Bruna, Vanden-Eijnden'19][5]

"Neuron birth-death dynamics accelerates gradient descent and converges asymptotically"

[5] Rotskoff, G., Jelassi, S., Bruna, J., & Vanden-Eijnden, E. (2019, May). Neuron birth-death dynamics accelerates gradient descent and converges asymptotically. In *International Conference on Machine Learning* (pp. 5508-5517). PMLR.

Some Extensions: Neurons Birth and Death

[Rotskoff, Jelassi, Bruna, Vanden-Eijnden'19][5]

"Neuron birth-death dynamics accelerates gradient descent and converges asymptotically"

- From empirical distribution to mean field distribution:

$$\mu_t^n(d\theta) = \frac{1}{n}\sum_{i=1}^{n} \delta_{\theta_i(t)}(d\theta) \to \mu_t(d\theta)$$

- satisfying PDE, for a potential V:

$$\partial_t \mu_t = \nabla \cdot (\mu_t \nabla V)$$

[5] Rotskoff, G., Jelassi, S., Bruna, J., & Vanden-Eijnden, E. (2019, May). Neuron birth-death dynamics accelerates gradient descent and converges asymptotically. In *International Conference on Machine Learning* (pp. 5508-5517). PMLR.

Some Extensions: Neurons Birth and Death

[Rotskoff, Jelassi, Bruna, Vanden-Eijnden'19][5]

"Neuron birth-death dynamics accelerates gradient descent and converges asymptotically"

- From empirical distribution to mean field distribution:

$$\mu_t^n(d\theta) = \frac{1}{n}\sum_{i=1}^{n} \delta_{\theta_i(t)}(d\theta) \to \mu_t(d\theta)$$

- satisfying PDE, for a potential V:

$$\partial_t \mu_t = \nabla \cdot (\mu_t \nabla V)$$

- Main idea: add **birth/death** (and keep mass constant):

$$\partial_t \mu_t = \nabla \cdot (\mu_t \nabla V) - \alpha V \mu_t + \alpha \bar{V} \mu_t$$

[5] Rotskoff, G., Jelassi, S., Bruna, J., & Vanden-Eijnden, E. (2019, May). Neuron birth-death dynamics accelerates gradient descent and converges asymptotically. In *International Conference on Machine Learning* (pp. 5508-5517). PMLR.

Some Extensions: Neurons Birth and Death

[Rotskoff, Jelassi, Bruna, Vanden-Eijnden'19][5]

"Neuron birth-death dynamics accelerates gradient descent and converges asymptotically"

- From empirical distribution to mean field distribution:

$$\mu_t^n(d\theta) = \frac{1}{n}\sum_{i=1}^{n}\delta_{\theta_i(t)}(d\theta) \to \mu_t(d\theta)$$

- satisfying PDE, for a potential V:

$$\partial_t \mu_t = \nabla \cdot (\mu_t \nabla V)$$

- Main idea: add **birth/death** (and keep mass constant):

$$\partial_t \mu_t = \nabla \cdot (\mu_t \nabla V) - \alpha V \mu_t + \alpha \bar{V} \mu_t$$

- $\rightarrow$ *global convergence to global minimizers* (see paper for assumptions)

[5] Rotskoff, G., Jelassi, S., Bruna, J., & Vanden-Eijnden, E. (2019, May). Neuron birth-death dynamics accelerates gradient descent and converges asymptotically. In *International Conference on Machine Learning* (pp. 5508-5517). PMLR.

Some Extensions: Adversarial Networks (GANs)

[Domingo-Enrich, Jelassi, Mensch, Rotskoff, Bruna'20][6]

"A mean-field analysis of two-player zero-sum games"

[6] Domingo-Enrich, C., Jelassi, S., Mensch, A., Rotskoff, G., & Bruna, J. (2020). A mean-field analysis of two-player zero-sum games. *Advances in neural information processing systems*.

Some Extensions: Adversarial Networks (GANs)

[Domingo-Enrich, Jelassi, Mensch, Rotskoff, Bruna'20][6]

"A mean-field analysis of two-player zero-sum games"

- **Goal:** mixed **Nash equilibrium** for $\ell(x, y)$, i.e., **saddle point** of

$$\mathcal{L}(\mu^x, \mu^y) = \int\int \ell(x, y) d\mu^x(x) d\mu^y(y)$$

- Finite number of parameters $\rightarrow$ Mean field

[6] Domingo-Enrich, C., Jelassi, S., Mensch, A., Rotskoff, G., & Bruna, J. (2020). A mean-field analysis of two-player zero-sum games. *Advances in neural information processing systems*.

Some Extensions: Adversarial Networks (GANs)

[Domingo-Enrich, Jelassi, Mensch, Rotskoff, Bruna'20][6]

"A mean-field analysis of two-player zero-sum games"

- **Goal:** mixed **Nash equilibrium** for $\ell(x, y)$, i.e., **saddle point** of

$$\mathcal{L}(\mu^x, \mu^y) = \int \int \ell(x, y) d\mu^x(x) d\mu^y(y)$$

- Finite number of parameters $\to$ Mean field
- **Gradient descent-ascent** $\Rightarrow$ PDE system:

$$\begin{cases} \partial_t \mu_t^x = \nabla \cdot (\mu_t^x \nabla_x V_x(\mu_t^y, x)), & \mu_0^x = \mu_{x,0} \\ \partial_t \mu_t^y = -\nabla \cdot (\mu_t^y \nabla_y V(\mu_t^x, y)), & \mu_0^y = \mu_{y,0} \end{cases}$$

with

$$\begin{cases} V_x(\mu^y, x) = \frac{\delta \mathcal{L}}{\delta \mu^x}(\mu^x, \mu^y)(x) = \int \ell(x, y) d\mu^y(y) \\ V_y(\mu^x, y) = \frac{\delta \mathcal{L}}{\delta \mu^y}(\mu^x, \mu^y)(y) = \int \ell(x, y) d\mu^x(x) \end{cases}$$

[6] Domingo-Enrich, C., Jelassi, S., Mensch, A., Rotskoff, G., & Bruna, J. (2020). A mean-field analysis of two-player zero-sum games. *Advances in neural information processing systems*.

Link with MKV Control

[Tzen, Raginsky'20][7]

"A mean-field theory of lazy training in two-layer neural nets: entropic regularization and controlled McKean-Vlasov dynamics"

[7] Tzen, B., & Raginsky, M. (2020). A mean-field theory of lazy training in two-layer neural nets: entropic regularization and controlled McKean-Vlasov dynamics. arXiv preprint arXiv:2002.01987.

Link with MKV Control

[Tzen, Raginsky'20][7]

"A mean-field theory of lazy training in two-layer neural nets: entropic regularization and controlled McKean-Vlasov dynamics"

- Adding entropic regularization with Gaussian prior: $\mathrm{KL}(\mu)$
  $\Rightarrow$ Unique minimizer
- **MKV optimal control** (aka MFC) formulation
- Optimality condition: HJB-KFP PDE system

[7] Tzen, B., & Raginsky, M. (2020). A mean-field theory of lazy training in two-layer neural nets: entropic regularization and controlled McKean-Vlasov dynamics. arXiv preprint arXiv:2002.01987.

Outline

1. MF Analysis of SGD for Wide NN

2. MFC Model for Deep Learning

3. MFG Model for Clustering Analysis

In a Nutshell

[E, Han, Li'19][8]:

"A mean-field optimal control formulation of deep learning"

Main points:

- Residual Neural Network as dynamical system
- Continuous time formulation via ODE
- Loss over a mean-field of samples
- MFC viewpoint: Pontryagin Maximum Principle & HJB equation

Related work: [E, Ma, Wu'20][9], [Li'20][10], [Lu, Ma, Lu, Lu, Ying'20][11], ...

[8] E, W., Han, J., & Li, Q. (2019). A mean-field optimal control formulation of deep learning. *Research in the Mathematical Sciences*, 6(1), 1-41.

[9] E, W., Ma, C., & Wu, L. (2020). Machine learning from a continuous viewpoint, I. *Science China Mathematics*, 63(11), 2233-2266.

[10] Li, Q. Dynamical Systems and Machine Learning. (Lecture notes for summer school on Machine Learning and Dynamical Systems at Peking University)

[11] Lu, Y., Ma, C., Lu, Y., Lu, J., & Ying, L. (2020, November). A mean field analysis of deep ResNet and beyond: Towards provably optimization via overparameterization from depth. In *International Conference on Machine Learning* (pp. 6426-6436). PMLR.

Deep Learning with RNN

- **Data set:** $S = \left\{(x_0^i, y_0^i), i = 1, \ldots, N_{samples}\right\}$, (input, output) $\sim \mu_0$
- Goal: Learn how to produce y given x by looking at S

Deep Learning with RNN

- **Data set:** $S = \left\{ (x_0^i, y_0^i), i = 1, \ldots, N_{samples} \right\}$, (input, output) $\sim \mu_0$
- Goal: Learn how to produce y given x by looking at S
- **Residual Neural Network (RNN):** feedforward dynamics $f : \mathbb{R}^d \times \Theta \to \mathbb{R}^d$,

$$\xi_0 = x \text{ (input)}, \qquad \xi_{t+1} = \xi_t + f(\xi_t, \theta_t), \quad t = 0, 1, \ldots, T-1,$$

where T = depth (number of layers)

Deep Learning with RNN

- **Data set:** $S = \left\{(x_0^i, y_0^i), i = 1, \ldots, N_{samples}\right\}$, (input, output) $\sim \mu_0$
- Goal: Learn how to produce y given x by looking at S
- **Residual Neural Network (RNN):** feedforward dynamics $f : \mathbb{R}^d \times \Theta \to \mathbb{R}^d$,

$$\xi_0 = x \text{ (input)}, \qquad \xi_{t+1} = \xi_t + f(\xi_t, \theta_t), \quad t = 0, 1, \ldots, T-1,$$

  where T = depth (number of layers)
- **Goal:** minimize (discrete-time) empirical loss over $\theta : \{0, \ldots, T\} \to \Theta$:

$$J_S(\theta) = \frac{1}{N_{samples}} \sum_{i=1}^{N_{samples}} \left[\Phi(\xi_T^i, y_0^i) + \sum_{t=0}^{T} L(\xi_t^i, \theta_t) \right]$$

  subject to

$$\begin{cases} \xi_0^i = x_0^i, & i = 1, \ldots, N_{samples} \\ \xi_{t+1}^i = \xi_t^i + f(\xi_t^i, \theta_t), & t = 0, \ldots, T-1, \end{cases}$$

  where
  - Φ = loss for not matching the output
  - L = regularizer

Deep Learning with RNN

- **Data set:** $S = \left\{(x_0^i, y_0^i), i = 1, \ldots, N_{samples}\right\}$, (input, output) $\sim \mu_0$
- Goal: Learn how to produce y given x by looking at S
- **Residual Neural Network (RNN):** feedforward dynamics $f : \mathbb{R}^d \times \Theta \to \mathbb{R}^d$,

$$\xi_0 = x \text{ (input)}, \qquad \xi_{t+1} = \xi_t + f(\xi_t, \theta_t), \quad t = 0, 1, \ldots, T-1,$$

  where T = depth (number of layers)
- **Goal:** minimize (discrete-time) empirical loss over $\theta : \{0, \ldots, T\} \to \Theta$:

$$J_S(\theta) = \frac{1}{N_{samples}} \sum_{i=1}^{N_{samples}} \left[\Phi(\xi_T^i, y_0^i) + \sum_{t=0}^{T} L(\xi_t^i, \theta_t) \right]$$

  subject to

$$\begin{cases} \xi_0^i = x_0^i, & i = 1, \ldots, N_{samples} \\ \xi_{t+1}^i = \xi_t^i + f(\xi_t^i, \theta_t), & t = 0, \ldots, T-1, \end{cases}$$

  where
  - Φ = loss for not matching the output
  - L = regularizer
- ⚠ Same θ used for all samples

Dynamical System Viewpoint

- **Deep RNN:** let depth increase but keep T fixed, i.e., let $\Delta t \to 0$
- Continuous time dynamics:

$$\xi_0 = x, \qquad \dot{\xi}_t = f(\xi_t, \theta_t), \quad t \in [0, T]$$

Dynamical System Viewpoint

- **Deep RNN:** let depth increase but keep T fixed, i.e., let $\Delta t \to 0$
- Continuous time dynamics:

$$\xi_0 = x, \qquad \dot{\xi}_t = f(\xi_t, \theta_t), \quad t \in [0, T]$$

- **Goal:** minimize continuous-time empirical loss over $\theta : [0, T] \to \Theta$:

$$J_S(\theta) = \frac{1}{N_{samples}} \sum_{i=1}^{N_{samples}} \left[\Phi(\xi_T^i, y_0^i) + \int_0^T L(\xi_t^i, \theta_t) dt \right]$$

subject to

$$\begin{cases} \xi_0^i = x_0^i, & i = 1, \ldots, N_{samples} \\ \dot{\xi}_t^i = f(\xi_t^i, \theta_t), & t \in [0, T] \end{cases}$$

MFC Formulation

- **Mean field** version when $N_{samples} \to \infty$

- **Mean field** version when $N_{samples} \to \infty$
- **Goal:** minimize continuous-time mean field loss over $\theta : [0, T] \to \Theta$:

$$J(\theta) = \mathbb{E}_{(x_0, y_0) \sim \mu_0} \left[\Phi(\xi_T, y_0) + \int_0^T L(\xi_t, \theta_t) dt \right]$$

subject to:

$$\begin{cases} \xi_0 = x_0 \\ \dot{\xi}_t = f(\xi_t, \theta_t), \quad t \in [0, T] \end{cases}$$

Main Results

Main theoretical results from [E, Han, Li'19]: optimality conditions through:

- HJB equation (on the Wasserstein space):

Main Results

Main theoretical results from [E, Han, Li'19]: optimality conditions through:

- HJB equation (on the Wasserstein space):

$$J(t,\mu,\theta) = \mathbb{E}_{(x_t,y_0)\sim\mu}\left[\Phi(\xi_T,y_0) + \int_t^T L(\xi_s,\theta_s)dt\right]$$
$$= \langle\tilde{\Phi}, \mu_T^{t,\mu,\theta}\rangle + \int_t^T \langle\tilde{L}(\cdot,\theta_s), \mu_s^{t,\mu,\theta}\rangle ds$$

Main Results

Main theoretical results from [E, Han, Li'19]: optimality conditions through:

- HJB equation (on the Wasserstein space):

$$
\begin{aligned}
J(t,\mu,\theta) &= \mathbb{E}_{(x_t,y_0)\sim\mu}\left[\Phi(\xi_T,y_0)+\int_t^T L(\xi_s,\theta_s)dt\right] \\
&= \langle\tilde{\Phi},\mu_T^{t,\mu,\theta}\rangle + \int_t^T \langle \tilde{L}(\cdot,\theta_s),\mu_s^{t,\mu,\theta}\rangle ds
\end{aligned}
$$

- Existence and uniqueness of viscosity solutions

Main theoretical results from [E, Han, Li'19]: optimality conditions through:

- HJB equation (on the Wasserstein space):

$$J(t,\mu,\theta) = \mathbb{E}_{(x_t,y_0)\sim\mu}\left[\Phi(\xi_T, y_0) + \int_t^T L(\xi_s, \theta_s)dt\right]$$
$$= \langle\tilde{\Phi}, \mu_T^{t,\mu,\theta}\rangle + \int_t^T \langle\tilde{L}(\cdot,\theta_s), \mu_s^{t,\mu,\theta}\rangle ds$$

- Existence and uniqueness of viscosity solutions
- Pontryagin Maximum Principle:

$$\begin{cases} \dot{\xi}_t^* = f(\xi_t^*, \theta_t^*), & x_t^* = x_0 \\ \dot{p}_t^* = -\nabla H_x(x_t^*, p_t^*, \theta_t^*), & p_T^* = -\nabla_x \Phi(x_T^*, y_0) \\ \mathbb{E}_{\mu_0}[H(x_t^*, p_t^*, \theta_t^*)] \geq \mathbb{E}_{\mu_0}[H(x_t^*, p_t^*, \theta_t)], & \forall \theta \in \Theta, \text{ a.e. } t \in [0,T] \end{cases}$$

Outline

In a Nutshell

[Aquilanti, Cacace, Camilli, De Maio'20a][12]:

"A mean field games approach to cluster analysis"

Main points:

- Data points: $\mathcal{X} = \{x_1, \ldots, x_I\}$, $x_i \in \mathbb{R}^d$
- Number of clusters: K
- Goal: Find a partition of $\mathcal{X}$ into K clusters $S_1, \ldots, S_K$
- Two algorithms: K-means & Expectation-Maximization
- Interpretation of optimality conditions as MFG

Related work: [Pequito *et al.*'11][13], [Coron'18][14], [Aquilanti *et al.*'20b][15]

[12] Aquilanti, L., Cacace, S., Camilli, F., & De Maio, R. (2020). A mean field games approach to cluster analysis. *Applied Mathematics & Optimization*, 1-25.

[13] Pequito, S., Aguiar, A.P., Sinopoli, B. & Gomes, D., Unsupervised learning of finite mixture models using Mean Field Games, in *Annual Allerton Conference on Communication, Control and Computing*, 2011, 321-328.

[14] Coron, J.L., Quelques exemples de jeux à champ moyen, Ph.D. thesis, Université Paris-Dauphine, 2018

[15] Aquilanti, L., Cacace, S., Camilli, F., & De Maio, R. (2020). A Mean Field Games model for finite mixtures of Bernoulli and Categorical distributions. arXiv preprint arXiv:2004.08119.

Cluster Analysis - K-Means

- K **clusters:** $(S_1, \ldots, S_K)$
- Barycentres: $\mu = (\mu_1, \ldots \mu_K) \in (\mathbb{R}^d)^k$
- Cluster assignment: $c = (c_1, \ldots, c_K), c_i \in \{1, \ldots, K\}$:

$$c_i = k \Leftrightarrow x_i \in S_k$$

- **Goal:** minimize over (μ, c)

$$J(\mu, c) = \sum_{i=1}^{I} \sum_{k=1}^{K} \mathbf{1}_{\{c_i = k\}} |x_i - \mu_k|^2$$

Cluster Analysis - K-Means

- K **clusters:** $(S_1, \ldots, S_K)$
- Barycentres: $\mu = (\mu_1, \ldots \mu_K) \in (\mathbb{R}^d)^k$
- Cluster assignment: $c = (c_1, \ldots, c_K), c_i \in \{1, \ldots, K\}$:

$$c_i = k \Leftrightarrow x_i \in S_k$$

- **Goal:** minimize over (μ, c)

$$J(\mu, c) = \sum_{i=1}^{I} \sum_{k=1}^{K} \mathbf{1}_{\{c_i = k\}} |x_i - \mu_k|^2$$

- **K-means algorithm:**

(i) **Cluster assignment:**

$$\begin{cases} c_i^{(\mathtt{n}+1)} & = \operatorname{argmin}_{c_i} J(\mu^{(\mathtt{n})}, c_i, c_{-i}^{(\mathtt{n})}) = \operatorname{argmin}_j |x_i - \mu_j^{(\mathtt{n})}|^2, \qquad i = 1, \ldots, I \\ S_k^{(\mathtt{n}+1)} & = \{x_i \in \mathcal{X} : c_i^{(\mathtt{n}+1)} = k\}, \qquad k = 1, \ldots, K \end{cases}$$

Cluster Analysis - K-Means

- K **clusters:** $(S_1, \ldots, S_K)$
- Barycentres: $\mu = (\mu_1, \ldots \mu_K) \in (\mathbb{R}^d)^k$
- Cluster assignment: $c = (c_1, \ldots, c_K), c_i \in \{1, \ldots, K\}$:

$$c_i = k \Leftrightarrow x_i \in S_k$$

- **Goal:** minimize over (μ, c)

$$J(\mu, c) = \sum_{i=1}^{I} \sum_{k=1}^{K} \mathbf{1}_{\{c_i = k\}} |x_i - \mu_k|^2$$

- **K-means algorithm:**
  (i) **Cluster assignment:**

$$\begin{cases} c_i^{(\mathtt{n}+1)} & = \operatorname{argmin}_{c_i} J(\mu^{(\mathtt{n})}, c_i, c_{-i}^{(\mathtt{n})}) = \operatorname{argmin}_j |x_i - \mu_j^{(\mathtt{n})}|^2, \qquad i = 1, \ldots, I \\ S_k^{(\mathtt{n}+1)} & = \{x_i \in \mathcal{X} : c_i^{(\mathtt{n}+1)} = k\}, \qquad k = 1, \ldots, K \end{cases}$$

  (ii) **Barycentre update:**

$$\mu_k^{(\mathtt{n}+1)} = \frac{1}{|S_k^{(\mathtt{n}+1)}|} \sum_{x_i \in S_k^{(\mathtt{n}+1)}}^{I} x_i, \qquad k = 1, \ldots, K$$

MFG Model for K-Means

Following [Coron'18]:

- Continuum of data points: $x \sim f$ for some PDF f
- Each point belongs to the cluster with the closest barycentre
  $\rightarrow$ minimization problem
- Barycentres positions depend on choices of other points
  $\rightarrow$ mean field coupling

MFG Model for K-Means

Following [Coron'18]:

- Continuum of data points: $x \sim f$ for some PDF f
- Each point belongs to the cluster with the closest barycentre
  $\rightarrow$ minimization problem
- Barycentres positions depend on choices of other points
  $\rightarrow$ mean field coupling
- **K-population MFG:**
  - $(m_1, \ldots, m_k)$: populations densities, corresponding to dynamics:

$$dX_k(t) = a_k(t)dt + \sqrt{2\epsilon}dW_k(t), t > 0, \qquad X_0 = x$$

  - $(u_1, \ldots, u_k)$: players' value functions: letting
    $\texttt{Bar}(m_k) = \frac{1}{\int_{\mathbb{R}^d} m_k(x)dx} \int_{\mathbb{R}^d} x m_k(x)dx,$

$$u_k(x) = \inf_{a_k} \mathbb{E}_x \left[\int_0^\infty e^{-\rho s} \left(\frac{1}{2}|a_k(s)|^2 + \underbrace{\kappa |X_k(s) - \texttt{Bar}(m_k(s))|^2}_{F(X_k(s), m_k(s))} \right) ds \right]$$

  - Clusters: $S_k = \{x \in \mathbb{R}^d : u_k(x) = \min_{j=1,\ldots,K} u_j(x)\}$

MFG Model for K-Means

Following [Coron'18]:

- Continuum of data points: $x \sim f$ for some PDF f
- Each point belongs to the cluster with the closest barycentre
  $\rightarrow$ minimization problem
- Barycentres positions depend on choices of other points
  $\rightarrow$ mean field coupling
- **K-population MFG:**
  - $(m_1, \dots, m_k)$: populations densities, corresponding to dynamics:
  $$dX_k(t) = a_k(t)dt + \sqrt{2\epsilon}dW_k(t), t > 0, \qquad X_0 = x$$
  - $(u_1, \dots, u_k)$: players' value functions: letting
  $\mathtt{Bar}(m_k) = \frac{1}{\int_{\mathbb{R}^d} m_k(x)dx} \int_{\mathbb{R}^d} x m_k(x)dx,$
  $$u_k(x) = \inf_{a_k} \mathbb{E}_x \left[\int_0^\infty e^{-\rho s} \left(\frac{1}{2}|a_k(s)|^2 + \underbrace{\kappa |X_k(s) - \mathtt{Bar}(m_k(s))|^2}_{F(X_k(s), m_k(s))} \right) ds \right]$$
  - Clusters: $S_k = \{x \in \mathbb{R}^d : u_k(x) = \min_{j=1,\dots,K} u_j(x)\}$
- Consistency rule:
$$\mathtt{Bar}(m_k) = \mathtt{Bar}(\mathbf{1}_{S_k} f)$$

MFG Model for K-Means

Following [Coron'18]:

- Continuum of data points: $x \sim f$ for some PDF f
- Each point belongs to the cluster with the closest barycentre
  $\rightarrow$ minimization problem
- Barycentres positions depend on choices of other points
  $\rightarrow$ mean field coupling
- **K-population MFG:**
  - $(m_1, \dots, m_k)$: populations densities, corresponding to dynamics:
  $$dX_k(t) = a_k(t)dt + \sqrt{2\epsilon}dW_k(t), t > 0, \qquad X_0 = x$$
  - $(u_1, \dots, u_k)$: players' value functions: letting
  $\texttt{Bar}(m_k) = \frac{1}{\int_{\mathbb{R}^d} m_k(x)dx} \int_{\mathbb{R}^d} x m_k(x)dx,$
  $$u_k(x) = \inf_{a_k} \mathbb{E}_x \left[\int_0^\infty e^{-\rho s} \left(\frac{1}{2}|a_k(s)|^2 + \underbrace{\kappa |X_k(s) - \texttt{Bar}(m_k(s))|^2}_{F(X_k(s), m_k(s))} \right) ds \right]$$
  - Clusters: $S_k = \{x \in \mathbb{R}^d : u_k(x) = \min_{j=1,\dots,K} u_j(x)\}$
- Consistency rule:
  $$\texttt{Bar}(m_k) = \texttt{Bar}(\mathbf{1}_{S_k} f)$$
- [Coron'18] proved "K-MFG PDE system $\leftrightarrow$ consistency rule"

MFG Model for K-Means – PDE System

Recall:

$$u_k(x) = \inf_{a_k} \mathbb{E}_x \left[\int_0^\infty e^{-\rho s} \left(\frac{1}{2} |a_k(s)|^2 + F(X_k(s), m_k(s)) \right) ds \right]$$

subj. to:

$$dX_k(t) = a_k(t) dt + \sqrt{2\epsilon} dW_k(t), t > 0, \qquad X_0 = x$$

and with $\texttt{Bar}(m_k) = \frac{1}{\int_{\mathbb{R}^d} m_k(x) dx} \int_{\mathbb{R}^d} x m_k(x) dx$

MFG Model for K-Means – PDE System

Recall:

$$u_k(x) = \inf_{a_k} \mathbb{E}_x \left[\int_0^\infty e^{-\rho s} \left(\frac{1}{2} |a_k(s)|^2 + F(X_k(s), m_k(s)) \right) ds \right]$$

subj. to:

$$dX_k(t) = a_k(t) dt + \sqrt{2\epsilon} dW_k(t), t > 0, \qquad X_0 = x$$

and with $\texttt{Bar}(m_k) = \frac{1}{\int_{\mathbb{R}^d} m_k(x) dx} \int_{\mathbb{R}^d} x m_k(x) dx$

K**-Population MFG PDE system:** for $k = 1, \ldots, K$,

$$\begin{cases} \rho u_k - \epsilon \Delta u_k(x) + \frac{1}{2} |Du_k(x)|^2 = F(x, m_k), & x \in \mathbb{R}^d, \\ \rho m_k(x) - \epsilon \Delta m_k(x) - \operatorname{div}(Du_k(x) m_k(x)) = \rho \tilde{f}_k & x \in \mathbb{R}^d, \end{cases}$$

MFG Model for K-Means – PDE System

Recall:

$$u_k(x) = \inf_{a_k} \mathbb{E}_x \left[\int_0^\infty e^{-\rho s} \left(\frac{1}{2}|a_k(s)|^2 + F(X_k(s), m_k(s)) \right) ds \right]$$

subj. to:

$$dX_k(t) = a_k(t)dt + \sqrt{2\epsilon}dW_k(t), t > 0, \qquad X_0 = x$$

and with $\mathtt{Bar}(m_k) = \frac{1}{\int_{\mathbb{R}^d} m_k(x)dx} \int_{\mathbb{R}^d} x m_k(x) dx$

K**-Population MFG PDE system:** for $k = 1, \dots, K$,

$$\begin{cases} \rho u_k - \epsilon \Delta u_k(x) + \frac{1}{2}|Du_k(x)|^2 = F(x, m_k), & x \in \mathbb{R}^d, \\ \rho m_k(x) - \epsilon \Delta m_k(x) - \operatorname{div}(Du_k(x) m_k(x)) = \rho \tilde{f}_k & x \in \mathbb{R}^d, \end{cases}$$

where $\kappa = \frac{1+\rho}{2}$, and $\tilde{f}_k$ is a Gaussian distribution with mean

$$\tilde{y}_k = \frac{\int_{S_k} x f(x) dx}{\int_{S_k} f(x) dx}$$

and variance ϵ, and the **cluster** $S_k = S_k(u)$ related to $\tilde{y}_k$ is defined by

$$S_k = \{x \in \mathbb{R}^d : u_k(x) = \min_{j=1,\dots,K} u_j(x)\}.$$

Cluster Analysis - Expectation Maximization

- Distribution $P(x) = \sum_{k=1}^{K} \alpha_k p_k(x|\theta_k)$, params. = $(\alpha_k, \theta_k)_k$, densities $(p_k)_k$
- **Goal:** Maximize log-likelihood:

$$\ln P(\mathcal{X}|\alpha, \theta) = \sum_{i=1}^{I} \ln \left(\sum_{k=1}^{K} \alpha_k p_k(x_i|\theta_k) \right)$$

Cluster Analysis - Expectation Maximization

- Distribution $P(x) = \sum_{k=1}^{K} \alpha_k p_k(x|\theta_k)$, params. = $(\alpha_k, \theta_k)_k$, densities $(p_k)_k$
- **Goal:** Maximize log-likelihood:

$$\ln P(\mathcal{X}|\alpha, \theta) = \sum_{i=1}^{I} \ln \left(\sum_{k=1}^{K} \alpha_k p_k(x_i|\theta_k) \right)$$

- Data completion: random $\mathcal{Y} = \{y_i\}_{i=1}^{I}$, $y_i = k \Leftrightarrow x_i$ generated by p_k
- Responsibility of x_i w.r.t. k-th cluster: $\gamma_k(x_i) = p_k(y_i = k|x_i, \theta_k)$

Cluster Analysis - Expectation Maximization

- Distribution $P(x) = \sum_{k=1}^{K} \alpha_k p_k(x|\theta_k)$, params. = $(\alpha_k, \theta_k)_k$, densities $(p_k)_k$
- **Goal:** Maximize log-likelihood:

$$\ln P(\mathcal{X}|\alpha, \theta) = \sum_{i=1}^{I} \ln \left(\sum_{k=1}^{K} \alpha_k p_k(x_i|\theta_k) \right)$$

- Data completion: random $\mathcal{Y} = \{y_i\}_{i=1}^{I}$, $y_i = k \Leftrightarrow x_i$ generated by p_k
- Responsibility of x_i w.r.t. k-th cluster: $\gamma_k(x_i) = p_k(y_i = k|x_i, \theta_k)$
- **Goal:** Maximize expected log-likelihood of complete data:

$$\mathbb{E}_{\mathcal{Y}}[\ln p(\mathcal{X}, \mathcal{Y}|\alpha, \theta)] = \sum_{i=1}^{I} \sum_{k=1}^{K} \gamma_k(x_i) \ln(\alpha_k p_k(x_i|\theta_k))$$

$\rightarrow$ optimality conditions for γ_k or α_k, θ_k

Cluster Analysis - Expectation Maximization

- Distribution $P(x) = \sum_{k=1}^{K} \alpha_k p_k(x|\theta_k)$, params. = $(\alpha_k, \theta_k)_k$, densities $(p_k)_k$
- **Goal:** Maximize log-likelihood:

$$\ln P(\mathcal{X}|\alpha, \theta) = \sum_{i=1}^{I} \ln \left(\sum_{k=1}^{K} \alpha_k p_k(x_i|\theta_k) \right)$$

- Data completion: random $\mathcal{Y} = \{y_i\}_{i=1}^{I}$, $y_i = k \Leftrightarrow x_i$ generated by p_k
- Responsibility of x_i w.r.t. k-th cluster: $\gamma_k(x_i) = p_k(y_i = k|x_i, \theta_k)$
- **Goal:** Maximize expected log-likelihood of complete data:

$$\mathbb{E}_{\mathcal{Y}}[\ln p(\mathcal{X}, \mathcal{Y}|\alpha, \theta)] = \sum_{i=1}^{I} \sum_{k=1}^{K} \gamma_k(x_i) \ln(\alpha_k p_k(x_i|\theta_k))$$

  $\rightarrow$ optimality conditions for γ_k or α_k, θ_k
- Special case: $p_k(\cdot|\theta_k) = \mathcal{N}(\cdot|\mu_k, \Sigma_k)$

Cluster Analysis - Expectation Maximization

- Distribution $P(x) = \sum_{k=1}^{K} \alpha_k p_k(x|\theta_k)$, params. = $(\alpha_k, \theta_k)_k$, densities $(p_k)_k$
- **Goal:** Maximize log-likelihood:

$$\ln P(\mathcal{X}|\alpha, \theta) = \sum_{i=1}^{I} \ln \left(\sum_{k=1}^{K} \alpha_k p_k(x_i|\theta_k) \right)$$

- Data completion: random $\mathcal{Y} = \{y_i\}_{i=1}^{I}$, $y_i = k \Leftrightarrow x_i$ generated by p_k
- Responsibility of x_i w.r.t. k-th cluster: $\gamma_k(x_i) = p_k(y_i = k|x_i, \theta_k)$
- **Goal:** Maximize expected log-likelihood of complete data:

$$\mathbb{E}_{\mathcal{Y}}[\ln p(\mathcal{X}, \mathcal{Y}|\alpha, \theta)] = \sum_{i=1}^{I} \sum_{k=1}^{K} \gamma_k(x_i) \ln(\alpha_k p_k(x_i|\theta_k))$$

$\rightarrow$ optimality conditions for γ_k or α_k, θ_k

- Special case: $p_k(\cdot|\theta_k) = \mathcal{N}(\cdot|\mu_k, \Sigma_k)$
- **EM algorithm:**

(i) **E-step:** posterior: $\gamma_k^{(\mathtt{n}+1)}(x_i) = P(y_i = k|x_i, \mu_k^{(\mathtt{n})}, \Sigma_k^{(\mathtt{n})}) = \ldots$

Cluster Analysis - Expectation Maximization

- Distribution $P(x) = \sum_{k=1}^{K} \alpha_k p_k(x|\theta_k)$, params. = $(\alpha_k, \theta_k)_k$, densities $(p_k)_k$
- **Goal:** Maximize log-likelihood:

$$\ln P(\mathcal{X}|\alpha, \theta) = \sum_{i=1}^{I} \ln \left(\sum_{k=1}^{K} \alpha_k p_k(x_i|\theta_k) \right)$$

- Data completion: random $\mathcal{Y} = \{y_i\}_{i=1}^{I}$, $y_i = k \Leftrightarrow x_i$ generated by p_k
- Responsibility of x_i w.r.t. k-th cluster: $\gamma_k(x_i) = p_k(y_i = k|x_i, \theta_k)$
- **Goal:** Maximize expected log-likelihood of complete data:

$$\mathbb{E}_{\mathcal{Y}}[\ln p(\mathcal{X}, \mathcal{Y}|\alpha, \theta)] = \sum_{i=1}^{I} \sum_{k=1}^{K} \gamma_k(x_i) \ln(\alpha_k p_k(x_i|\theta_k))$$

  $\rightarrow$ optimality conditions for γ_k or α_k, θ_k
- Special case: $p_k(\cdot|\theta_k) = \mathcal{N}(\cdot|\mu_k, \Sigma_k)$
- **EM algorithm:**
  (i) **E-step:** posterior: $\gamma_k^{(\mathtt{n}+1)}(x_i) = P(y_i = k|x_i, \mu_k^{(\mathtt{n})}, \Sigma_k^{(\mathtt{n})}) = \dots$
  (ii) **M-step:** params.: $\alpha_k^{(\mathtt{n}+1)} = \frac{\sum_i \gamma_k^{(\mathtt{n}+1)}(x_i)}{I}$, $\mu_k^{(\mathtt{n}+1)} = \frac{\sum_i x_i \gamma_k^{(\mathtt{n}+1)}(x_i)}{\sum_i \gamma_k^{(\mathtt{n}+1)}(x_i)}$, $\Sigma_k^{(\mathtt{n}+1)} = \dots$

MFG Model for EM

- Continuum of data points: $x \sim f$ for some PDF f
- Mixture: $m(x) = \sum_k \alpha_k m_k(x)$
- Responsibilities: $\gamma_k(x) = \frac{\alpha_k m_k(x)}{m(x)}$
- Mean and covariance: $\mu_k = \frac{\int_{\mathbb{R}^d} x \gamma_k(x) f(x) dx}{\int_{\mathbb{R}^d} \gamma_k(x) f(x) dx}, \qquad \Sigma_k = \dots$
- Cost:

$$J_k(x, a_k) = \lim_{T \to +\infty} \frac{1}{T} \mathbb{E}_x \left\{ \int_0^T \left[\frac{1}{2} |a_k(s)|^2 + F(X_k(s), m_k(X_k(s)), m(X_k(s))) \right] ds \right\},$$

subj. to:

$$dX_k(t) = a_k(t) dt + \sqrt{2\epsilon} dW_k(t), t > 0, \qquad X_0 = x$$

where ($m \rightsquigarrow \gamma_k \rightsquigarrow \mu_k$): $F(x, m_k, m) = \frac{1}{2}(x - \mu_k)^t (\Sigma_k^{-1})^t \Sigma_k^{-1} (x - \mu_k)$

MFG Model for EM

- Continuum of data points: $x \sim f$ for some PDF f
- Mixture: $m(x) = \sum_k \alpha_k m_k(x)$
- Responsibilities: $\gamma_k(x) = \frac{\alpha_k m_k(x)}{m(x)}$
- Mean and covariance: $\mu_k = \frac{\int_{\mathbb{R}^d} x\gamma_k(x) f(x) dx}{\int_{\mathbb{R}^d} \gamma_k(x) f(x) dx}, \qquad \Sigma_k = \dots$
- Cost:

$$J_k(x, a_k) = \lim_{T \to +\infty} \frac{1}{T} \mathbb{E}_x \left\{ \int_0^T \left[\frac{1}{2} |a_k(s)|^2 + F(X_k(s), m_k(X_k(s)), m(X_k(s))) \right] ds \right\},$$

subj. to:

$$dX_k(t) = a_k(t) dt + \sqrt{2\epsilon} dW_k(t), t > 0, \qquad X_0 = x$$

where ($m \rightsquigarrow \gamma_k \rightsquigarrow \mu_k$): $F(x, m_k, m) = \frac{1}{2}(x - \mu_k)^t (\Sigma_k^{-1})^t \Sigma_k^{-1} (x - \mu_k)$

- $m(x) = \sum_k \beta_k \mathcal{N}(x|\nu_k, T_k)$ is **consistent** with the data set f if:

$$\nu_k = \frac{\int x\gamma_k(x) f(x) dx}{\int \gamma_k(x) f(x) dx}, T_k = \epsilon \frac{\int (x - \nu_k)(x - \nu_k)^t \gamma_k(x) f(x) dx}{\int \gamma_k(x) f(x) dx}, \beta_k = \int \gamma_k(x) f(x) dx$$

- [Aquilanti *et al.*'20a]: "K-MFG PDE system $\leftrightarrow$ consistent density family"

MFG Model for EM – PDE System

Recall:

$$J_k(x, a_k) = \lim_{T \to +\infty} \frac{1}{T} \mathbb{E}_x \left\{ \int_0^T \left[\frac{1}{2} |a_k(s)|^2 + F_k(X_k(s), m_k(X_k(s)), m(X_k(s))) \right] ds \right\},$$

subj. to:

$$dX_k(t) = a_k(t)dt + \sqrt{2\epsilon} dW_k(t), t > 0, \qquad X_0 = x$$

MFG Model for EM – PDE System

Recall:

$$J_k(x, a_k) = \lim_{T \to +\infty} \frac{1}{T} \mathbb{E}_x \left\{ \int_0^T \left[\frac{1}{2} |a_k(s)|^2 + F_k(X_k(s), m_k(X_k(s)), m(X_k(s))) \right] ds \right\},$$

subj. to:

$$dX_k(t) = a_k(t)dt + \sqrt{2\epsilon}dW_k(t), t > 0, \qquad X_0 = x$$

K**-Population MFG PDE system:** for $k = 1, \ldots, K$,

$$\begin{cases} -\epsilon \Delta u_k(x) + \frac{1}{2}|Du_k(x)|^2 + \lambda_k = \frac{1}{2}(x - \mu_k)^t (\Sigma_k^{-1})^t \Sigma_k^{-1}(x - \mu_k), & x \in \mathbb{R}^d, \\ \epsilon \Delta m_k(x) + \operatorname{div}(m_k(x) Du_k(x)) = 0, & x \in \mathbb{R}^d, \\ \alpha_k = \int_{\mathbb{R}^d} \gamma_k(x) f(x) dx, & \\ m_k \geq 0, \ \int_{\mathbb{R}^d} m_k(x)dx = 1, u_k(\mu_k) = 0, & \end{cases}$$

where γ_k, μ_k, Σ_k are defined as previously

$$\gamma_k(x) = \frac{\alpha_k m_k(x)}{m(x)}, \mu_k = \frac{\int_{\mathbb{R}^d} x \gamma_k(x) f(x) dx}{\int_{\mathbb{R}^d} \gamma_k(x) f(x) dx}, \Sigma_k = \frac{\int_{\mathbb{R}^d} (x - \mu_k)(x - \mu_k)^t \gamma_k(x) f(x) dx}{\int_{\mathbb{R}^d} \gamma_k(x) f(x) dx}$$

Further results

Also in [Aquilanti *et al.*'20a]:

- EM algorithm & MFG in more general case than GMM
- Numerical results

In [Aquilanti *et al.*'20b]:

- Finite state space multi-pop. MFG PDE system
  $\leftrightarrow$ critical points of log-likelihood functional for a Bernoulli mixture
- MFG to compute the parameters of the mixture model

Summary

(Some) References

- Mean field approach to infinitely wide NN:
  [F. Bach, L. Chizat, *et al.*], [G. Rotskoff, R. Vanden-Eijnden, J. Bruna, *et al.*], [S. Mei, A. Montanari, P.M. Nguyen, *et al.*], [J. Sirignano, K. Spiliopoulos, *et al.*], . . .
- Link with MKV Control:
  [B. Tzen, M. Raginsky], . . .
- Mean field control & deep RNN:
  [W. E, J. Han, Q. Li], [W. E, C. Ma, L. Wu], [Y. Lu, C. Ma, Y. Lu, J. Lu, L. Ying], . . .
- Mean field Langevin dynamics & Optimal control:
  [K. Hu, A. Kazeykina, Z. Ren, *et al.*], . . .
- Mean field approach & relaxed control:
  [J.F. Jabir, D. Siska, L.Szpruch, *et al.*], [L. Bo, A. Capponi, H. Liao, *et al.*], . . .
- Multi-Population MFGs & cluster analysis:
  [J.L. Coron], [L. Aquilanti, S. Cacace, F. Camilli, R. De Maio], [S. Pequito, A.P. Aguiar, B. Sinopoli, D. Gomes], . . .
- Random Batch Method (RBM) & its mean field limit:
  [S. Jin, L. Li, J.G. Liu, *et al.*], . . .