Mean Field Approximation for Interaction Models

Sebastian Allmeier

December 2, 2021

Overview

Table of Contents

General Setting

- population n **interacting objects** which have (countably many) **states** $\mathcal{S}$
- object **interactions are stochastic**
- naturally described by $\mathbf{S}^{(n)}(t) = (S_1(t), \ldots, S_n(t)) \in \mathcal{S}^n$ (continuous time process)
- $S_k(t) \sim$ state of object k at time t

Many modeling possibilities

load balancing, epidemic modeling, caching, communication protocols, SSD garbage collection, malware propagation, ...

Example: Power-of-2-choice model (JSQ(2))

How to allocate jobs in large server farms?

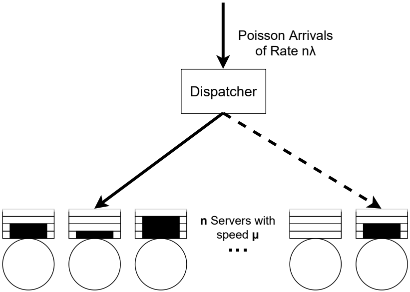
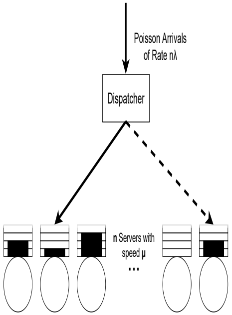


Figure: Power-of-2-choice - introduced by Mitzenmacher in 2001

a simple, effective distribution scheme
can be studied by using the mean field approximation

Example: Power-of-2-choice model (JSQ(2))



$S_i(t) \sim$ number of jobs in queue

- arrival rate $n\lambda$ (λ per server, < 1)
- dispatcher univ. and independently sam
- job added to shorter queue (equality broken at random)
- server service rate $\mu = 1$

Motivation

Mean Field Approximation

Goal: want to analyze transient and steady state behavior of the system

Problems:

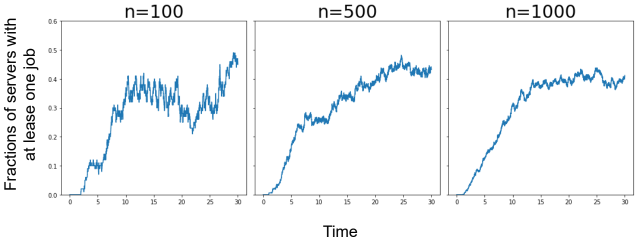
- exponentially growing state space $\mathcal{S}^n$
- predictions by simulation can be inefficient / inaccurate

Mean Field Approximation can help.

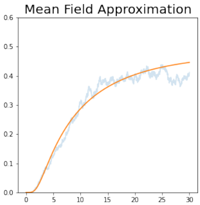
Examples

load balancing, epidemic modeling, caching, communication protocols, SSD garbage collection, malware propagation, ...

Mean Field Idea



$$\downarrow \quad n \to \infty$$



How to obtain the mean field limit?

Alternative system representations

$$S^{(n)}(t) \text{ with } S_k^{(n)}(t) \in \mathcal{S}, \;\; k = \{1...n\}$$

$$X^{(n)}(t) \text{ with } X_{k,s}^{(n)}(t) = \mathbf{1}_{\{S_k^{(n)}(t)=s\}}, \;\; k = \{1...n\}, i \in \mathbb{N}$$

$$Z^{(n)}(t) \text{ with } Z_s^{(n)}(t) = \frac{1}{n}\sum_{k=1}^{n} X_{k,s}^{(n)}(t) = \frac{1}{n}\sum_{k=1}^{n} \mathbf{1}_{\{S_k^{(n)}(t)=s\}}, \;\; i \in \mathbb{N}$$

all are continuous time Markov chains

for Z we use homogeneity (similar statistical behavior of objects) to simplify

How to obtain the mean field limit?

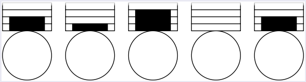
Alternative system representations

$$S^{(n)}(t) \text{ with } S_k^{(n)}(t) \in \mathcal{S},\ \ k = \{1...n\}$$

$$X^{(n)}(t) \text{ with } X_{k,s}^{(n)}(t) = \mathbf{1}_{\{S_k^{(n)}(t)=s\}},\ \ k = \{1...n\}, i \in \mathbb{N}$$

$$Z^{(n)}(t) \text{ with } Z_s^{(n)}(t) = \frac{1}{n}\sum_{k=1}^{n} X_{k,s}^{(n)}(t) = \frac{1}{n}\sum_{k=1}^{n} \mathbf{1}_{\{S_k^{(n)}(t)=s\}},\ \ i \in \mathbb{N}$$

all are continuous time Markov chains
for Z we use homogeneity (similar statistical behavior of objects) to simplify



$$\rightarrow Z(t) = (0.2,\ 0.4,\ 0.2,\ 0.2,\ 0,\ ...)$$

- arrival rate $n\lambda$ (λ per server, < 1)
- dispatcher univ. and independently samples two servers
- job added to shorter queue
  (equality broken at random)
- server service rate $\mu = 1$

Transitions

Arrivals: $Z \to Z + \frac{1}{n} e_i - \frac{1}{n} e_{i-1}$ at rate $n\lambda((\sum_{j \geq i-1} Z_j)^2 - (\sum_{j \geq i} Z_i)^2)$

Removals: $Z \to Z + \frac{1}{n} e_{i-1} - \frac{1}{n} e_i$ at rate $n\mu Z_i$

How to obtain the mean field limit?

Drift

For a given **state** $\boldsymbol{x}$ the drift is (informally) given by

$$\sum_y \left(x \nearrow y \right) \times p(x \to y) \;=:\; \underbrace{f(x)}_{\text{drift of the system}}$$

More formally, we look at
$f(\mathbf{z}) = \lim_{dt \to 0} \frac{1}{dt} \mathbb{E}\left[\mathbf{Z}(t + dt) - \mathbf{Z}(t) \mid \mathbf{Z}(t) = \mathbf{z}\right].$

How to obtain the mean field limit?

Drift of the the JSQ(2) model:

$$f_i(\mathbf{z}) = \mu z_{i+1} + \lambda((\sum_{j\geq i-1} z_j)^2 - (\sum_{j\geq i} z_i)^2)$$

$$- \mu z_i - \lambda((\sum_{j\geq i} z_j)^2 - (\sum_{j\geq i+1} z_i)^2), \ i \geq 1$$

$$f_0(\mathbf{z}) = \mu z_1 - \lambda((\sum_{j\geq 0} z_j)^2 - (\sum_{j\geq 1} z_i)^2)$$

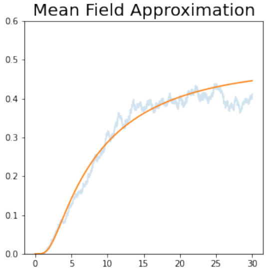
Mean Field Approximation

Solution to the initial value problem (IVP):

$$\dot{\mathbf{z}}(t) = f(\mathbf{z}), \ \mathbf{z}(0) = \mathbf{Z}(0)$$

Accuracy homogeneous case / known results

- $\sup_{s\leq t} \left\|Z^{(n)}(s) - z(s)\right\| \xrightarrow{n\to\infty} 0$ in probability [1]
- $\mathbb{E}\left[\left\|Z^{(n)}(s) - z(s)\right\|\right] = O(\frac{1}{\sqrt{n}})$
- $\mathbb{E}\left[Z^{(n)}(s)\right] - z(s) = O(\frac{1}{n})$



- same reasoning can be broadly applied
- results generalize to populations of n identical / homogeneous objects (use occupancy measure to describe system)

[1] $(\forall\epsilon > 0 \ \mathbb{P}(|Z^{(n)} - Z| > \epsilon) \to 0)$

Table of Contents

What can we say if objects are heterogeneous?*

* i.e. servers have different speeds,
people are more susceptible / recover faster,
cache objects have varying popularity

Classical Approach

- consider a finite and fixed number of classes C
- within each class $c \in C$ objects have identical behavior
- use mean field approach by letting the numbers of objects n_c of each class tend to infinity

Classical Approach

- consider a finite and fixed number of classes C
- within each class $c \in C$ objects have identical behavior
- use mean field approach by letting the numbers of objects n_c of each class tend to infinity

What if all objects have differing behavior (true heterogeneity)?

The Heterogeneous Framework

- n **interacting objects** which have (finitely many) **states** $\mathcal{S}$
- object **interactions are stochastic**
- described by $S^{(n)}(t) = (S_1(t), \dots, S_n(t)) \in \mathcal{S}^n$ (continuous time process)
- $S_k(t) \sim$ state of object k at time t
- can still use $X_{(k,s)}(t) = \mathbf{1}_{\{S_k(t)=s\}}$ to represent the system
- $Z(t) = (\frac{1}{n}\sum_k X_{k,s}(t))_s$ does not capture state of single objects

Heterogeneous Power-of-2-choice (JSQ(2)) model

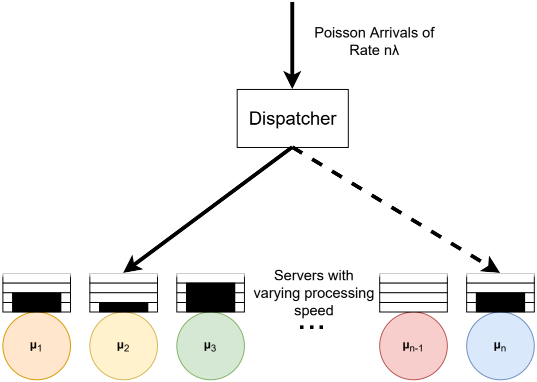
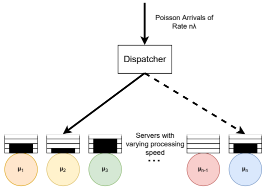


Figure: Power-of-2-choice model with differing server rates.

Transitions



$$\mathbf{X} \mapsto \mathbf{X} - e_{(k,i)} + e_{(k,i-1)}$$
$$\text{at rate } \mu_k X_{(k,i)} \quad \text{(unilateral)}$$

$$\mathbf{X} \mapsto \mathbf{X} + e_{(k,i+1)} - e_{(k,i)}$$
$$\text{at rate } \lambda X_{(k,i)}(\sum_k \sum_{j \geq i} \frac{X_{(k,j)}}{n} + \sum_k \sum_{j \geq i+1} \frac{X_{(k,j)}}{n})$$
$$\text{(pairwise)}$$

Definition of the Mean Field

The drift is

$$f(\mathbf{X}(t)) = \lim_{dt \to 0} \frac{1}{dt} \mathbb{E}\left[\mathbf{X}(t+dt) - \mathbf{X}(t) \mid \mathbf{X}(t) = \mathbf{x}\right]$$

- f can be expressed in unilateral and pairwise transition terms
- entries of f are polynomials

Definition of the Mean Field

The drift is

$$f(\mathbf{X}(t)) = \lim_{dt \to 0} \frac{1}{dt} \mathbb{E}\left[\mathbf{X}(t+dt) - \mathbf{X}(t) \mid \mathbf{X}(t) = \mathbf{x}\right]$$

- f can be expressed in unilateral and pairwise transition terms
- entries of f are polynomials
- mean field approximation is solution to the IVP:

$$\frac{d}{dt}\mathbf{x}(t) = f(\mathbf{x}(t)); \quad \mathbf{x}(0) = \mathbf{X}(0)$$

Results

Assumptions

Existence of uniform bound $\bar{r}$ independent of system size n for transition rates (unilateral, pairwise, ...).
By definition, the drift is a polynomial of degree d and
the mean field lies in some compact set $\subset [0,1]^{n \times S}$ $\rightarrow$ drift is Lipschitz

Results

Assumptions

Existence of uniform bound $\bar{r}$ independent of system size n for transition rates (unilateral, pairwise, ...).
By definition, the drift is a polynomial of degree d and
the mean field lies in some compact set $\subset [0,1]^{n \times S} \rightarrow$ drift is Lipschitz

Accuracy

For the mean field approximation $\mathbf{x}(t)$ we have

$$\mathbb{P}\left(S_k(t) = s\right) = \mathbb{E}\left[X_{(k,s)}(t)\right] = x_{(k,s)}(t) + O(1/n)$$

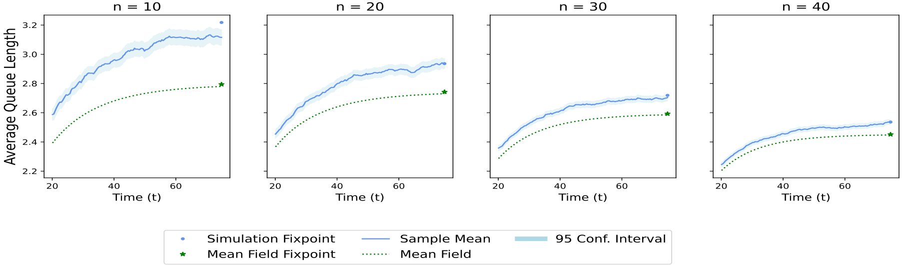


Figure: Average queue size; sample mean vs. mean field approximation vs. refined mean field approximation.

Idea of the Proof

- want to study $\mathbb{E}\left[X_{(k,s)}(t)\right] - x_{(k,s)}(t)$ by relating the generators of the stochastic system and ODE
- define $\psi(\tau) = \mathbb{E}\left[\mathbf{x}(\mathbf{X}(\tau), t-\tau)\right]$ with $\mathbf{x}(\mathbf{X}(\tau), t-\tau)$ the mean field approximation starting in $\mathbf{X}(\tau)$
- rewrite $\mathbb{E}[\mathbf{X}(t) - \mathbf{x}(\mathbf{X}(0), t)] = \psi(t) - \psi(0)$
- justify $\psi(t) - \psi(0) = \int_0^t \frac{d}{ds}\psi(s)ds$ and bound $\frac{d}{ds}\psi(s)$

Refinement

Possible to define a **refinement term $\boldsymbol{v(t)}$** following the ideas of [Gast, Van Houdt] [Gast et al.] to **increase accuracy**.
We call $\mathbf{x}(t) + \mathbf{v}(t)$ the refined mean field.

Increased Accuracy

$$\mathbb{P}\left(S_k(t) = s\right) = \mathbb{E}\left[X_{(k,s)}(t)\right] = x_{(k,s)}(t) + v_{(k,s)}(t) + O(1/n^2)$$

Proof Idea

- follows similar idea as for mean field
- use equality $\mathbb{E}[\mathbf{X}(t) - \mathbf{x}(\mathbf{X}(0), t)] = \psi(t) - \psi(0) = \int_0^t \frac{d}{ds}\psi(s)ds$
- look at $\mathbb{E}[\mathbf{X}(t) - \mathbf{x}(\mathbf{X}(0), t)] - \mathbf{v}(\mathbf{X}(0), t) = \int_0^t \frac{d}{ds}\psi(s) - \frac{d}{ds}v(s)ds$

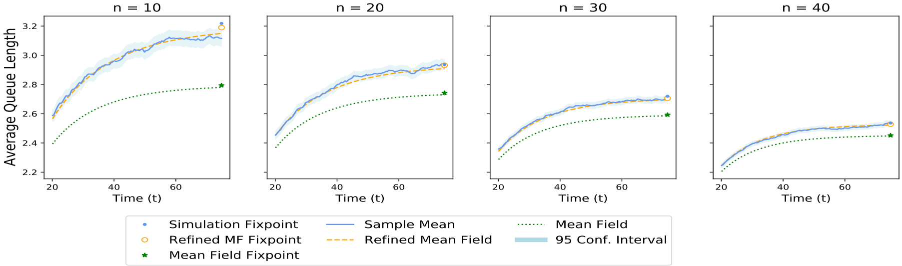


Figure: Average queue size; sample mean vs. mean field approximation vs. refined mean field approximation.

Table of Contents

Numerical Toolbox [2]

rmf_tool – A library to Compute (Refined) Mean Field Approximation(s)

Model Implementation

In [2]:
```
import rmf_tool.src.rmf_tool as rmf

# This code creates an object that represents a "density dependent population process"
ddpp = rmf.DDPP()

# Set parameters
mu, _lambda, K = 1.0, 0.9, 9
```

In [4]:
```
# Add transitions using mathematical formulation:
for i in range(K):
    if i >= 1:
        ddpp.add_transition(e(i), eval('lambda x: _lambda*(x[{}]*x[{}] - x[{}]*x[{}] )'.format(i-1,i-1,i,i) )) # arrivals
    if i < K-1 and i > 0:
        ddpp.add_transition(-e(i), eval('lambda x: mu*(x[{}] - x[{}])'.format(i,i+1))) # removals
```

Supports Density Dependent Population Processes & heterogeneous Framework

[2] https://github.com/ngast/rmf_tool

Numerical Toolbox [3]

In [15]:
```
# Set nitial state
ddpp.set_initial_state(e(0))

# Calculate mean field
T, x_transient = ddpp.ode(time=30)
```

In [16]:
```
# Simulate a trajectory for N=50
T_n50, X_n50 = ddpp.simulate(N=50,  time=30)
# and for N=1000
T_n1k, X_n1k = ddpp.simulate(N=1000,time=30)
```

Allows to easily obtain (for transient and steady state):
mean field, refined mean field approximation, simulations

[3] https://github.com/ngast/rmf_tool

Table of Contents

Conclusion

- mean field method is broadly applicable to population models
- historically used for homogeneous and partially heterogeneous systems
- can be applied to fully heterogeneous (population) interaction models

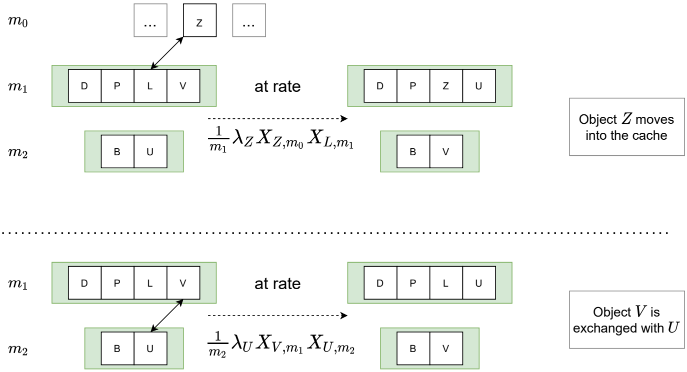
Conclusion

- mean field method is broadly applicable to population models
- historically used for homogeneous and partially heterogeneous systems
- can be applied to fully heterogeneous (population) interaction models

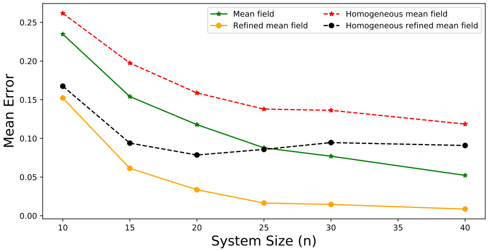
Thank you!

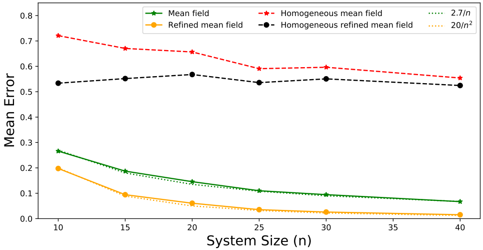
A Caching Example

Requests for object k arrive according to a Poisson process with intensity λ_k



Comparison of homogeneous and heterogeneous approximations





$$\dot{v}_{(k_1,s_1)}(t) = \sum_{u\in\mathcal{I}} \frac{\partial f_{(k_1,s_1)}}{\partial x_u}(\mathbf{x}(t))v_u(t) + \frac{1}{2}\sum_{u,l\in\mathcal{I}} \frac{\partial^2 f_{(k_1,s_1)}}{\partial x_l \partial x_u}(\mathbf{x}(t))w_{u,l}(t),$$

$$\dot{w}_{(k_1,s_1),(k_2,s_2)}(t) = \sum_{u\in\mathcal{I}} w_{u,(k_2,s_2)}(t)\frac{\partial f_{(k_1,s_1)}}{\partial x_u}(\mathbf{x}(t))$$

$$+ \sum_{u\in\mathcal{I}} w_{u,(k_1,s_1)}(t)\frac{\partial f_{(k_2,s_2)}}{\partial x_u}(\mathbf{x}(t)) + Q_{(k_1,s_1),(k_2,s_2)}(\mathbf{x}(t)),$$

$$Q_{(k,s),(k_1,s_1)}(\mathbf{x}) = \lim_{dt\to 0}\frac{1}{dt}\mathbb{E}\left[(\mathbf{X}(t+dt) - \mathbf{X}(t))^{\otimes 2}_{(k,s),(k_1,s_1)} \mid \mathbf{X}(t) = x\right]$$

References

Nicolas Gast and Benny Van Houdt 2017
A Refined Mean Field Approximation. Proceedings of the ACM on Measurement and Analysis of Computing Systems 1, 2

Nicolas Gast, Luca Bortolussi, and Mirco Tribastone 2019
Size Expansions of Mean Field Approximation: Transient and Steady-State Analysis. Performance Evaluation 129 (Feb. 2019)

Sebastian Allmeier and Nicolas Gast 2021
Mean Field and Refined Mean Field Approximations for Heterogeneous Systems: It Works! arXiv preprint arXiv:2111.01594

Sebastian Allmeier and Nicolas Gast 2021
rmf_tool – a library to compute (refined) mean field approximation(s). SIGMETRICS Perform. Eval. Rev. (Workshop).