Cooperating stochastic automata: approximate lumping an reversed process

Simonetta Balsamo   Gian-Luca Dei Rossi   Andrea Marin

Dipartimento di Scienze Ambientali, Informatica e Statistica
Università Ca' Foscari, Venezia

ISCIS '12, Paris, 3-4 October 2012

Context: Cooperating stochastic models

- Models with underlying Continuous Time Markov Chain (CTMC)
- Exploitation of compositionality in model definition
  - Each component is specified in isolation
  - Semantics of cooperation is defined so that the joint model can be algorithmically derived
- Stochastic automata considered here synchronise on the active/passive semantics
  - Performance Evaluation Process Algebra (PEPA) active/passive synchronisation
  - Buchholz's Communicating Markov Processes
  - Plateau's stochastic automata networks (SAN) with master/slave cooperation
  - ...

Motivation

- In general, the state-space's cardinality of the joint model grows exponentially with the number of components
- Steady-state analysis becomes quickly unfeasible
  - Space cost
  - Time cost
  - Numerical stability issues
- Workarounds
  - Approximate analysis (e.g. fluid)
  - Exploitation of the geometry of the state space
  - Product-form decomposition
  - Lumping
    - Approximate lumping

Previous work: Lumping on cooperating automata

Definition (Lumping condition)

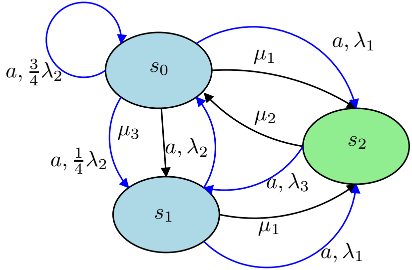
Given active automaton M_1, a set of labels $\mathcal{T}$, and a partition of the states of M_1 into N_1 clusters $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_{N_1}\}$, we say that $\mathcal{C}$ is an exact lumping for M_1 if:

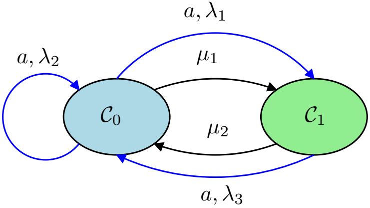
1. $\forall \mathcal{C}_i, \mathcal{C}_j,\, \mathcal{C}_i \neq \mathcal{C}_j,\, \forall s_1 \in \mathcal{C}_i \;\; \sum_{s_1' \in \mathcal{C}_k} q_1(s_1 \to s_1') = \tilde{q}_1(\mathcal{C}_i \to \mathcal{C}_j)$ not synchronising label
2. $\forall t \in \mathcal{T}, \forall \mathcal{C}_i, \mathcal{C}_j\,, \forall s_1 \in \mathcal{C}_i \;\; \sum_{s_1' \in \mathcal{C}_k} q_1^t(s_1 \to s_1') = \tilde{q}_1^t(\mathcal{C}_i \to \mathcal{C}_k)$

where $\varphi_1^t(s_1, \tilde{s}_1') = \sum_{s_1' \in \tilde{s}_1'} q_1^t(s_1, s_1')$.

- Reduce complexity GBEs' solution through component-wise lumping
  - If both automata have a spate-space of cardinality M, time cost reduces from $\mathcal{O}((MM)^3)$ to $\mathcal{O}((NM)^3)$, where N is the number of clusters in the lumping
- Intuition: *for each synchronising label the original and lumped automata must behave (in steady-state) equivalently*
- We treat non-synchronising transitions as a special case
- Conditions are stronger than the ones for regular lumpability and weaker than for PEPA strong equivalence

Example


s_0
s_1
s_2
$a, \frac{3}{4}\lambda_2$
a, λ_1
μ_1
μ_2
μ_3
a, λ_2
$a, \frac{1}{4}\lambda_2$
a, λ_3
μ_1
a, λ_1


a, λ_1
μ_1
a, λ_2
$\mathcal{C}_0$
$\mathcal{C}_1$
μ_2
a, λ_3

Marginal distribution

Theorem

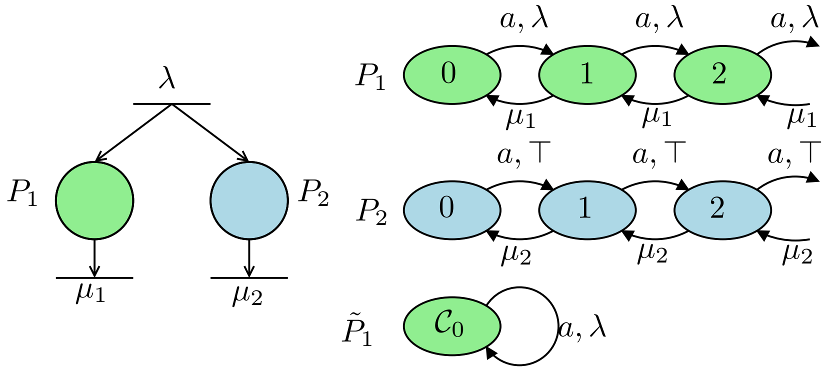
Let M_1 and M_2 be two cooperating automata, where M_2 is passive and M_1 active. If:

- *M_2 never blocks M_1*
- *$\tilde{M}_1$ is a lumped automaton of M_1*

Then the marginal steady state distribution of M_2 in the cooperations $M_1 \otimes M_2$ and $\tilde{M}_1 \otimes M_2$ are the same.

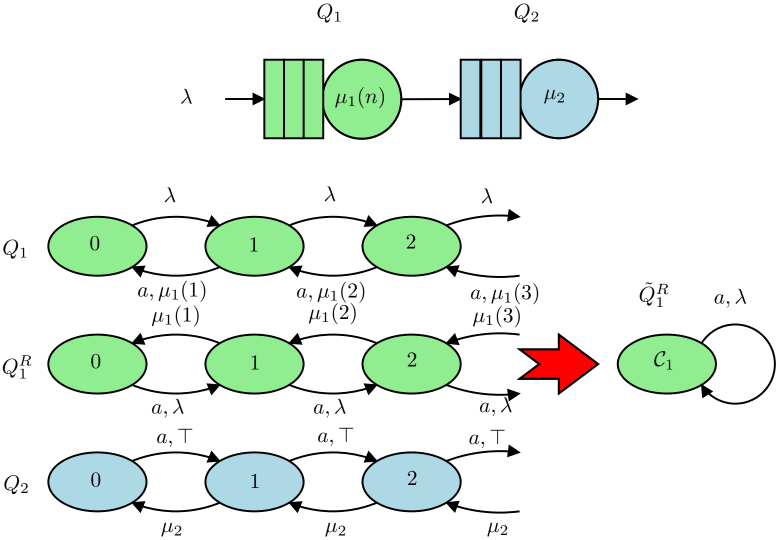
Note that ergodicity is assumed and the state-space of the joint process is the Cartesian product of the single automata state-spaces.

A trivial example



$$\pi_1(n) = \left(1 - \frac{\lambda}{\mu_1}\right)\left(\frac{\lambda}{\mu_1}\right)^n \quad \pi_2(n) = \left(1 - \frac{\lambda}{\mu_2}\right)\left(\frac{\lambda}{\mu_2}\right)^n$$

Another trivial one



Reversed lumping and product-forms

- Both previous examples allowed for a lumping into a single cluster
  - First is derived from the forward automaton
  - Second is derived from the reversed automaton
- In both cases we obtain the marginal distribution, but in the latter we also have product-form!
  - product-form $\Rightarrow$ the joint distribution is the product of the marginal ones

Corollary (Product-forms)

A synchronisation is in product-form if the reversed active automaton can be lumped into a single state

Note that, in general the marginal steady state distribution of M_2 in $\tilde{M}_1^R \otimes M_2 \simeq$ the one in $\tilde{M}_1 \otimes M_2$, and is equal in product-form models.

Approximation of marginal SSD through aggregation

- With our theorem we can reduce the cost to compute marginal steady state distributions of a cooperating automaton *if we're able to find an exact lumping of the other one*.
- What if this is not feasible or even possible?
  - We could try to find an approximated lumping.
  - Can be applied also to the reversed process.
- How we evaluate the quality of an approximation?
- How we can adapt clustering algorithms to use our definition of (approximated) exact lumping?

Evaluating the quality of an approximate lumping

How close is an arbitrary state partition $\mathcal{W}$ to an exact lumping?

- We measure the coefficient of variation of the outgoing fluxes $\phi_1^t(s_1)$ of the states in $\tilde{s}_1$.
- We further refine that measurement.

ϵ-error

Definition (ϵ-error)

Given model M_1 and a partition of states $\mathcal{W} = \{\tilde{1}, \ldots, \tilde{N}_1\}$, for all $\tilde{s}_1 \in \mathcal{W}$ and $t > 2$, we define:

$$\overline{\phi}_1^t(\tilde{s}_1) = \frac{\sum_{s_1 \in \tilde{s}_1} \pi_1(s_1)\phi_1^t(s_1)}{\sum_{s_1 \in \tilde{s}_1} \pi_1(s_1)}$$

$$\epsilon^t(\tilde{s}_1) = 1 - \exp\left(-\sqrt{\sum_{s_1 \in \tilde{s}_1} \frac{\pi_1(s_1)(\phi_1^t(s_1) - \overline{\phi}_1^t(\tilde{s}_1))^2}{\sum_{s \in \tilde{s}_1} \pi_1(s_1)}}\right).$$

where $\phi_1^t(s_1) = \sum_{s_1'=1}^{N_1} q_1^t(s_1, s_1')$.

δ-error

Definition (δ-error)

Given model M_1 and a partition of states $\mathcal{W} = \{\tilde{1}, \ldots, \tilde{N}_1\}$, for all $\tilde{s}_1, \tilde{s}'_1 \in \mathcal{W}$, we define:

$$\overline{\varphi}_1^t(\tilde{s}_1, \tilde{s}'_1) = \begin{cases} 0 & \tilde{s}_1 = \tilde{s}'_1 \wedge t = 1 \\ \frac{\left(\sum_{s_1 \in \tilde{s}_1} \pi_1(s_1) \varphi_1^t(s_1, \tilde{s}'_1)\right)}{\sum_{s_1 \in \tilde{s}_1} \pi_1(s_1)} & \text{otherwise} \end{cases}$$

$$\left(\sigma^t(\tilde{s}_1, \tilde{s}'_1)\right)^2 = \sum_{s_1 \in \tilde{s}_1} \frac{\pi_1(s_1)(\varphi_1^t(s_1, \tilde{s}'_1) - \overline{\varphi}_1^t(\tilde{s}_1, \tilde{s}'_1))^2}{\sum_{s \in \tilde{s}_1} \pi_1(s)}$$

$$\delta^t(\tilde{s}_1, \tilde{s}'_1) = 1 - e^{-\sigma(\tilde{s}_1, \tilde{s}'_1)}$$

where function φ_1^t has been defined in Lumping conditions.

An ideal algorithm

Definition (Ideal algorithm)

- **Input:** automata M_1, M_2, $\mathcal{T}$, tolerances $\epsilon \geq 0$, $\delta \geq 0$
- **Output:** marginal distribution π_1 of M_1; approximated marginal distribution of M_2

1. Find the minimum $\tilde{N}_1'$ such that there exists a partition $\mathcal{W} = \{\tilde{1}, \ldots, \tilde{N}_1'\}$ of the states of M_1 such that $\forall t \in \mathcal{T}, t > 2$ and $\forall \tilde{s}_1 \in \mathcal{W}\ \epsilon(\tilde{s}_1) \leq \epsilon$
2. Let $\mathcal{W}' \leftarrow \mathcal{W}$
3. Check if partition $\mathcal{W}'$ is such that $\forall t \in \mathcal{T}$, $\forall \tilde{s}_1, \tilde{s}_2 \in \mathcal{W}$, $\tilde{s}_1 \neq \tilde{s}_2$, $\delta^t(\tilde{s}_1, \tilde{s}_1') \leq \delta$. If this is true then return the marginal distribution of M_1 and the approximated of M_2 by computing the marginal distribution of $\tilde{M}_1 \otimes M_2$ and terminate.
4. Otherwise, refine partition $\mathcal{W}$ to obtain $\mathcal{W}^{new}$ such that the number of clusters of $\mathcal{W}^{new}$ is greater than the number of clusters in $\mathcal{W}'$. $\mathcal{W}' \leftarrow \mathcal{W}^{new}$. Repeat from Step 3

Constructing the approximate lumped automata

Definition (Approx. lumped automata)

Given active automaton M_1, a set of transition types $\mathcal{T}$, and a partition of the states of M_1 into $\tilde{N}_1$ clusters $\mathcal{W} = \{\tilde{1}, \tilde{2}, \ldots, \tilde{N}_1\}$, then we define the automaton $M_1^{\simeq}$ as follows:

$$
\begin{aligned}
\tilde{\mathbf{E}}_{11}(\tilde{s}_1, \tilde{s}_1') &= \begin{cases} \overline{\varphi}_1^1(\tilde{s}_1, \tilde{s}_1')\tilde{\lambda}_1^{-1} & \text{if } \tilde{s}_1 \neq \tilde{s}_2 \\ 0 & \text{otherwise} \end{cases} \\
\tilde{\mathbf{E}}_{12} &= \mathbf{I}, \\
\tilde{\mathbf{E}}_{1t}(\tilde{s}_1, \tilde{s}_1) &= \overline{\varphi}_1^t(\tilde{s}_1, \tilde{s}_1')\lambda_t^{-1} \quad t > 2
\end{aligned}
$$

where

$$
\tilde{\lambda}_t = \max_{\tilde{s}_1 = 1, \ldots, \tilde{N}_1} \left(\sum_{\tilde{s}_1' = 1}^{\tilde{N}_1} \overline{\varphi}_1^t(\tilde{s}_1, \tilde{s}_1') \right)
$$

are the rates associated with the transition types in the cooperation between $M_1^{\simeq}$ and M_2.

Initial clustering and refinement phase

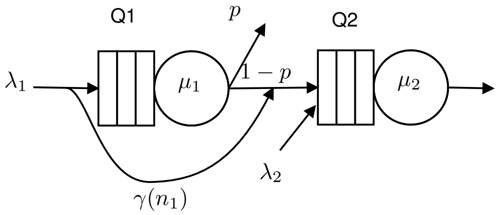
Initial clustering:

- similarity measure can be Euclidean distance between $(\phi_1^3(s_1), \ldots, \phi_1^T(s_1))$ and $(\phi_1^3(s_1'), \ldots, \phi_1^T(s'))$
- can be implemented using various algorithm
  - hierarchical clustering
  - K-means (but number of clusters must be decided a priori...)
  - ...

Refinement phase:

- using the tolerance constant δ
- distances between clusters depend on clusters themselves $\Longrightarrow$ K-means cannot be used.
- spectral analysis or iterative algorithms

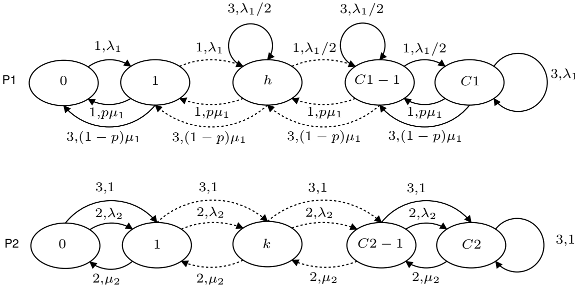
Example



where

$$\gamma(n_1) = \begin{cases} 0 & \text{if } n_1 \leq \left\lfloor \frac{C1}{2} \right\rfloor \\ \frac{\lambda_1}{2} & \text{if } \left\lfloor \frac{C1}{2} \right\rfloor < n_1 < C1 \\ \lambda_1 & \text{if } n_1 = C1 \end{cases}$$

Example



Not exactly lumpable. For $C1 = 20, C2 = 20, \lambda_1 = 6, \lambda_2 = 1, \mu_1 = 4, \mu_2 = 4, p = 0.7$, $\epsilon = 10^{-13}$ and $\delta = 0.95$ we could find

- $L_1 = \{0\}$, $L_2 = \{1, \ldots, 10\}$, $L_3 = \{11, \ldots, 19\}$ and $L_4 = \{20\}$ on the forward process
- $\overline{L_1} = \{0, \ldots, 10\}$, $\overline{L_2} = \{11, 12\}$, $\overline{L_3} = \{13, \ldots, 19\}$ and $\overline{L_4} = \{20\}$ on the reversed one

Comparison

	FW-Lump	RV-Lump	APF	FPA	Exact
KL div.	0.0065	0.0045	0.0451	0.0112	0
$E[N]$	11.62	11.55	9.990	11.80	11.33
Rel. err.	0.0259	0.0200	0.1178	0.0424	0

Where

- APF is the Approximated Product Form of order 4 [Buchholz, 2010]
- PFA is the Fixed Point Approximation [Miner et al., 2000]

Conclusion

- Lumping of automata can be applied to other formalisms
- Approximate lumpings can be used to derive approximate marginal distributions
  - Several examples show that in case of queueing networks lumping the reversed automata gives better approximations!
- Future works: definition of efficient algorithms
  - The algorithm proposed in [Gilmore et al., 2001] based on strong equivalence can be adapted to consider our notion of lumpability
    - also for reversed automata
    - optimality issues

References

[Balsamo et al., 2012] Balsamo, S., Dei Rossi, G., and Marin, A. (2012). Lumping and reversed processes in cooperating automata. In *LNCS 7314, Proc. of Int. Conf. ASMTA*, pages 212–226, Grenoble, FR. Springer.

[Buchholz, 2010] Buchholz, P. (2010). Product form approximations for communicating Markov processes. *Perf. Eval.*, 67(9):797 – 815. Special Issue: QEST 2008.

[Gilmore et al., 2001] Gilmore, S., Hillston, J., and Ribaudo, M. (2001). An Efficient Algorithm for Aggregating PEPA Models. *IEEE Trans. on Software Eng.*, 27(5):449–464.

[Miner et al., 2000] Miner, A. S., Ciardo, G., and Donatelli, S. (2000). Using the exact state space of a markov model to compute approximate stationary measures. In *Proc. of ACM SIGMETRICS*, pages 207–216, New York, NY, USA. ACM.

Thanks!

Thanks for the attention

any question?