

COMPUTATIONAL MODELLING ON COMPLEX NETWORKS DYNAMIC PROCESSES ON NETWORKS

Luca Bortolussi^{1,2}

¹Modelling and Simulation Group
Department of Computer Science
Universitat des Saarlandes

²Department of Mathematics and Geosciences
University of Trieste

Office 303, third floor, E1 3
luca@dmi.units.it or luca@cs.uni-saarland.de

TRIESTE, 2016 !!

OUTLINE

① INTRODUCTION TO DYNAMIC NETWORK MODELS

- Continuous Time Markov Chains
- Graph Rewriting Rules
- Simulating network models

② EPIDEMIC SPREADING AND CONTACT PROCESSES

- The homogeneous setting
- The heterogeneous setting

③ PAIR APPROXIMATION

OUTLINE

1 INTRODUCTION TO DYNAMIC NETWORK MODELS

- Continuous Time Markov Chains
- Graph Rewriting Rules
- Simulating network models

2 EPIDEMIC SPREADING AND CONTACT PROCESSES

- The homogeneous setting
- The heterogeneous setting

3 PAIR APPROXIMATION

THE MAIN IDEA

- There is a network $G = (V, E)$.
- Each node in the network is identified as a different agent, which are (inter)acting and can have many internal states.
- **Agent states S** are usually discrete (but can be continuous, too).
- For us, states will be discrete and finite, and we will represent them with colours.
- We will formally talk of labelled or typed graphs $G = (V, E, \sigma)$, with $\sigma : V \rightarrow S$.
- Agents interact. The network structure represent the possible interaction. Two agents can interact only if there is an edge between them.
- The graph structure is often considered static, but it can be also dynamically changing (node addition/ removal and edge addition/ removal).

IMPORTANCE

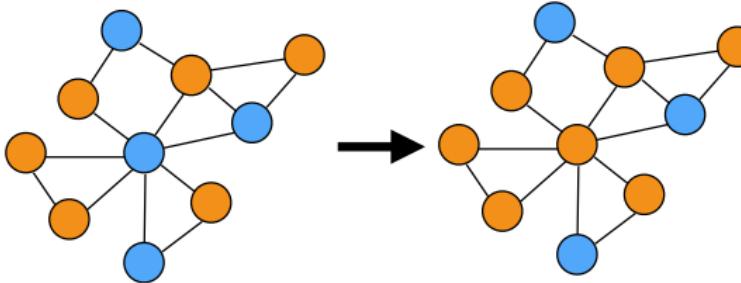
This agent-based framework is very general, and many different scenarios can be easily modelled. Examples include:

- The spreading of an epidemic in a population (of humans, of computers, ...);
- The spreading of a gossip in a social network.
- The dynamics of political opinions in a society.
- The dynamics of population of cells, e.g. tumour growth.
- The dynamics of inter-banking dependency.
- Biochemical reactions, like the formation of polymeric plaques in Alzheimer disease.
- ...

A SUPERSIMPLE MODEL

Consider the following supersimple scenario:

- Nodes have two states/ colours: orange and blue.
- If a blue node has an orange neighbour, it can turn orange.



Intuitively, if we iteratively apply this rule to a connected graph with at least one orange node, all nodes will eventually become orange. This notion of eventually requires us to specify a model of time and of the dynamics.

We will stick to a Markovian stochastic dynamics in continuous time.

EXPONENTIAL DISTRIBUTION

DEFINITION

A random variable $T : (\Omega, \mathcal{S}) \rightarrow [0, \infty]$ is $\text{Exp}(\lambda)$ iff

- Cdf is $\mathbb{P}(T < t) = 1 - e^{-\lambda t}$
- Survival probability is $\mathbb{P}(T > t) = e^{-\lambda t}$
- Density is $f_T(t) = \lambda e^{-\lambda t}, t \geq 0.$

The expected value of T is $\mathbb{E}(T) = \int_0^\infty \mathbb{P}(T > t) dt = \frac{1}{\lambda}$.

MEMORYLESS PROPERTY

$T \sim \text{Exp}(\lambda)$ if and only if the following **memoryless property** holds:

$$\mathbb{P}(T > s + t | T > s) = \mathbb{P}(T > t) \text{ for all } s, t \geq 0.$$

In fact

$$\mathbb{P}(T > s + t | T > s) = \mathbb{P}(T > s + t) / \mathbb{P}(T > s) = e^{-\lambda(s+t)} e^{\lambda s} = e^{-\lambda t}.$$

EXPONENTIAL DISTRIBUTION

DEFINITION

A random variable $T : (\Omega, \mathcal{S}) \rightarrow [0, \infty]$ is $\text{Exp}(\lambda)$ iff

- Cdf is $\mathbb{P}(T < t) = 1 - e^{-\lambda t}$
- Survival probability is $\mathbb{P}(T > t) = e^{-\lambda t}$
- Density is $f_T(t) = \lambda e^{-\lambda t}, t \geq 0.$

The expected value of T is $\mathbb{E}(T) = \int_0^\infty \mathbb{P}(T > t) dt = \frac{1}{\lambda}$.

INSTANTANEOUS FIRING PROBABILITY

An exponential distribution with rate λ can be seen as the **firing time** of an event who has **probability of firing between time t and $t + dt$ equal to λdt** .

Call $p(t) = \mathbb{P}\{T \geq t\}$. Then $p(t + dt) = p(t) \cdot (1 - \lambda dt)$, from which $\frac{dp(t)}{dt} = -\lambda p(t)$, that has solution $p(t) = e^{-\lambda t}$ (as $p(0) = 1$).

EXPONENTIAL DISTRIBUTION

DEFINITION

A random variable $T : (\Omega, \mathcal{S}) \rightarrow [0, \infty]$ is $\text{Exp}(\lambda)$ iff

- Cdf is $\mathbb{P}(T < t) = 1 - e^{-\lambda t}$
- Survival probability is $\mathbb{P}(T > t) = e^{-\lambda t}$
- Density is $f_T(t) = \lambda e^{-\lambda t}, t \geq 0.$

The expected value of T is $\mathbb{E}(T) = \int_0^\infty \mathbb{P}(T > t) dt = \frac{1}{\lambda}$.

THEOREM

Let I be a countable set and let $T_k, k \in I$, be independent random variables, with $T_k \sim \text{Exp}(q_k)$ and $q = \sum_{k \in I} q_k < \infty$. Set $T = \inf_k T_k$. Then this infimum is obtained at a unique random value K of I , with probability 1. Moreover, T and K are independent, $T \sim \text{Exp}(q)$ and $\mathbb{P}(K = k) = q_k/q$.

CTMC: DEFINITION

CONTINUOUS TIME MARKOV CHAIN

A Continuous Time Markov Chain is a (right-continuous) continuous-time stochastic process satisfying the **memoryless condition**: for each n , t_i and s_i :

$$\mathbb{P}(X_{t_n} = s_n \mid X_{t_0} = s_0, \dots, X_{t_{n-1}} = s_{n-1}) = \mathbb{P}(X_{t_n} = s_n \mid X_{t_{n-1}} = s_{n-1}).$$

CTMC AS A GRAPH

A CTMC on a state space S can be seen as a **labelled graph**. Each edge takes some time to be crossed, exponentially distributed with the rate labelling the edge.

In each state, there is a **race condition** between the different exiting edges: **the fastest is traversed**.

The memoryless property follows from that of the exponential distribution.

CTMC: Q MATRIX

CONTINUOUS TIME MARKOV CHAIN

A **Continuous Time Markov Chain** is a (right-continuous) continuous-time stochastic process satisfying the **memoryless condition**: for each n , t_i and s_i :

$$\mathbb{P}(X_{t_n} = s_n \mid X_{t_0} = s_0, \dots, X_{t_{n-1}} = s_{n-1}) = \mathbb{P}(X_{t_n} = s_n \mid X_{t_{n-1}} = s_{n-1}).$$

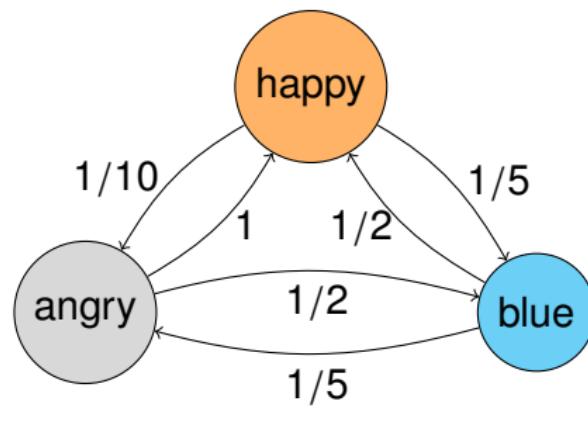
Q-MATRIX

A **Q-matrix** is the $|S| \times |S|$ matrix such that:

- ① $q_{ij} \geq 0$, $i \neq j$ is the rate of the exponential distribution giving the time needed to go from state s_i to state s_j
- ② $q_{ii} = -\sum_{j \neq i} q_{ij}$ is the opposite of the **exit rate** from state i .

Therefore, each row of the *Q*-matrix sums up to zero.

A SIMPLE EXAMPLE: THE MOOD CHAIN



$$S = \{ \textit{happy}, \textit{blue}, \textit{angry} \}$$

$$Q = \begin{pmatrix} -\frac{3}{10} & \frac{1}{5} & \frac{1}{10} \\ \frac{1}{2} & -\frac{7}{10} & \frac{1}{5} \\ 1 & \frac{1}{2} & -\frac{3}{2} \end{pmatrix}$$

JUMP CHAIN AND HOLDING TIMES

FACTORIZING EACH JUMP

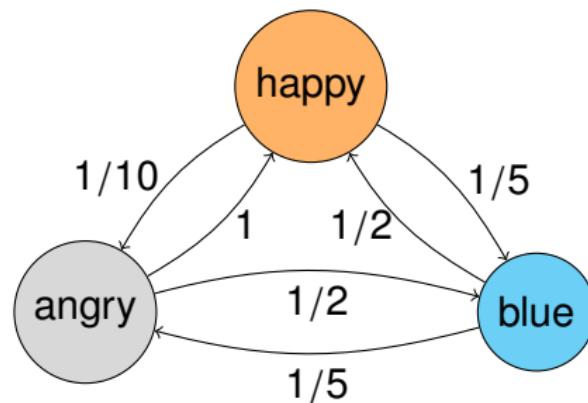
In each state i , we have a race condition between k transitions, each exponentially distributed with rate q_{ij} . Hence, the time spent is $T = \inf T_{ij}$. By the properties of the exponential distribution, we know that T has rate $q_i = \sum_j q_{ij}$, and that the transition that fires is independent from T and the next state j is chosen with probability q_{ij}/q_i .

JUMP CHAIN AND HOLDING TIMES

We can therefore factorize $X(t)$ into

- a DTMC Y_n , with probability matrix Π , defined by $\pi_{ij} = \frac{q_{ij}}{-q_{ii}}$, if $i \neq j$, and $\pi_{ii} = 0$;
- a sequence of **jump times** τ_n , where τ_n is the time of the n -th jump. Letting q_i the jump rate from state s_i , then $T_n = \tau_n - \tau_{n-1}$, the n -th **holding time**, is distributed exponentially with rate q_{Y_n} .
- Y_n and each T_i are **independent**.
- Hence $X(t) = Y_n$ for $\tau_n \leq t < \tau_{n+1}$.

A SIMPLE EXAMPLE: THE MOOD CHAIN



$$S = \{ \text{happy}, \text{blue}, \text{angry} \}$$

Jump chain

$$\Pi = \begin{pmatrix} 0 & \frac{2}{3} & \frac{1}{3} \\ \frac{5}{7} & 0 & \frac{2}{7} \\ \frac{2}{3} & \frac{1}{3} & 0 \end{pmatrix}$$

Exit rates

$$q = \left(\frac{3}{10}, \frac{7}{10}, \frac{3}{2} \right)$$

CHAPMAN-KOLMOGOROV EQUATIONS

Let $P_{ij}(t) = \mathbb{P}\{X(t) = s_j \mid X(0) = s_i\}$. Then

$$\begin{aligned} P_{ij}(t+s) &= \mathbb{P}\{X(t+s) = s_j \mid X(0) = s_i\} \\ &= \sum_k \mathbb{P}\{X(t+s) = s_j, X(t) = s_k \mid X(0) = s_i\} \\ &= \sum_k \mathbb{P}\{X(s) = s_j \mid X(0) = s_k\} \mathbb{P}\{X(t) = s_k \mid X(0) = s_i\} \\ &= \sum_k P_{ik}(s) P_{kj}(t). \end{aligned}$$

Hence $P(t)$, as a matrix, satisfies

$$P(t+s) = P(t)P(s) = P(s)P(t),$$

which is the **semigroup** property, also known as
Chapman-Kolmogorov equations.

KOLMOGOROV EQUATIONS

Using properties of the exponential, we can compute $P(dt)$:

- $P_{ij}(dt) = q_{ij}dt$, for $i \neq j$;
- $P_{ii}(dt) = 1 - \sum_{j \neq i} q_{ij}dt = 1 + q_{ii}dt$

Hence $P(dt) = I + Qdt$

From the CK equations: $P(t + dt) = P(t) + P(t)Qdt$, from which

$$\frac{dP(t)}{dt} = P(t)Q,$$

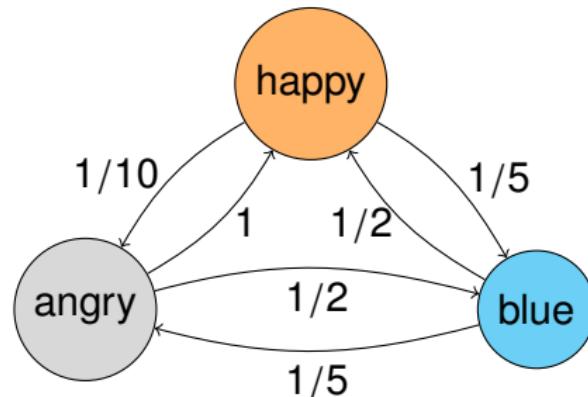
which is the **forward Kolmogorov equation**.

Using CK the other way round: $P(t + dt) = P(t) + QP(t)dt$, so

$$\frac{dP(t)}{dt} = QP(t),$$

which is the **backward Kolmogorov equation**.

A SIMPLE EXAMPLE: THE MOOD CHAIN

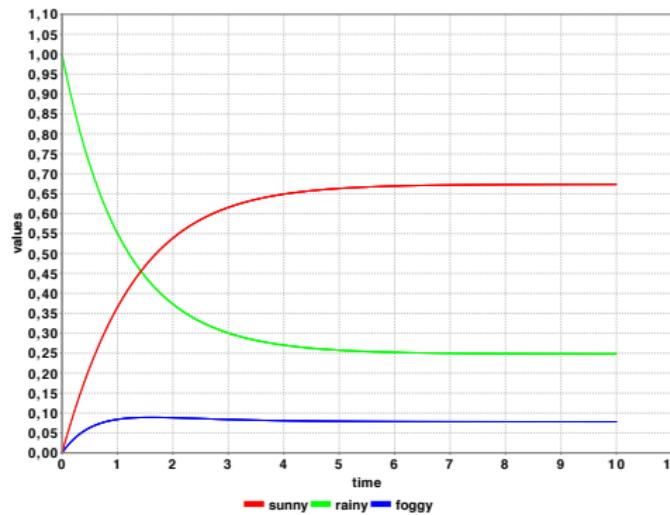
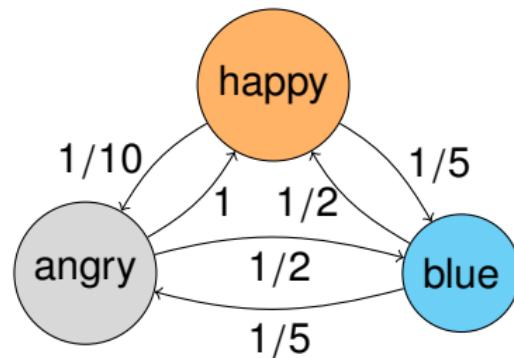


$$S = \{ \text{happy}, \text{blue}, \text{angry} \}$$

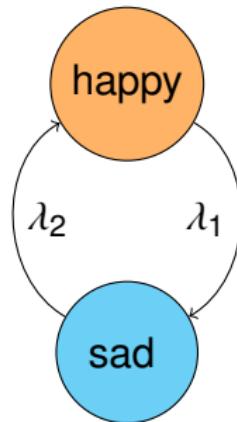
$$p_0 = (0, 1, 0) \quad p = p_0 P$$

$$\frac{d}{dt} p_0 P = p_0 P Q \Rightarrow \frac{d}{dt} p = p Q$$

A SIMPLE EXAMPLE: THE MOOD CHAIN

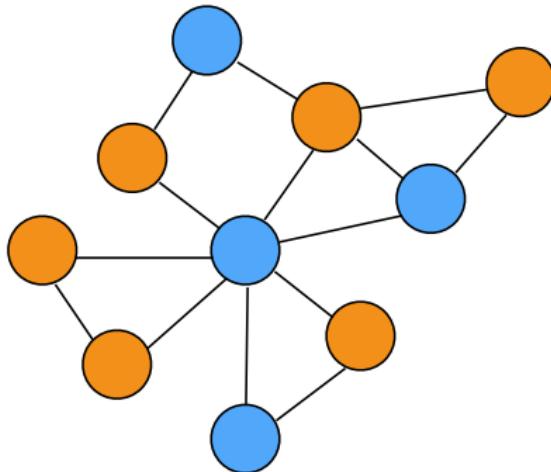


CTMC FOR NETWORK DYNAMICS



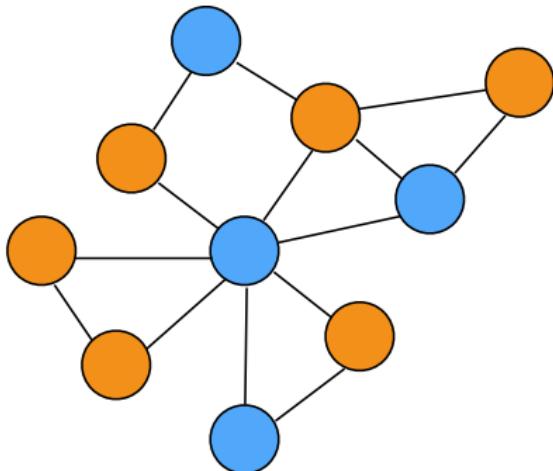
- Consider an individual which can be in two states: happy and sad.
- Surely, her happiness will be influenced by the status of other people she knows, and viceversa.
- Hence, we can look at a more interesting model, in which we consider the evolution of individual moods in the context of their social network!

CTMC FOR NETWORK DYNAMICS



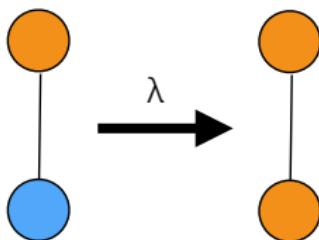
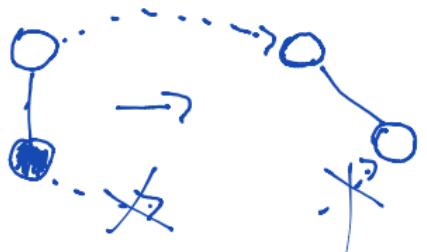
- Each individual agent is a node, which can be in different states (colours).
- Edges represent social connections, i.e. interaction capabilities.
- A state of our CTMC will then be a coloured network, and transition will modify it by changing the state/ colour of nodes and/or possibly changing the network topology.

CTMC FOR NETWORK DYNAMICS



- Intuitively, we want to define the CTMC dynamics in such a way that the change of state colour of one node is influenced by the state of (some of) its neighbours.
- For instance, we may want that a sad individual is more likely to become happy if she has many happy friends.
- We will achieve this by the notion of **graph rewriting rule**.

GRAPH REWRITING RULES



- Rules are like the one shown on the left.
- They have a left-hand-side and a right-hand-side, which are both graphs.
- Intuitively, a sub-graph like the one on the lhs can be transformed into the subgraph given in the rhs, at rate λ .
- Formally, a graph rewriting rule is a (partial) **graph morphism** between the subgraph in the lhs and the subgraph in the rhs.

GRAPH MORPHISMS

GRAPH MORPHISMS

A **graph morphism** $\rho : G \rightarrow H$ is a collection of two maps (ρ_V, ρ_E) such that

- $\rho_V : G_V \rightarrow H_V$
- $\rho_E : G_E \rightarrow H_E$
- ρ preserves the graph structure, i.e. for each $(v_1, v_2) \in G_E$, it holds $\rho_E(v_1, v_2) = (\rho_V(v_1), \rho_V(v_2))$

PARTIAL GRAPH MORPHISMS

A **partial graph morphism** $\rho : G \rightarrow H$ is a graph morphism for some subgraph $G' \subseteq G$.

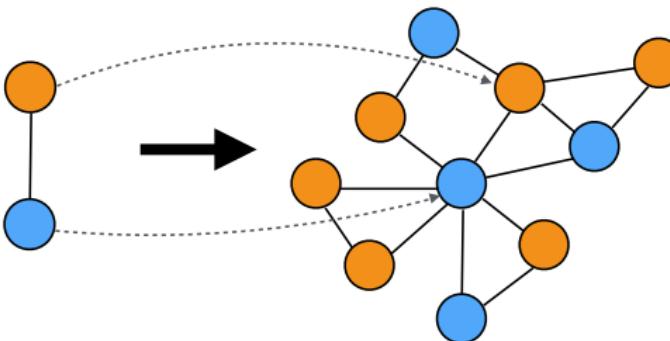
STATE PRESERVING GRAPH MORPHISMS

A **state preserving graph morphism** is a graph morphism satisfying the further constraint $\sigma(v) = \sigma(\rho_V(v))$, i.e. vertex states are preserved.

GRAPH EMBEDDINGS AND ISOMORPHISMS

GRAPH EMBEDDINGS

A **graph embedding** $\rho : G \rightarrow H$ is a total graph morphism such that ρ_V and ρ_E are injective.



GRAPH ISOMORPHISM

A **graph isomorphism** $\rho : G \rightarrow H$ is a total graph morphism such that ρ_V and ρ_E are bijective.

GRAPH TRANSFORMATION RULES

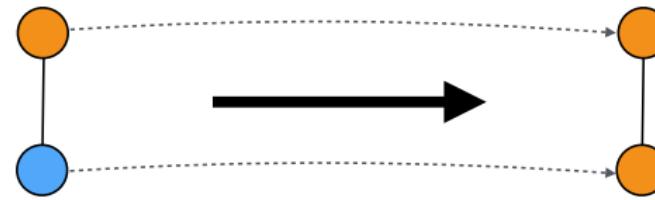
RULES

A **transformation rule** for a typed graph is a **partial graph morphism** between two graphs L and R , the left and right-hand sides of the rule: $\rho : L \rightarrow R$.

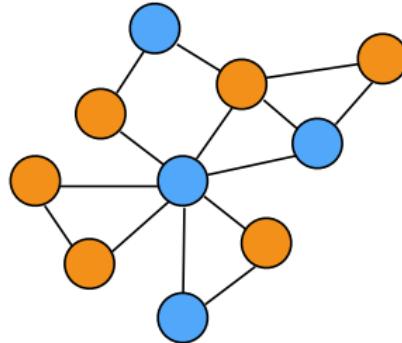
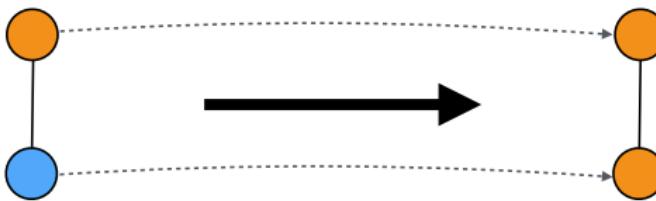
A rule $\rho : L \rightarrow R$ is applied to a graph G in the following way:

- First, we look for a state-preserving embedding $\mu : L \rightarrow G$.
- Then we replace L by R in G as follows:
 - nodes of L can be deleted by ρ , hence we delete them from $\mu(L)$, removing all incoming edges, too.
 - The subgraph of nodes of R having a preimage in L is modified according to the rule.
 - nodes in R but not in L are added also accordingly to ρ .

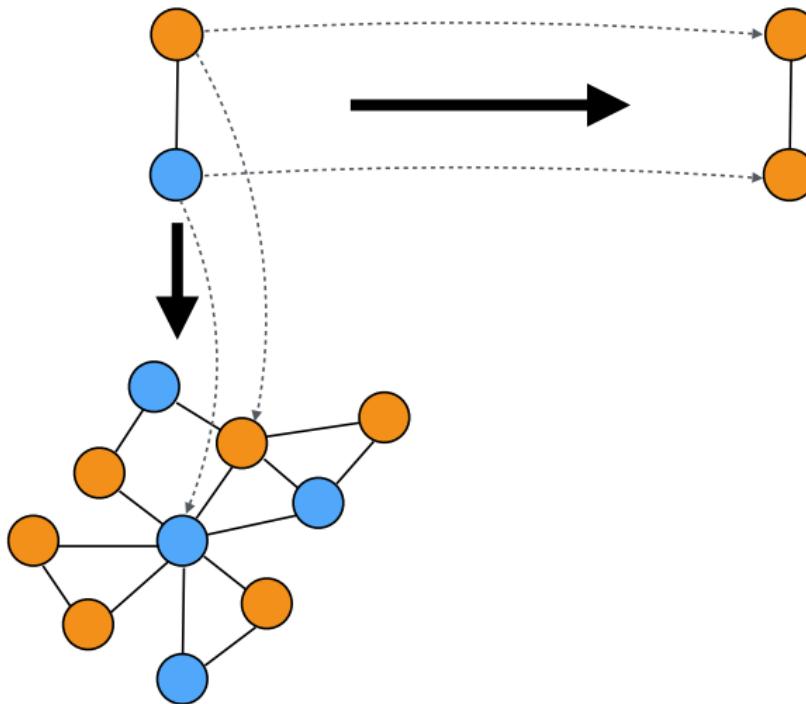
EXAMPLE OF RULE APPLICATION



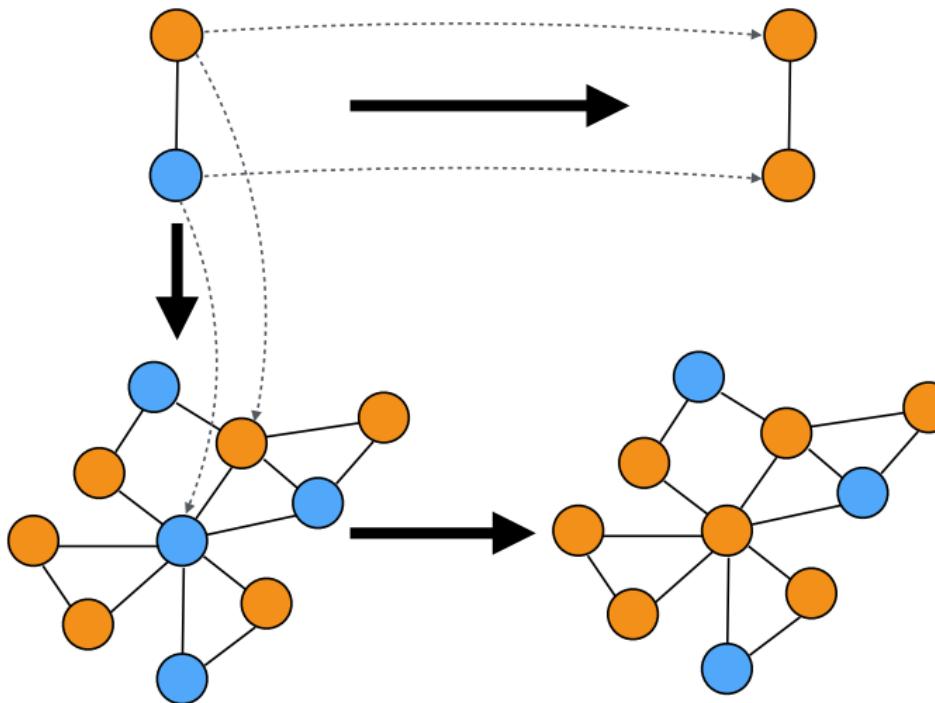
EXAMPLE OF RULE APPLICATION



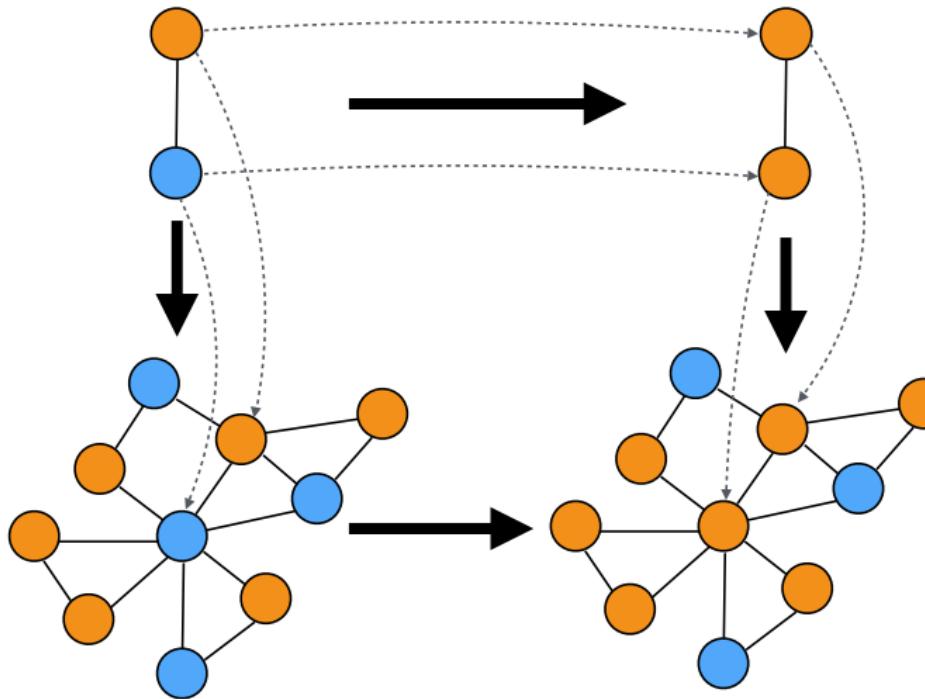
EXAMPLE OF RULE APPLICATION



EXAMPLE OF RULE APPLICATION



EXAMPLE OF RULE APPLICATION



GRAPH TRANSFORMATION SYSTEMS

GRAPH TRANSFORMATION SYSTEM

A graph transformation system (GTS) or graph rewriting system is specified by a set of rules $\{\rho_1, \dots, \rho_k\}$, and an initial graph G_0 .

All possible graphs reachable from G_0 , applying rules in each possible way, constitute the language of the GTS.

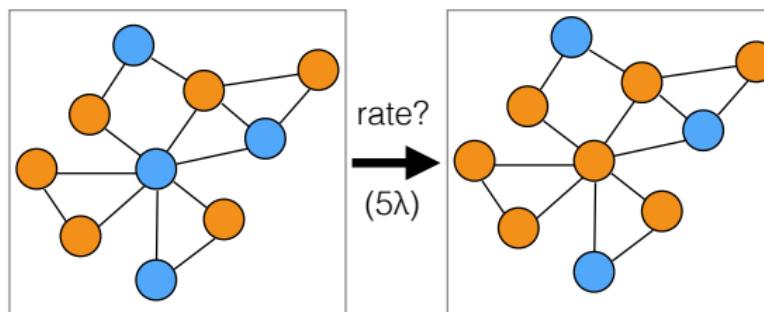
The approach sketched here is informal, but can be formalised either with a combinatorial construction or in an algebraic setting (category theory).

There are subtleties concerned with vertex deletion, node merging in L , and other amenities, though we will never use graph rewriting rules/ graph transformations in this sense, hence we will largely ignore them.

STOCHASTIC GRAPH TRANSFORMATION SYSTEMS

A stochastic graph transformation/ rewriting system is a GTS in which **each rule is associated with the rate of an exponential distribution.**

This determines a CTMC in which the states are (coloured/labeled) graphs, while transitions are given by rules being applied in one of the possible ways.



STOCHASTIC GRAPH TRANSFORMATION SYSTEMS

Given a rule $\rho : L \rightarrow R$ with rate λ , and the current state of the CTMC, given by the graph G , what is the **actual rate** of a rule ρ , i.e. the rate at which an observer sees a ρ event happening?

Intuitively, L can be matched in many different ways in G , via injective state-preserving morphisms μ , and each of these matches is a different application of the rule.

Hence, the rate of ρ is proportional to the different number of matches of L in G .

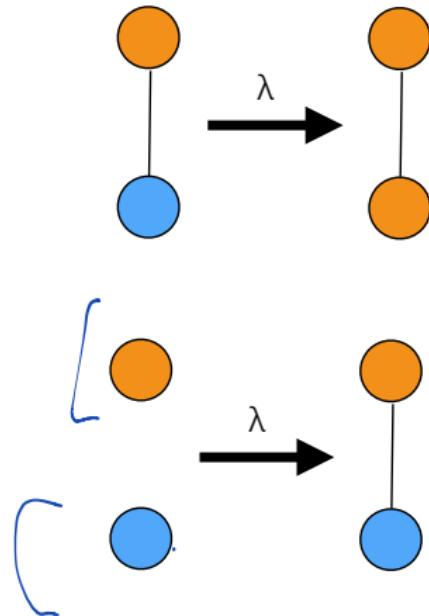
We denote this set as $\overbrace{\text{match}(L, G)}$, and $m(L, G) = |\text{match}(L, G)|$.

It follows that the total rate of ρ in G is $\lambda m(L, G)$.

How to actually count the number of matches? This is the subgraph isomorphism problem, which is NP-complete, but can be tackled efficiently by specialised heuristics.

For simple rules, this is in general easy.

STOCHASTIC GRAPH TRANSFORMATION SYSTEMS



Example: the rule for the mood.

In this case, we just need to count the number of edges with endpoints of different colours. This can be done efficiently.

In this case, we need to count the number of nodes of each colour, and take their product. This is usually the case when L is the union of disconnected components: we can just count them independently (paying attention to overlappings).

This is common for rule based modelling in systems biology, cf. κ -sim

CTMC SIMULATION

CTMC simulation means sampling possible execution traces. The best known algorithm is known as the Doob-Gillespie algorithm.

DOOB-GILLESPIE

It is based on the jump chain/ holding time decomposition. The basic loop, running until the target final time is reached, works at follows:
Let s be the current state.

- Compute all exit transitions from s and add up their rate. Let $\lambda_0 = \sum_j \lambda_j$ be such **exit rate**.
- Choose the next time sampling from an exponential distribution with rate λ_0 (i.e. $\tau = -\frac{1}{\lambda_0} \log U$, with U uniform in $[0, 1]$).
- Choose transition j with probability λ_j/λ_0 .
- Change state accordingly to transition j .

CTMC SIMULATION

In the context of rule-based modelling of GTS, the basic step of the Doob-Gillespie becomes as follows:

- ① Compute for each rule $\rho_j : L_j \rightarrow R_j$ (with rate λ_j) its actual rate, i.e. compute $match(L_j, G)$ and $m(L_j, G)$.
- ② Compute $\lambda_0 = \sum_{j \in \text{rules}} \lambda_j m(L_j, G)$.
- ③ Sample τ w.r.t $Exp(\lambda_0)$.
- ④ Choose a rule with probability $\lambda_j m(L_j, G) / \lambda_0$.
- ⑤ Choose a match of rule j in $match(L_j, G)$ with probability $1/m(L_j, G)$.
- ⑥ Update the system according to the rule/ apply the graph transformation.

OUTLINE

1

INTRODUCTION TO DYNAMIC NETWORK MODELS

- Continuous Time Markov Chains
- Graph Rewriting Rules
- Simulating network models

2

EPIDEMIC SPREADING AND CONTACT PROCESSES

- The homogeneous setting
- The heterogeneous setting

3

PAIR APPROXIMATION

EPIDEMIC SPREADING

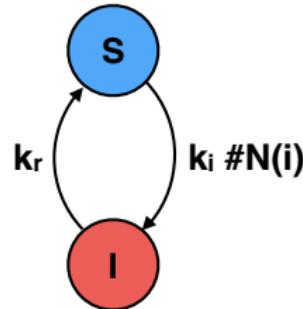


EPIDEMIC MODELLING

- The modelling of epidemic spreading has a centenary tradition, with the first non-linear ODE model dating back to 1920s.
- It is still a very active research area, with deep impact on policy making (e.g. CDC governative agency).
- Modelling is now a crucial ingredient on rapid assessment of the potential impact of policies to be undertaken in case of pandemics, including forced vaccination/ airport closure/ quarantene.
- Large scale, agent-based models are the most used ones in this respect, while simpler models are used to calibrate the more complex ones to reduce computational cost.

SIR-SIS EPIDEMIC MODELS

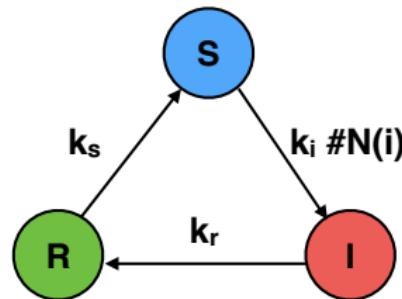
We consider simple agent based models of epidemics: SIS.



- Each agent can be in two states: Susceptible and Infected.
- Infection can be transmitted by contact of a susceptible and an infected individual.
- The more contacts with infected individuals, the more likely to get infected.

SIR-SIS EPIDEMIC MODELS

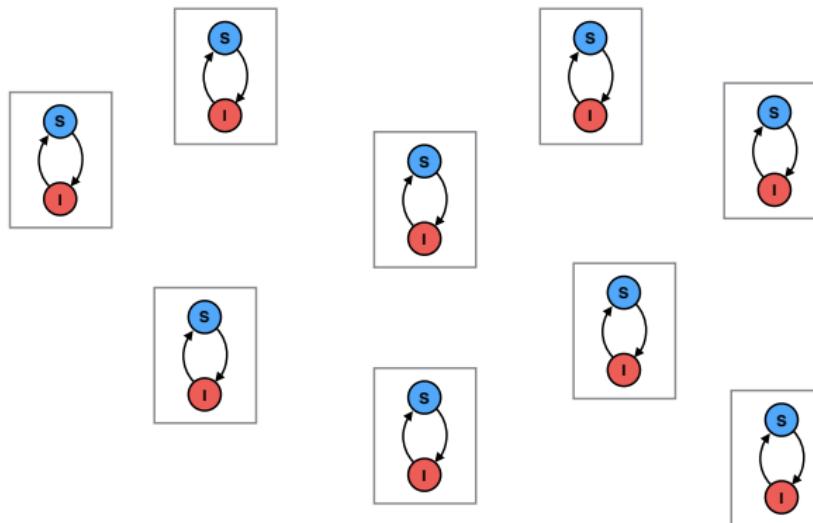
We consider simple agent based models of epidemics: SIR.



- Each agent can be in three states: Susceptible, Infected, and Recovered.
- Infection can be transmitted by contact, like in the SIS model.
- Recovered individuals are immune to infection, but may lose it (or not) after some time.

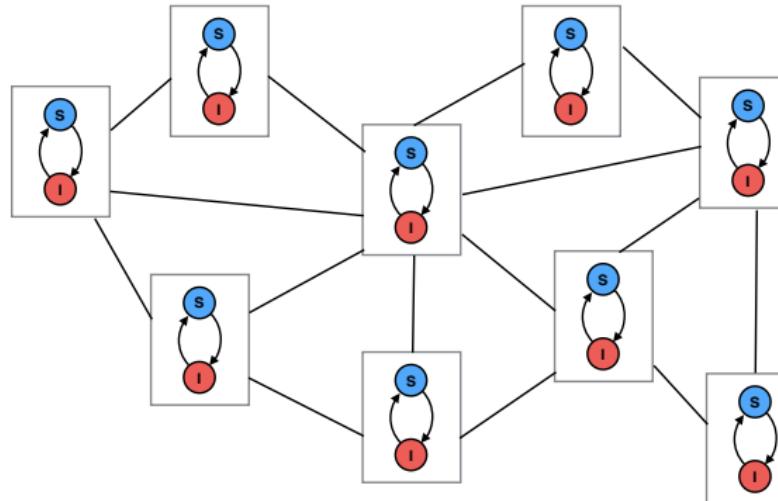
SIR-SIS EPIDEMIC MODELS

In SIS and SIR models, we consider many interacting agents.



SIR-SIS EPIDEMIC MODELS

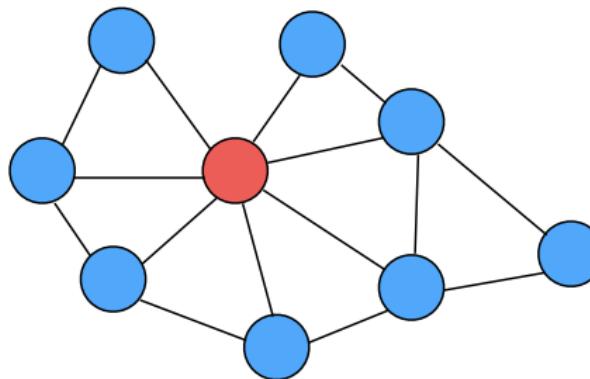
For the model to be more precise, we can represent explicitly the structure of such interactions, **linking two agents by an edge** if they can have a disease-spreading contact.



Hence, we have a **network model**.

SIR-SIS EPIDEMIC MODELS

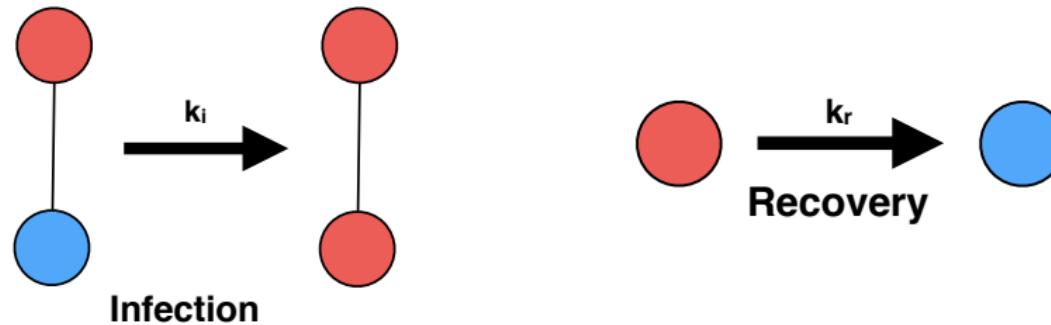
Formally, we can consider a graph (the interaction graph among individuals) in which each node has a state/ colour, taken from the set $\{S, I\}$ or $\{S, I, R\}$.



The dynamics at the network level can then be described by a set of **graph transformation rules**.

SIS GRAPH REWRITING RULES

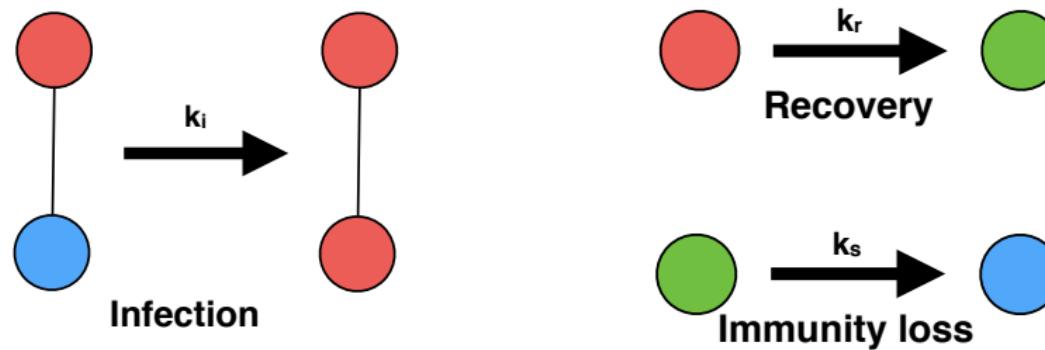
The SIS model has just two simple rules:



The infection rate c_i and the recovery rate c_r play a crucial role in defining the dynamics of the system.

SIS GRAPH REWRITING RULES

The SIR model, instead, has three simple rules:



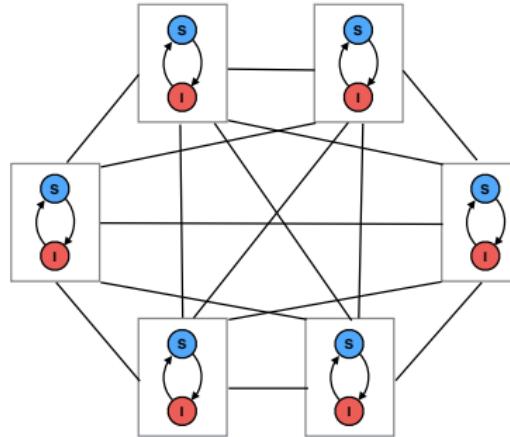
Also in this case, the infection rate c_i and the recovery rate c_r are the most interesting parameters.

SOME INTERESTING QUESTIONS ON SIR/SIS MODELS

- Will the epidemics die out before infecting a (large) fraction of the population?
- How can we assess the risk of pandemics, for a given set of parameters?
- Can we use this model to design vaccination policies?
- **How does the network topology impacts on the dynamics?**
- How can we analyse efficiently network-based epidemic models?
- Can we extend these models to deal with dynamic network topologies?

HOMOGENEOUS MODELS

The most common setting for epidemics models assumes an **homogeneous mixture** of the population, i.e. that the population is well mixed.



In the simplest case, this corresponds to the assumption that the underlying network is a **complete graph**.

COUNTING ABSTRACTION

- As everybody can interact with everybody, the network structure of a complete graph becomes irrelevant for the dynamics.
- Due to the nature of the rules, what really matters is how many individuals in the neighbourhood of a susceptible individual are infected. In the homogeneous case, this is nothing but the total number of infected.
- We can equivalently describe the system by a **counting abstraction**: we only need to know how many agents are in each possible state.
- In the SIR (SIS) model, we only need to introduce three (two) counting variables: X_S , X_I , X_R (X_S , X_I).
- In this way, we obtain a **classical population model**.

POPULATION MODELS

A (Markov) population model (MPM) is a tuple

$\mathcal{X} = (\mathbf{X}, \mathcal{D}, \mathcal{T}, \mathbf{x}_0)$, where:

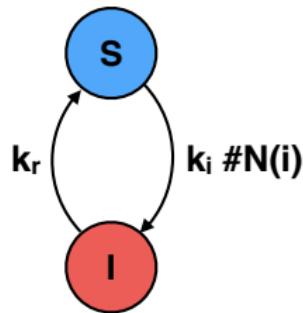
- ① \mathbf{X} is a vector of **variables** counting how many individuals are in each state.
- ② $\mathcal{D} = \prod_i \mathcal{D}_i \subset \mathbb{N}$ is the (countable) **state space**.
- ③ $\mathbf{x}_0 \in \mathcal{D}$ is the **initial state**.
- ④ $\eta_i \in \mathcal{T}$ are the **global transitions/ rules**, $\eta_i = (a, \mathbf{v}, r(\mathbf{X}))$
 - ① a — **event name** (optional).
 - ② $\mathbf{v} \in \mathbb{R}^n$ — **update vector** (from \mathbf{X} to $\mathbf{X} + \mathbf{v}$).
 - ③ $r : \mathcal{D} \rightarrow \mathbb{R}_{\geq 0}$ — **rate**, function of current population.

FROM MPM TO CTMC

The state space is \mathcal{D} . The infinitesimal generator Q is

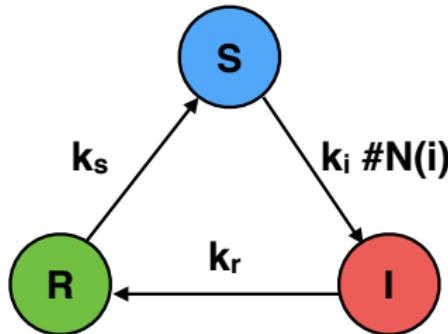
$$Q_{\mathbf{x}, \mathbf{y}} = \sum_{\eta \mid \mathbf{x} + \mathbf{v}_\eta = \mathbf{y}} r_\eta(\mathbf{x}), \quad \mathbf{x}, \mathbf{y} \in \mathcal{D}$$

POPULATION MODELS FOR SIS



- Variables:
 $X_S \in [0, N]$, $X_I \in [0, N]$,
 $X_I + X_S = N$.
- Infection transition:
 $(inf, (-1, 1), c_i X_S X_I)$
- Recovery transition:
 $(rec, (1, -1), c_r X_I)$.

POPULATION MODELS FOR SIR



- Variables:
 $X_S \in [0, N]$, $X_I \in [0, N]$,
 $X_R \in [0, N]$,
 $X_I + X_S + X_R = N$.
- Infection transition:
 $(inf, (-1, 1, 0), c_i X_S X_I)$
- Recovery transition:
 $(rec, (0, -1, 1), c_r X_I)$.
- Immunity loss transition:
 $(loss, (1, 0, -1), c_s X_R)$.

MASTER EQUATION

The Kolmogorov equation in the context of Population Processes is often known as **master equation**.

There is one equation per state $\mathbf{x} \in \mathcal{D}$, for the probability mass $P(\mathbf{x}, t)$, which considers the inflow and outflow of probability at time t .

$$\frac{dP(\mathbf{x}, t)}{dt} = \sum_{\eta \in \mathcal{T}} r_\eta(\mathbf{x} - \mathbf{v}_\eta) P(\mathbf{x} - \mathbf{v}_\eta, t) - \sum_{\eta \in \mathcal{T}} r_\eta(\mathbf{x}) P(\mathbf{x}, t)$$

However, the state space of population models is usually very large, making it impossible to solve the master equation directly. This is the (in)famous **state space explosion** problem.

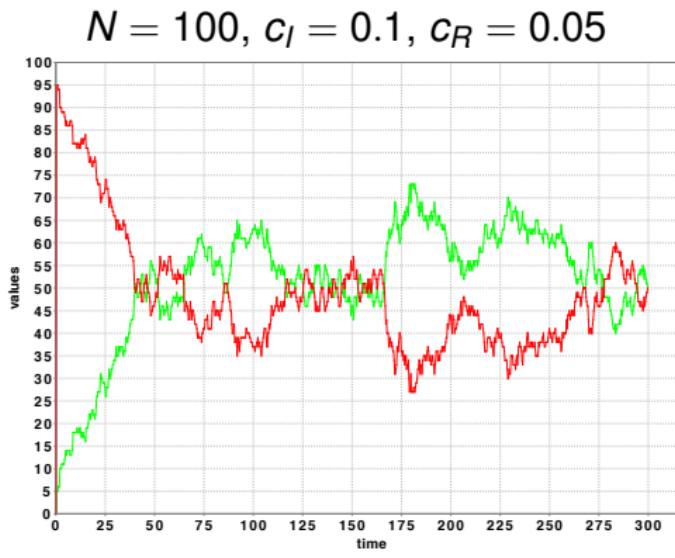
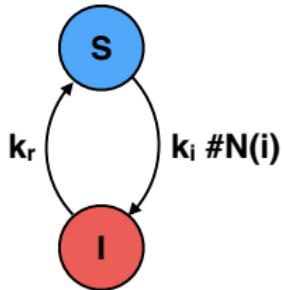
SIMULATION: GILLESPIE'S ALGORITHM REVISITED

The jump chain/ holding times based algorithm for simulating a population model $\mathcal{X} = (\mathbf{X}, \mathcal{D}, \mathcal{T}, \mathbf{x}_0)$ is the standard form of the Gillespie algorithm one can find in literature

THE MAIN LOOP FROM A STATE \mathbf{x}

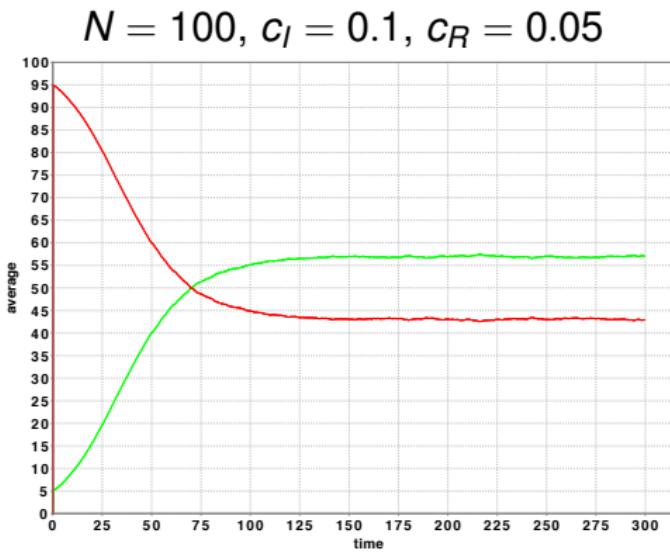
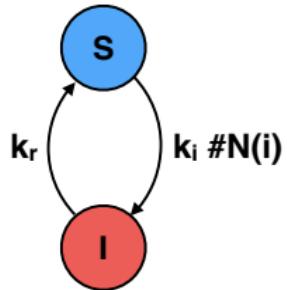
- ① Compute $r_\eta(\mathbf{x})$ for $\eta \in \mathcal{T}$ and $r_0 = \sum_\eta r_\eta(\mathbf{x})$
- ② Compute time step as $\tau = -1/r_0 \log(U)$
- ③ Choose next reaction η with probability $r_\eta(\mathbf{x})/r_0$
- ④ Update state from \mathbf{x} to $\mathbf{x} + \mathbf{v}_\eta$.

EXAMPLE: SIS MODEL



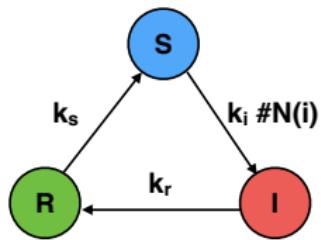
(1 run)

EXAMPLE: SIS MODEL

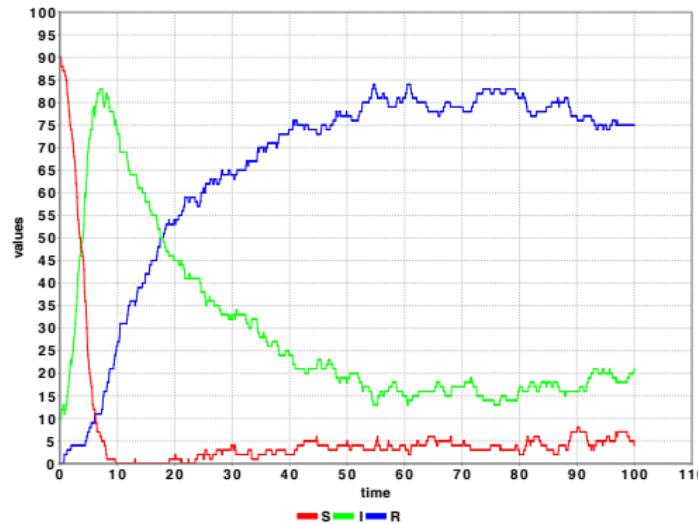


(average)

EXAMPLE: SIR MODEL

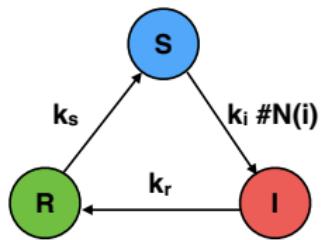


$$N = 100, c_I = 1, c_R = 0.05, c_S = 0.01$$

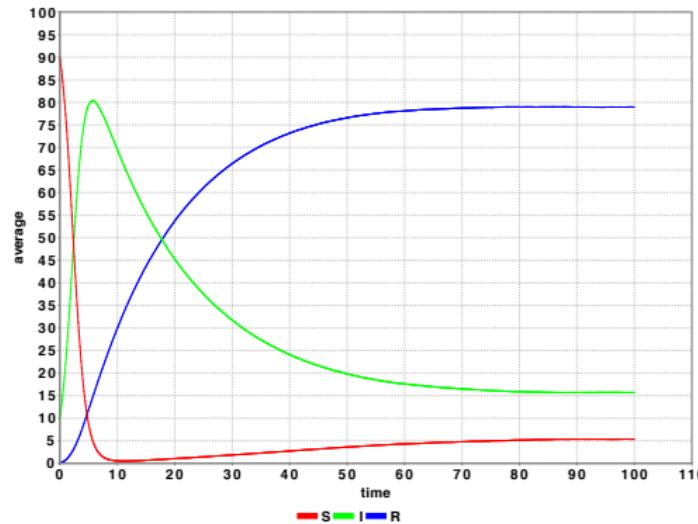


(1 run)

EXAMPLE: SIR MODEL



$$N = 100, c_I = 1, c_R = 0.05, c_S = 0.01$$



(average)

MEAN FIELD EQUATIONS

From the master equation, one can derive easily the following ordinary differential equation for the average of $\mathbb{E}[\mathbf{X}]$:

$$\frac{d}{dt}\mathbb{E}[\mathbf{X}](t) = \sum_{\eta} \mathbf{v}_{\eta} \mathbb{E}[r_{\eta}(\mathbf{X}(t))]$$

The mean-field assumption, in this setting, is to assume that

$$\mathbb{E}[r_{\eta}(\mathbf{X}(t))] \approx r_{\eta}(\mathbb{E}[\mathbf{X}](t))$$

This is exact only if r_{η} is linear, but not otherwise.

For the infection rate in the SIS/ SIR model $r_{inf}(\mathbf{X}) = c_i X_S X_I$:

$$\mathbb{E}[c_i X_S X_I] \approx c_i \mathbb{E}[X_S] \mathbb{E}[X_I]$$

amount to assuming that variables X_S and X_I are uncorrelated.
This is the same as assuming all agents to be independent!

DEEPER INTO MEAN FIELD EQUATIONS

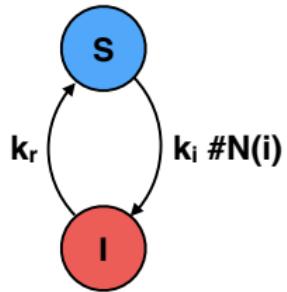
MOMENT CLOSURE

One way to see the mean-field equations is as a set of **approximate equations for the average**, obtained by “closing” the true equations limiting their dependence on moments of order one. A similar idea leads to approximate equations for higher order moments.

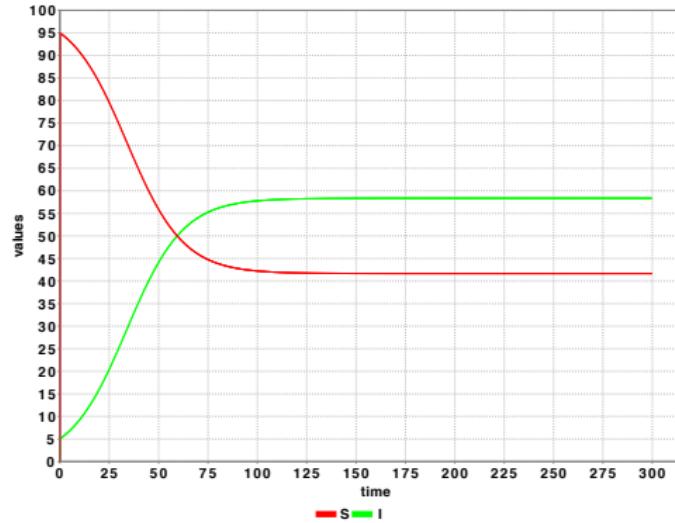
STOCHASTIC APPROXIMATION

An alternative point of view is that of stochastic approximation: one can prove that mean-field equations are **exact** in the limit of the population size going to infinity, by rescaling the process in terms of population densities and proving a **convergence** result (Kurtz theorem).

EXAMPLE: SIS MODEL

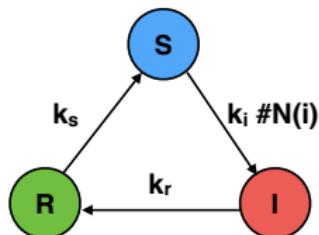


$$N = 100, c_I = 0.1, c_R = 0.05$$

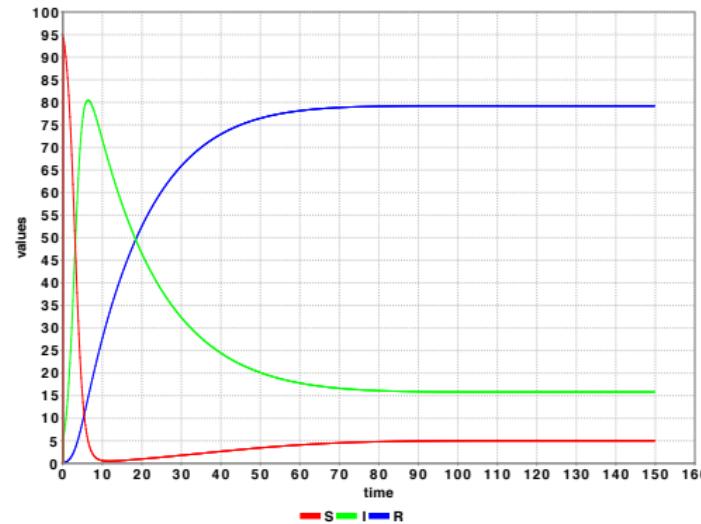


$$\begin{cases} \frac{d}{dt}X_S = -c_I X_S X_I + c_R X_I \\ \frac{d}{dt}X_I = c_I X_S X_I - c_R X_I \end{cases}$$

EXAMPLE: SIR MODEL



$$N = 100, c_I = 1, c_R = 0.05, c_S = 0.01$$



$$\left\{ \begin{array}{l} \frac{d}{dt}X_S = -c_I X_S X_I + c_S X_R \\ \frac{d}{dt}X_I = c_I X_S X_I - c_R X_I \\ \frac{d}{dt}X_R = c_R X_I - c_S X_R \end{array} \right.$$

THE SIS MODEL UNVEILED

- Consider the SIS model, for which $X_S + X_I = N$. We can normalise variables (divide by N , obtaining x_I and x_S) and get rid of x_S by setting $x_S = 1 - x_I$.
- The mean field ODE for x_I then becomes
$$\dot{x}_I = c_i x_I (1 - x_I) - k_r x_I.$$
 ←
- Looking for its steady state, setting $\dot{x}_I = 0$, we get
$$F(x_I) = x_I(R_0 - 1 - R_0 x_I) = 0,$$
 where $R_0 = \frac{c_i}{k_r}$ is known as the **basic reproductive number**.
- The two solutions are $x_I = 0$ and $x_I = 1 - 1/R_0$. The second solution is feasible (as $x_I \in [0, 1]$) only if $R_0 > 1$.
- Computing the derivative $\frac{\partial F(x_I)}{\partial x_I}$, we can show that $x_I = 0$ is stable if and only if $R_0 < 1$, while for $R_0 > 1$, $x_I = 1 - 1/R_0$ is stable.
- Hence, for $R_0 > 1$, there is an **endemic equilibrium**, while for $R_0 < 1$ the epidemic is **extinguished** quickly.

NETWORKS WITH CONSTANT DEGREES

We consider now a network of interactions in which each node has a constant degree k , e.g. a regular lattice.

In this setting, the neighbours of two random nodes are generally different, hence the counting abstraction is not exact anymore: we need to know how many neighbours of a given node are infected!

We can rely on a **mean-field assumption**, assuming that **nodes' states are independent**. Then the probability that a given neighbour is infected is $x_I = X_I/N$, and the fraction of infected neighbours is roughly kx_I .

NETWORKS WITH CONSTANT DEGREES

Under a mean-field hypothesis, we can still use the population-based counting abstraction, as an approximation:

- Let $c_i dt$ be the probability of infection via an infected neighbour in time dt .
- We can approximate the probability that a susceptible node is not infected in dt by $(1 - c_i dt)^{kx_I}$.
- The infection probability is $1 - (1 - c_i dt)^{kx_I} \approx c_i k x_I dt$.
- The mean field equation (SIS model, only infected) is therefore

$$\frac{d}{dt} x_I = c_i k x_I (1 - x_I) - c_r x_I$$

Hence in the mean field model in this case the rate is similar, but with a correction factor given by k to the rate c_i . The basic reproductive number is then $R_0 = \frac{c_i k}{c_r}$, where c_r is the recovery rate.

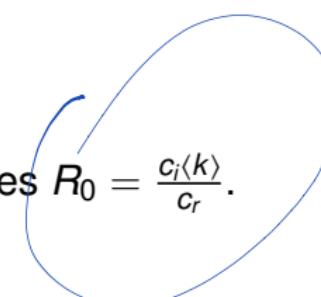
The condition $R_0 > 1$ remains the same.

EPIDEMICS ON A ERDOS-RENYI RANDOM GRAPH

- In an Erdos-Renyi random graph, the degree distribution of nodes is Poissonian, hence no more constant.
- However, for large graph size N , we can couple the population-based counting abstraction and the mean field assumption with the approximation $k \approx \langle k \rangle$.
- This gives, for the SIS model, a set of equations similar to the one above:

$$\frac{d}{dt}x_I = c_i \langle k \rangle x_I(1 - x_I) - c_r x_I$$

- Then the basic reproductive number becomes $R_0 = \frac{c_i \langle k \rangle}{c_r}$.



HETEROGENEOUS MODELS

Consider now an epidemic SIS or SIR model in a generic graph, possibly with an heterogeneous degree distribution. Will its behaviour be different than in the homogeneous case?

Intuitively, in a model with a skewed degree distribution, there will be several hubs, linked with many nodes. If these individuals are infected, they can act as super-spreaders. How can this impact on the dynamics?

In the following, we will set up a mean-field abstraction which works for general graph models, showing how the condition for having an **endemic epidemics** is changed.

DEGREE-BASED COUNTING ABSTRACTION

- In the general case, we rely on a degree-based counting abstraction.
- We denote by $X_{j,k}$ the number of nodes of degree k in state j , and by

$$x_{j,k} = \frac{X_{j,k}}{N_k} \quad \sum_j x_{j,k} = 1$$

the fraction of nodes of degree k that are in state j .

- It follows that the fraction of nodes in state j is $\sum_k x_{j,k}$

$$x_j = \sum_k P(k)x_{j,k}$$

- In the SIS model, we have the fraction of infected nodes of degree k , $x_{I,k}$, and the fraction of susceptible nodes of degree k , $x_{S,k}$.

DEGREE-BASED COUNTING ABSTRACTION

- We also introduce the quantity $\theta_{j,k}$ which denotes the fraction of neighbours of a degree- k node which are in state j .
- In a correlated network, we can express $\theta_{j,k}$ using the correlation probability $P(k'|k)$ of a node of degree k having a neighbour of degree k' :



$$\theta_{j,k} = \sum_{k'} x_{j,k'} P(k'|k)$$

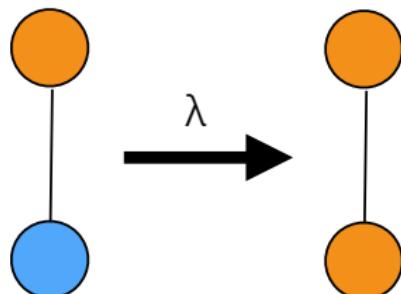
- If the network is uncorrelated, hence $P(k'|k) = \frac{k' P(k')}{\langle k \rangle}$, we have

$$\theta_j = \theta_{j,k} = \left[\sum_{k'} \frac{k' x_{j,k'} P(k')}{\langle k \rangle} \right]$$

independent of k .

DEGREE-BASED MEAN FIELD MODEL

The degree-based mean field assumption assumes independence of nodes but exploits the degree-based counting abstraction.



For a rule involving propagation of state via an edge (like the one on the left, i blue, j orange), the rate for a degree k node in state i of changing state to j can be written as

$$\lambda k x_{i,k} \theta_{j,k}$$

DEGREE-BASED MEAN FIELD MODEL

- For the epidemic spreading, this means that the probability of a node of degree k to be infected depends on the fraction of neighbours of degree k nodes which are infected, i.e. from $\theta_{I,k}$.
- The infection rate for susceptible nodes of degree k then is

$$c_i k x_{S,k} \theta_{I,k}$$

- The recovery rate for infected vertices of degree k is always
- The SIS model then becomes (we have one equation for each k):

$$c_r x_{I,k}$$

$$\frac{d}{dt} x_{I,k} = c_i k (1 - x_{I,k}) \theta_{I,k} - c_r x_{I,k}$$

STEADY STATE FOR THE SIS MODEL

We consider the SIS degree-based mean-field model for an uncorrelated network, so that $\theta_{I,k} = \theta_I$.

The equation is

$$\rightarrow \frac{d}{dt}x_{I,k} = c_i k(1 - x_{I,k})\theta_I - c_r x_{I,k}$$

with

$$\theta_I = \frac{1}{\langle k \rangle} \sum_{k'} k' x_{I,k'} P(k')$$

The steady state solution, solving the k dependent system of equations $\dot{x}_{I,k} = 0$ is

$$\left[x_{I,k} = \frac{k c_i \theta_I}{c_r + c_i k \theta_I} \right] \circlearrowleft$$

resulting in the following self consistency equation for θ_I :

$$\theta_I = \frac{1}{\langle k \rangle} \sum_{k'} k' \frac{k c_i \theta_I}{c_r + c_i k \theta_I} P(k') \quad \circlearrowleft$$

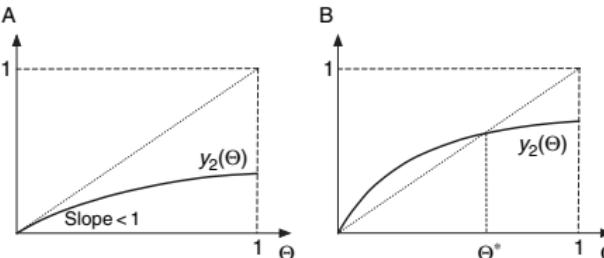
STEADY STATE FOR THE SIS MODEL

Consider the equation

$$\theta_I = \frac{1}{\langle k \rangle} \sum_{k'} k' \frac{k c_i \theta_I}{c_r + c_i k \theta_I} P(k')$$

- Zero is always a solution.
- Using geometrical arguments (on the slope of $\frac{1}{\langle k \rangle} \sum_{k'} k' \frac{k c_i \theta_I}{c_r + c_i k \theta_I} P(k')$ in $\theta_I = 0$),

c_i / c_r and heterogeneity ≥ 1



we obtain that there is a second solution (endemic equilibrium) iff

$$\frac{c_i \langle k^2 \rangle}{c_r \langle k \rangle} \geq 1$$

SIS MODEL ON POWER LAW NETWORKS

the condition for an endemic equilibrium is

$$\frac{c_i}{c_r} \frac{\langle k^2 \rangle}{\langle k \rangle} \geq 1$$

In a power law network, $\langle k^2 \rangle \rightarrow \infty$ for $N \rightarrow \infty$, hence in large network with power law distributions, **there is always an endemic equilibrium**, independently on the relative strength of infection and recovery, i.e. on the basic reproductive number.

This is the effect of super-spreaders!

An interesting example is given by the **spreading of viruses on internet**: even stupid viruses tend to persist, independently on the efforts to eradicate them.

VACCINATION

- Vaccination makes susceptible individuals **immune**.
- **Random vaccination** can be modeled by removing a fraction g of nodes at random from the network!
- Mathematically, we can capture this effect by **replacing the number of edges k with $(1 - g)k$** in formulas above.
- Hence the condition for emergence of an endemic state in an uncorrelated graph becomes

$$(1 - g)R_0 = (1 - g) \frac{c_i \langle k^2 \rangle}{c_r \langle k \rangle} \geq 1$$

- We see that we need to **vaccinate a fraction $g = 1 - \frac{1}{R_0}$** to avoid an epidemic.
- In a large power law network, then $g \approx 1$: **random vaccinations are ineffective**.
- **Targeted vaccinations** (i.e. vaccinating hubs), instead, are very effective.

OUTLINE

1

INTRODUCTION TO DYNAMIC NETWORK MODELS

- Continuous Time Markov Chains
- Graph Rewriting Rules
- Simulating network models

2

EPIDEMIC SPREADING AND CONTACT PROCESSES

- The homogeneous setting
- The heterogeneous setting

3

PAIR APPROXIMATION

MOTIVATION

- Consider the (SIS/ SIR) epidemic model. The global infection rate is proportional to the number of edges between susceptible and infected nodes.
- In the mean field equation, we approximated this number either by the product of X_S and X_I , or by a degree based expression $X_{S,k}k\theta_k$.
- An alternative would be to explicitly count the number of pairs of connected S and I nodes!
- This can be achieved introducing new counting variables for edges. In the SIS model, these are X_{SS} , $X_{SI} = X_{IS}$, X_{II} . But then we need to understand how the application of rules changes such counts.

COUNTING PAIRS

- We need to understand how the number of pairs of nodes (connected by an edge) **changes** when applying the graph transformation rules.
- It turns out that the rate of change will **depend on the number of instances of larger subgraphs**, in particular connected triplets of nodes.
- The pair approximation then consists in **approximating the number of triplets as a function of the number of pairs and vertices** in a given state.
- Pair approximation is thus a sort of **graph (moment) closure**, leading to an approximate set of differential equation for the average of pair counts.
- Similar ideas can be applied to count and approximate more complex subgraphs (e.g. triples).

PLAN

In the following, we will dig into pair approximation in steps:

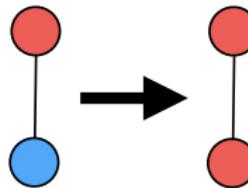
- ① Set up a way to express the speed of change of a given subgraph as a function of other subgraphs.
We will concentrate on pairs in the example, but sketch the method in general.
- ② Introduce the counting approximation for triples.
- ③ Show how this leads to the pair approximation equations.

BASICS

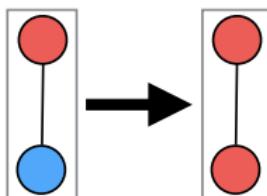
- Input: a graph transformation system with rules $\rho_j : L_j \rightarrow R_j$, and the current state, given by a graph G .
- Input: a graph pattern $P = (V_P, E_P, \sigma_P)$ with states, of which we want to understand how its number of instances in a graph G changes by application of the graph transformation rules ρ_j .
- Output: Two lists \mathcal{L}_L and \mathcal{L}_R of tuples of the form (n, H, c) , with $n \in \mathbb{N}$ the multiplicity, H a graph, and $c \in \mathbb{R}$ a rate.
- The first list \mathcal{L}_L describes the consumption rate of P , while the second list \mathcal{L}_R describes its production rate.

INTUITION

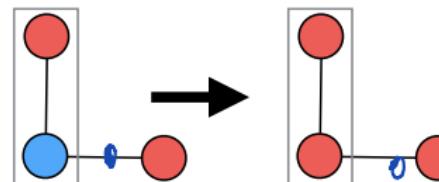
Consider the SIS model, the infection rule and the pattern P :



The pattern P can change by:



A direct application of the rule



An indirect application (the susceptible is infected by another neighbour).

GLUEING

- Hence, we need to find all possible ways in which a rule can modify a pattern P .
- This is a local operation: it depends only on the pattern P and on the left (L) or right-hand-side (R) of the rule.
- We need to combine/ glue together P and L (or R) to construct a local graph on which the application of the rule depends on.

MINIMAL GLUEING OF P AND L

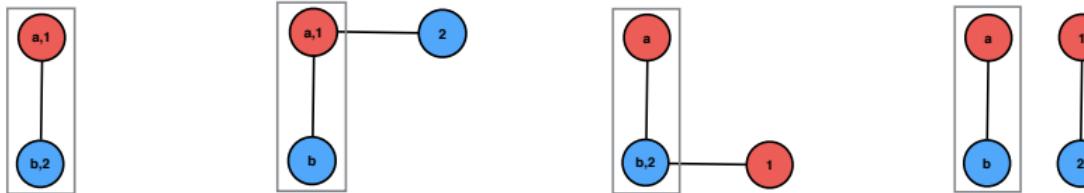
It is a graph H such that there are two injective morphisms $g_P : P \rightarrow H$ and $g_L : L \rightarrow H$ such that $H = g_P(P) \cup g_L(L)$, i.e. $g_P + g_L : P + L \rightarrow H$ is surjective.

EXAMPLE: GLUEING FOR SIS MODEL

Consider the SIS model, the infection rule and the pattern P :



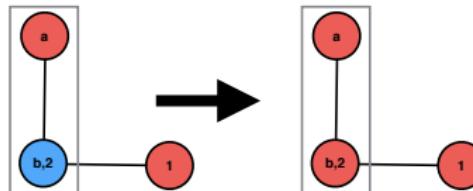
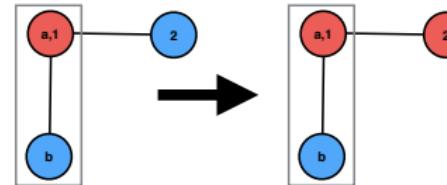
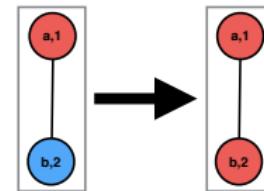
We have four possible minimal glueings between P and L :



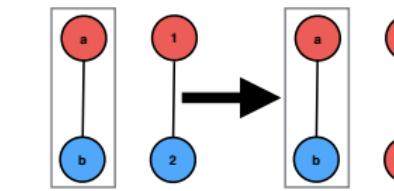
Warning: numbering nodes is essential in order to keep symmetries into account.

EVOLVING A GLUEING

Now we apply the rule to each glueing and check if and how the pattern has changed.

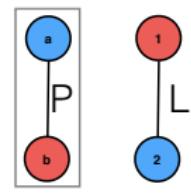
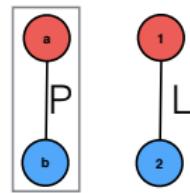


P decreases



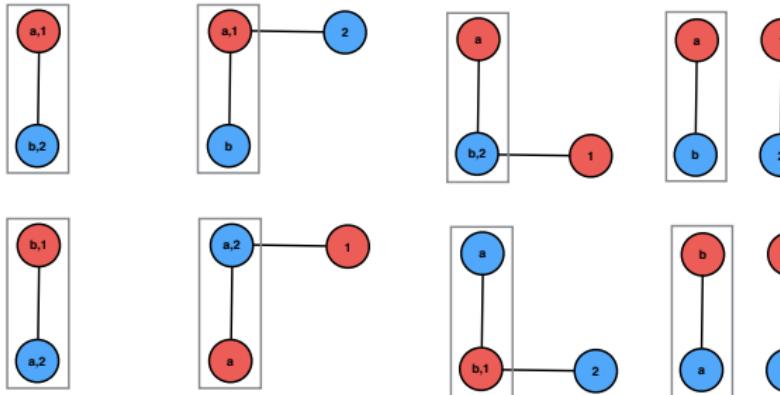
P unchanged

PATTERNS AND SYMMETRIES



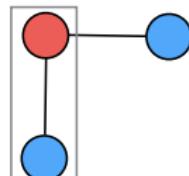
The patterns *IS* and *SI* are isomorphic.

Also the minimal glueings will be isomorphic, hence they will show the same behaviour. We can consider only one pattern.

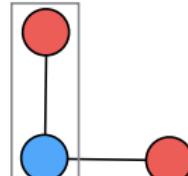


CONSTRUCTING THE CONSUMPTION LIST

- To construct the consumption list \mathcal{L}_L , we need to glue together P and L_j (for each rule j), and check for which glueings the application of ρ_j modifies the pattern P .
- Each of these glueings will be an instance in the consumption list: if P is modified, then one of its copies is lost.
- As counting is agnostic to pattern and rule numberings, we will **discard them**, and **identify isomorphic minimal glueings**, summing their multiplicities.
- Warning: new notation for pairs/ triples/ nodes counts!



[S/S]

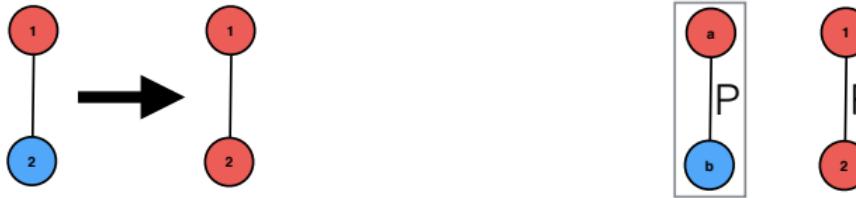


[IS/I]

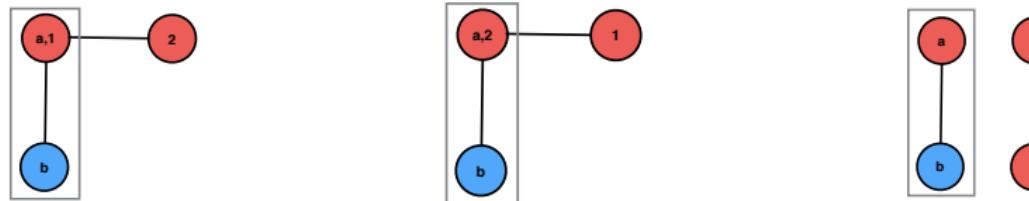
Consumption List \mathcal{L}_L
[(1, [S], c_i), (1, [IS], c_i)]

GLUEING P AND R

Consider the SIS model, the infection rule and the pattern P :

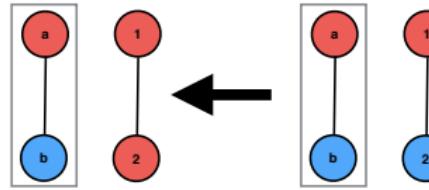
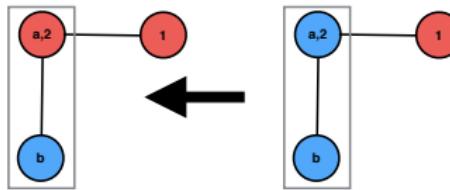


We have three possible minimal glueings between P and R :

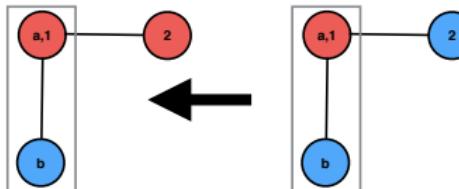


EVOLVING A GLUEING WITH R

In order to evolve a glueing with the r.h.s. of a rule, we have to apply the rule in the reverse order, and then check if the pattern is changed.



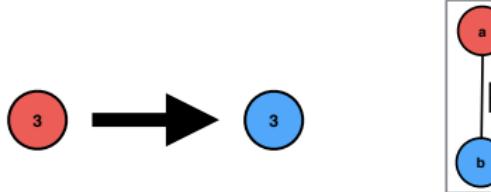
P changes only in the first case.



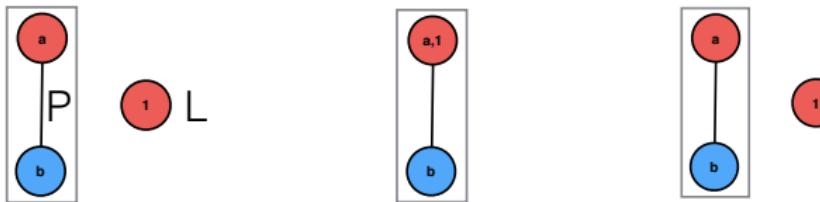
Notice that the two glueings on the left are isomorphic when numberings are removed, but they are not isomorphic w.r.t. rule applications (numbering has to be preserved).

EXAMPLE: SIS AND THE RECOVERY RULE

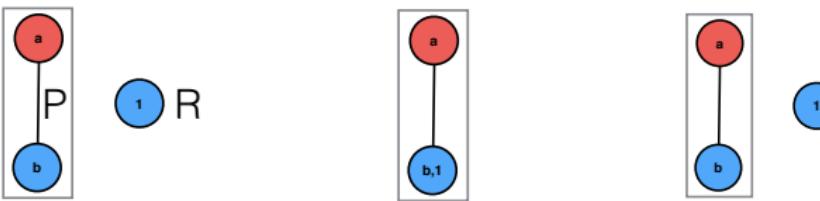
Consider the SIS model, the recovery rule, and the pattern P



Glueings with L

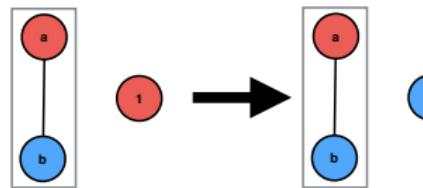
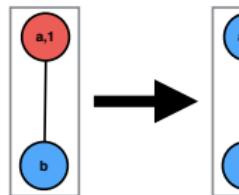


Glueings with R

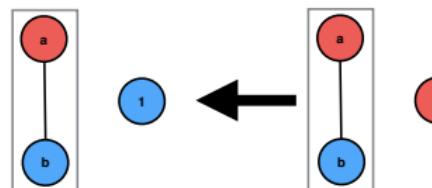
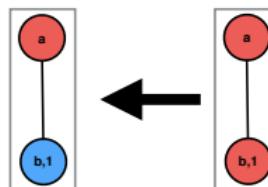


EVOLUTION OF GLUEINGS FOR RECOVERY RULE

Glueings with L



Glueings with R



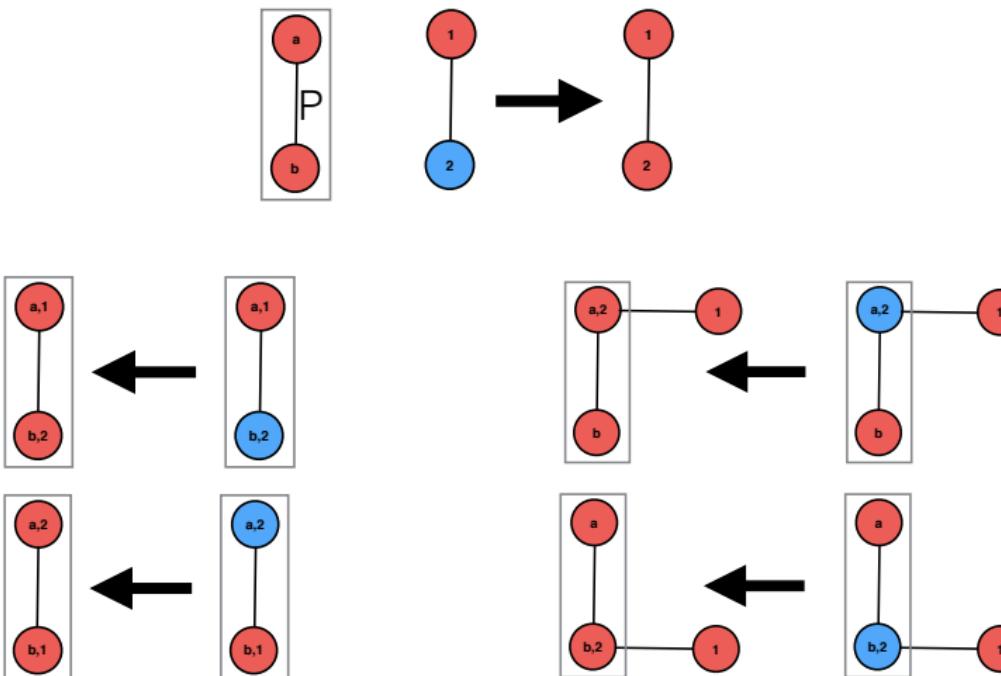
P changed

$$\mathcal{L}_L = [(1, [SI], c_r)]$$

P unchanged

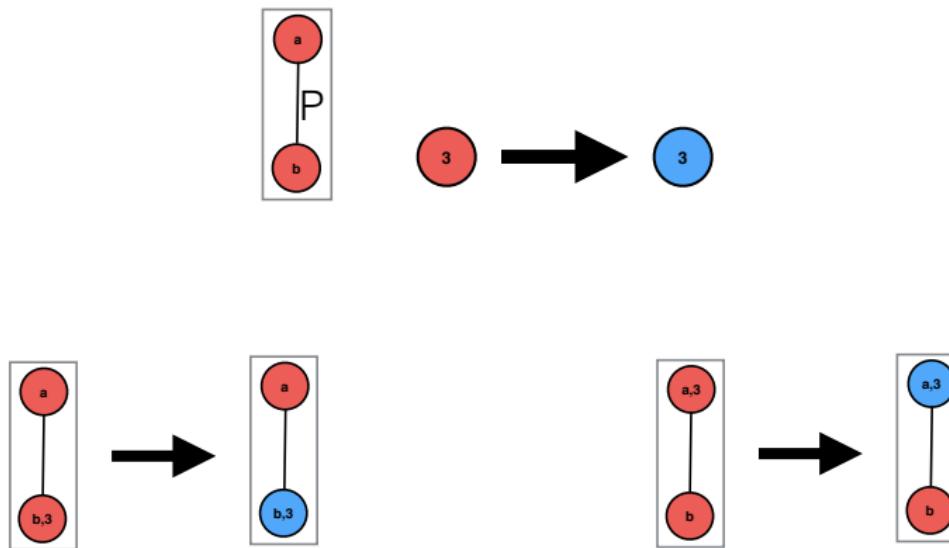
$$\mathcal{L}_R = [(1, [II], c_r)]$$

GLUEING FOR OTHER PATTERNS IN THE SIS MODEL



There is another way to glue P with R , but it is **irrelevant** (i.e. does not change P when the rule is reversed).

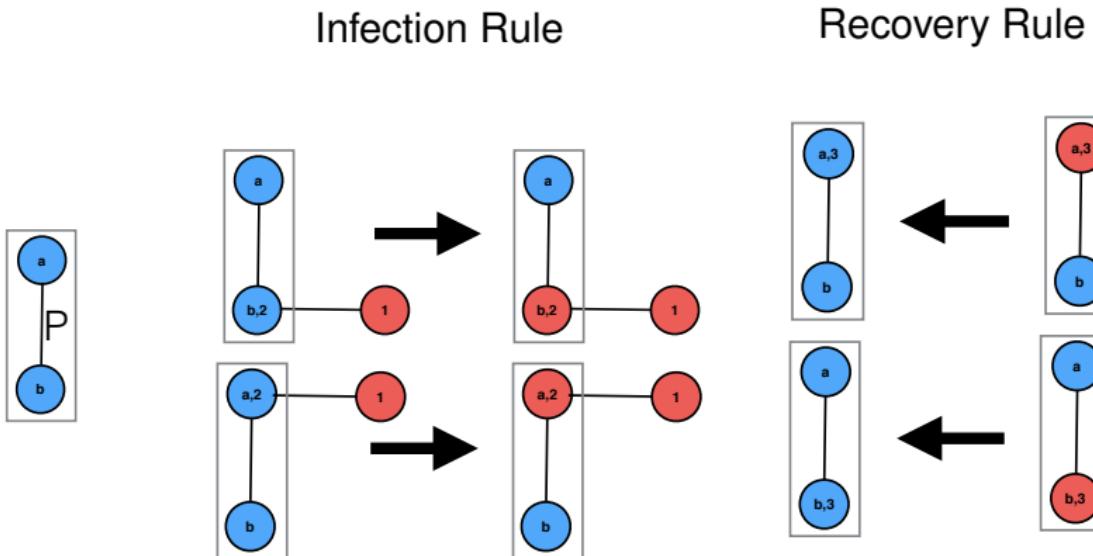
GLUEING FOR OTHER PATTERNS IN THE SIS MODEL



$$\mathcal{L}_L = [(2, [I], c_r)]$$

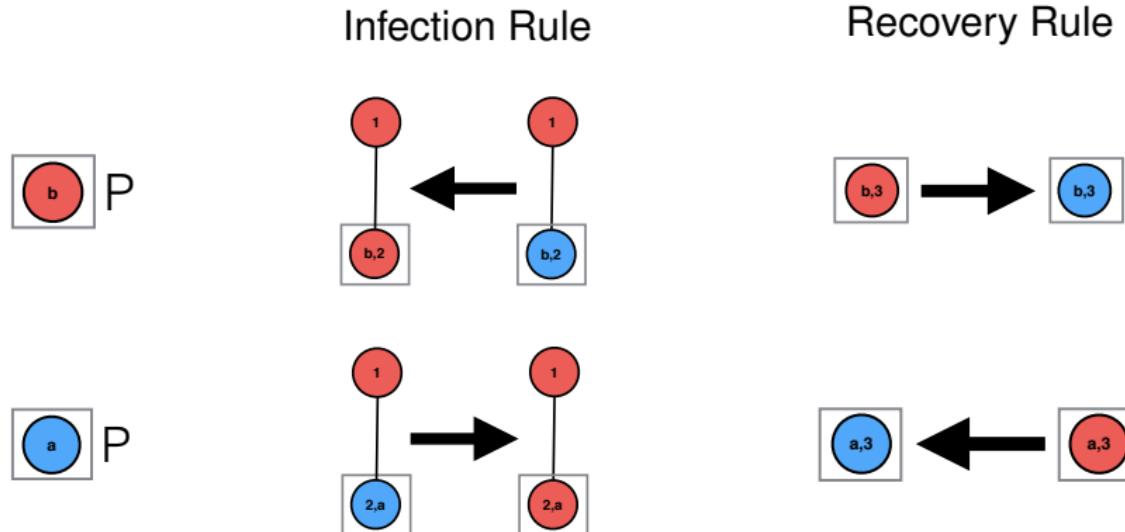
$$\mathcal{L}_R = [(2, [SI], c_i), (2, [ISI], c_i)]$$

GLUEING FOR OTHER PATTERNS IN THE SIS MODEL



$$\mathcal{L}_L = [(2, [SSI], c_i)] \quad \mathcal{L}_R = [(2, [SI], c_r)]$$

GLUEING FOR OTHER PATTERNS IN THE SIS MODEL



LISTS FOR THE SIS MODEL



$$\mathcal{L}_L = [(1, [SI], c_i)]; \mathcal{L}_R = [(1, [I], c_r)]$$



$$\mathcal{L}_L = [(1, [I], c_r)]; \mathcal{L}_R = [(1, [SI], c_i)]$$



$$\begin{aligned}\mathcal{L}_L &= [(1, [SI], c_i), (1, [ISI], c_i), (1, [SI], c_r)]; \\ \mathcal{L}_R &= [(1, [SSI], c_i), (1, [II], c_r)]\end{aligned}$$



$$\mathcal{L}_L = [(2, [SSI], c_i)]; \mathcal{L}_R = [(2, [SI], c_r)]$$



$$\mathcal{L}_L = [(2, [II], c_r)]; \mathcal{L}_R = [(2, [SI], c_i), (2, [ISI], c_i)]$$

RECAP: HOW TO GET THE LISTS

For each pattern P and each rule $L \xrightarrow{c} R$:

- ➊ Combine the pattern with L and generate numbered minimal glueings H_L^* in all possible (non-isomorphic) ways. For each H_L^* :
 - ➌ Apply the rule to H_L^* , $H_L^* \rightarrow J_L^*$, and check if the image of the pattern P in H_L^* is modified in J_L^* .
 - ➋ If so, add $(1, H_L, c)$ to consumption list, where H_L is the non-numbered version of H_L^* .
- ➋ Combine the pattern P with R to generate all possible numbered minimal glueings J_R^* . For each J_R^* :
 - ➌ Apply the rule inverse to J_R^* , obtaining H_R^* , and check if the image of the pattern P in J_R^* is modified in H_R^* .
 - ➋ If so, add $(1, H_R, c)$ to the production list, where H_R is the non-numbered graph obtained after applying the rule inverse.

For each list and each pair (k_1, H_1, c_1) and (k_2, H_2, c_2) , if $c_1 = c_2$, and H_1 is isomorphic with H_2 , then replace them with $(k_1 + k_2, H_1, c_1)$.

FROM LISTS TO AVERAGE EQUATIONS

From consumption and production lists of a given pattern P , we can easily construct ODEs for its average evolution:

$$\frac{d}{dt}[P] = \sum_{(h,H,c) \in \mathcal{L}_R} h \cdot c \cdot [H] - \sum_{(h,H,c) \in \mathcal{L}_L} h \cdot c \cdot [H]$$

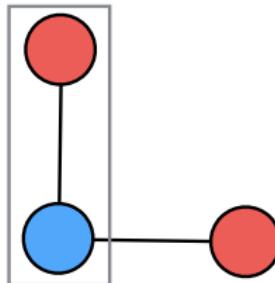
$$\left\{ \begin{array}{l} d[S]/dt = -c_i[SI] + c_r[I] \\ d[I]/dt = c_i[SI] - c_r[I] \\ d[SI]/dt = -c_i[SI] - c_i[ISI] - c_r[SI] + c_i[SSI] + c_r[II] \\ d[SS]/dt = -2c_i[SSI] + 2c_r[SI] \\ d[II]/dt = -2c_r[II] + 2c_i[SI] + 2c_i[ISI] \end{array} \right.$$

Notice: $[S] + [I]$ and $[SS] + [II] + 2[SI]$ are conserved quantities.

The problem is that equations for pairs are not closed, but depend on the counts of triples.

PAIR APPROXIMATION

The idea of pair approximation is to express the number of triples as a non-linear function of the number of pairs.



We can approximate the number of triples $[ABC]$ as

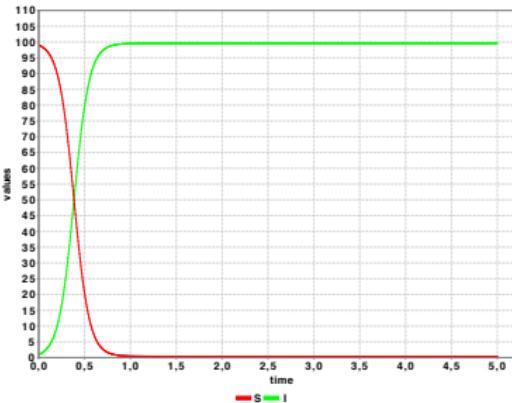
$$[ABC] \approx [AB][BC] \frac{1}{[B]}$$

The previous expression can be read as:

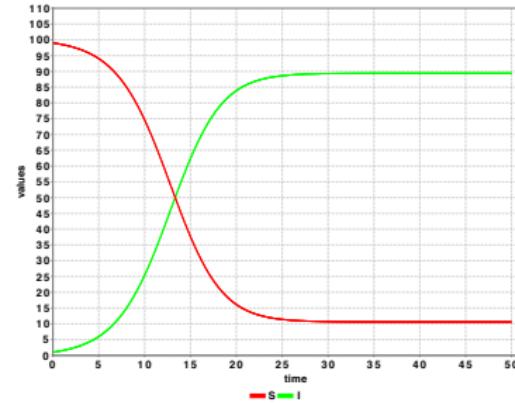
of edges $AB \times$ # of edges $BC \times$ prob. the B vertices coincide

EXAMPLE: THE SIS EPIDEMICS

We can construct the SIS pair approximation, assuming the process happens on a regular 2 dimensional lattice of 100 nodes, with only one vertex infected. Hence, we have $[SI](0) = 4$, $S(0) = 99$, $I(0) = 1$, and let $c_i = 0.12$, $c_r = 0.05$. Compare this with the basic SIS mean field model for the same initial condition and with the same rates.



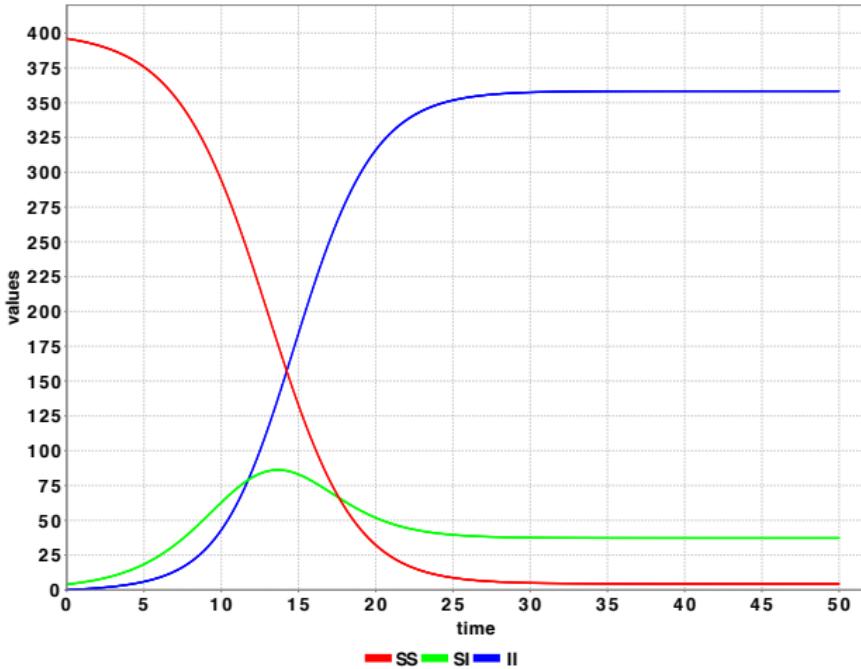
basic mean field model



pair approximation

EXAMPLE: THE SIS EPIDEMICS

We can also see how the number of pairs changes in the SIS model on a lattice!



PAIR APPROXIMATION

A SUSPICIOUS LOOK

Here we are ignoring the degree of the B vertex, when considering the probability of the two B vertices to coincide.

- Assume the central vertex B has degree k , and write it B_k . Then $[AB_k C]$ is given by the number of edges $[AB_k]$ times the probability that one of the remaining $k - 1$ edges out of B_k points to a C node.
- This last quantity can be computed as $B_k C$ edges / total number of edges out of B_k nodes (minus 1), i.e. as $\frac{[B_k C]}{k[B_k] - 1}$
- Thus we get

$$[AB_k C] \approx [AB_k] \frac{(k - 1)[B_k C]}{k[B_k] - 1} \approx \frac{k - 1}{k} \frac{[AB_k][B_k C]}{[B_k]} \approx \frac{[AB_k][B_k C]}{[B_k]}$$

VALIDITY OF THE PAIR APPROXIMATION

- We can always write

$$[ABC] = \sum_k [AB_k C]$$

- Then $[ABC]$ is exactly equal to $[AB][BC]/[B]$ if and only if the degree of B nodes is constant and equal to $k!$
- Hence pair approximation works for very regular graphs, like lattices.
- It also works for graphs whose degree is very close to its average value, like random Erdos-Renyi graphs.
- However, its validity for heterogeneous scale-free graphs is questionable.

DEGREE-BASED PAIR APPROXIMATION

- As in mean field, we can improve the approximation moving to a degree-based counting abstraction, this time also for pairs.
- This means counting the number of edges $[A_hB_k]$ between nodes of degree h in state A and nodes of degree k in state B .
- The evolution of these counts will depend on triples of the form $[A_hB_kC_l]$, and can be obtained by a similar minimal glueing construction, now carefully distinguishing nodes of different degree.
- The formula for pair approximation is similar to the degree based one.

PAIR APPROXIMATION AND CLUSTERING

- The number of triples $[XYZ]$ is the sum of the number of open triples and the number of triangles:

$$[XYZ] = [XYZ]_{open} + [XYZ]_{\Delta}$$

- The fraction of triples that are triangles, $[XYZ]_{\Delta}/[XYZ]$ is the clustering coefficient C of the network!
- Assume that the ratio of $[XYZ]$ triples with Y as middle vertex with triples $[X * Z]$ with arbitrary middle vertex for open or closed triples is the same (i.e., if XY and YZ are connected, then the fact that XZ is connected is independent of Y), so that

$$\frac{[XYZ]_{open}}{[X * Z]_{open}} = \frac{[XYZ]_{\Delta}}{[X * Z]_{\Delta}}$$

- Hence $[XYZ]_{\Delta} = \frac{[X * Z]_{\Delta}}{[X * Z]_{open}} [XYZ]_{open}$

PAIR APPROXIMATION AND CLUSTERING

- The number of open pairs $[X * Z]_{open}$ can be computed by observing that two random X and Z nodes will share a neighbour with probability $\frac{k \cdot k}{N}$ and will be directly connected with probability $\frac{k}{N}$, if the degree is constant and equal to k . Hence they will form an open triple with probability $\frac{k(k-1)}{N}$. As there are $[X][Z]$ pairs of such nodes, we have that

$$[X * Z]_{open} = \frac{k(k-1)}{N} [X][Z].$$

- The ratio of $[X * Z]_\Delta$ versus the total number of triangles is just the probability of X and Z being connected, which is $\frac{[XZ]}{kN}$. As there are $Ck(k-1)N$ triangles (why?), we have

$$[X * Z]_\Delta = C(k-1)[XZ]$$

PAIR APPROXIMATION AND CLUSTERING

Hence, we have the following correction due to the clustering coefficient:

$$[XYZ] \approx [XYZ_{open}]((1 - C) + C \frac{N}{k} \frac{[XZ]}{[X][Z]})$$

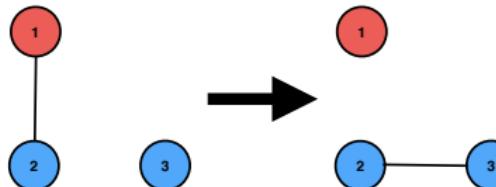
i.e.

$$[XYZ] \approx \frac{k-1}{k} \frac{[XY][YZ]}{[Y]} \left((1 - C) + C \frac{N}{k} \frac{[XZ]}{[X][Z]} \right)$$

ANOTHER EXAMPLE: ADAPTIVE NETWORK MODELS

Assuming a fixed topology of social interaction during an epidemics may not be realistic. People can tend to cut connections with infected people to avoid contagion.

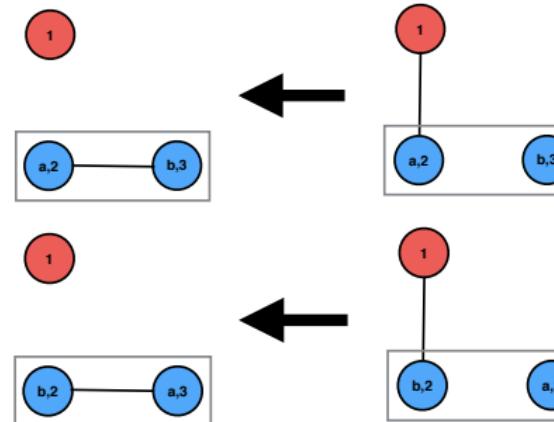
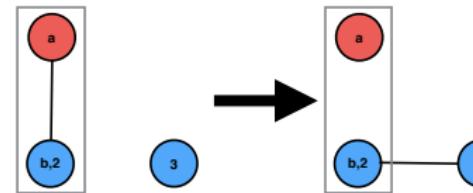
We can include this into the SIS model by adding another rule, firing with rate c_w



This rule changes the topology of the network, while the SIS rules change the state of nodes. Dynamic models with both such features are known as **adaptive or co-evolving network models**.

PAIR APPROXIMATION OF THE ADAPTIVE MODEL

We just need to include in the pair approximation lists the following terms for the specified patterns.



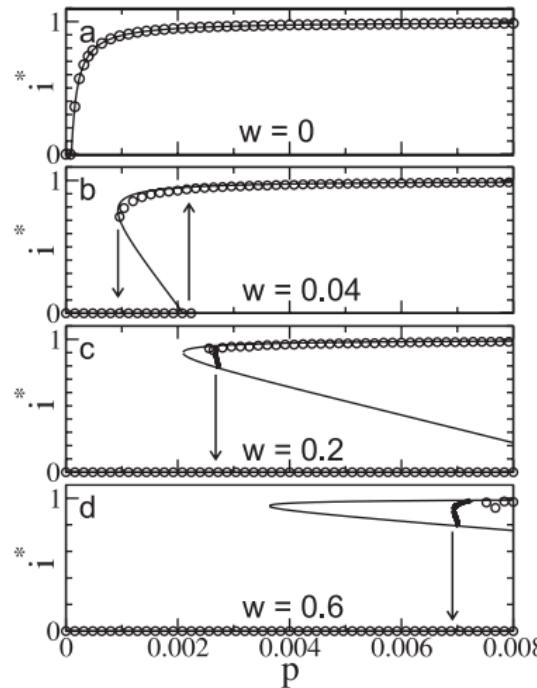
SIS ADAPTIVE MODEL

We assume the rewiring speed to depend only on the number of $[SI]$ pairs, not on the number of available S nodes to connect with, which are chosen at random.

Hence, we get the following equations.

$$\begin{cases} d[S]/dt = -c_i[SI] + c_r[I] \\ d[I]/dt = c_i[SI] - c_r[I] \\ d[SI]/dt = -c_w[SI] - c_i[SI] - c_i[ISI] - c_r[SI] + c_i[SSI] + c_r[II] \\ d[SS]/dt = -2c_i[SSI] + 2c_w[SI] + 2c_r[SI] \\ d[II]/dt = -2c_r[II] + 2c_i[SI] + 2c_i[ISI] \end{cases}$$

SIS ADAPTIVE MODEL



- This adaptive model shows a **bistable** behaviour.
 - For some values of the rewiring rate c_w , both the **endemic equilibrium** and the **epidemic-free** **equilibrium** are stable.
 - For large values of c_w , only the **epidemic-free** **equilibrium** survives.
- Gross et al. Epidemic Dynamics on an Adaptive Network, Phys. Rev. Lett. 96, 2006.