THESIS TITLE: Integrating Markov process and structural causal models enables counterfactual inference in complex systems

AUTHOR: Kaushal Pamesri

M.S. Thesis Approved as an elective towards the Master of Science Degree in Data Science.

Thesis Advisor

Date 08/07/2019

Thesis Reader

Date 08/07/2019

Thesis Reader

Date 08/15/2019

Thesis Reader

Date

Thesis Reader

Date

GRADUATE SCHOOL APPROVAL:

Director, Graduate School

Date 08/15/19

COPY RECEIVED IN GRADUATE SCHOOL OFFICE:

Recipient's Signature

Date

Distribution: Once completed, this form should be scanned and attached to the front of the electronic dissertation document (page 1). An electronic version of the document can then be uploaded to the Northeastern University-UMI website.
Integrating Markov Process And Structural Causal Models Enables Counterfactual Inference In Complex Systems

Author: Kaushal PANERI
Supervisor: Dr. Olga VITEK

A thesis submitted in fulfillment of the requirements for the degree of MS with Thesis in the

OLGA VITEK LAB - Statistical Methods For Studies of Biomolecular Systems
KHOURY College of Computer Sciences

August 23, 2019
Abstract

Dr. Olga Vitek
Khoury College of Computer Sciences

MS with Thesis

Integrating Markov Process And Structural Causal Models Enables Counterfactual Inference In Complex Systems

by Kaushal Paneri

Modeling causal relationships between components of dynamic systems helps predict the outcomes of interventions on the system. Upon an intervention, many systems reach a new equilibrium state. Once the equilibrium is observed, counterfactual inference predicts ways in which the equilibrium would have differed under another intervention. Counterfactual inference is key for optimal selection of interventions that yield the desired equilibrium state.

Complex dynamic systems are often described with Markov process (mechanistic) models, expressed as systems of ordinary or stochastic differential equations. They mimic interventions but do not support counterfactual inference. An alternative representation relies on structural causal models (SCMs). SCMs can represent the system at equilibrium, only require equilibrium data for parameter estimation, and support counterfactual inference. Unfortunately, multiple SCMs can represent the same observational or interventional distributions but provide different counterfactual insights. This drawback limits their practical use. Recent work has shown that for a Markov process with steady-state solution, it is possible to cast the system as an SCM.

Using two complex biomolecular systems, this thesis illustrates the steps of the approach and its implementation in the probabilistic programming language Pyro, scalable to realistic Markov process models with nonlinear dynamics. The SCM is defined in terms of the parameters and of the equilibrium dynamics of the Markov process model, and counterfactual inference flows from these settings. We further discuss different inference techniques for counterfactual inference and evaluate the identifiability of SCM via different experimental approaches and sensitivity analysis.
Acknowledgements

First and foremost, I would like to thank my parents and family for providing all the support that made me the person I am today.

Secondly, I would like to express sincere gratitude to my research advisor, Dr. Olga Vitek, for allowing me to conduct the research under her guidance. The mentorship was beyond an intellectual mentorship. The vision, sincerity, and motivation deeply inspired me. During my graduate work, she believed in me and supported through all the ups and downs. I feel fortunate to be a part of Olga Vitek Lab.

I would also like to express my sincere gratitude to Dr. Robert Ness. When I started with this project, I did not possess a firm grasp on many of the required background literature. He introduced to me the world of probabilistic programming, causality, and biology. Despite his busy schedule, he always got out of his way to answer all of my queries. I will always be thankful for the patience and admire the wit and intellectual curiosity.

I thank Dr. Jan-Willem van de Meent for agreeing for reading and providing critical comments on the thesis. During his course, Advanced machine learning and afterward, he continued supporting my research and was always available and willing to help.

And finally, the Olga Vitek Lab. Sara Taheri, Tyler Tuan, Sicheng Hao, Rashika Ramola, Shantanu Jain. Thank you all for all the support.
# Contents

Abstract iii

Acknowledgements v

1 Introduction 1
   1.1 Statement of biological problem ................................. 1
   1.2 Statement of statistical problem ................................ 1
   1.3 Contribution ...................................................... 2

2 Background 5
   2.1 Protein networks .................................................. 5
   2.2 Dynamical systems ................................................. 6
      2.2.1 Ordinary differential equations (ODE) models .............. 6
      2.2.2 Stochastic differential equation (SDE) models ............ 7
      2.2.3 Steady state ................................................... 9
   2.3 Case studies ....................................................... 9
      2.3.1 Elementary single protein reaction ......................... 10
      2.3.2 Mitogen-Activated Protein Kinase (MAPK) model .......... 10
      2.3.3 Insulin-like Growth Factor (IGF) model ................... 11
   2.4 Causal models and counterfactuals ............................... 11
      2.4.1 Structural Causal Models (SCM) ............................... 11
      2.4.2 Counterfactual inference .................................... 12
      2.4.3 Simulating interventions through ODE and SDE models .... 12

3 Related work 15
   3.1 Steady state reparameterization algorithm ...................... 15
   3.2 Properties of Markov process model M ............................ 18
   3.3 Inverse Binomial CDF transform is a monotonic conversion .... 18
   3.4 Converting case studies to an SCM at steady state ............. 18
      3.4.1 MAPK model .................................................... 18
      3.4.2 IGF model ...................................................... 19

4 Contribution 21
   4.1 Implementation of structural causal models in Pyro ............ 21
      4.1.1 Elementary single protein reaction .......................... 21
      4.1.2 Mitogen-Activated Protein Kinase (MAPK) model .......... 22
      4.1.3 Lessons learned ............................................... 23
   4.2 Inference of noise variables for counterfactual inference ...... 23
      4.2.1 Inverse CDF reparameterization: No need for inference ... 24
      4.2.2 Gaussian smoothing .......................................... 24
   4.3 Counterfactual robustness under model misspecification ........ 25
   4.4 Deterministic and stochastic counterfactual simulation and evaluation .................. 25
      4.4.1 MAPK model: Simulations and causal effect ............... 25
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4.2 IGF model: Simulations and causal effect</td>
<td>26</td>
</tr>
<tr>
<td>4.4.3 Counterfactual robustness under model misspecification</td>
<td>27</td>
</tr>
<tr>
<td>4.4.4 Implications of Gaussian approximation in MAPK and IGF models</td>
<td>27</td>
</tr>
<tr>
<td>4.4.5 Code and run time</td>
<td>28</td>
</tr>
<tr>
<td>4.5 Discussion</td>
<td>29</td>
</tr>
</tbody>
</table>

**Bibliography**
List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCM</td>
<td>Structural Causal Model</td>
</tr>
<tr>
<td>ODE</td>
<td>Ordinary Differential Equations</td>
</tr>
<tr>
<td>SDE</td>
<td>Stochastic Differential Equations</td>
</tr>
<tr>
<td>IGF</td>
<td>Insulin-like Growth Factor</td>
</tr>
<tr>
<td>MAPK</td>
<td>Mitogen-Activated Protein Kinase</td>
</tr>
<tr>
<td>CDF</td>
<td>Cumulative Distribution Function</td>
</tr>
<tr>
<td>PDF</td>
<td>Probability Density Function</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Statement of biological problem

Many problems in science can be viewed as studies of complex systems. They are governed by physical or abstract laws describing causal links between components of the system. Systems biology, in particular, deals with mathematical/mechanistic/Markov process models that describe the state of a system. Often, these systems contain elementary reactions between molecules. Concrete mathematical models such as ordinary differential equations (ODE) can be used to describe the temporal dynamics of deterministic systems and predict experimental observations. Under low concentration of molecules, stochastic models are used to predict the probabilistic outcome. An essential aspect of studying these systems is prediction the equilibrium behavior of the system and how the steady-state changes upon intervention [1].

- **Lack of prescribed formula for counterfactuals**
  
  For an observed outcome, the counterfactual inference can predict ways in which the equilibrium would have been different under different interventions. Unfortunately, Markov process models have no prescribed formula for counterfactual inference. Having observed equilibrium of a system trajectory in time, it is generally not possible to infer that trajectory from that observation. Even if the entire trajectory is observed, it is not clear how to re-play that trajectory with all things equal except those affected by the intervention. This makes counterfactual inference inaccessible for mechanistic modelers.

- **Model misspecification**
  
  These simulations can be highly susceptible to model-misspecification because of two main reasons. 1) Our limited understanding of molecular mechanisms, and 2) the fact that the models only describe a part of a broader molecular system, where the omitted components can affect the outcomes of the interventions. A biologist can have both experimental data from past intervention and a Markov process model, but not the possibility to conduct an experiment with a new intervention under the same external conditions. Therefore, rather than discard that data and only rely on the model, the investigator can use the data to more realistically assess the utility of various interventions in the past experimental setting. This can lead to more realistic biological insights.

1.2 Statement of statistical problem

Complex dynamical systems are often described with Markov process models, expressed as systems of discrete state continuous time Markov processes. They mimic interventions, but do not support counterfactual inference. An alternative representation relies on structural
equation models, often referred to as structural causal models (SCMs). SCMs can represent a stochastic model of a system at equilibrium, only require equilibrium data for parameter estimation, and support counterfactual inference [2].

- **Identifiability in SCMs**
  Unfortunately, multiple SCMs can represent the same observational or interventional distributions but provide different counterfactual insight. i.e., multiple SCMs may be consistent with the equilibrium distribution of the system components upon an intervention, but provide contradictory answers to the same counterfactual query [15, 16]. This limits their practical use.

- **Inference**
  SCMs have deterministic assignment of system components, and stochasticity resides in exogenous noise variables [16]. That means that variables are sampled from Dirac-delta distribution. Inference on Dirac-delta distribution is a well-known inference problem [20].

- **Model selection**
  The SCMs are parametric generative models, and parameter estimation from observational data leads to identifiability issues. This thesis contributes to the analysis of different frameworks for casting mechanistic models of a system observed at equilibrium as an SCM [6, 5]. The SCM retains the parameters in terms of the laws of the mechanistic model (e.g., its equilibrium solution). Counterfactual inference of the resulting SCM becomes rooted in the causal relationships described in the mechanistic model.

1.3 **Contribution**

- **Implementation of Structural Causal Models as probabilistic programs**
  Since structural causal models can be implemented as probabilistic programs, the first task is to implement them in a modern probabilistic programming language. This thesis demonstrates how SCMs can be implemented in Pyro and illustrate the process of carrying out interventions and counterfactual inference. Pyro is a PyTorch based probabilistic programming language that enables fast tensor-based inference techniques and includes Pearl’s do-operator [4, 15].

- **Enabling scalable inference techniques for SCMs**
  One of the essential steps in performing counterfactual inference on the resulting probabilistic program is inferring exogenous noise variables given observations. There are many challenges to make scalable inference work on SCM. i) SCMs have deterministic outputs, that means that variables are sampled from Dirac-delta distribution, and the stochasticity resides in the noise variables. Inference on Dirac-delta distribution is a well-known problem [20]. This thesis proposes Gaussian smoothing and shows empirically that it does not have any effect on the expected value of the resulting counterfactual distribution. ii) The technical difficulties of implementing PyTorch based variational inference techniques as Pyro does not have strong community support.

  This thesis also proposes a deterministic approach that eliminates the need for inference for SCMs in which variables are sampled from distributions for which their inverse CDFs are tractable.
• **Case Studies**
  This thesis provides multiple simple models as motivating examples. They establish a firm intuition of how these methods apply to more complex non-linear biological networks. This thesis provides probabilistic programs and methods for counterfactual inference on following two biological pathway models.

  – Mitogen-Activated Protein Kinase (MAPK) model [10]
  – Insulin-like Growth Factor (IGF) model [3]

• **Experimental Design and Evaluations**
  This thesis empirically demonstrates how initial rates and conditions flow through SCMs modeled at steady state. Evaluation methods include i) Sensitivity analysis on initial rates is performed to demonstrate the robustness of the proposed method ii) Visualizations demonstrating steady-state and causal effects.

• **Model Misspecification**
  Model misspecification is a prevalent issue in modeling biological systems due to stochasticity and unmeasured interactions. Counterfactual inference involves making use of past interventional or observational data to get new hypothetical interventions with the same initial conditions. This thesis shows that counterfactual inference is robust to model misspecification. Conditioning the misspecified model with past experimental data leads to counterfactual distribution that is very close to the true counterfactual.
Chapter 2

Background

2.1 Protein networks

Biomolecular networks can be categorized into three main groups: Gene regulatory, protein, and metabolic networks. Protein networks signify signaling pathways that integrate information about the internal and external environments and modulate both the metabolic and gene regulatory networks [19].

Protein networks are studied at different levels and are broadly classified into two categories: stoichiometric and non-stoichiometric. Non-stoichiometric networks often represent some physical or statistical associations between different proteins [19]. Such networks can be modeled using Bayesian networks ??.

Stoichiometric networks consider specific biochemical events, such as phosphorylation, and degradation, for which the stoichiometric relationships between proteins are known [19]. We consider stoichiometric networks as they describe underlying elementary chemical processes which give a physical causal relationship between system components. Signaling/-control protein networks are one of the types of protein networks that are extensively studied, and considerable detail has been accumulated in many different organisms. They operate as a signal processing network and are responsible for sensing external signals such as nutritional or cell signals such as insulin growth factor (IGF)[19]. For this thesis, phosphorylation is activation of protein. That means, in a network, if there is a directed edge from protein \( S \) to protein \( N \), \( S \) causes \( N \) to phosphorylate, which activates the downstream mechanism. The
signaling pathway ends on target proteins that lead to a change in cells behavior. Such targets include a variety of processes such as cytoskeleton, metabolic pathways, and gene regulatory proteins [19].

2.2 Dynamical systems

Many problems in science can be viewed as studies of complex systems. Examples include the economy, a disease population, an ecology, road traffic, or a cancer cell. Most complex systems are dynamic in nature. They are governed by physical or abstract laws (e.g., fluid dynamics) describing the causal links between components of the system. Mechanistic models (also called mathematical or dynamical models) express these laws in mathematical equations [7]. These equations describe the temporal dynamics of deterministic systems with ordinary differential equations (ODE) or, when the system is stochastic, with stochastic differential equations (SDE).

A primary goal of mechanistic modeling is to predict the outcome of interventions on the system, e.g., the outcome of introducing a new tax policy, or a drug in a cell. The intervention changes aspects of the model, such as the rate parameters or the initial conditions in an ODE or SDE. In many systems, it shifts the system from an old steady-state to dynamic transience, and then to a new steady state. Examples include a new price after a trade policy impacting supply [22, 24], or exposure of a drug in the biochemical network within a cell [21, 1]. Predicting the new steady-state behavior is the objective in many applications.

These computational models study the behavior of a complex system, and predict the effects of interventions to the system, through mathematical formalism and simulations [12]. Mechanistic models are a special case, describing the interactions between the model components in terms of abstract or physical processes rate laws with real-valued parameters rates [7]. In this work, we focus on SDE representations of mechanistic models, of which ODEs are a special case.

2.2.1 Ordinary differential equations (ODE) models

Ordinary differential equations models represents the physical laws as differential equations that describes the causal links between system components. In biological systems, if we assume continuum of values for protein concentration, we can model it deterministically with ODEs.

For example, consider a first-order irreversible degradation of a reactant $S$ to product $P$.

$$S \xrightarrow{v_1} P \quad (2.1)$$

The differential equation is given by the form

$$\frac{dS}{dt} = -v_1 S \quad \frac{dP}{dt} = v_1 S \quad (2.2)$$

Where $S$ is the molar amount of $S$. Once we have an ODE model, the next step is to solve the equations for their corresponding species by integrating them with respect to time $t$. Here, solving for $S$ yields
2.2. Dynamical systems

\[
\frac{dS}{dt} S = -v_1, \quad (2.3)
\]

\[
\int \frac{d}{dt} \ln S dt = -v_1 \int dt, \quad (2.4)
\]

\[
\ln S = -v_1 t + C. \quad (2.5)
\]

Here \( C \) is the constant of integration. If we assume that at \( t = 0, S = S_0 \), then \( C = \ln S_0 \). Substituting this result to the solution yields

\[
S = S_0 e^{-v_1 t},
\]

Here, \( v_1 \) rate parameter and \( S_0 \) is the starting condition. We will use this vocabulary when expressing equations for complex biological case studies.

2.2.2 Stochastic differential equation (SDE) models

Deterministic models of biochemical reactions ignore the fact that at the molecular level, concentrations can be described using discrete values representing the number of molecules. Problems arise when we deal with low concentrations. Brownian motion becomes a significant factor in determining reaction. A stochastic model thus, is required as the collisions between molecules are unpredictable. Let’s continue with the simple example 2.22. The derivation and notations below is motivated from “Systems biology: Introduction to pathway modeling” by Herebert M. Sauro Section 6.3 [19].

Suppose the \( cdt \) is the probability that a reactant molecule will react in the next time interval \( dt \). Here, \( c \) is the average probability that a reactant will react per unit time. We have a population of molecules in the system. Let \( h \) be the total number of reactant combinations. Let \( a = hc \). Thus, probability of a reaction occurring within entire population in \( dt \) time is \( hcdt \).

Suppose \( \bar{p} \) is the probability of no reaction from time 0 to \( t \). Then probability of no reaction from 0 to \( t + dt \) is

\[
\bar{p}(0, t + dt) = \bar{p}(0, t)\bar{p}(t, t + dt), \quad (2.6)
\]

\[
= \bar{p}(0, t)(1 - adt), \quad (2.7)
\]

\[
\bar{p}(0, t + dt) - \bar{p}(0, t) = \bar{p}(0, t)adt, \quad (2.8)
\]

\[
\frac{d\bar{p}(0, t)}{dt} = -\bar{p}(0, t)a. \quad (2.9)
\]

This differential equation above describes how probability of no reaction from 0 to \( t \) changes as we progress through time \( t \). Solving this equation for \( \bar{p}(0, t) \),

\[
\frac{d\bar{p}(0, t)}{dt} \frac{1}{\bar{p}(0, t)} = -a, \quad (2.10)
\]

\[
\ln \frac{\bar{p}(0, t)}{\bar{p}(0, t)} = -a, \quad (2.11)
\]

\[
\int \ln \frac{\bar{p}(0, t)}{\bar{p}(0, t)} dt = \int -adt, \quad (2.12)
\]

\[
\bar{p}(0, t) \approx e^{-at}. \quad (2.13)
\]
Chapter 2. Background

The probability density \( p(t) \) for the time \( t \) at which a reaction occurs is.

\[
p(t)dt = \delta(0,t)p(t+dt), \quad (2.14)
= \delta(0,t)dt, \quad (2.15)
p(t) = ae^{-at}. \quad (2.16)
\]

Equation 2.16 describes a PDF for \( t \). It’s simple to integrate PDF and get the CDF.

\[
\int p(t)dt = \int ae^{-at}dt, \quad (2.17)
u = P(t) = 1 - e^{-at}, \quad (2.18)
t = \frac{1}{a} \ln(1 - u), \quad (2.19)
\]

We can sample \( t \sim P(t) \) by

\[
t = P^{-1}(u) = \frac{1}{a} \ln(u) \quad \quad \quad u \sim Uniform(0,1) \quad (2.21)
\]

Equation 2.21 is the core equation for Gillespie Stochastic Simulation Algorithm (Gillespie SSA). Algorithm 1 describes the method of simulation used for all the complex systems described in the thesis. This algorithm outputs one simulation that can be one of many trajectories that could have been unfolded. For this project, we repeatedly applied this algorithm with same conditions and combined multiple trajectories with a single average trajectory.

**Algorithm 1 Gillespie SSA[19]**

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_0 )</td>
<td>Simulation trajectory</td>
</tr>
<tr>
<td>( x_0 )</td>
<td></td>
</tr>
<tr>
<td>( c )</td>
<td></td>
</tr>
<tr>
<td>( t_{max} )</td>
<td></td>
</tr>
</tbody>
</table>

1: **procedure** GILLESPIE\((t_0, x_0, c, t_{max})\)
2: Set \( t = t_0 \)
3: Set \( S = S_0 \)
4: Compute \( h \) (reactant combinations) depending on the nature of the reaction
5: Evaluate propensity function \( a = hc \)
6: Draw a uniform random number \( u \sim Uniform(0,1) \)
7: **\( \triangleright \) Determine time \( \tau \), when the next reaction will take place**
8: \( \tau = \frac{1}{a} \ln(1 - u) \)
9: **\( \triangleright \) Determine the new state and update time**
10: \( t = t + \tau \)
11: \( S = S - 1 \)
12: Is \( t > t_{max} \)?
13: No, go to step 3.
14: Yes, finished.
2.2.3 Steady state

A steady-state can be a stationary solution, single point, or an equilibrium point. Note that steady-state does not mean thermodynamic equilibrium.

Steady-state is one of the most important states to consider in a dynamical model, which is a reference point from which to consider the behavior of the model. At steady state, the concentration of all proteins is constant. In other words, at steady state, the rate of change $\frac{dS}{dt} = 0, \forall S$, but at the same time net rate is non-zero. That means, the matter can be exchanged with the surroundings in an open system, but the total concentration of protein remains constant.

There are two ways we can derive the steady-state solution of a Markov process model. The easy way is to set the derivatives to 0. This may not be possible for complicated models, then simplifying assumptions makes complex models have a steady-state. In that case, we can get the solution by taking limit $\lim_{t \to \infty}$.

Consider the ODE example Eq. (2.2). Let us add surrounding rate parameters in order to get a reasonable steady state.

\[
\begin{align*}
\rightarrow S & \rightarrow P \rightarrow \\
S & = \frac{v_0}{v_1} \quad P = \frac{v_0}{v_2}
\end{align*}
\]

Then the change of $S$ and $P$ through time is

\[
\frac{dS}{dt} = v_0 - v_1 S \quad \frac{dP}{dt} = v_1 S - v_2 P
\]

Using the first way and setting both the equations to 0, we get

\[
S = \frac{v_0}{v_1} \quad P = \frac{v_0}{v_2}
\]

2.3 Case studies

Define a real valued random variables $X(t) = \{ X_i(t) : i \in J \}$ where $J$ is a set of indices $J := 1, ..., D$. The variables reflect the state of the $D$ system components in time. We use capital letters to refer to random variables, lower case letters to refer to instances of random variables, normal font for a single variable, and boldface for a tuple of variables. A Markov process model $M$ is defined by master equations, a coupled set of SDEs that describe the probabilities of the states of the system components at a time [23]. Denote $P^M(t)$ the probability density of $X(t)$, and $p^M_{X_i}(t)$ the marginal probability of $X_i(t)$. Then the master equations are

\[
\frac{dP^M_{X_i}(t)}{dt} = g_i \left( t, v_i, \text{PA}_{M,i}(t) \right), \quad X_i(0) = (x_0)_i, \quad \forall i \in J
\]

The function $g_i$ determines the probability of a state change between $X_i(t)$ and $X_i(s), s > t, v_i$ is a set of parameters of the rate laws, and $x_0$ is an initial condition. $\text{PA}_{M,i}(t) \subseteq X(t) \setminus X_i(t)$ is the set of parents of variable $X_i(t)$, i.e. variables that regulate $X_i(t)$. Because of this structure, Markov process models are often visualized as directed graphs, with one node per variable and a directed edge from $X_i(t)$ to $X_j(t)$ if $\frac{dP^M_{X_j}(t)}{dt}$ depends on $X_i(t)$.

Like ODEs, we can also solve SDEs for corresponding $p^M_{X_i}$. Note that, here the solution will be the probability distribution (PDF) of $X_i$. 
2.3.1 Elementary single protein reaction

Fig. (2.2) illustrates a dynamic system, where component $y$ is regulated by components $x_1$ and $x_2$. Each component is in either an active or inactive state. The model is a simplified representation of a biochemical reaction under a mass action kinetic assumption [1]. The laws governing the regulation are

\[ x_1^{\text{on}} + y^{\text{off}} \xrightarrow{v_1} x_1^{\text{on}} + y^{\text{on}} \]  
\[ x_2^{\text{on}} + y^{\text{on}} \xrightarrow{v_2} x_2^{\text{on}} + y^{\text{off}} \]  

In this example, the Markov process model $M$ includes reactions 2.24 and 2.25, rates $v = \{v_1, v_2\}$, the reactions regulating the state of $x_1$ and $x_2$, as well as the initial states. Algorithm 5 summarizes the steps of converting such Markov process model into an SCM. The steps of the algorithm are a series of mathematical derivations (as opposed to a pseudocode for a computational implementation). Below we illustrate these steps for the component $y$.

**Finding the steady-state probability distribution for $Y(t)$**

The hazard rate function of a biochemical reaction is the probability that the reaction occurs in a given instant. It is determined by the particle counts at that instant, and by the rate parameters. Let $h_1(Y(t))$ and $h_2(Y(t))$ represent the hazard rate functions for activation and deactivation respectively:

\[ h_1(Y(t)) = v_1 X_1(t)(T_y - Y(t)) \]  
\[ h_2(Y(t)) = v_2 X_2(t)Y(t) \]

$T_y$ here is total number of particles of $Y$. Let $S_1$ and $S_2$ denote the change in particle count after reactions in Eq. (2.24).

\[ S_1 = 1 \]  
\[ S_2 = -1 \]

The Kolmogorov forward equations determine the change in $\pi(y, t)$ as the system evolves in time.

\[ \frac{d}{dt} \pi(y, t) = \sum_{i=1}^{2} (h_i(y - S_i)\pi(y - S_i, t) - h_i(y)\pi(y, t)) \]

Let $E_\pi(\cdot)$ denote the conditional expectation function over $\pi(y, t)$, i.e. $E_\pi(f(Y(t))) = \sum_{y=0}^{T_y} f(y)\pi(y, t)$.

2.3.2 Mitogen-Activated Protein Kinase (MAPK) model

The system The mitogen-activated protein kinase (MAPK) pathway is important in many biological processes, such as determination of cell fate. It is a cascade of three proteins, a
2.4. Causal models and counterfactuals

MAPK, a MAPK kinase (MAP2K), and a MAPK kinase kinase (MAP3K), represented with a causal diagram [9, 17]

Here E1 is an input signal to the pathway. The cascade relays the signal from one protein to the next by changing the count of proteins in an active state.

**The biochemical reactions** A protein molecule is in an active state if it has one or more attached phosphoryl groups. Each arrow in Eq. (2.28) combines the reactions of phosphorylation (i.e., activation) and dephosphorylation (i.e., deactivation). For example, E1 → MAP3K combines two reactions

\[ E1 + \text{MAP3K} \xrightarrow{v_{act}^{K3}} E1 + \text{P-MAP3K}, \quad \text{P-MAP3K} \xrightarrow{v_{inh}^{K3}} \text{MAP3K}. \]  

(2.29)

In the first reaction in Eq. (2.29), a particle of the input signal E1 binds (i.e., activates) a molecule of MAP3K to produce MAP3K with an attached phosphoryl. The rate parameter associated with this reaction is \( v_{act}^{K3} \). In the second reaction, phosphorylated MAP3K loses its phosphoryl (i.e., deactivates), with the rate \( v_{inh}^{K3} \). The remaining arrows in Eq. (2.28) aggregate similar reactions and rate pairs.

**The Markov process model** Let \( K3(t) \), \( K2(t) \) and \( K(t) \) denote the counts of phosphorylated MAP3K, MAP2K, and MAPK at time \( t \). Let \( T_{K3} \), \( T_{K2} \), and \( T_K \) represent the total amount of each of the three proteins, and \( E1 \) the total amount of input, which we assume are constant in time. We model the system as a continuous time discrete state Markov process \( M \) with hazard rates functions in Table 2.1.

<table>
<thead>
<tr>
<th>activation hazard</th>
<th>MAP3K</th>
<th>MAP2K</th>
<th>MAPK</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v_{act}^{K3}(T_{K3} - K3(t)) )</td>
<td>( v_{act}^{K2}(T_{K2} - K2(t)) )</td>
<td>( v_{act}^{K}(T_k - K(t)) )</td>
<td></td>
</tr>
</tbody>
</table>

**Table 2.1**: The hazard functions in MAPK, specified according to mass action enzyme kinetics.

### 2.4.1 Structural Causal Models (SCM)

For a same system with the set of components \( J \), a structural causal model \( C \) is often visualized as a causal directed graph \( G \). The model consists of [15, 16]:

1. Random variables \( X = \{ X_i; i \in J \} \) corresponding to the states of the system (and to the nodes in \( G \))
Chapter 2. Background

Figure 2.3: IGF signaling. The top nodes are receptors for the epidermal growth factor (EGF) and the insulin growth factor (IGF). Downstream of the receptors are several canonical signaling pathways (including Raf-Mek-Erk, a renamed equivalent of Eq. (2.28)). Each reaction has a single rate parameter. The auto-deactivation reactions are not pictured.

2. A distribution $P_C^i$ on a set of independent random variables $N = \{N_i; i \in J\}$ called noise variables.

3. A set of functions $f = \{f_i, i \in J\}$ called structural assignments, such that

$$X_i := f_i(\text{PA}_{C,i}, N_i), \forall i \in J$$

(2.30)

where $\text{PA}_{C,i} \subseteq X \setminus X_i$ are the parents of $X_i$ in $G$.

$C$ is a generative model that entails an observational distribution $P_C$. This means that a procedure that first samples noise values from $P_C^N$, and then sets the values of $X$ deterministically with $f$, generates samples from $P_C$. This is viewed as the generating process for the observed $X$.

$d$ is the number of variables in the model. $N_i$ is called noise variable for variable $j$. The value of each variable is set deterministically by a function $f_j$ for the $j$th random variable called a structural assignment, such that $X_j := f_j(\text{PA}_{C,j}, N_j), j = 1, \ldots, d$.

2.4.2 Counterfactual inference
We use Pearl’s $\text{do}$-operator to represent an intervention [15].

Consider an SCM $C = (S, P_N)$ over nodes $X$. Given some observations $x$, we define a counterfactual SCM by replacing the distribution of noise variables:

$$C^{\text{do}}(X=x) := (S, P_N^{\text{do}}(X=x))$$

Where $P_N^{\text{do}}(X=x)$ is the new set of noise variables need not be jointly independent anymore. Counterfactual statement can be seen as $\text{do}$-statements in the new counterfactual SCM.

Algorithm 2 shows the Bayesian counterfactual inference.

2.4.3 Simulating interventions through ODE and SDE models
We consider intervention on the rate parameters. ODE and SDE models can efficiently simulate new steady-state given the changed rate parameters. We then define the causal effect as the difference between the new steady-state vs. the previous steady-state. This is same as the counterfactual statement in the corresponding SCM. Observing a sample; what would
2.4. Causal models and counterfactuals

Algorithm 2 Counterfactual inference on SCM

Inputs: Prior distribution on exogenous noise NPrior
Structural causal model C
Observed endogenous variables $X = x$
Counterfactual interventions $X = \neg x$
Desired sample size ssize

Output: ssize samples from $p_{C;X=x,do(X=\neg x)}$

1: procedure CFQUERY(C, NPrior, x, $\neg x$, ssize)
2:   ▶ Create “observation” and “intervention” models
3:   obsModel ← Condition(C, $X = x$)
4:   intModel ← Do(C, $X = \neg x$)
5:   ▶ Infer noise distribution with observation model
6:   NPosterior ← Infer(obsModel, NPrior)
7:   ▶ Simulate from intervention model w/ updated noise
8:   samples = array(ssize)
9:   for i in (0:ssize) do
10:      samples[i] ← intModel(NPosterior)
11: return samples

have happened, had the activation rate of the upstream protein was higher? Algorithms (3) and (4), respectively, are used to simulate interventions and compute causal effects through ODE and SDE models.
Algorithm 3 Deterministic counterfactual simulation and evaluation with Markov process model

Inputs: Markov process model $M$
Rate sets $v$ and $v'$
steady-state time point $T$
Index of counterfactual intervention target $i$
Index of counterfactual query target $j$

Output: Structural causal model $C$

1: procedure CF-ODE($M$, $v$, $v'$, $T$, $i$, $j$)
2:   $\blacktriangleright$ Simulate a deterministic equilibrium using $v$
3:   $x = \text{ODEsolve}(M, v)[T]$
4:   $\blacktriangleright$ Simulate a deterministic steady-state using $v'$
5:   $x' = \text{ODEsolve}(M, v')[T]$
6:   $\blacktriangleright$ Calculate difference in counterfactual target
7:   for index $k$ in array $\Delta$ do
8:     $\delta_{\text{true}} = x'_j - x_j$
9:     $\blacktriangleright$ Simulate CF value from SCM
10:    $x^* \sim P_{C:X=x,do(X_i=x'_i)}$
11:    $\blacktriangleright$ Calculate difference
12:    $\Delta[k] = x^*_j - x_j$
13:    $\blacktriangleright$ Compare $\Delta$ to $\delta_{\text{true}}$
14:    histogram($\Delta$, vertical-line = $\delta_{\text{true}}$)

Algorithm 4 Stochastic counterfactual simulation and evaluation with Markov process model

Inputs: Markov process model $M$
Rate sets $v$ and $v'$
steady-state time point $T$
Index of counterfactual intervention target $i$
Index of counterfactual query target $j$
List of random seeds $S$

Output: Structural causal model $C$

1: procedure CF-SDE($M$, $v$, $v'$, $T$, $i$, $j$, $S$)
2:   $\blacktriangleright$ Simulate a deterministic steady-state using $v'$
3:   $x'd' = \text{ODEsolve}(M, v')[T]$
4:   for index $k$ in $S$, & collectors $\Delta_M$, $\Delta_C$ do
5:     $\blacktriangleright$ Simulate a stochastic equilibrium using $v$
6:     $x^s = \text{SDEsolve}(M, v, \text{seed} = S[k])[T]$
7:     $\blacktriangleright$ Simulate a stochastic steady-state using $v'$
8:     $x'^s = \text{SDEsolve}(M, v', \text{seed} = S[k])[T]$
9:     $\blacktriangleright$ Calculate difference in counterfactual target
10:    $\Delta_M[k] = x'^s_j - x^s_j$
11:    $\blacktriangleright$ Simulate CF value from SCM using $x'^s$
12:    $x_j^s \sim P_{C:X=x,do(X_i=x'_i)}$
13:    $\blacktriangleright$ Calculate difference
14:    $\Delta_C[k] = x^*_j - x^s_j$
15:    $\blacktriangleright$ Compare $\Delta_C$ to $\Delta_M$
16:   overlayHistograms($\Delta_C, \Delta_M$)
Chapter 3

Related work

3.1 Steady state reparameterization algorithm

Algorithm 5 summarizes the proposed steps of converting the Markov process model into an SCM. The steps are a series of mathematical derivations (as opposed to a pseudocode for a computational implementation). Below we illustrate these steps for the component Y in the elementary single protein reaction example.

Algorithm 5  Convert Markov process into SCM
Inputs: Markov process model $M$
Output: Structural causal model $C$

1: procedure GetSCM($M$)
2:   ▶ Solve master equation
3:     $p_M(t) := \int_0^t \frac{dp_M(t)}{dt} dt$
4:   ▶ Find the steady-state distribution
5:     $p_M = \lim_{t \to \infty} p_M(t)$
6:   ▶ Use $p_M$ to define generative model $G$
7:     $G := \{x \sim p_M\}$
8:   ▶ Convert the generative model to an SCM
9:     ▶ that entails $p_M$
10:    $C := \{N \sim p_C, X = f(X, N) : p_C \approx p_M\}$
11: return $C$

Building a probability model of the system

Let $X_1(t)$, $X_2(t)$, and $Y(t)$ represent the total active-state particle count in a cell of $X_1$, $X_2$, and $Y$ respectively at time $t$. Let $T_y$ represent the total particle count (active and inactive) in a cell of $Y$, such that $T_y - Y(t)$ is the number of inactive particles of $Y$ at time $t$.

Let $\pi(y, t) = P(Y(t) = y | T_y, X_1(t), X_2(t))$ represent the conditional probability distribution of $Y(t)$. Each particle of $Y$ is in active state with some probability, i.e. a Bernoulli trial. Therefore $Y(t)$ is the sum of Bernoulli random variables such that $\pi(y, t)$ is a Binomial distribution with $T_y$ trials.

Let $\theta_y(X_1(t), X_2(t))$ denote the probability that a particle of $Y$ is in active state at time $t$. This probability is needed to fully specify the Binomial distribution. The following derivation demonstrates how $\theta_y(X_1(t), X_2(t))$ is a function of $X_1(t)$, $X_2(t)$.

Finding the steady-state probability distribution for $Y(t)$

The hazard rate function of a biochemical reaction is the probability that the reaction occurs in a given instant. It is determined by the particle counts at that instant, and by the rate parameters. Let $h_1(Y(t))$ and $h_2(Y(t))$ represent the hazard rate functions for activation and
deactivation respectively:

\begin{align*}
    h_1(Y(t)) &= v_1 X_1(t)(T_y - Y(t)) \\
    h_2(Y(t)) &= v_2 X_2(t) Y(t)
\end{align*}

(3.1)

(3.2)

Let $S_1$ and $S_2$ denote the change in particle count after reactions in Eq. (2.24).

\begin{align*}
    S_1 &= 1 \\
    S_2 &= -1
\end{align*}

The Kolmogorov forward equations determine the change in $\pi(y, t)$ as the system evolves in time.

\[
    \frac{d}{dt} \pi(y, t) = \sum_{i=1}^{2} (h_i(y - S_i) \pi(y - S_i, t) - h_i(y) \pi(y, t))
\]

Let $E_\pi(\cdot)$ denote the conditional expectation function over $\pi(y, t)$, i.e. $E_\pi(f(Y(t))) \equiv \sum_{y=0}^{T_y} f(y) \pi(y, t)$. **Solve the master equation** (line 3). Solving $P^M_{Y(t)}$ in Eq. (3.9) is equivalent to solving the ordinary differential equation on the expectation $E(Y(t))$ of $Y(t)$ over $P^M_{Y(t)}$:

\[
    \frac{d}{dt} E(Y(t)) = v_1 X_1(t) T - (v_1 X_1(t) + v_2 X_2(t)) E(Y(t))
\]

(3.3)

This has an analytical solution, where $g(u) = \frac{u}{u+1}$:

\[
    \frac{E(Y(t))}{T} = e^{-t(v_1 X_1(t) + v_2 X_2(t))} + g(v_1 X_1(t) + v_2 X_2(t))
\]

(3.4)

Finally, $Y(t)$ is a count of binary state variables with the same probability of being activated at a given instant. Then $P^M_{Y(t)}$ must be Binomial distribution with $T$ trials, and trial probability $E(Y(t)) / T$.

If hazards are zero or first order, then the linearity property of the expectation operator allows for an analytical solution. Without loss of generality, we demonstrate this with the motivating example. Substituting in the hazard functions in Eq. (3.2):

\[
    \frac{d}{dt} E_\pi(Y(t)) = \sum_{i=1}^{2} S_i E_\pi(h_i(Y(t)))
\]

\[
    = E_\pi v_1 X_1(t)(T_y - Y(t)) - E_\pi v_2 X_2(t) Y(t)
\]

\[
    = v_1 X_1(t) T_y - v_1 X_1(t) E_\pi(Y(t)) - v_2 X_2(t) E_\pi(Y(t))
\]

\[
    = v_1 X_1(t) T_y - (v_1 X_1(t) + v_2 X_2(t)) E_\pi(Y(t))
\]

(3.5)

Let $\theta_y(X_1(t), X_2(t))$ denote the probability that a particle of $Y$ is in active state at time $t$. $\theta_y(X_1(t), X_2(t))$ is a function $X_1(t), X_2(t)$. This follows from the fact that $\pi(y, t)$ has a Binomial distribution, and therefore $\theta_y$ is determined by the expectation $E_\pi(Y(t))$.

\[
    \theta_y(X_1(t), X_2(t)) \equiv \frac{E_\pi(Y(t))}{T_y}
\]

(3.6)
3.1. Steady state reparameterization algorithm

Substituting Equation 3.6 into 3.5:

\[
\frac{d}{dt} E\pi(Y(t)) = v_1 X_1(t) T_y - (v_1 X_1(t) + v_2 X_2(t)) T_y \theta_y(X_1(t), X_2(t))
\]

\[
\frac{d}{dt} \theta_y(X_1(t), X_2(t)) = v_1 X_1(t) - (v_1 X_1(t) + v_2 X_2(t)) \theta_y(X_1(t), X_2(t))
\] (3.7)

The analytical solution is:

\[
\theta_y(X_1(t), X_2(t)) = e^{-t(v_1 X_1(t)+v_2 X_2(t))} + \frac{v_1 X_1(t)}{v_1 X_1(t) + v_2 X_2(t)}
\] (3.8)

Therefore the probability distribution of \(Y(t)\) is given by

\[
Y(t) \sim \text{Binomial}(T_y, \theta_y(X_1(t), X_2(t)))
\] (3.9)

\(\theta_y(X_1(t), X_2(t))\) achieves steady-state when \(\frac{d}{dt} \theta_y(X_1(t), X_2(t)) = 0\). The solution is found by setting the left-hand side in Eq. (3.7) to 0, or alternatively, taking the limit in time of Eq. (3.8)

\[
\lim_{t \to \infty} \theta_y(X_1(t), X_2(t)) = \frac{v_1 X_1(t)}{v_1 X_1(t) + v_2 X_2(t)}
\] (3.10)

\(\theta_y(X_1(t), X_2(t))\) is the only component of the Binomial probability distribution of \(Y(t)\) that varies in time. The steady-state solution of Eq. (3.8) also provides the steady-state distribution of \(Y(t)\) (also referred to as stationary or invariant distribution in stochastic process literature). For simplicity we assume that the counts of active \(X_1\) and \(X_2\) are also the results of processes with stationary distributions. Let \(Y, X_1\) and \(X_2\) represent steady state active particle counts for \(Y, X_1,\) and \(X_2\). Then the steady-state distribution of \(Y\) is given by:

\[
\theta_y(X_1, X_2) := \frac{v_1 X_1}{v_1 X_1 + v_2 X_2}
\] (3.11)

\[
Y \sim \text{Binomial}(T_y, \theta_y(X_1, X_2))
\] (3.12)

Use \(p^M\) to define generative model \(G\) (line 7). Let \(\theta_{X_1}\) and \(\theta_{X_1}\) be the probability parameters for the steady-state Binomial distributions \(p^M_{X_1}\) and \(p^M_{X_2}\). Let \(\theta_Y(X_1, X_2) = \frac{E(Y)}{T}\) be the probability parameter for the steady-state Binomial distribution \(p^M_Y\). Then we define a generative model \(G\) as:

\[
G := \{X_1 \sim \text{Binom}(T, \theta_{X_1}); X_2 \sim \text{Binom}(T, \theta_{X_2}); Y \sim \text{Binom}(T, \theta_Y(X_1, X_2))\}
\] (3.13)

Convert the generative model to an SCM that entails \(p^M\) (line 10). We rely on a method of monotonic conversion, which restricts the class of possible SCMs to those with consistent counterfactual quantities (such as the probability of necessity, i.e. the probability that \(Y\) would not have been activated without \(X_1\)) [15]. For each structural assignment \(X_i = f_i(\text{PA}_{C,i}, N_i), \forall i \in J\) the method enforces the property \(E[X_i \mid \text{do}(	ext{PA}_{C,i} = y)] \geq E[X_i \mid \text{do}(	ext{PA}_{C,i} = y')] \Rightarrow f_i(y, n_i) \geq f_i(y', n_i)\forall n_i\).

For this example we selected a monotonic conversion by means of the inverse CDF transform. Denote \(F^{-1}(u, n, p)\) the inverse CDF of the Binomial distribution, where \(0 < u < 1\), and \(n\) (number of trials) and \(p\) (success probability) are the parameters of the Binomial
distribution. Then the SCM \( C \) that entails \( P^M \) is defined as

\[
N_{X_1}, N_{X_2}, N_Y \overset{\text{ind}}{\sim} \text{Uniform}(0,1); \\
C = \{ X_1 := F^{-1}(N_{X_1}, T, \theta_{X_1}); X_2 := F^{-1}(N_{X_2}, T, \theta_{X_2}); Y := F^{-1}(N_Y, T, \theta_Y(X_1, X_2)) \}
\]  

(3.14)

3.2 Properties of Markov process model \( M \)

We assume a Markov process models with a unique steady state distribution and no cycles. Therefore, the steady-state distribution is factorized according to a directed acyclic graph, given by solving each \( \frac{dP(X(t))}{dt} \) for each variable \( X \) in the model. We cast this distribution as a causal Bayesian network, with the conditional probability distributions given by the steady-state distributions.

Furthermore, we assume that hazard functions are zero or first order.

3.3 Inverse Binomial CDF transform is a monotonic conversion

**Definition 3.3.1. Monotonic condition.** A variable \( Y \) is said to be monotonic relative to variable \( X \) in a given structural causal model if and only if, given \( X = x \) and noise variable \( N = n \), the structural assignment \( f_Y(x, n) \) is monotonic in \( x \) for all \( n \). If the monotonicity condition is true, then \( E(Y|do(X = x)) \geq E(Y|do(X = x')) \Rightarrow f_Y(x, n) \geq f_Y(x', n) \) \( \forall n \) [14, 15].

**Definition 3.3.2. Monotonic conversion.** A monotonic conversion is a conversion of a probabilistic generative model \( Y \) to a structural assignment (which assigns a value of \( Y \) to a deterministic function of a random noise input) such that the assignment satisfies the monotonic condition.

**Lemma** 3.3.1. Let \( N \) be a noise variable with a Uniform(0,1), and let \( n \) be a sample of \( N \). Let random variable \( Y \) be generated from a probabilistic generative model \( Y \sim \text{Binomial}(T, \theta(x)) \). Let \( T \) be the total number of trials, and \( \theta(x) \) be the success probability, where \( \{ x, \theta(x) : x \in \mathbb{N}, 0 < \theta(x) < 1 \} \). Assume that \( \theta(x) \) is monotonic in \( x \) (as in Eq. (3.11), except with \( X_2 \) held constant). Let \( F^{-1}(\theta(x), T, n) \) denote the inverse Binomial CDF of \( Y \), parameterized by \( \theta(x) \) and \( T \). If \( E(Y|do(X = x)) \geq E(Y|do(X = x')) \), then the inverse CDF of \( Y \) \( f_Y(x, n) = F^{-1}(\theta(x), T, n) \) is a monotonic conversion.

**Proof.** Let \( y = E(Y|do(X = x)) \) and \( y' = E(Y|do(X = x')) \). Given \( y \) and \( y' \), there exists a value of \( n^* \in (0, 1) \) such that \( y = F^{-1}(\theta(x), T, n^*) \) and \( y' = F^{-1}(\theta(x'), T, n^*) \). Therefore, if \( y \geq y' \) then \( F^{-1}(\theta(x), T, n^*) \geq F^{-1}(\theta(x'), T, n^*) \).

\[ \square \]

3.4 Converting case studies to an SCM at steady state

3.4.1 MAPK model

Solve stochastic process’s master equation (Algorithm 5 line 3). As in the motivating example, we compactly express \( \frac{dP^M(t)}{dt} \) in terms of the expectation \( E(K3(t)) \) over \( P^M(t) \).
3.4. Converting case studies to an SCM at steady state

The master equation for K3(t) is \( \frac{dE(K3(t))}{dt} = v_{K3}^{inf}E1(T_{K3} - K3(t)) - v_{K3}^{inh}K3(t) \). We derive similar master equations for K2(t) and K(t). We solve the ODE above:

\[
\frac{E(K3(t))}{T_{K3}} = e^{-t(v_{K3}^{act} + v_{K3}^{inh})} + \frac{v_{K3}^{act}E1}{v_{K3}^{act}E1 + v_{K3}^{inh}}
\]

(3.15)

and obtain the steady-state by taking the limit \( t \to \infty \). The first term in Eq. (3.15) goes to 0:

\[
\frac{E(K3)}{T_{K3}} = \frac{v_{K3}^{act}E1}{v_{K3}^{act}E1 + v_{K3}^{inh}}
\]

(3.16)

**Find the steady-state distribution** (Algorithm 5 line 5) Each active-state MAPK protein has a Binomial marginal distribution. Let \( \theta_{K3}(E1) \) denote the probability that a MAP3K particle is active at steady-state given E1. After solving the master equation,

\[
\theta_{K3}(E1) = \frac{E(K3)}{T_{K3}} = \frac{v_{K3}^{act}E1}{v_{K3}^{act}E1 + v_{K3}^{inh}}
\]

(3.17)

Extending this solution to MAP2K and MAPK leads to probabilities

\[
\theta_{K3}(E1) = \frac{v_{K3}^{act}E1}{v_{K3}^{act}E1 + v_{K3}^{inh}} \theta_{K2}(K3) = \frac{v_{K3}^{act}K3}{v_{K3}^{act}K3 + v_{K3}^{inh}}; \theta_{K}(K2) = \frac{v_{K}^{act}K2}{v_{K}^{act}K2 + v_{K}^{inh}}
\]

(3.18)

and the following steady-state distributions:

\[
P_{K3}^{M} \equiv \text{Binomial}(T_{K3}, \theta_{K3}(E1)); P_{K2}^{M} \equiv \text{Binomial}(T_{K2}, \theta_{K2}(K3)); P_{K}^{M} \equiv \text{Binomial}(T_{K}, \theta_{K}(K2))\]

Use \( P_{K}^{M} \) to define generative model \( G \) (Algorithm 5 line 7). From here it is straightforward to create a generative model that entails \( P_{K}^{M} \):

\[
G := \{K3 \sim \text{Binom}(T_{K3}, \theta_{K3}(E1)); K2 \sim \text{Binom}(T_{K2}, \theta_{K2}(K3)); K \sim \text{Binom}(T_{K}, \theta_{K}(K2))\}
\]

Use \( P_{K}^{M} \) to define generative model \( G \) (Algorithm 5 line 10). Here the challenge is in expressing the stochasticity in \( G \), while defining K3, K2, K as deterministic functions of the noise variables \( N_{K}, N_{K2}, N_{K3} \). Instead of using the inverse binomial CDF, we demonstrate the use of a differentiable monotonic conversion, so that we can validate approximate counterfactual inference with stochastic gradient descent. We achieve this by first applying a Gaussian approximation to the Binomial distribution, and then applying the “reparameterization trick” used in variational autoencoders [18] (combined in helper function \( q \) in Eq. (4.5)).

\[
N_{K}, N_{K2}, N_{K3} \sim N(0, 1); q(\theta, T, N) = N \cdot (T(1 - \theta))^{1/2} + \theta T
\]

(3.21)

\[
C = \{K3 := q(\theta_{K3}(E1), T_{K3}, N_{K3}); K2 := q(\theta_{K2}(K3), T_{K2}, N_{K2}); K := q(\theta_{K}(K2), T_{K}, N_{K})\}
\]

The Gaussian approximation facilitates the gradient-based inference in line 6 of Algorithm 2. Despite the approximation, the resulting SCM is still defined in terms of \( \theta \). In this manner the SCM retains the biological mechanisms and the interpretation of the Markov process model.

3.4.2 IGF model

The derivations for the growth factor signaling system align closely with that of the motivating example and of the MAPK model. For variable \( X_i \) with parents \( PA_{M,j} \), we partition each parent set into activators and inhibitors \( PA_{M,j} = \{PA_{M,j}^{act}, PA_{M,j}^{inh}\} \). The rate parameters are
also partitioned into \( v = \{ v^{\text{act}}, v^{\text{inh}} \} \). For each \( X_i \) the probability for particle activation at steady-state is:

\[
\theta_{X_i}(P_{A_{M,i}}) = \frac{v^{\text{act}}P_{A_{M,i}}^{\text{act}}}{v^{\text{act}}P_{A_{M,i}}^{\text{act}} + v^{\text{inh}}P_{A_{M,i}}^{\text{inh}}} \tag{3.23}
\]

Next, we derive an SCM using the same Normal approximation to the Binomial distribution as in the MAPK pathway.
Chapter 4

Contribution

4.1 Implementation of structural causal models in Pyro

Structural causal models resulting from Algorithm 5 can be implemented as deterministic subroutines in a modern probabilistic programming language. In this chapter, we use Pyro to illustrate the implementation and the process of performing interventions and counterfactual inference. Pyro is a PyTorch based probabilistic programming language that enables fast tensor-based inference techniques and includes Pearl’s \( d \)-operator. \[4, 15\].

4.1.1 Elementary single protein reaction

Here is the structure causal model for elementary single protein reaction at steady state derived from Algorithm 5.

\[
N_{X_1}, N_{X_2}, N_{X_3} \overset{\text{ind}}{\sim} \text{Uniform}(0, 1)
\]

\[
X_1 := F^{-1}(N_{X_1}, T, \theta_{X_1})
\]

\[
X_2 := F^{-1}(N_{X_2}, T, \theta_{X_2})
\]

\[
Y := F^{-1}(N_{X}, T, \theta_Y(X_1, X_2))
\]

Where \( F \) is Binomial CDF. Note that for reparameterizing binomial random variables, we can either use gaussian approximation or directly reparameterize using inverse CDF. To cover both the cases, this model is using direct reparameterization and the MAPK model uses Gaussian approximation of Binomial variables.

```python
import pyro
import torch
from pyro.distributions import Uniform, Delta
from pyro import sample
from scipy.stats import binom

# Reaction rate
v1, v2 = torch.Tensor([0.8]), torch.Tensor([0.2])

def theta_Y(X1, X2):
    return v1 * X1 / (v1 * X1 + v2 * X2)

def f_inv(N, probs, num_trials=100):
    # detaching tensors to get them to work with scipy ppf function
    N = N.detach().numpy()
    probs = probs.detach().numpy()
    return torch.Tensor(binom.ppf(N, n=num_trials, p=probs))

def elementary_reaction(thetaX1, thetaX2):
    # Sample noises
    N_X1 = sample('N_X1', Uniform(0, 1))
    N_X2 = sample('N_X2', Uniform(0, 1))
    N_Y = sample('N_Y', Uniform(0, 1))

    # X1, X2, Y set deterministically
    X1 = sample('X1', Delta(f_inv(N_X1, thetaX1)))
    X2 = sample('X2', Delta(f_inv(N_X2, thetaX2)))
    Y = sample('Y', Delta(f_inv(theta_Y(X1, X2), N_Y)))
```


4.1.2 Mitogen-Activated Protein Kinase (MAPK) model

Here is the structure causal model for MAPK model at steady state derived from Algorithm 5.

\[
\begin{align*}
\theta_{K3}(E1) &= \frac{v_{K3}E1}{v_{K3}E1 + \alpha_{K3}} \\
\theta_{K2}(K3) &= \frac{v_{K2}K3}{v_{K2}K3 + \alpha_{K2}} \\
\theta_{K}(K2) &= \frac{v_{K}K2}{v_{K}K2 + \alpha_{K}}
\end{align*}
\]

\[
N_{K}, N_{K2}, N_{K3} \sim \mathcal{N}(0, 1)
\]

\[
g(\theta, T, N) = N \cdot (T\theta(1 - \theta))^{1/2} + \theta T
\]

\[
C := \begin{cases} 
K3 &= g(\theta_{K3}(E1), T_{K3}, N_{K3}) \\
K2 &= g(\theta_{K2}(K3), T_{K2}, N_{K2}) \\
K &= g(\theta_{K}(K2), T_{K}, N_{K})
\end{cases}
\]

Listing 4.1: Elementary single protein reaction model in Pyro

```python
import pyro
import torch
from pyro.distributions import Normal, Delta
from pyro import sample

rates = {
    'K3_activate': 0.1,
    'K3_deactivate': 0.1,
    'K2_activate': 0.1,
    'K2_deactivate': 2,
    'K_activate': 0.1,
    'K_deactivate': 1
}
totals = {
    'K3': 100,
    'K2': 100,
    'K': 100
}
def g(a):
    return a / (a + 1)
def f_K3(N):
    E1 = tensor(1.)
    weight = rates['K3_activate'] / rates['K3_deactivate']
    mu = totals['K3'] * weight
    sigma = mu * (1. - p)
    K3 = N * sigma + mu
    return K3
def f_K2(K3, N):
    weight = rates['K2_activate'] / rates['K2_deactivate']
    mu = totals['K2'] * weight
    sigma = mu * (1. - p)
    K2 = N * sigma + mu
    return K2
def f_K(K2, N):
    weight = rates['K_activate'] / rates['K_deactivate']
    mu = totals['K'] * weight
    sigma = mu * (1. - p)
    K = N * sigma + mu
    return K
```
4.2 Inference of noise variables for counterfactual inference

Algorithm 2 Step 6 is the inference step, where noise variables are inferred given an observation. It is challenging to do inference on SCMs as the observed system variables come from...
deterministic function. Therefore, they can be sampled from Dirac-delta in Pyro. Due to its point mass, all the well-known sampling and variational inference algorithms fail to work with Dirac-delta. [20, 13] In this section, two alternatives are discussed that enables inference in SCMs.

4.2.1 Inverse CDF reparameterization: No need for inference

If a variable is sampled from a distribution that has a tractable inverse CDF, we can reparameterize it with Uniform(0,1). That is, if $X$ is sampled from a distribution with CDF $F$, then it can be reparameterized as follows.

$$N_X \sim \text{Uniform}(0, 1)$$

$$X = F^{-1}(N_X)$$

$F^{-1}$ is a non-decreasing deterministic function that maps probabilities to the value of variable. If mechanisms of an SCM consist of tractable inverse CDFs, we can use CDF to get the noise variables. There is no need for inference.

For Elementary single protein reaction case study, Algo 2 Step 6 can be implemented as

```python
def F(x, prob):
    return binom.cdf(x, 100, prob) # Assuming 100 total particles of each protein
def get_noise(X1, X2, Y, thetaX1, thetaX2):
    N_X1 = F(X1, thetaX1)
    N_X2 = F(X2, 100, thetaX2)
    N_Y = F(Y, thetaY(X1, X2))
    return {
        'N_X1': N_X1,
        'N_X2': N_X2,
        'N_Y': N_Y
    }
```

LISTING 4.3: Inference in elementary single protein reaction

4.2.2 Gaussian smoothing

By definition, Dirac-delta distributions can be achieved by taking limit

$$\text{Dirac}(x) = \lim_{\sigma \to 0} \mathcal{N}(x, \sigma^2)$$

Even though variance in the resulting counterfactual distribution is essential for analysis, causal inference literature primarily focuses on the causal effect that puts heavy emphasis on expected value. If we tweak the definition of SCM as, instead of deterministic structural assignment, the variables are centered with small variance, this opens up SCMs to all different inference techniques, as long as the quantity of interest is the expected value of the resulting distribution. We call an SCM with this property as Soft SCM.

```python
def MAPK_soft():
    N_K3 = sample('N_K3', Normal(0, 1))
    N_K2 = sample('N_K2', Normal(0, 1))
    N_K = sample('N_K', Normal(0, 1))
    K3 = sample('K3', Normal(f_K3(N_K3), 1))
    K2 = sample('K2', Normal(f_K2(K3, N_K2), 1))
    K = sample('K', Normal(f_K(K2, N_K), 1))
    return {
        'K3': K3,
        'K2': K2,
        'K': K
    }

def infer_dist(MAPK_soft):
    return pyro.infer.Importance(MAPK_soft, num_samples=5000).run()
```

LISTING 4.4: Inference on soft SCM for MAPK model
4.3 Counterfactual robustness under model misspecification

All the complex systems covered in this thesis have steady-state, and that can be derived from their mechanistic nature and appropriate kinetics assumptions. A valid question one can ask is, why do we care for counterfactual reasoning in such systems if we can get the causal effect from simulations?

Systems biology models are subject to misspecification, due to (1) our limited understanding of molecular mechanisms, and (2) the fact that the models only describe a part of a broader molecular system, where the omitted components can affect the outcomes of the interventions. A biologist can have both experimental data from past intervention and a mechanistic model, but not the possibility to experiment with a new intervention under the exact same external conditions. Therefore, rather than discard that data and only rely on the model, the investigator can use the data to more realistically assess the utility of various interventions in the past experimental setting. This can lead to more realistic biological insights.

Suppose we know the model of a system at steady-state, and its true initial parameters. Consider an experiment where a biologist sets the model parameters that will be subject to misspecification. Because the misspecified model is conditioned with the true data as the part of the process of getting counterfactuals, the resulting counterfactual will be closer if not equivalent to the true causal effect.

4.4 Deterministic and stochastic counterfactual simulation and evaluation

4.4.1 MAPK model: Simulations and causal effect

Infer noise distribution with observation model (Algorithm 2 line 6) We use stochastic variational inference ([8]) to infer and update $N_{K3}$, $N_{K2}$ and $N_{K}$ from the observation model, and independent Normal distributions as approximating distributions.

Simulate from intervention model with updated noise (Algorithm 2 line 10) After updating the noise distributions, we generate the target distribution of the intervention model.

Deterministic and stochastic counterfactual simulation and evaluation (Algorithms 3 and 4). Fig. (4.2)(a) illustrates that the simulated trajectories converge in steady state. Since we relied on the Gaussian approximation to the Binomial in constructing $C$, we would expect worse results if we were to set the rates on or near the boundaries 0 and 100, where the approximation is weak. Fig. (4.2)(b) shows that for each set of rates, the causal effects from the SCM’s counterfactual distribution were centered around the ground truth simulated deterministically using Eq. (3.15) and corresponding equations for $K2$ and $K$. The SCM’s distribution had less variance, likely due to the posterior approximation step in the variational inference.

The data We simulated the counts of protein particles using the Markov process model with rate parameters $v_{k3}^{act}$, $v_{k2}^{act}$, $v_{k}^{act}$ and deactivation rate parameters $v_{k3}^{inh}$, $v_{k2}^{inh}$, $v_{k}^{inh}$. We conducted three simulation experiments with three sets of rates, all consistent with a low concentration in a cell-sized volume (see table 4.1). The initial conditions assumed 1 particle of E1, 100 particles of the unphosphorylated form of each protein, and 0 particles of the phosphorylated form.

The counterfactual of interest Let $K3$, $K2$ and $K$ denote the observed counts of phosphorylated MAP3K, MAP2K, and MAPK at 100 seconds, which corresponds to an steady-state for all the rates. Let $K3'$ be the count of phosphorylated MAP3K generated by 3 times smaller $v_{k3}^{act}$. Thus $v' = [v_{k3}^{act}/3, v_{k3}^{act}, v_{k2}^{act}, v_{k2}^{inh}, v_{K}^{act}, v_{K}^{inh}]$. We pose the counterfactual question: “Having observed the steady-state particle counts $K3$, $K2$ and $K$, what would have been the count of $K$ if we had $K3'$?”.
The evaluation We derive the SCM $C$ of the Markov process model and evaluate the counterfactual distribution $P_{C;K3=Z',K2=Y,K=Z,do(K3=Z')}$ where $Z'$ is the expected steady-state value associated with $v'$. We evaluate this counterfactual statement as described in Algorithms 3 and 4 (with 500 seeds). If the counterfactuals from the converted SCMs are consistent with the Markov process models, their histograms from Algorithms 3 and 4 should overlap.

Table 4.1: The rates parameters in MAPK model. Each row corresponds to a set $v$.

<table>
<thead>
<tr>
<th>Exp 1</th>
<th>v_{act3K}</th>
<th>MAP3K on</th>
<th>v_{inh3K}</th>
<th>MAP3K off</th>
<th>v_{act2K}</th>
<th>MAP2K on</th>
<th>v_{inh2K}</th>
<th>MAP2K off</th>
<th>v_{actK}</th>
<th>MAPK on</th>
<th>v_{inhK}</th>
<th>MAPK off</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exp 1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>2.0</td>
<td>0.1</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exp 2</td>
<td>0.2</td>
<td>0.3</td>
<td>0.2</td>
<td>3.0</td>
<td>0.2</td>
<td>1.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exp 3</td>
<td>0.1</td>
<td>0.3</td>
<td>0.5</td>
<td>5.0</td>
<td>0.3</td>
<td>4.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. (4.2)(a) illustrates that the simulated trajectories converge in steady state. Since we relied on the Gaussian approximation to the Binomial in constructing $C$, we would expect worse results if we were to set the rates on or near the boundaries 0 and 100, where the approximation is weak. Fig. (4.2)(b) shows that for each set of rates, the causal effects from the SCM’s counterfactual distribution were centered around the ground truth simulated deterministically using Eq. (3.15) and corresponding equations for $K2$ and $K$. The SCM’s distribution had less variance, likely due to the posterior approximation step in the variation inference.

4.4.2 IGF model: Simulations and causal effect

The data We simulated the counts of protein particles using the Markov process model with rates in Tables 2 and 3. The other settings are as in MAPK model. The initial condition assumed 37 particles of EGFR, 5 particles of IGFR, 100 particles of the unphosphorylated form of other proteins, and 0 particles of the phosphorylated form.

Table 4.2: The activation rates parameters in IGF model. Activation rates are formatted as $v_{act}$child-parent

<table>
<thead>
<tr>
<th>Rates</th>
<th>$v_{act}$SOS-EGFR</th>
<th>$v_{act}$SOS-IGFR</th>
<th>$v_{act}$Raf-SOS</th>
<th>$v_{act}$PI3K-EGFR</th>
<th>$v_{act}$PI3K-IGFR</th>
<th>$v_{act}$PI3K-Ras</th>
<th>$v_{act}$AKT-PI3K</th>
<th>$v_{act}$Raf-Ras</th>
<th>$v_{act}$Mek-Raf</th>
<th>$v_{act}$Erk-Mek</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rates</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.05</td>
<td>0.05</td>
</tr>
</tbody>
</table>
4.4. Deterministic and stochastic counterfactual simulation and evaluation

### Table 4.3: The deactivation rates parameters in IGF model. All but one rate are auto-deactivation. $v_{inh}$ formats as $v_{child-parent}^inh$.

<table>
<thead>
<tr>
<th>Rates</th>
<th>$v_{inh}$ SOS</th>
<th>$v_{inh}$ Ras</th>
<th>$v_{inh}$ PI3K</th>
<th>$v_{inh}$ AKT</th>
<th>$v_{inh}$ Raf-AKT</th>
<th>$v_{inh}$ Mek</th>
<th>$v_{inh}$ Erk</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.3</td>
<td>0.01</td>
<td>0.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>

The counterfactual of interest

Let $R'$ be the number of phosphorylated particles of Ras at steady-state, achieved with $v_{act}^{Ras-SOS} = v_{act}^{Ras-SOS}/6$. We pose the counterfactual: Having observed the number of phosphorylated particles of each protein before the intervention, what would be the number of particles of Erk if the intervention had fixed $Ras = R'$? Unlike the MAPK pathway, where the intervention on MAP3K affects the counterfactual target MAPK through a direct path, this system has two paths from Ras to Erk. One path goes directly through Raf, and the other through a mediating path PI3K $\rightarrow$ AKT. This challenges the algorithm to address multiple paths of influence.

The evaluation

We consider the rates $v_{act}^{Ras-SOS}/6$, the counterfactual distribution $P_{C;X_i=x_i;do(Ras=R')}$, and the Algorithms 3 and 4 (with 300 seeds).

Fig. (4.5)(b) plots deterministic and stochastic time courses for active states counts of the proteins in the pathway. Fig. (4.5)(c) illustrates that the counterfactual inference was successful despite the increased model complexity and size.

**Fig. 4.3:** IGF model (b) Deterministic and stochastic trajectories of the active-state proteins in the system. Horizontal lines are the expected values at steady-state. (c) Histogram of causal effects, defined as differences between the “observed” and the “counterfactual” trajectories of ERK at steady-state.

4.4.3 Counterfactual robustness under model misspecification

Fig. (4.4) shows that causal effects derived from misspecified model with true data is closer to the true causal effects, than causal effects from misspecified model conditioned with the misspecified data. For this experiment, true rate parameters were kept constant and misspecified parameters were randomly sampled from a uniform distribution (0.1,0.5).

4.4.4 Implications of Gaussian approximation in MAPK and IGF models

In the MAPK and IGF model, there are two levels of Gaussian approximation done for mathematical convenience and inference, which does not violate the findings. The first approximation is the approximation of Binomial to Gaussian, and the second is the replacement of Dirac-delta to Gaussians. As we saw in the inference part and the results, the second one adds variance but does not change the quantity we care that is the mean. The first approximation is...
problematic for simulating data with SCMs. The noise variables are sampled from standard normal distributions. If the sampled noise variable is above or below +3 or -3, SCM gives outliers. The CDF reparameterization, however, is safe to use for simulations. Hence, if the data generation is the main objective, then the Binomial to Gaussian approximation is not a good idea.

Here are the scatterplots with outliers comparing the data with SDE with the same initial conditions and rate parameters. It turns out that removing the outliers will give the same steady-state relationship between the proteins.

4.4.5 Code and run time

All the experiments were performed on a Macbook pro with intel 2.2 GHz core i7 and 16 GB RAM. The counterfactual inference (Algorithm 2) on the MAPK and growth factor models with one sample was computed in 4.89 seconds and 10.4 seconds with stochastic variational inference with 1000 steps. For the MAPK model, Algorithm 3 with trajectories of length 30 took 0.1 seconds, and Algorithm 4 with 1000 seeds took 1800 seconds. For the IGF model, the Algorithm 3 with trajectories of length 30 took 0.9 and Algorithm 4 with 300 seeds took 1245 seconds.

The code and results are available at https://github.com/kaushalpaneri/ode2scm.
4.5 Discussion

This work demonstrated the process of converting a mechanistic model to SCM, aiming to perform counterfactual inference at steady-state. We presented general principles and illustrated the specific steps in the case of real-life biochemical networks. Empirical validation shows that the counterfactual inference from the resulting SCM is rooted in the causal dynamic processes of the mechanistic model. We further demonstrated the implementation of these models in modern tensor-based probabilistic programming, showing that this approach can scale to relatively complex models.

Continuation of this work could further build on modeling and inference of experimental data, e.g., in the absence of a closed-form steady-state solutions. As SCMs and Markov process models both allow cycles, that is a logical avenue for future work. More investigation in models with higher-order hazard functions as well as investigating the impact of different types of intervention can also be the future directions. Moreover, combining the insight from this work with the advances in deep causal generative implicit models [11] can help to enrich these models with a more explicit mechanistic insight.
Bibliography


