# Part VI

Bayesian Solution of Ordinary Differential Equations

J. SKILLING, Cambridge University.

In: Maximum Entropy and Bayesian Methods, Springer Netherlands, 23-37, 1992.

> *In the numerical solution of ordinary differential equations, a function $y(x)$ is to be reconstructed from knowledge of the functional form of its derivative: $dy/dx = f(x, y)$, together with an appropriate boundary condition. The derivative $f$ is evaluated at a sequence of suitably chosen points $(x_k, y_k)$, from which the form of $y(\cdot)$ is estimated. This is an inference problem, which can and perhaps should be treated by Bayesian techniques. As always, the inference appears as a probability distribution prob$(y(\cdot))$, from which random samples show the probabilistic reliability of the results.*

Bayesian Solution of Ordinary Differential Equations

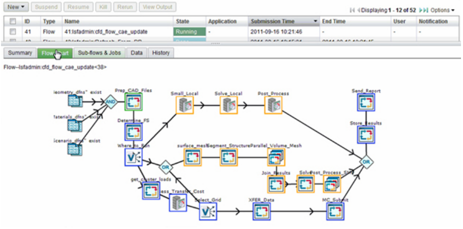
J. SKILLING, Cambridge University.

In: Maximum Entropy and Bayesian Methods, Springer Netherlands, 23-37, 1992.

> *In the numerical solution of ordinary differential equations, a function $y(x)$ is to be reconstructed from knowledge of the functional form of its derivative: $dy/dx = f(x, y)$, together with an appropriate boundary condition. The derivative $f$ is evaluated at a sequence of suitably chosen points $(x_k, y_k)$, from which the form of $y(\cdot)$ is estimated. This is an inference problem, which can and perhaps should be treated by Bayesian techniques. As always, the inference appears as a probability distribution prob$(y(\cdot))$, from which random samples show the probabilistic reliability of the results.*

# Twelfth Job: Introduction to Graphical Models

Numerical analysis for the "drag and drop" era of computational pipelines:



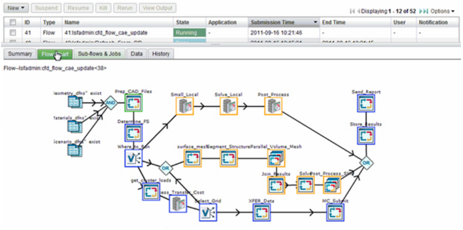

[Fig: IBM High Performance Computation]

The sophistication and scale of modern computer models creates an urgent need to better understand the propagation and accumulation of numerical error within arbitrary - often large - pipelines of computation, so that "numerical risk" to end-users can be controlled.

$\Longrightarrow$ Need to consider graphical representations of computation.

# Recall Our Motivation: Computational Pipelines

Numerical analysis for the "drag and drop" era of computational pipelines:




[Fig: IBM High Performance Computation]

The sophistication and scale of modern computer models creates an urgent need to better understand the propagation and accumulation of numerical error within arbitrary - often large - pipelines of computation, so that "numerical risk" to end-users can be controlled.

$\Longrightarrow$ Need to consider graphical representations of computation.

A probabilistic graphical model comprises of:

- A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with vertex set $\mathcal{V} = \{v_1, \dots, v_p\}$ and edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$.
- Semantics that associate the graph to statements about conditional (in)dependence of random variables $\{X_v\}_{v \in \mathcal{V}}$.

Recall: $X$ is C.I. of $Y$ given $Z \leftrightarrow X \perp\!\!\!\perp Y | Z \leftrightarrow p_{X,Y|Z}(x, y) = p_{X|Z}(x) p_{Y|Z}(y)$

Example: In a Gaussian graphical model:

- The random variables $\{X_v\}_{v \in \mathcal{V}}$ are jointly Gaussian.
- The edges are undirected (i.e. $(i, j) \in \mathcal{E}$ iff $(j, i) \in \mathcal{E}$).
- Two vertices $v_1$ and $v_2$ are **not** connected by an edge iff $X_{v_1} \perp\!\!\!\perp X_{v_2} | X_{\mathcal{V} \setminus \{v_1, v_2\}}$.

Fact: The edge structure of a Gaussian graphical model characterises the sparsity structure of the associated precision matrix.

A probabilistic graphical model comprises of:

- A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with vertex set $\mathcal{V} = \{v_1, \ldots, v_p\}$ and edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$.
- Semantics that associate the graph to statements about conditional (in)dependence of random variables $\{X_v\}_{v \in \mathcal{V}}$.

Recall: $X$ is C.I. of $Y$ given $Z \leftrightarrow X \perp\!\!\!\perp Y | Z \leftrightarrow p_{X,Y|Z}(x, y) = p_{X|Z}(x) p_{Y|Z}(y)$

Example: In a Gaussian graphical model:

- The random variables $\{X_v\}_{v \in \mathcal{V}}$ are jointly Gaussian.
- The edges are undirected (i.e. $(i, j) \in \mathcal{E}$ iff $(j, i) \in \mathcal{E}$).
- Two vertices $v_1$ and $v_2$ are **not** connected by an edge iff $X_{v_1} \perp\!\!\!\perp X_{v_2} | X_{\mathcal{V} \setminus \{v_1, v_2\}}$.

Fact: The edge structure of a Gaussian graphical model characterises the sparsity structure of the associated precision matrix.

A probabilistic graphical model comprises of:

- A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with vertex set $\mathcal{V} = \{v_1, \ldots, v_p\}$ and edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$.
- Semantics that associate the graph to statements about conditional (in)dependence of random variables $\{X_v\}_{v \in \mathcal{V}}$.

Recall: $X$ is C.I. of $Y$ given $Z \leftrightarrow X \perp\!\!\!\perp Y | Z \leftrightarrow p_{X,Y|Z}(x, y) = p_{X|Z}(x) p_{Y|Z}(y)$

Example: In a Gaussian graphical model:

- The random variables $\{X_v\}_{v \in \mathcal{V}}$ are jointly Gaussian.
- The edges are undirected (i.e. $(i, j) \in \mathcal{E}$ iff $(j, i) \in \mathcal{E}$).
- Two vertices $v_1$ and $v_2$ are **not** connected by an edge iff $X_{v_1} \perp\!\!\!\perp X_{v_2} | X_{\mathcal{V} \setminus \{v_1, v_2\}}$.

Fact: The edge structure of a Gaussian graphical model characterises the sparsity structure of the associated precision matrix.

A probabilistic graphical model comprises of:

- A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with vertex set $\mathcal{V} = \{v_1, \dots, v_p\}$ and edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$.
- Semantics that associate the graph to statements about conditional (in)dependence of random variables $\{X_v\}_{v \in \mathcal{V}}$.

Recall: $X$ is C.I. of $Y$ given $Z \leftrightarrow X \perp\!\!\!\perp Y|Z \leftrightarrow p_{X,Y|Z}(x,y) = p_{X|Z}(x)p_{Y|Z}(y)$

Example: In a Gaussian graphical model:

- The random variables $\{X_v\}_{v \in \mathcal{V}}$ are jointly Gaussian.
- The edges are undirected (i.e. $(i,j) \in \mathcal{E}$ iff $(j,i) \in \mathcal{E}$).
- Two vertices $v_1$ and $v_2$ are **not** connected by an edge iff $X_{v_1} \perp\!\!\!\perp X_{v_2} | X_{\mathcal{V} \setminus \{v_1, v_2\}}$.

Fact: The edge structure of a Gaussian graphical model characterises the sparsity structure of the associated precision matrix.

A probabilistic graphical model comprises of:

- A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with vertex set $\mathcal{V} = \{v_1, \dots, v_p\}$ and edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$.
- Semantics that associate the graph to statements about conditional (in)dependence of random variables $\{X_v\}_{v \in \mathcal{V}}$.

NB: More complicated graphical structures have also been developed.

Example: In a Gaussian graphical model:

- The random variables $\{X_v\}_{v \in \mathcal{V}}$ are jointly Gaussian.
- The edges are undirected (i.e. $(i,j) \in \mathcal{E}$ iff $(j,i) \in \mathcal{E}$).
- Two vertices $v_1$ and $v_2$ are **not** connected by an edge iff $X_{v_1} \perp\!\!\!\perp X_{v_2} | X_{\mathcal{V} \setminus \{v_1, v_2\}}$.

Fact: The edge structure of a Gaussian graphical model characterises the sparsity structure of the associated precision matrix.

A probabilistic graphical model comprises of:

- A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with vertex set $\mathcal{V} = \{v_1, \dots, v_p\}$ and edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$.
- Semantics that associate the graph to statements about conditional (in)dependence of random variables $\{X_v\}_{v \in \mathcal{V}}$.

NB: More complicated graphical structures have also been developed.

Example: In a Gaussian graphical model:

- The random variables $\{X_v\}_{v \in \mathcal{V}}$ are jointly Gaussian.
- The edges are undirected (i.e. $(i, j) \in \mathcal{E}$ iff $(j, i) \in \mathcal{E}$).
- Two vertices $v_1$ and $v_2$ are **not** connected by an edge iff $X_{v_1} \perp\!\!\!\perp X_{v_2} | X_{\mathcal{V} \setminus \{v_1, v_2\}}$.

Fact: The edge structure of a Gaussian graphical model characterises the sparsity structure of the associated precision matrix.

A probabilistic graphical model comprises of:

- A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with vertex set $\mathcal{V} = \{v_1, \dots, v_p\}$ and edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$.
- Semantics that associate the graph to statements about conditional (in)dependence of random variables $\{X_v\}_{v \in \mathcal{V}}$.

NB: More complicated graphical structures have also been developed.

Example: In a Gaussian graphical model:

- The random variables $\{X_v\}_{v \in \mathcal{V}}$ are jointly Gaussian.
- The edges are undirected (i.e. $(i,j) \in \mathcal{E}$ iff $(j,i) \in \mathcal{E}$).
- Two vertices $v_1$ and $v_2$ are **not** connected by an edge iff $X_{v_1} \perp\!\!\!\perp X_{v_2} | X_{\mathcal{V} \setminus \{v_1, v_2\}}$.

Fact: The edge structure of a Gaussian graphical model characterises the sparsity structure of the associated precision matrix.

Example: In a Bayesian network:

- The edges are directed (i.e. $(i,j) \in \mathcal{E} \implies (j,i) \notin \mathcal{E}$).
- No directed cycles exist (in particular, $(i,i) \notin \mathcal{E}$).
- Two sets $Q, R \subseteq \mathcal{V}$ are d-separated by a set $S \subseteq \mathcal{V} \setminus Q \cup R$ iff $X_Q \perp\!\!\!\perp X_R | X_S$. ("global Markov property")

Let $\text{an}_{\mathcal{G}}(S)$ denote the *ancestors* of the set $S$ in $\mathcal{G}$. (NB: This includes $S$ itself.)

Given a directed, acyclic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we say that two sets $Q, R \subseteq \mathcal{V}$ are d-separated by a set $S \subseteq \mathcal{V} \setminus Q \cup R$ iff the following holds:

- Form $\mathcal{G}'$, the subgraph induced by $\text{an}_{\mathcal{G}}(Q \cup R \cup S)$.
- Form $\mathcal{G}''$ by connecting any un-connected parents (with an undirected edge) in $\mathcal{G}'$.
- Remove all arrowheads in $\mathcal{G}''$, to obtain an undirected "moral" graph $\mathcal{G}_{\text{moral}}$.
- Check that all paths from vertices in $Q$ to vertices in $R$ in $\mathcal{G}_{\text{moral}}$ pass through a vertex in $S$.

Example: In a Bayesian network:

- The edges are directed (i.e. $(i,j) \in \mathcal{E} \implies (j,i) \notin \mathcal{E}$).
- No directed cycles exist (in particular, $(i,i) \notin \mathcal{E}$).
- Two sets $Q, R \subseteq \mathcal{V}$ are d-separated by a set $S \subseteq \mathcal{V} \setminus Q \cup R$ iff $X_Q \perp\!\!\!\perp X_R | X_S$. ("global Markov property")

Let $\mathrm{an}_{\mathcal{G}}(S)$ denote the *ancestors* of the set $S$ in $\mathcal{G}$. (NB: This includes $S$ itself.)

Given a directed, acyclic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we say that two sets $Q, R \subseteq \mathcal{V}$ are d-separated by a set $S \subseteq \mathcal{V} \setminus Q \cup R$ iff the following holds:

- Form $\mathcal{G}'$, the subgraph induced by $\mathrm{an}_{\mathcal{G}}(Q \cup R \cup S)$.
- Form $\mathcal{G}''$ by connecting any un-connected parents (with an undirected edge) in $\mathcal{G}'$.
- Remove all arrowheads in $\mathcal{G}''$, to obtain an undirected "moral" graph $\mathcal{G}_{\text{moral}}$.
- Check that all paths from vertices in $Q$ to vertices in $R$ in $\mathcal{G}_{\text{moral}}$ pass through a vertex in $S$.

Example: In a Bayesian network:

- The edges are directed (i.e. $(i,j) \in \mathcal{E} \implies (j,i) \notin \mathcal{E}$).
- No directed cycles exist (in particular, $(i,i) \notin \mathcal{E}$).
- Two sets $Q, R \subseteq \mathcal{V}$ are d-separated by a set $S \subseteq \mathcal{V} \setminus Q \cup R$ iff $X_Q \perp\!\!\!\perp X_R | X_S$. ("global Markov property")

Let $\text{an}_{\mathcal{G}}(S)$ denote the *ancestors* of the set $S$ in $\mathcal{G}$. (NB: This includes $S$ itself.)

Given a directed, acyclic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we say that two sets $Q, R \subseteq \mathcal{V}$ are d-separated by a set $S \subseteq \mathcal{V} \setminus Q \cup R$ iff the following holds:

- Form $\mathcal{G}'$, the subgraph induced by $\text{an}_{\mathcal{G}}(Q \cup R \cup S)$.
- Form $\mathcal{G}''$ by connecting any un-connected parents (with an undirected edge) in $\mathcal{G}'$.
- Remove all arrowheads in $\mathcal{G}''$, to obtain an undirected "moral" graph $\mathcal{G}_{\text{moral}}$.
- Check that all paths from vertices in $Q$ to vertices in $R$ in $\mathcal{G}_{\text{moral}}$ pass through a vertex in $S$.

Example: In a Bayesian network:

- The edges are directed (i.e. $(i,j) \in \mathcal{E} \implies (j,i) \notin \mathcal{E}$).
- No directed cycles exist (in particular, $(i,i) \notin \mathcal{E}$).
- Two sets $Q, R \subseteq \mathcal{V}$ are d-separated by a set $S \subseteq \mathcal{V} \setminus Q \cup R$ iff $X_Q \perp\!\!\!\perp X_R | X_S$. ("global Markov property")

Let $\text{an}_{\mathcal{G}}(S)$ denote the *ancestors* of the set $S$ in $\mathcal{G}$. (NB: This includes $S$ itself.)

Given a directed, acyclic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we say that two sets $Q, R \subseteq \mathcal{V}$ are d-separated by a set $S \subseteq \mathcal{V} \setminus Q \cup R$ iff the following holds:

- Form $\mathcal{G}'$, the subgraph induced by $\text{an}_{\mathcal{G}}(Q \cup R \cup S)$.
- Form $\mathcal{G}''$ by connecting any un-connected parents (with an undirected edge) in $\mathcal{G}'$.
- Remove all arrowheads in $\mathcal{G}''$, to obtain an undirected "moral" graph $\mathcal{G}_{\text{moral}}$.
- Check that all paths from vertices in $Q$ to vertices in $R$ in $\mathcal{G}_{\text{moral}}$ pass through a vertex in $S$.

Example: In a Bayesian network:

- The edges are directed (i.e. $(i,j) \in \mathcal{E} \implies (j,i) \notin \mathcal{E}$).
- No directed cycles exist (in particular, $(i,i) \notin \mathcal{E}$).
- Two sets $Q, R \subseteq \mathcal{V}$ are d-separated by a set $S \subseteq \mathcal{V} \setminus Q \cup R$ iff $X_Q \perp\!\!\!\perp X_R | X_S$. ("global Markov property")

Let $\text{an}_{\mathcal{G}}(S)$ denote the *ancestors* of the set $S$ in $\mathcal{G}$. (NB: This includes $S$ itself.)

Given a directed, acyclic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we say that two sets $Q, R \subseteq \mathcal{V}$ are d-separated by a set $S \subseteq \mathcal{V} \setminus Q \cup R$ iff the following holds:

- Form $\mathcal{G}'$, the subgraph induced by $\text{an}_{\mathcal{G}}(Q \cup R \cup S)$.
- Form $\mathcal{G}''$ by connecting any un-connected parents (with an undirected edge) in $\mathcal{G}'$.
- Remove all arrowheads in $\mathcal{G}''$, to obtain an undirected "moral" graph $\mathcal{G}_{\text{moral}}$.
- Check that all paths from vertices in $Q$ to vertices in $R$ in $\mathcal{G}_{\text{moral}}$ pass through a vertex in $S$.

Example: In a Bayesian network:

- The edges are directed (i.e. $(i,j) \in \mathcal{E} \implies (j,i) \notin \mathcal{E}$).
- No directed cycles exist (in particular, $(i,i) \notin \mathcal{E}$).
- Two sets $Q, R \subseteq \mathcal{V}$ are d-separated by a set $S \subseteq \mathcal{V} \setminus Q \cup R$ iff $X_Q \perp\!\!\!\perp X_R | X_S$. ("global Markov property")

Let $\mathrm{an}_{\mathcal{G}}(S)$ denote the *ancestors* of the set $S$ in $\mathcal{G}$. (NB: This includes $S$ itself.)

Given a directed, acyclic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we say that two sets $Q, R \subseteq \mathcal{V}$ are d-separated by a set $S \subseteq \mathcal{V} \setminus Q \cup R$ iff the following holds:

- Form $\mathcal{G}'$, the subgraph induced by $\mathrm{an}_{\mathcal{G}}(Q \cup R \cup S)$.
- Form $\mathcal{G}''$ by connecting any un-connected parents (with an undirected edge) in $\mathcal{G}'$.
- Remove all arrowheads in $\mathcal{G}''$, to obtain an undirected "moral" graph $\mathcal{G}_{\mathrm{moral}}$.
- Check that all paths from vertices in $Q$ to vertices in $R$ in $\mathcal{G}_{\mathrm{moral}}$ pass through a vertex in $S$.

# Bayesian Network

Example: In a Bayesian network:

- The edges are directed (i.e. $(i,j) \in \mathcal{E} \implies (j,i) \notin \mathcal{E}$).
- No directed cycles exist (in particular, $(i,i) \notin \mathcal{E}$).
- Two sets $Q, R \subseteq \mathcal{V}$ are d-separated by a set $S \subseteq \mathcal{V} \setminus Q \cup R$ iff $X_Q \perp\!\!\!\perp X_R | X_S$. ("global Markov property")

Let $\text{an}_{\mathcal{G}}(S)$ denote the *ancestors* of the set $S$ in $\mathcal{G}$. (NB: This includes $S$ itself.)

Given a directed, acyclic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we say that two sets $Q, R \subseteq \mathcal{V}$ are d-separated by a set $S \subseteq \mathcal{V} \setminus Q \cup R$ iff the following holds:

- Form $\mathcal{G}'$, the subgraph induced by $\text{an}_{\mathcal{G}}(Q \cup R \cup S)$.
- Form $\mathcal{G}''$ by connecting any un-connected parents (with an undirected edge) in $\mathcal{G}'$.
- Remove all arrowheads in $\mathcal{G}''$, to obtain an undirected "moral" graph $\mathcal{G}_{\text{moral}}$.
- Check that all paths from vertices in $Q$ to vertices in $R$ in $\mathcal{G}_{\text{moral}}$ pass through a vertex in $S$.

Example: In a Bayesian network:

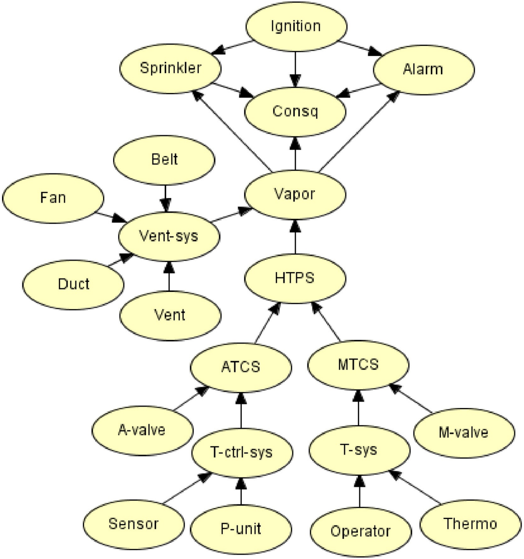
- The edges are directed (i.e. $(i,j) \in \mathcal{E} \implies (j,i) \notin \mathcal{E}$).
- No directed cycles exist (in particular, $(i,i) \notin \mathcal{E}$).
- Two sets $Q, R \subseteq \mathcal{V}$ are d-separated by a set $S \subseteq \mathcal{V} \setminus Q \cup R$ iff $X_Q \perp\!\!\!\perp X_R | X_S$. ("global Markov property")

Let $\mathrm{an}_{\mathcal{G}}(S)$ denote the *ancestors* of the set $S$ in $\mathcal{G}$. (NB: This includes $S$ itself.)

Given a directed, acyclic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we say that two sets $Q, R \subseteq \mathcal{V}$ are d-separated by a set $S \subseteq \mathcal{V} \setminus Q \cup R$ iff the following holds:

- Form $\mathcal{G}'$, the subgraph induced by $\mathrm{an}_{\mathcal{G}}(Q \cup R \cup S)$.
- Form $\mathcal{G}''$ by connecting any un-connected parents (with an undirected edge) in $\mathcal{G}'$.
- Remove all arrowheads in $\mathcal{G}''$, to obtain an undirected "moral" graph $\mathcal{G}_{\text{moral}}$.
- Check that all paths from vertices in $Q$ to vertices in $R$ in $\mathcal{G}_{\text{moral}}$ pass through a vertex in $S$.

# Exercise: Is Vent-sys d-separated from HTPS?



## Local Markov Property

For a Bayesian network;

$$X_v \perp\!\!\!\perp X_{\mathrm{Nd}(v)\setminus\mathrm{Pa}(v)} | X_{\mathrm{Pa}(v)}$$

where, according to the graph $\mathcal{G}$,

- $\mathrm{Nd}(v)$ are the non-descendants of $v \in \mathcal{V}$
- $\mathrm{Pa}(v)$ are the parents of $v \in \mathcal{V}$.

Message: "Only immediate parents matter".

Practically important, as it allows local verification of the global Markov property.

# Local Markov Property for Bayesian Networks

## Local Markov Property

For a Bayesian network;

$$X_v \perp\!\!\!\perp X_{\mathrm{Nd}(v)\setminus\mathrm{Pa}(v)} | X_{\mathrm{Pa}(v)}$$

where, according to the graph $\mathcal{G}$,

- $\mathrm{Nd}(v)$ are the non-descendants of $v \in \mathcal{V}$
- $\mathrm{Pa}(v)$ are the parents of $v \in \mathcal{V}$.

Message: "Only immediate parents matter".

Practically important, as it allows local verification of the global Markov property.

# Local Markov Property for Bayesian Networks

## Local Markov Property

For a Bayesian network;

$$X_v \perp\!\!\!\perp X_{\mathrm{Nd}(v) \setminus \mathrm{Pa}(v)} | X_{\mathrm{Pa}(v)}$$

where, according to the graph $\mathcal{G}$,

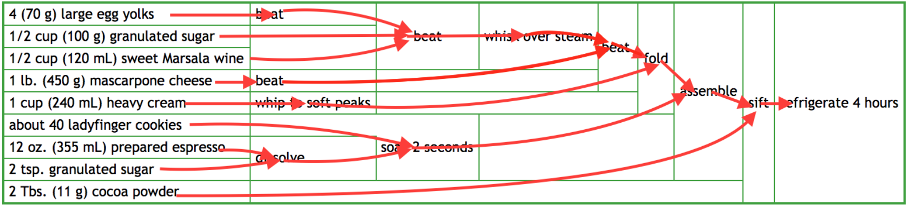
- $\mathrm{Nd}(v)$ are the non-descendants of $v \in \mathcal{V}$
- $\mathrm{Pa}(v)$ are the parents of $v \in \mathcal{V}$.

Message: "Only immediate parents matter".

Practically important, as it allows local verification of the global Markov property.

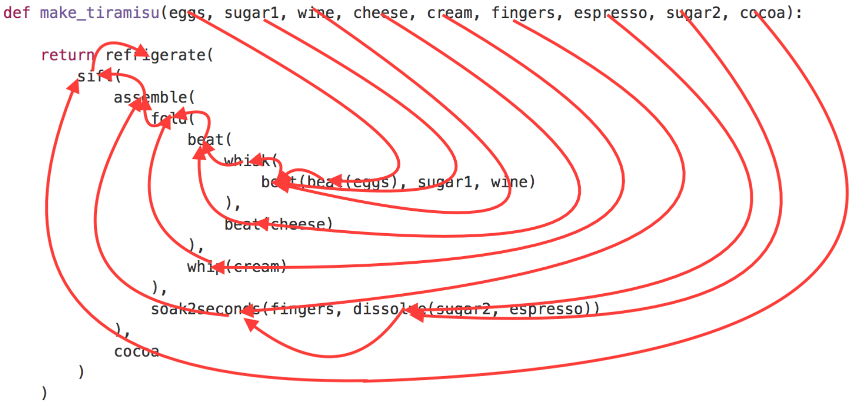
# Computational Work-Flows

Is there a connection to computation?



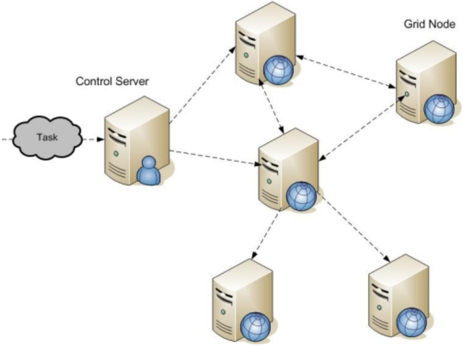

[Image from Li Haoyi's blog]

In particular, is there a connection to functional programming?




[Image from Li Haoyi's blog]

Or a connection to grid or cloud computing?



# Thirteenth Job: Pipelines of Computation

# Example: Split Integration

Consider estimation of the integral

$$\int_0^1 x(t)\mathrm{d}t$$

with a Bayesian probabilistic numerical method (Bayesian quadrature), based on the information $\{x(t_1), \ldots, x(t_{2m})\}$, where $t_1 = 0$, $t_m = 0.5$, $t_{2m} = 1$.

Under what circumstances can this computation be partitioned into two independent sub-computations?

$$\int_0^1 u(x)\mathrm{d}x = \int_0^{0.5} u(x)\mathrm{d}x + \int_{0.5}^1 u(x)\mathrm{d}x$$

Consider estimation of the integral

$$\int_0^1 x(t)\mathrm{d}t$$

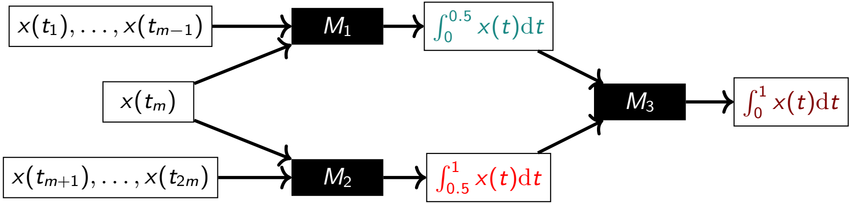
with a Bayesian probabilistic numerical method (Bayesian quadrature), based on the information $\{x(t_1), \dots, x(t_{2m})\}$, where $t_1 = 0$, $t_m = 0.5$, $t_{2m} = 1$.

Under what circumstances can this computation be partitioned into two independent sub-computations?

$$\int_0^1 u(x)\mathrm{d}x = \int_0^{0.5} u(x)\mathrm{d}x + \int_{0.5}^1 u(x)\mathrm{d}x$$

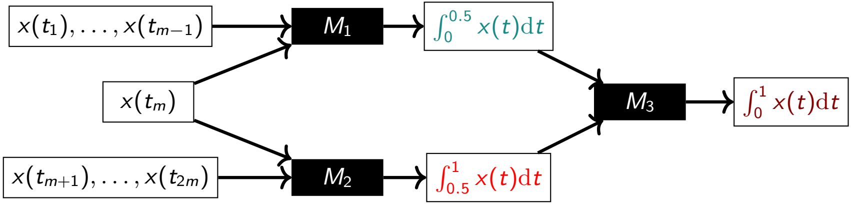
# Example: Split Integration

Let's attempt to represent this with a graphical model, which we will call a pipeline:




- Nodes are of two kinds: *Information* nodes □, and *method* nodes ■.
- The graph is bipartite, so that edges connect a method node to an information node or vice-versa. That is, edges are of the form □ → ■ or ■ → □.
- There are in general $n$ method nodes, each with a unique label in $\{1, \ldots, n\}$.
- The method node labelled $i$ has $m(i)$ parents and one child. Its in-edges are assigned a unique label in $\{1, \ldots, m(i)\}$.
- There is a unique terminal node and it is the child of method node $n$. This represents the principal quantity of interest $Q(x) = \int_0^1 x(t)\mathrm{d}t$.

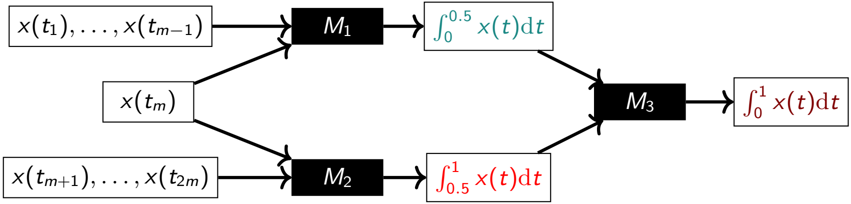
Let's attempt to represent this with a graphical model, which we will call a pipeline:




- Nodes are of two kinds: *Information* nodes □, and *method* nodes ■.
- The graph is bipartite, so that edges connect a method node to an information node or vice-versa. That is, edges are of the form □ → ■ or ■ → □.
- There are in general $n$ method nodes, each with a unique label in $\{1, \ldots, n\}$.
- The method node labelled $i$ has $m(i)$ parents and one child. Its in-edges are assigned a unique label in $\{1, \ldots, m(i)\}$.
- There is a unique terminal node and it is the child of method node $n$. This represents the principal quantity of interest $Q(x) = \int_0^1 x(t)\mathrm{d}t$.

# Example: Split Integration

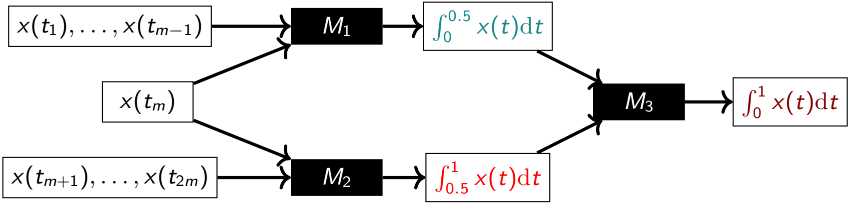
Let's attempt to represent this with a graphical model, which we will call a pipeline:




- Nodes are of two kinds: *Information* nodes □, and *method* nodes ■.
- The graph is bipartite, so that edges connect a method node to an information node or vice-versa. That is, edges are of the form □ → ■ or ■ → □.
- There are in general $n$ method nodes, each with a unique label in $\{1, \ldots, n\}$.
- The method node labelled $i$ has $m(i)$ parents and one child. Its in-edges are assigned a unique label in $\{1, \ldots, m(i)\}$.
- There is a unique terminal node and it is the child of method node $n$. This represents the principal quantity of interest $Q(x) = \int_0^1 x(t)\mathrm{d}t$.

# Example: Split Integration

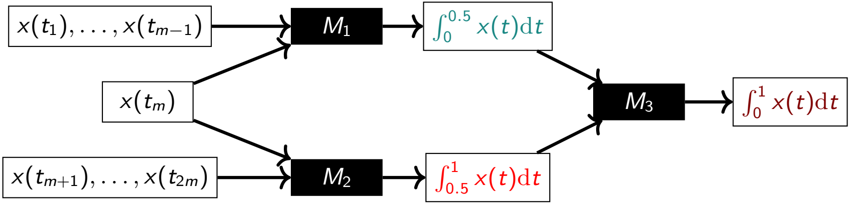
Let's attempt to represent this with a graphical model, which we will call a pipeline:




- Nodes are of two kinds: *Information* nodes $\square$, and *method* nodes $\blacksquare$.
- The graph is bipartite, so that edges connect a method node to an information node or vice-versa. That is, edges are of the form $\square \to \blacksquare$ or $\blacksquare \to \square$.
- There are in general $n$ method nodes, each with a unique label in $\{1, \ldots, n\}$.
- The method node labelled $i$ has $m(i)$ parents and one child. Its in-edges are assigned a unique label in $\{1, \ldots, m(i)\}$.
- There is a unique terminal node and it is the child of method node $n$. This represents the principal quantity of interest $Q(x) = \int_0^1 x(t)\mathrm{d}t$.

# Example: Split Integration

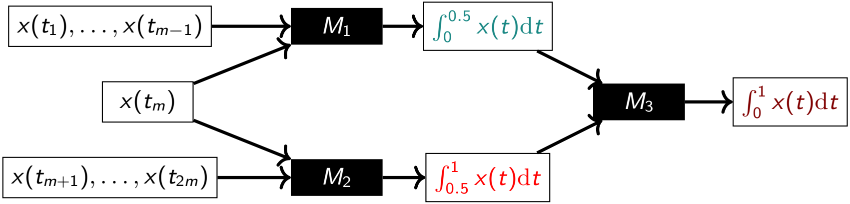
Let's attempt to represent this with a graphical model, which we will call a pipeline:




- Nodes are of two kinds: *Information* nodes $\square$, and *method* nodes $\blacksquare$.
- The graph is bipartite, so that edges connect a method node to an information node or vice-versa. That is, edges are of the form $\square \to \blacksquare$ or $\blacksquare \to \square$.
- There are in general $n$ method nodes, each with a unique label in $\{1, \dots, n\}$.
- The method node labelled $i$ has $m(i)$ parents and one child. Its in-edges are assigned a unique label in $\{1, \dots, m(i)\}$.
- There is a unique terminal node and it is the child of method node $n$. This represents the principal quantity of interest $Q(x) = \int_0^1 x(t)\mathrm{d}t$.

# Example: Split Integration

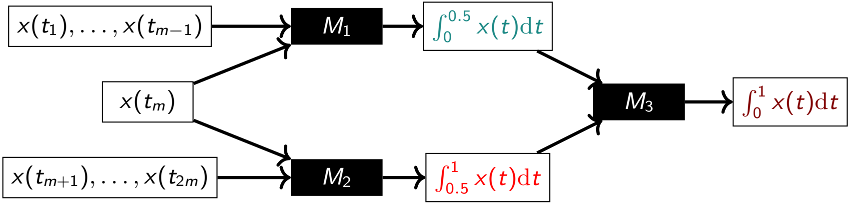
Let's attempt to represent this with a graphical model, which we will call a pipeline:




- Nodes are of two kinds: *Information* nodes $\square$, and *method* nodes $\blacksquare$.
- The graph is bipartite, so that edges connect a method node to an information node or vice-versa. That is, edges are of the form $\square \to \blacksquare$ or $\blacksquare \to \square$.
- There are in general $n$ method nodes, each with a unique label in $\{1, \ldots, n\}$.
- The method node labelled $i$ has $m(i)$ parents and one child. Its in-edges are assigned a unique label in $\{1, \ldots, m(i)\}$.
- There is a unique terminal node and it is the child of method node $n$. This represents the principal quantity of interest $Q(x) = \int_0^1 x(t)\mathrm{d}t$.

# Example: Split Integration

Let's attempt to represent this with a graphical model, which we will call a pipeline:

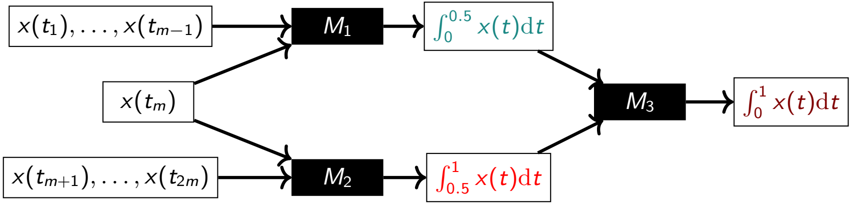



Let

- $M_1$ be Bayesian quadrature for $\int_0^{0.5} x(t)\mathrm{d}t$.
- $M_2$ be Bayesian quadrature for $\int_{0.5}^{1} x(t)\mathrm{d}t$.
- $M_3$ be the trivial probabilistic numerical method that adds its two inputs..

# Example: Split Integration

Let's attempt to represent this with a graphical model, which we will call a pipeline:

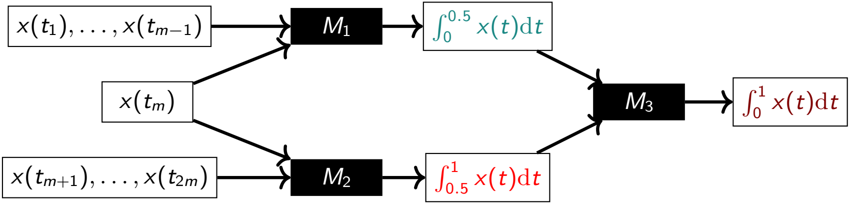



Q: When is the output of the pipeline Bayesian?

I.e. When does the output of the pipeline coincide with standard Bayesian quadrature performed on the full information $\{x(t_1), \ldots, x(t_{2m})\}$?

# Dependency Graphs

The abstract structure of the graph allows us to give a rigorous answer:



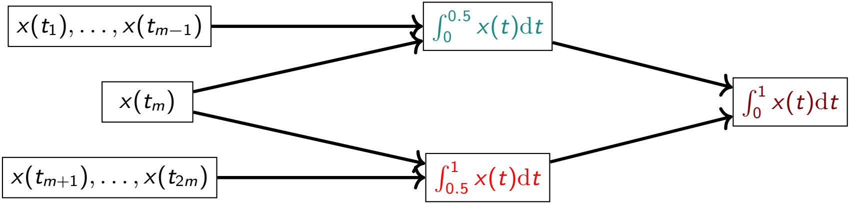

Pipeline

If we restrict attention to Bayesian probabilistic numerical methods, then $M_1$, $M_2$ and $M_3$ are uniquely determined by the prior distribution $P_x$ for the integrand.

So we can delete the method nodes to obtain the dependency graph of a pipeline.

Starting to look like a Bayesian network...

The abstract structure of the graph allows us to give a rigorous answer:



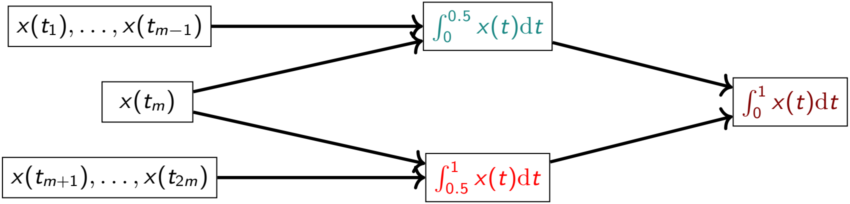

Dependency Graph

If we restrict attention to Bayesian probabilistic numerical methods, then $M_1$, $M_2$ and $M_3$ are uniquely determined by the prior distribution $P_x$ for the integrand.

So we can delete the method nodes to obtain the dependency graph of a pipeline.

Starting to look like a Bayesian network...

The abstract structure of the graph allows us to give a rigorous answer:



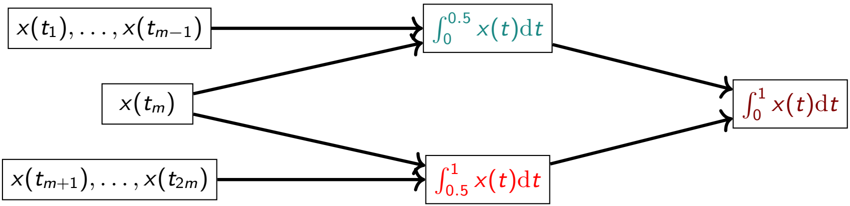

Dependency Graph

If we restrict attention to Bayesian probabilistic numerical methods, then $M_1$, $M_2$ and $M_3$ are uniquely determined by the prior distribution $P_x$ for the integrand.

So we can delete the method nodes to obtain the dependency graph of a pipeline.

Starting to look like a Bayesian network...

The abstract structure of the graph allows us to give a rigorous answer:

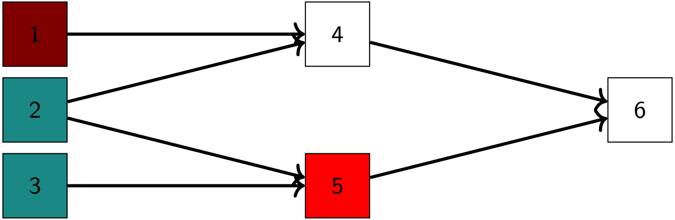



Dependency Graph

If we restrict attention to Bayesian probabilistic numerical methods, then $M_1$, $M_2$ and $M_3$ are uniquely determined by the prior distribution $P_x$ for the integrand.

So we can delete the method nodes to obtain the dependency graph of a pipeline.

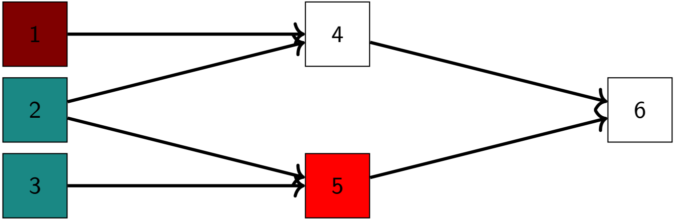
Starting to look like a Bayesian network...




Associate each node $i$ with a random variable $X_i$.

e.g. $X_1 = \{x(t_1), \ldots, x(t_{m-1})\}$, $X_4 = \int_0^{0.5} x(t) \mathrm{d}t$.

A prior $P_x$ is coherent for the dependency graph if each $X_i \perp\!\!\!\perp X_{\mathrm{Nd}(i) \setminus \mathrm{Pa}(i)} \mid X_{\mathrm{Pa}(i)}$.

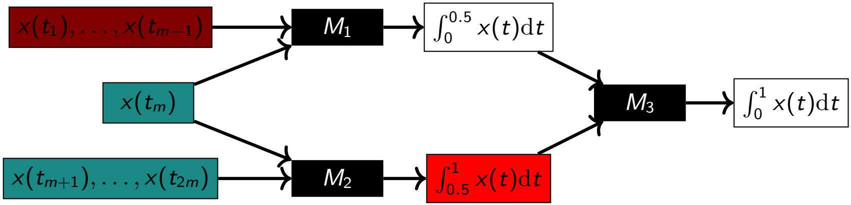



Associate each node $i$ with a random variable $X_i$.

e.g. $X_1 = \{x(t_1), \ldots, x(t_{m-1})\}$, $X_4 = \int_0^{0.5} x(t)\mathrm{d}t$.

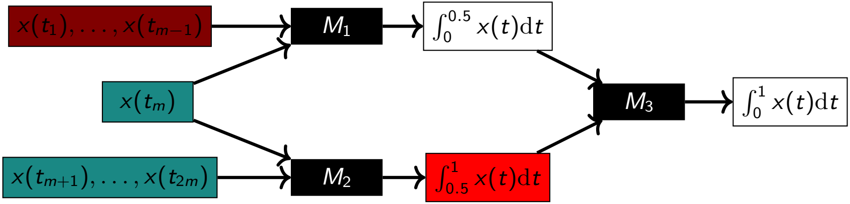
A prior $P_x$ is <u>coherent</u> for the dependency graph if each $X_i \perp\!\!\!\perp X_{\mathrm{Nd}(i) \setminus \mathrm{Pa}(i)} \mid X_{\mathrm{Pa}(i)}$.

# Split Integration: Coherence



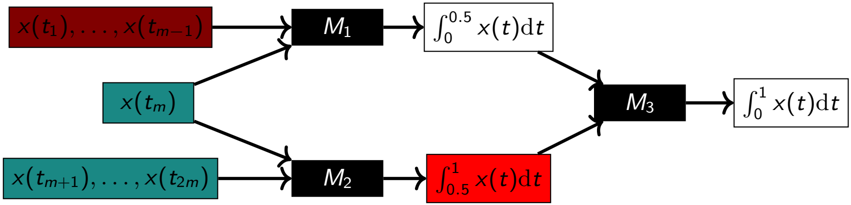

Thus in this instance we would ask whether $\int_{0.5}^{1} x(t)\mathrm{d}t$ is independent of $x(t_1), \ldots, x(t_{m-1})$ given $x(t_m), \ldots, x(t_{2m})$?

This is not true in general, but sometimes holds - e.g. a Wiener process prior $P_x$.




Thus in this instance we would ask whether $\int_{0.5}^{1} x(t)\mathrm{d}t$ is independent of $x(t_1), \dots, x(t_{m-1})$ given $x(t_m), \dots, x(t_{2m})$?

This is not true in general, but sometimes holds - e.g. a Wiener process prior $P_x$.




Thus in this instance we would ask whether $\int_{0.5}^1 x(t)\mathrm{d}t$ is independent of $x(t_1), \ldots, x(t_{m-1})$ given $x(t_m), \ldots, x(t_{2m})$?

This is <u>not</u> true in general, but sometimes holds - e.g. a Wiener process prior $P_x$.

The process illustrated here can be made formal:

A pipeline is Bayesian for estimation of its output if:

1. The prior $P_x$ is coherent for the dependency graph associated to the pipeline.
2. The methods $M_i$ are Bayesian probabilistic numerical methods.

**Open Question**: Can a similar notion of coherence be developed for non-Bayesian probabilistic numerical methods?

**Point for Discussion**: How to do split integration in $d \geq 2$ dimensions?

The process illustrated here can be made formal:

A pipeline is Bayesian for estimation of its output if:

1. The prior $P_x$ is coherent for the dependency graph associated to the pipeline.
2. The methods $M_i$ are Bayesian probabilistic numerical methods.

**Open Question**: Can a similar notion of coherence be developed for non-Bayesian probabilistic numerical methods?

**Point for Discussion**: How to do split integration in $d \geq 2$ dimensions?

The process illustrated here can be made formal:

A pipeline is Bayesian for estimation of its output if:

1. The prior $P_x$ is coherent for the dependency graph associated to the pipeline.
2. The methods $M_i$ are Bayesian probabilistic numerical methods.

**Open Question**: Can a similar notion of coherence be developed for non-Bayesian probabilistic numerical methods?

**Point for Discussion**: How to do split integration in $d \geq 2$ dimensions?

# Conclusion

In Part VI it has been argued that:

- Computational work-flow can be related to graphical models used in statistical applications.
- Bayesian probabilistic numerical methods induce a joint distribution over unknown objects, whose conditional (in)dependence structure can be represented with a pipeline graph.
- The local Markov property can be used to check whether a large pipeline of Bayesian probabilistic numerical methods is coherent.

END OF PART VI

# Conclusion

In Part VI it has been argued that:

- Computational work-flow can be related to graphical models used in statistical applications.
- Bayesian probabilistic numerical methods induce a joint distribution over unknown objects, whose conditional (in)dependence structure can be represented with a pipeline graph.
- The local Markov property can be used to check whether a large pipeline of Bayesian probabilistic numerical methods is coherent.

END OF PART VI

# Conclusion

In Part VI it has been argued that:

- Computational work-flow can be related to graphical models used in statistical applications.
- Bayesian probabilistic numerical methods induce a joint distribution over unknown objects, whose conditional (in)dependence structure can be represented with a pipeline graph.
- The local Markov property can be used to check whether a large pipeline of Bayesian probabilistic numerical methods is coherent.

END OF PART VI

# Conclusion

In Part VI it has been argued that:

- Computational work-flow can be related to graphical models used in statistical applications.
- Bayesian probabilistic numerical methods induce a joint distribution over unknown objects, whose conditional (in)dependence structure can be represented with a pipeline graph.
- The local Markov property can be used to check whether a large pipeline of Bayesian probabilistic numerical methods is coherent.

END OF PART VI

# Conclusion

In Part VI it has been argued that:

- Computational work-flow can be related to graphical models used in statistical applications.
- Bayesian probabilistic numerical methods induce a joint distribution over unknown objects, whose conditional (in)dependence structure can be represented with a pipeline graph.
- The local Markov property can be used to check whether a large pipeline of Bayesian probabilistic numerical methods is coherent.

END OF PART VI

# Minimal Take-Home Message

"All uncertainty is of one kind" $\sim$ Phil Dawid.

Download link: `oates.work/dobbiaco`

J. Chang and D. Pollard. Conditioning as disintegration. *Statistica Neerlandica*, 51(3):287–317, 1997.

I. Cialenco, G. E. Fasshauer, and Q. Ye. Approximation of stochastic partial differential equations by a kernel-based collocation method. *International Journal of Computer Mathematics*, 89 (18):2543–2561, 2012.

J. Cockayne, C. Oates, T. J. Sullivan, and M. Girolami. Probabilistic meshless methods for partial differential equations and Bayesian inverse problems, 2016. arXiv:1605.07811v1.

J. Cockayne, C. Oates, T. Sullivan, and M. Girolami. Bayesian probabilistic numerical methods. *arXiv:1702.03673*, 2017.

P. R. Conrad, M. Girolami, S. Särkkä, A. M. Stuart, and K. C. Zygalakis. Statistical analysis of differential equations: introducing probability measures on numerical solutions. *Statistics and Computing*, 2016.

S. L. Cotter, G. O. Roberts, A. M. Stuart, D. White, et al. Mcmc methods for functions: modifying old algorithms to make them faster. *Statistical Science*, 28(3):424–446, 2013.

T. Cui, J. Martin, Y. M. Marzouk, A. Solonen, and A. Spantini. Likelihood-informed dimension reduction for nonlinear inverse problems. *Inverse Problems*, 30(11):114015, 2014.

C. Dellacherie and P. Meyer. *Probabilities and Potential*. North-Holland, Amsterdam, 1978.

T. E. Hull and J. R. Swenson. Tests of probabilistic models for propagation of roundoff errors. *Communications of the ACM*, 9(2):108–113, 1966.

# References II

D. Isaacson, J. L. Mueller, J. C. Newell, and S. Siltanen. Reconstructions of chest phantoms by the d-bar method for electrical impedance tomography. *IEEE Transactions on Medical Imaging*, 23(7):821–828, 2004.

J. B. Kadane and G. W. Wasilkowski. *Bayesian Statistics*, chapter Average Case $\epsilon$-Complexity in Computer Science: A Bayesian View, pages 361–374. Elsevier, North-Holland, 1985.

A. N. Kolmogorov. *Foundations of Probability*. 1933.

C. Ley, G. Reinert, Y. Swan, et al. Stein's method for comparison of univariate distributions. *Probability Surveys*, 14:1–52, 2017.

C. Oates, S. Niederer, A. Lee, F. Briol, and M. Girolami. Probabilistic models for integration error in assessment of functional cardiac models. *arXiv:1606.06841*, 2017.

C. J. Oates, J. Cockayne, and R. G. Aykroyd. Bayesian probabilistic numerical methods for industrial process monitoring. *In preparation*, 2016a.

C. J. Oates, J. Cockayne, F.-X. Briol, and M. Girolami. Convergence rates for a class of estimators based on stein's identity. *arXiv:1603.03220*, 2016b.

H. Owhadi. Bayesian numerical homogenization. *Multiscale Modeling & Simulation*, 13(3): 812–828, 2015.

S. Särkkä. Linear operators and stochastic partial differential equations in gaussian process regression. *Artificial Neural Networks and Machine Learning–ICANN 2011*, pages 151–158, 2011.

A. M. Stuart. Inverse problems: A Bayesian perspective. *Acta Numerica*, 19:451–559, May 2010.

T. J. Sullivan. Well-posed Bayesian inverse problems and heavy-tailed stable quasi-Banach space priors, 2016. arXiv:1605.05898.

B. Szabó, A. van der Vaart, J. van Zanten, et al. Frequentist coverage of adaptive nonparametric bayesian credible sets. *The Annals of Statistics*, 43(4):1391–1428, 2015.

H. Wendland. *Scattered data approximation*, volume 17. Cambridge university press, 2004.